NXP Semiconductors
Data Sheet: Technical Data

Document Number: MM912_637D1
Rev. 5.0, 8/2016

Intelligent integrated precision battery sensor

MM912_637

Battery Monitoring System

EP SUFFIX (WF-TYPE)
98ASA00343D
48-PIN QFN

The MM912I637 (96 kB) and MM912J637 (128 kB) are fully integrated LIN Battery monitoring devices, based on NXP SMARTMOS and S12 MCU Technology.

The device supports precise current measurement via an external shunt resistor, and precise battery voltage measurement via a series resistor directly at the battery plus pole. The integrated temperature sensor combined in the close proximity to the battery, allows battery temperature measurement.

The integrated LIN 2.1 interface makes the sensor feedback available on the LIN Bus.

Features

- Battery voltage measurement
- Battery current measurement in up to 8 ranges
- On chip temperature measurement
- Normal and two low-power modes
- Current threshold detection and current averaging in standby => wake-up from low-power mode
- Triggered wake-up from LIN and periodic wake-up
- Signal low pass filtering (current, voltage)
- PGA (programmable low-noise gain amplifier) with automatic gain control
- Accurate internal oscillator (an external quartz oscillator may be used for extended accuracy)
- Communication via a LIN 2.1, LIN2.0 bus interface
- S12 microcontroller with 128 kByte flash, 6.0 kByte RAM, 4.0 kByte data flash
- Background debug module
- External temperature sensor option (TSUP, VTEMP)
- Optional 2nd external voltage sense input (VOPT)
- 4 x 5.0 V GPIO including one wake-up capable high voltage input (PTB3/L0)
- 8 x MCU general purpose I/O including SPI functionality
- Industry standard EMC compliance

Applications

- 12 V Lead-acid battery monitoring

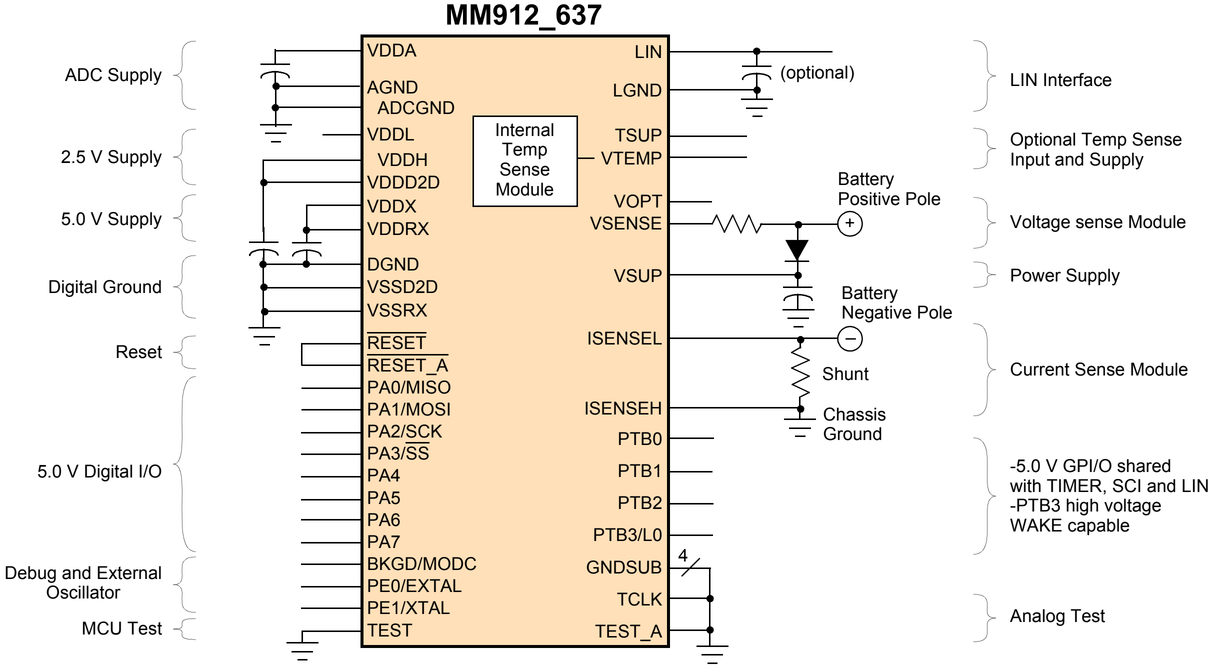


Figure 1. Simplified application diagram

Table of Contents

1 Ordering information

Table 1. Ordering information

Device [1]	Temperature range (T_A)	Package	Maximum input voltage	Analog option	Flash (kB)
MM912I637AM2EP	-40 °C to 125 °C	48 QFN-EP	42 V	2	96
MM912J637AM2EP					128
MM912I637AV1EP	-40 °C to 105 °C			1	96
MM912J637AV1EP					128

Notes

1. To order parts in tape & reel, add the R2 suffix to the part number.

Table 2. Analog options

Feature	Analog option 1	Analog option 2
Cranking Mode	Not Characterized or Tested	Fully Characterized and Tested
External Wake-up (PTB3/L0)	No	Yes
External Temperature Sensor Option (VTEMP)	No	Yes
Optional 2nd External Voltage Sense Input (VOPT)	No	Yes

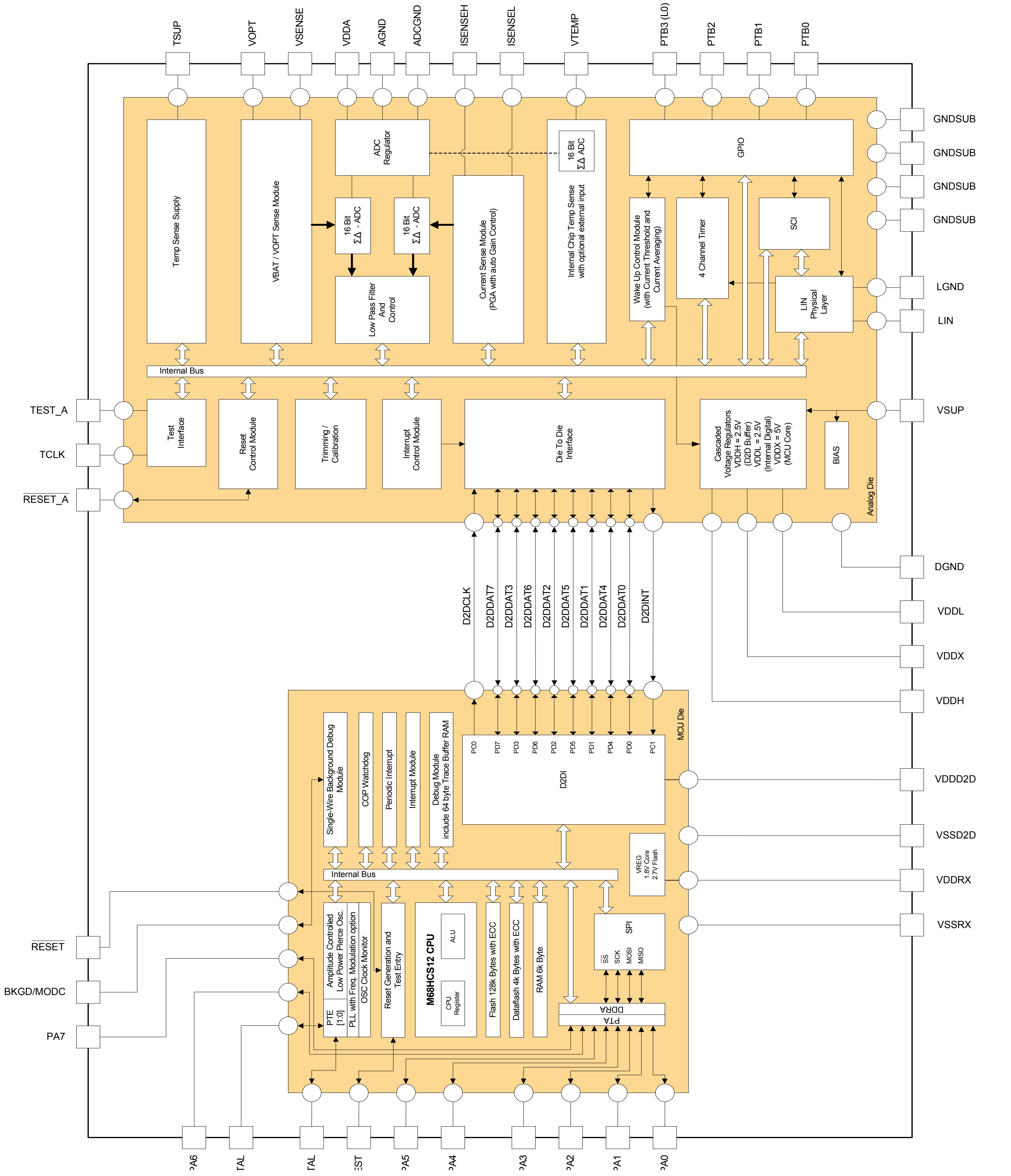


Figure 2. Sample block diagram

2 Pin assignment

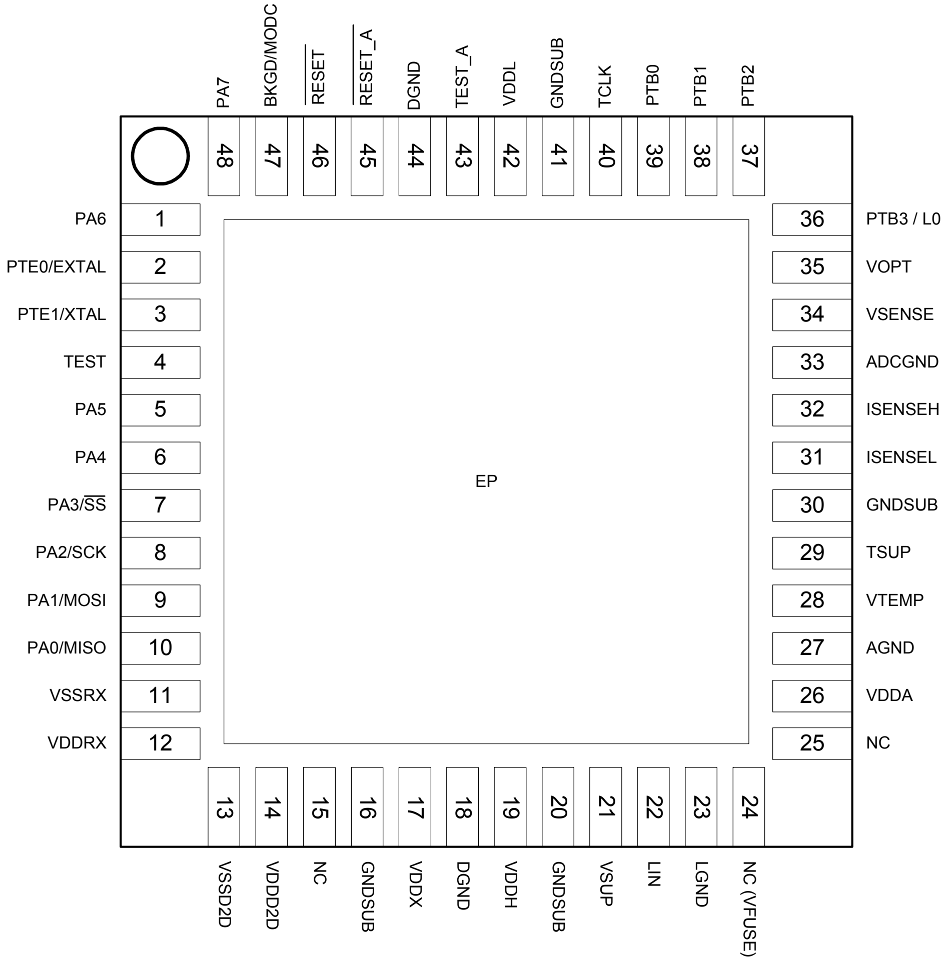


Figure 3. MM912_637 pin connections

2.1 MM912_637 pin description

The following table gives a brief description of all available pins on the MM912_637 device. Refer to the highlighted chapter for detailed information

Table 3. MM912_637 pin description (continued)

Pin #	Pin Name	Formal Name	Description
1	PA6	MCU PA6	General purpose port A input or output pin 6. See Section 4.17, "MCU - port integration module (9S12I128PIMV1)".
2	PE0/EXTAL	MCU Oscillator	EXTAL in one of the optional crystal/resonator drivers and external clock pins, and the PE0 port may be used as a general purpose I/O. On reset, all the device clocks are derived from the internal reference clock. See Section 4.23, "S12 clock, reset, and power management unit (9S12I128PIMV1)".
3	PE1/XTAL	MCU Oscillator	XTAL is one of the optional crystal/resonator drivers and external clock pins, and the PE1 port may be used as a general purpose I/O. On reset all the device clocks are derived from the internal reference clock. See Section 4.23, "S12 clock, reset, and power management unit (9S12I128PIMV1)".

Table 3. MM912_637 pin description (continued)

Pin #	Pin Name	Formal Name	Description
4	TEST	MCU Test	This input only pin is reserved for test. This pin has a pull-down device. The TEST pin must be tied to VSSRX in user mode.
5	PA5	MCU PA5	General purpose port A input or output pin 5. See Section 4.17, “MCU - port integration module (9S12I128PIMV1)".
6	PA4	MCU PA4	General purpose port A input or output pin 4. See Section 4.17, “MCU - port integration module (9S12I128PIMV1)".
7	PA3	MCU PA3 / $\overline{SS}$	General purpose port A input or output pin 3, shared with the $\overline{SS}$ signal of the integrated SPI interface. See Section 4.17, “MCU - port integration module (9S12I128PIMV1)".
8	PA2	MCU PA2 / SCK	General purpose port A input or output pin 2, shared with the SCLK signal of the integrated SPI interface. See Section 4.17, “MCU - port integration module (9S12I128PIMV1)".
9	PA1	MCU PA1 / MOSI	General purpose port A input or output pin 1, shared with the MOSI signal of the integrated SPI interface. See Section 4.17, “MCU - port integration module (9S12I128PIMV1)".
10	PA0	MCU PA0 / MISO	General purpose port A input or output pin 0, shared with the MISO signal of the integrated SPI interface. See Section 4.17, “MCU - port integration module (9S12I128PIMV1)".
11	VSSRX	MCU 5.0 V Ground	External ground for the MCU - VDDRX return path.
12	VDDRX	MCU 5.0 V Supply	5.0 V MCU power supply. MCU core- (internal 1.8 V regulator) and flash (internal 2.7 V regulator) supply. This pin must be connected to VDDX.
13	VSSD2D	MCU 2.5 V Ground	External ground for the MCU - VDDD2D return path.
14	VDDD2D	MCU 2.5 V Supply	2.5 V MCU power supply. Die to die buffer supply. This pin must be connected to VDDH.
15	NC	Not connected	This pin must be grounded in the application.
16	GNDSUB	Substrate Ground	Substrate ground connection to improve EMC behavior.
17	VDDX	Voltage Regulator Output 5.0 V	5.0 V main voltage regulator output pin. An external capacitor (C_{VDDX}) is needed. See Section 4.3, “Analog die - power, clock and resets - PCR".
18	DGND	Digital Ground	This pin is the device digital ground connection. See Section 4.3, “Analog die - power, clock and resets - PCR".
19	VDDH	Voltage Regulator Output 2.5 V	2.5 V high power main voltage regulator output pin to be connected with the VDDD2D MCU pin. An external capacitor (C_{VDDH}) is needed. See Section 4.3, “Analog die - power, clock and resets - PCR".
20	GNDSUB	Substrate Ground	Substrate ground connection to improve EMC behavior.
21	VSUP	Power Supply	This pin is the device power supply pin. A reverse battery protection diode is required. See Section 4.3, “Analog die - power, clock and resets - PCR".
22	LIN	LIN Bus I/O	This pin represents the single-wire bus transmitter and receiver. See Section 4.12, “LIN".
23	LGND	LIN Ground Pin	This pin is the device LIN ground connection. See Section 4.3, “Analog die - power, clock and resets - PCR".
24	NC	Not connected (reserved)	This pin must be grounded in the application.
25	NC	Not connected	This pin must be grounded in the application.
26	VDDA	Analog Voltage Regulator Output	Low power analog voltage regulator output pin, permanently supplies the analog front end. An external capacitor (C_{VDDA}) is needed. See Section 4.3, “Analog die - power, clock and resets - PCR".
27	AGND	Analog Ground	This pin is the device analog voltage regulator and LP oscillator ground connection. See Section 4.3, “Analog die - power, clock and resets - PCR".
28	VTEMP	Temperature Sensor Input	External temperature sensor input. See Section 4.7, “Temperature measurement - TSENSE".
29	TSUP	Temperature Sensor Supply Output	Supply for the external temperature sensor. TSUP frequency compensation option to allow capacitor CTSUP. See Section 4.7, “Temperature measurement - TSENSE".
30	GNDSUB	Substrate Ground	Substrate ground connection to improve EMC behavior.

Table 3. MM912_637 pin description (continued)

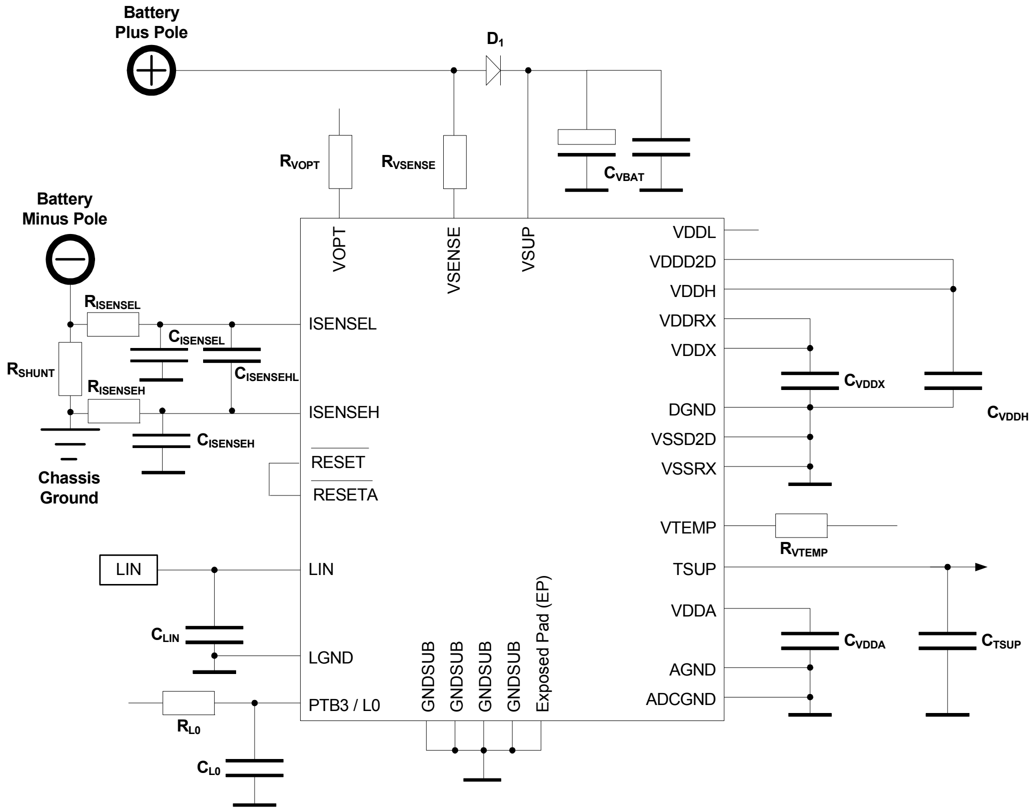
Pin #	Pin Name	Formal Name	Description
31	ISENSEL	Current Sense L	Current sense input “Low”. This pin is used in combination with ISENSEH to measure the voltage drop across a shunt resistor. See Section 4.5, “Current measurement - ISENSE".
32	ISENSEH	Current Sense H	Current sense input “high”. This pin is used in combination with ISENSEL to measure the voltage drop across a shunt resistor. See Section 4.5, “Current measurement - ISENSE".
33	ADCGND	Analog Digital Converter Ground	Analog digital converter ground connection. See Section 4.3, “Analog die - power, clock and resets - PCR".
34	VSENSE	Voltage Sense	Precision battery voltage measurement input. This pin can be connected directly to the battery line for voltage measurements. The voltage preset at this input is scaled down by an internal voltage divider. The pin is self protected against reverse battery connections. An external resistor (R_{VSENSE}) is needed for protection. See Section 4.6, “Voltage measurement - VSENSE".
35	VOPT	Optional Voltage Sense	Optional voltage measurement input. See Section 4.6, “Voltage measurement - VSENSE".
36	PTB3 / L0	General Purpose Input 3 - High Voltage Input 0	This is the high voltage general purpose input pin 3, based on VDDX with the following shared functions: • Internal clamping structure to operate as a high voltage input (L0). When used as high voltage input, a series resistor (R_{L0}) and capacitor to GND (C_{L0}) must be used to protect against automotive transients, when used to connect outside the PCB. • 5.0 V (VDDX) digital port input • Selectable internal pull-down resistor • Selectable wake-up input during low power mode. • Selectable timer channel input • Selectable connection to the LIN / SCI (Input only) See Section 4.11, “General purpose I/O - GPIO".
37	PTB2	General Purpose I/O 2	This is the general purpose I/O pin 2 based on VDDX with the following shared functions: • Bidirectional 5.0 V (VDDX) digital port I/O • Selectable internal pull-up resistor • Selectable timer channel input/output • Selectable connection to the LIN / SCI See Section 4.11, “General purpose I/O - GPIO".
38	PTB1	General Purpose I/O 1	This is the general purpose I/O pin 1, based on VDDX with the following shared functions: • Bidirectional 5.0 V (VDDX) digital port I/O • Selectable internal pull-up resistor • Selectable timer channel input/output • Selectable connection to the LIN / SCI See Section 4.11, “General purpose I/O - GPIO".
39	PTB0	General Purpose I/O 0	This is the general purpose I/O pin 0 based on VDDX with the following shared functions: • Bidirectional 5.0 V (VDDX) digital port I/O • Selectable internal pull-up resistor • Selectable timer channel input/output • Selectable connection to the LIN / SCI See Section 4.11, “General purpose I/O - GPIO".
40	TCLK	Test Clock Input	Test mode clock input pin for Test mode only. This pin must be grounded in user mode.
41	GNDSUB	Substrate Ground	Substrate ground connection to improve EMC behavior.
42	VDDL	Low Power Voltage Regulator Output	2.5 V low power voltage regulator output pin. See Section 4.3, “Analog die - power, clock and resets - PCR".
43	TEST_A	Test Mode	Analog die Test mode pin for Test mode only. This pin must be grounded in user mode.
44	DGND	Digital Ground	This pin is the device digital ground connection. See Section 4.3, “Analog die - power, clock and resets - PCR".
45	$\overline{\text{RESET\_A}}$	Reset I/O	Reset output pin of the analog die in Normal mode. Bidirectional reset I/O of the analog die in Stop mode. Active low signal with internal pull-up to VDDX. This pin must be connected to RESET. See Section 4.3, “Analog die - power, clock and resets - PCR".

Table 3. MM912_637 pin description (continued)

Pin #	Pin Name	Formal Name	Description
46	$\overline{\text{RESET}}$	MCU Reset	Bidirectional reset I/O pin of the MCU die. Active low signal with internal pull-up to VDDRX. This pin must be connected to RESETA. See Section 4.3, "Analog die - power, clock and resets - PCR".
47	BKGD	MCU Background Debug and Mode	The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as an MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has a pull-up device. See Section 4.20, "MCU - debug module (9S12I128PIMV1)".
48	PA7	MCU PA7	General purpose port A input or output pin 7. See Section 4.17, "MCU - port integration module (9S12I128PIMV1)".

2.2 Recommended external components

Figure 4 and Table 4 list the required / recommended / optional external components for the application.



Note: Module GND connected to Battery Minus or Chassis Ground – based on configuration.

Figure 4. Required/recommended external components

Table 4. Required/recommended external components

Name	Description	Value	Connection	Comment
D_1	Reverse Battery Diode	n.a.	VSUP-VBAT	
C_{VBAT}	Battery Blocking Capacitor	4.7 µF/100 nF	VSUP-GND	Ceramic
R_{VSENSE}	VSENSE Current Limitation	2.2 kΩ	VSENSE-VBAT	
R_{VOPT}	VOPT Current Limitation	2.2 kΩ	VOPT-signal	optional[2]

Table 4. Required/recommended external components

Name	Description	Value	Connection	Comment
R_{SHUNT}	Current Shunt Resistor	100 μΩ	ISENSEH-ISENSEL	
$R_{ISENSEL}$	EMC Resistor	500 Ω max		select for best EMC performance
$R_{ISENSEH}$	EMC Resistor	500 Ω max		select for best EMC performance
$C_{ISENSEL}$	EMC Capacitor	TBD		select for best EMC performance
$C_{ISENSEHL}$	EMC Capacitor	TBD		select for best EMC performance
$C_{ISENSEH}$	EMC Capacitor	TBD		select for best EMC performance
C_{VDDH}	Blocking Capacitor	1.0 μF	VDDH-GND	
C_{VDDX}	Blocking Capacitor	220 nF	VDDX-GND	
C_{VDDA}	Blocking Capacitor	47 nF	VDDA-GND	
C_{VDDL}	Blocking Capacitor	n.a.	VDDL-GND	not required
C_{LIN}	LIN Bus Filter	n.a.	LIN-LGND	not required
R_{L0}	PTB3 / L0 Current Limitation	47 kΩ	L0	
C_{L0}	PTB3 / L0 ESD Protection	47 nF	L0-GND	
C_{TSUP}	Blocking Capacitor	220 pF	TSUP-GND	not required(3)
R_{VTEMP}	VTEMP Current Limitation	20 kΩ	VTEMP-signal	optional(2)

Notes

2. Required if extended EMC protection is needed
3. If an external temperature sensor is used, EMC compliance may require the addition of CTSUP. In this case the ECAP bit must be set to ensure the stability of the TSUP power supply circuit. See Section 4.7.1.2, “Block diagram".

2.3 Pin structure

Table 5 documents the individual pin characteristic.

Table 5. Pin type/structure

Pin #	Pin name	Alternative pin function	Power supply	Structure
1	PA6	n.a.	VDDRX	n.a.
2	PE0	EXTAL	VDDRX	PUPEE / OSCPINS_EN
3	PE1	XTAL	VDDRX	PUPEE / OSCPINS_EN
4	TEST	n.a.	n.a.	n.a.
5	PA5	n.a.	VDDRX	n.a.
6	PA4	n.a.	VDDRX	n.a.
7	PA3	SS	VDDRX	n.a.
8	PA2	SCK	VDDRX	n.a.
9	PA1	MOSI	VDDRX	n.a.
10	PA0	MISO	VDDRX	n.a.
11	VSSRX	n.a.		GND
12	VDDRX	n.a.		
13	VSSD2D	n.a.		GND

Table 5. Pin type/structure

Pin #	Pin name	Alternative pin function	Power supply	Structure
14	VDDD2D	n.a.		
15	NC	n.a.		
16	GNDSUB	n.a.		GND
17	VDDX	n.a.	VDDX	
18	DGND	n.a.	GND	B2B-Diode to GNDSUB
19	VDDH	n.a.	VDDH	Negative Clamp Diode, Dynamic ESD (transient protection)
20	GNDSUB	n.a.	GND	GNDSUB
21	VSUP	n.a.	VSUP	Negative Clamp Diode, >42 V ESD
22	LIN	n.a.	VSUP	No Negative Clamping Diode (-40 V), >42 V ESD
23	LGND	n.a.	GND	B2B-Diode to GNDSUB
24	NC	n.a.	n.a.	Negative Clamp Diode, >15 V ESD
25	NC	n.a.	n.a.	n.a.
26	VDDA	n.a.	VDDA	Negative Clamp Diode, Dynamic ESD (transient protection)
27	AGND	n.a.	GND	B2B-Diode to GNDSUB
28	VTEMP		VDDA	Negative Clamp Diode, >6.0 V ESD
29	TSUP		TSUP	Negative Clamp Diode, Dynamic ESD (transient protection)
30	GNDSUB		GND	GND
31	ISENSEL		n.a.	Negative Clamp Diode, 2nd Clamp Diode to VDDA
32	ISENSEH		n.a.	Negative Clamp Diode, 2nd Clamp Diode to VDDA
33	ADCGND		GND	B2B-Diode to GNDSUB
34	VSENSE		n.a.	No Negative Clamping Diode (-40 V), >42 V ESD
35	VOPT		n.a.	No Negative Clamping Diode (-40 V), >42 V ESD
36	PTB3 / L0		VDDRX	Negative Clamp Diode, >6.0 V ESD
37	PTB2		VDDRX	Negative Clamp, Dynamic 5.5 V ESD
38	PTB1		VDDRX	Negative Clamp, Dynamic 5.5 V ESD
39	PTB0		VDDRX	Negative Clamp, Dynamic 5.5 V ESD
40	TCLK		VDDRX	Negative Clamp, Dynamic 5.5 V ESD
41	GNDSUB		GND	GND
42	VDDL		VDDL	Negative Clamp Diode, Dynamic ESD (transient protection)
43	TEST_A		VDDRX	Negative Clamp, positive 10 V Clamp
44	DGND		GND	B2B-Diode to GNDSUB
45	$\overline{\text{RESET\_A}}$		VDDRX	Negative Clamp, positive 10 V Clamp
46	$\overline{\text{RESET}}$		VDDRX	Pull-up

Table 5. Pin type/structure

Pin #	Pin name	Alternative pin function	Power supply	Structure
47	BKGD	MODC	VDDRX	BKPUE
48	PA7		VDDRX	n.a.

3 Electrical characteristics

3.1 General

This section contains electrical information for the microcontroller, as well as the MM912_637 analog die.

3.2 Absolute maximum ratings

Absolute maximum ratings are stress ratings only. A functional operation under or outside these maximums is not guaranteed. Stress beyond these limits may affect the reliability, or cause permanent damage of the device.

This device contains circuitry protecting against damage due to high static voltage or electrical fields. However, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate voltage level. All voltages are with respect to ground, unless otherwise noted.

Table 6. Absolute maximum electrical ratings - analog die

Ratings	Symbol	Value	Unit
VSUP pin voltage	V_{VSUP}	-0.3 to 42	V
VSENSE pin voltage with 2.2 k serial resistor [4]	V_{VSENSE}	-16 to 42	V
VOPT pin voltage with 2.2 k serial resistor	V_{VOPT}	-16 to 42	V
VTEMP pin voltage	V_{VTEMP}	-0.3 to V_{DDA}+0.25	V
ISENSEH and ISENSEL pin voltage	V_{ISENSE}	-0.5 to V_{DDA}+0.25	V
ISENSEH and ISENSEL pin current	I_{ISENSE}	-1 to 1	mA
LIN pin voltage	V_{BUS}	-33 to 42	V
LIN pin current (internally limited)	I_{BUSLIM}	on page 17	mA
L0 pin voltage with R_{PTB3}	V_{PTB3}	-0.3 to 42 max.	V
Input / Output pins PTB[0:2] voltage	V_{PTB0-2}	-0.3 to V_{DDX}+0.5	V
Pin voltage at VDDX	V_{DDX}	-0.3 to 5.75	V
Pin voltage at VDDH	V_{DDH}	-0.3 to 2.75	V
VDDH output current	I_{VDDH}	internally limited	A
VDDX output current	I_{VDDX}	internally limited	A
TCLK pin voltage	V_{TCLK}	-0.3 to V_{DDX}+0.5	V
$\overline{\text{RESET\_A}}$ pin voltage	V_{IN}	-0.3 to V_{DDX}+0.5	V

Notes

[4] It has to be assured by the application circuit that these limits will not be exceeded, e.g. by ISO pulse 1.

Table 7. Maximum electrical ratings - MCU die (continued)

Ratings	Symbol	Value	Unit
5.0 V supply voltage	V_{DDRX}	-0.3 to 6.0	V
2.5 V supply voltage	V_{DDD2D}	-0.3 to 3.6	V
Digital I/O input voltage (PTA0...7)	V_{IN}	-0.3 to 6.0	V
EXTAL, XTAL	V_{IN}	-0.3 to 2.16	V
Instantaneous maximum current single pin limit for all digital I/O pins[5]	I_D	-25 to 25	mA

Table 7. Maximum electrical ratings - MCU die (continued)

Ratings	Symbol	Value	Unit
Instantaneous maximum current single pin limit for EXTAL, XTAL	I_{DL}	-25 to 25	mA

Notes

5. All digital I/O pins are internally clamped to V_{SSRX} and V_{DDRX}.

Table 8. Maximum thermal ratings

Ratings	Symbol	Value	Unit
Storage temperature	T_{STG}	-55 to 150	°C
Package thermal resistance [6]	$R_{\theta JA}$	25 typ.	°C/W

Notes

6. $R_{\theta JA}$ value is derived using a JEDEC 2s2p test board

3.3 Operating conditions

This section describes the operating conditions of the device. Conditions apply to all the following data, unless otherwise noted.

Table 9. Operating conditions [7]

Ratings	Symbol	Value	Unit
Functional operating supply voltage - Device is fully functional. All features are operating.	V_{SUP}	3.5 to 28	V
Extended range for RAM Content is guaranteed. Other device functionary is limited. With cranking mode enabled (seeSection 4.3.3.4, "Low-voltage operation - cranking mode device option").	V_{SUPL}	2.5 to 3.5	V
Functional operating VSENSE voltage [8]	V_{SENSE}	0 to 28	V
Functional operating VOPT voltage	V_{OPT}	0 to 28	V
External temperature sense input - VTEMP	V_{TEMP}	0 to 1.25	V
LIN output voltage range	V_{VSUP_LIN}	7.0 to 18	V
ISENSEH / ISENSEL terminal voltage	V_{ISENSE}	-0.3 to 0.3	V
MCU 5.0 V supply voltage	V_{DDRX}	3.13 to 5.5	V
MCU 2.5 V supply voltage	V_{DDD2D}	2.25 to 3.6	V
MCU oscillator	f_{OSC}	4.0 to 16	MHz
MCU bus frequency	f_{BUS}	max. 32.768	MHz
Operating ambient temperature	T_A	-40 to 125	°C
Operating junction temperature - analog die	T_{J_A}	-40 to 150	°C
Operating junction temperature - MCU die	T_{J_M}	-40 to 150	°C

Notes

7. The parametric data are guaranteed while the pins are within Operating Conditions. Other conditions are presented at the top of the parametric tables or noted into parameters.
8. Values V_{SENSE} > 28 V are flagged in the VSENSE

3.4 Supply currents

This section describes the current consumption characteristics of the device, as well as the conditions for the measurements.

3.4.1 Measurement conditions

All measurements are without output loads. The currents are measured in MCU special single chip mode, and the CPU code is executed from RAM, unless otherwise noted.

For Run and Wait current measurements, PLL is on and the reference clock is the IRC1M, trimmed to 1.024 MHz. The bus frequency is 32.768 MHz and the CPU frequency is 65.536 MHz. Table 10 and Table 11 show the configuration of the CPMU module for Run, Wait, and Stop current measurements. Table 12 shows the configuration of the peripherals for run current measurements

Table 10. CPUM configuration for run/wait and full stop current measurement

CPMU register	Bit settings/conditions
CPMUSYNR	VCOFRQ[1:0]=01,SYNDIV[5:0] = 32.768 MHz
CPMUPOSTDIV	POSTDIV[4:0]=0,
CPMUCLKS	PLLSEL=1
CPMUOSC	OSCE=0, Reference clock for PLL is f_{REF}=f_{IRC1M} trimmed to 1.024 MHz

Table 11. CPMU configuration for pseudo stop current measurements

CPMU register	Bit settings/conditions
CPMUCLKS	PLLSEL=0, PSTP=1, PRE=PCE=RTIOSCSEL=COPOSCSEL=1
CPMUOSC	OSCE=1, External square wave on EXTAL f_{EXTAL}=16 MHz, V_{IH}= 1.8 V, V_{IL}=0 V
CPMURTI	RTDEC=0, RTR[6:4]=111, RTR[3:0]=1111;
CPMUCOP	WCOP=1, CR[2:0]=111

Table 12. MCU peripheral configurations for run supply current measurements

Peripheral	Configuration
SPI	configured to master mode, continuously transmit data (0x55 or 0xAA) at 4.0 Mbit/s
D2DI	continuously read data
COP	COP Watchdog Rate 2^{24}
RTI	enabled, RTI Control Register (RTICTL) set to $FF
DBG	The module is disabled.

Table 13. Analog die configurations for normal mode supply current measurements

Peripheral	Configuration
D2D	maximum frequency (32.768 MHz)
LIN	enabled, 50% dominant, 50% recessive
TIMER	enabled, all channels active in output compare mode with minimum timeout
LTC	enabled, maximum timeout
SCI	continuously transmitting data (0x55 or 0xAA) with 19.2 kBit/s
Acquisition Channels	current/voltage: highest sampling rate (8.0 kHz), LPF enabled, chopper and compensation enabled, automatic gain adjustment enabled temperature: internal temperature measurement enabled, 1.0 kHz sampling rate

Table 14. Supply currents[9]

Ratings	Symbol	Min.	Typ.[11]	Max.	Unit
MM912_637 combined consumption					
Normal mode current both dice	I_{RUN}			25	mA
Stop Mode current measured at VSUP	I_{STOP}				μA
Continuous base current [11]					
-40 °C i_{STOP_S}		–	100	120	
85 °C [10]		–	105	190	
125 °C		–	106	230	
Pseudo Stop Current, RTI and COP enabled [11]					
-40 °C i_{STOP_S}		–	410	490	
85 °C [10]		–	450	520	
125 °C		–	520	590	
Stop Current during Cranking Mode					
-40 °C i_{STOP_S}		–	105	130	
85 °C [10]		–	125	230	
125 °C		–	185	270	
Current adder during current trigger event - (typ. 10 ms duration[13], temperature measurement = OFF)		–	1500	1750	
Sleep Mode measured at VSUP	I_{SLEEP}				μA
-40 °C i_{STOP_S}		–	60	80	
85 °C [10]		–	60	130	
125 °C		–	80	140	
Current adder during current trigger event - (typ. 10 ms duration [13], temperature measurement = OFF)		–	1500	1750	
Analog die contribution - excluding MCU and external load current, (3.5 V ≤ V_{SUP} ≤ 28 V; -40 °C ≤ T_A ≤ 125 °C)					
Normal mode current measured at V_{SUP}	I_{NORMAL}		1.5	4.0	mA
MCU die contribution, V_{DDRX} = 5.5 V					
Run Current, T_A = 125 °C	I_{RUN}		13.5	18.8	mA
Wait current, T_A = 125 °C	I_{WAIT}		7.0	8.8	mA

Notes

[9] See Table 10, Table 11, Table 12, and Table 13 for conditions. Currents measured in Test mode with external loads (100 pF) and the external clock at 64 MHz.

[10] Not tested in production, guaranteed by characterization

[11] Typical values noted reflect the approximate parameter mean at T_A = 25 °C.

[12] From V_{SUP} 6.0 to 28 V

[13] Duration based on channel configuration. 10ms typical for Decimation Factor = 512, Chopper = ON.

3.5 Static electrical characteristics

All characteristics noted under conditions 3.5 V ≤ V_{SUP} ≤ 28 V, -40 °C ≤ T_A ≤ 125 °C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25 °C under nominal conditions, unless otherwise noted.

3.5.1 Static electrical characteristics analog die

Table 15. Static electrical characteristics - power supply

Ratings	Symbol	Min.	Typ.	Max.	Unit
Low Voltage Reset L (POR) Assert (measured on VDDL) Cranking Mode Disabled	V_{PORL}	1.75	1.9	2.1	V
Low Voltage Reset L (POR) Deassert (measured on VDDL) Cranking Mode Disabled	V_{PORH}	1.85	2.1	2.35	V
Low Voltage Reset L (POR) Assert (measured on VDDL) Cranking Mode Enabled[14]	V_{PORCL}	1.0	1.3	1.7	V
Low Voltage Reset A (LVRA) Assert (measured on VDDA)	V_{LVRAL}	1.9	2.05	2.2	V
Low Voltage Reset A (LVRA) Deassert (measured on VDDA)	V_{LVRAH}	2.0	2.15	2.3	V
Low Voltage Reset X (LVRX) Assert (measured on VDDX)	V_{LVRXL}	2.5	2.75	3.0	V
Low Voltage Reset X (LVRX) Deassert (measured on VDDX)	V_{LVRXH}	2.7	2.95	3.25	V
Low Voltage Reset H (LVRH) Assert (measured on VDDH)	V_{LVRHL}	1.9	2.075	2.2	V
Low Voltage Reset H (LVRH) Deassert (measured on VDDH)	V_{LVRHH}	2.05	2.175	2.3	V
Undervoltage Interrupt (UVI) Assert (measured on VSUP), Cranking Mode Disabled	V_{UVIL}	4.55	5.2	6.1	V
Undervoltage Interrupt (UVI) Deassert (measured on VSUP), Cranking Mode Disabled	V_{UVIH}	4.7	5.4	6.2	V
Undervoltage Cranking Interrupt (UVI) Assert (measured on VSUP) Cranking Mode Enabled	V_{UVCIL}	3.4	3.6	4.0	V
Undervoltage Cranking Interrupt (UVI) Deassert (measured on VSUP) Cranking Mode Enabled	V_{UVCIH}	3.5	3.8	4.1	V
VSENSE/VOPT High Voltage Warning Threshold Assert[15]	V_{TH}		28		V

Notes

[14] Deassert with Cranking off = V_{PORH}

[15] 5.0 V < V_{SUP} < 28 V, Digital Threshold at the end of channel chain (incl. compensation)

Table 16. Static electrical characteristics - resets

Ratings	Symbol	Min.	Typ.	Max.	Unit
Low-state Output Voltage I_{OUT} = 2.0 mA	V_{OL}			0.8	V
Pull-up Resistor	R_{RPU}	25		50	kΩ
Low-state Input Voltage	V_{IL}			$0.3V_{DDX}$	V
High-state Input Voltage	V_{IH}	$0.7V_{DDX}$			V
Reset Release Voltage (VDDX)	V_{RSTRV}	0	0.02	1.0	V
RESET_A pin Current Limitation	I_{LIMRST}			10	mA

Table 17. Static electrical characteristics - voltage regulator outputs

Ratings	Symbol	Min.	Typ.	Max.	Unit
Analog Voltage Regulator - VDDA[16]					
Output Voltage 1.0 mA ≤ I_{VDDA} ≤ 1.5 mA	V_{DDA}	2.25	2.5	2.75	V
Output Current Limitation	I_{VDDA}			10	mA
Low Power Digital Voltage Regulator - VDDL[16]					
Output Voltage	V_{DDL}	2.25	2.5	2.75	V
High Power Digital Voltage Regulator - VDDH[17]					
Output Voltage 1.0 mA ≤ I_{VDDH} ≤ 30 mA	V_{DDH}	2.4	2.5	2.75	V
Output Current Limitation	I_{VDDH}			65	mA
5.0 V Voltage Regulator - VDDX[17]					
Output Voltage 1.0 mA ≤ I_{VDDX} ≤ 30 mA	V_{DDX}	3.15	5.0	5.9	V
Output Current Limitation	I_{VDDX}	45	60	80	mA

Notes

[16] No additional current must be taken from those outputs.

[17] The specified current ranges does include the current for the MCU die. No external loads recommended.

Table 18. Static electrical characteristics - LIN physical layer interface - LIN

Ratings	Symbol	Min.	Typ.	Max.	Unit
Current Limitation for Driver dominant state. V_{BUS} = 18 V	I_{BUSLIM}	40	120	200	mA
Input Leakage Current at the Receiver incl. Pull-up Resistor R_{SLAVE}; Driver OFF; V_{BUS} = 0 V; V_{BAT} = 12 V	$I_{BUS\_PAS\_DOM}$	-1.0			mA
Input Leakage Current at the Receiver incl. Pull-up Resistor R_{SLAVE}; Driver OFF; 8.0 V < V_{BAT} < 18 V; 8.0 V < V_{BUS} < 18 V; $V_{BUS} \geq V_{BAT}$	$I_{BUS\_PAS\_REC}$			20	µA
Input Leakage Current; GND Disconnected; $GND_{DEVICE} = V_{SUP}$; 0 < V_{BUS} < 18 V; V_{BAT} = 12 V	$I_{BUS\_NO\_GND}$	-1.0		1.0	mA
Input Leakage Current; V_{BAT} disconnected; V_{SUP_DEVICE} = GND; 0 < V_{BUS} < 18 V	$I_{BUS\_NO\_BAT}$			100	µA
Receiver Input Voltage; Receiver Dominant State	V_{BUSDOM}			0.4	V_{SUP}
Receiver Input Voltage; Receiver Recessive State	V_{BUSREC}	0.6			V_{SUP}
Receiver Threshold Center ($V_{TH\_DOM} + V_{TH\_REC}$)/2	V_{BUS_CNT}	0.475	0.5	0.525	V_{SUP}
Receiver Threshold Hysteresis ($V_{TH\_REC} - V_{TH\_DOM}$)	V_{BUS_HYS}			0.175	V_{SUP}
Voltage Drop at the serial Diode	D_{SER_INT}	0.3	0.7	1.0	V
LIN Pull-up Resistor	R_{SLAVE}	20	30	60	kΩ
Low Level Output Voltage, I_{BUS}=40 mA	V_{DOM}			0.3	V_{SUP}
High Level Output Voltage, I_{BUS}=-10 µA, R_L=33 kΩ	V_{REC}	VSUP-1			V
J2602 Detection Deassert Threshold for VSUP level	V_{J2602H}	5.9	6.3	6.7	V
J2602 Detection Assert Threshold for VSUP level	V_{J2602L}	5.8	6.2	6.6	V
J2602 Detection Hysteresis	$V_{J2602HYS}$	70	190	250	mV
BUS Wake-up Threshold	V_{LINWUP}	4.0	5.25	6.0	V

Table 19. Static electrical characteristics - high voltage input - PTB3 / L0

Ratings	Symbol	Min.	Typ.	Max.	Unit
Wake-up Threshold - Rising Edge	V_{WTHR}	1.3	2.6	3.4	V
Input High Voltage (digital Input)	V_{IH}	$0.7V_{DDX}$		$V_{DDX}+0.3$	V
Input Low Voltage (digital Input)	V_{IL}	$V_{SS}-0.3$		$0.35V_{DDX}$	V
Input Hysteresis	V_{HYS}	50	140	200	mV
Internal Clamp Voltage	V_{L0CLMP}	4.9	6.0	7.0	V
Input Current PTB3 / L0 (V_{IN} = 42 V; R_{L0}=47 kΩ)	I_{IN}			1.1	mA
Internal pull-down resistance[18]	R_{PD}	50	100	200	kΩ
PTB3 / L0 Series Resistor	R_{PTB3}	42.3	47	51.7	kΩ
PTB3 / L0 Capacitor	C_{L0}	42.3	47	51.7	nF

Notes

[18] Disabled by default.

Table 20. Static electrical characteristics - general purpose I/O - PTB[0...2]

Ratings	Symbol	Min.	Typ.	Max.	Unit
Input High Voltage	V_{IH}	$0.7V_{DDX}$		$V_{DDX}+0.3$	V
Input Low Voltage	V_{IL}	$V_{SS}-0.3$		$0.35V_{DDX}$	V
Input Hysteresis	V_{HYS}	50	140	200	mV
Input Leakage Current (pins in high-impedance input mode) (V_{IN} = V_{DDX} or V_{SSX})	I_{IN}	-1.0		1.0	µA
Output High Voltage (pins in output mode) Full drive I_{OH} = –5.0 mA	V_{OH}	$V_{DDX}-0.8$			V
Output Low Voltage (pins in output mode) Full drive I_{OL} = 5.0 mA	V_{OL}			0.8	V
Internal Pull-up Resistance (V_{IH} min. > Input voltage > V_{IL} max)[19]	R_{PUL}	25	37.5	50	kΩ
Input Capacitance	C_{IN}		6.0		pF
Maximum Current All PTB Combined[20]	I_{BMAX}	-17		17	mA
Output Drive strength at 10 MHz	C_{OUT}			100	pF

Notes

[19] Disabled by default.

[20] Overall VDDR Regulator capability to be considered.

Table 21. Static electrical characteristics - current sense module[21] (continued)

Ratings	Symbol	Min.	Typ.	Max.	Unit
Gain Error with temperature based gain compensation adjustment[22], [23]	$I_{GAINERR}$	-0.5	+/-0.1	0.5	%
Offset Error[24],[25]	$I_{OFFSETERR}$			0.5	µV
Resolution	I_{RES}		0.1		µV
ISENSEH, ISENSEL terminal voltage differential signal voltage range	V_{INC} V_{IND}	-300 -200		300 200	mV
Differential Leakage Current: differential voltage between ISENSEH/ ISENSEL, ≤200 mV	I_{SENSE_DLC}	-2.0		2.0	nA

Table 21. Static electrical characteristics - current sense module[21] (continued)

Ratings	Symbol	Min.	Typ.	Max.	Unit
Wake-up Current Threshold Resolution	$I_{RESWAKE}$		0.2		µV
Resistor Threshold for OPEN Detection	R_{OPEN}	0.8	1.25	1.8	MΩ

Notes

21. 3.5 V ≤ V_{SUP} ≤ 28 V, after applying default trimming values - see Section 5, “MM912_637 - trimming".
22. Gain Compensation adjustment on calibration request interrupt with TCALSTEP
23. ±0.65%, including lifetime drift for gain 256 and 512
24. Chopper Mode = ON, Gain with automatic gain control enabled
25. Parameter not tested. Guaranteed by design and characterization

Table 22. Static electrical characteristics - voltage sense module[26]

Ratings	Symbol	Min.	Typ.	Max.	Unit
Gain Error[27] 18 V < V_{IN} ≤ 28 V 3.5 V ≤ V_{IN} < 5.0 V[28] 5.0 V ≤ V_{IN} ≤ 18 V[28],[30]	$V_{GAINERR}$	 -0.5 -0.25 -0.15	 0.1 0.1 0.1	 0.5 0.25 0.15	%
Offset Error[29],[31]	$V_{OFFSETERR}$	-1.5		1.5	mV
Resolution with R_{VSENSE} = 2.2 kΩ	V_{RES}			0.5	mV

Notes

26. 3.5 V ≤ V_{SUP} ≤ 28 V, after applying default trimming values - see Section 5, “MM912_637 - trimming".
27. Including resistor mismatch drift
28. Gain Compensation adjustment on calibration request interrupt with TCALSTEP
29. Chopper Mode = ON.
30. ±0.2%, including lifetime drift
31. Parameter not tested. Guaranteed by design and characterization.

Table 23. Static electrical characteristics - temperature sense module[32]

Ratings	Symbol	Min.	Typ.	Max.	Unit
Measurement Range	T_{RANGE}	-40		150	°C
Accuracy -40 °C ≤ T_A ≤ 60 °C[33] -40 °C ≤ T_A ≤ 150 °C	T_{ACC}	 -2.0 -3.0		 2.0 3.0	K
Resolution	T_{RES}		8.0		mK
TSUP Voltage Output, 10 µA ≤ I_{TSUP} ≤ 100 µA	V_{TSUP}	1.1875	1.25	1.3125	V
TSUP Capacitor with ECAP = 1	C_{TSUP}	209	220	231	pF
Max Calibration Request Interrupt Temperature Step	$T_{CALSTEP}$	-25		25	K

Notes

32. 3.5 V ≤ V_{SUP} ≤ 28 V, after applying default trimming values - see Section 5, “MM912_637 - trimming".
33. Temperature not tested in production. Guaranteed by design and characterization.

3.5.2 Static electrical characteristics MCU die

Table 24. Static electrical characteristics - MCU

Ratings	Symbol	Min.	Typ.	Max.	Unit
Power On Reset Assert (measured on VDDRX)	V_{PORA}	0.6	0.9	-	V
Power On Reset Deassert (measured on VDDRX)	V_{PORD}	-	0.95	1.6	V
Low Voltage Reset Assert (measured on VDDRX)	V_{LVRA}	2.97	3.06	-	V
Low Voltage Reset Deassert (measured on VDDRX)	V_{LVRD}	-	3.09	3.3	V
Low Voltage Interrupt Assert (measured on VDDRX)	V_{LVIA}	4.06	4.21	4.36	V
Low Voltage Interrupt Deassert (measured on VDDRX)	V_{LVID}	4.19	4.34	4.49	V

Table 25. Static electrical characteristics - oscillator (OSCLCP)

Ratings	Symbol	Min.	Typ.	Max.	Unit
Startup Current	i_{OSC}	100			µA
Input Capacitance (EXTAL, XTAL pins)	C_{IN}		7.0		pF
EXTAL Pin Input Hysteresis	$V_{HYS,EXTAL}$	—	180	—	mV
EXTAL Pin oscillation amplitude (loop controlled Pierce)	$V_{PP,EXTAL}$	—	0.9	—	V

Table 26. 5.0 V I/O characteristics for all I/O pins except EXTAL, XTAL, TEST, D2DI, and supply pins (4.5 V < V_{DDRX} < 5.5 V; T_J: –40 °C to +150 °C, unless otherwise noted) (continued)

Ratings	Symbol	Min.	Typ.	Max.	Unit
Input High Voltage	V_{IH}	0.65*V_{DDRX}	—	—	V
Input High Voltage	V_{IH}	—	—	V_{DDRX}+0.3	V
Input Low Voltage	V_{IL}	—	—	0.35*V_{DDRX}	V
Input Low Voltage	V_{IL}	V_{SSRX}–0.3	—	—	V
Input Hysteresis	V_{HYS}		250	—	mV
Input Leakage Current (pins in high-impedance input mode)[34] V_{IN} = V_{DDRX} or V_{SSRX}	I_{IN}	–1.00	—	1.00	µA
Input Leakage Current (pins in high-impedance input mode)[35] V_{IN} = V_{DDX} or V_{SSX} T_A = –40 °C T_A = 25 °C T_A = 70 °C T_A = 85 °C T_A = 105 °C T_A = 110 °C T_A = 120 °C T_A = 125 °C T_A = 130 °C T_A = 150 °C	I_{IN}		±1.0 ±1.0 ±8.0 ±14 ±26 ±32 ±40 ±60 ±74 ±92 ±240		nA
Output High Voltage (pins in output mode), I_{OH} = –4.0 mA	V_{OH}	V_{DDRX} – 0.8	—	—	V
Output Low Voltage (pins in output mode), I_{OL} = 4.0 mA	V_{OL}	—	—	0.8	V
Internal Pull-up Current, V_{IH} min > input voltage > V_{IL} max	I_{PUL}	-10	—	-130	µA
Internal Pull-down Current, V_{IH} min > input voltage > V_{IL} max	I_{PDH}	10	—	130	µA
Input Capacitance	C_{in}	—	7	—	pF

Table 26. 5.0 V I/O characteristics for all I/O pins except EXTAL, XTAL, TEST, D2DI, and supply pins (4.5 V < V_{DDRX} < 5.5 V; T_J: –40 °C to +150 °C, unless otherwise noted) (continued)

Ratings	Symbol	Min.	Typ.	Max.	Unit
Injection Current[36] Single pin limit Total device Limit, sum of all injected currents	I_{ICS} I_{ICP}	–2.5 –25	—	2.5 25	mA

Notes

34. Maximum leakage current occurs at maximum operating temperature. Current decreases by approximately one-half for each 8.0 °C to 12 °C in the temperature range from 50 °C to 125 °C.
35. Maximum leakage current occurs at maximum operating temperature. Current decreases by approximately one-half for each 8.0 °C to 12 °C in the temperature range from 50 °C to 125 °C.
36. Refer to Section 3.5.2.1, “Current injection" for more details

3.5.2.1 Current injection

The power supply must maintain regulation within the V_{DDX} operating range during instantaneous and operating maximum current conditions. If positive injection current ($V_{IN} > V_{DDX}$) is greater than I_{DDX}, the injection current may flow out of VDDX and could result in the external power supply going out of regulation. Ensure that the external V_{DDX} load will shunt current greater than the maximum injection current. This will be the greatest risk when the MCU is not consuming power; e.g., if no system clock is present, or if the clock rate is very low, which would reduce overall power consumption.

3.6 Dynamic electrical characteristics

Dynamic characteristics noted under conditions 3.5 V ≤ V_{SUP} ≤ 28 V, -40 °C ≤ T_A ≤ 125 °C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25 °C under nominal conditions, unless otherwise noted.

3.6.1 Dynamic electrical characteristics analog die

Table 27. Dynamic electrical characteristics - modes of operation

Ratings	Symbol	Min.	Typ.	Max.	Unit
Low Power Oscillator Frequency	f_{OSCL}	—	512	—	kHz
Low Power Oscillator Tolerance over full temperature range Analog Option 2 Analog Option 1	f_{TOL_A}	-4.0 -5.0	— —	4.0 5.0	%
Low Power Oscillator Tolerance - synchronized ALFCLK[37] ALF clock cycle = 1.0 ms ALF clock cycle = 2.0 ms ALF clock cycle = 4.0 ms ALF clock cycle = 8.0 ms	f_{TOLC_A}	f_{TOL}-0.2 f_{TOL}-0.1 f_{TOL}-0.05 f_{TOL}-0.025	f_{TOL}	f_{TOL}+0.2 f_{TOL}+0.1 f_{TOL}+0.05 f_{TOL}+0.025	%

Notes

37. Parameter not tested. Guaranteed by design and characterization.

Table 28. Dynamic electrical characteristics - die to die interface - D2D

Ratings	Symbol	Min.	Typ.	Max.	Unit
Operating Frequency (D2DCLK, D2D[0:3])	f_{D2D}	—	—	32.768	MHz

Table 29. Dynamic electrical characteristics - resets

Ratings	Symbol	Min.	Typ.	Max.	Unit
Reset Deglitch Filter Time	t_{RSTDF}	1.0	2.0	3.2	µs
Reset Time for watchdog and Hardware Reset (RESETA pin set low)	t_{RSTRT}	—	32	—	µs

Table 30. Dynamic electrical characteristics - wake-up / cyclic sense

Ratings	Symbol	Min.	Typ.	Max.	Unit
Cyclic Wake-up Time[38]	t_{WAKEUP}	ALFCLK	—	TIM4CH	ms
Cyclic Current Measurement Step Width[39]	t_{STEP}	ALFCLK	—	16Bit	ms

Notes

[38] Cyclic wake-up on ALFCLK clock based 16 Bit TIMER with maximum 128x prescaler (min 1x)

[39] Cyclic wake-up on ALFCLK clock with 16 Bit programmable counter

Table 31. Dynamic electrical characteristics - window watchdog

Ratings	Symbol	Min.	Typ.	Max.	Unit
Initial Non-window Watchdog Timeout	t_{IWDTO}	see Figure 39			ms

Table 32. Dynamic electrical characteristics - LIN physical layer interface - LIN (continued)

Ratings	Symbol	Min.	Typ.	Max.	Unit
Bus Wake-up Deglitcher (Sleep and Stop Mode)	t_{PROPWL}	60	80	100	µs
Fast Bit Rate (Programming Mode)	BR_{FAST}	—	—	100	kBit/s
Propagation delay of receiver	t_{RX_PD}	—	—	6.0	µs
Symmetry of receiver propagation delay rising edge w.r.t. falling edge	t_{RX_SYM}	-2.0	—	2.0	µs
LIN driver - 20.0 kBit/s; bus load conditions (C_{BUS}; R_{BUS}): 1.0 nF; 1.0 kΩ / 6,8 nF;660 Ω / 10 nF;500 Ω					
Duty Cycle 1: $TH_{REC(MAX)} = 0.744 \times V_{SUP}$ $TH_{DOM(MAX)} = 0.581 \times V_{SUP}$ $7.0\ V \leq V_{SUP} \leq 18\ V$; $t_{BIT} = 50\ \mu s$; $D1 = t_{BUS\_REC(MIN)}/(2 \times t_{BIT})$	D1	0.396	—	—	
Duty Cycle 2: $TH_{REC(MIN)} = 0.422 \times V_{SUP}$ $TH_{DOM(MIN)} = 0.284 \times V_{SUP}$ $7.6\ V \leq V_{SUP} \leq 18\ V$; $t_{Bit} = 50\ \mu s$ $D2 = t_{BUS\_REC(MAX)}/(2 \times t_{BIT})$	D2	—	—	0.581	
LIN driver - 10.0 kBit/s; bus load conditions (C_{BUS}; R_{BUS}): 1.0 nF; 1.0 kΩ / 6,8 nF;660 Ω / 10 nF;500 Ω					
Duty Cycle 3: $TH_{REC(MAX)} = 0.778 \times V_{SUP}$ $TH_{DOM(MAX)} = 0.616 \times V_{SUP}$ $7.0\ V \leq V_{SUP} \leq 18\ V$; $t_{Bit} = 96\ \mu s$ $D3 = t_{BUS\_REC(MIN)}/(2 \times t_{BIT})$	D3	0.417	—	—	
Duty Cycle 4: $TH_{REC(MIN)} = 0.389 \times V_{SUP}$ $TH_{DOM(MIN)} = 0.251 \times V_{SUP}$ $7.6\ V \leq V_{SUP} \leq 18\ V$; $t_{BIT} = 96\ \mu s$ $D4 = t_{BUS\_REC(MAX)}/(2 \times t_{BIT})$	D4	—	—	0.590	

Table 32. Dynamic electrical characteristics - LIN physical layer interface - LIN (continued)

Ratings	Symbol	Min.	Typ.	Max.	Unit
LIN Transmitter Timing, (V_{SUP} from 7.0 to 18 V) - See Figure 5					
Transmitter Symmetry t_{TRAN_SYM} < MAX(ttran_sym60%, t_{TRAN_SYM}40%) t_{TRAN_SYM}60% = t_{TRAN_PDF}60% - t_{TRAN_PDR}60% t_{TRAN_SYM}40% = t_{TRAN_PDF}40% - t_{TRAN_PDR}40%	t_{TRAN_SYM}	-7.25	0	7.25	µs

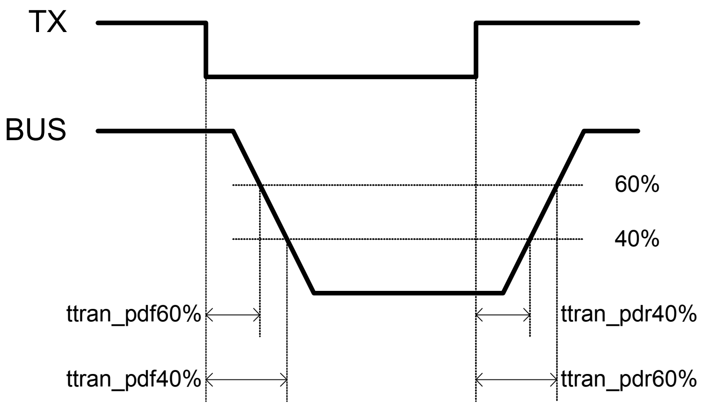


Figure 5. LIN transmitter timing

Table 33. Dynamic electrical characteristics - general purpose I/O - PTB3 / L0]

Ratings	Symbol	Min.	Typ.	Max.	Unit
Wake-up Glitch Filter Time	t_{WUPF}		20		µs

Table 34. Dynamic electrical characteristics - general purpose I/O - PTB[0...2]

Ratings	Symbol	Min.	Typ.	Max.	Unit
GPIO Digital Frequency	f_{PTB}			10	MHz
Propagation Delay - Rising Edge[40]	t_{PDr}			20	ns
Rise Time - Rising Edge[40]	t_{RISE}			17.5	ns
Propagation Delay - Falling Edge[40]	t_{PDf}			20	ns
Rise Time - Falling Edge[40]	t_{FALL}			17.5	ns

Notes

40. Load PTBx = 100 pF

Table 35. Dynamic Electrical Characteristics - Current Sense Module

Ratings	Symbol	Min	Typ	Max	Unit
Frequency Attenuation[41],[42] <100 Hz (f_{PASS}) >500 Hz (f_{STOP})		40		3.0	dB
Signal Update Rate[43]	$f_{IUPDATE}$	0.5		8.0	kHz
Signal Path Match with Voltage Channel	$f_{IVMATCH}$		2.0		µs
Gain Change Duration (Automatic GCB active)[44]	t_{GC}			14	µs

Notes

41. Characteristics identical to Voltage Sense Module
42. With default LPF coefficients
43. After passing decimation filter
44. Parameter not tested. Guaranteed by design and characterization.

Table 36. Dynamic electrical characteristics - voltage sense module

Ratings	Symbol	Min.	Typ.	Max.	Unit
Frequency attenuation[45],[46] 95...105 Hz (f_{PASS}) >500 Hz (f_{STOP})		40		3.0	dB
Signal update rate[47]	$f_{VUPDATE}$	0.5		8.0	kHz
Signal path match with Current Channel[48]	$f_{IVMATCH}$		2.0		µs

Notes

45. Characteristics identical to Voltage Sense Module
46. With default LPF coefficients
47. After passing decimation filter
48. Parameter not tested. Guaranteed by design and characterization.

Table 37. Dynamic electrical characteristics - temperature sense module

Ratings	Symbol	Min.	Typ.	Max.	Unit
Signal Update Rate[49]	$f_{TUPDATE}$	1.0		4.0	kHz

Notes

49. 1.0 kHz with Chopper Enabled, 4.0 kHz with Chopper Disabled (fixed decimeter = 128)

3.6.2 Dynamic electrical characteristics MCU die

3.6.2.1 NVM

3.6.2.1.1 Timing parameters

The time base for all NVM program or erase operations is derived from the bus clock using the FCLKDIV register. The frequency of this derived clock must be set within the limits specified as f_{NVMOP}. The NVM module does not have any means to monitor the frequency, and will not prevent program or erase operations at frequencies above or below the specified minimum. When attempting to program or erase the NVM module at a lower frequency, a full program or erase transition is not assured.

The following sections provide equations which can be used to determine the time required to execute specific flash commands. All timing parameters are a function of the bus clock frequency, f_{NVMBUS}. All program and erase times are also a function of the NVM operating frequency, f_{NVMOP}. A summary of key timing parameters can be found in Table 38.

3.6.2.1.1.1 Erase verify all blocks (blank check) (FCMD=0x01)

The time required to perform a blank check on all blocks is dependent on the location of the first non-blank word starting at relative address zero. It takes one bus cycle per phrase to verify, plus a setup of the command. Assuming that no non-blank location is found, then the time to erase verify all blocks is given by:

$$t_{check} = 35500 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.2 Erase verify block (blank check) (FCMD=0x02)

The time required to perform a blank check is dependent on the location of the first non-blank word starting at relative address zero. It takes one bus cycle per phrase to verify, plus a setup of the command.

Assuming that no non-blank location is found, then the time to erase verify a P-Flash block is given by:

$$t_{pcheck} = 33500 \cdot \frac{1}{f_{NVMBUS}}$$

Assuming that no non-blank location is found, then the time to erase verify a D-Flash block is given by:

$$t_{dcheck} = 2800 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.3 Erase verify p-flash section (FCMD=0x03)

The maximum time to erase verify a section of P-Flash depends on the number of phrases being verified (N_{VP}) and is given by:

$$t \approx (450 + N_{VP}) \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.4 Read once (FCMD=0x04)

The maximum read once time is given by:

$$t = 400 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.5 Program p-flash (FCMD=0x06)

The programming time for a single phrase of four P-Flash words and the two seven-bit ECC fields is dependent on the bus frequency, f_{NVMBUS}, as well as on the NVM operating frequency, f_{NVMOP}.

The typical phrase programming time is given by:

$$t_{ppgm} \approx 164 \cdot \frac{1}{f_{NVMOP}} + 2000 \cdot \frac{1}{f_{NVMBUS}}$$

The maximum phrase programming time is given by:

$$t_{ppgm} \approx 164 \cdot \frac{1}{f_{NVMOP}} + 2500 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.6 Program once (FCMD=0x07)

The maximum time required to program a P-Flash Program Once field is given by:

$$t \approx 164 \cdot \frac{1}{f_{NVMOP}} + 2150 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.7 Erase all blocks (FCMD=0x08)

The time required to erase all blocks is given by:

$$t_{mass} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 70000 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.8 Erase p-flash block (FCMD=0x09)

The time required to erase the P-Flash block is given by:

$$t_{pmass} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 67000 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.9 Erase p-flash sector (FCMD=0x0A)

The typical time to erase a 512-byte P-Flash sector is given by:

$$t_{pera} \approx 20020 \cdot \frac{1}{f_{NVMOP}} + 700 \cdot \frac{1}{f_{NVMBUS}}$$

The maximum time to erase a 512-byte P-Flash sector is given by:

$$t_{pera} \approx 20020 \cdot \frac{1}{f_{NVMOP}} + 1400 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.10 Unsecure flash (FCMD=0x0B)

The maximum time required to erase and unsecure the Flash is given by:
(for 128 kByte P-Flash and 4.0 kByte D-Flash)

$$t_{uns} \approx 100100 \cdot \frac{1}{f_{NVMOP}} + 70000 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.11 Verify backdoor access key (FCMD=0x0C)

The maximum verify back door access key time is given by:

$$t = 400 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.12 Set user margin level (FCMD=0x0D)

The maximum set user margin level time is given by:

$$t = 350 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.13 Set field margin level (FCMD=0x0E)

The maximum set field margin level time is given by:

$$t = 350 \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.14 Erase verify d-flash section (FCMD=0x10)

The time required to Erase Verify D-Flash for a given number of words N_W is given by:

$$t_{dcheck} \approx (450 + N_W) \cdot \frac{1}{f_{NVMBUS}}$$

3.6.2.1.1.15 Program d-flash (FCMD=0x11)

D-Flash programming time is dependent on the number of words being programmed and their location with respect to a row boundary, since programming across a row boundary requires extra steps. The D-Flash programming time is specified for different cases: 1,2,3,4 words and 4 words across a row boundary. The typical D-Flash programming time is given by the following equation, where N_W denotes the number of words; BC=0 if no row boundary is crossed and BC=1, if a row boundary is crossed:

$$t_{dpgm} \approx \left((14 + (54 \cdot N_W) + (14 \cdot BC)) \cdot \frac{1}{f_{NVMOP}}\right) + \left((500 + (525 \cdot N_W) + (100 \cdot BC)) \cdot \frac{1}{f_{NVMBUS}}\right)$$

The maximum D-Flash programming time is given by:

$$t_{dpgm} \approx \left((14 + (54 \cdot N_W) + (14 \cdot BC)) \cdot \frac{1}{f_{NVMOP}}\right) + \left((500 + (750 \cdot N_W) + (100 \cdot BC)) \cdot \frac{1}{f_{NVMBUS}}\right)$$

3.6.2.1.1.16 Erase d-flash sector (FCMD=0x12)

Typical D-Flash sector erase times, expected on a new device where no margin verify fails occur, is given by:

$$t_{dera} \approx 5025 \cdot \frac{1}{f_{NVMOP}} + 700 \cdot \frac{1}{f_{NVMBUS}}$$

Maximum D-Flash sector erase times is given by:

$$t_{dera} \approx 20100 \cdot \frac{1}{f_{NVMOP}} + 3400 \cdot \frac{1}{f_{NVMBUS}}$$

The D-Flash sector erase time is ~5.0 ms on a new device and can extend to ~20 ms as the flash is cycled.

Table 38. NVM timing characteristics (FTMRC)

Rating	Symbol	Min.	Typ.[50]	Max.[51]	Unit[52]
Bus Frequency	f_{NVMBUS}	1.0	—	32.768	MHz
Operating Frequency	f_{NVMOP}	0.8	1.0	1.05	MHz
Erase All Blocks (mass erase) Time	t_{MASS}	—	100	130	ms
Erase Verify All Blocks (blank check) Time	t_{CHECK}	—	—	35500	t_{CYC}
Unsecure Flash Time	t_{UNS}	—	100	130	ms
P-flash Block Erase Time	t_{PMASS}	—	100	130	ms
P-flash Erase Verify (blank check) Time	t_{PCHECK}	—	—	33500	t_{CYC}
P-flash Sector Erase Time	t_{PERA}	—	20	26	ms
P-flash Phrase Programming Time	t_{PPGM}	—	226	285	µs
D-flash Sector Erase Time	t_{DERA}	—	5[53]	26	ms
D-flash Erase Verify (blank check) Time	t_{DCHECK}	—	—	2800	t_{CYC}
D-flash One Word Programming Time	t_{DPGM1}	—	100	107	µs
D-flash Two Word Programming Time	t_{DPGM2}	—	170	185	µs
D-flash Three Word Programming Time	t_{DPGM3}	—	241	262	µs
D-flash Four Word Programming Time	t_{DPGM4}	—	311	339	µs
D-flash Four Word Programming Time Crossing Row Boundary	t_{DPGM4C}	—	328	357	µs

Notes

[50] Typical program and erase times are based on typical f_{NVMOP} and maximum f_{NVMBUS}

[51] Maximum program and erase times are based on minimum f_{NVMOP} and maximum f_{NVMBUS}

[52] $t_{CYC} = 1 / f_{NVMBUS}$

[53] Typical value for a new device

3.6.2.1.2 NVM reliability parameters

The reliability of the NVM blocks is guaranteed by stress test during qualification, constant process monitors, and burn-in to screen early life failures. The data retention and program/erase cycling failure rates are specified at the operating conditions noted. The program/erase cycle count on the sector is incremented every time a sector or mass erase event is executed.

Table 39. NVM reliability characteristics[54]

Rating	Symbol	Min.	Typ.	Max.	Unit
Data retention at an average junction temperature of T_{JAVG} = 85 °C[54] after up to 10,000 program/erase cycles	t_{NVMRET}	20	100[56]	—	Years
Program Flash number of program/erase cycles (-40 °C ≤ T_J ≤ 150 °C)	n_{FLPE}	10 K	100 K[57]	—	Cycles
Data retention at an average junction temperature of T_{JAVG} = 85 °C[54] after up to 50,000 program/erase cycles	t_{NVMRET}	5.0	100[56]	—	Years
Data retention at an average junction temperature of T_{JAVG} = 85 °C[54] after up to 10,000 program/erase cycles	t_{NVMRET}	10	100[56]	—	Years
Data retention at an average junction temperature of T_{JAVG} = 85 °C[54] after less than 100 program/erase cycles	t_{NVMRET}	20	100[56]	—	Years
Data Flash number of program/erase cycles (-40 °C ≤ T_J ≤ 150°C)	n_{FLPE}	50 K	500 K[57]	—	Cycles

Notes

54. Conditions are shown in Table 9, unless otherwise noted
55. T_{JAVG} does not exceed 85 °C in a typical temperature profile over the lifetime of a consumer, industrial, or automotive application.
56. Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25°C using the Arrhenius equation. For additional information on how NXP defines Typical Data Retention, refer to Engineering Bulletin EB618
57. Spec table quotes typical endurance evaluated at 25°C for this product family. For additional information on how NXP defines Typical Endurance, refer to Engineering Bulletin EB619.

3.6.2.2 Phase locked loop

3.6.2.2.1 Jitter definitions

With each transition of the feedback clock, the deviation from the reference clock is measured and input voltage to the VCO is adjusted accordingly.The adjustment is done continuously with no abrupt changes in the VCOCLK frequency. Noise, voltage, temperature, and other factors, cause slight variations in the control loop resulting in a clock jitter. This jitter affects the real minimum and maximum clock periods as illustrated in Figure 6.

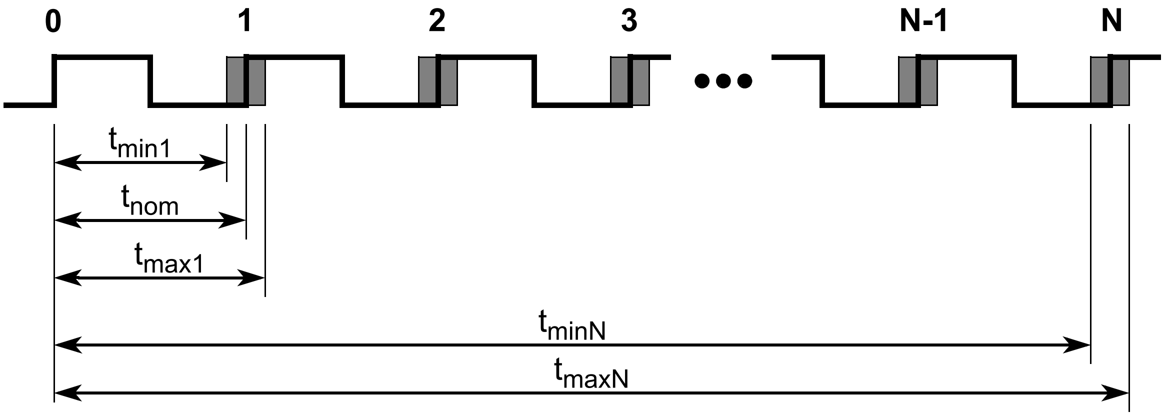


Figure 6. Jitter definitions

The relative deviation of t_{NOM} is at its maximum for one clock period, and decreases towards zero for larger number of clock periods (N).

Defining the jitter as:

$$J(N) = max\left(\left|1 - \frac{t_{max}(N)}{N \cdot t_{nom}}\right|, \left|1 - \frac{t_{min}(N)}{N \cdot t_{nom}}\right|\right)$$

For N < 100, the following equation is a good fit for the maximum jitter:

$$J(N) = \frac{j_1}{\sqrt{N}}$$

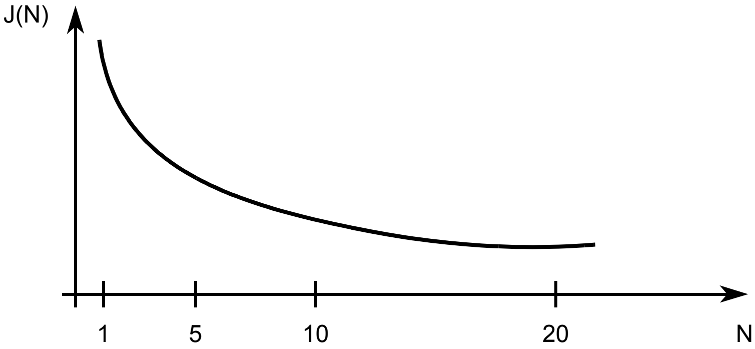


Figure 7. Maximum bus clock jitter approximation

NOTE

On timers and serial modules a prescaler will eliminate the effect of the jitter to a large extent.

3.6.2.2.2 Electrical characteristics for the PLL

Table 40. PLL characteristics

Rating	Symbol	Min	Typ	Max	Unit
VCO Frequency During System Reset	f_{VCORST}	8		32	MHz
VCO Locking Range	f_{VCO}	32.768		65.536	MHz
Lock Detection	$\|\Delta_{LOCK}\|$	0		1.5	%[58]
Un-lock Detection	$\|\Delta_{UNL}\|$	0.5		2.5	%[58]
Time to Lock	t_{LOCK}			150 + 256/f_{REF}	µs
Jitter Fit Parameter 1[59]	j_1			1.2	%

Notes

[58] % deviation from target frequency

[59] f_{REF} = 1.024 MHz, f_{BUS} = 32.768 MHz equivalent f_{PLL} = 65.536 MHz, REFRQ=00, SYNDIV=$1F, VCOFRQ=01, POSTDIV=$00

3.6.2.3 Reset, oscillator and internal clock generation

Table 41. Dynamic electrical characteristics - MCU clock generator

Ratings	Symbol	Min.	Typ.	Max.	Unit
Bus Frequency	f_{BUS}	—	—	32.768	MHz
Internal Reference Frequency	f_{IRC1M_TRIM}	—	1.024	—	MHz
Internal Clock Frequency Tolerance[60],[61] Analog Option 2 Analog Option 1	f_{TOL}	 -1.0 -1.2	 — —	 1.0 1.2	%
Clock Frequency Tolerance with External Oscillator[62]	t_{TOLEXT}	-0.5		0.5	%
Crystal Oscillator Range	f_{OSC}	4.0		16	MHz
Oscillator Start-up Time (LCP, 4.0 MHz)[63]	t_{UPOSC}	—	2.0	10	ms
Oscillator Start-up Time (LCP, 8.0 MHz)[63]	t_{UPOSC}	—	1.6	8.0	ms
Oscillator Start-up Time (LCP, 16 MHz)[63]	t_{UPOSC}	—	1.0	5.0	ms
Clock Monitor Failure Assert Frequency	f_{CMFA}	200	400	1000	kHz

Notes

[60] -40 °C ≤ T_A ≤ 125 °C

[61] ±1.3%, including lifetime drift

[62] Dependent on the external OSC

[63] These values apply for carefully designed PCB layouts with capacitors that match the crystal/resonator requirements

3.6.2.4 Reset characteristics

Table 42. Reset and stop characteristics[64]

Rating	Symbol	Min.	Typ.	Max.	Unit
Reset Input Pulse Width, minimum input time	PW_{RSTL}	2.0			t_{VCORST}
Startup from Reset	n_{RST}		768		t_{VCORST}
STOP Recovery Time	t_{STP_REC}		50		µs

Notes

[64] Conditions are shown in Table 9 unless otherwise noted

3.6.2.5 SPI timing

This section provides electrical parameters and ratings for the SPI. The measurement conditions are listed in Table 43.

Table 43. Measurement conditions

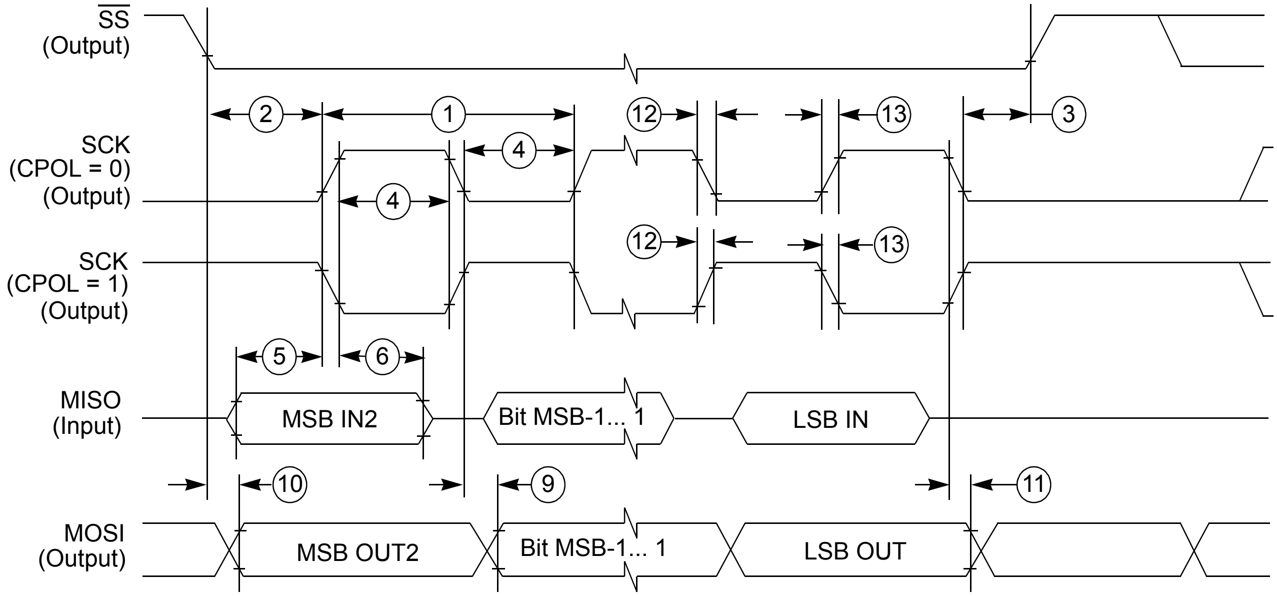
Description	Value	Unit
Drive mode	Full drive mode	—
Load capacitance C_{LOAD}[65], on all outputs	50	pF

Notes

[65] Conditions are shown in Table 9 unless otherwise noted

3.6.2.5.1 Master mode

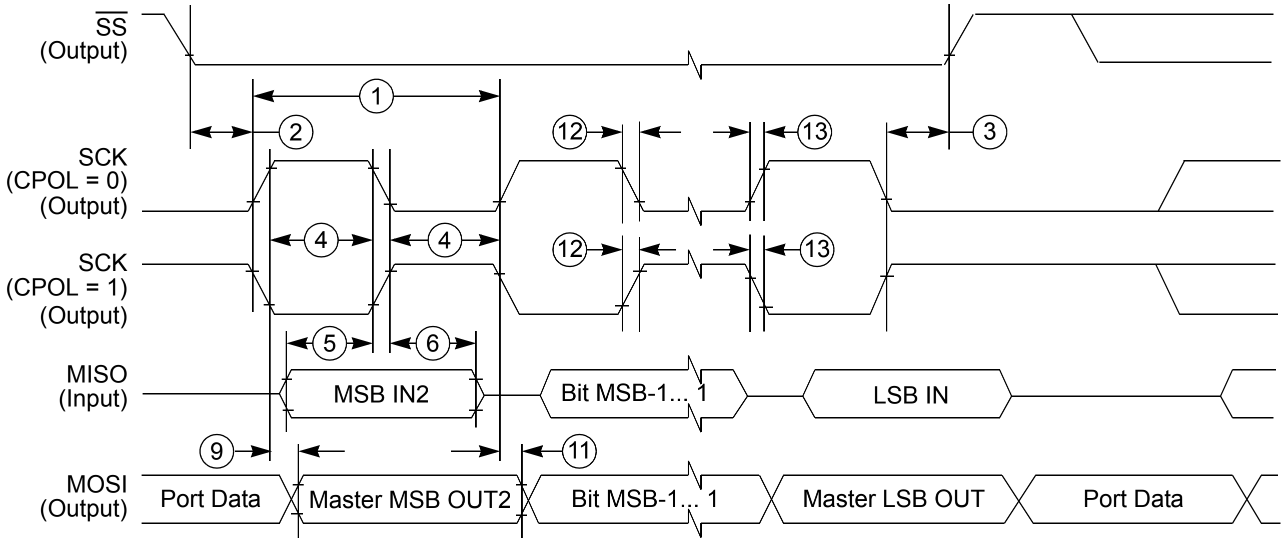
The timing diagram for master mode with transmission format CPHA = 0 is depicted in Figure 8.



1. If configured as an output.
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1, bit 2... MSB.

Figure 8. SPI master timing (CPHA = 0)

The timing diagram for master mode with transmission format CPHA=1 is depicted in Figure 9.



1.If configured as output
2. LSBF = 0. For LSBF = 1, bit order is LSB, bit 1,bit 2... MSB.

Figure 9. SPI master timing (CPHA = 1)

The timing characteristics for master mode are listed in Table 44.

Table 44. SPI master mode timing characteristics

Num	C	Characteristic	Symbol	Min.	Typ.	Max.	Unit
1	D	SCK Frequency	f_{SCK}	1/2048	—	1/2	f_{BUS}
1	D	SCK Period	t_{SCK}	2.0	—	2048	t_{BUS}
2	D	Enable Lead Time	t_{LEAD}	—	1/2	—	t_{SCK}
3	D	Enable Lag Time	t_{LAG}	—	1/2	—	t_{SCK}
4	D	Clock (SCK) High or Low Time	t_{WSCK}	—	1/2	—	t_{SCK}
5	D	Data Setup Time (inputs)	t_{SU}	8.0	—	—	ns
6	D	Data Hold Time (inputs)	t_{HI}	8.0	—	—	ns
9	D	Data Valid After SCK Edge	t_{VSCK}	—	—	29	ns
10	D	Data Valid After $\overline{SS}$ Fall (CPHA = 0)	t_{VSS}	—	—	15	ns
11	D	Data Hold Time (outputs)	t_{HO}	20	—	—	ns
12	D	Rise and Fall Time Inputs	t_{RFI}	—	—	8.0	ns
13	D	Rise and Fall Time Outputs	t_{RFO}	—	—	8.0	ns

3.6.2.5.2 Slave mode

The timing diagram for slave mode with transmission format CPHA = 0 is depicted in Figure 10.

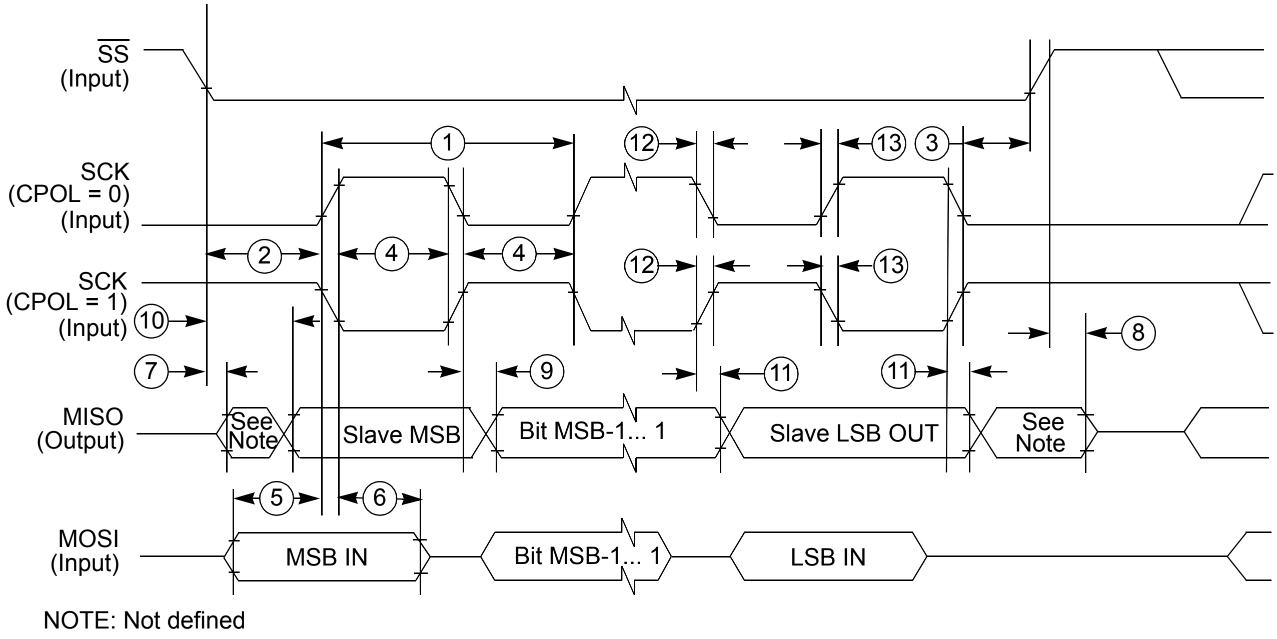
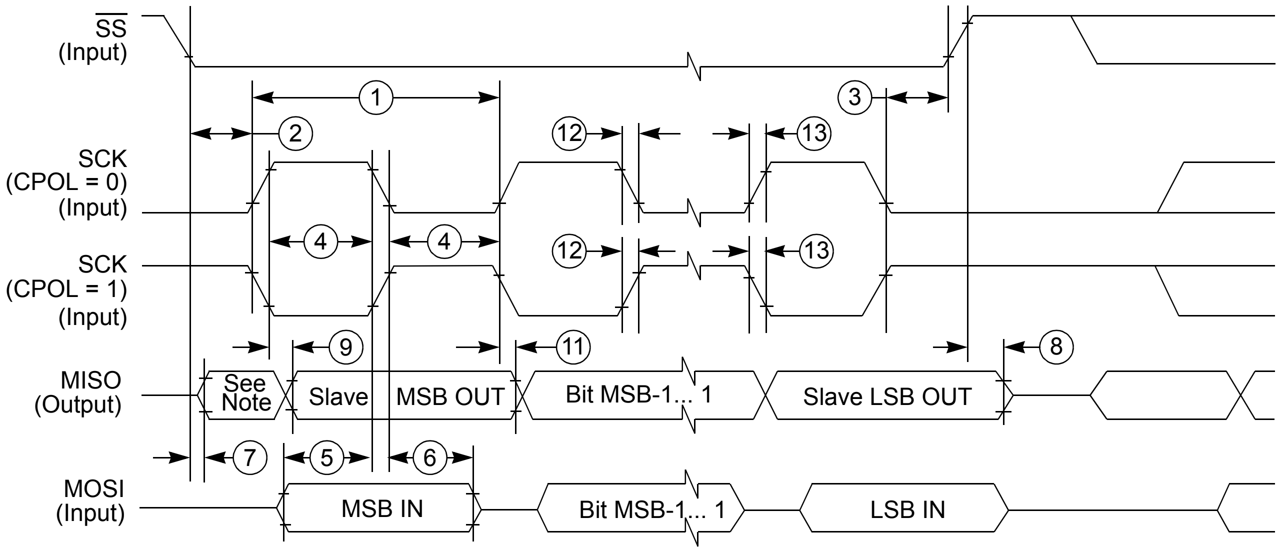


Figure 10. SPI slave timing (CPHA = 0)

The timing diagram for slave mode with transmission format CPHA = 1 is depicted in Figure 11.



NOTE: Not defined

Figure 11. SPI slave timing (CPHA = 1)

The timing characteristics for slave mode are listed in Table 45.

Table 45. SPI slave mode timing characteristics

Num	C	Characteristic	Symbol	Min.	Typ.	Max.	Unit
1	D	SCK Frequency	f_{SCK}	DC	—	1/4	f_{BUS}
1	D	SCK Period	t_{SCK}	4.0	—	∞	f_{BUS}
2	D	Enable Lead Time	t_{LEAD}	4.0	—	—	f_{BUS}
3	D	Enable Lag Time	t_{LAG}	4.0	—	—	f_{BUS}
4	D	Clock (SCK) High or Low Time	t_{WSCK}	4.0	—	—	f_{BUS}
5	D	Data Setup Time (inputs)	t_{SU}	8.0	—	—	ns
6	D	Data Hold Time (inputs)	t_{HI}	8.0	—	—	ns
7	D	Slave Access Time (time to data active)	t_A	—	—	20	ns
8	D	Slave MISO Disable Time	t_{DIS}	—	—	22	ns
9	D	Data Valid After SCK Edge	t_{VSCK}	—	—	$29 + 0.5 \cdot t_{BUS}$[66]	ns
10	D	Data Valid After $\overline{SS}$ Fall	t_{VSS}	—	—	$29 + 0.5 \cdot t_{BUS}$[66]	ns
11	D	Data Hold Time (outputs)	t_{HO}	20	—	—	ns
12	D	Rise and Fall Time Inputs	t_{RFI}	—	—	8.0	ns
13	D	Rise and Fall Time Outputs	t_{RFO}	—	—	8.0	ns

Notes

[66] 0.5 t_{BUS} added due to internal synchronization delay

3.7 Thermal protection characteristics

Characteristics noted under conditions 3.5 V ≤ V_{SUP} ≤ 28 V, -40 °C ≤ T_A ≤ 125 °C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25 °C under nominal conditions, unless otherwise noted.

Table 46. Thermal characteristics

Ratings	Symbol	Min	Typ	Max	Unit
VDDH/VDDA/VDDX High Temperature Warning (HTI) Threshold Hysteresis	T_{HTI} T_{HTI_H}	110	125 10	140	°C
VDDH/VDDA/VDDX Overtemperature Shutdown Threshold Hysteresis	T_{SD} T_{SD_H}	155	165 10	180	°C
LIN Overtemperature Shutdown	T_{LINSD}	150	165	180	°C
LIN Overtemperature Shutdown Hysteresis	T_{LINSD_HYS}		20		°C

3.8 Electromagnetic compatibility (EMC)

All ESD testing is in conformity with the CDF-AEC-Q100 stress test qualification for automotive grade integrated circuits. During the device qualification, ESD stresses are performed for the Human Body Model (HBM), Machine Model (MM), Charge Device Model (CDM), as well as LIN transceiver specific specifications.

A device will be defined as a failure, if after exposure to ESD pulses, the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature, followed by hot temperature, unless specified otherwise in the device specification. The immunity against transients for the LIN, PTB3/L0, VSENSE, ISENSEH, ISENSEL, and VSUP, is specified according to the LIN Conformance Test Specification - Section LIN EMC Test Specification (ISO7637-2), refer to the LIN Conformance Test Certification Report - available as separate document.

Table 47. Electromagnetic compatibility

Ratings	Symbol	Value/limit	Unit
ESD - Human Body Model (HBM) following AEC-Q100 / JESD22-A114 (C_{ZAP} = 100 pF, R_{ZAP} = 1500 Ω) - LIN (all GNDs shorted) - All other Pins	V_{HBM}	±8.0 ±2.0	kV
ESD - Charged Device Model (CDM) following AEC-Q100 Corner Pins All other Pins	V_{CDM}	±750 ±500	V
ESD - Machine Model (MM) following AEC-Q100 (C_{ZAP} = 200 pF, R_{ZAP} = 0 Ω), All Pins	V_{MM}	±200	V
Latch-up current at T_A = 125 °C[67]	I_{LAT}	±100	mA
ESD GUN - LIN Conformance Test Specification[68], unpowered, contact discharge. (C_{ZAP}= 150 pF, R_{ZAP} = 330 Ω); LIN (no bus filter C_{BUS}); VSENSE with serial R_{VSENSE}; VSUP with C_{VSUP}; PTB3 with serial R_{PTB3}		± 6000	V
ESD GUN - IEC 61000-4-2 Test Specification[69], unpowered, contact discharge. (C_{ZAP}= 150 pF, R_{ZAP} = 330 Ω); LIN (no bus filter C_{BUS}); VSENSE with serial R_{VSENSE}; VSUP with C_{VSUP}; PTB3 with serial R_{PTB3}		± 6000	V
ESD GUN - ISO10605[69], unpowered, contact discharge, C_{ZAP}= 150 pF, R_{ZAP} = 2.0 kΩ; LIN (no bus filter C_{BUS}); VSENSE with serial R_{VSENSE}; VSUP with C_{VSUP}; PTB3 with serial R_{PTB3}		± 8000	V
ESD GUN - ISO10605[69], powered, contact discharge, C_{ZAP}= 330 pF, R_{ZAP} = 2.0 kΩ; LIN (no bus filter C_{BUS}); VSENSE with serial R_{VSENSE}; VSUP with C_{VSUP}; PTB3 with serial R_{PTB3}		± 8000	V

Notes

[67] Input Voltage Limit = -2.5 to 7.5 V

[68] Certification available on request

[69] Tested internally only, following the reference document test procedure.

4 Functional description and application information

This chapter describes the MM912_637 dual die device functions on a block by block base. The following symbols are shown on all module cover pages to distinguish between the module location being the MCU die or the analog die:

The documented module is physically located on the Analog die. This applies to Section 4.2, “MM912_637 - analog die overview" through Section 4.15, “Die to die interface - target".

The documented module is physically located on the Microcontroller die. This applies to Section 4.2, “MM912_637 - analog die overview" through Section 4.26, “MCU - die-to-die initiator (9S12I128PIMV1)".

Sections concerning both die or the complete device will not have a specific indication (e.g. Section 5, “MM912_637 - trimming").

4.1 Introduction

Many types of electronic control units (ECUs) are connected to and supplied from the main car battery in modern cars. Depending on the cars mode of operation (drive, start, stop, standby), the battery must deliver different currents to the different ECUs. The vehicle power management has several sub-functions, like control of the set-point value of the power generator, dynamic load management during drive, start, stop, and standby mode.

The Application Specific Integrated Circuit (ASIC) allows for two application circuits, depending on whether the bias current of the MM912_637 itself shall be included into the current measurement.

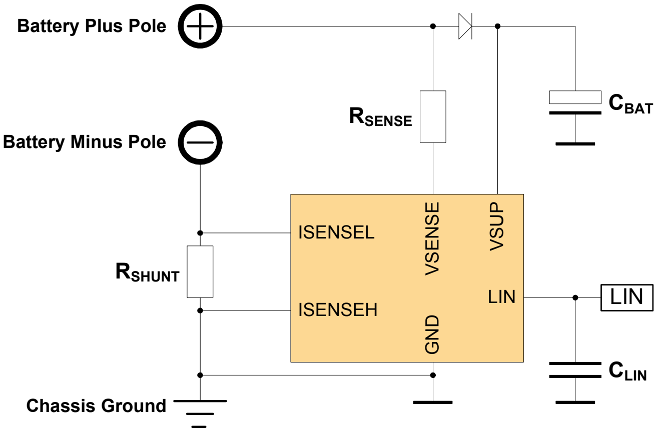


Figure 12. Typical IBS application (device GND = chassis GND)

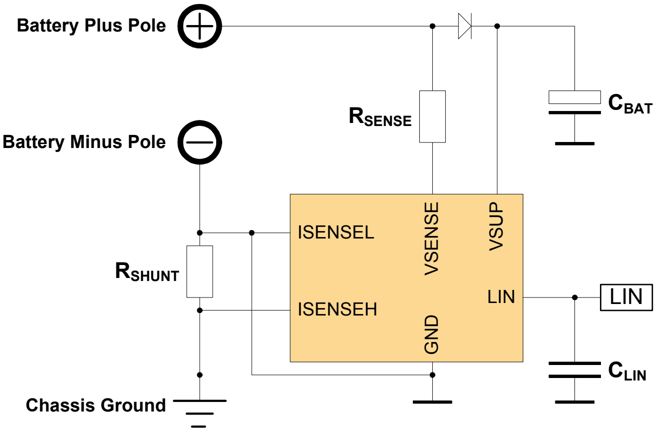


Figure 13. Typical IBS application (device GND = battery minus)

The vehicle power system needs actual measurement data from the battery, mainly voltage, current, and temperature. Out of these measurement data, it needs calculated characteristics, such as dynamic internal battery resistance. Therefore, an intelligent battery sensor (IBS) module is required.

To efficiently measure the battery voltage, current, and temperature, the IBS module is directly connected to and supplied from the battery. It is located directly on the negative pole of the battery; the supply of the IBS module comes from 'KL30'. The battery current is measured via a low-ohmic shunt resistor, connected between the negative pole of the battery and the chassis ground of the car. The battery voltage is measured at 'KL30'.

The data communication between the IBS module and the higher level ECU is done via a LIN interface. The MM912_637 is able to measure its junction temperature. That temperature is the basis for a model in software that calculates the battery temperature out of the junction temperature. An optional external temperature sense input is provided as well.

4.1.1 Device register map

Table 48 shows the device register memory map overview.

Table 48. Device register memory map overview

Address	Module	Size (Bytes)
0x0000–0x0003	PIM (port integration module)	4
0x0004–0x0009	Reserved	6
0x000A–0x000B	MMC (memory map control)	2
0x000C–0x000D	PIM (port integration module)	2
0x000E–0x000F	Reserved	2
0x0010–0x0015	MMC (memory map control)	8
0x0016–0x0019	Reserved	2
0x001A–0x001B	Device ID register	2
0x001C–0x001E	Reserved	4
0x001F	INT (interrupt module)	1
0x0020–0x002F	DBG (debug module)	16
0x0030–0x0033	Reserved	4
0x0034–0x003F	CPMU (clock and power management)	12
0x0040–0x00D7	Reserved	152
0x00D8–0x00DF	D2DI (die 2 die initiator)	8
0x00E0–0x00E7	Reserved	32
0x00E8–0x00EF	SPI (serial peripheral interface)	8
0x00F0–0x00FF	Reserved	32
0x0100–0x0113	FTMRC control registers	20
0x0114–0x011F	Reserved	12
0x0120–0x017F	PIM (port integration module)	96
0x0180–0x01EF	Reserved	112
0x01F0–0x01FC	CPMU (clock and power management)	13
0x01FD–0x01FF	Reserved	3
0x0200-0x02FF	D2DI (die 2 die initiator, blocking access window)	256
0x0300–0x03FF	D2DI (die 2 die initiator, non-blocking write window)	256

NOTE

The reserved register space shown in Table 48 is not allocated to any module. This register space is reserved for future use. Writing to these locations has no effect. Read access to these locations returns a zero.

4.1.2 Detailed module register map

Table 49 to Table 62 show the detailed module maps of the MM912_637.

Table 49. 0x0000–0x0009 port integration module (PIM) 1 of 3

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0000	PTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
		W								
0x0001	PTE	R	0	0	0	0	0	0	PE1	PE0
		W								
0x0002	DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
0x0003	DDRE	R	0	0	0	0	0	0	DDRE1	DDRE0
		W								
0x0004-0x0009	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 50. 0x000A–0x000B memory map control (MMC) 1 of 2

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x000B	MODE	R	MODC	0	0	0	0	0	0	0
		W								

Table 51. 0x000C–0x000F port integration module (PIM) map 2 of 3

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x000C	PUCR	R	0	BKPUE	0	0	0	0	PDPEE	0
		W								
0x000D	RDRIV	R	0	0	0	0	RDRD	RDRC	0	0
		W								
0x000E-0x000F	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 52. 0x0010–0x0019 memory map control (MMC) 2 of 2

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0011	DIRECT	R	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
		W								
0x0012-0x 0014	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0015	PPAGE	R	0	0	0	0	PIX3	PIX2	PIX1	PIX0
		W								
0x0016-0x 0019	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 53. 0x001A–0x001E miscellaneous peripheral

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x001A	PARTIDH	R	PARTIDH							
		W								
0x001B	PARTIDL	R	PARTIDL							
		W								
0x001C-0x 001E	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 54. 0x001F interrupt module (S12SINT)

0x001F	IVBR	R	IVB_ADDR[7:0]
		W	

Table 55. 0x0020–0x002F debug module (S12XDBG)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0020	DBGC1	R	ARM	0	0	BDM	DBGBRK	0	COMRV	
		W		TRIG						
0x0021	DBGSR	R	TBF[70]	0	0	0	0	SSF2	SSF1	SSF0
		W								
0x0022	DBGTCR	R	0	TSOURCE	0	0	TRCMOD		0	TALIGN
		W								
0x0023	DBGC2	R	0	0	0	0	0	0	ABCM	
		W								
0x0024	DBGTBH	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		W								
0x0025	DBGTBL	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		W								

Table 55. 0x0020–0x002F debug module (S12XDBG) (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0026	DBGCNT	R	TBF[70]	0	CNT					
		W								
0x0027	DBGSCRX	R	0	0	0	0	SC3	SC2	SC1	SC0
		W								
0x0027	DBGMFR	R	0	0	0	0	0	MC2	MC1	MC0
		W								
0x0028[71]	DBGACTL	R	SZE	SZ	TAG	BRK	RW	RWE	NDB	COMPE
		W								
0x0028[72]	DBGBCTL	R	SZE	SZ	TAG	BRK	RW	RWE	0	COMPE
		W								
0x0028[73]	DBGCCTL	R	0	0	TAG	BRK	RW	RWE	0	COMPE
		W								
0x0029	DBGXAH	R	0	0	0	0	0	0	Bit 17	Bit 16
		W								
0x002A	DBGXAM	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002B	DBGXAL	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x002C	DBGADH	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002D	DBGADL	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x002E	DBGADHM	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002F	DBGADLM	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								

Notes

[70] This bit is visible at DBGCNT[7] and DBGSR[7]

[71] This represents the contents if the Comparator A control register is blended into this address.

[72] This represents the contents if the Comparator B control register is blended into this address.

[73] This represents the contents if the Comparator C control register is blended into this address.

Table 56. 0x0034–0x003F Clock and Power Management (CPMU) 1 of 2

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0034	CPMU SYNR	R	VCOFRQ[1:0]		SYNDIV[5:0]					
		W								
0x0035	CPMU REFDIV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
		W								

Table 56. 0x0034–0x003F Clock and Power Management (CPMU) 1 of 2 (continued)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0036	CPMU POSTDIV	R	0	0	0	POSTDIV[4:0]				
		W								
0x0037	CPMUFLG	R	RTIF	PORF	LVRF	LOCKIF	LOCK	ILAF	OSCIF	UPOSC
		W								
0x0038	CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
		W								
0x0039	CPMUCLKS	R	PLLSEL	PSTP	0	0	PRE	PCE	RTI OSCSEL	COP OSCSEL
		W								
0x003A	CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
		W								
0x003B	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								
0x003C	CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		W			WRTMASK					
0x003D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x003E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x003F	CPMU ARMCOP	R	0	0	0	0	0	0	0	0
		W	Bit 7	6	5	4	3	2	1	Bit 0

Table 57. 0x00D8–0x00DF Die 2 Die Initiator (D2DI) 1 of 3

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00D8	D2DCTL0	R	D2DEN	D2DCW	D2DSWAI	0	0	0	D2DCLKDIV[1:0]	
		W								
0x00D9	D2DCTL1	R	D2DIE	0	0	0	TIMOUT[3:0]			
		W								
0x00DA	D2DSTAT0	R	ERRIF	ACKERF	CNCLF	TIMEF	TERRF	PARF	PAR1	PAR0
		W								
0x00DB	D2DSTAT1	R	D2DIF	D2DBSY	0	0	0	0	0	0
		W								
0x00DC	D2DADRHI	R	RWB	SZ8	0	NBLK	0	0	0	0
		W								
0x000D	D2DADRLO	R	ADR[7:0]							
		W								
0x00DE	D2DDATAHI	R	DATA[15:8]							
		W								

Table 57. 0x00D8–0x00DF Die 2 Die Initiator (D2DI) 1 of 3 (continued)

0x00DF	D2DDATALO	R	DATA[7:0]							
		W								

Table 58. 0x00E8–0x00EF serial peripheral interface (SPI)

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x00E8	SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
		W								
0x00E9	SPICR2	R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
		W								
0x00EA	SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
		W								
0x00EB	SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
		W								
0x00EC	SPIDRH	R	R15	R14	R13	R12	R11	R10	R9	R8
		W	T15	T14	T13	T12	T11	T10	T9	T8
0x00ED	SPIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
		W	T7	T6	T5	T4	T3	T2	T1	T0
0x00EE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x00EF	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 59. 0x0100–0x0113 flash control & status register FTMRC

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0100	FCLKDIV	R	FDIVLD	FDIVLCK	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
		W								
0x0101	FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
		W								
0x0102	FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
		W								
0x0103	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0104	FCNFG	R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
		W								
0x0105	FERCNFG	R	0	0	0	0	0	0	DFDIE	SFDIE
		W								
0x0106	FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
		W								

Table 59. 0x0100–0x0113 flash control & status register FTMRC (continued)

0x0107	FERSTAT	R	0	0	0	0	0	0	DFDIF	SFDIF
		W								
0x0108	FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
		W								
0x0109	DFPROT	R	DPOPEN	0	0	0	DPS3	DPS2	DPS1	DPS0
		W								
0x010A	FCCOBHI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
		W								
0x010B	FCCOBLO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
		W								
0x010C-0x010F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0110	FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
		W								
0x0111-0x0113	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 60. 0x0120 port integration module (PIM) 2 of 2

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0120	PTIA	R	PTIA7	PTIA6	PTIA5	PTIA4	PTIA3	PTIA2	PTIA1	PTIA0
		W								
0x0121	PTIE	R	0	0	0	0	0	0	PTIE1	PTIE0
		W								
0x0122-0x017F	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 61. 0x01F0–0x01FF clock and power management (CPMU) 2of 2

Address	Name		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x01F0	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x01F1	CPMU LVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W								
0x01F2-0x01F7	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x01F8	CPMU IRCTRIMH	R	TCTRIM[3:0]				0	0	IRCTRIM[9:8]	
		W								
0x01F9	CPMU IRCTRIML	R	IRCTRIM[7:0]							
		W								

Table 61. 0x01F0–0x01FF clock and power management (CPMU) 2of 2 (continued)

0x01FA	CPMUOSC	R	OSCE	OSCBW	OSCPINS_ EN	OSCFILT[4:0]				
		W								
0x01FB	CPMUPROT	R	0	0	0	0	0	0	0	PROT
		W								
0x01FC	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3

Offset[74]	Name		15 7	14 6	13 5	12 4	11 3	10 2	9 1	8 0
0x00	PCR_CTL PCR Control Register	R	0	0	0	0	0	0	0	0
		W	HTIEM	UVIEM	HWRM	0	PFM[1:0]		OPMM[1:0]	
		R	HTIE	UVIE	0	0	PF[1:0]		OPM[1:0]	
		W			HWR	0				
0x02	PCR_SR (hi)	R	HTF	UVF	HWRF	WDRF	HVRF	LVRF	WULTCF	WLPMF
	PCR Status Register	W	Write 1 will clear the flags							
0x03	PCR_SR (lo)	R	WUAHTH F	WUCTHF	WUCALF	WULINF	WUPTB3 F	WUPTB2 F	WUPTB1 F	WUPTB0 F
	PCR Status Register	W	Write 1 will clear the flags							
0x04	PCR_PRESC PCR 1.0 ms prescaler	R	PRESC[15:0]							
		W								
		R								
		W								
0x06	PCR_WUE (hi)	R	WUAHTH	WUCTH	WUCAL	WULIN	WUPTB3	WUPTB2	WUPTB1	WUPTB0
	Wake-up Enable Register	W								
0x07	PCR_WUE (lo)	R	WULTC	0	0	0	0	0	0	0
	Wake-up Enable Register	W								
0x08	INT_SRC (hi)	R	TOV	CH3	CH2	CH1	CH0	LTI	HTI	UVI
	Interrupt source register	W								
0x09	INT_SRC (lo)	R	0	0	CAL	LTC	CVMI	RX	TX	ERR
	Interrupt source register	W								
0x0A	INT_VECT	R	0	0	0	0	IRQ[3:0]			
	Interrupt vector register	W								
0x0B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0C	INT_MSK (hi)	R	TOVM	CH3M	CH2M	CH1M	CH0M	LTIM	HTIM	UVIM
	Interrupt mask register	W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x0D	INT_MSK (lo)	R	0	0	CALM	LTCM	CVMM	RXM	TXM	ERRM
	Interrupt mask register	W								
0x0E	TRIM_ALF (hi)	R	PRDF	0	0	APRESC[12:8]				
	Trim for accurate 1.0 ms low freq clock	W								
0x0F	TRIM_ALF (lo)	R	APRESC[7:0]							
	Trim for accurate 1.0 ms low freq clock	W								
0x10	WD_CTL Watchdog control register	R	0	0	0	0	0	0	0	0
		W	Reserved					WDTOM[2:0]		
		R	Reserved	0	0	0	0	WDTO[2:0]		
		W								
0x12	WD_SR	R	0	0	0	0	0	0	WDOFF	WDWO
	Watchdog status register	W								
0x13	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x14	WD_RR	R	WDR[7:0]							
	Watchdog rearm register	W								
0x15	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x16	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x17	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x18	SCIBD (hi)	R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
	SCI Baud Rate Register	W								
0x19	SCIBD (lo)	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	SCI Baud Rate Register	W								
0x1A	SCIC1	R	LOOPS	0	RSRC	M	0	ILT	PE	PT
	SCI Control Register 1	W								
0x1B	SCIC2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	SCI Control Register 2	W								
0x1C	SCIS1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	SCI Status Register 1	W								
0x1D	SCIS2	R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
	SCI Status Register 2	W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x1E	SCIC3	R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
	SCI Control Register 3	W								
0x1F	SCID	R	R7	R6	R5	R4	R3	R2	R1	R0
	SCI Data Register	W	T7	T6	T5	T4	T3	T2	T1	T0
0x20	TIOS	R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
	Timer Input Capture/Output Compare Select	W								
0x21	CFORC	R	0	0	0	0	0	0	0	0
	Timer Compare Force Register	W					FOC3	FOC2	FOC1	FOC0
0x22	OC3M	R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
	Output Compare 3 Mask Register	W								
0x23	OC3D	R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
	Output Compare 3 Data Register	W								
0x24	TCNT (hi)	R	TCNT[15:0]							
	Timer Count Register	W								
0x25	TCNT (lo)	R								
	Timer Count Register	W								
0x26	TSCR1	R	TEN	0	0	TFFCA	0	0	0	0
	Timer System Control Register 1	W								
0x27	TTOV	R	0	0	0	0	TOV3	TOV2	TOV1	TOV0
	Timer Toggle Overflow Register	W								
0x28	TCTL1	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	Timer Control Register 1	W								
0x29	TCTL2	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	Timer Control Register 2	W								
0x2A	TIE	R	0	0	0	0	C3I	C2I	C1I	C0I
	Timer Interrupt Enable Register	W								
0x2B	TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	Timer System Control Register 2	W								
0x2C	TFLG1	R	0	0	0	0	C3F	C2F	C1F	C0F
	Main Timer Interrupt Flag 1	W								
0x2D	TFLG2	R	TOF	0	0	0	0	0	0	0
	Main Timer Interrupt Flag 2	W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x2E	TC0 (hi)	R	TC0[15:0]							
	Timer Input Capture/Output Compare Register 0	W								
0x2F	TC0 (lo)	R								
	Timer Input Capture/Output Compare Register 0	W								
0x30	TC1 (hi)	R	TC1[15:0]							
	Timer Input Capture/Output Compare Register 1	W								
0x31	TC1 (lo)	R								
	Timer Input Capture/Output Compare Register 1	W								
0x32	TC2 (hi)	R	TC2[15:0]							
	Timer Input Capture/Output Compare Register 2	W								
0x33	TC2 (lo)	R								
	Timer Input Capture/Output Compare Register 2	W								
0x34	TC3 (hi)	R	TC3[15:0]							
	Timer Input Capture/Output Compare Register 3	W								
0x35	TC3 (lo)	R								
	Timer Input Capture/Output Compare Register 3	W								
0x36	TIMTST	R	0	0	0	0	0	0	TCBYP	0
	Timer Test Register	W								
0x37	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x38	LTC_CTL (hi)	R	0	0	0	0	0	0	0	0
	Life Time Counter control register	W	LTCIEM							LTCEM
0x39	LTC_CTL (lo)	R	LTCIE	0	0	0	0	0	0	LTCE
	Life Time Counter control register	W								
0x3A	LTC_SR	R	LTCOF	0	0	0	0	0	0	0
	Life Time Counter status register	W	1 will clr							
0x3B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3C	LTC_CNT1 Life Time Counter Register	R	LTC[31:16]							
		W								
		R								
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x3E	LTC_CNT0 Life Time Counter Register	R	LTC[15:0]							
		W								
		R								
		W								
0x40	GPIO_CTL GPIO control register	R	0	0	0	0	0	0	0	0
		W		DIR2M	DIR1M	DIR0M	PE3M	PE2M	PE1M	PE0M
		R	0	DIR2	DIR1	DIR0	PE3	PE2	PE1	PE0
		W								
0x42	GPIO_PUC	R	0	0	0	0	PDE3	PUE2	PUE1	PUE0
	GPIO pull up/down configuration	W								
0x43	GPIO_DATA	R	0	0	0	0	PD3	PD2	PD1	PD0
	GPIO port data register	W								
0x44	GPIO_IN0	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 0 input configuration	W								
0x45	GPIO_OUT0	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 0 output configuration	W								PTBX0
0x46	GPIO_IN1	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 1 input configuration	W								
0x47	GPIO_OUT1	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 1 output configuration	W								PTBX1
0x48	GPIO_IN2	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 2 input configuration	W								
0x49	GPIO_OUT2	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 2 output configuration	W								PTBX2
0x4A	GPIO_IN3	R	PTWU	PTWU	TCAP3	TCAP2	TCAP1	TCAP0	0	0
	Port 3 input configuration	W								
0x4B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4F	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x50	LIN_CTL LIN control register	R	0	0	0	0	0	0	0	0
		W	OTIEM			TXDM	LVSDM	ENM	SRSM[1:0]	
		R	OTIE	0	0	TXD	LVSD	EN	SRS[1:0]	
		W								
0x52	LIN_SR (hi)	R	OT	0	HF	0	UV	0	0	0
	LIN status register	W	Write 1 will clear the flags							
0x53	LIN_SR (lo)	R	RDY	0	0	0	0	0	RX	TX
	LIN status register	W								
0x54	LIN_TX	R	0	0	0	0	0	0	FROMPTB	FROMSCI
	LIN transmit line definition	W								
0x55	LIN_RX	R	0	0	0	0	0	0	TOPTB	TOSCI
	LIN receive line definition	W								
0x56	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x57	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x58	ACQ_CTL Acquisition control register	R	0	0	0	0	0	0	0	0
		W	AHCRM	OPTEM	OPENEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
		R	0	OPTE	OPENE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
		W	AHCR							
0x5A	ACQ_SR (hi)	R	AVRF	PGAG	VMOW	CMOW	ETM	ITM	VM	CM
	Acquisition status register	W	Write 1 will clear the flags							
0x5B	ACQ_SR (lo)	R	OPEN	0	0	VTH	ETCHOP	ITCHOP	VCHOP	CCHOP
	Acquisition status register	W								
0x5C	ACQ_ACC1 Acquisition chain control 1	R	0	0	0	0	0	0	0	0
		W	TCOMPM	VCOMPM	CCOMPM	LPFENM	ETCHOPM	ITCHOPM	CVCHOPM	AGENM
		R	TCOMP	VCOMP	CCOMP	LPFEN	ETCHOP	ITCHOP	CVCHOP	AGEN
		W								
0x5E	ACQ_ACC0 Acquisition chain control 0	R	0	0	0	0	0	0	0	0
		W	ZEROM	ECAPM	TADCGM	VADCGM	CADCGM	TDENM	VDENM	CDENM
		R	ZERO	ECAP	TADCG	VADCG	CADCG	TDEN	VDEN	CDEN
		W								
0x60	ACQ_DEC	R	0	0	0	0	0	DEC[2:0]		
	Decimation rate	W								
0x61	ACQ_BGC	R	0	0	BGADC[1:0]		BGLDO	BG3EN	BG2EN	BG1EN
	BandGap control	W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x62	ACQ_GAIN	R	0	0	0	0	0	IGAIN[2:0]		
	PGA gain	W								
0x63	ACQ_GCB	R	D[7:0]							
	GCB threshold	W								
0x64	ACQ_ITEMP (hi)	R	ITEMP[15:8]							
	Internal temperature measurement	W								
0x65	ACQ_ITEMP (lo)	R	ITEMP[7:0]							
	Internal temperature measurement	W								
0x66	ACQ_ETEMP (hi)	R	EEMP[15:8]							
	External temperature measurement	W								
0x67	ACQ_ETEMP (lo)	R	EEMP[7:0]							
	External temperature measurement	W								
0x68	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x69	ACQ_CURR1	R	CURR[23:16]							
	Current measurement	W								
0x6A	ACQ_CURR0 Current measurement	R	CURR[15:8]							
		W								
		R	CURR[7:0]							
		W								
0x6C	ACQ_VOLT Voltage measurement	R	VOLT[15:8]							
		W								
		R	VOLT[7:0]							
		W								
0x6E	ACQ_LPFC	R	0	0	0	0	LPFC[3:0]			
	Low pass filter coefficient number	W								
0x6F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x70	ACQ_TCMP Low power trigger current measurement period	R	TCMP[15:0]							
		W								
		R								
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x72	ACQ_THF	R	THF[7:0]							
	Low power current threshold filtering period	W								
0x73	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x74	ACQ_CVCR (hi)	R	0	0	0	0	0	0	0	0
	I and V chopper control register	W			DBTM[1:0]		IIRCM[2:0]			PGAFM
0x75	ACQ_CVCR (lo)	R	0	0	DBT[1:0]		IIRC[2:0]			PGAF
	I and V chopper control register	W								
0x76	ACQ_CTH	R	CTH[7:0]							
	Low power current threshold	W								
0x77	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x78	ACQ_AHTH1 (hi)	R	0	AHTH[30:16]						
	Low power Ah counter threshold	W								
0x79	ACQ_AHTH1 (lo)	R								
	Low power Ah counter threshold	W								
0x7A	ACQ_AHTH0 (hi)	R	AHTH[15:0]							
	Low power Ah counter threshold	W								
0x7B	ACQ_AHTH0 (lo)	R								
	Low power Ah counter threshold	W								
0x7C	ACQ_AHC1 (hi)	R	AHC[31:24]							
	Low power Ah counter	W								
0x7D	ACQ_AHC1 (lo)	R	AHC[23:16]							
	Low power Ah counter	W								
0x7E	ACQ_AHC0 (hi)	R	AHC[15:8]							
	Low power Ah counter	W								
0x7F	ACQ_AHC0 (lo)	R	AHC[7:0]							
	Low power Ah counter	W								
0x80	LPF_A0 (hi)	R	A0[15:0]							
	A0 filter coefficient	W								
0x81	LPF_A0 (lo)	R								
	A0 filter coefficient	W								
0x82	LPF_A1 (hi)	R	A1[15:0]							
	A1 filter coefficient	W								
0x83	LPF_A1 (lo)	R								
	A1 filter coefficient	W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x84	LPF_A2 (hi)	R	A2[15:0]							
	A2 filter coefficient	W								
0x85	LPF_A2 (lo)	R								
	A2 filter coefficient	W								
0x86	LPF_A3 (hi)	R	A3[15:0]							
	A3 filter coefficient	W								
0x87	LPF_A3 (lo)	R								
	A3 filter coefficient	W								
0x88	LPF_A4 (hi)	R	A4[15:0]							
	A4 filter coefficient	W								
0x89	LPF_A4 (lo)	R								
	A4 filter coefficient	W								
0x8A	LPF_A5 (hi)	R	A5[15:0]							
	A5 filter coefficient	W								
0x8B	LPF_A5 (lo)	R								
	A5 filter coefficient	W								
0x8C	LPF_A6 (hi)	R	A6[15:0]							
	A6 filter coefficient	W								
0x8D	LPF_A6 (lo)	R								
	A6 filter coefficient	W								
0x8E	LPF_A7 (hi)	R	A7[15:0]							
	A7 filter coefficient	W								
0x8F	LPF_A7 (lo)	R								
	A7 filter coefficient	W								
0x90	LPF_A8 (hi)	R	A8[15:0]							
	A8 filter coefficient	W								
0x91	LPF_A8 (lo)	R								
	A8 filter coefficient	W								
0x92	LPF_A9 (hi)	R	A9[15:0]							
	A9 filter coefficient	W								
0x93	LPF_A9 (lo)	R								
	A9 filter coefficient	W								
0x94	LPF_A10 (hi)	R	A10[15:0]							
	A10 filter coefficient	W								
0x95	LPF_A10 (lo)	R								
	A10 filter coefficient	W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0x96	LPF_A11 (hi)	R	A11[15:0]							
	A11 filter coefficient	W								
0x97	LPF_A11 (lo)	R								
	A11 filter coefficient	W								
0x98	LPF_A12 (hi)	R	A12[15:0]							
	A12 filter coefficient	W								
0x99	LPF_A12 (lo)	R								
	A12 filter coefficient	W								
0x9A	LPF_A13 (hi)	R	A13[15:0]							
	A13 filter coefficient	W								
0x9B	LPF_A13 (lo)	R								
	A13 filter coefficient	W								
0x9C	LPF_A14 (hi)	R	A14[15:0]							
	A14 filter coefficient	W								
0x9D	LPF_A14 (lo)	R								
	A14 filter coefficient	W								
0x9E	LPF_A15 (hi)	R	A15[15:0]							
	A15 filter coefficient	W								
0x9F	LPF_A15 (lo)	R								
	A15 filter coefficient	W								
0xA0	COMP_CTL Compensation control register	R	0	0	0	0	0	0		0
		W	BGCALM[1:0]		PGAZM	PGAOM	DIAGVM	DIAGIM		CALIEM
		R	BGCAL[1:0]		PGAZ	PGAO	DIAGV	DIAGI		CALIE
		W								
0xA2	COMP_SR	R	0	BGRF	0	PGAOF	0	0	0	CALF
	Compensation status register	W	Write 1 will clear the flags							
0xA3	COMP_TF	R	0	0	0	0	0	TMF[2:0]		
	Temperature filtering period	W								
0xA4	COMP_TMAX Max temp before recalibration	R	TCMAX[15:0]							
		W								
		R								
		W								
0xA6	COMP_TMIN Min temp before recalibration	R	TCMIN[15:0]							
		W								
		R								
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15 7	14 6	13 5	12 4	11 3	10 2	9 1	8 0
0xA8	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xA9	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xAA	COMP_VO	R	VOC[7:0]							
	Offset voltage compensation	W								
0xAB	COMP_IO	R	COC[7:0]							
	Offset current compensation	W								
0xAC	COMP_VSG Gain voltage compensation vsense channel	R	0	0	0	0	0	0	VSGC[9:8]	
		W								
		R	VSGC[7:0]							
		W								
0xAE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xAF	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xB0	COMP_IG4 Gain current compensation 4	R	0	0	0	0	0	0	IGC4[9:8]	
		W								
		R	IGC4[7:0]							
		W								
0xB2	COMP_IG8 Gain current compensation 8	R	0	0	0	0	0	0	IGC8[9:8]	
		W								
		R	IGC8[7:0]							
		W								
0xB4	COMP_IG16 Gain current compensation 16	R	0	0	0	0	0	0	IGC16[9:8]	
		W								
		R	IGC16[7:0]							
		W								
0xB6	COMP_IG32 Gain current compensation 32	R	0	0	0	0	0	0	IGC32[9:8]	
		W								
		R	IGC32[7:0]							
		W								
0xB8	COMP_IG64 Gain current compensation 64	R	0	0	0	0	0	0	IGC64[9:8]	
		W								
		R	IGC64[7:0]							
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0xBA	COMP_IG128 Gain current compensation 128	R	0	0	0	0	0	0	IGC128[9:8]	
		W								
		R	IGC128[7:0]							
		W								
0xBC	COMP_IG256 Gain current compensation 256	R	0	0	0	0	0	0	IGC256[9:8]	
		W								
		R	IGC256[7:0]							
		W								
0xBE	COMP_IG512 Gain current compensation 512	R	0	0	0	0	0	0	IGC512[9:8]	
		W								
		R	IGC512[7:0]							
		W								
0xC0	COMP_PGAO4 Offset PGA compensation 4	R	0	0	0	0	0	PGAOC4[10:8]		
		W								
		R	PGAOC4[7:0]							
		W								
0xC2	COMP_PGAO8 Offset PGA compensation 8	R	0	0	0	0	0	PGAOC8[10:8]		
		W								
		R	PGAOC8[7:0]							
		W								
0xC4	COMP_PGAO16 Offset PGA compensation 16	R	0	0	0	0	0	PGAOC16[10:8]		
		W								
		R	PGAOC16[7:0]							
		W								
0xC6	COMP_PGAO32 Offset PGA compensation 32	R	0	0	0	0	0	PGAOC32[10:8]		
		W								
		R	PGAOC32[7:0]							
		W								
0xC8	COMP_PGAO64 Offset PGA compensation 64	R	0	0	0	0	0	PGAOC64[10:8]		
		W								
		R	PGAOC64[7:0]							
		W								
0xCA	COMP_PGAO128 Offset PGA compensation 128	R	0	0	0	0	0	PGAOC128[10:8]		
		W								
		R	PGAOC128[7:0]							
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15 7	14 6	13 5	12 4	11 3	10 2	9 1	8 0
0xCC	COMP_PGAO256 Offset PGA compensation 256	R	0	0	0	0	0	PGAOC256[10:8]		
		W								
		R	PGAOC256[7:0]							
		W								
0xCE	COMP_PGAO512 Offset PGA compensation 512	R	0	0	0	0	0	PGAOC512[10:8]		
		W								
		R	PGAOC512[7:0]							
		W								
0xD0	COMP_ITO	R	ITOC[7:0]							
	Internal temp. offset compensation	W								
0xD1	COMP_ITG	R	ITGC[7:0]							
	Internal temp. gain compensation	W								
0xD2	COMP_ETO	R	ETOC[7:0]							
	External temp. offset compensation	W								
0xD3	COMP_ETG	R	ETGC[7:0]							
	External temp. gain compensation	W								
0xD4	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD5	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD6	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD7	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD8	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD9	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xDA	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xDB	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xDC	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15 7	14 6	13 5	12 4	11 3	10 2	9 1	8 0
0xDD	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xDE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xDF	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xE0	TRIM_BG0 (hi)	R	0	0	TCIBG2[2:0]			TCIBG1[2:0]		
	Trim bandgap 0	W								
0xE1	TRIM_BG0 (lo)	R	0	0	IBG2[2:0]			IBG1[2:0]		
	Trim bandgap 0	W								
0xE2	TRIM_BG1 (hi)	R	UBG3	DBG3	TCBG2[2:0]			TCBG1[2:0]		
	Trim bandgap 1	W								
0xE3	TRIM_BG1 (lo)	R	0	0	0	0	0	SLPBG[2:0]		
	Trim bandgap 1	W								
0xE4	TRIM_BG2 (hi)	R	V1P2BG2[3:0]				V1P2BG1[3:0]			
	Trim bandgap 2	W								
0xE5	TRIM_BG2 (lo)	R	V2P5BG2[3:0]				V2P5BG1[3:0]			
	Trim bandgap 2	W								
0xE6	TRIM_LIN	R	0	0	0	0	0	0	0	LIN
	Trim LIN	W								
0xE7	TRIM_LVT	R	0	0	0	0	0	0	0	LVT
	Trim low voltage threshold	W								
0xE8	TRIM_OSC (hi)	R				LPOSC[12:0]				
	Trim LP oscillator	W								
0xE9	TRIM_OSC (lo)	R								
	Trim LP oscillator	W								
0xEA	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xEB	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xEC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xED	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xEE	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 62. Analog die registers - 0x0200–0x02FF D2D blocking access (D2DI) 2 of 3/0x0300–0x03FF D2D non blocking access (D2DI) 3 of 3 (continued)

Offset[74]	Name		15	14	13	12	11	10	9	8
			7	6	5	4	3	2	1	0
0xEF	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF0	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF1	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF2	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF3	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF4	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF5	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF6	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF7	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF8	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF9	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFA	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFB	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFD	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFF	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes

[74] Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

4.2 MM912_637 - analog die overview

4.2.1 Introduction

The MM912_637 analog die implements all system base functionality to operate the integrated microcontroller, and delivers application specific input capturing.

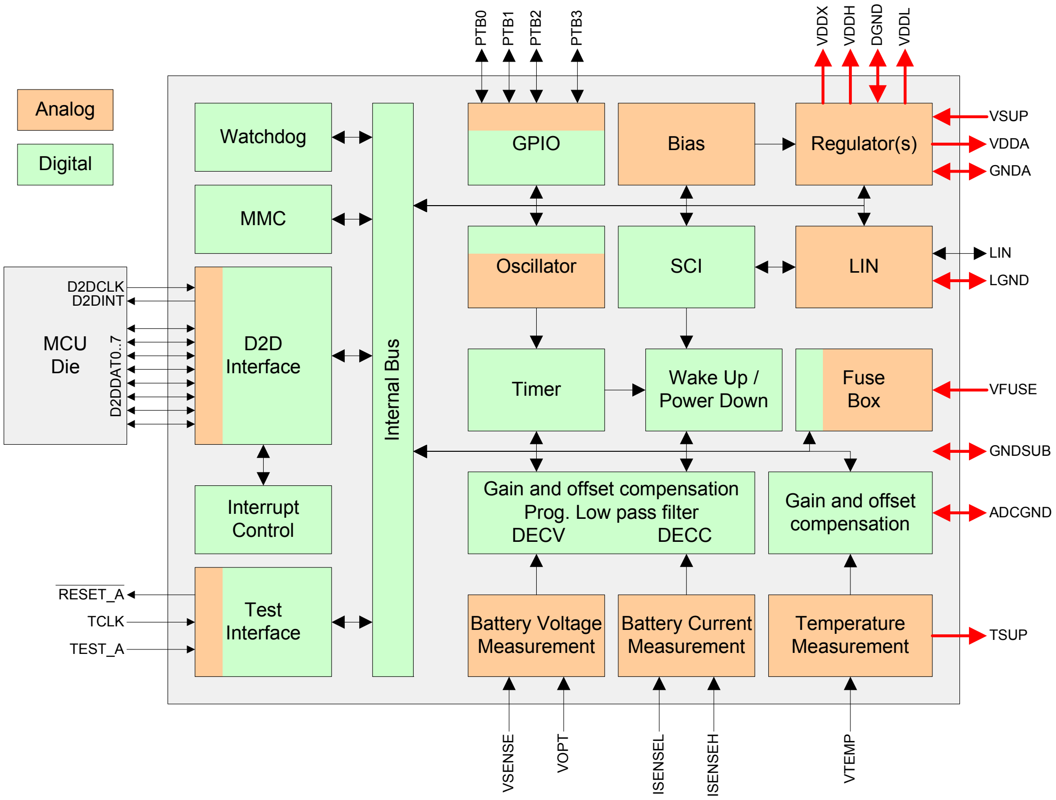


Figure 14. Analog die block overview

The following chapters describe the analog die functionality on a module by module basis.

4.2.2 Analog die options

NOTE

This document describes the features and functions of Analog Option 2 (all modules available and tested). Beyond this chapter, there will be no additional note or differentiation between the different implementations.

The following section describes the differences between analog die options 1 and 2.

Table 63. Analog options

Feature	Analog option 1	Analog option 2
Cranking Mode	Not Characterized or Tested	Fully Characterized and Tested
External Wake-up (PTB3/L0)	No	Yes
External Temperature Sensor Option (VTEMP)	No	Yes
Optional 2nd External Voltage Sense Input (VOPT)	No	Yes

4.2.2.1 Cranking mode

For devices with Analog Option 1 (Cranking mode not characterized), the following considerations are to be made:

4.2.2.1.1 Data sheet considerations

In Analog Option 1 devices, Operation in Cranking mode is neither characterized not tested. All data sheet parameters and descriptions relating to Cranking mode operation apply to Analog Option 2 devices only.

4.2.2.2 External wake-up (PTB3/L0)

For devices with Analog Option 1 (External Wake-up not available), the following considerations are to be made:

4.2.2.2.1 Register considerations

Table 64. Wake-up enable register (PCR_WUE (hi))

Offset [75] 0x06 Access: User read/write

	7	6	5	4	3	2	1	0
R W	WUAHTH	WUCTH	WUCAL	WULIN	WUPTB3	WUPTB2	WUPTB1	WUPTB0
Reset	0	0	0	0	0	0	0	0

Notes
75. Offset related to 0x0200 for blocking access and 0x300 for non-blocking access within the global address space.

For Analog Option 1 devices, WUPTB3 must be set to 0 (wake-up on a GPIO 3 event disabled).

4.2.2.3 External temperature sensor option (VTEMP)

For devices with Analog Option 1 (External Temperature Sensor Option not available), the following considerations are to be made:

4.2.2.3.1 Pinout considerations

Pin	Pin name for option 2	Pin name for option 1	Comment
28	VTEMP	NC	NC pin should be connected to GND
29	TSUP	NC	Pin should be left unconnected

4.2.2.3.2 Register considerations

Table 65. Acquisition control register (ACQ_CTL)

Offset [76],[77] 0x58 Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	AHCRM	OPTEM	OPENEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	0	OPTE	OPENE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
W	AHCR							

Table 65. Acquisition control register (ACQ_CTL) (continued)

Offset [76],[77] 0x58 Access: User read/write

Reset	0	0	0	0	0	0	0	0

Notes

[76] Offset related to 0x0200 for blocking access and 0x300 for non-blocking access within the global address space.

[77] This register is 16-bit access only.

For Analog Option 1 devices, ETMEN must be set to 0 (external temperature measurement disabled).

4.2.2.4 Optional 2nd external voltage sense input (VOPT)

For devices with Analog Option 1 (Optional 2nd External Voltage Sense Input not available), the following considerations are to be made:

4.2.2.4.1 Pinout considerations

Pin	Pin name for option 2	Pin name for option 1	Comment
28	VOPT	NC	NC pin should be connected to GND

4.2.2.4.2 Register considerations

Table 66. Acquisition control register (ACQ_CTL)

Offset [78],[79] 0x58 Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	AHCRM	OPTEM	OPENEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	0	OPTE	OPENE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
W	AHCR							
Reset	0	0	0	0	0	0	0	0

Notes

[78] Offset related to 0x0200 for blocking access and 0x300 for non-blocking access within the global address space.

[79] This register is 16-bit access only.

For Analog Option 1 devices, OPTE must be set to 0 (VSENSE routed to ADC).

4.3 Analog die - power, clock and resets - PCR

4.3.1 Introduction

The following chapter describes the MM912_637's system base functionality primary location on the analog die. The chapter is divided in the following sections:

1. 4.3.2, "Device operating modes"
2. 4.3.3, "Power management"
3. 4.3.4, "Wake-up sources"
4. 4.3.5, "Device clock tree"
5. 4.3.6, "System resets"
6. 4.3.7, "PCR - memory map and registers"

4.3.2 Device operating modes

The MM912_637 features three main operation modes: normal operation, stop mode, and sleep mode. The full signal conditioning and measurements are permanently running in normal operation mode. The total current consumption of the MM912_637 is reduced in the two low power modes.

The analog die of the MM912_637 is still partially active and able to monitor the battery current, temperature, activities on the LIN interface and L0 terminal, during both low power modes.

4.3.2.1 Operating mode overview

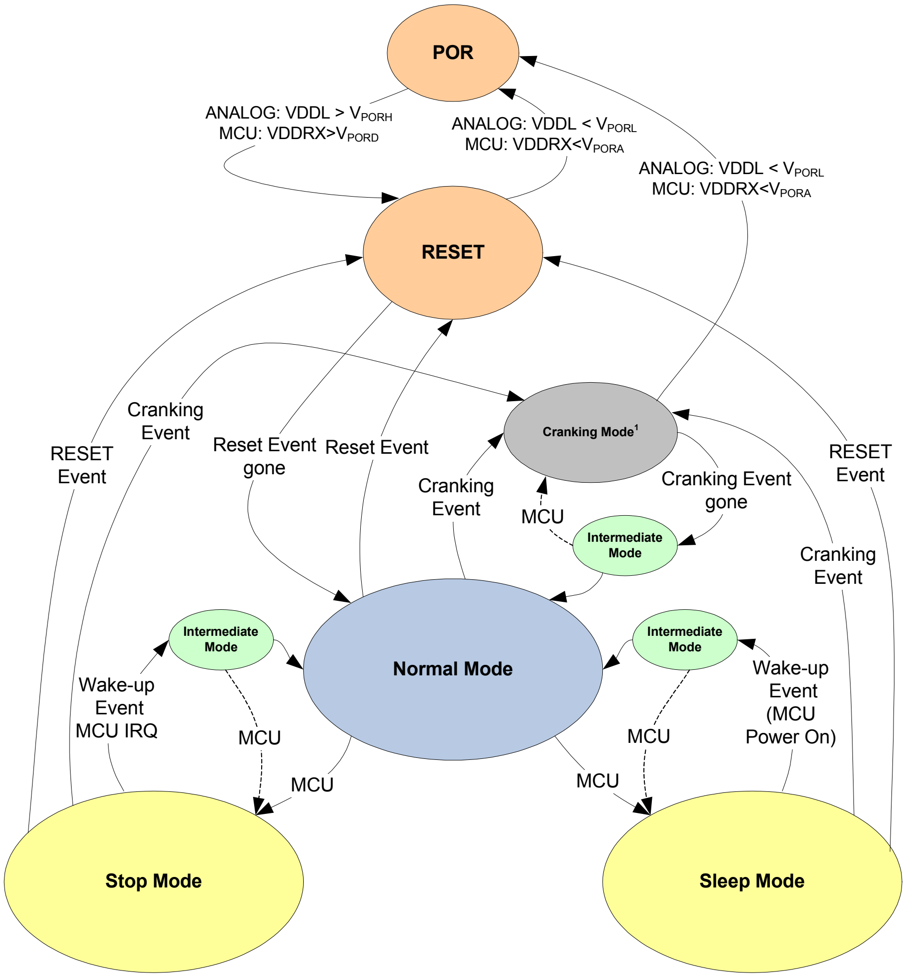
- Normal Mode
  - All device modules active
  - Microcontroller fully supplied
  - D2DCLK active analog die clock source
  - Window watchdog clocked by the low power oscillator (LPCLK) to operate on independent clock
- Stop Mode
  - MCU in low power mode, MCU regulator supply (VDDX) with reduced current capability
  - D2D interface supply disabled (VDDH=OFF)
  - Unused analog blocks disabled
  - Watchdogs = OFF
  - LIN wake-up, calibration request wake-up, cyclic wake-up, external wake-up, current threshold wake-up, and lifetime counter wake-up optional
  - Current Measurement / current averaging and temperature measurement optional
- Sleep Mode
  - MCU powered down (VDDH and VDDX = OFF)
  - Unused Analog Blocks disabled
  - Watchdogs = OFF
  - LIN wake-up, calibration request wake-up, cyclic wake-up, external wake-up, current threshold wake-up, and lifetime counter wake-up optional
  - Current measurement / current averaging and temperature measurement optional
- Intermediate Mode
  - Every transition from Stop or Sleep into Normal mode will go through an intermediate mode where the analog die clock is not yet switched to the D2D clock. If required, the MM912_637 analog die can be put back to low power mode without changing the frequency domain.
- Reset Mode
  - Every reset source within the analog die will bring the system into a Reset state
- Power On Reset Mode
  - For both low voltage thresholds are defined to indicate a loss of internal state.
- Cranking Mode[80]
  - Special Mode implemented to guarantee the RAM content being valid though very low power conditions.

Notes

[80] Not available on all device derivatives

4.3.2.2 Operating mode transitions

The device operating modes are controlled by the microcontroller, as well as external and internal wake-up sources. Figure 15 shows the basic principal.



[1]) Cranking Mode not available on all device derivatives

Figure 15. Modes of operation - transitions

4.3.2.3 Power on reset - POR

During system startup, or in any other case when MCU_VDD drops below VPORA (MCU), or VDDL drops below VPORL (analog die), a Power On Reset (POR) condition is reached. The MCU (PORF) / analog die (LVRF) will indicate this state, setting the corresponding power on reset flag. The primary consequence of entering POR is that the RAM or analog register content can no longer be guaranteed.

4.3.2.4 RESET - mode

If any of the analog die reset conditions are present, the MM912_637 analog die will enter Reset mode. During that mode, the analog die will issue the $\overline{\text{RESET\_A}}$ pin to be pulled down to reset the microcontroller die. Entering Reset mode will reset the analog die registers to their default values.

The cause of the last reset is flagged in the PCR status register (PCR_SR (hi)).

4.3.2.5 Normal mode

During Normal mode operation, all modules are operating and the microcontroller is fully supplied.

4.3.2.6 Cranking mode[81]

A specific power down behavior has been implemented to allow the MCU memory (RAM) content to be guaranteed during very low supply voltage conditions. The difference between the device behavior, with or without the cranking mode feature enabled, is described in Section 4.3.3.3, "Power up/power down behavior".

Notes

81. Not available on all device derivatives

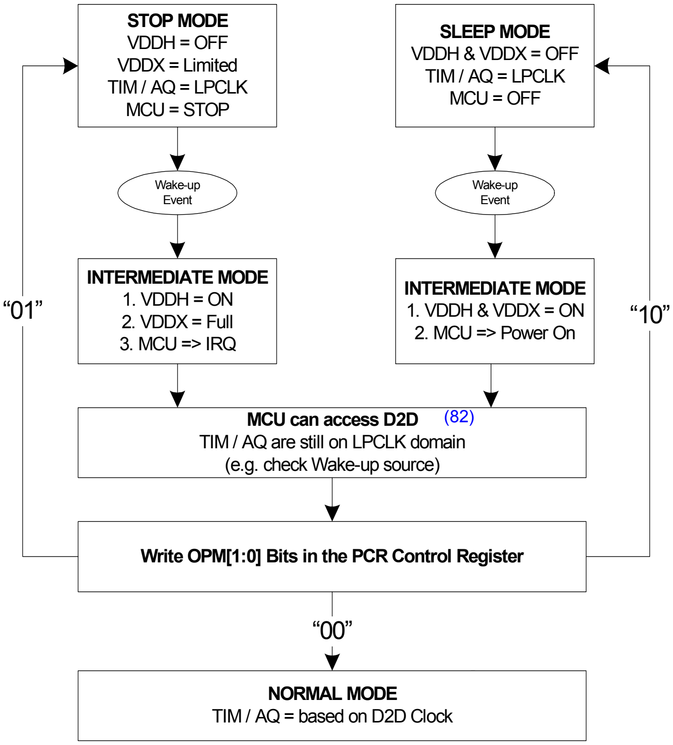
4.3.2.7 Intermediate mode

As the channel acquisition and the timer modules are switched to the LPCLK, while the MM912_637 is operating in one of the two low power modes, the Intermediate mode has been implemented, to be able to go back to low power mode without the transition into the D2D Clock domain.

NOTE

The flag indicating the last wake-up source must be cleared before re-entering low power mode!

Once awakened, the MCU instructs the analog die to transit to Normal mode by writing "00" to the OPM bits in the PCR Control Register. See Figure 16 for details.



Notes

82. As the Life Time Counter has to be configured into normal mode (no access to LTC_CNT[1..0] is possible during intermediate mode), if not reconfigured it will continue to increment starting from 0.

Figure 16. Low power mode to normal mode transition through the intermediate mode

4.3.2.8 Low-power modes

In low power mode, the MM912_637 is still active to monitor the battery current (triggered current measurement for current threshold detection and current accumulator function), and activities on the LIN interface and wake-up inputs. A cyclic wake-up using timer module is implemented for timed wake-up. Temperature measurements are optional to detect an out of calibration condition.

The Life Time counter is also incremented during Low Power mode, to issue a Wake-up on overflow. See Section 4.14, "Life time counter (LTC)" for additional details.

The average current consumption is reduced, and based on the actual low power mode, the active modules, and the wake-up timing.

NOTE

To avoid any lock condition, no analog die interrupt should be enabled or pending when entering LPM. To accomplish that condition, the analog die interrupts should be masked and served before writing the PCR_CTL register.

The MCU interrupts should be enabled right before the STOP command, to avoid any interrupt to be handled in between.

A wake-up from any of the low power modes will reset the window watchdog equal to a standard reset.

4.3.2.8.1 Sleep mode

Writing the PCR Control Register (PCR_CTL) with OPM=10, the MM912_637 will enter Sleep mode with the configured wake-up sources (see Section 4.3.4, “Wake-up sources").

NOTE

The power supply to the MCU will be turned off during Sleep mode. To safely approach this condition, the MCU should be put into a safe state (e.g STOP).

During Sleep mode, the only active voltage regulator is VDDL, supplying the low power oscillator (LPOSC), and the permanently supplied digital blocks.

When an enabled wake-up condition occurs, the shutdown voltage regulators are re-enabled, and once their outputs are above reset threshold, the $\overline{\text{RESET\_A}}$ signal is released, and the microcontroller will start its normal operation. The wake-up source is flagged in the PCR Status Register (PCR_SR (hi)).

The microcontroller has to move from intermediate mode to Normal mode, by writing the OPM=00, to allow a controlled transition into the D2D Clock domain.

4.3.2.8.2 Stop mode

Writing the PCR Control Register (PCR_CTL) with OPM=01, the MM912_637 analog die will enter Stop mode with the configured wake-up sources (see Section 4.3.4, “Wake-up sources"), after the D2DCLK signal has been stopped by the MCU die entering Stop.

NOTE

After writing the PCR Control Register (PCR_CTL) with OPM=01, the register content of the SCI (S08SCIV4) and TIMER (TIM16B4C) module registers are read only until Normal mode is entered again. This is important in case the MCU does not effectively enter STOP, due to an IRQ pending from one of the two blocks. (Having any analog die IRQ allowed when entering Low Power mode is not recommended).

During Stop mode, the MM912_637 has the same behavior as during Sleep mode, except VDDX is still powered by the internal Clamp_5v, to supply the MCU STOP mode current. As this current is limited, the MCU die must be switched into STOP mode after sending the Stop command for the analog die.

If any enabled wake up condition occurs, the shutdown voltage regulators are re-enabled, and once their outputs are above the reset threshold, VDDX is switched to the main regulator, an D2D interrupt (D2DINT) is issued to wake-up the MCU, and the microcontroller will continue its normal operation. The wake-up source is flagged in the PCR Status Register (PCR_SR (hi)).

The microcontroller has to acknowledge the Normal mode by writing the OPM=00. This allows a controlled transition into the D2D Clock domain. If the clock domain transition is not required, the microcontroller may issue a sleep / stop mode entry instead (see Section 4.3.5, “Device clock tree" for details on the limitations during the intermediate state).

At start up or after wake-up, it is required to wait to ensure the PLL is locked, if the PLL is enabled previous to accessing the analog die through the D2D interface.

NOTE

After writing the PCR Control Register (PCR_CTL) with OPM=01, writing OPM=00 (Normal mode) is allowed to wake-up the analog die. The reduced current capability of the MCU regulator supply (VDDX) has to be considered.

4.3.3 Power management

To support the various operating modes and modules in the MM912_637, the following power management architecture has been implemented.

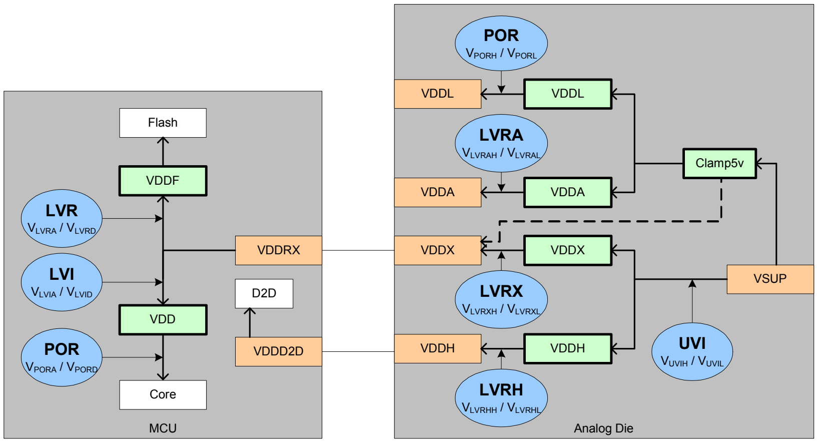


Figure 17. System voltage monitoring

4.3.3.1 Detailed power block description

See recommended external components under Section 2.2, “Recommended external components"”.

4.3.3.1.1 VSUP

VSUP is the system power supply input, and must be reverse battery protected by an external diode. VSUP is monitored for undervoltage conditions (UVI). Once VSUP drops below V_{UVIL} an undervoltage interrupt (LVI) is issued.

NOTE

If the device has the cranking mode feature enabled, the undervoltage threshold would be V_{UVCIL} instead of V_{UVIL}.

4.3.3.1.2 VDDL

VDDL is the low power 2.5 V digital supply voltage, supplying the permanently active blocks. It is based on the internal Clamp5v voltage and always on. It is available externally, but must not be connected to any load.

4.3.3.1.3 VDDX

VDDX is the Normal mode 5.0 V regulator output, suppling the LIN block and the microcontroller via the VDDX pin. During STOP and SLEEP mode operation, the VDDX regulator is shut down (Clamp5v does supply the MCU during STOP mode).

4.3.3.1.4 VDDH

VDDH is the Normal mode 2.5 V regulator output, suppling only active blocks during Normal mode and the MCU Die to Die Interface, via the VDDH terminal. The VDDH regulator is shut down during both low power modes.

4.3.3.1.5 VDDA

VDDA is the 2.5 V analog supply voltage, active during Normal mode and I/T acquisitions. No external load must be connected to the VDDA terminal.

4.3.3.2 Power supply by module

The following table summarized the active regulators vs. module for the different operating modes.

Table 67. Power supply by module

Module/block	VDDH	VDDA	VDDL	VDDX
Gain Control Block (GCB)(83)	X	X		
Programmable Gain Amplifier (PGA)(84)		X		
I/T - ADC Converters(84)		X		
V - ADC Converters(83)		X		
Temperature Sensor(84)		X		
LIN(83)			X	X
D2D(83)	X			
LPOSC(85)			X	
Permanent Digital(85)			X	
Normal Mode Digital(83)	X			

Notes

83. Enabled in Normal Mode only
84. Enabled when a measuring in Low Power mode and always in Normal mode
85. Permanently enabled

4.3.3.3 Power up/power down behavior

Several system voltage monitors have been implemented in both die, to guarantee a defined power up and power down system behavior. See Figure 17 for the various sensing points. The individual threshold levels are specified in Table 15 for the analog die, and Table 24 for the microcontroller.

NOTE

To differentiate between the MCU and analog die thresholds, the following symbol scheme is defined:

V_{xxxxA} - MCU Assert Level (lower threshold for low voltage events)
V_{xxxxD} - MCU Deassert Level (higher threshold for low voltage events)
V_{xxxxH} - Analog Die High Threshold Level (deassert threshold for low voltage events)
V_{xxxxL} - Analog Die Low Threshold Level (assert threshold for low voltage events)

4.3.3.4 Low-voltage operation - cranking mode device option

Based on the device option (“Cranking” or “Non-cranking”), the MM912_637 will behave different during “Loss of Power” conditions. The “Cranking” option is an option, allowing lower voltage operations to guarantee the MCU memory content during a standard cranking situation.

As illustrated in Figure 18, the cranking mode is introduced to maintain both die in a STOP mode alike state. The MCU die will remain in STOP with the RAM content being guaranteed until the PORA level is reached for the VDDRX supply.

The analog die will enter “Cranking Mode” upon the MCU command out of Normal Mode, or when it reaches V_{UVCIL} during STOP Mode, with the LVT bit set in the TRIM_LVT register.

NOTE

Executing STOP with VSUP < V_{UVCIL} and LVT = 1, the MM912_637 will immediately enter Cranking Mode.

During Cranking Mode, the analog die will gate its internal oscillator to stop all ongoing acquisitions during the low power condition. Returning from Cranking mode will appear as a wake-up from undervoltage interrupt (UVI=1). The analog die will be in Intermediate mode after wake-up, and could be sent into Normal mode (Stop, Sleep), by writing the OPM bits.

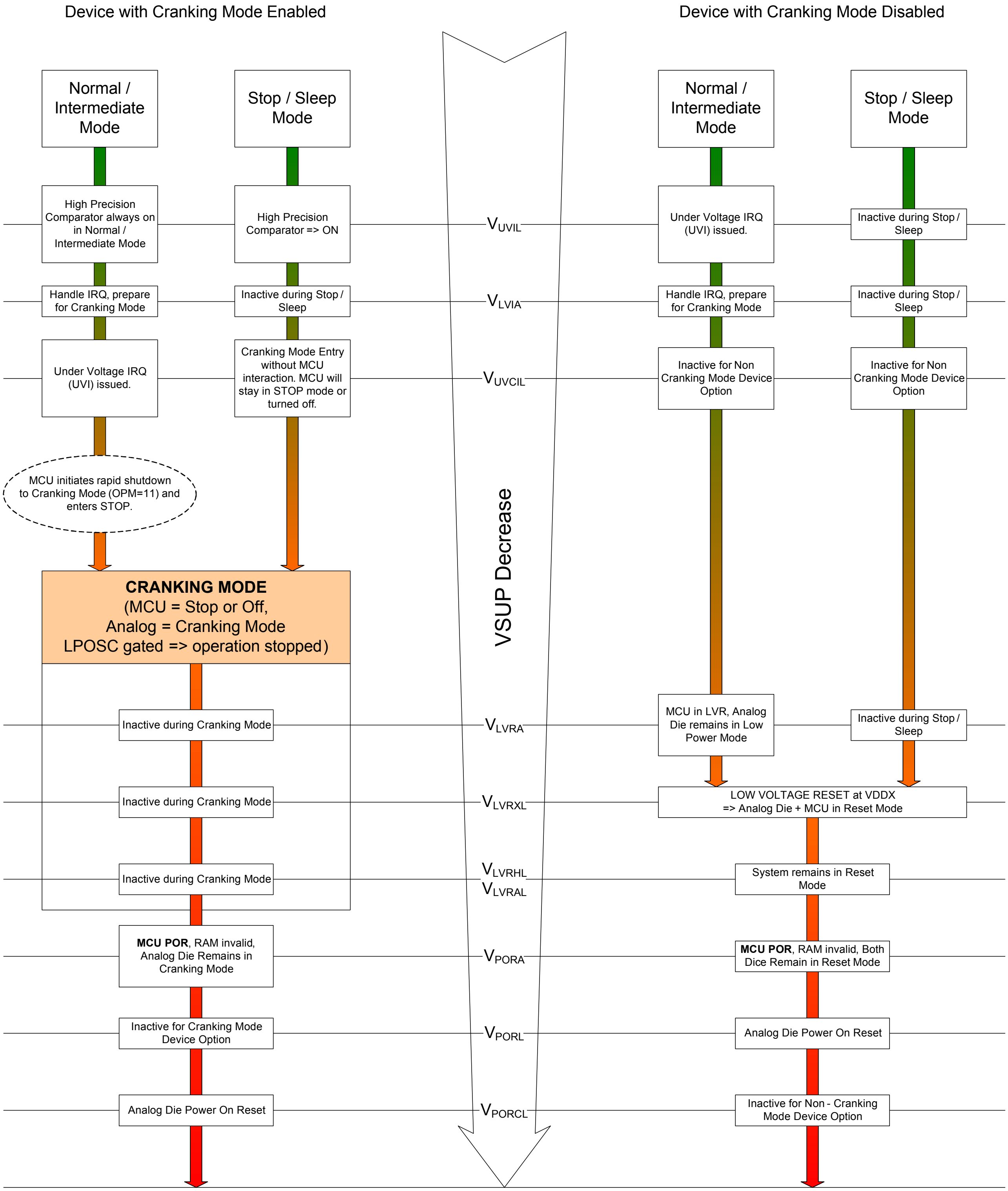

Device with Cranking Mode Enabled
Device with Cranking Mode Disabled
Normal / Intermediate Mode
Stop / Sleep Mode
Normal / Intermediate Mode
Stop / Sleep Mode
High Precision Comparator always on in Normal / Intermediate Mode
High Precision Comparator => ON
V_{UVIL}
Under Voltage IRQ (UVI) issued.
Inactive during Stop / Sleep
Handle IRQ, prepare for Cranking Mode
Inactive during Stop / Sleep
V_{LVIA}
Handle IRQ, prepare for Cranking Mode
Inactive during Stop / Sleep
Under Voltage IRQ (UVI) issued.
Cranking Mode Entry without MCU interaction. MCU will stay in STOP mode or turned off.
V_{UVCIL}
Inactive for Non Cranking Mode Device Option
Inactive for Non Cranking Mode Device Option
MCU initiates rapid shutdown to Cranking Mode (OPM=11) and enters STOP.
VSUP Decrease
CRANKING MODE
(MCU = Stop or Off,
Analog = Cranking Mode
LPOSC gated => operation stopped)
Inactive during Cranking Mode
V_{LVRA}
MCU in LVR, Analog Die remains in Low Power Mode
Inactive during Stop / Sleep
Inactive during Cranking Mode
V_{LVRXL}
LOW VOLTAGE RESET at VDDX => Analog Die + MCU in Reset Mode
Inactive during Cranking Mode
V_{LVRHL}
V_{LVRAL}
System remains in Reset Mode
MCU POR, RAM invalid, Analog Die Remains in Cranking Mode
V_{PORA}
MCU POR, RAM invalid, Both Dice Remain in Reset Mode
Inactive for Cranking Mode Device Option
V_{PORL}
Analog Die Power On Reset
Analog Die Power On Reset
V_{PORCL}
Inactive for Non - Cranking Mode Device Option

Figure 18. Power down sequence

4.3.4 Wake-up sources

Several wake-up sources have been implemented in the MM912_637, to exit from Sleep or Stop mode. Figure 19 shows the wake-up sources and the corresponding configuration and status bits.

To indicate the internal wake-up signal, a routing of the internal wake-up signal to the PTBx output (WKUP) is implemented. See Section 4.11, “General purpose I/O - GPIO", for additional details on the required configuration.

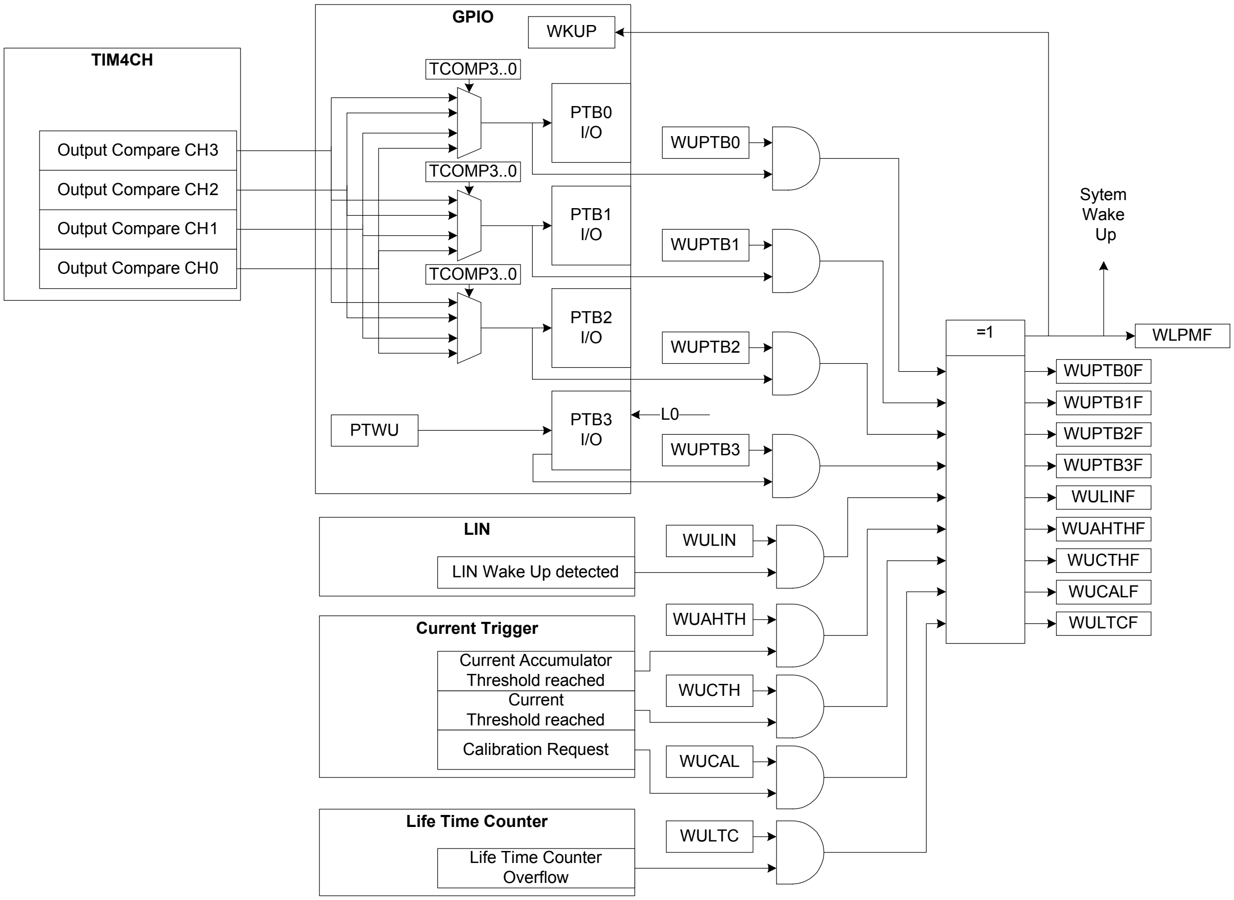


Figure 19. Wake-up sources

4.3.4.1 Wake-up source details

4.3.4.1.1 Cyclic current acquisition/calibration temperature check

A configurable (ACQ_TCMP) independent low power mode counter/trigger, based on the ALFCLK, has been implemented to trigger a cyclic current measurement during the low power modes. To validate that the temperature is still within the calibration range, the temperature measurement can be enabled during this event as well.

As a result of the cyclic conversions, three wake-up conditions are implemented.

- Current Threshold Wake-up
- Current Averaging Wake-up
- Calibration Request Wake-up

The configuration of the counter and the cyclic measurements is part of the acquisition paragraph (see Section 4.8, “Channel acquisition"). The actual cyclic measurement does not wake-up the microcontroller unless one of the three wake-up conditions become valid.

4.3.4.1.1.1 Current threshold wake-up

Every cyclic current measurement result (absolute content of the ADC result I_CURR register) is compared with a programmable unsigned current threshold (CTH in the ACQ_CTH register).

The comparison is done with the CTH content left - shifted by 1, as shown in Table 68.

Table 68. Current Threshold Comparison

	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTH[7:0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CTH[7:0]								0
ABS(CURR[23:0])	X	ABS(CURR[23:0])																						

If the absolute result is greater or equal to the programmed and shifted threshold, a filter counter is incremented (decremented if below). If the filter counter (8-Bit) reaches the programmable low power current threshold filtering period (ACQ_THF), a wake-up initiated if the Current Threshold Wake-up is enabled (WUCTH). The filter counter is reset every time a low power mode is entered. The implementation is shown in Figure 20.

The wake-up source is flagged with the WUCTHF Bit.

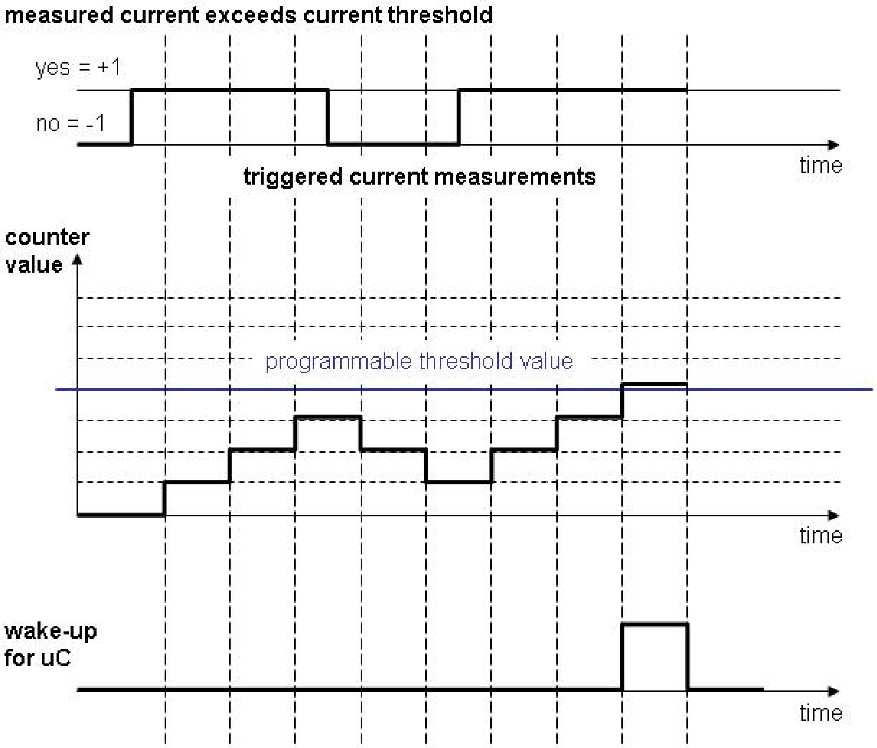

measured current exceeds current threshold
yes = +1
no = -1
time
triggered current measurements
counter value
programmable threshold value
time
wake-up for uC
time

Figure 20. Current threshold - wake-up counter

4.3.4.1.1.2 Current ampere hour threshold wake-up

As shown in Figure 21, every cyclic current measurement (signed content of the ADC result ACQ_CURR register) is added to the 32-Bit (signed) current accumulator (ACQ_AHC) (both in two’s complement format). If the absolute accumulator value reaches (|ACQ_AHC| ≥ ACQ_AHTH), the absolute programmable 31-Bit current threshold (ACQ_AHTH), a wake-up is initiated if the Current AH Threshold Wake-up is enabled (WUAHTH). The accumulator could be reset by writing 1 into the AHCR register.

The Ampere Hour Counter is counting after wake-up. In normal mode, the accumulator register ACQ_AHC can be read out anytime. The wake-up source is flagged with the WUAHTHF Bit.

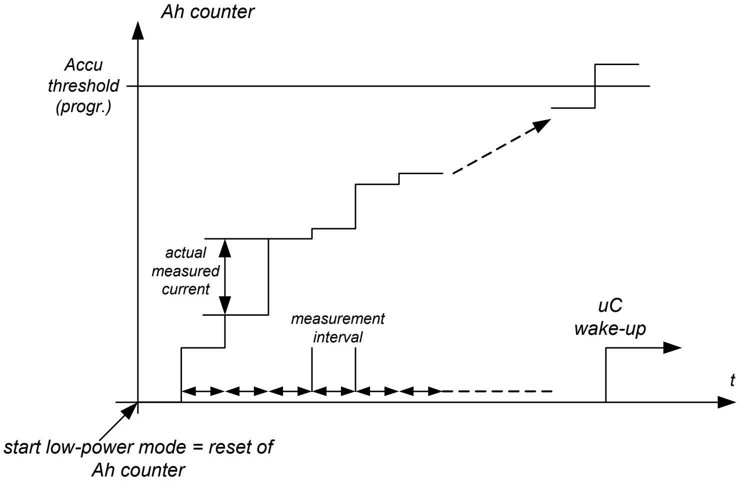


Figure 21. Ah counter function

4.3.4.1.1.3 Calibration request wake-up

Once the temperature measured during the cyclic sense is indicating a potential “out of calibration” situation, a wake-up is issued if the Calibration Request Wake-up is enabled (WUCAL). For additional details, refer to Section 4.8.5, “Calibration". The wake-up source is flagged with the WUCALF Bit.

4.3.4.1.2 Timed wake-up

To generate a programmable wake-up timer, the integrated 4 Channel Timer Module is supplied, during both low power modes and running on the ALFCLK clock. To wake-up from one of the low power modes, the output compare signal (OC) of any of the 4 channels can be routed to the PTB[2:0] logic (standard feature also in Normal mode). Enabling the corresponding Wake-up Enable Bit (WUPTBx) will generate the wake up, once the timer output compare becomes active.

NOTE

Only the internal GPIO logic is active during the low power modes. The Port I/O structures will not be active.

To allow an accurate wake-up configuration during the clock transition, the timer should be configured before entering one of the low power modes, without the Timer Enable Bit (TEN) being set. Setting the Timer Wake-up Enable Bit (WUPTB) will enable the TIMER interrupts as wake-up sources, and cause the Timer Enable Bit (TEN) to be set, once the timer clock domain was changed to the ALFCLK clock supplied by LPOSC.

During low-power mode, only current and temperature measurements are performed, so only the current measurement channel is active with the temperature channel being optional - the voltage measurement channel is inactive. To reduce further the power consumption, only triggered current measurements are done. For this purpose, an independent Timer Module is used to periodically start a current measurement after a programmable time (ACQ_TCMP).

4.3.4.1.3 Wake-up from LIN

During Low Power mode, operation of the transmitter of the physical layer is disabled. The receiver remain, active and able to detect wake-up events on the LIN bus line. For further details, refer to Section 4.12, “LIN". A dominant level longer than t_{WUPF} followed by a rising edge, will generate a wake-up event if the WULIN is enabled. The wake-up source is flagged with the WULINF Bit.

NOTE

If the LIN module is disabled (LIN_CTL:EN=0), no wake-up will be issued after the dominant to recessive transition, when the device goes to low power mode, while the LIN bus is in the DOMINANT STATE.

If the LIN module is enabled (LIN_CTL:EN=1), the device will wake-up after the dominant to recessive transition, when the device goes to low power mode, while the LIN bus is in the DOMINANT STATE.

A full dominant -> recessive -> dominant sequence, during low power mode, will wake-up the device in both cases.

4.3.4.1.4 Wake-up on wake-up pin high level

Once a Wake-up signal (high level) is detected on the PTB3/L0 input, with the Wake-up Enable Bit (WUPTB3) and the port configuration bit (PTWU) set, a wake-up is issued. The wake-up source is flagged with the WUPTB3F Bit.

4.3.4.1.5 Wake-up on life time counter overflow

The life time counter continues to run during low power mode, if configured. Once the counter overflows with the life time counter wake-up enabled (WULTC=1), a wake-up is issued. The wake-up source is flagged with the WULTC Bit. Life Time Counter has to be configured in Normal mode only.

4.3.4.1.6 General wake-up indicator

To indicate the system has been awakened after power up, the WLPMF flag will be set.

4.3.5 Device clock tree

4.3.5.1 Clock scheme overview

There are two system oscillators implemented. The low power oscillator is located on the analog die, and is supplied permanently and has a nominal frequency of f_{OSCL} (512 kHz), providing a LPCLK clock signal. It is primarily used in low power mode, and as an independent clock source for the watchdog during Normal mode.

The high power oscillator is basically the internal or external microcontroller oscillator (active only during normal mode). The high power oscillator is distributed to the analog die via the D2DCLK (via configurable MCU prescalers), and there it’s divided into two clocks (D2DSCLK and D2DFCLK), based on the PRESC[15:0] prescaler. For the D2DSCLK, an additional 2 Bit divider PF[1:0] is implemented [86]. During Normal mode, D2DSCLK is continuously synchronizing the LPCLK, to create the accurate ALFCLK (See Section 4.3.5.2, “ALFCLK calibration"), D2DCLK is the clock source of the TIM16B4C (Timer), and S08SCIV4 (SCI) module with a fixed by 4 divider.

Notes

[86] PF[1:0] is not implemented as a simple divider. To accomplish a D2DSCLK period ranging from 1.0 ms to 8.0 ms, the following scheme is used: 00 - 1; 01 - 2; 10 - 4; 11 - 8.

D2DSCLK - D2D Slow Clock (1... 0.125 kHz) ***Eqn. 1***

$$\left(D2DSCLK = \frac{D2DCLK}{(2^{PF[1,0]}) \times (PRESC[15,0])}\right)$$

D2DFCLK - D2D Fast Clock (512 kHz) ***Eqn. 2***

$$\left(D2DFCLK = \frac{D2DCLK}{2 \times (PRESC[15,10] + PRESC[9])}\right)$$

During low power mode, D2DCLK is not available. The low power oscillator is the only system clock. Figure 22 and Figure 23 show the different clock sources for normal and low power mode.

NOTE

D2DFCLK has to be set to match 512 kHz, resulting in D2DSCLK being 1.0, 2.0, 4.0, or 8.0 kHz, based on PF[1:0]

The minimum value for PRESC[15:0] has to be 0x0400. Any value lower than 0x0400 will result in faulty behavior and is not recommended. Values of 0x0003 or less are not stored by the internal logic.

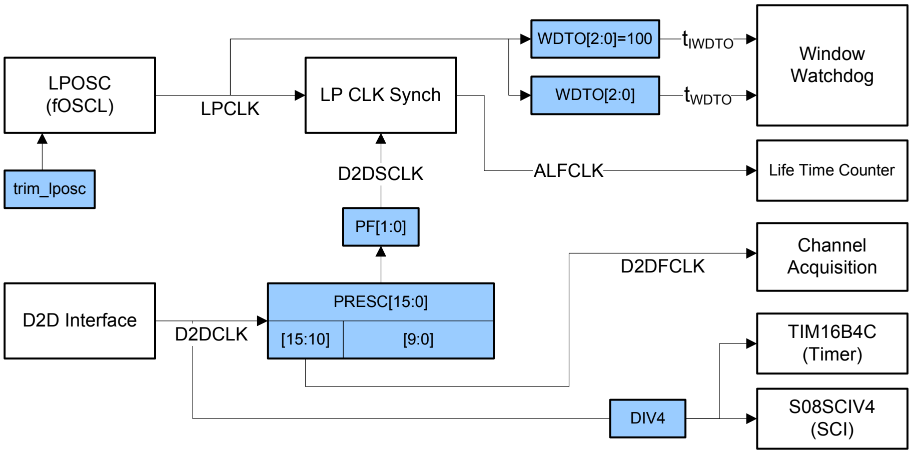


Figure 22. Clock tree overview - normal mode

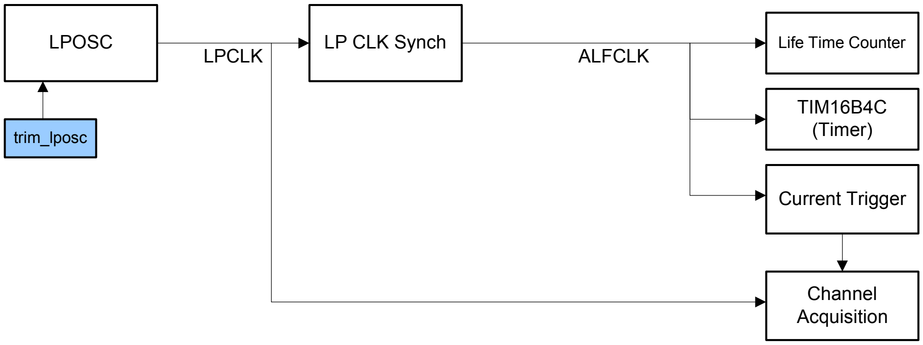


Figure 23. Clock tree overview - low power modes

4.3.5.2 ALFCLK calibration

To increase the accuracy of the 1.0 kHz (or 2.0, 4.0, 8.0 kHz based on PF[1:0]) system clock (ALFCLK), the low power oscillator (LPCLK) is synchronized to the more precise D2DCLK, via the D2DSCLK signal. The “Calibrated Low Power Clock” (ALFCLK) could be trimmed to the D2DCLK accuracy plus a maximum error adder of 1 LPCLK period, by internally counting the number of periods of the LPCLK (512 kHz) during a D2DSCLK period. The APRESC[12:0] register will represent the calculated internal prescaler. The PRDF bit (Prescaler Ready flag) will indicate the synchronization complete after a power up or prescaler (PRESC/PF) change.

The adjustment is continuously performed during Normal mode. During low power mode (STOP or SLEEP), the last adjustment factor would be used.

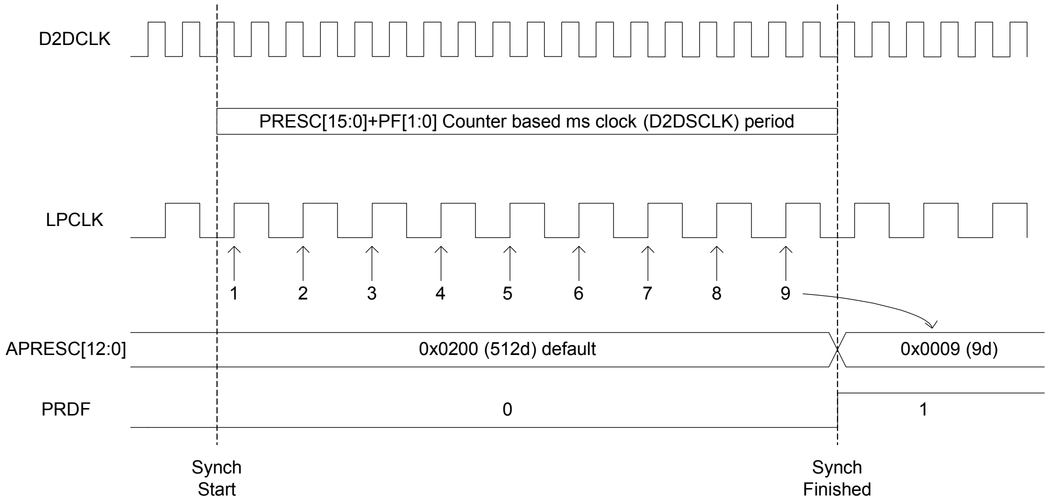


Figure 24. ALF clock calibration procedure during normal mode

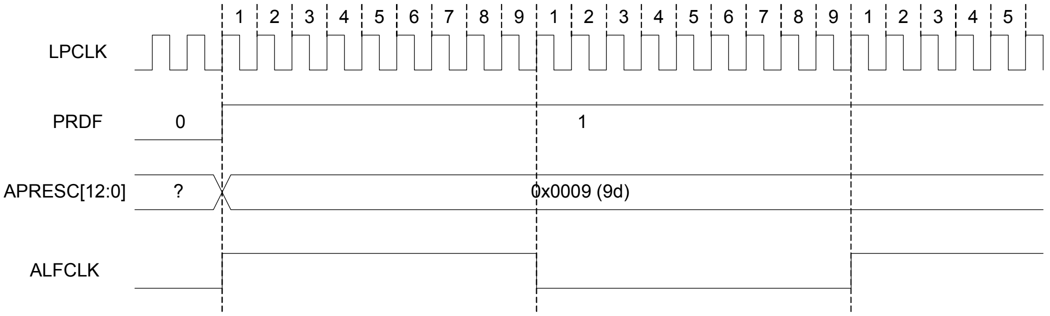


Figure 25. ALFCLK after calibration

4.3.5.3 Recommended clock settings

Considering the system is running on the internal oscillator, Table 69 shows the recommended clock settings to achieve the optimal 512 kHz D2DFCLK. For details on the MCU divider settings, including POSTDIV and SYNDIV, see Section 4.23, “S12 clock, reset, and power management unit (9S12I128PIMV1)". The D2D initiator module includes D2DCLKDIV see Section 4.26, “MCU - die-to-die initiator (9S12I128PIMV1)".

Table 69. Recommended clock settings

f_{D2D} / MHz				POSTDIV for (SYNDIV=fVCO in MHz)																	Divider for[87] D2DFCLK=512kHz	PRESC[15:9] (dec)[87]
D2DCLKDIV=1 (f_{BUS})	D2DCLKDIV=2	D2DCLKDIV=3	D2DCLKDIV=4	31=65.536	30=63.488	29=61.440	28=59.392	27=57.344	26=55.296	25=53.248	24=51.200	23=49.152	22=47.104	21=45.056	20=43.008	19=40.960	18=38.912	17=36.864	16=34.816	15=32.768		
32.768	16.384		8.192	0																	64	63;64
31.744					0																62	61;62
30.720	15.360	10.240				0															60	59;60
29.696							0														58	57;58
28.672	14.336		7.168					0													56	55;56
27.648		9.216							0												54	53;54
26.624	13.312									0											52	51;52

Table 69. Recommended clock settings (continued)

f_{D2D} / MHz				POSTDIV for (SYNDIV=fVCO in MHz)																	Divider for[87] D2DFCLK=512kHz	PRESC[15:9] (dec)[87]
D2DCLKDIV=1 (f_{BUS})	D2DCLKDIV=2	D2DCLKDIV=3	D2DCLKDIV=4	31=65.536	30=63.488	29=61.440	28=59.392	27=57.344	26=55.296	25=53.248	24=51.200	23=49.152	22=47.104	21=45.056	20=43.008	19=40.960	18=38.912	17=36.864	16=34.816	15=32.768		
25.600											0										50	49;50
24.576	12.288	8.192	6.144									0									48	47;48
23.552													0								46	45;46
22.528	11.264													0							44	43;44
21.504		7.168													0						42	41;42
20.480	10.240		5.120													0					40	39;40
19.456																	0				48	47;48
18.432	9.216	6.144																0			36	35;36
17.408																			0		34	33;34
16.384	8.192		4.096	1																0	32	31;32
15.360		5.120				1															30	29;30
14.336	7.168							1													28	27;28
13.312										1											26	25;26
12.288	6.144	4.096	3.072									1									24	23;24
11.264														1							22	21;22
10.240	5.120					2										1					20	19;20
9.216		3.072							2									1			18	17;18
8.192	4.096		2.048	3								2								1	16	15;16
7.168								3							2						14	13;14
6.144	3.072	2.048				4						3						2			12	11;12
5.120						5					4					3					10	9;10
4.096	2.048			7				6				5				4				3	8	7;8
3.072						9			8			7			6			5			6	5;6
2.048				15		14		13		12		11		10		9		8		7	4	4

Notes

[87] For D2DCLKDIV=1

4.3.6 System resets

To guarantee safe operation, several RESET sources have been implemented in the MM912_637 device. Both the MCU and the analog die are designed to initiate reset events on internal sources and the MCU is capable of being reset by external events including the analog die reset output. The analog die can be reset through RESETA in stop and cranking mode only. In normal mode, the MCU can reset the analog die only by writing 1 into HWR bit (remind the mask bit HWRM).

4.3.6.1 Device reset overview

The MM912_637 reset concept includes two external reset signals, RESET (MCU) and RESET_A (analog Die). Figure 26 illustrates the general configuration.

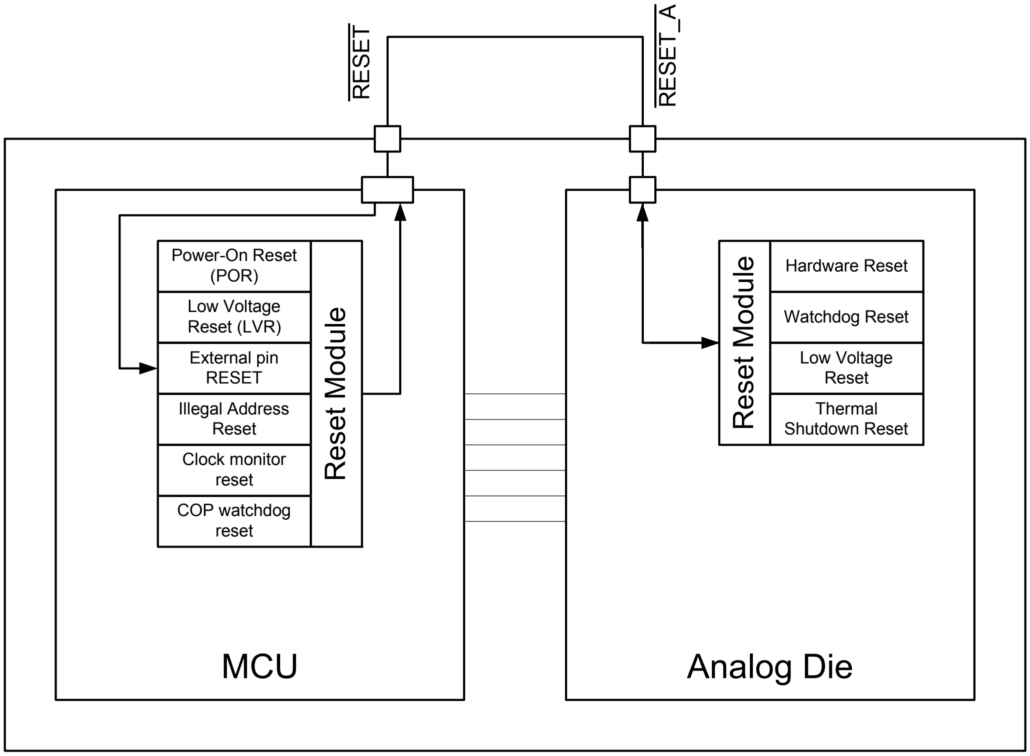


Figure 26. Device reset overview

Both RESET and RESET_A signals are low active I/Os, based on the 5.0 V supply (VDDRX for RESET and VDDX for RESET_A).

4.3.6.2 Analog die reset implementation

There are 7 internal reset sources implemented in the analog die of the MM912_637 that causing the internal analog die status to be reset to default (Internal analog RST), and to trigger an external reset, activating the RESET_A pin. In addition, during stop and cranking mode, an external reset at the RESET_A pin will also reset the analog die.

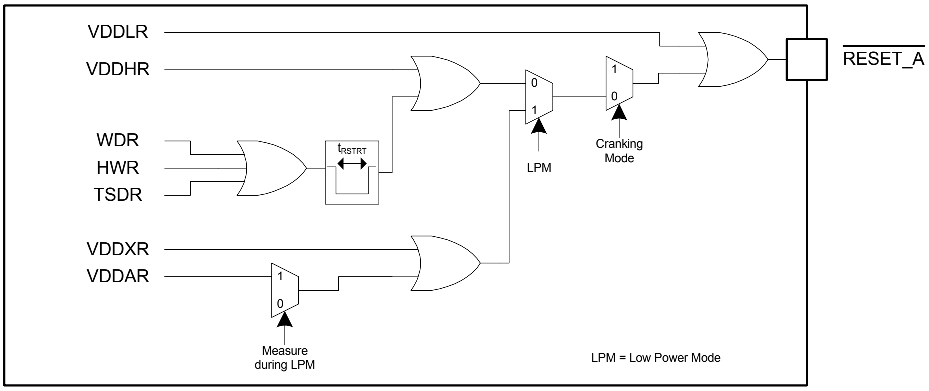


Figure 27. Analog die reset implementation

The WDR and HWR will issue a reset on the $\overline{\text{RESET\_A}}$ pin during t_{RSTRT} (see Table 29). The other reset VDDLR, VDDHR, VDDXR, VDDAR, and TSDR will drive the $\overline{\text{RESET\_A}}$ pin as long as the condition is present. During cranking mode [88], only the VDDLR is active. During Low Power modes, only VDDXR and VDDAR are active reset sources. VDDAR is only active during active measurement in LPM. VDDXR and VDDAR are not active in Normal mode.

Notes

[88] Not available on all device derivatives

4.3.6.3 Reset source summary

- HWR - Hardware Reset
  - — Forced internal reset caused by writing the HWR bit in the PCR_CTL register. The source will be indicated by the HWRF bit.
- WDR - Watchdog Reset
  - — Window watchdog failure. The source will be indicated by the WDRF bit.
- LVR - Low Voltage Reset
  - — The Voltage at the VDDL, VDDH, VDDX, or VDDA has dropped below its reset threshold level. The source will be indicated for the VDDL by the LVRF + HVRF, for the VDDA by the AVRF, and for the VDDH by the HVRF bit. VDDX resets are not indicated via individual reset flags. See Figure 27 for dependencies.
- TSDR - Temperature Shutdown Reset
  - — The critical shutdown temperature threshold has been reached. VDDA, VDDX, and VDDH will be disabled as long as the overtemperature condition is pending[89] and the reset source is indicated by the HTF bit.
- External Reset
  - — During stop and cranking[89] mode, a low signal at the $\overline{\text{RESET\_A}}$ pin will reset the analog die. Since this condition can only be initiated by the microcontroller, no specific indicator flag is implemented.

Notes

[89] Resulting in a VDDH Low Voltage Reset taking over the reset after the 2 LPCLK reset pulse

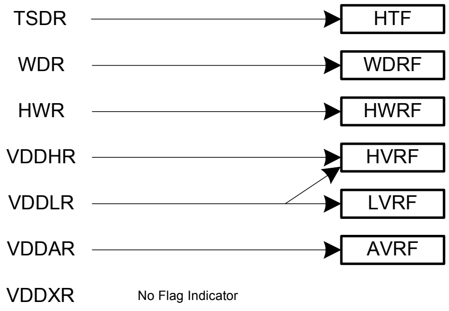


Figure 28. Reset status information

4.3.7 PCR - memory map and registers

4.3.7.1 Overview

This section provides a detailed description of the memory map and registers.

4.3.7.2 Module memory map

The memory map for the Analog Die - Power, Clock and Resets - PCR module is given in Table 62

Table 70. Module Memory Map

Offset [90],[91]	Name		7	6	5	4	3	2	1	0
0x00	PCR_CTL PCR Control Register	R	0	0	0	0	0	0	0	0
		W	HTIEM	UVIEM	HWRM	0	PFM[1:0]		OPMM[1:0]	
		R	HTIE	UVIE	0	0	PF[1:0]		OPM[1:0]	
		W			HWR	0				
0x02	PCR_SR (hi)	R	HTF	UVF	HWRF	WDRF	HVRF	LVRF	WULTCF	WLPMF
	PCR Status Register	W	Write 1 will clear the flags							
0x03	PCR_SR (lo)	R	WUAHTHF	WUCTHF	WUCALF	WULINF	WUPTB3F	WUPTB2F	WUPTB1F	WUPTB0F
	PCR Status Register	W	Write 1 will clear the flags							
0x04	PCR_PRESC PCR 1.0 ms prescaler	R	PRESC[15:0]							
		W								
		R								
		W								
0x06	PCR_WUE (hi)	R	WUAHTH	WUCTH	WUCAL	WULIN	WUPTB3	WUPTB2	WUPTB1	WUPTB0
	Wake-up Enable Register	W								
0x07	PCR_WUE (lo)	R	WULTC	0	0	0	0	0	0	0
	Wake-up Enable Register	W								
0x0E	TRIM_ALF (hi)	R	PRDF	0	0	APRESC[12:8]				
	Trim for accurate 1.0 ms low freq clock	W								
0x0F	TRIM_ALF (lo)	R	APRESC[7:0]							
	Trim for accurate 1.0 ms low freq clock	W								

Notes

[90] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[91] Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

4.3.7.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

4.3.7.3.1 PCR control register (PCR_CTL)

Table 71. PCR control register (PCR_CTL)

Offset [92], [93] 0x00 Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	HTIEM	UVIEM	HWRM	0	PFM[1:0]		OPMM[1:0]	
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	HTIE	UVIE	0	0	PF[1:0]		OPM[1:0]	
W			HWR	0				
Reset	0	0	0	0	0	0	0	0

Notes

[92] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[93] Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

Table 72. PCR control register (PCR_CTL) - register field descriptions

Field	Description
15 HTIEM	High temperature interrupt enable mask 0 - writing the HTIE bit will have no effect 1 - writing the HTIE bit will be effective
14 UVIEM	Supply undervoltage interrupt enable mask 0 - writing the UVIE bit will have no effect 1 - writing the UVIE bit will be effective
13 HWRM	Hardware reset mask 0 - writing the HWR bit will have no effect 1 - writing the HWR bit will be effective
12 Reserved	Reserved. Must remain “0”
11-10 PFM[1:0]	Prescaler factor mask 00,01,10 - writing the PF bits will have no effect 1 - writing the PF bits will be effective
9-8 OPMM[1:0]	Operation mode mask 00,01,10 - writing the OPM bits will have no effect 11 - writing the OPM bits will be effective
7 HTIE	High Temperature Interrupt enable. Writing only effective with corresponding mask bit HTIEM set. 0 - High temperature interrupt (HTI) enabled 1 - High temperature interrupt (HTI) disabled
6 UVIE	Low supply voltage interrupt enable. Writing only effective with corresponding mask bit UVIEM set. 0 - Low supply voltage interrupt (UVI) enabled 1 - Low supply voltage interrupt (UVI) disabled
5 HWR	Hardware Reset. Writing only effective with corresponding mask bit HWRM set. Write only. 0 - No effect 1 - All analog die digital logic is reset and external reset ($\overline{\text{RESET\_A}}$) is set to reset the MCU.
4 Reserved	Reserved. Must remain “0”
3-2 PF[1:0]	1.0 ms Prescaler. Writing only effective with corresponding mask bits PFM set to 11. 00 - 1 01 - 2 10 - 4 11 - 8
1-0 OPM[1:0]	Operation mode select. Writing only effective with “11” mask bits OPMM set to 11. 00 - Normal mode 01 - Stop mode 10 - Sleep mode 11 with Cranking feature disabled - same effect as 01 (STOP mode) 11 with Cranking feature enabled - Cranking mode

4.3.7.3.2 PCR status register (PCR_SR (hi))

Table 73. PCR status register (PCR_SR (hi))

Offset[94] 0x02 Access: User read/write

	7	6	5	4	3	2	1	0
R	HTF	UVF	HWRF	WDRF	HVRF	LVRF	WULTCF	WLPMF
W	Write 1 will clear the flags[95]							
Reset	0	0	0/1	0/1	0/1	0/1	0	0

Notes

[94] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[95] HTF and UVF represent the current status and cannot be cleared. Writing 1 to HTF / UVF will clear the Interrupt flag in the Interrupt Source Register and Interrupt Vector Register instead.

Table 74. PCR status register (PCR_SR (hi)) - register field descriptions

Field	Description
7 HTF	High Temperature Condition Flag. This bit is set once a temperature warning is detected, or the last reset being caused by a temperature shutdown event (TSDR). Writing HTF=1 will clear the flag and the interrupt flag in the Interrupt Source Register and Interrupt Vector Register, if the condition is gone. 0 - No High Temperature condition detected. 1 - High Temperature condition detected or last reset = TSDR.
6 UVF	Supply Undervoltage Condition Flag. This bit is set once a undervoltage warning is detected. Writing UVF=1 will clear the flag and the Interrupt flag in the Interrupt Source Register and Interrupt Vector Register, if the condition is gone (UVF=0). 0 - No undervoltage condition detected. 1 - Undervoltage condition detected.
5 HWRF	Hardware Reset Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a HWR command.
4 WDRF	Watchdog Reset Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by the analog die window watchdog.
3 HVRF	VDDH Low Voltage Reset Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a low voltage condition at the VDDH regulator. (LVRF = 0) 1 - Last reset was caused by a low voltage condition at the VDDL regulator. (LVRF = 1)
2 LVRF	VDDL Low Voltage (POR) Reset Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a low voltage condition at the VDDL regulator. (Power on Reset - POR)
1 WULTCF	Life Time Counter Wake-up Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last Wake-up was caused by a life time counter overflow
0 WLPMF	Wake-up after Low Power Mode Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after Low Power mode.

4.3.7.3.3 PCR status register (PCR_SR (lo))

Table 75. PCR status register (PCR_SR (lo))

Offset[96] 0x03

Access: User read/write

	7	6	5	4	3	2	1	0
R	WUAHTHF	WUCTHF	WUCALF	WULINF	WUPTB3F	WUPTB2F	WUPTB1F	WUPTB0F
W	Write 1 will clear the flags							
Reset	0	0	0	0	0	0	0	0

Notes

96. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 76. PCR status register (PCR_SR (lo)) - register field descriptions

Field	Description
7 WUAHTHF	Wake-up on Ah counter threshold Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after Ah counter threshold reached.
6 WUCTHF	Wake-up on current threshold Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after current threshold reached.
5 WUCALF	Wake-up on calibration request flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after calibration request.
4 WULINF	Wake-up on LIN flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after LIN wake-up detected
3 WUPTB3F	Wake-up on GPIO 3 event (L0 external wake-up) flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 3 event
2 WUPTB2F	Wake-up on GPIO 2 event (TIMER output compare) flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 2 event
1 WUPTB1F	Wake-up on GPIO 1 event (TIMER output compare) flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 1 event
0 WUPTB0F	Wake-up on GPIO 0 event (TIMER output compare) flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 0 event

4.3.7.3.4 PCR 1.0 ms prescaler (PCR_PRESC)

Table 77. PCR 1.0 ms Prescaler (PCR_PRESC)

Offset [97],[98] 0x04 Access: User read/write

	15	14	13	12	11	10	9	8
R W	PRESC[15:8]							
Reset	0	1	1	1	1	1	0	1
	7	6	5	4	3	2	1	0
R W	PRESC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes

[97] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[98] This Register is 16 Bit access only.

Table 78. PCR 1.0 ms prescaler (PCR_PRESC) - register field descriptions

Field	Description
15-0 PRESC[15:0]	1.0 ms Prescaler, used to derive D2DSCLK and D2DFCLK from the D2DCLK signal. See 4.3.5, “Device clock tree" for details.

4.3.7.3.5 Wake-up enable register (PCR_WUE (hi))

Table 79. Wake-up enable register (PCR_WUE (hi))

Offset[99] 0x06 Access: User read/write

	7	6	5	4	3	2	1	0
R W	WUAHTH	WUCTH	WUCAL	WULIN	WUPTB3	WUPTB2	WUPTB1	WUPTB0
Reset	0	0	0	0	0	0	0	0

Notes

[99] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 80. Wake-up enable register (PCR_WUE (hi)) - register field descriptions

Field	Description
7 WUAHTH	0 - Wake-up on Ah counter disabled 1 - Wake-up on Ah counter enabled
6 WUCTH	0 - Wake-up on current threshold disabled 1 - Wake-up on current threshold enabled
5 WUCAL	0 - Wake-up on calibration request disabled 1 - Wake-up on calibration request enabled
4 WULIN	0 - Wake-up on LIN disabled 1 - Wake-up on LIN enabled
3 WUPTB3	0 - Wake-up on GPIO 3 event disabled 1 - Wake-up on GPIO 3 event enabled

Table 80. Wake-up enable register (PCR_WUE (hi)) - register field descriptions (continued)

Field	Description
2 WUPTB2	0 - Wake-up on GPIO 2 event disabled 1 - Wake-up on GPIO 2 event enabled
1 WUPTB1	0 - Wake-up on GPIO 1 event disabled 1 - Wake-up on GPIO 1 event enabled
0 WUPTB0	0 - Wake-up on GPIO 0 event disabled 1 - Wake-up on GPIO 0 event enabled

4.3.7.3.6 Wake-up enable register (PCR_WUE (lo))

Table 81. Wake-up enable register (PCR_WUE (lo))

Offset[100] 0x07

Access: User read/write

	7	6	5	4	3	2	1	0
R	WULTC	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[100] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 82. Wake-up enable register (PCR_WUE (lo)) - register field descriptions

Field	Description
7 WULTC	0 - Wake-up on Life Timer Counter Overflow disabled 1 - Wake-up on Life Timer Counter Overflow enabled

4.3.7.3.7 Trim for accurate 1.0 ms low freq clock (TRIM_ALF (hi))

Table 83. Trim for accurate 1.0 ms low freq clock (TRIM_ALF (hi))

Offset[101] 0x0E

Access: User read

	15	14	13	12	11	10	9	8
R	PRDF	0	0	APRESC[12:8]				
W								

Notes

[101] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.3.7.3.8 Trim for accurate 1.0 ms low freq clock (TRIM_ALF (lo))

Table 84. Trim for accurate 1.0 ms low freq clock (TRIM_ALF (lo))

Offset[102] 0x0F

Access: User read

	7	6	5	4	3	2	1	0
R	APRESC[7:0]							
W								

Notes

[102] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 85. Trim for accurate 1.0 ms low freq clock (TRIM_ALF (lo)) - register field descriptions

Field	Description
15 PRDF	ALFCLK Prescaler ready Flag 0 - The ALFCLK synchronization after power up or PRESC[15:0] / PF[1:0] change is not completed. 1 - The ALFCLK synchronization is complete. The ALFCLK signal is synchronized to the D2DCLK.
12-0 APRESC[12:0]	ALFCLK Prescaler This read only value represents the current ALFCLK prescaler value. With the synchronization complete (PRDF=1), the prescaler is used to create the calibrated clock for the Life Time Counter (Normal mode and Low Power mode), and Timer and Current trigger (Low Power Mode only), based on the low power oscillator. After Power Up, the APRESC register is reset to 0x0200 (512dec) until the first synchronization is complete. This will initialize the ALFCLK to 1.0 kHz.

4.4 Interrupt module - IRQ

4.4.1 Introduction

Several interrupt sources are implemented on the analog die to indicate important system conditions. Those Interrupt events are signalized via the D2DINT signal to the microcontroller. See Section 4.18, “MCU - interrupt module (S12S9S12I128PIMV1V1)".

4.4.2 Interrupt source identification

Once an Interrupt is signalized, there are two options to identify the corresponding source(s).

NOTE

The following Interrupt source registers (Interrupt Source Mirror and Interrupt Vector Emulation by Priority) are indicators only. After identifying the interrupt source, the acknowledgement of the interrupt has to be performed in the corresponding block.

4.4.2.1 Interrupt source mirror

All Interrupt sources in the MM912_637 analog die are mirrored to a special Interrupt Source Register (INT_SRC). This register is read only and will indicate all currently pending Interrupts. Reading this register will not acknowledge any interrupt. An additional D2D access is necessary to serve the specific module.

4.4.2.2 Interrupt vector emulation by priority

To allow a vector based interrupt handling by the MCU, the number of the highest prioritized interrupt pending is returned in the Interrupt Vector Register (INT_VECT). Reading this register will not acknowledge an interrupt. An additional D2D access is necessary to serve the specific module.

4.4.3 Interrupt global mask

The Global Interrupt mask registers INT_MSK (hi) and INT_MSK (lo) are implemented to allow a global enable / disable of all analog die Interrupt sources. The individual blocks mask registers should be used to control the individual sources.

4.4.4 Interrupt sources

The following Interrupt sources are implemented on the analog die.

Table 86. Interrupt sources

IRQ	Description
UVI	Undervoltage Interrupt (or wake-up from Cranking mode)
HTI	High Temperature Interrupt

Table 86. Interrupt sources

IRQ	Description
LTI	LIN Driver Overtemperature Interrupt
CH0	TIM Channel 0 Interrupt
CH1	TIM Channel 1 Interrupt
CH2	TIM Channel 2 Interrupt
CH3	TIM Channel 3 Interrupt
TOV	TIM Timer Overflow Interrupt
ERR	SCI Error Interrupt
TX	SCI Transmit Interrupt
RX	SCI Receive Interrupt
CVMI	Current / Voltage Measurement Interrupt
LTC	Lifetime Counter Interrupt
CAL	Calibration Request Interrupt

4.4.4.1 Undervoltage interrupt (UVI)

This maskable interrupt signalizes a undervoltage condition on the VSUP supply input. Acknowledge the interrupt by writing a 1 into the UVF Bit in the PCR Status Register (PCR_SR (hi)). The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. The UVF Bit represents the current condition, and might not be set after an interrupt was signalized by the interrupt source registers. See Section 4.3, “Analog die - power, clock and resets - PCR" for details on the PCR Status Register (PCR_SR (hi)), including masking information.

NOTE

The undervoltage interrupt is not active in devices with the Cranking mode enabled. For those devices, the undervoltage threshold is used to enable the high precision low voltage threshold during Stop/Sleep mode. Once the device wakes up from cranking mode, the UVI flag is indicating the wake-up source.

4.4.4.2 High temperature interrupt (HTI)

This maskable interrupt signalizes a high temperature condition on the analog die. The sensing element is located close to the major thermal contributors, the system voltage regulators.

Acknowledge the interrupt by writing a 1 into the HTF Bit in the PCR Status Register (PCR_SR (hi)). The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. The HTF Bit represents the current condition and might not be set after an interrupt was signalized by the interrupt source registers. See Section 4.3, “Analog die - power, clock and resets - PCR" for details on the PCR Status Register (PCR_SR (hi)), including masking information.

4.4.4.3 LIN driver overtemperature interrupt (LTI)

Acknowledge the interrupt by reading the LIN Register - LINR. The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. See Section 4.12, “LIN" for details on the LIN Register, including masking information.

4.4.4.4 TIM channel 0 interrupt (CH0)

See Section 4.10, “Basic timer module - TIM (TIM16B4C)".

4.4.4.5 TIM channel 1 interrupt (CH1)

See Section 4.10, “Basic timer module - TIM (TIM16B4C)".

4.4.4.6 TIM channel 2 interrupt (CH2)

See Section 4.10, “Basic timer module - TIM (TIM16B4C)".

4.4.4.7 TIM channel 3 interrupt (CH3)

See Section 4.10, “Basic timer module - TIM (TIM16B4C)".

4.4.4.8 TIM timer overflow interrupt (TOV)

See Section 4.10, “Basic timer module - TIM (TIM16B4C)".

4.4.4.9 SCI error interrupt (ERR)

See Section 4.13, “Serial communication interface (S08SCIV4)".

4.4.4.10 SCI transmit interrupt (TX)

See Section 4.13, “Serial communication interface (S08SCIV4)".

4.4.4.11 SCI receive interrupt (RX)

See Section 4.13, “Serial communication interface (S08SCIV4)".

4.4.4.12 Current/voltage measurement interrupt (CVMI)

Indicates the current or voltage measurement finished (VM or CM bit set). See Section 4.8, “Channel acquisition".

4.4.4.13 Life time counter interrupt (LTC)

In case a Life Time Counter overflow occurs with the corresponding interrupt enabled, the LTC interrupt is issued. See Section 4.14, “Life time counter (LTC)".

4.4.4.14 Calibration request interrupt (CAL)

Once a request for re-calibration is present (Temperature out of pre-set range), the Calibration Interrupt is issued. After a calibration wake-up, the reading of ACQ_ITEMP must be done after waiting for the latency of the temperature acquisition chain. Then ACQ_ITEMP is valid to adjust the compensation over temperature. See full documentation on the interrupt source inSection 4.8, “Channel acquisition".

4.4.5 IRQ - memory map and registers

4.4.5.1 Overview

This section provides a detailed description of the memory map and registers.

4.4.5.2 Module memory map

The memory map for the IRQ module is given in Table 62

Table 87. Module memory map

Offset[103]	Name		7	6	5	4	3	2	1	0
0x08	INT_SRC (hi)	R	TOV	CH3	CH2	CH1	CH0	LTI	HTI	UVI
	Interrupt source register	W								
0x09	INT_SRC (lo)	R	0	0	CAL	LTC	CVMI	RX	TX	ERR
	Interrupt source register	W								
0x0A	INT_VECT	R	0	0	0	0	IRQ[3:0]			
	Interrupt vector register	W								
0x0B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0C	INT_MSK (hi)	R	TOVM	CH3M	CH2M	CH1M	CH0M	LTIM	HTIM	UVIM
	Interrupt mask register	W								
0x0D	INT_MSK (lo)	R	0	0	CALM	LTCM	CVMM	RXM	TXM	ERRM
	Interrupt mask register	W								

Notes

[103] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.4.5.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

4.4.5.3.1 Interrupt source register (INT_SRC (hi))

Table 88. Interrupt source register (INT_SRC (hi))

Offset[104] 0x08 Access: User read

	7	6	5	4	3	2	1	0
R	TOV	CH3	CH2	CH1	CH0	LTI	HTI	UVI
W								

Notes

[104] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 89. Interrupt Source Register (INT_SRC (hi)) - register field descriptions

Field	Description
7 TOV	TIM16B4C - Timer overflow interrupt status 0 - No timer overflow interrupt pending 1 - Timer overflow interrupt pending
6 CH3	TIM16B4C - TIM channel 3 interrupt status 0 - No channel 3 interrupt pending 1 - Channel 3 interrupt pending

Table 89. Interrupt Source Register (INT_SRC (hi)) - register field descriptions (continued)

Field	Description
5 CH2	TIM16B4C - TIM channel 2 interrupt status 0 - No channel 2 interrupt pending 1 - Channel 2 interrupt pending
4 CH1	TIM16B4C - TIM channel 1 interrupt status 0 - No channel 1 interrupt pending 1 - Channel 1 interrupt pending
3 CH0	TIM16B4C - TIM channel 0 interrupt status 0 - No channel 0 interrupt pending 1 - Channel 0 interrupt pending
2 LTI	LIN Driver overtemperature interrupt status 0 - No LIN driver overtemperature interrupt 1 - LIN driver overtemperature interrupt
1 HTI	High temperature interrupt status 0 - No high temperature interrupt pending 1 - High temperature interrupt pending
0 UVI	Undervoltage interrupt pending or wake-up from Cranking mode status 0 - No undervoltage Interrupt pending or wake-up from Cranking mode 1 - Undervoltage interrupt pending or wake-up from Cranking mode

4.4.5.3.2 Interrupt source register (INT_SRC (lo))

Table 90. Interrupt source register (INT_SRC (lo))

Offset[105] 0x09 Access: User read

	7	6	5	4	3	2	1	0
R	0	0	CAL	LTC	CVMI	RX	TX	ERR
W								

Notes

[105] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 91. Interrupt source register (INT_SRC (lo)) - register field descriptions

Field	Description
5 CAL	Calibration request interrupt status 0 - No calibration request interrupt pending 1 - Calibration request interrupt pending
4 LTC	Life time counter interrupt status 0 - No life time counter interrupt pending 1 - Life time counter interrupt pending
3 CVMI	Current / Voltage measurement interrupt status 0 - No Current / Voltage measurement interrupt pending 1 - Current / Voltage measurement interrupt pending
2 RX	SCI receive interrupt status 0 - No SCI receive interrupt pending 1 - SCI receive interrupt pending

Table 91. Interrupt source register (INT_SRC (lo)) - register field descriptions (continued)

Field	Description
1 TX	SCI transmit interrupt status 0 - No SCI transmit interrupt pending 1 - SCI transmit interrupt pending
0 ERR	SCI error interrupt status 0 - No SCI transmit interrupt pending 1 - SCI transmit interrupt pending

4.4.5.3.3 Interrupt vector register (INT_VECT)

Table 92. Interrupt vector register (INT_VECT)

Offset[106] 0x0A

Access: User read

	7	6	5	4	3	2	1	0
R	0	0	0	0	IRQ			
W								

Notes

[106] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 93. Interrupt vector register (INT_VECT) - register field descriptions

Field	Description
4-0 IRQ	Represents the highest prioritized interrupt pending. See Table 94. If no interrupt is pending, the result will be 0.

Table 94. Interrupt vector/priority

IRQ	Description	IRQ	Priority
-	No interrupt pending or wake-up from Stop mode	0x00	-
UVI	Undervoltage interrupt or wake-up from Cranking mode	0x01	1 (highest)
HTI	High temperature interrupt	0x02	2
LTI	LIN driver overtemperature interrupt	0x03	3
CH0	TIM channel 0 interrupt	0x04	4
CH1	TIM channel 1 interrupt	0x05	5
CH2	TIM channel 2 interrupt	0x06	6
CH3	TIM channel 3 interrupt	0x07	7
TOV	TIM timer overflow interrupt	0x08	8
ERR	SCI error interrupt	0x09	9
TX	SCI transmit interrupt	0x0A	10
RX	SCI receive interrupt	0x0B	11
CVMI	Acquisition interrupt	0x0C	12
LTC	Life time counter interrupt	0x0D	13
CAL	Calibration request interrupt	0x0E	14 (lowest)

4.4.5.3.4 Interrupt mask register (INT_MSK (hi))

Table 95. Interrupt mask register (INT_MSK (hi))

Offset[107] 0x0C Access: User read/write

	7	6	5	4	3	2	1	0
R W	TOVM	CH3M	CH2M	CH1M	CH0M	LTIM	HTIM	UVIM
Reset	0	0	0	0	0	0	0	0

Notes

[107] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 96. Interrupt mask register (INT_MSK (hi)) - register field descriptions

Field	Description
7 TOVM	Timer overflow interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
6 CH3M	Timer channel 3 interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
5 CH2M	Timer channel 2 interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
4 CH1M	Timer channel 1 interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
3 CH0M	Timer channel 1 interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
2 LTIM	LIN driver overtemperature interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
1 HTIM	High temperature interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
0 UVIM	Undervoltage interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled

4.4.5.3.5 Interrupt mask register (INT_MSK (lo))

Table 97. Interrupt mask register (INT_MSK (lo))

Offset[108] 0x0D Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	CALM	LTCM	CVMM	RXM	TXM	ERRM
W								
Reset	0	0	0	0	0	0	0	0

Notes

[108] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 98. Interrupt mask register (INT_MSK (lo)) - register field descriptions

Field	Description
5 CALM	Calibration request interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
4 LTCM	Life time counter interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
3 CVMM	Current / Voltage measurement interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
2 RXM	SCI receive interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
1 TXM	SCI transmit interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled
0 ERRM	SCI error interrupt mask 0 - Interrupt enabled 1 - Interrupt disabled

4.5 Current measurement - ISENSE

4.5.1 Introduction

This chapter only gives a summary of the current sense module. Refer to Section 4.8, “Channel acquisition" for the complete description of all acquisition channels, including the current measurement channel.

4.5.1.1 Features

- Dedicated 16 Bit Sigma Delta (ΣΔ) ADC
- Programmable Gain Amplifier (PGA) with 8 programmable gain factors
- Gain Control Block (GCB) for automatic gain adjustment
- Simultaneous Sampling with Voltage Channel
- Programmable Gain and Offset Compensation
- Optional Chopper Mode with moving average
- SINC3 + IIR Stage
- Calibration mode to compute compensation buffers
- Programmable Low Pass Filter (LPF), configuration shared with the Voltage Measurement Channel
- Optional Shunt resistor sensing feature
- Triggered Sampling during Low Power Mode with programmable wake-up conditions

4.5.1.2 Block diagram

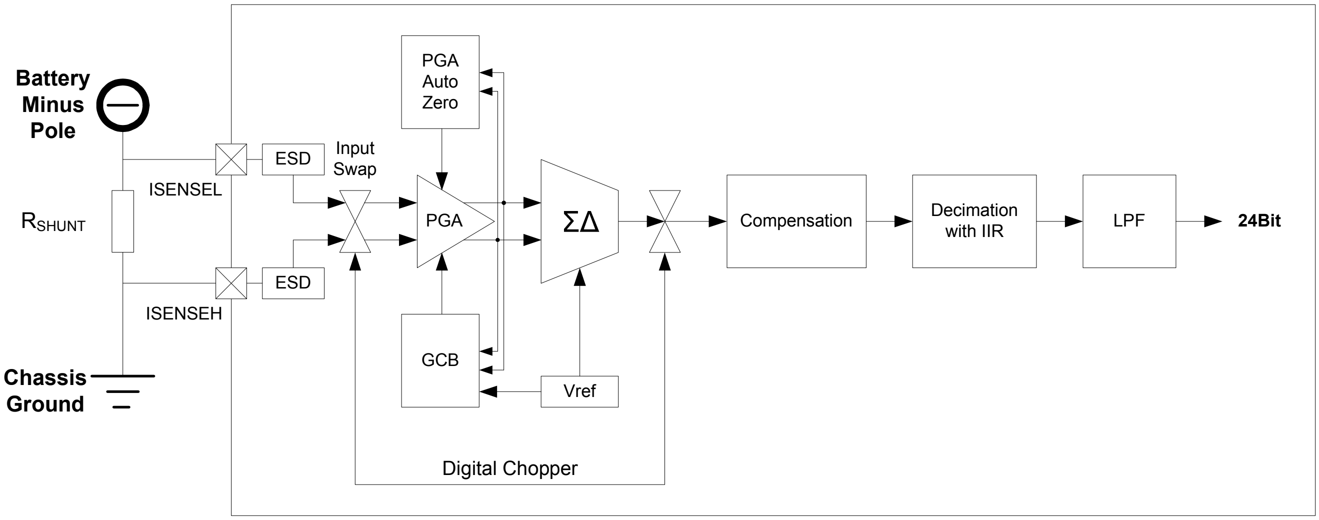


Figure 29. Current measurement channel

The battery current is measured by measuring the voltage drop V_{DROP} over an external shunt resistor, connected to ISENSEH and ISENSEL. V_{DROP}, and is defined as the differential voltage between the ISENSEL and ISENSEH inputs (V_{DROP}=ISENSEL-ISENSEH). A positive voltage drop means a positive current is flowing, and vice versa.

If the GND pin of the module is connected to ISENSEH, the measured current includes the supply current of the MM912_637 (current flows back to negative battery pole). If the GND pin is connected to the ISENSEL input, the supply current of the MM912_637 is not measured. However, the voltage at the ISENSEH input could go below GND (see max ratings). In this case, the current measurement still functions as specified.

4.6 Voltage measurement - VSENSE

4.6.1 Introduction

This chapter only gives a summary of the voltage sense module. Refer to Section 4.8, “Channel acquisition" for the complete description of all acquisition channels, including the voltage measurement channel.

4.6.1.1 Features

- Dedicated 16 Bit Sigma Delta (ΣΔ) ADC
- Fixed High Precision Divider
- Optional External Voltage Input “VOPT”
- Simultaneous Sampling with Current Channel
- Programmable Gain and Offset Compensation
- Calibration mode to compute compensation buffers
- Optional Chopper mode with moving average
- SINC3 + IIR Stage
- Programmable Low Pass Filter (LPF), Configuration shared with Current Measurement Channel

4.6.1.2 Block diagram

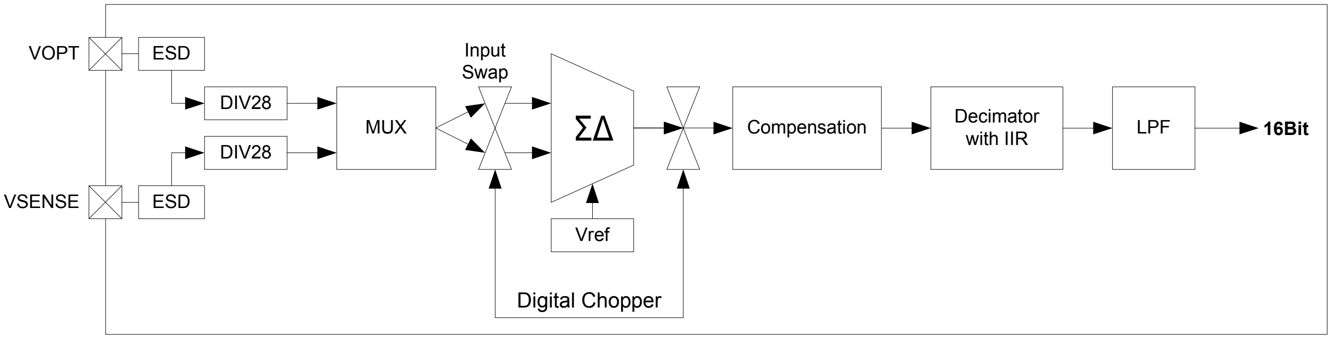


Figure 30. Voltage Measurement Channel

The battery voltage is measured by default, via the VSENSE input. A high precision divider stage scales down the battery voltage by a fixed factor K =1/28, to a voltage below the internal reference voltage of the Sigma Delta ADC (VSENSE*K < V_{REF}). If an optional external voltage is measured, the multiplexer (MUX) is selected to feed the V_{OPT} input to the buffer.

4.7 Temperature measurement - TSENSE

4.7.1 Introduction

This chapter only gives a summary of the temperature sense module. Refer to Section 4.8, “Channel acquisition" for the complete description of all acquisition channels, including the temperature measurement channel.

4.7.1.1 Features

- Internal on chip Temperature Sensor
- Optional External Temperature Sensor Input (VTEMP)
- Dedicated 16-Bit Sigma Delta ADC
- Programmable Gain and Offset Compensation
- Optional External Sensor Supply (TSUP) with selectable capacitor
- Optional Measurement during Low Power mode to trigger recalibration

4.7.1.2 Block diagram

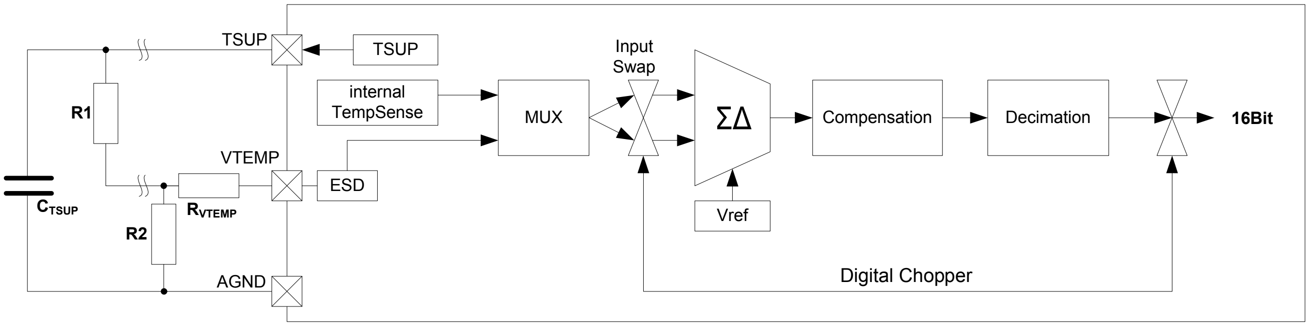


Figure 31. Temperature measurement channel

NOTE

To minimize ground shift effects while using the external sensor option, R2 must be placed as close to the AGND pin as possible. C_{TSUP} is optional. The supply output must be configured to operate with the capacitor.

4.8 Channel acquisition

4.8.1 Introduction

This chapter documents the current, voltage, and temperature acquisition flow. The chapter is structured in the following sections.

- Section 4.8.2, “Channel structure overview"
- Section 4.8.3, “Current and voltage measurement"
  - Section 4.8.3.1, “Shunt sense, PGA, and GCB (current channel only)"
  - Section 4.8.3.2, “Voltage sense multiplexer (voltage channel only)"
  - Section 4.8.3.3, “Sigma delta converter"
  - Section 4.8.3.4, “Compensation"
  - Section 4.8.3.5, “IIR/decimation/chopping stage"
  - Section 4.8.3.6, “Low pass filter"
  - Section 4.8.3.7, “Format and clamping"
- Section 4.8.4, “Temperature measurement channel"
  - Section 4.8.4.1, “Compensation"
- Section 4.8.5, “Calibration"
- Section 4.8.6, “Memory map and registers"

4.8.2 Channel structure overview

The MM912_637 offers three parallel measurement channels. Current, Voltage, and Temperature. The Voltage Channel is shared between the VSENSE and VOPT voltage source, the Temperature channel between ETEMP and ITEMP.

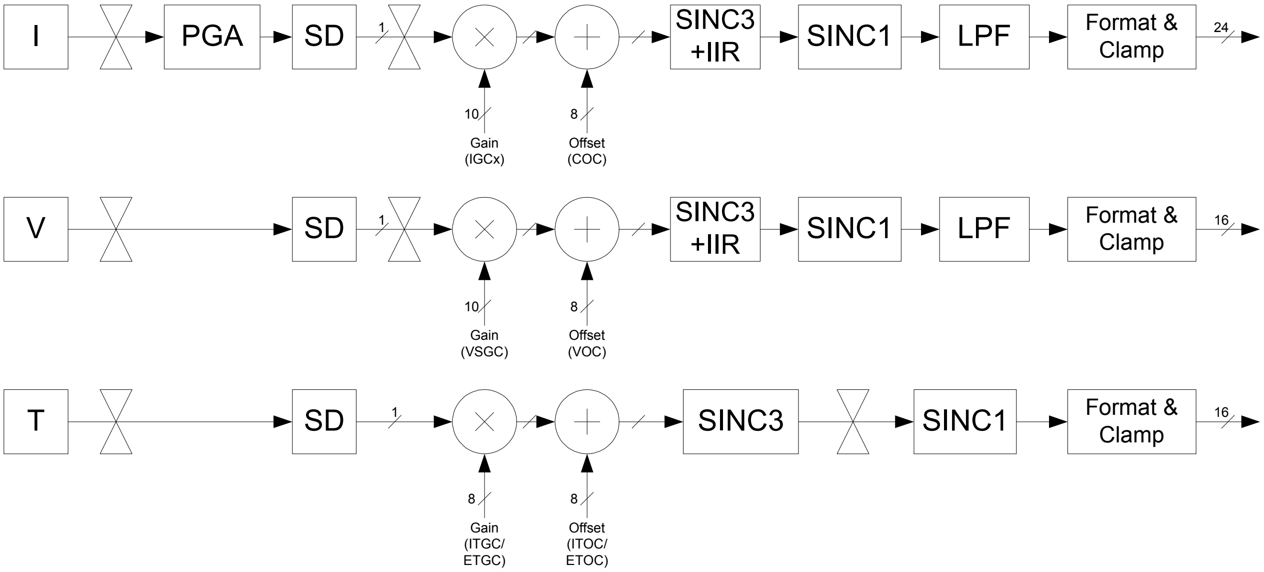


Figure 32. Simplified measurement channel

Figure 33 shows an overview of the detailed dependencies between the control and status registers and the channels. Refer to the following sections of this chapter for details.

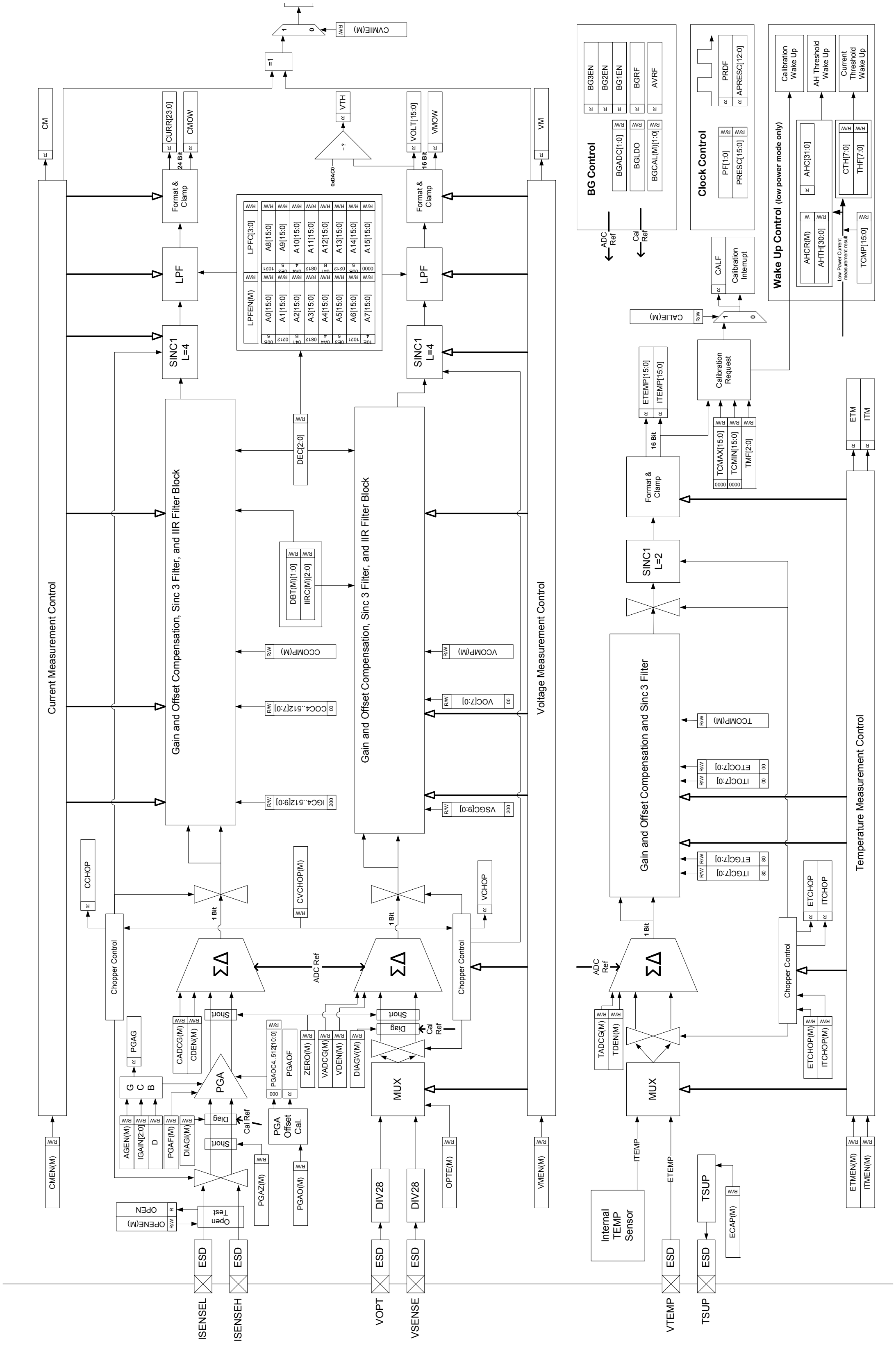


Figure 33. Channel complete overview

4.8.3 Current and voltage measurement

To guarantee synchronous voltage and current acquisition, both channels are implemented equal in terms of digital signal conditioning and timing. The analog signal conditioning, before the Sigma Delta Converter, is different to match the different sources.

4.8.3.1 Shunt sense, PGA, and GCB (current channel only)

Current Channel specific analog signal conditioning.

4.8.3.1.1 Shunt sense

An optional current sense feature is implemented to sense the presence of the current shunt resistor. Setting the OPEN bit (ACQ_CTL register), will activate the feature. The OPEN bit (ACQ_SR register) will indicate the shunt resistor open. The sense feature will detect an open condition for a shunt resistance R_{SHUNT} > ROPEN.

4.8.3.1.2 Programmable gain amplifier (PGA)

To allow a wide range of current levels to be measured, a programmable gain amplifier is implemented. Following the input chopper (see Section 4.8.3.5, “IIR/decimation/chopping stage"), the differential voltage is amplified by one of the 8 gains controlled by the Gain Control Block. The PGA has an internal offset compensation feature - see Section 4.8.4.1, “Compensation" and Section 4.8.5, “Calibration" for details.

4.8.3.1.3 Gain control block (GCB)

To allow a transparent Gain adjustment with minimum MCU load, an automatic gain control has been implemented. The absolute output of the PGA is constantly compared with a programmable up and down threshold (ACQ_GCB register). The threshold is a D/A output according Table 99.

Table 99. Gain control block - register

ACQ_GCB D[7:0]	GCB high (up) threshold	ACQ_GCB D[7:0]	GCB low (down) threshold
0000xxxx	1/16 V_{REF}	xxxx0000	0
0001xxxx	2/16 V_{REF}	xxxx0001	1/16 V_{REF}
0010xxxx	3/16 V_{REF}	xxxx0010	2/16 V_{REF}
0011xxxx	4/16 V_{REF}	xxxx0011	3/16 V_{REF}
0100xxxx	5/16 V_{REF}	xxxx0100	4/16 V_{REF}
0101xxxx	6/16 V_{REF}	xxxx0101	5/16 V_{REF}
0110xxxx	7/16 V_{REF}	xxxx0110	6/16 V_{REF}
0111xxxx	8/16 V_{REF}	xxxx0111	7/16 V_{REF}
1000xxxx	9/16 V_{REF}	xxxx1000	8/16 V_{REF}
1001xxxx	10/16 V_{REF}	xxxx1001	9/16 V_{REF}
1010xxxx	11/16 V_{REF}	xxxx1010	10/16 V_{REF}
1011xxxx	12/16 V_{REF}	xxxx1011	11/16 V_{REF}
1100xxxx	13/16 V_{REF}	xxxx1100	12/16 V_{REF}
1101xxxx	14/16 V_{REF}	xxxx1101	13/16 V_{REF}
1110xxxx	15/16 V_{REF}	xxxx1110	14/16 V_{REF}
1111xxxx	16/16 V_{REF}	xxxx1111	15/16 V_{REF}

Once the programmed threshold is reached, the gain is adjusted to the next level. The currently active gain setting can be read in the IGAIN[2:0] register. Once the gain has been adjusted by the GCB, the PGAG bit will be set.

The automatic Gain Control can be disabled by clearing the AGEN bit. In this case, writing the IGAIN[2:0] register will allow manual gain control.

NOTE

> The IGAIN[2:0] register content does determine the offset compensation register access, as there are 8 individual offset register buffers implemented, accessed through the same COC[7:0] register.

4.8.3.2 Voltage sense multiplexer (voltage channel only)

A multiplexer has been implemented to select between the VSENSE or VOPT voltage input. The OPTE bit controls the multiplexer. Both input signals are divided by a fixed DIV28 divider.

NOTE

> There is no further state machine separation of the two voltage channels. The software has to assure all compensation registers are configured properly after changing the multiplexer. Both voltage source conversion results will be stored in the same result register.

The divided and multiplexed voltages will be routed through the optional chopper (see Section 4.8.3.5, “IIR/decimation/chopping stage") before entering the Sigma Delta converter stage.

4.8.3.3 Sigma delta converter

4.8.3.3.1 Overview

A high resolution ADC is needed for current and battery voltage measurements of the MM912_637. A second order sigma delta modulator based architecture is chosen.

4.8.3.4 Compensation

Following the optional chopper stage, the sigma delta bit stream is first gain and then offset compensated using the compensation registers.

The compensation stages for both channels can be completely bypassed by clearing the CCOMP / VCOMP bits.

4.8.3.4.1 Gain compensation

Table 100 shows the gain compensation register for the current and voltage channel. At system startup, the factory trimmed values have to be copied into the VSGC and IGCx registers (see Section 5.2, “IFR trimming content and location").

NOTE

> There are 8 individual Gain compensation registers for the current measurement channels different PGA gains with 8 individual gain trim values present in the IFR trimming flash.
>
> Based on the voltage channel multiplexer configuration, a different trim gain compensation value has to be used in the compensation register. The compensation register content has to be updated when changing the multiplexer setting.

Table 100. Gain compensation - voltage and current channel

VSGC[9:0] IGCx[9:0]	Voltage channel gain	Current channel gain
0x3FF	1.3174	1.7832
0x3FE	1.3169	1.7822
0x3FD	1.3164	1.7812
.	.	.
.	.	.
0x203	1.0694	1.2872
0x202	1.0689	1.2862

Table 100. Gain compensation - voltage and current channel (continued)

VSGC[9:0] IGCx[9:0]	Voltage channel gain	Current channel gain
0x201	1.0684	1.2852
0x200 (default)	1.0679	1.2842
0x1FF	1.0674	1.2832
0x1FE	1.0669	1.2822
0x1FD	1.0664	1.2812
.	.	.
.	.	.
0x002	0.8189	0.7862
0x001	0.8184	0.7852
0x000	0.8179	0.7842

4.8.3.4.2 Offset compensation

Table 101 shows the offset compensation register for the current and voltage channel. At system startup, the factory trimmed values have to be copied into the VOC and COC registers (see Section 5.2, “IFR trimming content and location").

NOTE

Based on the voltage channel multiplexer and copper configuration, a different trim offset compensation value has to be used in the compensation register. The compensation register content has to be updated when changing the multiplexer setting.

While there is only one offset compensation register VOC[7:0] for the voltage channel, there are 8 individual offset compensation registers for the current channel. The access happens through the COC[7:0] register mapped, based on the IGAIN[2:0] register content.

Table 101. Offset compensation - voltage and current channel

VOC[7:0] COC[7:0]	Voltage channel offset[(109)]	Current channel offset[(109)]
0x7F	+9.073	+15.092
0x7E	+9.002	+14.974
0x7D	+8.93	+14.855
.	.	.
.	.	.
0x03	0.214	+0.357
0x02	0.143	+0.238
0x01	0.071	+0.119
0x00 (default)	0	0
0xFF	-0.071	-0.119
0xFE	-0.143	-0.238
0xFD	-0.214	-0.357
.	.	.
.	.	.
0x82	-9.002	-14.974

Table 101. Offset compensation - voltage and current channel (continued)

VOC[7:0] COC[7:0]	Voltage channel offset[109]	Current channel offset[109]
0x81	-9.073	-15.092
0x80	-9.145	-15.211

Notes

[109] SD input related (mV)

4.8.3.5 IIR/decimation/chopping stage

4.8.3.5.1 Functional description

The chopper frequency is set to one eighth of the decimator frequency (512 kHz typ). On each phase, four decimation cycles are necessary to get a steady signal. The equation of the IIR is $y_{n+1}=\alpha.x_n+(1-\alpha).y_n$.

The α parameter can be configured by the IIRC[2:0] register. See Section 4.8.6.3.18, “I and V chopper control register (ACQ_CVCR (lo))".

The decimation process is then completed by a programmable (DEC[2:0]) sinc3 filter, which outputs a 0.5...8 kS/s signal. The modulated noise is removed by an averaging filter (SINC1; L=4), which has an infinite rejection at the chopping frequency.

4.8.3.5.2 Latency and throughput

- The throughput is 512 kHz/DF with DF configurable from 64 to 1024.
- The latency is given by (4+3*IIR+3*Avger+N_LPF)*DF/512 kHz where:
  - IIR=1 if IIR is enabled (0 otherwise),
  - Avger=1 if the chopper mode is activated (0 otherwise),
  - N_LPF is the LPF coefficient number.

4.8.3.6 Low pass filter

To achieve the required attenuation of the measured voltage and current signals in the frequency domain, a programmable low-pass filter following the SINC3+IIR filter, is implemented for both channels with shared configuration registers to deliver the equivalent filtering.

The following filter characteristic is implemented:

- F_{PASS} = 100 Hz (Att100 Hz)
- F_{STOPP} = 500 Hz (Att500 Hz)

The number of filter coefficients used can be programmed in the ACQ_LPFC[3:0] register. The filter can be bypassed completely clearing the LPFEN bit.

The filter uses an algorithmic and logic unit (ALU) for calculating the filtered output data, depending on the incoming data stream at “DATA IN” and the low-pass coefficients (A0...15) at the input “COEFF”, 16-bit width of each coefficient (See 4.8.6.3.22, “Low pass filter coefficient Ax (LPF_Ax (hi))"). The filter structure calculates during one cycle (Tcyc=1/Fadc) the filtered data output.

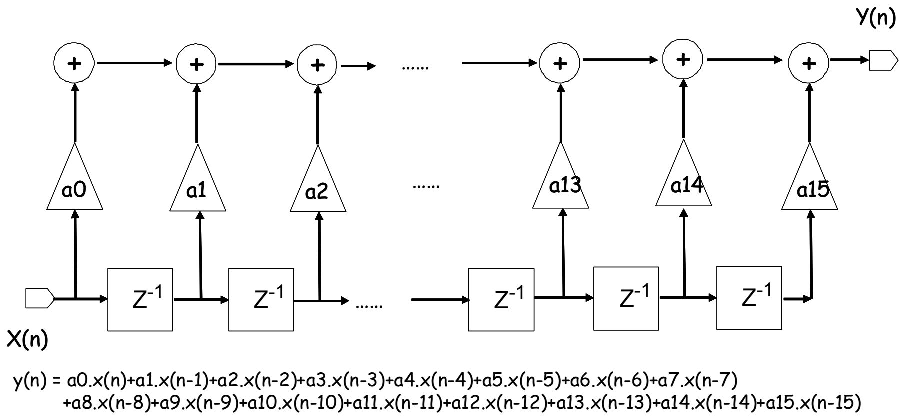

Y(n)
+
a0
a1
a2
......
a13
a14
a15
Z-1
X(n)
y(n) = a0.x(n)+a1.x(n-1)+a2.x(n-2)+a3.x(n-3)+a4.x(n-4)+a5.x(n-5)+a6.x(n-6)+a7.x(n-7)
+a8.x(n-8)+a9.x(n-9)+a10.x(n-10)+a11.x(n-11)+a12.x(n-12)+a13.x(n-13)+a14.x(n-14)+a15.x(n-15)

Figure 34. FIR structure

Z^{-1} Unit delay is done at a programmable frequency, depending on the decimation factor programmed in the DEC[2:0] register. See Table 119.

NOTE

> There is no decimation from SINC3 to the LPF output, LPF uses same output rate than decimator. It's therefore possible to select an output update rate independent of the filter characteristic and bandwidth.

The coefficient vector consists of 16*16-bit elements and is free programmable, the maximum response time for 16 coefficients structure is 16*1/output rate. The following filter function can be realized.

$$H_{LP}(z) = \sum_{i=0}^{M} a_i * z^{-i}$$

LP filter function ***Eqn. 3***

The coefficients aj are the elements of the coefficient vector and determine the filter function. M <= 16. It's possible to realize FIR filter functions. A typical total frequency response of the decimator and the programmable LP filter is given in Figure 35.

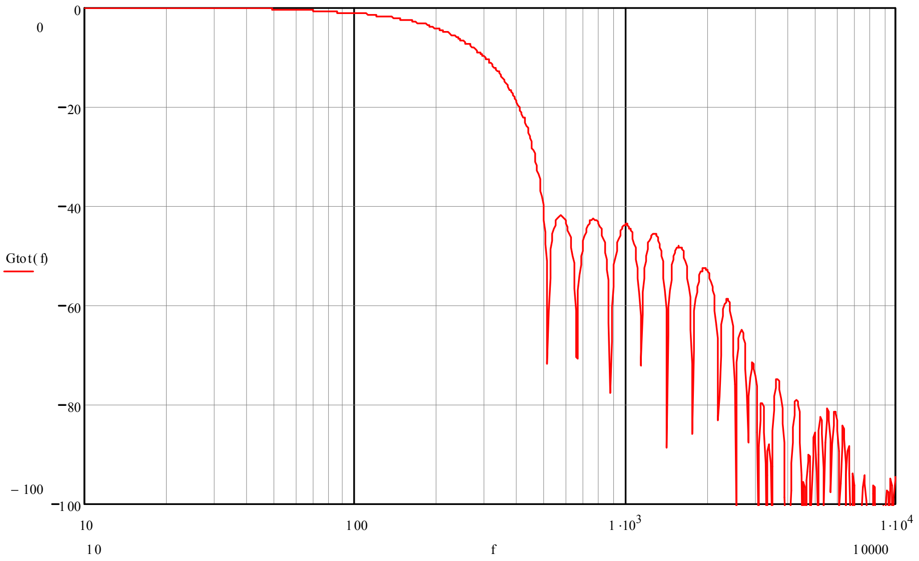


Figure 35. Typical total filter response Sinc3 (D=128), LP filter (FIR type with 15 coefficients used)

4.8.3.7 Format and clamping

The output data stream is formatted into its final size for both channels (16-Bit for Voltage and 24-Bit for Current). The current result will contain the gain information as part of the result. See Section 4.8.6.3.12, “Current measurement result (ACQ_CURR1 / ACQ_CURR0)" and Section 4.8.6.3.13, “Voltage measurement result (ACQ_VOLT)". Both results are written into the corresponding result registers and will issue an IRQ if enabled.

The internal voltage measurement results (no compensation active) are clamped to maximum and minimum values of 0xFFFF and 0x0000 respectively. Terminal voltages outside this range will result in the respective max or min clamped values. The internal current measurement results (no compensation active) are clamped to maximum and minimum values of 0x0FFFF and 0x10000 respectively. Terminal voltages outside this range will result in the respective max or min clamped values.

NOTE

Both channels will perform synchronized conversions when enabled with a single write to the ACQ_CTL register.

As the voltage channel is not active during low power mode, the synchronicity might not be given after wake-up, and has to be re-established by restarting both channels.

Entering low power mode with the current / temperature channel enabled will have the channel(s) remain active during low power mode.

4.8.4 Temperature measurement channel

The MM912_637 can measure the temperature from an internal built-in temperature sensor, or from an external temperature sensor connected to the VTEMP pin. The external temperature sensor is supplied via the TSUP pin. The measurement channel is the same for the internal and external temperature sensor.

The temperature measurement channel uses the same Sigma Delta (SD) converter implementation as the current and voltage channel, followed by a fixed decimation (L=128). A selectable chopper mode is implemented to compensate for offset errors. Once the chopper is enabled, an average (sinc1, L=2) is active.

Once the measurement is enabled, the temperature result registers are updated with the channel update rate. When both measurements are enabled, both temperature sensors are measured successively where the measurement is started with the internal sensor.

The internal temperature measurement result (no compensation active) of 0x0000 represents 0K, the maximum 0xFFFF = 523K (typ). The result data is stored into the result registers ACQ_ITEMP and ACQ_ETEMP (both 16-bit). During an over range event, the ADC is limited to the maximum value. The result of the internal temperature measurement is utilized to generate the calibration request. See Section 4.8.5, “Calibration".

4.8.4.1 Compensation

The compensation for the temperature channels is implemented similar to the current and voltage channel.

Table 102. Gain compensation - temperature channel

ITGC[7:0] ETGC[7:0]	Temperature channel gain compensation
0xFF	1.124
0xFE	1.123
0xFD	1.122
.	.
.	.
0x83	1.003
0x82	1.002
0x81	1.001
0x80 (default)	1.000
0x7F	0.999

Table 102. Gain compensation - temperature channel (continued)

ITGC[7:0] ETGC[7:0]	Temperature channel gain compensation
0x7E	0.998
0x7D	0.997
.	.
.	.
0x02	0.877
0x01	0.876
0x00	0.875

Table 103. Offset compensation - temperature channel[110]

ITOC[7:0] ETOC[7:0]	Temperature channel offset compensation[111]
0x7F	+9.689
0x7E	+9.613
0x7D	+9.537
.	.
.	.
0x03	+0.229
0x02	+0.153
0x01	+0.076
0x00 (default)	0
0xFF	-0.076
0xFE	-0.153
0xFD	-0.229
.	.
.	.
0x82	-9.613
0x81	-9.689
0x80	-9.766

Notes

[110] Typical values based on default gain setting

[111] SD input related (mV)

NOTE

Factory trimmed compensation values are only available for the internal temperature channel.

4.8.5 Calibration

To ensure the maximum precision of the current and voltage sense module, several stages of calibration are implemented to compensate temperature effects. The calibration concept combines the availability of FLASH and the temperature information to guarantee the measurement accuracy under all functional conditions. The trimming and calibration procedures are split in three different categories: Power On-, Calibration Request-, and Optional Verification Procedures.

4.8.5.1 System power on procedure

Several device parameters are guaranteed with full precision after system trimming only. During final test of the device, trim values are computed, verified, and stored into the system FLASH memory. To ensure optimum system performance, the following power on procedure has to be performed during power on. As the device is typically constantly powered during its operation, this operation has to be performed typically one time only. During a system power loss or low power reset condition, the application software has to ensure the procedure executes again.

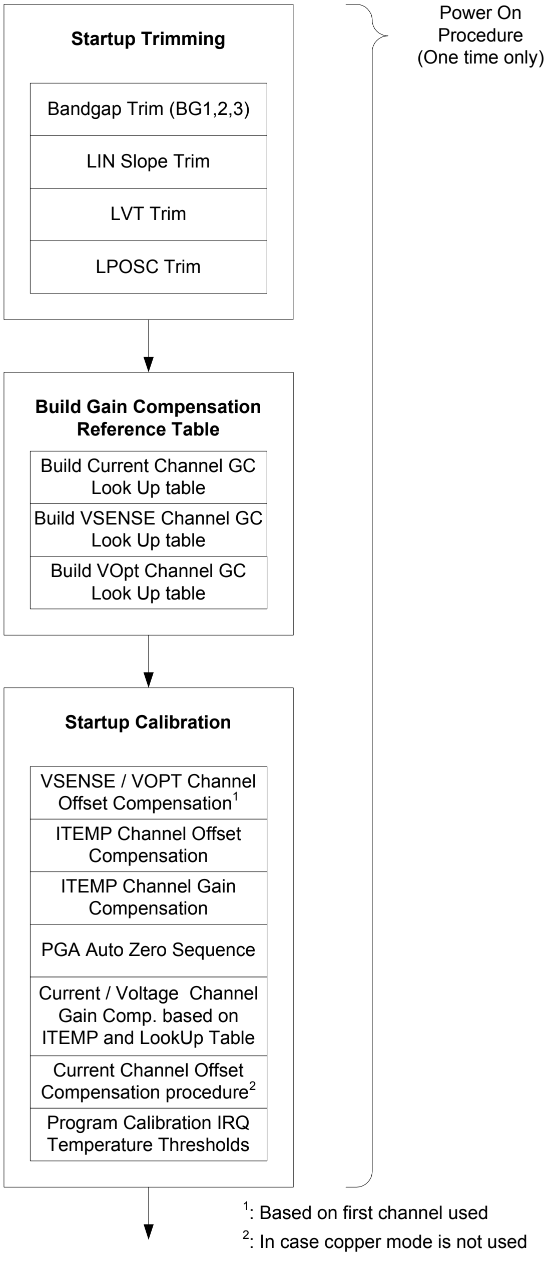

Startup Trimming
Bandgap Trim (BG1,2,3)
LIN Slope Trim
LVT Trim
LPOSC Trim
Build Gain Compensation Reference Table
Build Current Channel GC Look Up table
Build VSENSE Channel GC Look Up table
Build VOpt Channel GC Look Up table
Startup Calibration
VSENSE / VOPT Channel Offset Compensation1
ITEMP Channel Offset Compensation
ITEMP Channel Gain Compensation
PGA Auto Zero Sequence
Current / Voltage Channel Gain Comp. based on ITEMP and LookUp Table
Current Channel Offset Compensation procedure2
Program Calibration IRQ Temperature Thresholds
Power On Procedure (One time only)
1: Based on first channel used
2: In case copper mode is not used

Figure 36. Power on procedure

4.8.5.1.1 Startup trimming

To ensure all analog die modules are being trimmed properly, the following FLASH information (located in the MCU IFR from 0x01_80D0 to 0x01_80D9) has to be copied to the analog die register 0xE0 to 0xE9. This trimming includes the Band Gap Reference adjustment for the 3 system Band Gap circuits, The LIN slope adjustment (TRIM_LIN), the Low Voltage Threshold (TRIM_LVT), and the Low Power Oscillator (TRIM_OSC). See Section 5, “MM912_637 - trimming".

NOTE

The LPOSC[12:0] trim will adjust the low power oscillator to its specified accuracy. This will result in the dependent Watchdog timing to be accurate after writing the trimming information.

4.8.5.1.2 Gain compensation look up table

In order to prepare the system for the optional calibration interrupt service during operation, it is be beneficial to create a look up table for the voltage and current channel gain compensation over temperature.

For all current and voltage channel gain buffers, there are corresponding ROOM temperature optimum trim values stored in the IFR FLASH. For HOT (125 °C) and COLD (-40 °C) temperature, the adjustment towards the ROOM value is stored.

Table 104. Gain compensation buffer optimum

Global address	OFFSET		Byte description								Content
	HEX	DEC	7	6	5	4	3	2	1	0	
0x01_80C0	00	00							IGC4[9:8]		Current Channel Gain (4) Compensation - Room Temp
0x01_80C1	01	01	IGC4[7:0]								
0x01_80C2	02	02							IGC8[9:8]		Current Channel Gain (8) Compensation - Room Temp
0x01_80C3	03	03	IGC8[7:0]								
0x01_80C4	04	04							IGC16[9:8]		Current Channel Gain (16) Compensation - Room Temp
0x01_80C5	05	05	IGC16[7:0]								
0x01_80C6	06	06							IGC32[9:8]		Current Channel Gain (32) Compensation - Room Temp
0x01_80C7	07	07	IGC32[7:0]								
0x01_80C8	08	08							IGC64[9:8]		Current Channel Gain (64) Compensation - Room Temp
0x01_80C9	09	09	IGC64[7:0]								
0x01_80CA	0A	10							IGC128[9:8]		Current Channel Gain (128) Compensation - Room Temp
0x01_80CB	0B	11	IGC128[7:0]								
0x01_80CC	0C	12							IGC256[9:8]		Current Channel Gain (256) Compensation - Room Temp
0x01_80CD	0D	13	IGC256[7:0]								
0x01_80CE	0E	14							IGC512[9:8]		Current Channel Gain (512) Compensation - Room Temp
0x01_80CF	0F	15	IGC512[7:0]								
0x01_80DE	1E	30							VSGC[9:8]		VSENSE Channel Gain Compensation - Room Temp
0x01_80DF	1F	31	VSGC[7:0]								
0x01_80E0	20	32							VOGC[9:8]		VOPT Channel Gain Compensation - Room Temp
0x01_80E1	21	33	VOGC[7:0]								
0x01_80EC	2C	44	COMP_VSG_COLD[7:0]								VSENSE Channel Gain Compensation - COLD Temp[112]
0x01_80ED	2D	45	COMP_VSG_HOT[7:0]								VSENSE Channel Gain Compensation - HOT Temp[112]
0x01_80EE	2E	46	COMP_VOG_COLD[7:0]								VOPT Channel Gain Compensation - COLD Temp[112]
0x01_80EF	2F	47	COMP_VOG_HOT[7:0]								VOPT Channel Gain Compensation - HOT Temp[112]
0x01_80F0	30	48	IGC4_COLD[7:0]								Current Channel Gain (4) Compensation - COLD Temp[112]
0x01_80F1	31	49	IGC4_HOT[7:0]								Current Channel Gain (4) Compensation - HOT Temp[112]

Table 104. Gain compensation buffer optimum (continued)

Global address	OFFSET		Byte description								Content
	HEX	DEC	7	6	5	4	3	2	1	0	
0x01_80F2	32	50	IGC8_COLD[7:0]								Current Channel Gain (8) Compensation - COLD Temp[112]
0x01_80F3	33	51	IGC8_HOT[7:0]								Current Channel Gain (8) Compensation - HOT Temp[112]
0x01_80F4	34	52	IGC16_COLD[7:0]								Current Channel Gain (16) Compensation - COLD Temp[112]
0x01_80F5	35	53	IGC16_HOT[7:0]								Current Channel Gain (16) Compensation - HOT Temp[112]
0x01_80F6	36	54	IGC32_COLD[7:0]								Current Channel Gain (32) Compensation - COLD Temp[112]
0x01_80F7	37	55	IGC32_HOT[7:0]								Current Channel Gain (32) Compensation - HOT Temp[112]
0x01_80F8	38	56	IGC64_COLD[7:0]								Current Channel Gain (64) Compensation - COLD Temp[112]
0x01_80F9	39	57	IGC64_HOT[7:0]								Current Channel Gain (64) Compensation - HOT Temp[112]
0x01_80FA	3A	58	IGC128_COLD[7:0]								Current Channel Gain (128) Compensation - COLD Temp[112]
0x01_80FB	3B	59	IGC128_HOT[7:0]								Current Channel Gain (128) Compensation - HOT Temp[112]
0x01_80FC	3C	60	IGC256_COLD[7:0]								Current Channel Gain (256) Compensation - COLD Temp[112]
0x01_80FD	3D	61	IGC256_HOT[7:0]								Current Channel Gain (256) Compensation - HOT Temp[112]
0x01_80FE	3E	62	IGC512_COLD[7:0]								Current Channel Gain (512) Compensation - COLD Temp[112]
0x01_80FF	3F	63	IGC512_HOT[7:0]								Current Channel Gain (512) Compensation - HOT Temp[112]

Notes

[112] 7-Bit character with bit 7 (MSB) as sign (0 = “+”; 1 = “-”) with the difference to the corresponding room temperature value (e.g. 10000010 = “-2”).

To create the look up table, a linear interpolation of the gain adjustment has to be done between the three given temperatures, based on the temperature step width specified (TCALSTEP).

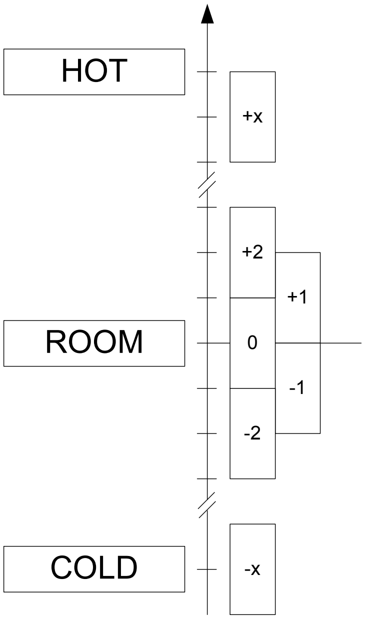


Figure 37. Look up table creation

4.8.5.1.3 Startup calibration

The power on trimming / calibration procedure is finalized by performing the start up calibration.

4.8.5.1.3.1 VSENSE/VOPT channel offset compensation and ITEMP channel gain/offset compensation

Copying the default compensation values, according to Table 105, will establish the optimum offset compensation for the VSENSE and VOPT channels, as well as the optimum gain and offset compensation for the internal temperature sensor.

Table 105. Voltage/temp trim

Global Address	OFFSET		Byte description								Target register	
	HEX	DEC	7	6	5	4	3	2	1	0	Name	Offset
0x01_80DA	1A	26	VOC_S[7:0]								COMP_VO	0xAA[113]
0x01_80DB	1B	27	VOC_O[7:0]								COMP_VO	0xAA[113]
0x01_80DC	1C	28	VOC_S_CHOP[7:0] (Chopper Mode)								COMP_VO	0xAA[113]
0x01_80DD	1D	29	VOC_O_CHOP[7:0] (Chopper Mode)								COMP_VO	0xAA[113]
0x01_80E2	22	34	ITO[7:0]								COMP_ITO	0xD0
0x01_80E3	23	35	ITG[7:0]								COMP_ITG	0xD1

Notes

[113] Based on the selection of the voltage measurement source (VSENSE or VOPT) and the activation of chopper mode. VOC_S is for VSENSE with Chopper mode OFF, VOC_O for VOPT with Chopper mode OFF, VOC_S_CHOP for VSENSE with Chopper mode ON, VOC_O_CHOP for VOPT with Chopper mode ON.

4.8.5.1.3.2 PGA auto zero sequence

The following procedure has to be performed for the PGA (Programmable Gain Amplifier) Auto Zero (AZ).

1. Write a “1” to the PGAO bit and its mask in the COMP_CTL register (0xA0)
2. Approximately 6.5 ms later, PGAOF will become set at to “1” (Flag needs to be polled)
3. Exit the PGAO mode by writing “0” in PGA0 and its mask being a “1”
4. Clear the PGAOF flag by writing “1”

NOTE

The new offset compensation data can be observed in the (PGAOC4...512[10:0]) registers.

The sequence will require 3352 clock cycles of the D2DFCLK (512kHz), typically 6.5 ms.

4.8.5.1.3.3 Current and voltage channel gain compensation based on ITEMP from look up table

After the first reading of the temperature channel measurement, the current and voltage channel gain compensation buffers must be written with the corresponding look up table value (see Section 4.8.5.1.2, “Gain compensation look up table").

4.8.5.1.3.4 Current channel offset compensation procedure (chopper off only)

If the chopper feature is not used for the current measurement channel, the offset should be compensated using the following procedure with the highest decimation selected and the LPF active.

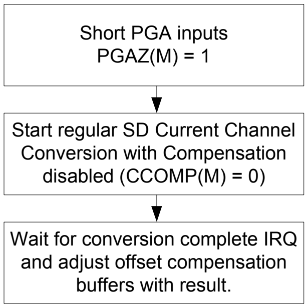

Short PGA inputs
PGAZ(M) = 1
Start regular SD Current Channel
Conversion with Compensation
disabled (CCOMP(M) = 0)
Wait for conversion complete IRQ
and adjust offset compensation
buffers with result.

Figure 38. Current channel offset compensation sequence

4.8.5.1.3.5 Program calibration IRQ temperature thresholds

To finalize the startup sequence, the new temperature limits must be programmed into the Calibration Temperature Limits (TCMAX[15:0] and TCMIN[15:0]), and the Calibration Request interrupt must be enabled.

4.8.5.2 Calibration request procedure

During normal system operation (in Normal and Low Power mode), a calibration request interrupt / wake-up will indicate the device temperature changed outside the range for the programmed Current and Voltage Channel Gain Compensation.

During a calibration request interrupt (wake-up), the Current and Voltage Channel Gain Compensation buffers have to be updated with the corresponding values stored in the look up table created upon system start up (see Section 4.8.5.1.2, “Gain compensation look up table"). The the new temperature limits must be programmed into the Calibration Temperature Limits (TCMAX[15:0] and TCMIN[15:0]) before leaving the interrupt service routine.

4.8.5.3 Verification procedures

As an optional feature, upon application requirement, the proper function of the current and voltage measurement channels can be verified by connecting a special calibration reference to the input of the channels.

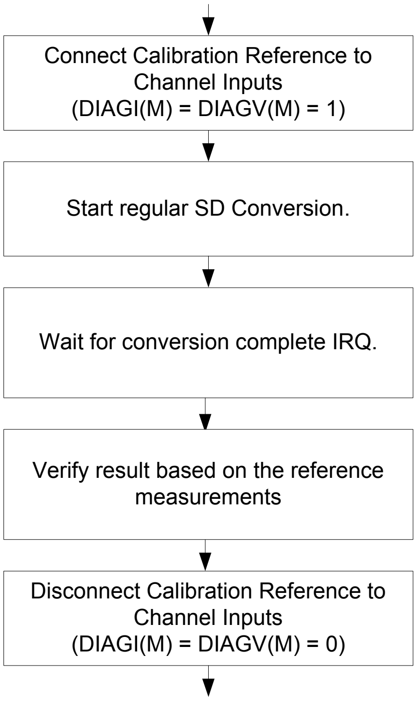

Connect Calibration Reference to Channel Inputs
(DIAGI(M) = DIAGV(M) = 1)
Start regular SD Conversion.
Wait for conversion complete IRQ.
Verify result based on the reference measurements
Disconnect Calibration Reference to Channel Inputs
(DIAGI(M) = DIAGV(M) = 0)

Table 106 shows the location of the diagnostic reference measurements.

Note: This table is unpopulated for Analog Option 1 devices.

Table 106. Diagnostic measurement flash location

Global Address	OFFSET		Byte description							
	HEX	DEC	7	6	5	4	3	2	1	0
0x01_80E4	24	36	BG3 diag measurement from Vsense channel after cal at room							
0x01_80E5	25	37								
0x01_80E6	26	38	BG3 diag measurement from Vopt channel after cal at room							
0x01_80E7	27	39								
0x01_80E8	28	40	BG3 diag measurement from I channel (gain4) at room							
0x01_80E9	29	41								
0x01_80EA	2A	42								

4.8.6 Memory map and registers

4.8.6.1 Overview

This section provides a detailed description of the memory map and registers.

4.8.6.2 Module memory map

The memory map for the Acquisition, Compensation, and LPF module is given in Table 62.

Table 107. Module memory map

Offset [114][115]	Name		7	6	5	4	3	2	1	0
0x58	ACQ_CTL Acquisition control register	R	0	0	0	0	0	0	0	0
		W	AHCRM	OPTEM	OPENEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
		R	0	OPTE	OPENE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
		W	AHCR							
0x5A	ACQ_SR (hi)	R	AVRF	PGAG	VMOW	CMOW	ETM	ITM	VM	CM
	Acquisition status register	W	Write 1 will clear the flags							
0x5B	ACQ_SR (lo)	R	OPEN	0	0	VTH	ETCHOP	ITCHOP	VCHOP	CCHOP
	Acquisition status register	W								
0x5C	ACQ_ACC1 Acquisition chain control 1	R	0	0	0	0	0	0	0	0
		W	TCOMPM	VCOMPM	CCOMPM	LPFENM	ETCHOP M	ITCHOPM	CVCHOP M	AGENM
		R	TCOMP	VCOMP	CCOMP	LPFEN	ETCHOP	ITCHOP	CVCHOP	AGEN
		W								
0x5E	ACQ_ACC0 Acquisition chain control 0	R	0	0	0	0	0	0	0	0
		W	ZEROM	ECAPM	TADCGM	VADCGM	CADCGM	TDENM	VDENM	CDENM
		R	ZERO	ECAP	TADCG	VADCG	CADCG	TDEN	VDEN	CDEN
		W								
0x60	ACQ_DEC	R	0	0	0	0	0	DEC[2:0]		
	Decimation Rate	W								
0x61	ACQ_BGC	R	0	0	BGADC[1:0]		BGLDO	BG3EN	BG2EN	BG1EN
	BandGap control	W								
0x62	ACQ_GAIN	R	0	0	0	0	0	IGAIN[2:0]		
	PGA gain	W								
0x63	ACQ_GCB	R	D[7:0]							
	GCB threshold	W								
0x64	ACQ_ITEMP (hi)	R	ITEMP[15:8]							
	Internal temp. measurement result	W								
0x65	ACQ_ITEMP (lo)	R	ITEMP[7:0]							
	Internal temp. measurement result	W								
0x66	ACQ_ETEMP (hi)	R	EEMP[15:8]							
	External temp. measurement result	W								
0x67	ACQ_ETEMP (lo)	R	EEMP[7:0]							
	External temp. measurement result	W								
0x68	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x69	ACQ_CURR1	R	CURR[23:16]							
	Current measurement result	W								
0x6A	ACQ_CURR0 Current measurement result	R	CURR[15:8]							
		W								
		R	CURR[7:0]							
		W								

Table 107. Module memory map (continued)

Offset (114)(115)	Name		7	6	5	4	3	2	1	0
0x6C	ACQ_VOLT Voltage measurement result	R	VOLT[15:8]							
		W								
		R	VOLT[7:0]							
		W								
0x6E	ACQ_LPFC	R	0	0	0	0	LPFC[3:0]			
	Low pass filter coefficient number	W								
0x6F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x70	ACQ_TCMP Low power trigger current measurement period	R	TCMP[15:0]							
		W								
		R								
		W								
0x72	ACQ_THF	R	THF[7:0]							
	Low power current threshold filtering period	W								
0x73	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x74	ACQ_CVCR (hi)	R	0	0	0	0	0	0	0	0
	I and V chopper control register	W			DBTM[1:0]		IIRCM[2:0]			PGAFM
0x75	ACQ_CVCR (lo)	R	0	0	DBT[1:0]		IIRC[2:0]			PGAF
	I and V chopper control register	W								
0x76	ACQ_CTH	R	CTH[7:0]							
	Low power current threshold	W								
0x77	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x78	ACQ_AHTH1 (hi)	R	0							
	Low power Ah counter threshold	W				AHTH[30:16]				
0x79	ACQ_AHTH1 (lo)	R								
	Low power Ah counter threshold	W								
0x7A	ACQ_AHTH0 (hi)	R	AHTH[15:0]							
	Low power Ah counter threshold	W								
0x7B	ACQ_AHTH0 (lo)	R								
	Low power Ah counter threshold	W								
0x7C	ACQ_AHC1 (hi)	R	AHC[31:24]							
	Low power Ah counter	W								
0x7D	ACQ_AHC1 (lo)	R	AHC[23:16]							
	Low power Ah counter	W								
0x7E	ACQ_AHC0 (hi)	R	AHC[15:8]							
	Low power Ah counter	W								
0x7F	ACQ_AHC0 (lo)	R	AHC[7:0]							
	Low power Ah counter	W								
0x80	LPF_A0 (hi)	R	A0[15:0]							
	A0 filter coeff	W								
0x81	LPF_A0 (lo)	R								
	A0 filter coeff	W								

Table 107. Module memory map (continued)

Offset (114)(115)	Name		7	6	5	4	3	2	1	0
0x82	LPF_A1 (hi)	R	A1[15:0]							
	A1 filter coeff	W								
0x83	LPF_A1 (lo)	R								
	A1 filter coeff	W								
0x84	LPF_A2 (hi)	R	A2[15:0]							
	A2 filter coeff	W								
0x85	LPF_A2 (lo)	R								
	A2 filter coeff	W								
0x86	LPF_A3 (hi)	R	A3[15:0]							
	A3 filter coeff	W								
0x87	LPF_A3 (lo)	R								
	A3 filter coeff	W								
0x88	LPF_A4 (hi)	R	A4[15:0]							
	A4 filter coeff	W								
0x89	LPF_A4 (lo)	R								
	A4 filter coeff	W								
0x8A	LPF_A5 (hi)	R	A5[15:0]							
	A5 filter coeff	W								
0x8B	LPF_A5 (lo)	R								
	A5 filter coeff	W								
0x8C	LPF_A6 (hi)	R	A6[15:0]							
	A6 filter coeff	W								
0x8D	LPF_A6 (lo)	R								
	A6 filter coeff	W								
0x8E	LPF_A7 (hi)	R	A7[15:0]							
	A7 filter coeff	W								
0x8F	LPF_A7 (lo)	R								
	A7 filter coeff	W								
0x90	LPF_A8 (hi)	R	A8[15:0]							
	A8 filter coeff	W								
0x91	LPF_A8 (lo)	R								
	A8 filter coeff	W								
0x92	LPF_A9 (hi)	R	A9[15:0]							
	A9 filter coeff	W								
0x93	LPF_A9 (lo)	R								
	A9 filter coeff	W								
0x94	LPF_A10 (hi)	R	A10[15:0]							
	A10 filter coeff	W								
0x95	LPF_A10 (lo)	R								
	A10 filter coeff	W								
0x96	LPF_A11 (hi)	R	A11[15:0]							
	A11 filter coeff	W								
0x97	LPF_A11 (lo)	R								
	A11 filter coeff	W								

Table 107. Module memory map (continued)

Offset [114][115]	Name		7	6	5	4	3	2	1	0
0x98	LPF_A12 (hi)	R	A12[15:0]							
	A12 filter coeff	W								
0x99	LPF_A12 (lo)	R								
	A12 filter coeff	W								
0x9A	LPF_A13 (hi)	R	A13[15:0]							
	A13 filter coeff	W								
0x9B	LPF_A13 (lo)	R								
	A13 filter coeff	W								
0x9C	LPF_A14 (hi)	R	A14[15:0]							
	A14 filter coeff	W								
0x9D	LPF_A14 (lo)	R								
	A14 filter coeff	W								
0x9E	LPF_A15 (hi)	R	A15[15:0]							
	A15 filter coeff	W								
0x9F	LPF_A15 (lo)	R								
	A15 filter coeff	W								
0xA0	COMP_CTL Compensation control register	R	0	0	0	0	0	0		0
		W	BGCALM[1:0]		PGAZM	PGAOM	DIAGVM	DIAGIM		CALIEM
		R	BGCAL[1:0]		PGAZ	PGAO	DIAGV	DIAGI		CALIE
		W								
0xA2	COMP_SR	R	0	BGRF	0	PGAOF	0	0	0	CALF
	Compensation status register	W	Write 1 will clear the flags							
0xA3	COMP_TF	R	0	0	0	0	0	TMF[2:0]		
	Temperature filtering period	W								
0xA4	COMP_TMAX Max temp before recalibration	R	TCMAX[15:0]							
		W								
		R								
		W								
0xA6	COMP_TMIN Min temp before recalibration	R	TCMIN[15:0]							
		W								
		R								
		W								
0xA8	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xA9	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xAA	COMP_VO	R	VOC[7:0]							
	Offset voltage compensation	W								
0xAB	COMP_IO	R	COC[7:0]							
	Offset current compensation window	W								
0xAC	COMP_VSG Gain voltage comp. vsense channel	R	0	0	0	0	0	0	VSGC[9:8]	
		W								
		R	VSGC[7:0]							
		W								

Table 107. Module memory map (continued)

Offset[114][115]	Name		7	6	5	4	3	2	1	0
0xAE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xAF	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xB0	COMP_IG4 Gain current compensation 4	R	0	0	0	0	0	0	IGC4[9:8]	
		W								
		R	IGC4[7:0]							
		W								
0xB2	COMP_IG8 Gain current compensation 8	R	0	0	0	0	0	0	IGC8[9:8]	
		W								
		R	IGC8[7:0]							
		W								
0xB4	COMP_IG16 Gain current compensation 16	R	0	0	0	0	0	0	IGC16[9:8]	
		W								
		R	IGC16[7:0]							
		W								
0xB6	COMP_IG32 Gain current compensation 32	R	0	0	0	0	0	0	IGC32[9:8]	
		W								
		R	IGC32[7:0]							
		W								
0xB8	COMP_IG64 Gain current compensation 64	R	0	0	0	0	0	0	IGC64[9:8]	
		W								
		R	IGC64[7:0]							
		W								
0xBA	COMP_IG128 Gain current compensation 128	R	0	0	0	0	0	0	IGC128[9:8]	
		W								
		R	IGC128[7:0]							
		W								
0xBC	COMP_IG256 Gain current compensation 256	R	0	0	0	0	0	0	IGC256[9:8]	
		W								
		R	IGC256[7:0]							
		W								
0xBE	COMP_IG512 Gain current compensation 512	R	0	0	0	0	0	0	IGC512[9:8]	
		W								
		R	IGC512[7:0]							
		W								
0xC0	COMP_PGAO4 Offset PGA compensation 4	R	0	0	0	0	0	PGAOC4[10:8]		
		W								
		R	PGAOC4[7:0]							
		W								
0xC2	COMP_PGAO8 Offset PGA compensation 8	R	0	0	0	0	0	PGAOC8[10:8]		
		W								
		R	PGAOC8[7:0]							
		W								

Table 107. Module memory map (continued)

Offset [114][115]	Name		7	6	5	4	3	2	1	0
0xC4	COMP_PGAO16 Offset PGA compensation 16	R	0	0	0	0	0	PGAOC16[10:8]		
		W								
		R	PGAOC16[7:0]							
		W								
0xC6	COMP_PGAO32 Offset PGA compensation 32	R	0	0	0	0	0	PGAOC32[10:8]		
		W								
		R	PGAOC32[7:0]							
		W								
0xC8	COMP_PGAO64 Offset PGA compensation 64	R	0	0	0	0	0	PGAOC64[10:8]		
		W								
		R	PGAOC64[7:0]							
		W								
0xCA	COMP_PGAO128 Offset PGA compensation 128	R	0	0	0	0	0	PGAOC128[10:8]		
		W								
		R	PGAOC128[7:0]							
		W								
0xCC	COMP_PGAO256 Offset PGA compensation 256	R	0	0	0	0	0	PGAOC256[10:8]		
		W								
		R	PGAOC256[7:0]							
		W								
0xCE	COMP_PGAO512 Offset PGA compensation 512	R	0	0	0	0	0	PGAOC512[10:8]		
		W								
		R	PGAOC512[7:0]							
		W								
0xD0	COMP_ITO	R	ITOC[7:0]							
	Internal temp. offset compensation	W								
0xD1	COMP_ITG	R	ITGC[7:0]							
	Internal temp. gain compensation	W								
0xD2	COMP_ETO	R	ETOC[7:0]							
	External temp. offset compensation	W								
0xD3	COMP_ETG	R	ETGC[7:0]							
	External temp. gain compensation	W								
0xD4	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD5	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD6	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xD7	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes

114. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
115. Register Offset with the “lo” address value not shown have to be accessed in 16Bit mode. 8 Bit access will not function.

4.8.6.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

4.8.6.3.1 Acquisition control register (ACQ_CTL)

Table 108. Acquisition control register (ACQ_CTL)

Offset [116],[117] 0x58

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	AHCRM	OPTEM	OPENEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	0	OPTE	OPENE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
W	AHCR							
Reset	0	0	0	0	0	0	0	0

Notes

[116] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[117] This Register is 16-Bit access only.

Table 109. Acquisition control register (ACQ_CTL) - register field descriptions

Field	Description
15 AHCRM	Ampere Hour Counter Reset - Mask 0 - writing the AHCR Bit will have no effect 1 - writing the AHCR Bit will be effective
14 OPTEM	Optional Voltage Sense Enable - Mask 0 - writing the OPTE Bit will have no effect 1 - writing the OPTE Bit will be effective
13 OPENEM	Enable Shunt Resistor Open Detection - Mask 0 - writing the OPENE Bit will have no effect 1 - writing the OPENE Bit will be effective
12 CVMIEM	Current / Voltage Measurement Interrupt Enable - Mask 0 - writing the CVMIE Bit will have no effect 1 - writing the CVMIE Bit will be effective
11 ETMENM	External Temperature Measurement Enable - Mask 0 - writing the ETMEN Bit will have no effect 1 - writing the ETMEN Bit will be effective
10 ITMENM	Internal Temperature Measurement Enable - Mask 0 - writing the ITMEN Bit will have no effect 1 - writing the ITMEN Bit will be effective
9 VMENM	Voltage Measurement Enable - Mask 0 - writing the VMEN Bit will have no effect 1 - writing the VMEN Bit will be effective
8 CMENM	Current Measurement Enable - Mask 0 - writing the CMEN Bit will have no effect 1 - writing the CMEN Bit will be effective

Table 109. Acquisition control register (ACQ_CTL) - register field descriptions (continued)

Field	Description
7 AHCR	Ampere Hour Counter Reset, this write only bit will reset the ACQ_AHC register. 0 - no effect 1 - ACQ_AHC reset to 0x00000000
6 OPTE	Optional Voltage Sense Enable (Voltage Channel Multiplexer Control) 0 - VSENSE routed to ADC 1 - VOPT routed to ADC
5 OPENE	Enable Shunt Resistor Open Detection 0 - Shunt resistor open detection disabled, the OPEN bit must be ignored 1 - Shunt resistor open detection enabled, OPEN bit will indicate status
4 CVMIE	Current / Voltage Measurement Interrupt Enable 0 - current and voltage measurement interrupt disabled 1 - current and voltage measurement interrupt enabled
3 ETMEN	External Temperature Measurement Enable 0 - external temperature measurement disabled 1 - external temperature measurement enabled
2 ITMEN	Internal Temperature Measurement Enable 0 - internal temperature measurement disabled 1 - internal temperature measurement enabled
1 VMEN	Voltage Measurement Enable 0 - voltage measurement disabled 1 - voltage measurement enabled
0 CMEN	Current Measurement Enable 0 - current measurement disabled 1 - current measurement enabled

4.8.6.3.2 Acquisition status register (ACQ_SR (hi))

Table 110. Acquisition status register (ACQ_SR (hi))

Offset[118] 0x5A

Access: User read/write

	7	6	5	4	3	2	1	0
R	AVRF	PGAG	VMOW	CMOW	ETM	ITM	VM	CM
W	Write 1 will clear the flags							
Reset	0	0	0	0	0	0	0	0

Notes

[118] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 111. Acquisition status register (ACQ_SR (hi)) - register field descriptions

Field	Description
7 AVRF	VDDA Low Voltage Reset Flag. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a low voltage condition at the VDDA regulator.
6 PGAG	PGA Gain Change Flag[119]. Writing this bit to logic 1 will clear the flag. 0 - PGA gain has not changed since last flag clear 1 - PGA gain has changed since last flag clear
5 VMOW	Voltage Measurement Result Overwritten[119]. Writing this bit to logic 1 will clear the flag. 0 - Voltage measurement result register VOLT[15:0] not overwritten[120] since last VMOW flag clear 1 - Voltage measurement result register VOLT[15:0] overwritten[120] since last VMOW flag clear

Table 111. Acquisition status register (ACQ_SR (hi)) - register field descriptions (continued)

Field	Description
4 CMOW	Current Measurement Result Overwritten[119]. Writing this bit to logic 1 will clear the flag. 0 - Current measurement result register CURR[15:0] not overwritten[120] since last CMOW flag clear 1 - Current measurement result register CURR[15:0] overwritten[120] since last CMOW flag clear
3 ETM	End of Measurement - External Temperature[119]. Writing this bit to logic 1 will clear the flag. 0 - No external temperature measurement completed since last ETM clear 1 - External temperature measurement completed since last ETM clear
2 ITM	End of Measurement - Internal Temperature[119]. Writing this bit to logic 1 will clear the flag. 0 - No internal temperature measurement completed since last ITM clear 1 - Internal temperature measurement completed since last ITM clear
1 VM	End of Measurement - Voltage. Writing this bit to logic 1 will clear the flag. 0 - No voltage measurement completed since last VM clear 1 - Voltage measurement completed since last VM clear
0 CM	End of Measurement - Current. Writing this bit to logic 1 will clear the flag. 0 - No current measurement completed since last CM clear 1 - Current measurement completed since last CM clear

Notes

[119] No Interrupts issued for those flags

[120] Overwritten - new result latched before previous result was read

4.8.6.3.3 Acquisition status register (ACQ_SR (lo))

Table 112. Acquisition status register (ACQ_SR (lo))

Offset[121] 0x5B

Access: User read

	7	6	5	4	3	2	1	0
R	OPEN	0	0	VTH	ETCHOP	ITCHOP	VCHOP	CCHOP
W								

Notes

[121] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 113. Acquisition status register (ACQ_SR (lo)) - register field descriptions

Field	Description
7 OPEN	Shunt Resistor Open Detection Status (Normal mode only, only functional if OPENE=1) 0 - Shunt resistor detected 1 - Shunt resistor disconnected
4 VTH	Digital Voltage High Threshold Reached 0 - Voltage measurement result for VSENSE / VOPT below V_{TH} (0xDAC0: equivalent to 28 V at 0.5 mV LSB weighing) 1 - Voltage measurement result for VSENSE / VOPT above or equal V_{TH} (0xDAC0: equivalent to 28 V at 0.5 mV LSB weighing)
3 ETCHOP	Chopping Active Status - External Temperature 0 - Chopper for external temperature measurement disabled 1 - Chopper for external temperature measurement enabled
2 ITCHOP	Chopping Active Status - Internal Temperature 0 - Chopper for internal temperature measurement disabled 1 - Chopper for internal temperature measurement enabled

Table 113. Acquisition status register (ACQ_SR (lo)) - register field descriptions (continued)

Field	Description
1 VCHOP	Chopping Active Status - Voltage 0 - Chopper for voltage measurement disabled 1 - Chopper for voltage measurement enabled
0 CCHOP	Chopping Active Status - Current 0 - Chopper for current measurement disabled 1 - Chopper for current measurement enabled

4.8.6.3.4 Acquisition chain control 1 (ACQ_ACC1)

Table 114. Acquisition chain control 1 (ACQ_ACC1)

Offset [122][123] 0x5C

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	TCOMPM	VCOMPM	CCOMPM	LPFENM	ETCHOPM	ITCHOPM	CVCHOPM	AGENM
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	TCOMP	VCOMP	CCOMP	LPFEN	ETCHOP	ITCHOP	CVCHOP	AGEN
Reset	1	1	1	0	0	0	0	1

Notes

[122] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[123] This Register is 16 Bit access only.

Table 115. Acquisition chain control 1 (ACQ_ACC1) - register field descriptions

Field	Description
15 TCOMPM	Temperature Measurement Channel - Compensation Enable - Mask 0 - writing the TCOMP bit will have no effect 1 - writing the TCOMP bit will be effective
14 VCOMPM	Voltage Measurement Channel - Compensation Enable - Mask 0 - writing the VCOMP bit will have no effect 1 - writing the VCOMP bit will be effective
13 CCOMPM	Current Measurement Channel - Compensation Enable - Mask 0 - writing the CCOMP bit will have no effect 1 - writing the CCOMP bit will be effective
12 LPFENM	LPF Enable - Mask 0 - writing the LPFEN bit will have no effect 1 - writing the LPFEN bit will be effective
11 ETCHOPM	Chopping Enable - External Temperature Measurement Channel - Mask 0 - writing the ETCHOP bit will have no effect 1 - writing the ETCHOP bit will be effective
10 ITCHOPM	Chopping Enable - Internal Temperature Measurement Channel - Mask 0 - writing the ITCHOP bit will have no effect 1 - writing the ITCHOP bit will be effective

Table 115. Acquisition chain control 1 (ACQ_ACC1) - register field descriptions (continued)

Field	Description
9 CVCHOPM	Chopping Enable - Voltage Measurement Channel - Mask 0 - writing the CVCHOP bit will have no effect 1 - writing the CVCHOP bit will be effective
8 AGENM	Automatic Gain Control Enable - Mask 0 - writing the AGEN bit will have no effect 1 - writing the AGEN bit will be effective
7 TCOMP	Temperature Measurement Channel - Compensation Enable 0 - Temperature measurement channel offset and gain compensation disabled 1 - Temperature measurement channel offset and gain compensation enabled
6 VCOMP	Voltage Compensation Enable 0 - Voltage measurement channel offset and gain compensation disabled 1 - Voltage measurement channel offset and gain compensation enabled
5 CCOMP	Current Compensation Enable 0 - Current measurement channel offset and gain compensation disabled 1 - Current measurement channel offset and gain compensation enabled
4 LPFEN	LPF Enable 0 - Low pass filter for current and voltage channel disabled 1 - Low pass filter for current and voltage channel enabled
3 ETCHOP	Chopping Enable - External Temperature 0 - Chopper mode for external temperature measurement disabled 1 - Chopper mode for external temperature measurement enabled
2 ITCHOP	Chopping Enable - Internal Temperature 0 - Chopper mode for internal temperature measurement disabled 1 - Chopper mode for internal temperature measurement enabled
1 CVCHOP	Chopping Enable - Voltage 0 - Chopper mode for voltage and current measurement disabled 1 - Chopper mode for voltage and current measurement enabled
0 AGEN	Automatic Gain Control Enable 0 - Automatic gain control disabled (manual gain control via IGAIN[2:0]) 1 - Automatic gain control enabled

4.8.6.3.5 Acquisition chain control 0 (ACQ_ACC0)

Table 116. Acquisition chain control 0 (ACQ_ACC0)

Offset [124],[125] 0x5E

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	ZEROM	ECAPM	TADCGM	VADCGM	CADCGM	TDENM	VDENM	CDENM
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	ZERO	ECAP	TADCG	VADCG	CADCG	TDEN	VDEN	CDEN
Reset	0	0	1	1	1	0	0	0

Notes

[124] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[125] This Register is 16 Bit access only.

Table 117. Acquisition chain control 0 (ACQ_ACC0) - register field descriptions

Field	Description
15 ZEROM	Current and Voltage Sigma Delta Input Short - Mask 0 - writing the ZERO bit will have no effect 1 - writing the ZERO bit will be effective
14 ECAPM	TSUP External Capacitor - Mask 0 - writing the ECAP bit will have no effect 1 - writing the ECAP bit will be effective
13 TADCGM	Temperature ADC Gain Select - Mask 0 - writing the TADCG bit will have no effect 1 - writing the TADCG bit will be effective
12 VADCGM	Voltage ADC Gain Select - Mask 0 - writing the VADCG bit will have no effect 1 - writing the VADCG bit will be effective
11 CADCGM	Current ADC Gain Select - Mask 0 - writing the CADCG bit will have no effect 1 - writing the CADCG bit will be effective
10 TDENM	100ns Clock delay - Internal Temperature - Mask 0 - writing the TDEN bit will have no effect 1 - writing the TDEN bit will be effective
9 VDENM	100ns Clock delay - Voltage - Mask 0 - writing the VDEN bit will have no effect 1 - writing the VDEN bit will be effective
8 CDENM	100ns Clock delay - Current - Mask 0 - writing the CDEN bit will have no effect 1 - writing the CDEN bit will be effective
7 ZERO	Current and Voltage Sigma Delta Input Short (to perform Offset Compensation measurement) 0 - Sigma delta inputs not shorted 1 - Current and voltage sigma delta inputs shorted
6 ECAP	TSUP External Capacitor select 0 - TSUP frequency compensation disabled. No capacitor at pin. 0 - TSUP frequency compensation enabled. Capacitor C_{TSUP} allowed at pin.
5 TADCG	Temperature ADC Gain Select; Test purpose only, Default value (1) must be used 0 - Temperature ADC - gain adjustment 1 - Temperature ADC - standard gain (default)
4 VADCG	Voltage ADC Gain Select; Test purpose only; Default value (1) must be used 0 - Voltage ADC - gain adjustment 1 - Voltage ADC - standard gain (default)
3 CADCG	Current ADC Gain Select; Test purpose only; Default value (1) must be used 0 - Current ADC - gain adjustment 1 - Current ADC - standard gain (default)
2 TDEN	Timing delay - Temperature 0 - standard timing for temperature measurement channel 1 - additional SD converter input delay (typ. 100 ns) for temperature measurement channel
1 VDEN	Timing delay - Voltage 0 - standard timing for Voltage measurement channel 1 - additional SD converter input delay (typ. 100 ns) for voltage measurement channel
0 CDEN	Timing delay - Current 0 - standard timing for current measurement channel 1 - additional SD converter input delay (typ. 100 ns) for current measurement channel

4.8.6.3.6 Decimation rate (ACQ_DEC)

Table 118. Decimation rate (ACQ_DEC)

Offset[126] 0x60

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	DEC[2:0]		
W								
Reset	0	0	0	0	0	1	0	0

Notes

126. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 119. Decimation rate (ACQ_DEC) - register field descriptions

Field	Description
2-0 DEC[2:0]	Decimation Rate Selection (Combined decimation rate of first and second sinc3 decimator; Fist decimator is fixed to D=8) 000 - D = 512 (Channel Output Rate = 1.0 kHz) 001 - D = 64 (Channel Output Rate = 8.0 kHz) 010 - D = 128 (Channel Output Rate = 4.0 kHz) 011 - D = 256 (Channel Output Rate = 2.0 kHz) 100 - D = 512 (Channel Output Rate = 1.0 kHz), (default) 101 - D = 1024 (Channel Output Rate = 500 Hz) 110 - D = 512 (Channel Output Rate = 1.0 kHz) 111 - D = 512 (Channel Output Rate = 1.0 kHz)

4.8.6.3.7 BandGap control (ACQ_BGC)

Table 120. BandGap control (ACQ_BGC)

Offset[127] 0x61

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	BGADC		BGLDO	BG3EN	BG2EN	BG1EN
W								
Reset	0	0	0	1	1	0	0	0

Notes

127. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 121. BandGap control (ACQ_BGC) - register field descriptions

Field	Description
5-4 BGADC	ADC Bandgap select 00 - n.a. (not allowed - VDDA Reset) 01 - BG1 reference selected for the AD converters (default) 10 - BG2 reference selected for the AD converters 11 - BG3 reference selected for the AD converters
3 BGLDO	LDO (Low Dropout Regulator) Bandgap select 0 - BG2 selected as voltage regulator reference 1 - BG1 selected as voltage regulator reference (default)
2 BG3EN	Bandgap 3 Status 0 - Bandgap 3 disabled 1 - Bandgap 3 enabled

Table 121. BandGap control (ACQ_BGC) - register field descriptions

Field	Description
1 BG2EN	Bandgap 2 Status 0 - Bandgap 2 disabled 1 - Bandgap 2 enabled
0 BG1EN	Bandgap 1 Status 0 - Bandgap 1 disabled 1 - Bandgap 1 enabled

4.8.6.3.8 PGA gain (ACQ_GAIN)

Table 122. PGA gain (ACQ_GAIN)

Offset[128] 0x62

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	IGAIN[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

Notes

[128] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 123. PGA gain (ACQ_GAIN) - register field descriptions

Field	Description
2-0 IGAIN[2:0]	PGA Gain Register - Writing will select (manually override) the PGA gain if the automatic gain control is disabled (AGEN=0). Reading will return current gain setting (including the auto gain). The register content will also determine the current channel offset compensation buffer accessed through the COC[7:0] register. 000 - PGA Gain = 4 001 - PGA Gain = 8 010 - PGA Gain = 16 011 - PGA Gain = 32 100 - PGA Gain = 64 101 - PGA Gain = 128 110 - PGA Gain = 256 111 - PGA Gain = 512

4.8.6.3.9 GCB threshold (ACQ_GCB)

Table 124. GCB threshold (ACQ_GCB)

Offset[129] 0x63

Access: User read/write

	7	6	5	4	3	2	1	0
R	D (hi)				D (lo)			
W								
Reset	0	0	0	0	0	0	0	0

Notes

[129] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 125. GCB threshold (ACQ_GCB) - register field descriptions

Field	Description
7-4 D[7:4]	Gain Control Block (GCB) - 4 Bit Gain “Up” Threshold. See Section 4.8.3.1.3, “Gain control block (GCB)".
3-0 D[3:0]	Gain Control Block (GCB) - 4 Bit Gain “Down” Threshold. See Section 4.8.3.1.3, “Gain control block (GCB)".

4.8.6.3.10 Internal temp. measurement result (ACQ_ITEMP (hi) / ACQ_ITEMP (lo))

Table 126. Internal temp. measurement result (ACQ_ITEMP (hi) / ACQ_ITEMP (lo))

Offset[130] 0x64 / 0x65 Access: User read

	7	6	5	4	3	2	1	0
R	ITEMP[15:8]							
W								
R	ITEMP[7:0]							
W								

Notes

[130] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 127. Internal Temp. Measurement Result (ACQ_ITEMP (hi) / ACQ_ITEMP (lo)) - Register Field Descriptions

Field	Description
15-0 ITEMP[15:0]	Internal Temperature Measurement - 16 Bit ADC Result Register (unsigned Integer)

4.8.6.3.11 External temp. measurement result (ACQ_ETEMP (hi) / ACQ_ETEMP (lo))

Table 128. External temp. measurement result (ACQ_ETEMP (hi) / ACQ_ETEMP (lo))

Offset[131] 0x66 / 0x67 Access: User read

	7	6	5	4	3	2	1	0
R	ETEMP[15:8]							
W								
R	ETEMP[7:0]							
W								

Notes

[131] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 129. External temp. measurement result (ACQ_ETEMP (hi) / ACQ_ETEMP (lo)) - register field descriptions

Field	Description
15-0 ETEMP[15:0]	External Temperature Measurement - 16 Bit ADC Result Register (unsigned Integer)

4.8.6.3.12 Current measurement result (ACQ_CURR1 / ACQ_CURR0)

Table 130. Current measurement result (ACQ_CURR1 / ACQ_CURR0)

Offset[132] 0x69[133] / 0x6A[134]

Access: User read

	7	6	5	4	3	2	1	0
R	CURR[23:16]							
W								
R	CURR[15:8]							
W								
R	CURR[7:0]							
W								

Notes

[132] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[133] 0x69 for 8-Bit access. 0x68 for 16-Bit access.

[134] This Register is 16-Bit access only.

Table 131. Current measurement result (ACQ_CURR1 / ACQ_CURR0) - register field descriptions

Field	Description
CURR[23:0]	Two's complement 24-Bit signed integer result register for the current measurement channel.
23-16 CURR[23:16]	Current Measurement - High Byte Result Register, 8 or 16-Bit read operation.
15-0 CURR[15:0]	Current Measurement - Low Word Result Register, 16-Bit read operation only.

4.8.6.3.13 Voltage measurement result (ACQ_VOLT)

Table 132. Voltage measurement result (ACQ_VOLT)

Offset [135][136] 0x6C

Access: User read

	7	6	5	4	3	2	1	0
R	VOLT[15:8]							
W								
R	VOLT[7:0]							
W								

Notes

[135] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[136] This Register is 16-Bit access only.

Table 133. Voltage measurement result (ACQ_VOLT) - register field descriptions

Field	Description
15-0 VOLT[15:0]	Unsigned 16-Bit integer result register for the voltage measurement channel.

4.8.6.3.14 Low pass filter coefficient number (ACQ_LPFC)

Table 134. Low pass filter coefficient number (ACQ_LPFC)

Offset[137] 0x6E

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	LPFC[3:0]			
W								
Reset	0	0	0	0	1	1	1	0

Notes

137. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 135. Low pass filter coefficient number (ACQ_LPFC) - register field descriptions

Field	Description
3-0 LPFC[3:0]	Low Pass Filter Coefficient Number. Defines the highest coefficient Number used. 0000 - LPF used with Coefficient A0 0001 - LPF used with Coefficient A0...A1 .... 1111 - LPF used with Coefficient A0...A15

4.8.6.3.15 Low power trigger current measurement period (ACQ_TCMP)

Table 136. Low power trigger current measurement period (ACQ_TCMP)

Offset[138][139] 0x70

Access: User read / write

	7	6	5	4	3	2	1	0
R W	TCMP[15:8]							
Reset	0	0	0	0	0	0	0	0
R W	TCMP[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes

138. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
139. This Register is 16-Bit access only.

Table 137. Low power trigger current measurement period (ACQ_TCMP) - register field descriptions

Field	Description
15-0 TCMP[15:0]	Low power trigger current measurement period (Trigger counter based on ALFCLK). See Section 4.3.4.1.1, “Cyclic current acquisition/calibration temperature check".

NOTE

The cyclic acquisition period must be greater than the acquisition time. See Section 4.8.3.5.2, “Latency and throughput"” for estimation. A continuous acquisition is still possible by using TCMP=0.

The Low Power Trigger Current counter is an up counting counter starting at 0. It is increment according to the Low Power Clock. In Low Power mode, when Low Power Trigger Current counter is equal to the Low Power Trigger Currrent Measurement Period, the device will start a current and temperature acquisition according to the setting of the current and internal temperature acquisition channels.

4.8.6.3.16 Low power current threshold filtering period (ACQ_THF)

Table 138. Low power current threshold filtering period (ACQ_THF)

Offset[140] 0x72 Access: User read / write

	7	6	5	4	3	2	1	0
R W	THF[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes

[140] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 139. Low power current threshold filtering period (ACQ_THF) - register field descriptions

Field	Description
7-0 THF[7:0]	Low power current threshold wake up filtering period. See Section 4.3.4.1.1, “Cyclic current acquisition/calibration temperature check".

4.8.6.3.17 I and V chopper control register (ACQ_CVCR (hi))

Table 140. I and V chopper control register (ACQ_CVCR (hi))

Offset[141] 0x74 Access: User write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W			DBTM		IIRCM			PGAFM
Reset	0	0	0	0	0	0	0	0

Notes

[141] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 141. I and V chopper control register (ACQ_CVCR (hi)) - register field descriptions

Field	Description
5-4 DBTM[1:0]	Hold Time After Chopper Swap - Mask 0 - writing the DBT bits will have no effect 1 - writing the DBT bits will be effective
3-1 IIRCM[2:0]	IIR Low Pass Filter Configuration - Mask 0 - writing the IIRC bits will have no effect 1 - writing the IIRC bits will be effective
0 PGAFM	PGA fast mode enable - Mask 0 - writing the PGAF bit will have no effect 1 - writing the PGAF bit will be effective

4.8.6.3.18 I and V chopper control register (ACQ_CVCR (lo))

Table 142. I and V chopper control register (ACQ_CVCR (lo))

Offset[142] 0x75 Access: User write

	7	6	5	4	3	2	1	0
R	0	0	DBT		IIRC			PGAF
W								
Reset	0	0	0	0	1	1	1	1

Notes

[142] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 143. I and V chopper control register (ACQ_CVCR (lo)) - register field descriptions

Field	Description
5-4 DBT[1:0]	Hold Time After Chopper Swap 00 - Hold after swap disabled 01 - 3 x 64 kHz cycles hold time for the SINC3-L8 10 - 4 x 64 kHz cycles hold time for the SINC3-L8 11 - 5 x 64 kHz cycles hold time for the SINC3-L8
3-1 IIRC[2:0]	IIR Low Pass Filter Coefficient (α) 000 - 1/8 001 - 1/16 010 - 1/32 011 - 1/64 100 - 1/128 101 - IIR disabled 110 - IIR disabled 111 - IIR disabled
0 PGAF	PGA fast mode enable 0 - PGA capacitor swap disabled (slow mode). 1 - PGA capacitors swapped during chopper

NOTE

During Low Power mode: 0x15; (00010101b) is recommend for ACQ_CVCR (DBT =01, IIRC = 010, PGAF = 1)

4.8.6.3.19 Low power current threshold (ACQ_CTH)

Table 144. Low power current threshold (ACQ_CTH

Offset[143] 0x76 Access: User read / write

	7	6	5	4	3	2	1	0
R	CTH[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes

[143] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 145. Low power current threshold (ACQ_CTH - register field descriptions

Field	Description
7-0 CTH[7:0]	Low power current threshold See Section 4.3.4.1.1.1, “Current threshold wake-up" for details.

4.8.6.3.20 Low power Ah counter threshold (ACQ_AHTH1 (hi) / ACQ_AHTH1 (lo) / ACQ_AHTH0 (hi) / ACQ_AHTH0 (lo))

Table 146. Low power Ah counter threshold (ACQ_AHTH1 (hi) / ACQ_AHTH1 (lo) / ACQ_AHTH0 (hi) / ACQ_AHTH0 (lo))

Offset[144] 0x78 / 0x79 / 0x7A / 0x7B Access: User read / write

	7	6	5	4	3	2	1	0
R	0	AHTH[30:0]						
W								
R								
W								
R								
W								
R								
W								
Reset	0	0	0	0	0	0	0	0

Notes

144. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 147. Low power Ah counter threshold (ACQ_AHTH1 (hi) / ACQ_AHTH1 (lo) / ACQ_AHTH0 (hi) / ACQ_AHTH0 (lo)) - register field descriptions

Field	Description
30-0 AHTH[30:0]	Low power Ah counter threshold. Absolute (unsigned) 31-Bit integer. Reading one 16-Bit part of the register will buffer the second. Reading the second will unlock the buffer. See Section 4.3.4.1.1.2, “Current ampere hour threshold wake-up". for details on the Register.

4.8.6.3.21 Low power Ah counter (ACQ_AHC1 (hi) / ACQ_AHC1 (lo) / ACQ_AHC0 (hi) / ACQ_AHC0 (lo))

Table 148. Low power Ah counter (ACQ_AHC1 (hi) / ACQ_AHC1 (lo) / ACQ_AHC0 (hi) / ACQ_AHC0 (lo))

Offset[145] 0x7C / 0x7D / 0x7E / 0x7F — Access: User read

	7	6	5	4	3	2	1	0
R	AHC[31:0]							
W								
R	AHC[23:16]							
W								
R	AHC[15:8]							
W								
R	AHC[7:0]							
W								

Notes

[145] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 149. Low power Ah counter (ACQ_AHC1 (hi) / ACQ_AHC1 (lo) / ACQ_AHC0 (hi) / ACQ_AHC0 (lo)) - register field descriptions

Field	Description
31-0 AHC[31:0]	Low power Ah counter (32-Bit signed integer, two's complement). Reading one 16-Bit part of the register will buffer the second. Reading the second will unlock the buffer. See Section 4.3.4.1.1.2, "Current ampere hour threshold wake-up".

4.8.6.3.22 Low pass filter coefficient Ax (LPF_Ax (hi))

Table 150. Low pass filter coefficient Ax (LPF_Ax (hi))

Offset[146] 0x80...0x9E — Access: User read/write

	7	6	5	4	3	2	1	0
R	Ax[15:8]							
W								
Reset	see Table 153							

Notes

[146] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.8.6.3.23 Low pass filter coefficient Ax (LPF_Ax (lo))

Table 151. Low pass filter coefficient Ax (LPF_Ax (lo))

Offset[147] 0x81...0x9F — Access: User read/write

	7	6	5	4	3	2	1	0
R	Ax[7:0]							
W								
Reset	see Table 153							

Notes

[147] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 152. Low pass filter coefficient Ax - register field descriptions

Field	Description
15-0 Ax[15:0]	Low Pass Filter Coefficient Value. x = 0...15. Data Format: MSB = Sign (“1” minus). [14:0] integer.

Table 153. Low pass filter coefficient Ax - reset values

Field	Reset Value	Field	Reset Value
A0	0x00F5	A8	0x1021
A1	0x0312	A9	0x0E35
A2	0x051F	A10	0x0B44
A3	0x0852	A11	0x0852
A4	0x0B44	A12	0x051F
A5	0x0E35	A13	0x0312
A6	0x1021	A14	0x00F5
A7	0x10E5	A15	0x0000

4.8.6.3.24 Compensation control register (COMP_CTL)

Table 154. Compensation control register (COMP_CTL)

Offset[148][149] 0xA0

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	BGCALM		PGAZM	PGAOM	DIAGVM	DIAGIM		CALIEM
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	BGCAL		PGAZ	PGAO	DIAGV	DIAGI	0	CALIE
W								
Reset	1	0	0	0	0	0	0	0

Notes

148. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
149. This Register is 16-Bit access only.

Table 155. Compensation control register (COMP_CTL) - register field descriptions

Field	Description
15-14 BGCALM	Calibration Band Gap Select - Mask 0 - writing the corresponding BGCAL bits will have no effect 1 - writing the corresponding BGCAL bits will be effective
13 PGAZM	PGA Input Zero - Mask 0 - writing the PGAZ bit will have no effect 1 - writing the PGAZ bit will be effective
12 PGAOM	PGA Offset Calibration - Mask 0 - writing the PGAO bit will have no effect 1 - writing the PGAO bit will be effective

Table 155. Compensation control register (COMP_CTL) - register field descriptions (continued)

Field	Description
11 DIAGVM	Diagnostic Mode Voltage Channel - Mask 0 - writing the DIAGV bit will have no effect 1 - writing the DIAGV bit will be effective
10 DIAGIM	Diagnostic Mode Current Channel - Mask 0 - writing the DIAGI bit will have no effect 1 - writing the DIAGI bit will be effective
8 CALIEM	Calibration IRQ Enable - Mask 0 - writing the CALIE bit will have no effect 1 - writing the CALIE bit will be effective
7-6 BGCAL	Calibration Band Gap Select 00 - Bandgap disconnected from calibration 01 - BG1 selected as calibration reference 10 - BG2 selected as calibration reference (default) 11 - BG3 selected as calibration reference
5 PGAZ	PGA Input Zero 0 - Programmable gain amplifier inputs in normal operation 1 - Programmable gain amplifier inputs shorted for Calibration
4 PGAO	PGA Offset Calibration Start 0 - PGA normal operation 1 - PGA internal offset calibration start (PGAOF will indicate calibration complete). PGAZ has to be set to 1 during calibration. The bit will remain set after the calibration is complete. It has to be cleared by writing 0 before it can be set to start the next calibration. The current measurement channel has to be enabled (ACQ_CTL[CMEN]=1) in order to perform the PGA offset compensation.
3 DIAGV	Diagnostic Mode Voltage Channel 0 - Calibration reference disconnected from the voltage channel input 1 - Calibration reference connected to the voltage channel input for calibration. Manual conversion needed to measure reference
2 DIAGI	Diagnostic Mode Current Channel 0 - Calibration reference disconnected from the current channel input 1 - Calibration reference connected to the current channel input for calibration. Manual conversion needed to measure reference
0 CALIE	Calibration IRQ Enable 0 - Calibration request interrupt disabled 1 - Calibration request interrupt enabled. A temperature “out of calibration range” will cause a calibration interrupt request

4.8.6.3.25 Compensation status register (COMP_SR)

Table 156. Compensation status register (COMP_SR)

Offset[150] 0xA2 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	BGRF	0	PGAOF	0	0	0	CALF
W	Write 1 will clear the flags and will start next calibration steps							
Reset	0	0	0	0	0	0	0	0

Notes

[150] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 157. Compensation status register (COMP_SR) - register field descriptions

Field	Description
6 BGRF	Band Gap Reference Status Flag 0 - Indicates the reference bandgap has not been set / applied 1 - Reference bandgap has been set. Writing 1 will clear the flag
4 PGAOF	PGA Internal Offset Compensation Complete Flag 0 - PGA offset compensation ongoing or not started since last flag clear 1 - PGA offset compensation finished since last flag clear. Writing 1 will clear the flag
0 CALF	Calibration Request Status Flag 0 - No Temperature out of range condition detected 1 - Temperature out of range condition detected. Writing 1 will clear the flag

4.8.6.3.26 Temperature filtering period (COMP_TF)

Table 158. Temperature filtering period (COMP_TF)

Offset[151] 0xA3

Access: User read / write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TMF[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

Notes

[151] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 159. Temperature filtering period (COMP_TF) - register field descriptions

Field	Description
2-0 TMF[2:0]	Recalibration Temperature Filtering period. Defines the number of measurements above / below the Max / Min thresholds that are required before a calibration request is detected.

4.8.6.3.27 Max. temp. before recalibration (COMP_TMAX)

Table 160. Max. temp. before recalibration (COMP_TMAX)

Offset[152][153] 0xA4

Access: User read/write

	15	14	13	12	11	10	9	8
R	TCMAX[15:8]							
W								
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	TCMAX[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes

[152] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[153] This Register is 16 Bit access only.

Table 161. Max. temp. before recalibration (COMP_TMAX) - register field descriptions

Field	Description
15-0 TCMAX[15:0]	Maximum Temperature before recalibration. Once the internal temperature measurement result is above or equal to TCMAX, the TMF filter counter is increased, if below, the counter is decreased.

4.8.6.3.28 Min.temp. before recalibration (COMP_TMIN)

Table 162. Min. temp. before recalibration (COMP_TMIN)

Offset [154][155] 0xA6

Access: User read/write

	15	14	13	12	11	10	9	8
R W	TCMIN[15:8]							
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	TCMIN[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes

[154] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[155] This Register is 16-Bit access only.

Table 163. Min. Temp. before recalibration (COMP_TMIN) - register field descriptions

Field	Description
15-0 TCMIN[15:0]	Minimum Temperature before recalibration. Once the internal temperature measurement result is below TCMIN, the TMF filter counter is increased, if above or equal, the counter is decreased.

4.8.6.3.29 Offset voltage compensation (COMP_VO)

Table 164. Offset voltage compensation (COMP_VO)

Offset[156] 0xAA

Access: User read/write

	7	6	5	4	3	2	1	0
R W	VOC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes

[156] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 165. Offset voltage compensation (COMP_VO) - register field descriptions

Field	Description
7-0 VOC[7:0]	Voltage Offset Compensation Buffer. This register contains the voltage channel offset compensation as an 8-bit signed char (two complement). 0x7F = max, 0x80 =min.

4.8.6.3.30 Offset current compensation window (COMP_IO)

Table 166. Offset current compensation window (COMP_IO)

Offset[157] 0xAB

Access: User read/write

	7	6	5	4	3	2	1	0
R	COC[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes

[157] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 167. Offset current compensation window (COMP_IO) - register field descriptions

Field	Description
7-0 COC[7:0]	Current Offset Compensation Buffer window for the 8 current compensation values stored. The content of the IGAIN[2:0] register will determine the compensation buffer accessed through the COC[7:0] register. This register contains the current channel offset compensation as 8-bit signed char (two complement). 0x7F = max, 0x80 =min.

4.8.6.3.31 Gain voltage comp. V_{SENSE} channel (COMP_VSG)

Table 168. Gain voltage comp. V_{SENSE} channel (COMP_VSG)

Offset [158][159] 0xAC

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	VSGC[9:8]	
W								
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	VSGC[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes

[158] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[159] This Register is 16 Bit access only.

Table 169. Gain voltage comp. V_{SENSE} channel (COMP_VSG) - register field descriptions

Field	Description
9-0 VSGC[9:0]	Voltage Channel Gain Compensation Buffer. This register contains the voltage channel gain compensation as 10-bit special coded value. Refer to Section 4.8.3.4, “Compensation" for details.

4.8.6.3.32 8 x gain current compensation 4...512 (COMP_IG4... COMP_IG512)

Table 170. 8 x Gain current compensation 4...512 (COMP_IG4... COMP_IG512)

Offset (160)(161) 0xB0... 0xBE

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	IGC4...512 (hi) [9:8]	
W								
Reset	0	0	0	0	0	0	1	0
	7	6	5	4	3	2	1	0
R	IGC4...512 (lo) [7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes

[160] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[161] This Register is 16 Bit access only.

Table 171. 8 x gain current compensation 4...512 (COMP_IG4… COMP_IG512) - register field descriptions

Field	Description
9-0 IGC4[9:0] IGC8[9:0] IGC16[9:0] IGC32[9:0] IGC64[9:0] IGC128[9:0] IGC256[9:0] IGC512[9:0]	Individual Current Gain Compensation Buffers for the 8 Gain configurations. Those registers contain the current channel gain compensation as 10-bit special coded value. Refer to Section 4.8.3.4, “Compensation" for details.

4.8.6.3.33 8 x offset PGA compensation (COMP_PGAO4...COMP_PGAO512)

Table 172. 8 x offset PGA compensation (COMP_PGAO4... COMP_PGAO512)

Offset (162)(163) 0xC0... 0xCE

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	PGAOC4...512 (hi) [10:8]		
W								
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	PGAOC4...512 (lo) [7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes

[162] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[163] This Register is 16 Bit access only.

Table 173. 8 x offset PGA compensation (COMP_PGAO4...COMP_PGAO512) - register field descriptions

Field	Description
10-0 PGAOC4[10:0] PGAOC8[10:0] PGAOC16[10:0] PGAOC32[10:0] PGAOC64[10:0] PGAOC128[10:0] PGAOC256[10:0] PGAOC512[10:0]	Individual PGA Offset Compensation Buffers for the 8 Gain configurations. Those registers contain the PGA Offset compensation as 11-bit special coded value. Refer to Section 4.8.3.4, “Compensation" for details.

4.8.6.3.34 Internal temp. offset compensation (COMP_ITO)

Table 174. Internal temp. offset compensation (COMP_ITO)

Offset[164] 0xD0 Access: User read/write

	7	6	5	4	3	2	1	0
R	ITOC[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes

[164] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 175. Internal temp. offset compensation (COMP_ITO) - register field descriptions

Field	Description
7-0 ITOC[7:0]	Internal Temperature Offset Compensation Buffer. This register contains the Internal Temperature Offset compensation as 8-bit signed char (two complement). Refer to Section 4.8.3.4, “Compensation" for details.

4.8.6.3.35 Internal temp. gain compensation (COMP_ITG)

Table 176. Internal temp. gain compensation (COMP_ITG)

Offset[165] 0xD1 Access: User read/write

	7	6	5	4	3	2	1	0
R	ITGC[7:0]							
W								
Reset	1	0	0	0	0	0	0	0

Notes

[165] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 177. Internal temp. gain compensation (COMP_ITG) - register field descriptions

Field	Description
7-0 ITGC[7:0]	Internal Temperature Gain Compensation Buffer. This register contains the Internal Temperature Gain compensation as 8-bit special coded value. Refer to Section 4.8.3.4, “Compensation" for details.

4.8.6.3.36 External temp. offset compensation (COMP_ETO)

Table 178. External temp. offset compensation (COMP_ETO)

Offset[166] 0xD2 Access: User read/write

	7	6	5	4	3	2	1	0
R W	ETOC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes

[166] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 179. External temp. offset compensation (COMP_ETO) - register field descriptions

Field	Description
7-0 ETOC[7:0]	External Temperature Offset Compensation Buffer. This register contains the External Temperature Offset compensation as 8-bit signed char (two complement). Refer to Section 4.8.3.4, “Compensation" for details.

4.8.6.3.37 External temp. gain compensation (COMP_ETG)

Table 180. External temp. gain compensation (COMP_ETG)

Offset[167] 0xD3 Access: User read/write

	7	6	5	4	3	2	1	0
R W	ETGC[7:0]							
Reset	1	0	0	0	0	0	0	0

Notes

[167] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 181. External temp. gain compensation (COMP_ETG) - register field descriptions

Field	Description
7-0 ETGC[7:0]	External Temperature Gain Compensation Buffer. This register contains the External Temperature Gain compensation as 8-bit special coded value. Refer to Section 4.8.3.4, “Compensation" for details.

4.9 Window watchdog

The MM912_637 analog die includes a configurable window watchdog which is active in Normal mode. The watchdog module is based on the Low Power Oscillator (LPCLK) to operate independently from the MCU based D2DCLK clock. The watchdog timeout (t_{WDTO}) can be configured between 4.0 ms and 2048 ms using the watchdog control register (WD_CTL).

NOTE

As the watchdog timing is based on the LPCLK, its accuracy is based on the trimming applied to the TRIM_OSC register. The given timeout values are typical values only.

During Low Power mode, the watchdog feature is not active, a D2D read during Stop mode will have the WDOFF bit set. After wake-up and transition to Normal mode, the watchdog is reset to the same state as when following a Power-On-Reset (POR).

To clear the watchdog counter, a alternating write has to be performed to the watchdog rearm register (WD_RR). The first write after the wake-up or $\overline{\text{RESET\_A}}$ has been released has to be 0xAA, the next one has to be 0x55.

After the wake-up or $\overline{\text{RESET\_A}}$ has been released, there will be a standard (non window) watchdog active with a fixed timeout of t_{IWDTO} (t_{WDTO} = b100 = 256 ms). The Watchdog Window Open (WDWO) bit is set during that time.

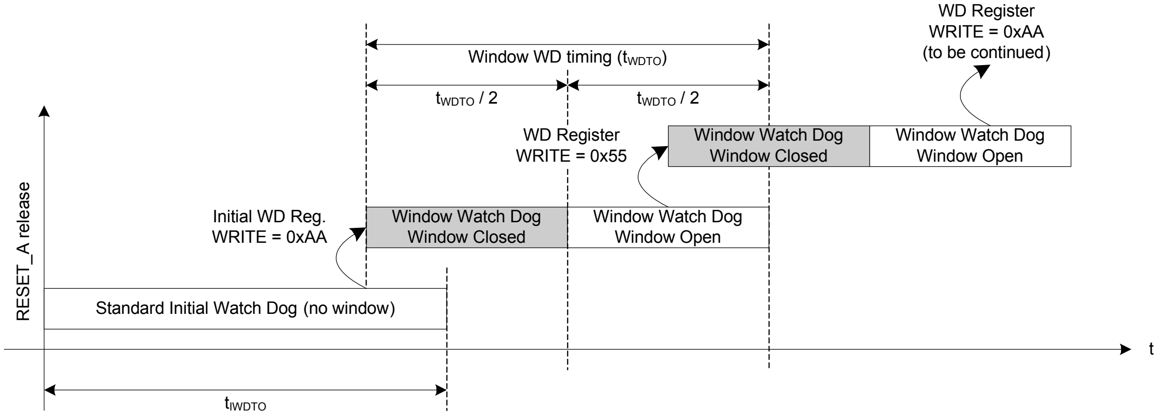

WD Register
WRITE = 0xAA
(to be continued)
Window WD timing (t_{WDTO})
t_{WDTO} / 2
t_{WDTO} / 2
WD Register
WRITE = 0x55
Window Watch Dog Window Closed
Window Watch Dog Window Open
Initial WD Reg.
WRITE = 0xAA
Window Watch Dog Window Closed
Window Watch Dog Window Open
RESET_A release
Standard Initial Watch Dog (no window)
t
t_{IWDTO}

Figure 39. MM912_637 Analog die watchdog operation

To change from the standard initial watchdog to the window watchdog, the initial counter reset has to be performed by writing 0xAA to the Watchdog rearm register (WD_RR) before t_{IWDTO} is reached.

NOTE

An immediate trimming of the low power oscillator after reset release assures t_{IWDTO} being at the maximum accuracy. See chapter 5.2.2.5, “Low power oscillator trimming (TRIM_OSC)".

If the t_{IWDTO} timeout is reached with no counter reset or a value different from 0xAA written to the WD_RR, a watchdog reset will occur.

Once entering window watchdog mode, the first half of the time, t_{WDTO} is forbidden for a counter reset. To reset the watchdog counter, an alternating write of 0x55 and 0xAA has to be performed within the second half of the t_{WDTO}. A Window Open (WDWO) flag will indicate the current status of the window. A timeout or wrong value written to the WD_RR will force a watchdog reset.

If the first write to the WD_CTL register is 000 (WD OFF), the WD will be disabled[168]. If a different cycle time is written or the WD is refreshed with the default Window (100) unchanged, no further “000” write will be effective (a change of cycle time would still be possible).

Notes

[168] The Watchdog can be enabled any time later.

4.9.1 Memory map and registers

4.9.1.1 Overview

This section provides a detailed description of the memory map and registers.

4.9.1.2 Module memory map

The memory map for the Watchdog module is given in Table 62

Table 182. Module memory map

Offset [169],[170]	Name		7	6	5	4	3	2	1	0
0x10	WD_CTL Watchdog control register	R	0	0	0	0	0	0	0	0
		W	Reserved					WDTOM[2:0]		
		R	Reserved	0	0	0	0	WDTO[2:0]		
		W								
0x12	WD_SR	R	0	0	0	0	0	0	WDOFF	WDWO
	Watchdog status register	W								

Table 182. Module memory map (continued)

Offset [169],[170]	Name		7	6	5	4	3	2	1	0
0x13	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x14	WD_RR	R	WDR[7:0]							
	Watchdog rearm register	W								
0x15	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x16	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x17	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes

[169] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[170] Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

4.9.1.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bits and field function follow the register diagrams, in bit order.

4.9.1.3.1 Watchdog control register (WD_CTL)

Table 183. Watchdog control register (WD_CTL)

Offset [171],[172] 0x10

Access: User write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	Reserved					WDTOM		
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	Reserved	0	0	0	0	WDTO		
W								
Reset	1	0	0	0	0	1	0	0

Notes

[171] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[172] This Register is 16 Bit access only.

Table 184. Watchdog control register (WD_CTL) - register field descriptions

Field	Description
15 Reserved	This reserved bit is used for test purpose. Software has to write a 0 into this bit.
10-8 WDTOM[2:0]	Watchdog Timeout - Mask any setting other than 111 - writing the WDTO bits will have no effect 111 - writing the WDTO bits will be effective

Table 184. Watchdog control register (WD_CTL) - register field descriptions (continued)

Field	Description
7 Reserved	This reserved bit is used for test purpose. Software has to write a 0 into this bit.
2-0 WDTO[2:0]	Watchdog Timeout Configuration - configuring the watchdog timeout duration t_{WDTO}. 000 - Watchdog OFF 001 - 4.0 ms 010 - 16.0 ms 011 - 64.0 ms 100 - 256 ms (default) 101 - 512 ms 110 - 1024 ms 111 - 2048 ms

4.9.1.3.2 Watchdog status register (WD_SR)

Table 185. Watchdog status register (WD_SR)

Offset[173] 0x12 Access: User read

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	WDOFF	WDWO
W								

Notes

[173] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 186. Watchdog status register (WD_SR) - register field descriptions

Field	Description
1 WDOFF	Watchdog Status - Indicating the watchdog module being enabled/disabled 1 - Watchdog Off 0 - Watchdog Active
0 WDWO	Watchdog Window Status 1 - Open - Indicating the watchdog window is currently open for counter reset. 0 - Closed - Indicating the watchdog window is currently closed for counter reset. Resetting the watchdog with the window closed will cause a watchdog - reset.

4.9.1.3.3 Watchdog rearm register (WD_RR)

Table 187. Watchdog rearm register (WD_RR)

Offset[174] 0x14 Access: User read/write

	7	6	5	4	3	2	1	0
R W	WDR							
Reset	0	0	0	0	0	0	0	0

Notes

[174] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 188. Watchdog rearm register (WD_RR) - register field descriptions

Field	Description
7-0 WDR[7:0]	Watchdog rearm register- Writing this register with the correct value (0xAA alternating 0x55) while the window is open will reset the watchdog counter. Writing the register while the watchdog is disabled will have no effect.

4.10 Basic timer module - TIM (TIM16B4C)

4.10.1 Introduction

4.10.1.1 Overview

The basic timer consists of a 16-bit, software-programmable counter driven by a seven stage programmable prescaler. This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from microseconds to many seconds.

This timer contains four complete input capture/output compare channels [IOC 3:0]. The input capture function is used to detect a selected transition edge and record the time. The output compare function is used for generating output signals or for timer software delays.

Full access for the counter registers or the input capture/output compare registers should take place in a16-bit word access. Accessing high bytes and low bytes separately for all of these registers may not yield the same result as accessing them in one word.

4.10.1.2 Features

The TIM16B4C includes these distinctive features:

- Four input capture/output compare channels.
- Clock prescaler
- 16-bit counter
- Timer counter reset on Output Compare 3

4.10.1.3 Modes of operation

The TIM16B4C is driven by the D2DCLK / 4 during Normal mode and the ALFCLK during Low Power mode.

4.10.1.4 Block diagram

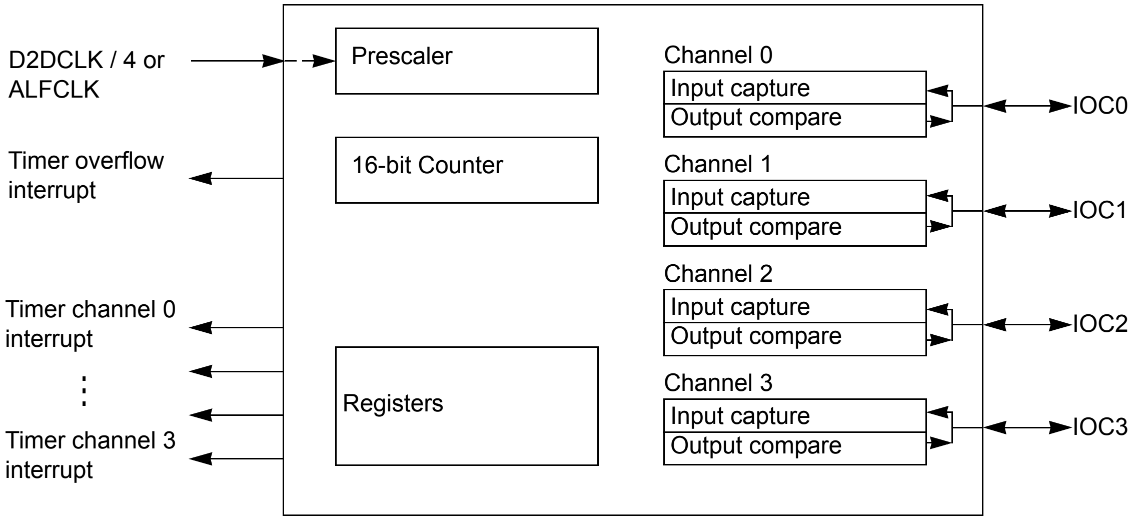


Figure 40. Timer block diagram

For more information on the respective functional descriptions see Section 4.10.4, “Functional description" of this chapter.

4.10.2 Signal description

4.10.2.1 Overview

The TIM16B4C module can be used as regular time base, or can be internally routed to the PTB and LIN module. Refer to the corresponding sections for further details, see Section 4.12, “LIN" and Section 4.11, “General purpose I/O - GPIO". In addition, the TIM16B4C module is used during Low Power mode to determine the cyclic wake-up and current measurement timing (Section 4.3, “Analog die - power, clock and resets - PCR")

4.10.2.2 Detailed signal descriptions

4.10.2.2.1 IOC3 – input capture and output compare channel 3

This pin serves as the input capture or output compare for channel 3.

4.10.2.2.2 IOC2 – input capture and output compare channel 2

This pin serves as the input capture or output compare for channel 2.

4.10.2.2.3 IOC1 – input capture and output compare channel 1

This pin serves as the input capture or output compare for channel 1.

4.10.2.2.4 IOC0 – input capture and output compare channel 0

This pin serves as the input capture or output compare for channel 0.

4.10.3 Memory map and registers

4.10.3.1 Overview

This section provides a detailed description of all memory and registers.

4.10.3.2 Module memory map

The memory map for the TIM16B4C module is given in Table 62.

Table 189. Module memory map

Offset (175)	Name		7	6	5	4	3	2	1	0
0x20	TIOS	R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
	Timer Input Capture/Output Compare Select	W								
0x21 (176)	CFORC	R	0	0	0	0	0	0	0	0
	Timer Compare Force Register	W					FOC3	FOC2	FOC1	FOC0
0x22	OC3M	R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
	Output Compare 3 Mask Register	W								
0x23	OC3D	R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
	Output Compare 3 Data Register	W								

Table 189. Module memory map (continued)

Offset (175)	Name		7	6	5	4	3	2	1	0
0x24 (177)	TCNT (hi)	R	TCNT							
	Timer Count Register	W								
0x25 (177)	TCNT (lo)	R								
	Timer Count Register	W								
0x26	TSCR1	R	TEN	0	0	TFFCA	0	0	0	0
	Timer System Control Register 1	W								
0x27	TTOV	R	0	0	0	0	TOV3	TOV2	TOV1	TOV0
	Timer Toggle Overflow Register	W								
0x28	TCTL1	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	Timer Control Register 1	W								
0x29	TCTL2	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	Timer Control Register 2	W								
0x2A	TIE	R	0	0	0	0	C3I	C2I	C1I	C0I
	Timer Interrupt Enable Register	W								
0x2B	TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	Timer System Control Register 2	W								
0x2C	TFLG1	R	0	0	0	0	C3F	C2F	C1F	C0F
	Main Timer Interrupt Flag 1	W								
0x2D	TFLG2	R	TOF	0	0	0	0	0	0	0
	Main Timer Interrupt Flag 2	W								
0x2E (178)	TC0 (hi)	R	TC0							
	Timer Input Capture/Output Compare Register 0	W								
0x2F (178)	TC0 (lo)	R								
	Timer Input Capture/Output Compare Register 0	W								
0x30 (178)	TC1 (hi)	R	TC1							
	Timer Input Capture/Output Compare Register 1	W								
0x31 (178)	TC1 (lo)	R								
	Timer Input Capture/Output Compare Register 1	W								
0x32 (178)	TC2 (hi)	R	TC2							
	Timer Input Capture/Output Compare Register 2	W								
0x33 (178)	TC2 (lo)	R								
	Timer Input Capture/Output Compare Register 2	W								

Table 189. Module memory map (continued)

Offset [175]	Name		7	6	5	4	3	2	1	0
0x34 [178]	TC3 (hi)	R	TC3							
	Timer Input Capture/Output Compare Register 3	W								
0x35 [178]	TC3 (lo)	R								
	Timer Input Capture/Output Compare Register 3	W								
0x36 [177]	TIMTST	R	0	0	0	0	0	0	TCBYP	0
	Timer Test Register	W								

Notes

[175] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[176] Always reads $00.

[177] Only writable in special modes. (Refer to the SOC Guide for different modes).

[178] A write to these registers has no meaning or effect during input capture.

4.10.3.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

4.10.3.3.1 Timer input capture/output compare select (TIOS)

Table 190. Timer input capture/output compare select (TIOS)

Offset[179] 0x20

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[179] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 191. TIOS - register field descriptions

Field	Description
3-0 IOS[3-0]	Input Capture or Output Compare Channel Configuration 0 - The corresponding channel acts as an input capture. 1 - The corresponding channel acts as an output compare.

4.10.3.3.2 Timer compare force register (CFORC)

Table 192. Timer compare force register (CFORC)

Offset[180] 0x21 Access: User write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W					FOC3	FOC2	FOC1	FOC0
Reset	0	0	0	0	0	0	0	0

Notes

[180] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 193. CFORC - register field descriptions

Field	Description
3-0 FOC[3-0]	Force Output Compare Action for Channel 3-0 0 - Force output compare action disabled. Input capture or output compare channel configuration 1 - Force output compare action enabled

A write to this register with the corresponding (FOC 3:0) data bit(s) set causes the action programmed for output compare on channel “n” to occur immediately.The action taken is the same as if a successful comparison had just taken place with the TCn register, except the interrupt flag does not get set.

NOTE

A successful channel 3 output compare overrides any channel 2:0 compare. If a forced output compare on any channel occurs at the same time as the successful output compare, then a forced output compare action will take precedence and the interrupt flag will not get set.

4.10.3.3.3 Output compare 3 mask register (OC3M)

Table 194. Output compare 3 mask register (OC3M)

Offset[181] 0x22 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[181] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 195. OC3M - register field descriptions

Field	Description
3-0 OC3M[3-0]	Output Compare 3 Mask “n” Channel bit 0 - Does not set the corresponding port to be an output port 1 - Sets the corresponding port to be an output port when this corresponding TIOS bit is set to be an output compare

Setting the OC3Mn (n ranges from 0 to 2) will set the corresponding port to be an output port when the corresponding TIOSn (n ranges from 0 to 2) bit is set to be an output compare.

NOTE

A successful channel 3 output compare overrides any channel 2:0 compares. For each OC3M bit that is set, the output compare action reflects the corresponding OC3D bit.

4.10.3.3.4 Output compare 3 data register (OC3D)

Table 196. Output compare 3 data register (OC3D)

Offset[182] 0x23

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
W								
Reset	0	0	0	0	0	0	0	0

Notes

182. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 197. OC3D - register field descriptions

Field	Description
3 OC3D3	Output Compare 3 Data for Channel 3
2 OC3D2	Output Compare 3 Data for Channel 2
1 OC3D1	Output Compare 3 Data for Channel 1
0 OC3D0	Output Compare 3 Data for Channel 0

NOTE

A channel 3 output compare will cause bits in the output compare 3 data register to transfer to the timer port data register if the corresponding output compare 3 mask register bits are set.

4.10.3.3.5 Timer count register (TCNT)

Table 198. Timer count register (TCNT)

Offset[183] 0x24, 0x25

Access: User read (anytime)/write (special mode)

	15	14	13	12	11	10	9	8
R	tcnt15	tcnt14	tcnt13	tcnt12	tcnt11	tcnt10	tcnt9	tcnt8
W								
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	tcnt7	tcnt6	tcnt5	tcnt4	tcnt3	tcnt2	tcnt1	tcnt0
W								
Reset	0	0	0	0	0	0	0	0

Notes

183. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 199. TCNT - register field descriptions

Field	Description
15-0 tcnt[15-0]	16-Bit Timer Count Register

NOTE

The 16-bit main timer is an up counter. Full access to the counter register should take place in one clock cycle. A separate read/write for high bytes and low bytes will give a different result than accessing them as a word. The period of the first count after a write to the TCNT registers may be a different length, because the write is not synchronized with the prescaler clock.

4.10.3.3.6 Timer system control register 1 (TSCR1)

Table 200. Timer system control register 1 (TSCR1)

Offset[184] 0x26 Access: User read/write

	7	6	5	4	3	2	1	0
R	TEN	0	0	TFFCA	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[184] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 201. TSCR1 - register field descriptions

Field	Description
7 TEN	Timer Enable 1 = Enables the timer. 0 = Disables the timer. (Used for reducing power consumption).
4 TFFCA	Timer Fast Flag Clear All 1 = For TFLG1 register, a read from an input capture or a write to the output compare channel [TC 3:0] causes the corresponding channel flag, CnF, to be cleared. For TFLG2 register, any access to the TCNT register clears the TOF flag. This has the advantage of eliminating software overhead in a separate clear sequence. Extra care is required to avoid accidental flag clearing due to unintended accesses. 0 = Allows the timer flag clearing.

4.10.3.3.7 Timer toggle on overflow register 1 (TTOV)

Table 202. Timer toggle on overflow register 1 (TTOV)

Offset[185] 0x27 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	TOV3	TOV2	TOV1	TOV0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[185] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 203. TTOV - register field descriptions

Field	Description
3-0 TOV[3-0]	Toggle On Overflow Bits 1 = Toggle output compare pin on overflow feature enabled. 0 = Toggle output compare pin on overflow feature disabled.

NOTE

TOVn toggles the output compare pin on overflow. This feature only takes effect when the corresponding channel is configured for an output compare mode. When set, an overflow toggle on the output compare pin takes precedence over forced output compare events.

4.10.3.3.8 Timer control register 1 (TCTL1)

Table 204. Timer control register 1 (TCTL1)

Offset[186] 0x28 Access: User read/write

	7	6	5	4	3	2	1	0
R W	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
Reset	0	0	0	0	0	0	0	0

Notes

[186] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 205. TCTL1 - register field descriptions

Field	Description
7,5,3,1 OMn	Output Mode bit
6,4,2,0 OLn	Output Level bit

NOTE

These four pairs of control bits are encoded to specify the output action to be taken as a result of a successful Output Compare on “n” channel. When either OMn or OLn, the pin associated with the corresponding channel becomes an output tied to its IOC. To enable output action by the OMn and OLn bits on a timer port, the corresponding bit in OC3M should be cleared.

Table 206. Compare result output action

OMn	OLn	Action
0	0	Timer disconnected from output pin logic
0	1	Toggle OCn output line
1	0	Clear OCn output line to zero
1	1	Set OCn output line to one

4.10.3.3.9 Timer control register 2 (TCTL2)

Table 207. Timer control register 2 (TCTL2)

Offset[187] 0x29 Access: User read/write

	7	6	5	4	3	2	1	0
R W	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
Reset	0	0	0	0	0	0	0	0

Notes

[187] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 208. TCTL2 - register field descriptions

Field	Description
EDGnB,EDGnA	Input Capture Edge Control

These four pairs of control bits configure the input capture edge detector circuits.

Table 209. Edge detector circuit configuration

EDGnB	EDGnA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

4.10.3.3.10 Timer interrupt enable register (TIE)

Table 210. Timer interrupt enable register (TIE)

Offset[188] 0x2A Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	C3I	C2I	C1I	C0I
W								
Reset	0	0	0	0	0	0	0	0

Notes

[188] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 211. TIE - register field descriptions

Field	Description
3-0 C[3-0]I	Input Capture/Output Compare Interrupt Enable. 1 = Enables corresponding Interrupt flag (CnF of TFLG1 register) to cause a hardware interrupt 0 = Disables corresponding Interrupt flag (CnF of TFLG1 register) from causing a hardware interrupt

4.10.3.3.11 Timer system control register 2 (TSCR2)

Table 212. Timer system control register 2 (TSCR2)

Offset[189] 0x2B Access: User read/write

	7	6	5	4	3	2	1	0
R	TOI	0	0	0	TCRE	PR2	PR1	PR0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[189] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 213. TIE - register field descriptions

Field	Description
7 TOI	Timer Overflow Interrupt Enable 1 = Hardware interrupt requested when TOF flag set in TFLG2 register. 0 = Hardware Interrupt request inhibited.
3 TCRE	TCRE — Timer Counter Reset Enable 1 = Enables timer counter reset by a successful output compare on channel 3 0 = Inhibits timer counter reset and counter continues to run.
3-0 PR[2:0]	Timer Prescaler Select These three bits select the frequency of the timer prescaler clock derived from the bus clock as shown in Table 214.

NOTE

This mode of operation is similar to an up-counting modulus counter.

If register TC3 = $0000 and TCRE = 1, the timer counter register (TCNT) will stay at $0000 continuously. If register TC3 = $FFFF and TCRE = 1, TOF will not be set when the timer counter register (TCNT) is reset from $FFFF to $0000.

The newly selected prescale factor will not take effect until the next synchronized edge, where all prescale counter stages equal zero.

Table 214. Timer clock selection

PR2	PR1	PR0	Timer Clock[190]
0	0	0	TimerClk / 1
0	0	1	TimerClk / 2
0	1	0	TimerClk / 4
0	1	1	TimerClk / 8
1	0	0	TimerClk / 16
1	0	1	TimerClk / 32
1	1	0	TimerClk / 64
1	1	1	TimerClk / 128

Notes

[190] TimerClk = D2DCLK/4 or ALFCLK

4.10.3.3.12 Main timer interrupt flag 1 (TFLG1)

Table 215. Main timer interrupt flag 1 (TFLG1)

Offset[191] 0x2C

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	C3F	C2F	C1F	C0F
W								
Reset	0	0	0	0	0	0	0	0

Notes

[191] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 216. TFLG1 - register field descriptions

Field	Description
3-0 C[3:0]F	Input Capture/Output Compare Channel Flag. 1 = Input capture or output compare event occurred 0 = No event (input capture or output compare event) occurred.

NOTE

These flags are set when an input capture or output compare event occurs. Flag set on a particular channel is cleared by writing a one to that corresponding CnF bit. Writing a zero to CnF bit has no effect on its status. When TFFCA bit in TSCR register is set, a read from an input capture or a write into an output compare channel will cause the corresponding channel flag CnF to be cleared.

4.10.3.3.13 Main timer interrupt flag 2 (TFLG2)

Table 217. Main timer interrupt flag 2 (TFLG2)

Offset[192] 0x2D

Access: User read/write

	7	6	5	4	3	2	1	0
R	TOF	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[192] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 218. TFLG2 - register field descriptions

Field	Description
7 TOF	Timer Overflow Flag 1 = Indicates that an interrupt has occurred (Set when 16-bit free-running timer counter overflows from $FFFF to $0000) 0 = Flag indicates an interrupt has not occurred.

NOTE

The TFLG2 register indicates when an interrupt has occurred. Writing a one to the TOF bit will clear it. Any access to TCNT will clear TOF bit of TFLG2 register if the TFFCA bit in TSCR register is set.

4.10.3.3.14 Timer input capture/output compare registers (TC3 - TC0)

Table 219. Timer input capture/output compare register 0 (TC0)

Offset[193] 0x2E, 0x2F

Access: User read/write

	15	14	13	12	11	10	9	8
R W	tc0_15	tc0_14	tc0_13	tc0_12	tc0_11	tc0_10	tc0_9	tc0_8
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	tc0_7	tc0_6	tc0_5	tc0_4	tc0_3	tc0_2	tc0_1	tc0_0
Reset	0	0	0	0	0	0	0	0

Notes

[193] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 220. Timer input capture/output compare register 1(TC1)

Offset[194] 0x30, 0x31

Access: User read/write

	15	14	13	12	11	10	9	8
R W	tc1_15	tc1_14	tc1_13	tc1_12	tc1_11	tc1_10	tc1_9	tc1_8
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	tc1_7	tc1_6	tc1_5	tc1_4	tc1_3	tc1_2	tc1_1	tc1_0
Reset	0	0	0	0	0	0	0	0

Notes

[194] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 221. Timer input capture/output compare register 2(TC2)

Offset[195] 0x32, 0x33

Access: User read/write

	15	14	13	12	11	10	9	8
R W	tc2_15	tc2_14	tc2_13	tc2_12	tc2_11	tc2_10	tc2_9	tc2_8
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	tc2_7	tc2_6	tc2_5	tc2_4	tc2_3	tc2_2	tc2_1	tc2_0
Reset	0	0	0	0	0	0	0	0

Notes

[195] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 222. Timer input capture/output compare register 3(TC3)

Offset[196] 0x34, 0x35 Access: User read/write

	15	14	13	12	11	10	9	8
R W	tc3_15	tc3_14	tc3_13	tc3_12	tc3_11	tc3_10	tc3_9	tc3_8
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	tc3_7	tc3_6	tc3_5	tc3_4	tc3_3	tc3_2	tc3_1	tc3_0
Reset	0	0	0	0	0	0	0	0

Notes

[196] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 223. TCn - register field descriptions

Field	Description
15-0 tcn[15-0]	16 Timer Input Capture/Output Compare Registers

NOTE

TRead anytime. Write anytime for output compare function. Writes to these registers have no effect during input capture.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read/Write access in byte mode for high byte should takes place before low byte otherwise it will give a different result.

4.10.4 Functional description

4.10.4.1 general

This section provides a complete functional description of the timer TIM16B4C block. Refer to the detailed timer block diagram in Figure 41 as necessary.

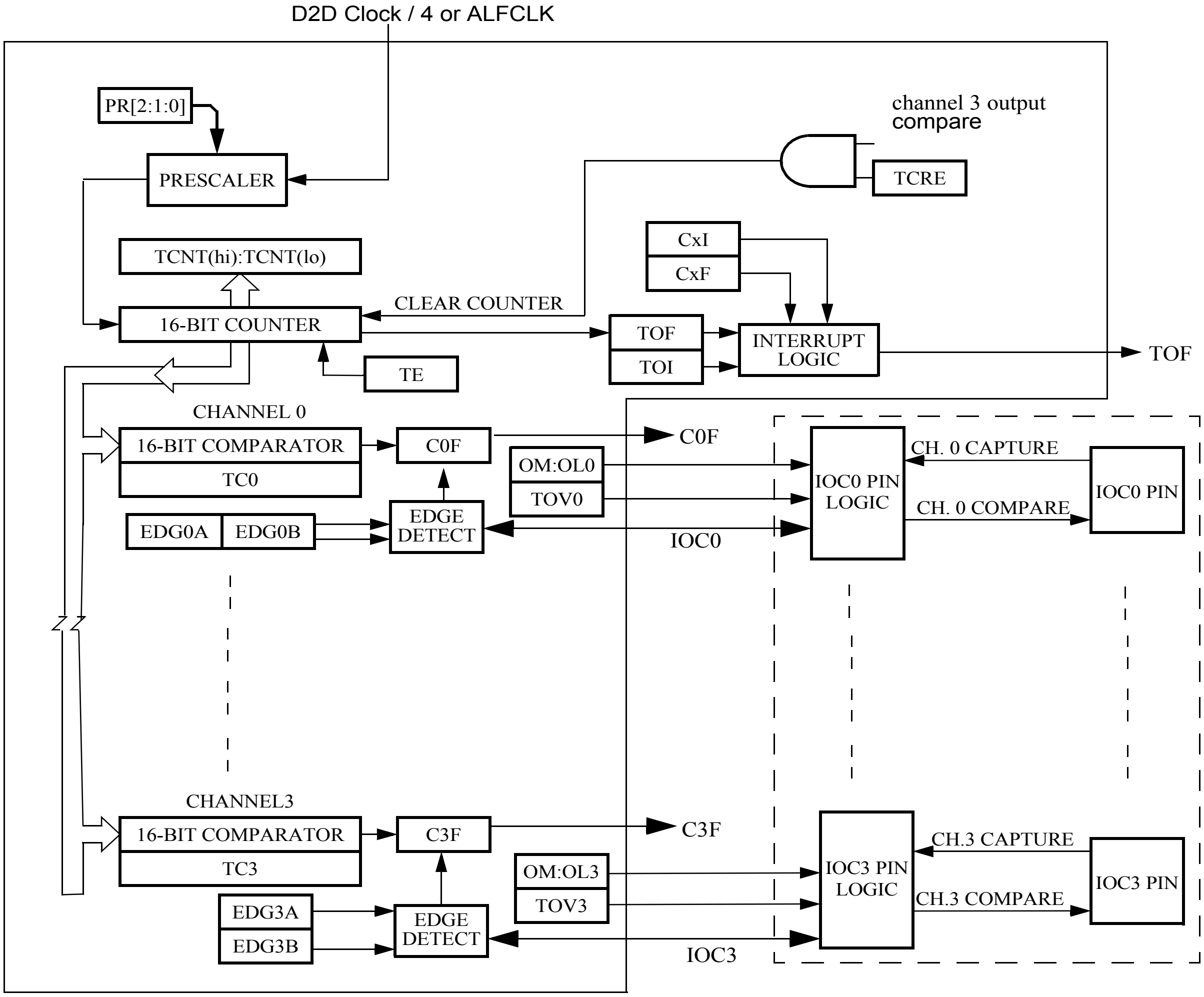


Figure 41. Detailed timer block diagram

4.10.4.2 Prescaler

The prescaler divides the bus clock by 1, 2, 4, 8,16, 32, 64, or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in the timer system control register 2 (TSCR2).

4.10.4.3 Input capture

Clearing the I/O (input/output) select bit, IOSn, configures channel n as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCn.

The minimum pulse width for the input capture input is greater than two bus clocks. An input capture on channel n sets the CnF flag. The CnI bit enables the CnF flag to generate interrupt requests.

4.10.4.4 Output compare

Setting the I/O select bit, IOSn, configures channel n as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin. An output compare on channel n sets the CnF flag. The CnI bit enables the CnF flag to generate interrupt requests.

The output mode and level bits, OMn and OLn, select set, clear, toggle on output compare. Clearing both OMn and OLn disconnects the pin from the output logic. Setting a force output compare bit, FOCn, causes an output compare on channel n. A forced output compare does not set the channel flag.

A successful output compare on channel 3 overrides output compares on all other output compare channels. The output compare 3 mask register masks the bits in the output compare 3 data register. The timer counter reset enable bit, TCRE, enables channel 3 output compares to reset the timer counter.Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general-purpose output, the last value written to the bit appears at the pin.

4.10.5 Resets

4.10.5.1 General

The reset state of each individual bit is listed within the Register Description Section 4.10.3, “Memory map and registers", which details the registers and their bit-fields.

4.10.6 Interrupts

4.10.6.1 General

This section describes interrupts originated by the TIM16B4C block. Table 224 lists the interrupts generated by the TIM16B4C to communicate with the MCU.

Table 224. TIM16B4C interrupts

Interrupt	Offset	Vector	Priority	Source	Description
C[3:0]F	-	-	-	Timer Channel 3-0	Active high timer channel interrupts 3-0
TOF	-	-	-	Timer Overflow	Timer Overflow interrupt

4.10.6.2 Description of interrupt operation

The TIM16B4C uses a total of 5 interrupt vectors. The interrupt vector offsets and interrupt numbers are chip dependent. More information on interrupt vector offsets and interrupt numbers can be found in Section 4.4, “Interrupt module - IRQ".

Channel [3:0] Interrupt

These active high outputs is asserted by the module to request a timer channel 3 – 0 interrupt following an input capture or output compare event on these channels [3-0]. For the interrupt to be asserted on a specific channel, the enable, CnI bit of TIE register should be set. These interrupts are serviced by the system controller.

4.10.6.2.1 Timer overflow interrupt (TOF)

This active high output will be asserted by the module to request a timer overflow interrupt, following the timer counter overflow when the overflow enable bit (TOI) bit of TFLG2 register is set. This interrupt is serviced by the system controller.

4.11 General purpose I/O - GPIO

4.11.1 Introduction

The 3 General Purpose I/Os (PTB0...2) are multipurpose ports, making internal signals available externally and providing digital inputs. L0 (PTB3) offers an additional wake-up on rising edge during low power mode.

Additional routing options allow connections to the LIN, TIMER, and SCI module.

4.11.2 Features

- Internal Clamping Structure to operate as High Voltage Input (PTB3/L0 only).
- 5.0 V (VDDX) digital port Input/Output (PTB3/L0 only as Input)
- Selectable internal pull-up (PTB3/L0 pull-down) resistor
- Selectable Wake-up Input during Low Power mode (PTB3/L0 - rising edge only).
- Selectable Timer Channel Input / Output
- Selectable connection to LIN / SCI

4.11.3 Block diagram

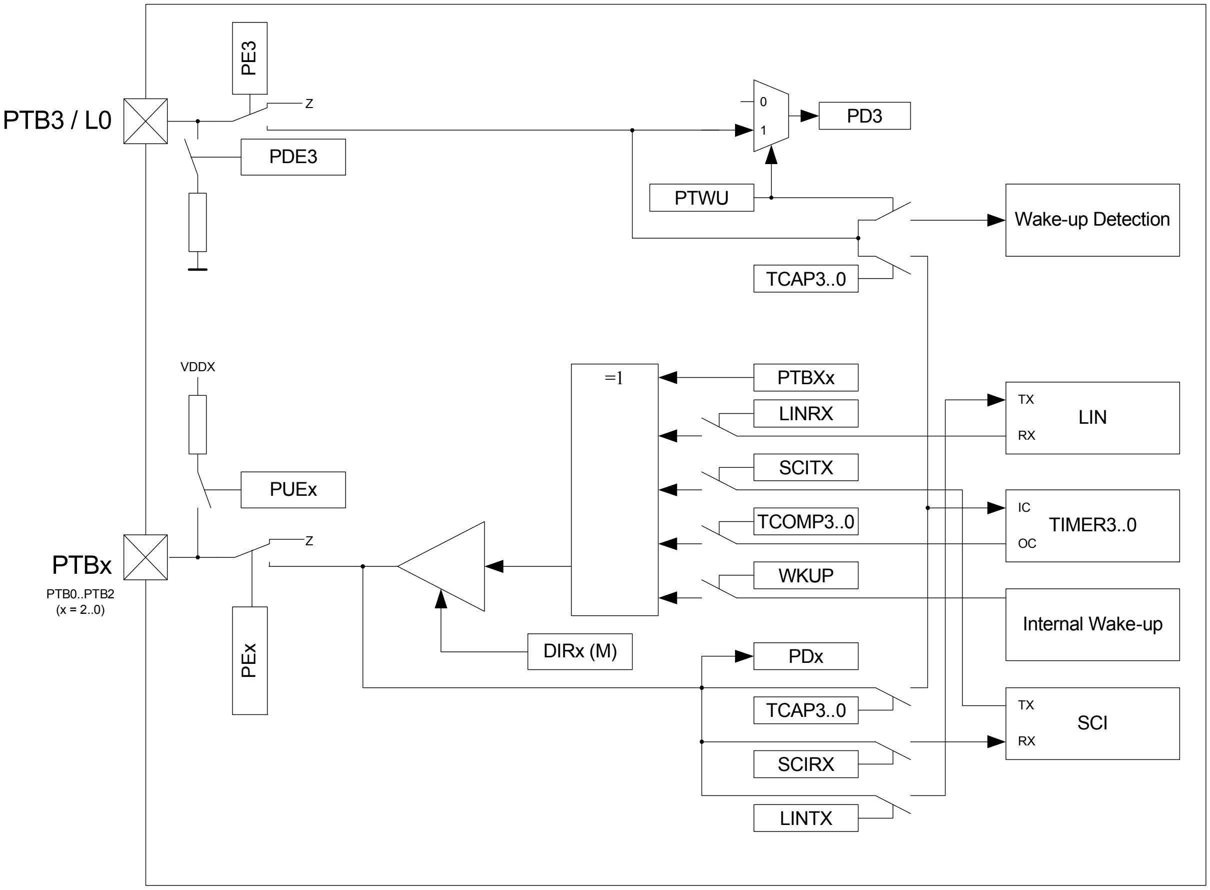


Figure 42. General purpose I/O - block diagram

4.11.4 High voltage wake-up input - PTB3 / L0

To offer robust high voltage wake-up capabilities, the following structure is implemented for PTB3/L0.

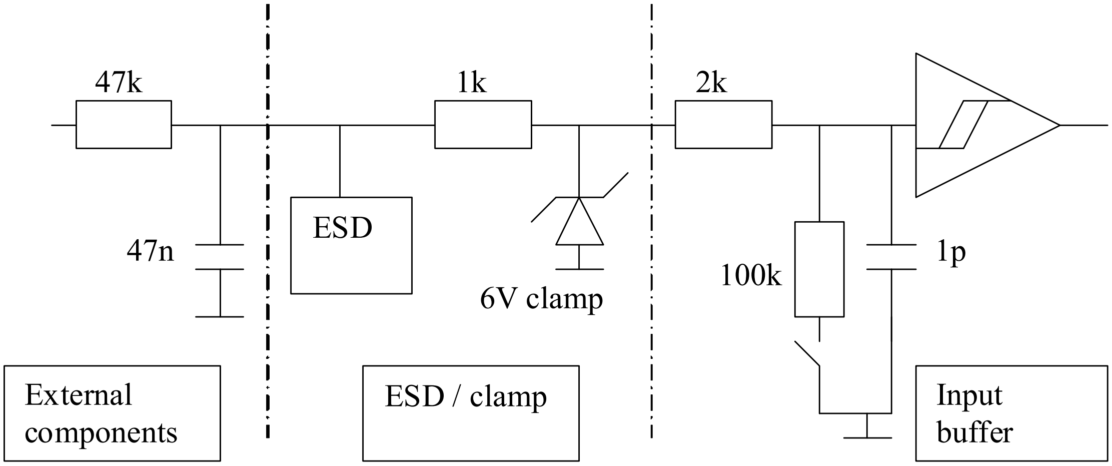


Figure 43. L0 / PTB3 input structure (typical values indicated)

NOTE

Due the different implementation of the L0/PTB3, the PTWU bit needs to be set in the GPIO_IN3 register, to read the port status PD3 during Normal mode.

4.11.4.1 Modes of operation

The full GPIO functionality is only available during Normal mode. The only features in available in both low power modes is the PTB3/L0 external wake-up and the wake-up routing of the timer output compare.

NOTE

TCOMP3...0 needs to be configured to allow timer output compare interrupts to generate a system wake-up.

4.11.5 Memory map and registers

4.11.5.1 Overview

This section provides a detailed description of the memory map and registers.

4.11.5.2 Module memory map

The memory map for the GPIO module is given in Table 62

Table 225. Module memory map

Offset [197], [198]	Name		7	6	5	4	3	2	1	0
0x40	GPIO_CTL GPIO control register	R	0	0	0	0	0	0	0	0
		W		DIR2M	DIR1M	DIR0M	PE3M	PE2M	PE1M	PE0M
		R	0	DIR2	DIR1	DIR0	PE3	PE2	PE1	PE0
		W								
0x42	GPIO_PUC	R	0	0	0	0	PDE3	PUE2	PUE1	PUE0
	GPIO pull up configuration	W								
0x43	GPIO_DATA	R	0	0	0	0	PD3	PD2	PD1	PD0
	GPIO port data register	W								
0x44	GPIO_IN0	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 0 input configuration	W								
0x45	GPIO_OUT0	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 0 output configuration	W								PTBX0
0x46	GPIO_IN1	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 1 input configuration	W								
0x47	GPIO_OUT1	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 1 output configuration	W								PTBX1
0x48	GPIO_IN2	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 2 input configuration	W								
0x49	GPIO_OUT2	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 2 output configuration	W								PTBX2
0x4A	GPIO_IN3	R	PTWU	PTWU	TCAP3	TCAP2	TCAP1	TCAP0	0	0
	Port 3 input configuration	W								
0x4B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4C	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4E	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x4F	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes

[197] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[198] Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

4.11.5.3 Register descriptions

4.11.5.3.1 GPIO control register (GPIO_CTL)

Table 226. GPIO control register (GPIO_CTL)

Offset [199],[200] 0x40

Access: User read/write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W		DIR2M	DIR1M	DIR0M	PE3M	PE2M	PE1M	PE0M
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	0	DIR2	DIR1	DIR0	PE3	PE2	PE1	PE0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[199] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[200] Those Registers are 16-Bit access only.

Table 227. GPIO control register (GPIO_CTL)

Field	Description
14 DIR2M	Data Direction PTB2 - Mask 0 - writing the DIR2 bit will have no effect 1 - writing the DIR2 bit will be effective
13 DIR1M	Data Direction PTB1 - Mask 0 - writing the DIR1 bit will have no effect 1 - writing the DIR1 bit will be effective
12 DIR0M	Data Direction PTB0 - Mask 0 - writing the DIR0 bit will have no effect 1 - writing the DIR0 bit will be effective
11 PE3M	Port 3 Enable - Mask 0 - writing the PE3 bit will have no effect 1 - writing the PE3 bit will be effective
10 PE2M	Port 2 Enable - Mask 0 - writing the PE2 bit will have no effect 1 - writing the PE2 bit will be effective
9 PE1M	Port 1 Enable - Mask 0 - writing the PE1 bit will have no effect 1 - writing the PE1 bit will be effective
8 PE0M	Port 0 Enable - Mask 0 - writing the PE0 bit will have no effect 1 - writing the PE0 bit will be effective
6 DIR2	Data Direction PTB2 0 - PTB2 configured as Input 1 - PTB2 configured as Output
5 DIR1	Data Direction PTB1 0 - PTB1 configured as Input 1 - PTB1 configured as Output

Table 227. GPIO control register (GPIO_CTL) (continued)

Field	Description
4 DIR0	Data Direction PTB0 0 - PTB0 configured as Input 1 - PTB0 configured as Output
3 PE3	Port 3 Enable[201] 0 - PTB3 Disabled (Z state) 1 - PTB3 Enabled (I)
2 PE2	Port 2 Enable[201] 0 - PTB2 disabled (Z state) 1 - PTB2 enabled (I/O)
1 PE1	Port 1 Enable[201] 0 - PTB1 disabled (Z state) 1 - PTB1 enabled (I/O)
0 PE0	Port 0 Enable[201] 0 - PTB0 disabled (Z state) 1 - PTB0 enabled (I/O)

Notes

[201] The port logic is always enabled. Setting PEx will connect the logic to the port I/O buffers.

4.11.5.3.2 GPIO pull-up configuration (GPIO_PUC)

Table 228. GPIO pull-up configuration (GPIO_PUC)

Offset[202] 0x42 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	PDE3	PUE2	PUE1	PUE0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[202] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 229. GPIO pull-up configuration (GPIO_PUC)

Field	Description
3 PDE3	PTB3 Pull-down Enable 0 - PTB3 pull-down disabled 1 - PTB3 pull-down enabled
2 PUE2	PTB2 Pull-up Enable 0 - PTB2 pull-up disabled 1 - PTB2 pull-up enabled
1 PUE1	PTB1 Pull-up Enable 0 - PTB1 pull-up disabled 1 - PTB1 pull-up enabled
0 PUE0	PTB0 Pull-up Enable 0 - PTB0 pull-up disabled 1 - PTB0 pull-up enabled

4.11.5.3.3 GPIO port data register (GPIO_DATA)

Table 230. GPIO port data register (GPIO_DATA)

Offset[203] 0x43 Access: User read

	7	6	5	4	3	2	1	0
R	0	0	0	0	PD3[204]	PD2	PD1	PD0
W								

Notes

[203] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[204] Due the different implementation of the L0/PTB3, PTWU needs to be set in the GPIO_IN3 to read the PD3 port status during normal mode.

Table 231. GPIO port data register (GPIO_DATA)

Field	Description
3 PD3	PTB3 Data Register A read returns the value of the PTB3 buffer.
2 PD2	PTB2 Data Register A read returns the value of the PTB2 buffer.
1 PD1	PTB1 Data Register A read returns the value of the PTB1 buffer.
0 PD0	PTB0 Data Register A read returns the value of the PTB0 buffer.

4.11.5.3.4 Port 0 input configuration (GPIO_IN0)

Table 232. Port 0 input configuration (GPIO_IN0)

Offset[205] 0x44 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[205] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 233. Port 0 input configuration (GPIO_IN0)

Field	Description
6 TCAP3	PTB0 - Timer Input Capture Channel 3 0 - PTB0 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB0 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	PTB0 - Timer Input Capture Channel 2 0 - PTB0 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB0 Input buffer routed to Timer Channel 2 - Input Capture
4 TCAP1	PTB0 - Timer Input Capture Channel 1 0 - PTB0 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB0 Input buffer routed to Timer Channel 1 - Input Capture
3 TCAP0	PTB0 - Timer Input Capture Channel 0 0 - PTB0 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB0 Input buffer routed to Timer Channel 0 - Input Capture

Table 233. Port 0 input configuration (GPIO_IN0) (continued)

Field	Description
2 SCIRX	PTB0 - SCI Module Rx Input 0 - PTB0 Input buffer disconnected from SCI Module Rx Input 1 - PTB0 Input buffer routed to SCI Module Rx Input
1 LINTX	PTB0 - LIN Module Tx Input 0 - PTB0 Input buffer disconnected from LIN Module Tx Input 1 - PTB0 Input buffer routed to LIN Module Tx Input

4.11.5.3.5 Port 0 output configuration (GPIO_OUT0)

Table 234. Port 0 output configuration (GPIO_OUT0)

Offset[206] 0x45

Access: User read/write

	7	6	5	4	3	2	1	0
R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
W								PTBX0
Reset	0	0	0	0	0	0	0	0

Notes

[206] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 235. Port 0 output configuration (GPIO_OUT0)

Field	Description
7 WKUP	PTB0 - Wake-up output 0 - Internal wake-up signal disconnected from PTB0 output buffer OR gate 1 - Internal wake-up signal connected to PTB0 output buffer OR gate
6 TCOMP3	PTB0 - Timer Channel 3 - Output Compare output 0 - Timer Channel 3 - output compare disconnected from PTB0 output buffer OR gate 1 - Timer Channel 3 - output compare connected to PTB0 output buffer OR gate
5 TCOMP2	PTB0 - Timer Channel 2 - Output Compare output 0 - Timer Channel 2 - output compare disconnected from PTB0 output buffer OR gate 1 - Timer Channel 2 - output compare connected to PTB0 output buffer OR gate
4 TCOMP1	PTB0 - Timer Channel 1 - Output Compare output 0 - Timer Channel 1 - output compare disconnected from PTB0 output buffer OR gate 1 - Timer Channel 1 - output compare connected to PTB0 output buffer OR gate
3 TCOMP0	PTB0 - Timer Channel 0 - Output Compare output 0 - Timer Channel 0 - output compare disconnected from PTB0 output buffer OR gate 1 - Timer Channel 0 - output compare connected to PTB0 output buffer OR gate
2 SCITX	PTB0 - SCI TX Output 0 - SCI TX output disconnected from PTB0 output buffer OR gate 1 - SCI TX output connected to PTB0 output buffer OR gate
1 LINRX	PTB0 - LIN RX Output 0 - LIN RX output disconnected from PTB0 output buffer OR gate 1 - LIN RX output connected to PTB0 output buffer OR gate
0 PTBX0	PTB0 - Output Buffer Control 0 - PTB0 output buffer OR gate input = 0 1 - PTB0 output buffer OR gate input = 1

4.11.5.3.6 Port 1 input configuration (GPIO_IN1)

Table 236. Port 1 input configuration (GPIO_IN1)

Offset[207] 0x46 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	
W								
Reset	0	0	0	0	0	0	0	0

Notes

[207] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 237. Port 1 input configuration (GPIO_IN1)

Field	Description
6 TCAP3	PTB1 - Timer Input Capture Channel 3 0 - PTB1 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	PTB1 - Timer Input Capture Channel 2 0 - PTB1 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 2 - Input Capture
4 TCAP1	PTB1 - Timer Input Capture Channel 1 0 - PTB1 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 1 - Input Capture
3 TCAP0	PTB1 - Timer Input Capture Channel 0 0 - PTB1 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 0 - Input Capture
2 SCIRX	PTB1 - SCI Module Rx Input 0 - PTB1 Input buffer disconnected from SCI Module Rx Input 1 - PTB1 Input buffer routed to SCI Module Rx Input
1 LINTX	PTB1 - LIN Module Tx Input 0 - PTB1 Input buffer disconnected from LIN Module Tx Input 1 - PTB1 Input buffer routed to LIN Module Tx Input

4.11.5.3.7 Port 1 output configuration (GPIO_OUT1)

Table 238. Port 1 output configuration (GPIO_OUT1)

Offset[208] 0x47 Access: User read/write

	7	6	5	4	3	2	1	0
R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
W								PTBX1
Reset	0	0	0	0	0	0	0	0

Notes

[208] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 239. Port 1 output configuration (GPIO_OUT1)

Field	Description
7 WKUP	PTB1 - Wake-up output 0 - Internal wake-up signal disconnected from PTB1 output buffer OR gate 1 - Internal wake-up signal connected to PTB1 output buffer OR gate
6 TCOMP3	PTB1 - Timer Channel 3 - Output Compare output 0 - Timer Channel 3 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 3 - output compare connected to PTB1 output buffer OR gate
5 TCOMP2	PTB1 - Timer Channel 2 - Output Compare output 0 - Timer Channel 2 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 2 - output compare connected to PTB1 output buffer OR gate
4 TCOMP1	PTB1 - Timer Channel 1 - Output Compare output 0 - Timer Channel 1 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 1 - output compare connected to PTB1 output buffer OR gate
3 TCOMP0	PTB1 - Timer Channel 0 - Output Compare output 0 - Timer Channel 0 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 0 - output compare connected to PTB1 output buffer OR gate
2 SCITX	PTB1 - SCI TX Output 0 - SCI TX output disconnected from PTB1 output buffer OR gate 1 - SCI TX output connected to PTB1 output buffer OR gate
1 LINRX	PTB1 - LIN RX Output 0 - LIN RX output disconnected from PTB1 output buffer OR gate 1 - LIN RX output connected to PTB1 output buffer OR gate
0 PTBX1	PTB1 - Output Buffer Control 0 - PTB1 output buffer OR gate input = 0 1 - PTB1 output buffer OR gate input = 1

4.11.5.3.8 Port 2 input configuration (GPIO_IN2)

Table 240. Port 2 input configuration (GPIO_IN2)

Offset[209] 0x48 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[209] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 241. Port 2 input configuration (GPIO_IN2)

Field	Description
6 TCAP3	PTB2 - Timer Input Capture Channel 3 0 - PTB2 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	PTB2 - Timer Input Capture Channel 2 0 - PTB2 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 2 - Input Capture
4 TCAP1	PTB2 - Timer Input Capture Channel 1 0 - PTB2 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 1 - Input Capture

Table 241. Port 2 input configuration (GPIO_IN2) (continued)

Field	Description
3 TCAP0	PTB2 - Timer Input Capture Channel 0 0 - PTB2 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 0 - Input Capture
2 SCIRX	PTB2 - SCI Module Rx Input 0 - PTB2 Input buffer disconnected from SCI Module Rx Input 1 - PTB2 Input buffer routed to SCI Module Rx Input
1 LINTX	PTB2 - LIN Module Tx Input 0 - PTB2 Input buffer disconnected from LIN Module Tx Input 1 - PTB2 Input buffer routed to LIN Module Tx Input

4.11.5.3.9 Port 2 output configuration (GPIO_OUT2)

Table 242. Port 2 output configuration (GPIO_OUT2)

Offset[209] 0x49

Access: User read/write

	7	6	5	4	3	2	1	0
R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
W								PTBX2
Reset	0	0	0	0	0	0	0	0

Notes

210. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 243. Port 2 output configuration (GPIO_OUT2)

Field	Description
7 WKUP	PTB2 - Wake-up output 0 - Internal wake-up signal disconnected from PTB2 output buffer OR gate 1 - Internal wake-up signal connected to PTB2 output buffer OR gate
6 TCOMP3	PTB2 - Timer Channel 3 - Output Compare output 0 - Timer Channel 3 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 3 - output compare connected to PTB2 output buffer OR gate
5 TCOMP2	PTB2 - Timer Channel 2 - Output Compare output 0 - Timer Channel 2 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 2 - output compare connected to PTB2 output buffer OR gate
4 TCOMP1	PTB2 - Timer Channel 1 - Output Compare output 0 - Timer Channel 1 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 1 - output compare connected to PTB2 output buffer OR gate
3 TCOMP0	PTB2 - Timer Channel 0 - Output Compare output 0 - Timer Channel 0 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 0 - output compare connected to PTB2 output buffer OR gate
2 SCITX	PTB2 - SCI TX Output 0 - SCI TX output disconnected from PTB2 output buffer OR gate 1 - SCI TX output connected to PTB2 output buffer OR gate
1 LINRX	PTB2 - LIN RX Output 0 - LIN RX output disconnected from PTB2 output buffer OR gate 1 - LIN RX output connected to PTB2 output buffer OR gate
0 PTBX2	PTB2 - Output Buffer Control 0 - PTB2 output buffer OR gate input = 0 1 - PTB2 output buffer OR gate input = 1

4.11.5.3.10 Port 3 input configuration (GPIO_IN3)

Table 244. Port 3 input configuration (GPIO_IN3)

Offset[211] 0x4A

Access: User read/write

	7	6	5	4	3	2	1	0
R	PTWU	TCAP3	TCAP2	TCAP1	TCAP0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[211] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 245. Port 3 input configuration (GPIO_IN3)

Field	Description
7 PTWU	PTB3 Wake-up 0 - PTB3 Input buffer low power mode wake-up circuity disabled 1 - PTB3 Input buffer low power mode wake-up circuity enabled
6 TCAP3	PTB3 - Timer Input Capture Channel 3 0 - PTB3 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	PTB3 - Timer Input Capture Channel 2 0 - PTB3 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 2 - Input Capture
4 TCAP1	PTB3 - Timer Input Capture Channel 1 0 - PTB3 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 1 - Input Capture
3 TCAP0	PTB3 - Timer Input Capture Channel 0 0 - PTB3 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 0 - Input Capture

4.12 LIN

4.12.1 Introduction

The LIN bus pin provides a physical layer for single-wire communication in automotive applications. The LIN physical layer is designed to meet the LIN physical layer version 2.0 / 2.1 and J2602 specification, and has the following features:

- LIN physical layer 2.0 / 2.1 / J2602 compliant
- Slew rate selection 20 kBit, 10 kBit, and fast Mode (100 kBit)
- Overtemperature Shutdown - HTI
- Permanent Pull-up in Normal mode 30 kΩ, 1.0 MΩ in low power
- Current limitation
- Special J2602 compliant configuration
- Direct Rx / Tx access
- Optional external Rx / Tx access and routing to the TIMER Input through PTBx

The LIN driver is a low side MOSFET with current limitation and thermal shutdown. An internal pull-up resistor with a serial diode structure is integrated, so no external pull-up components are required for the application in a slave node. The fall time from dominant to recessive and the rise time from recessive to dominant is controlled. The symmetry between both slopes is guaranteed.

4.12.2 Overview

4.12.2.1 Block diagram

Figure 44 shows the basic function of the LIN module.

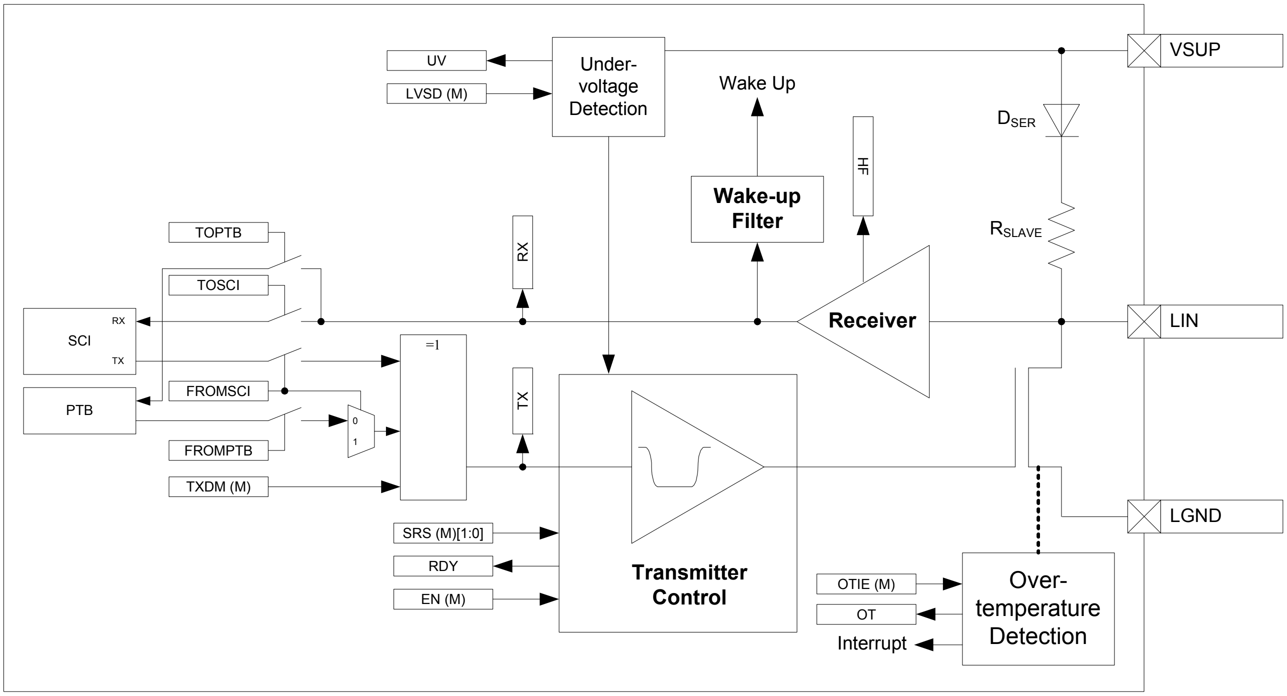


Figure 44. LIN module block diagram

4.12.2.2 LIN pin

The LIN pin offers high susceptibility immunity level from external disturbance, guaranteeing communication during external disturbances. See Section 3.8, “Electromagnetic compatibility (EMC)".

4.12.2.3 Slew rate selection

The slew rate can be selected for optimized operation at 10 kBit/s and 20 kBit/s as well as a fast baud rate (100 kBit) for test and programming. The slew rate can be adapted with the bits SRS[1:0] in the LIN Control Register (LIN_CTL). The initial slew rate is 20 kBit/s.

4.12.2.4 Overtemperature shutdown (LIN interrupt)

The output low side FET (transmitter) is protected against overtemperature conditions. In an overtemperature condition, the transmitter will be shut down, and the TO bit in the LIN Control Register (LIN_CTL) is set as long as the condition is present.

If the OTIEM bit is set in the LIN Status Register (LIN_SR), an Interrupt IRQ will be generated. Acknowledge the interrupt by writing a “1” in the LIN Status Register (LIN_SR). To issue a new interrupt, the condition has to vanish and reoccur.

The transmitter is automatically re-enabled once the overtemperature condition is gone and TxD is High.

4.12.2.5 Low power mode and wake-up feature

During Low Power mode operation, the transmitter of the physical layer is disabled. The receiver is still active and able to detect wake-up events on the LIN bus line.

A dominant level longer than t_{PROPWL}, followed by a rising edge will generate a wake-up event and be reported in the Wake-up Source Register (WSR).

4.12.2.6 J2602 compliance

A Low Voltage Shutdown feature was implemented to allow controlled LIN driver behavior under low voltage conditions at VSUP. If LVSD is set, once VSUP is below the threshold VJ2602H, the LIN transmitter is not turned dominant again. The condition is indicated by the UV flag.

4.12.2.7 Transmit/receiving line definition

The LIN module can be connected to the SCI or PTB module, or can be directly controlled by the TXDM / RX bit

4.12.2.8 Transmitter enable/ready

The LIN transmitter must be enabled before transmission is possible (EN). The RDY bit is set to 1 about 50 µs after the LIN transmitter is enabled. This is due to the initialization time for the LIN transmitter, under some low voltage conditions.

During this period (LIN enabled to RDY = 1), the LIN is forced to a recessive state.

4.12.3 Memory map and registers

4.12.3.1 Overview

This section provides a detailed description of the memory map and registers.

4.12.3.2 Module memory map

The memory map for the LIN module is given in Table 62

Table 246. Module memory map

Offset [212],[213]	Name		7	6	5	4	3	2	1	0
0x50	LIN_CTL LIN control register	R	0	0	0	0	0	0	0	0
		W	OTIEM			TXDM	LVSDM	ENM	SRSM	
		R	OTIE	0	0	TXD	LVSD	EN	SRS	
		W								
0x52	LIN_SR (hi)	R	OT	0	HF	0	UV	0	0	0
	LIN status register	W	Write 1 will clear the flags							
0x53	LIN_SR (lo)	R	RDY	0	0	0	0	0	RX	TX
	LIN status register	W								
0x54	LIN_TX	R	0	0	0	0	0	0	FROMPTB	FROMSCI
	LIN transmit line definition	W								
0x55	LIN_RX	R	0	0	0	0	0	0	TOPTB	TOSCI
	LIN receive line definition	W								
0x56	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x57	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes

[212] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[213] This Register is 16-Bit access only.

4.12.3.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of the register bit and field function follow the register diagrams, in bit order.

4.12.3.3.1 LIN control register (LIN_CTL)

Table 247. LIN control register (LIN_CTL)

Offset [214] ,[215] 0x50

Access: User write

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	OTIEM			TXDM	LVSDM	ENM	SRSM	
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	OTIE	0	0	TXD	LVSD	EN	SRS	
W								
Reset	0	0	0	0	0	0	0	0

Notes

[214] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

[215] This Register is 16-Bit access only.

Table 248. LIN control register (LIN_CTL) - register field descriptions

Field	Description
15 OTIEM	LIN Overtemperature Interrupt Enable - Mask 0 - writing the OTIE Bit will have no effect 1 - writing the OTIE Bit will be effective
12 TXDM	IN - Direct Transmitter Control - Mask 0 - writing the TXD Bit will have no effect 1 - writing the TXD Bit will be effective
11 LVSDM	LIN - Low Voltage Shutdown Disable (J2602 Compliance Control) - Mask 0 - writing the LVSD Bit will have no effect 1 - writing the LVSD Bit will be effective
10 ENM	LIN Module Enable - Mask 0 - writing the EN Bit will have no effect 1 - writing the EN Bit will be effective
9-8 SRSM[1:0]	LIN - Slew Rate Select - Mask 00,01,10 - writing the SRS Bits will have no effect 11 - writing the SRS Bits will be effective
7 OTIE	LIN Overtemperature Interrupt Enable 0 - LIN overtemperature interrupt disabled 1 - LIN overtemperature interrupt enabled
4 TXD	IN - Direct Transmitter Control 0 - Transmitter not controlled 1 - Transmitter dominant
3 LVSD	LIN - Low Voltage Shutdown Disable (J2602 Compliance Control) 0 - LIN will be remain in recessive state in case of V_{SUP} undervoltage condition 1 - LIN will stay functional even with a V_{SUP} undervoltage condition

Table 248. LIN control register (LIN_CTL) - register field descriptions (continued)

Field	Description
2 EN	LIN Module Enable 0 - LIN module disabled 1 - LIN module enabled
1-0 SRS[1:0]	LIN - Slew Rate Select 00 - Normal slew rate (20 kBit) 01 - Slow slew rate (10.4 kBit) 10 - Fast slew rate (100 kbit) 11 - normal Slew Rate (20 kBit)

4.12.3.3.2 LIN status register (LIN_SR (hi))

Table 249. LIN status register (LIN_SR (hi))

Offset[216] 0x52 Access: User read/write

	7	6	5	4	3	2	1	0
R	OT	0	HF	0	UV	0	0	0
W	Write 1 will clear the flags							
Reset	0	0	0	0	0	0	0	0

Notes

[216] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 250. LIN status register (LIN_SR (hi)) - register field descriptions

Field	Description
7 OT	LIN Overtemperature Status. This bit is latched and has to be reset by writing 1 into OT bit. 0 - No LIN overtemperature condition detected 1 - LIN overtemperature condition detected
5 HF	LIN HF (High Frequency) Condition Status indicating HF (DPI) disturbance in the LIN module. This bit is latched and has to be reset by writing 1 into HF bit. 0 - No LIN HF (DPI) condition detected 1 - LIN HF (DPI) condition detected
3 UV	LIN Undervoltage Status. This threshold is used for the J2602 feature as well. This bit is latched and has to be reset by writing 1 into UV bit. 0 - No LIN undervoltage condition detected 1 - LIN undervoltage condition detected

4.12.3.3.3 LIN status register (LIN_SR (lo))

Table 251. LIN status register (LIN_SR (lo))

Offset[217] 0x53 Access: User read

	7	6	5	4	3	2	1	0
R	RDY	0	0	0	0	0	RX	TX
W								

Notes

[217] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 252. LIN status register (LIN_SR (lo)) - register field descriptions

Field	Description
1 RDY	Transmitter Ready Status 0 - Transmitter not ready 1 - Transmitter ready
1 RX	Current RX status 0 - Rx recessive 1 - Rx dominant
0 TX	Current TX status 0 - Tx recessive 1 - Tx dominant

4.12.3.3.4 LIN transmit line definition (LIN_TX)

Table 253. LIN transmit line definition (LIN_TX)

Offset[218] 0x54 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	FROMPTB	FROMSCI
W								
Reset	0	0	0	0	0	0	0	0

Notes

[218] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 254. LIN transmit line definition (LIN_TX) - register field descriptions

Field	Description
1 FROMPTB	LIN_TX internally routed from PTB. See Section 4.11, “General purpose I/O - GPIO" for details.[219] 0 - LIN transmitter disconnected from PTB module. 1 - LIN transmitter connected to the PTB module.
0 FROMSCI	LIN_TX internally routed from SCI[219] 0 - LIN transmitter disconnected from SCI module. 1 - LIN transmitter connected to the SCI module.

Notes

[219] In case both, FROMPTB and FROMSCI are selected, the SCI has priority and the PTB signal is ignored. In any case, the signal is logically ORed with the TXD direct transmitter control.

4.12.3.3.5 LIN receive line definition (LIN_RX)

Table 255. LIN receive line definition (LIN_RX)

Offset[220] 0x55 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	TOPTB	TOSCI
W								
Reset	0	0	0	0	0	0	0	0

Notes

[220] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 256. LIN receive line definition (LIN_RX) - register field descriptions

Field	Description
1 TOPTB	LIN_RX internally routed to PTB 0 - LIN receiver disconnected from PTB module. 1 - LIN receiver connected to the PTB module.
0 TOSCI	LIN_RX internally routed to SCI 0 - LIN receiver disconnected from SCI module. 1 - LIN receiver connected to the SCI module.

NOTE

In order to route the RX signal to the Timer Input capture, one of the PTBx must be configured as a pass through.

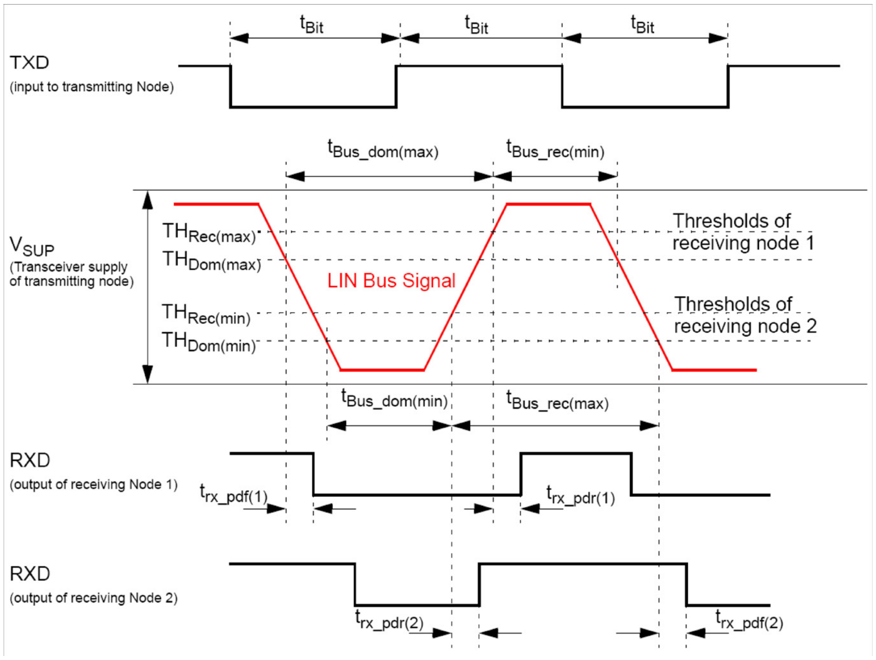


Figure 45. Definition of LIN bus timing parameters

4.13 Serial communication interface (S08SCIV4)

4.13.1 Introduction

4.13.1.1 Features

Features of SCI module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- Interrupt-driven or polled operation:
  - Transmit data register empty and transmission complete
  - Receive data register full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN

- Hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Receiver wake-up by idle-line or address-mark
- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output polarity

4.13.1.2 Modes of operation

See Section 4.13.3, “Functional description", for details concerning SCI operation in these modes:

- 8- and 9-bit data modes
- Loop mode
- Single-wire mode

4.13.1.3 Block diagram

Figure 46 shows the transmitter portion of the SCI.

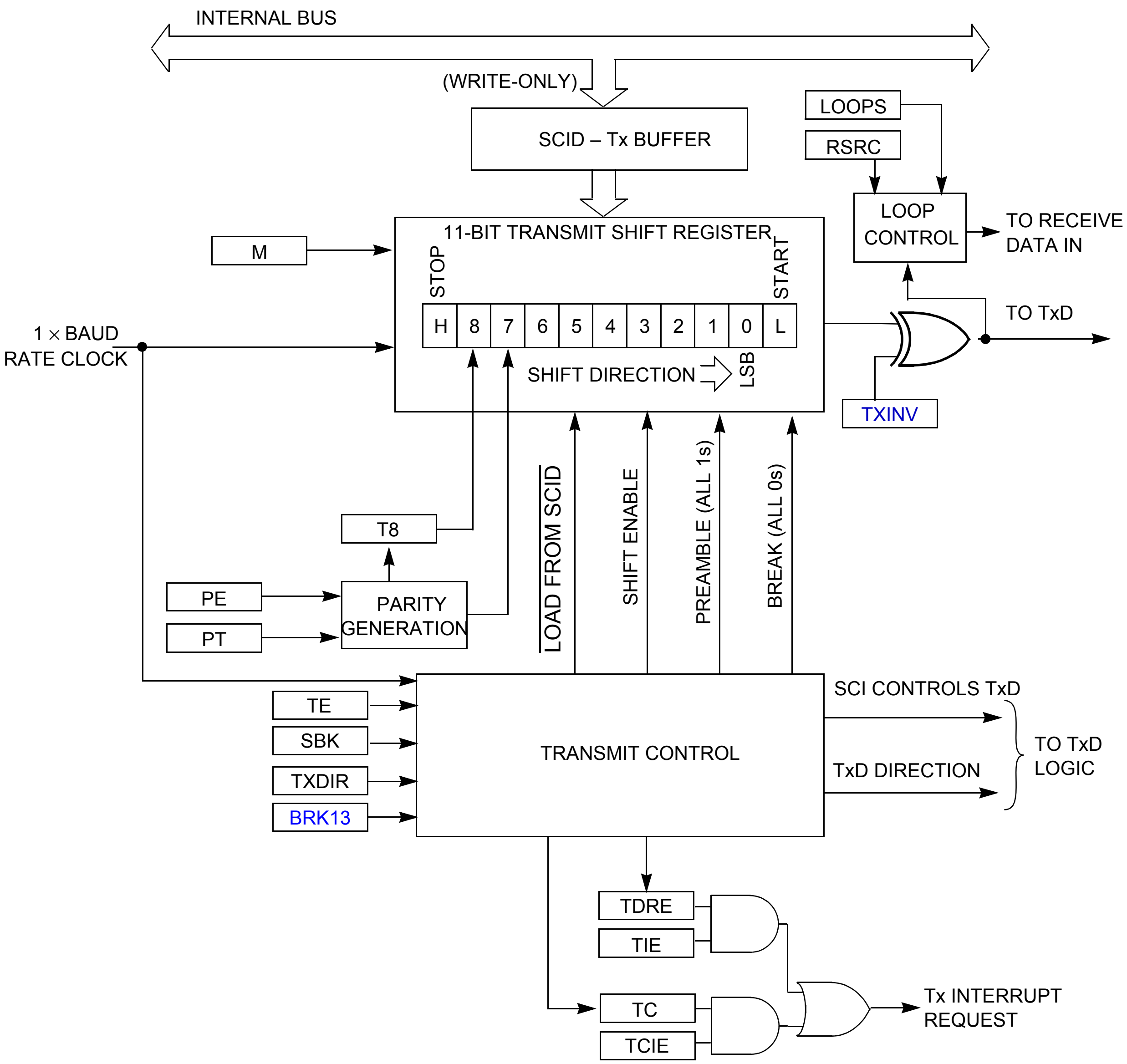


Figure 46. SCI transmitter block diagram

Figure 47 shows the receiver portion of the SCI.

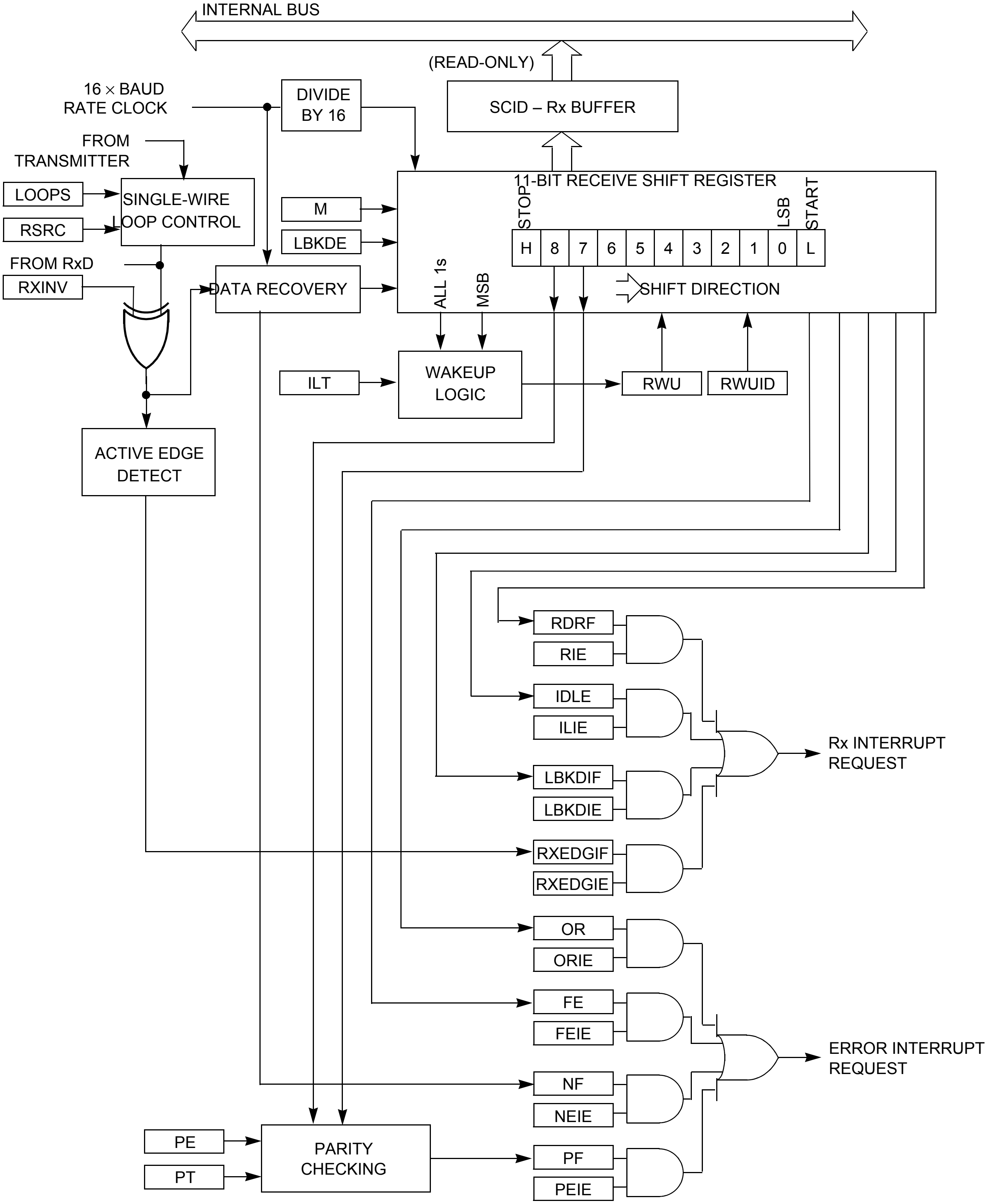


Figure 47. SCI receiver block diagram

4.13.2 Memory map and registers

4.13.2.1 Overview

This section provides a detailed description of the memory map and registers.

4.13.2.2 Module memory map

The memory map for the S08SCIV4 module is given in Table 62.

Table 257. Module memory map

Offset[221]	Name		7	6	5	4	3	2	1	0
0x18	SCIBD (hi)	R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
	SCI Baud Rate Register	W								
0x19	SCIBD (lo)	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	SCI Baud Rate Register	W								
0x1A	SCIC1	R	LOOPS	0	RSRC	M	0	ILT	PE	PT
	SCI Control Register 1	W								
0x1B	SCIC2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	SCI Control Register 2	W								
0x1C	SCIS1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	SCI Status Register 1	W								
0x1D	SCIS2	R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
	SCI Status Register 2	W								
0x1E	SCIC3	R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
	SCI Control Register 3	W								
0x1F	SCID	R	R7	R6	R5	R4	R3	R2	R1	R0
	SCI Data Register	W	T7	T6	T5	T4	T3	T2	T1	T0

Notes

[221] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.13.2.3 Register definition

The SCI has eight 8-bit registers to control baud rate, select SCI options, report SCI status, and for transmit/receive data.

4.13.2.3.1 SCI baud rate registers (SCIBD (hi), SCIBD (lo))

This pair of registers control the prescale divisor for SCI baud rate generation. To update the 13-bit baud rate setting [SBR12:SBR0], first write to SCIBD (hi) to buffer the high half of the new value, and then write to SCIBD (lo). The working value in SCIBD (hi) does not change until SCIBD (lo) is written.

SCIBDL is reset to a non-zero value, so after reset the baud rate generator remains disabled until the first time the receiver or transmitter is enabled (RE or TE bits in SCIC2 are written to 1).

Table 258. SCI baud rate register (SCIBD (hi))

Offset[222] 0x18 Access: User read/write

	7	6	5	4	3	2	1	0
R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
W								
Reset	0	0	0	0	0	0	0	0

□ = Unimplemented or Reserved

Notes

222. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 259. SCIBD (hi) field descriptions

Field	Description
7 LBKDIE	**LIN Break Detect Interrupt Enable (for LBKDIF)** 0 Hardware interrupts from LBKDIF disabled (use polling). 1 Hardware interrupt requested when LBKDIF flag is 1.
6 RXEDGIE	**RxD Input Active Edge Interrupt Enable (for RXEDGIF)** 0 Hardware interrupts from RXEDGIF disabled (use polling). 1 Hardware interrupt requested when RXEDGIF flag is 1.
4:0 SBR[12:8]	**Baud Rate Modulo Divisor** — The 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(64×BR). See BR bits in Table 260.

Table 260. SCI baud rate register (SCIBDL)

Offset[223] 0x19 Access: User read/write

	7	6	5	4	3	2	1	0
R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
W								
Reset	0	0	0	0	0	1	0	0

Notes

223. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 261. SCIBDL field descriptions

Field	Description
7:0 SBR[7:0]	**Baud Rate Modulo Divisor** — These 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(64×BR). See also BR bits in Table 258.

4.13.2.3.2 SCI control register 1 (SCIC1)

This read/write register is used to control various optional features of the SCI system.

Table 262. SCI control register 1 (SCIC1)

Offset[224] 0x1A

Access: User read/write

	7	6	5	4	3	2	1	0
R	LOOPS	0	RSRC	M	0	ILT	PE	PT
W								
Reset	0	0	0	0	0	0	0	0

Notes

[224] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 263. SCIC1 field descriptions

Field	Description
7 LOOPS	**Loop Mode Select** — Selects between loop back modes and normal 2-pin full-duplex modes. When LOOPS = 1, the transmitter output is internally connected to the receiver input. 0 Normal operation — RxD and TxD use separate pins. 1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input. (See RSRC bit.) RxD pin is not used by SCI.
5 RSRC	**Receiver Source Select** — This bit has no meaning or effect unless the LOOPS bit is set to 1. When LOOPS = 1, the receiver input is internally connected to the TxD pin and RSRC determines whether this connection is also connected to the transmitter output. 0 Provided LOOPS = 1, RSRC = 0 selects internal loop back mode and the SCI does not use the RxD pins. 1 Single-wire SCI mode where the TxD pin is connected to the transmitter output and receiver input.
4 M	**9-Bit or 8-Bit Mode Select** 0 Normal — start + 8 data bits (LSB first) + stop. 1 Receiver and transmitter use 9-bit data characters start + 8 data bits (LSB first) + 9th data bit + stop.
2 ILT	**Idle Line Type Select** — Setting this bit to 1 ensures that the stop bit and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. Refer to Section 4.13.3.3.2.1, “Idle-line wake-up" for more information. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	**Parity Enable** — Enables hardware parity generation and checking. When parity is enabled, the most significant bit (MSB) of the data character (eighth or ninth data bit) is treated as the parity bit. 0 No hardware parity generation or checking. 1 Parity enabled.
0 PT	**Parity Type** — Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even. 0 Even parity. 1 Odd parity.

4.13.2.3.3 SCI control register 2 (SCIC2)

This register can be read or written at any time.

Table 264. SCI control register 2 (SCIC2)

Offset[225] 0x1B Access: User read/write

	7	6	5	4	3	2	1	0
R W	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

Notes

225. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 265. SCIC2 field descriptions

Field	Description
7 TIE	**Transmit Interrupt Enable (for TDRE)** 0 Hardware interrupts from TDRE disabled (use polling). 1 Hardware interrupt requested when TDRE flag is 1.
6 TCIE	**Transmission Complete Interrupt Enable (for TC)** 0 Hardware interrupts from TC disabled (use polling). 1 Hardware interrupt requested when TC flag is 1.
5 RIE	**Receiver Interrupt Enable (for RDRF)** 0 Hardware interrupts from RDRF disabled (use polling). 1 Hardware interrupt requested when RDRF flag is 1.
4 ILIE	**Idle Line Interrupt Enable (for IDLE)** 0 Hardware interrupts from IDLE disabled (use polling). 1 Hardware interrupt requested when IDLE flag is 1.
3 TE	**Transmitter Enable** 0 Transmitter off. 1 Transmitter on. TE must be 1 in order to use the SCI transmitter. When TE = 1, the SCI forces the TxD pin to act as an output for the SCI system. When the SCI is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single SCI communication line (TxD pin). TE also can be used to queue an idle character by writing TE = 0 then TE = 1 while a transmission is in progress. Refer to Section 4.13.3.2.1, “Send break and queued idle" for more details. When TE is written to 0, the transmitter keeps control of the port TxD pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to revert to a general-purpose I/O pin.
2 RE	**Receiver Enable** — When the SCI receiver is off, the RxD pin reverts to being a general-purpose port I/O pin. If LOOPS = 1 the RxD pin reverts to being a general-purpose I/O pin even if RE = 1. 0 Receiver off. 1 Receiver on.
1 RWU	**Receiver Wake-up Control** — This bit can be written to 1 to place the SCI receiver in a standby state where it waits for automatic hardware detection of a selected wake-up condition. The wake-up condition is either an idle line between messages (WAKE = 0, idle-line wake-up), or a logic 1 in the most significant data bit in a character (WAKE = 1, address-mark wake-up). Application software sets RWU and (normally) a selected hardware condition automatically clears RWU. Refer to Section 4.13.3.3.2, “Receiver wake-up operation" for more details. 0 Normal SCI receiver operation. 1 SCI receiver in standby waiting for wake-up condition.
0 SBK	**Send Break** — Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 (13 or 14 if BRK13 = 1) bit times of logic 0 are queued as long as SBK = 1. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. Refer to Section 4.13.3.2.1, “Send break and queued idle" for more details. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

4.13.2.3.4 SCI status register 1 (SCIS1)

This register has eight read-only status flags. Writes have no effect. Special software sequences (which do not involve writing to this register) are used to clear these status flags.

Table 266. SCI status register 1 (SCIS1)

Offset[226] 0x1C

Access: User read/write

	7	6	5	4	3	2	1	0
R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W								
Reset	1	1	0	0	0	0	0	0

□ = Unimplemented or Reserved

Notes

[226] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 267. SCIS1 field descriptions

Field	Description
7 TDRE	**Transmit Data Register Empty Flag** — TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCIS1 with TDRE = 1 and then write to the SCI data register (SCID). 0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.
6 TC	**Transmission Complete Flag** — TC is set out of reset and when TDRE = 1 and no data, preamble, or break character is being transmitted. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete). TC is cleared automatically by reading SCIS1 with TC = 1 and then doing one of the following three things: • Write to the SCI data register (SCID) to transmit new data • Queue a preamble by changing TE from 0 to 1 • Queue a break character by writing 1 to SBK in SCIC2
5 RDRF	**Receive Data Register Full Flag** — RDRF becomes set when a character transfers from the receive shifter into the receive data register (SCID). If the OR bit is set, further received characters are lost and RDRF bit does not set. To clear RDRF, read SCIS1 with RDRF = 1 and then read the SCI data register (SCID). 0 Receive data register empty. 1 Receive data register full.
4 IDLE	**Idle Line Flag** — IDLE is set when the SCI receive line becomes idle for a full character time after a period of activity. When ILT = 0, the receiver starts counting idle bit times after the start bit. So if the receive character is all 1s, these bit times and the stop bit time count toward the full character time of logic high (10 or 11 bit times depending on the M control bit) needed for the receiver to detect an idle line. When ILT = 1, the receiver doesn't start counting idle bit times until after the stop bit. So the stop bit and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line. To clear IDLE, read SCIS1 with IDLE = 1 and then read the SCI data register (SCID). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE will get set only once even if the receive line remains idle for an extended period. 0 No idle line detected. 1 Idle line was detected.
3 OR	**Receiver Overrun Flag** — OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from SCID yet. In this case, the new character (and all associated error information) is lost because there is no room to move it into SCID. To clear OR, read SCIS1 with OR = 1 and then read the SCI data register (SCID). 0 No overrun. 1 Receive overrun (new SCI data lost).
2 NF	**Noise Flag** — The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bit. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF will be set at the same time as the flag RDRF gets set for the character. To clear NF, read SCIS1 and then read the SCI data register (SCID). 0 No noise detected. 1 Noise detected in the received character in SCID.

Table 267. SCIS1 field descriptions (continued)

Field	Description
1 FE	**Framing Error Flag** — FE is set at the same time as RDRF when the receiver detects a logic 0 where the stop bit was expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, read SCIS1 with FE = 1 and then read the SCI data register (SCID). 0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.
0 PF	**Parity Error Flag** — PF is set at the same time as RDRF when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, read SCIS1 and then read the SCI data register (SCID). 0 No parity error. 1 Parity error.

4.13.2.3.5 SCI status register 2 (SCIS2)

This register has one read-only status flag.

Table 268. SCI status register 2 (SCIS2)

Offset[227] 0x1D

Access: User read/write

	7	6	5	4	3	2	1	0
R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
W								
Reset	0	0	0	0	0	0	0	0

□ = Unimplemented or Reserved

Notes

227. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 269. SCIS2 field descriptions

Field	Description
7 LBKDIF	**LIN Break Detect Interrupt Flag** — LBKDIF is set when the LIN break detect circuitry is enabled and a LIN break character is detected. LBKDIF is cleared by writing a “1” to it. 0 No LIN break character has been detected. 1 LIN break character has been detected.
6 RXEDGIF	**RxD Pin Active Edge Interrupt Flag** — RXEDGIF is set when an active edge (falling if RXINV = 0, rising if RXINV=1) on the RxD pin occurs. RXEDGIF is cleared by writing a “1” to it. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
4 RXINV[228]	**Receive Data Inversion** — Setting this bit reverses the polarity of the received data input. 0 Receive data not inverted 1 Receive data inverted
3 RWUID	**Receive Wake Up Idle Detect**— RWUID controls whether the idle character that wakes up the receiver sets the IDLE bit. 0 During receive standby state (RWU = 1), the IDLE bit does not get set upon detection of an idle character. 1 During receive standby state (RWU = 1), the IDLE bit gets set upon detection of an idle character.
2 BRK13	**Break Character Generation Length** — BRK13 is used to select a longer transmitted break character length. Detection of a framing error is not affected by the state of this bit. 0 Break character is transmitted with length of 10 bit times (11 if M = 1) 1 Break character is transmitted with length of 13 bit times (14 if M = 1)
1 LBKDE	**LIN Break Detection Enable**— LBKDE is used to select a longer break character detection length. While LBKDE is set, framing error (FE) and receive data register full (RDRF) flags are prevented from setting. 0 Break character detection enabled. 1 Break character detection disabled.

Table 269. SCIS2 field descriptions (continued)

Field	Description
0 RAF	**Receiver Active Flag** — RAF is set when the SCI receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. This status flag can be used to check whether an SCI character is being received before instructing the MCU to go to stop mode. 0 SCI receiver idle waiting for a start bit. 1 SCI receiver active (RxD input not idle).

Notes

[228] Setting RXINV inverts the RxD input for all cases: data bits, start and stop bits, break, and idle.

When using an internal oscillator in a LIN system, it is necessary to raise the break detection threshold by one bit time. Under the worst case timing conditions allowed in LIN, it is possible that a 0x00 data character can appear to be 10.26 bit times long at a slave which is running 14% faster than the master. This would trigger normal break detection circuitry, which is designed to detect a 10 bit break symbol. When the LBKDE bit is set, framing errors are inhibited and the break detection threshold changes from 10 bits to 11 bits, preventing false detection of a 0x00 data character as a LIN break symbol.

4.13.2.3.6 SCI control register 3 (SCIC3)

Table 270. SCI control register 3 (SCIC3)

Offset[229] 0x1E

Access: User read/write

	7	6	5	4	3	2	1	0
R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
W								
Reset	0	0	0	0	0	0	0	0

□ = Unimplemented or Reserved

Notes

[229] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 271. SCIC3 field descriptions

Field	Description
7 R8	**Ninth Data Bit for Receiver** — When the SCI is configured for 9-bit data (M = 1), R8 can be thought of as a ninth receive data bit to the left of the MSB of the buffered data in the SCID register. When reading 9-bit data, read R8 before reading SCID, because reading SCID completes automatic flag clearing sequences, which could allow R8 and SCID to be overwritten with new data.
6 T8	**Ninth Data Bit for Transmitter** — When the SCI is configured for 9-bit data (M = 1), T8 may be thought of as a ninth transmit data bit to the left of the MSB of the data in the SCID register. When writing 9-bit data, the entire 9-bit value is transferred to the SCI shift register after SCID is written, so T8 should be written (if it needs to change from its previous value) before SCID is written. If T8 does not need to change in the new value (such as when it is used to generate mark or space parity), it need not be written each time SCID is written.
5 TXDIR	**TxD Pin Direction in Single-wire Mode** — When the SCI is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this bit determines the direction of data at the TxD pin. 0 TxD pin is an input in single-wire mode. 1 TxD pin is an output in single-wire mode.
4 TXINV[230]	**Transmit Data Inversion** — Setting this bit reverses the polarity of the transmitted data output. 0 Transmit data not inverted 1 Transmit data inverted
3 ORIE	**Overrun Interrupt Enable** — This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled (use polling). 1 Hardware interrupt requested when OR = 1.
2 NEIE	**Noise Error Interrupt Enable** — This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled (use polling). 1 Hardware interrupt requested when NF = 1.

Table 271. SCIC3 field descriptions (continued)

Field	Description
1 FEIE	**Framing Error Interrupt Enable** — This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled (use polling). 1 Hardware interrupt requested when FE = 1.
0 PEIE	**Parity Error Interrupt Enable** — This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled (use polling). 1 Hardware interrupt requested when PF = 1.

Notes

230. Setting TXINV inverts the TxD output for all cases: data bits, start and stop bits, break, and idle.

4.13.2.3.7 SCI data register (SCID)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for the SCI status flags.

Table 272. SCI data register (SCID)

Offset[231] 0x1D　　Access: User read/write

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Notes

231. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.13.3 Functional description

The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The SCI comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the SCI, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the SCI.

4.13.3.1 Baud rate generation

Figure 48 shows the clock source for the SCI baud rate generator is the D2D clock / 4.

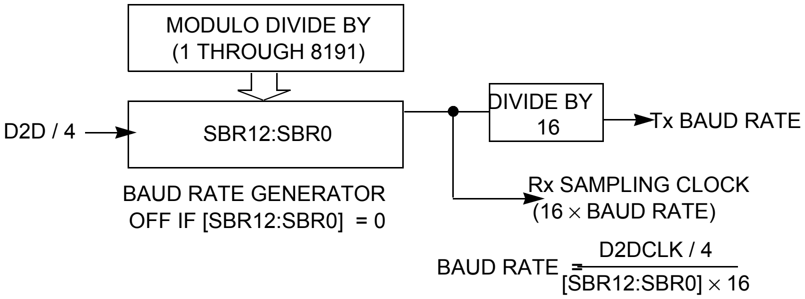


Figure 48. SCI baud rate generation

SCI communications require the transmitter and receiver (which typically derive baud rates from independent clock sources) to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition, but in the worst case, there are no such transitions in the full 10- or 11-bit time character frame so any mismatch in baud rate is accumulated for the whole character time. For a NXP Semiconductor

SCI system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ±4.5 percent for 8-bit data format and about ±4.0 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates that exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

4.13.3.2 Transmitter functional description

This section describes the overall block diagram for the SCI transmitter, as well as specialized functions for sending break and idle characters. The transmitter block diagram is shown in Figure 46.

The transmitter output (TxD) idle state defaults to logic high (TXINV = 0 following reset). The transmitter output is inverted by setting TXINV = 1. The transmitter is enabled by setting the TE bit in SCIC2. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the SCI data register (SCID).

The central element of the SCI transmitter is the transmit shift register that is either 10 or 11 bits long depending on the setting in the M control bit. For the remainder of this section, we will assume M = 0, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new SCI character, the value waiting in the transmit data register is transferred to the shift register (synchronized with the baud rate clock) and the transmit data register empty (TDRE) status flag is set to indicate another character may be written to the transmit data buffer at SCID.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the TxD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TxD high, waiting for more characters to transmit.

Writing 0 to TE does not immediately release the pin to be a general purpose I/O pin. Any transmit activity that is in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

4.13.3.2.1 Send break and queued idle

The SBK control bit in SCIC2 is used to send break characters which were originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0 (10 bit times including the start and stop bits). A longer break of 13 bit times can be enabled by setting BRK13 = 1. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 1 and then write 0 to the SBK bit. This action queues a break character to be sent as soon as the shifter is available. If SBK is still 1 when the queued break moves into the shifter (synchronized to the baud rate clock), an additional break character is queued. If the receiving device is another NXP Semiconductor SCI, the break characters will be received as 0s in all eight data bits and a framing error (FE = 1) occurs.

When idle-line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the TE bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while TE = 0, the SCI transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while TE = 0, set the general-purpose I/O controls so the pin that is shared with TxD is an output driving a logic 1. This ensures that the TxD line will look like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to TE. The length of the break character is affected by the BRK13 and M bits as shown in Table 273.

Table 273. Break character length

BRK13	M	Break character length
0	0	10 bit times
0	1	11 bit times
1	0	13 bit times
1	1	14 bit times

4.13.3.3 Receiver functional description

In this section, the receiver block diagram (Figure 47) is used as a guide for the overall receiver functional description. The data sampling technique used to reconstruct receiver data is then described in more detail. Finally, two variations of the receiver wake-up function are explained.

The receiver input is inverted by setting RXINV = 1. The receiver is enabled by setting the RE bit in SCIC2. Character frames consist of a start bit of logic 0, eight (or nine) data bits (LSB first), and a stop bit of logic 1. For information about 9-bit data mode, refer to Section 4.13.3.5.1, “8- and 9-bit data modes". For the remainder of this discussion, we assume the SCI is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF) status flag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared automatically by a 2-step sequence which is normally satisfied in the course of the user’s program that handles receive data. Refer to Section 4.13.3.4, “Interrupts and status flags" for more details about flag clearing.

4.13.3.3.1 Data sampling technique

The SCI receiver uses a 16× baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The 16× baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for that bit, the noise flag (NF) will be set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately. The receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

4.13.3.3.2 Receiver wake-up operation

Receiver wake-up is a hardware mechanism that allows an SCI receiver to ignore the characters in a message that is intended for a different SCI receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCIC2. When RWU bit is set, the status flags associated with the receiver (with the exception of the idle bit, IDLE, when RWUID bit is set) are inhibited from setting, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force RWU to 0 so all receivers wake up in time to look at the first character(s) of the next message.

4.13.3.3.2.1 Idle-line wake-up

When WAKE = 0, the receiver is configured for idle-line wake-up. In this mode, RWU is cleared automatically when the receiver detects a full character time of the idle-line level. The M control bit selects 8-bit or 9-bit data mode that determines how many bit times of idle are needed to constitute a full character time (10 or 11 bit times because of the start and stop bits).

When RWU is one and RWUID is zero, the idle condition that wakes up the receiver does not set the IDLE flag. The receiver wakes up and waits for the first data character of the next message which will set the RDRF flag and generate an interrupt if enabled. When RWUID is one, any idle condition sets the IDLE flag and generates an interrupt if enabled, regardless of whether RWU is zero or one.

The idle-line type (ILT) control bit selects one of two ways to detect an idle line. When ILT = 0, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When ILT = 1, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

4.13.3.3.2.2 Address-mark wake-up

When WAKE = 1, the receiver is configured for address-mark wake-up. In this mode, RWU is cleared automatically when the receiver detects a logic 1 in the most significant bit of a received character (eighth bit in M = 0 mode and ninth bit in M = 1 mode). Address-mark wake-up allows messages to contain idle characters but requires that the MSB be reserved for use in address frames. The logic 1 MSB of an address frame clears the RWU bit before the stop bit is received and sets the RDRF flag. In this case, the character with the MSB set is received even though the receiver was sleeping during most of this character time.

4.13.3.4 Interrupts and status flags

The SCI system has three separate interrupt vectors to reduce the amount of software needed to isolate the cause of the interrupt. One interrupt vector is associated with the transmitter for TDRE and TC events. Another interrupt vector is associated with the receiver for RDRF, IDLE, RXEDGIF, and LBKDIF events, and a third vector is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can still be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The SCI transmitter has two status flags that optionally can generate hardware interrupt requests. Transmit data register empty (TDRE) indicates when there is room in the transmit data buffer to write another transmit character to SCID. If the transmit interrupt enable (TIE) bit is set, a hardware interrupt will be requested whenever TDRE = 1. Transmit complete (TC) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware interrupt will be requested whenever TC = 1. Instead of hardware interrupts, software polling may be used to monitor the TDRE and TC status flags if the corresponding TIE or TCIE local interrupt masks are 0s.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared by reading SCIS1 while RDRF = 1 and then reading SCID.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, SCIS1 must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the RxD line remains idle for an extended period of time. IDLE is cleared by reading SCIS1 while IDLE = 1 and then reading SCID. After IDLE has been cleared, it cannot become set again until the receiver has received at least one new character and has set RDRF.

If the associated error was detected in the received character that caused RDRF to be set, the error flags — noise flag (NF), framing error (FE), and parity error flag (PF) — get set at the same time as RDRF. These flags are not set in overrun cases.

If RDRF was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (OR) flag gets set instead the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the RxD serial data input pin causes the RXEDGIF flag to set. The RXEDGIF flag is cleared by writing a “1” to it. This function does depend on the receiver being enabled (RE = 1).

4.13.3.5 Additional SCI functions

The following sections describe additional SCI functions.

4.13.3.5.1 8- and 9-bit data modes

The SCI system (transmitter and receiver) can be configured to operate in 9-bit data mode by setting the M control bit in SCIC1. In 9-bit mode, there is a ninth data bit to the left of the MSB of the SCI data register. For the transmit data buffer, this bit is stored in T8 in SCIC3. For the receiver, the ninth bit is held in R8 in SCIC3.

For coherent writes to the transmit data buffer, write to the T8 bit before writing to SCID.

If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to T8 again. When data is transferred from the transmit data buffer to the transmit shifter, the value in T8 is copied at the same time data is transferred from SCID to the shifter.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wake-up so the ninth data bit can serve as the wake-up bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

4.13.3.5.2 Stop mode operation

During all stop modes, clocks to the SCI module are halted. In stop1 and stop2 modes, all SCI register data is lost and must be re-initialized upon recovery from these two stop modes. No SCI module registers are affected in stop3 mode.

The receive input active edge detect circuit is still active in stop3 mode, but not in stop2. An active edge on the receive input brings the CPU out of stop3 mode if the interrupt is not masked (RXEDGIE = 1).

Note, because the clocks are halted, the SCI module will resume operation upon exit from stop (only in stop3 mode). Software should ensure stop mode is not entered while there is a character being transmitted out of or received into the SCI module.

4.13.3.5.3 Loop mode

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the SCI, so it reverts to a general purpose port I/O pin.

4.13.3.5.4 Single-wire operation

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Single-wire mode is used to implement a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general-purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIC3 controls the direction of serial data on the TxD pin. When TXDIR = 0, the TxD pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD pin is an output driven by the transmitter. In single-wire mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters that are sent out by the transmitter.

4.14 Life time counter (LTC)

4.14.1 Introduction

The Life Time Counter is implemented as flexible counter running in both, low power (STOP and SLEEP) and normal modes.

It is based on the ALFCLK clock featuring IRQ and Wake Up capabilities on the Life Time Counter Overflow. The Wake Up on overflow would be indicated in the PCR_SR register WULTCF bit. The Life Time Counter must be set in Normal mode. The Life Time Counter is an up counter.

4.14.2 Memory map and registers

4.14.2.1 Overview

This section provides a detailed description of the memory map and registers.

4.14.2.2 Module memory map

The memory map for the LTC module is in Table 62

Table 274. Module memory map

Offset[232]	Name		7	6	5	4	3	2	1	0
0x38	LTC_CTL (hi)	R	0	0	0	0	0	0	0	0
	Life Time Counter control register	W	LTCIEM							LTCEM
0x39	LTC_CTL (lo)	R	LTCIE	0	0	0	0	0	0	LTCE
	Life Time Counter control register	W								
0x3A	LTC_SR	R	LTCOF	0	0	0	0	0	0	0
	Life Time Counter status register	W	1 = clear							
0x3B	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 274. Module memory map (continued)

Offset[232]	Name		7	6	5	4	3	2	1	0
0x3C	LTC_CNT1 Life Time Counter Register	R	LTC[31:0]							
		W								
		R								
		W								
0x3E	LTC_CNT0 Life Time Counter Register	R								
		W								
		R								
		W								

Notes

[232] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

4.14.2.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

4.14.2.3.1 Life time counter control register (LTC_CTL (hi))

Table 275. Life time counter control register (LTC_CTL (hi))

Offset[233] 0x38

Access: User write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	LTCIEM							LTCEM
Reset	0	0	0	0	0	0	0	0

Notes

[233] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 276. Life time counter control register (LTC_CTL (hi)) - register field descriptions

Field	Description
0 LTCEM	Life Time Counter Enable Mask 0 - writing the LTCE Bit will have no effect 1 - writing the LTCE Bit will be effective
7 LTCIEM	Life Time Counter Interrupt Enable Mask 0 - writing the LTCIE Bit will have no effect 1 - writing the LTCIE Bit will be effective

4.14.2.3.2 Life time counter control register (LTC_CTL (lo))

Table 277. Life time counter control register (LTC_CTL (lo))

Offset[234] 0x39

Access: User read/write

	7	6	5	4	3	2	1	0
R	LTCIE	0	0	0	0	0	0	LTCE
W								
Reset	0	0	0	0	0	0	0	0

Notes

[234] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 278. Life time counter control register (LTC_CTL (lo)) - register field descriptions

Field	Description
0 LTCE	Life Time Counter Enable 1 - Life time counter module enabled. Counter will be incremented with based on the ALFCLK frequency. 0 - Life time counter module disabled. Counter content will remain.[235]
7 LTCIE	Life Time Counter Interrupt Enable 1 - Life time counter overflow will generate an interrupt request. 0 - Life time counter overflow will not generate an interrupt request.

Notes

[235] The first period after enable might be shorted due to the asynchronous clocks.

4.14.2.3.3 Life time counter status register (LTC_SR)

Table 279. Life time counter status register (LTC_SR)

Offset[236] 0x3A

Access: User read/write

	7	6	5	4	3	2	1	0
R	LTCOF	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes

[236] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 280. Life time counter status register (LTC_SR) - register field descriptions

Field	Description
0 LTCOF	Life Time Counter Overflow Flag. Writing 1 will clear the flag. 1 - Life time counter overflow detected. 0 - No life time counter overflow since last clear

4.14.2.3.4 Life time counter register (LTC_CNT1, LTC_CNT0)

Table 281. Life time counter register (LTC_CNT1, LTC_CNT0)

Offset (237),(238) 0x3C, 0x3E

Access: User read/write

	7	6	5	4	3	2	1	0
R	LTC[31:16]							
W								
R								
W								
R	LTC[15:0]							
W								
R								
W								
Reset	0	0	0	0	0	0	0	0

Notes

237. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.
238. Those Registers are 16-Bit access only.

Table 282. Life time counter register (LTC_CNT1, LTC_CNT0) - register field descriptions

Field	Description
0-31 LTC[31:0]	Life Time Counter Register The two 16-Bit words of the 32-Bit Life Time Counter register represent the current counter status. Whenever the microcontroller performs a reading operation on one of the 16Bit registers, the Life Time Counter is stopped until the remaining 16-Bit register is read, to prevent loss of information. After the second part is read, the LTC continues automatically. Write operations should be performed with the Life Time Counter disabled to prevent a loss of data.

4.15 Die to die interface - target

The D2D Interface is the bus interface to the Microcontroller. Access to the MM912_637 analog die is controlled by the D2D Interface module. This section describes the functionality of the die-to-die target block (D2D).

4.15.1 Overview

The D2D is the target for a data transfer from the target to the initiator (MCU). The initiator provides a set of configuration registers and two memory mapped 256 Byte address windows. When writing to a window a transaction is initiated sending a write command, followed by an 8-bit address and the data byte or word is received from the initiator. When reading from a window a transaction is received with the read command, followed by an 8-bit address. The target then responds with the data. The basic idea is that a peripheral located on the MM912_637 analog die, can be addressed like an on-chip peripheral.

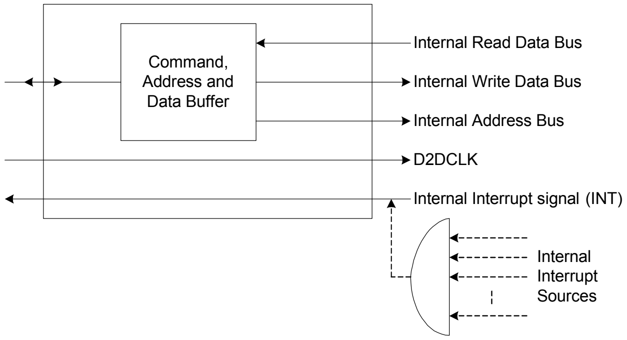


Figure 49. Die to die interface

Features:

- software transparent register access to peripherals on the MM912_637 analog die
- 256 Byte address window
- supports blocking read or write as well as non-blocking write transactions
- 8 bit physical bus width
- automatic synchronization of the target when initiator starts driving the interface clock
- generates transaction and error status as well as EOT acknowledge
- providing single interrupt interface to D2D Initiator

4.15.2 Low power mode operation

The D2D module is disabled in SLEEP and STOP mode. In Stop mode, the D2DINT signal is used to wake-up a powered down MCU after re-enabling the D2D interface. As the MCU could wake-up without the MM912_637 analog die, a special command will be recognized as wake-up event during Stop mode. See Section 4.3, “Analog die - power, clock and resets - PCR".

4.15.2.1 Normal mode

While in Normal, D2DCLK acts as an input only with pull present. D2D[7:0] operates as input/output with pull-down always present. D2DINT acts as an output only.

4.15.2.2 Sleep mode/stop mode

While in Sleep mode, all Interface data pins are pulled down to DGND to reduce power consumption.

4.16 Embedded microcontroller - overview

4.16.1 Introduction

The S12 Central Processing Unit (CPU) offers 128 kB of Flash memory and 6.0 kB of system SRAM, up to eight general purpose I/Os, an on-chip oscillator and clock multiplier, one Serial Peripheral Interface (SPI), an interrupt module, and debug capabilities via the on-chip debug module (DBG), in combination with the Background Debug mode (BDM) interface. Additionally, there is a die-to-die initiator (D2DI) which represents the communication interface to the companion (analog) die.

4.16.2 Features

This section describes the key features of the MM912_637 micro controller die.

4.16.2.1 Chip-level features

On-chip modules available within the family include the following features:

- S12 CPU core (CPU12_V1)
- 128 kByte on-chip flash with ECC
- 4.0 kbyte on-chip data flash with ECC
- 6.0 kbyte on-chip SRAM
- Phase locked loop (IPLL) frequency multiplier with internal filter
- 4.0–16 MHz amplitude controlled Pierce oscillator
- 1.024 MHz internal RC oscillator
- One serial peripheral interface (SPI) module
- On-chip voltage regulator (VREG) for regulation of input supply and all internal voltages
- Die to Die Initiator (D2DI)

4.16.3 Module features

The following sections provide more details of the modules implemented on the MC9S12I128.

4.16.3.1 S12 16-bit central processor unit (CPU)

S12 CPU is a high-speed 16-bit processing unit:

- Full 16-bit data paths supports efficient arithmetic operation and high speed math execution
- Includes many single-byte instructions. This allows much more efficient use of ROM space
- Extensive set of indexed addressing capabilities, including:
  - Using the stack pointer as an indexing register in all indexed operations
  - Using the program counter as an indexing register in all but auto increment/decrement mode
  - Accumulator offsets using A, B, or D accumulators
  - Automatic index predecrement, preincrement, postdecrement, and postincrement (by –8 to +8)

4.16.3.2 On-chip flash with ECC

On-chip flash memory on the MM912_637 features the following:

- 128 kbyte of program flash memory
  - 32 data bits plus 7 syndrome ECC (Error Correction Code) bits allow single bit error correction and double fault detection
  - Erase sector size 512 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads
  - Protection scheme to prevent accidental program or erase
- 4.0 kbyte data flash memory
  - 16 data bits plus 6 syndrome ECC (Error Correction Code) bits allow single bit error correction and double-bit error detection
  - Erase sector size 256 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads

4.16.3.3 On-chip SRAM

- 6.0 kBytes of general purpose RAM

4.16.3.4 Main external oscillator (XOSC)

- Loop controlled Pierce oscillator using a 4.0 MHz to 16 MHz crystal or resonator
  - Current gain control on amplitude output
  - Signal with low harmonic distortion
  - Low power
  - Good noise immunity

- — Eliminates need for external current limiting resistor
- — Transconductance sized for optimum start-up margin for typical crystals

4.16.3.5 Internal RC oscillator (IRC)

- Trimmable internal reference clock
  - Frequency: 1.024 MHz

4.16.3.6 Internal phase-locked-loop (IPLL)

- Phase-locked-loop clock frequency multiplier
  - No external components required
  - Reference divider and multiplier allow large variety of clock rates
  - Automatic bandwidth control mode for low-jitter operation
  - Automatic frequency lock detector
  - Configurable option to spread spectrum for reduced EMC radiation (frequency modulation)
  - Reference clock sources:
    - External 4.0–16 MHz resonator/crystal (XOSC)
    - Internal 1.024 MHz RC oscillator (IRC)

4.16.3.7 System integrity support

- Power-on reset (POR)
- System reset generation
- Illegal address detection with reset
- Low voltage detection with interrupt or reset
- Real time interrupt (RTI)
- Computer operating properly (COP) watchdog
  - Configurable as window COP for enhanced failure detection
  - Initialized out of reset using option bits located in flash memory
- Clock monitor supervising the correct function of the oscillator

4.16.3.8 Serial peripheral interface module (SPI)

- Configurable 8- or 16-bit data size
- Full duplex or single-wire bidirectional
- Double buffered transmit and receive
- Master or slave mode
- MSB-first or LSB-first shifting
- Serial clock phase and polarity options

4.16.3.9 On-chip voltage regulator (VREG)

- Linear voltage regulator with bandgap reference
- Low voltage detect (LVD) with low voltage interrupt (LVI)
- Power-on reset (POR) circuit
- Low voltage reset (LVR)

4.16.3.10 Background debug (BDM)

- Non-intrusive memory access commands
- Supports in-circuit programming of on-chip nonvolatile memory

4.16.3.11 Debugger (DBG)

- Trace buffer with depth of 64 entries
- Three comparators (A, B and C)
  - — Comparator A compares the full address bus and full 16-bit data bus
  - — Exact address or address range comparisons
- Two types of comparator matches
  - — Tagged: This matches just before a specific instruction begins execution
  - — Force: This is valid on the first instruction boundary after a match occurs
- Four trace modes
- Four stage state sequencer

4.16.3.12 Die to die initiator (D2DI)

- Up to 2.0 Mbyte/s data rate
- Configurable 4-bit or 8-bit wide data path

4.16.4 Block diagram

Figure 50 shows a block diagram of the MC9S12I128 device.

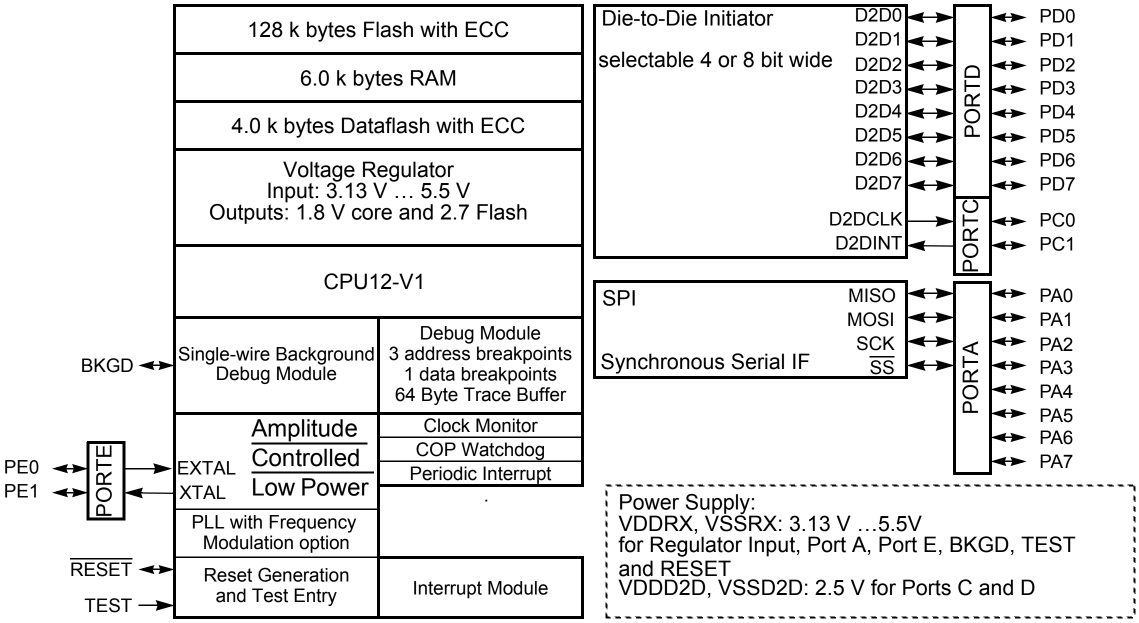


Figure 50. MC9S12I128 block diagram

4.16.5 Device memory map

Table 283 shows the device register memory map.

Table 283. Device register memory map

Address	Module	Size (Bytes)
0x0000–0x0009	PIM (port integration module)	10
0x000A–0x000B	MMC (memory map control)	2
0x000C–0x000D	PIM (port integration module)	2
0x000E–0x000F	Reserved	2

Table 283. Device register memory map (continued)

Address	Module	Size (Bytes)
0x0010–0x0015	MMC (memory map control)	8
0x0016–0x0019	Reserved	2
0x001A–0x001B	Device ID register	2
0x001C–0x001E	Reserved	4
0x001F	INT (interrupt module)	1
0x0020–0x002F	DBG (debug module)	16
0x0030–0x0033	Reserved	4
0x0034–0x003F	CPMU (clock and power management)	12
0x0040–0x00D7	Reserved	152
0x00D8–0x00DF	D2DI (die 2 die initiator)	8
0x00E0–0x00E7	Reserved	32
0x00E8–0x00EF	SPI (serial peripheral interface)	8
0x00F0–0x00FF	Reserved	32
0x0100–0x0113	FTMRC control registers	20
0x0114–0x011F	Reserved	12
0x0120–0x017F	PIM (port integration module)	96
0x0180–0x01EF	Reserved	112
0x01F0–0x01FC	CPMU (clock and power management)	13
0x01FD–0x01FF	Reserved	3
0x0200-0x02FF	D2DI (die 2 die initiator, blocking access window)	256
0x0300–0x03FF	D2DI (die 2 die initiator, non-blocking write window)	256

NOTE

Reserved register space shown in Table 283 is not allocated to any module. This register space is reserved for future use. Writing to these locations have no effect. Read access to these locations returns zero.

Figure 51 shows MM912_637 CPU and BDM local address translation to the global memory map. It indicates also the location of the internal resources in the memory map. The whole 256 k global memory space is visible through the P-Flash window located in the 64 k local memory map located at 0x8000 - 0xBFFF using the PPAGE register.

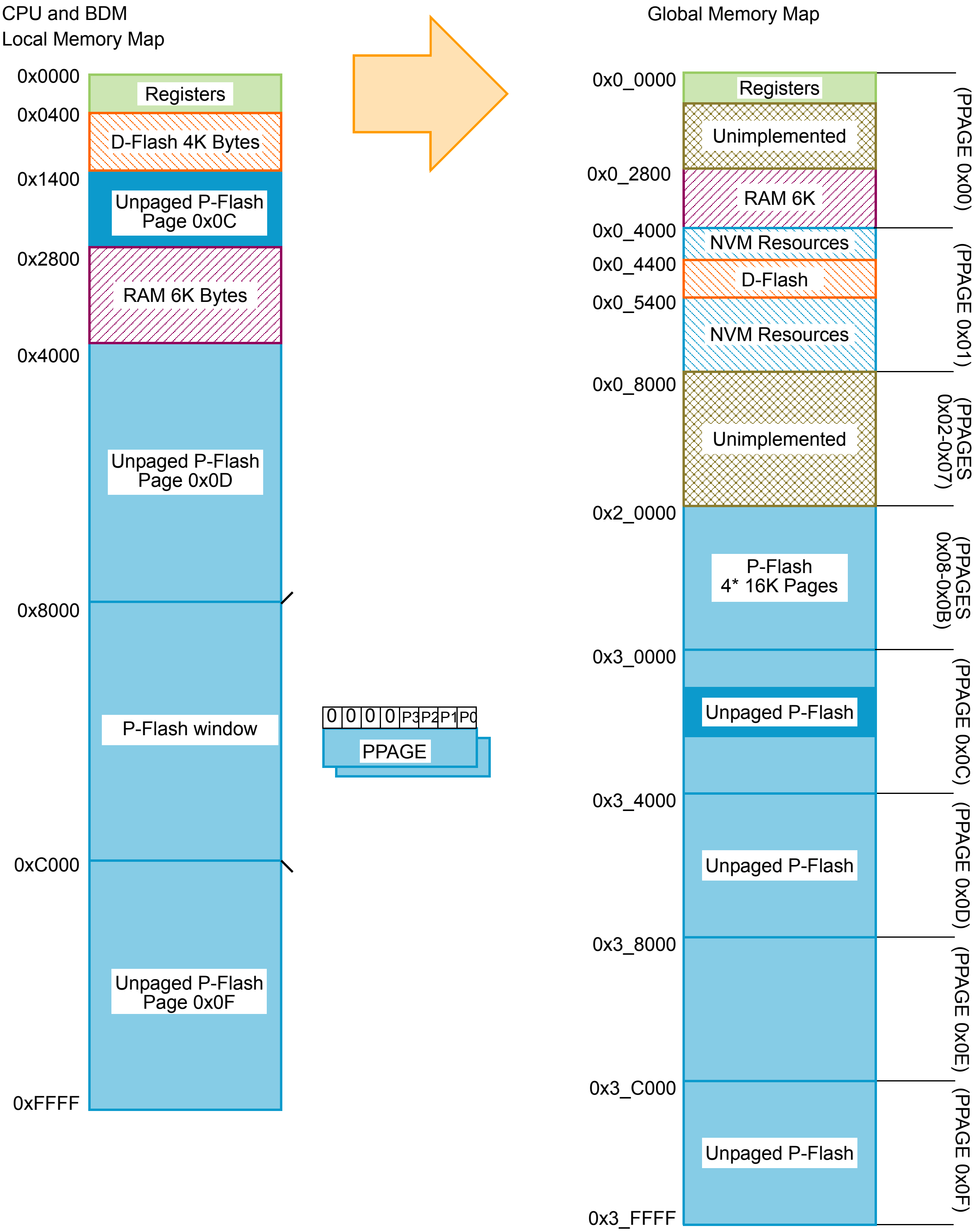


Figure 51. MC9S12I128 global memory map

4.16.6 Part ID assignments

The part ID is located in two 8-bit registers PARTIDH and PARTIDL (addresses 0x001A and 0x001B). The read-only value is a unique part ID for each revision of the chip. Table 284 shows the assigned part ID number and Mask Set number.

The Version ID in Table 284 is a word located in a flash information row. The version ID number indicates a specific version of internal NVM controller.

Table 284. Assigned part ID numbers

Device	Mask set number	Part ID[239]	Version ID
MM912_637	0M96X	0x3880	0x0000

Notes

239. The coding is as follows:
Bit 15-12: Major family identifier
Bit 11-6: Minor family identifier
Bit 5-4: Major mask set revision number including FAB transfers
Bit 3-0: Minor — non full — mask set revision

4.16.7 System clock description

Refer to Section 4.23, “S12 clock, reset, and power management unit (9S12I128PIMV1)" for the system clock description.

4.16.8 Modes of operation

The MCU can operate in different modes. These are described in Section 4.16.8.1, “Chip configuration summary".

The MCU can operate in different power modes to facilitate power saving when full system performance is not required. These are described in Section 4.16.8.2, “Low power operation". Some modules feature a software programmable option to freeze the module status while the background debug module is active to facilitate debugging.

4.16.8.1 Chip configuration summary

The different modes and the security state of the MCU affect the debug features (enabled or disabled). The operating mode out of reset is determined by the state of the MODC signal during reset (see Table 285). The MODC bit in the MODE register shows the current operating mode and provides limited mode switching during operation. The state of the MODC signal is latched into this bit on the rising edge of $\overline{\text{RESET}}$.

Table 285. Chip modes

Chip modes	MODC
Normal single chip	1
Special single chip	0

4.16.8.1.1 Normal single-chip mode

This mode is intended for normal device operation. The opcode from the on-chip memory is being executed after reset (requires the reset vector to be programmed correctly). The processor program is executed from internal memory.

4.16.8.1.2 Special single-chip mode

This mode is used for debugging single-chip operation, boot-strapping, or security related operations. The background debug module BDM is active in this mode. The CPU executes a monitor program located in an on-chip ROM. BDM firmware waits for additional serial commands through the BKGD pin.

4.16.8.2 Low power operation

The MM912_637 has two static low-power modes Pseudo Stop and Stop mode. For a detailed description refer to the S12CPMU section.

4.16.9 Security

The MCU security mechanism prevents unauthorized access to the Flash memory. Refer to Section 4.21, “MCU - security (S12XS9S12I128PIMV1V2)", Section 4.22.4.1, “Security", and Section 4.25.5, “Security".

4.16.10 Resets and interrupts

Consult the S12 CPU manual and the S12SINT section for information on exception processing.

4.16.10.1 Resets

Table 286 lists all Reset sources and the vector locations. Resets are explained in detail in Section 4.23, “S12 clock, reset, and power management unit (9S12I128PIMV1)".

Table 286. Reset sources and vector locations

Vector address	Reset source	CCR mask	Local enable
$FFFE	Power-On Reset (POR)	None	None
$FFFE	Low Voltage Reset (LVR)	None	None
$FFFE	External pin $\overline{\text{RESET}}$	None	None
$FFFE	Illegal Address Reset	None	None
$FFFC	Clock monitor reset	None	OSCE Bit in CPMUOSC register
$FFFA	COP watchdog reset	None	CR[2:0] in CPMUCOP register

4.16.10.2 Interrupt vectors

Table 287 lists all interrupt sources and vectors in the default order of priority. The interrupt module (see Section 4.18, “MCU - interrupt module (S12S9S12I128PIMV1V1)") provides an interrupt vector base register (IVBR) to relocate the vectors.

Table 287. Interrupt vector locations (Sheet 2 of 2)

Vector Address[240]	Interrupt Source	CCR Mask	Local Enable	Wake-up from STOP	Wake-up from WAIT
Vector base + $F8	Unimplemented instruction trap	None	None	-	-
Vector base+ $F6	SWI	None	None	-	-
Vector base+ $F4	D2DI Error Interrupt	X Bit	None	Yes	Yes
Vector base+ $F2	D2DI External Interrupt	I bit	D2DCTL (D2DIE)	Yes	Yes
Vector base+ $F0	RTI timeout interrupt	I bit	CPMUINT (RTIE)	3.22.6 Interrupts	
Vector base + $EE to Vector base + $DA	Reserved				
Vector base + $D8	SPI	I bit	SPICR1 (SPIE, SPTIE)	No	Yes
Vector base + $D6 to Vector base + $CA	Reserved				
Vector base + $C8	Oscillator status interrupt	I bit	CPMUINT (OSCIE)	No	No
Vector base + $C6	PLL lock interrupt	I bit	CPMUINT (LOCKIE)	No	No
Vector base + $C4 to Vector base + $BC	Reserved				
Vector base + $BA	FLASH error	I bit	FERCNFG (SFDIE, DFDIE)	No	No

Table 287. Interrupt vector locations (Sheet 2 of 2)

Vector Address[240]	Interrupt Source	CCR Mask	Local Enable	Wake-up from STOP	Wake-up from WAIT
Vector base + $B8	FLASH command	I bit	FCNFG (CCIE)	No	Yes
Vector base + $B6 to Vector base + $8C	Reserved				
Vector base + $8A	Low-voltage interrupt (LVI)	I bit	CPMUCTRL (LVIE)	No	Yes
Vector base + $88 to Vector base + $82	Reserved				
Vector base + $80	Spurious interrupt	—	None	-	-

Notes

[240] 16-bit vector address based

4.16.10.3 Effects of reset

When a reset occurs, MCU registers and control bits are initialized. Refer to the respective block sections for register reset states. On each reset, the Flash module executes a reset sequence to load Flash configuration registers.

4.16.10.3.1 Flash configuration reset sequence phase

The Flash module will hold CPU on each reset activity while loading Flash module registers from the Flash memory. If double faults are detected in the reset phase, Flash module protection and security may be active on leaving reset. This is explained in more detail in the Flash module, Section 4.25.6, “Initialization".

4.16.10.3.2 Reset while flash command active

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

4.16.10.3.3 I/O pins

Refer to the PIM section for reset configurations of all peripheral module ports.

4.16.10.3.4 Memory

The RAM arrays are not initialized out of reset.

4.16.11 COP configuration

The COP timeout rate bits CR[2:0] and the WCOP bit in the CPMUCOP register at address 0x003C are loaded from the Flash register FOPT. See Table 288 and Table 289 for coding. The FOPT register is loaded from the Flash configuration field byte at global address 0x3_FF0E during the reset sequence.

Table 288. Initial COP rate configuration

NV[2:0] in FOPT register	CR[2:0] in COPCTL register
000	111
001	110
010	101
011	100
100	011

Table 288. Initial COP rate configuration (continued)

NV[2:0] in FOPT register	CR[2:0] in COPCTL register
101	010
110	001
111	000

Table 289. Initial WCOP configuration

NV[3] in FOPT register	WCOP in COPCTL register
1	0
0	1

4.17 MCU - port integration module (9S12I128PIMV1)

4.17.1 Introduction

The Port Integration Module (PIM) establishes the interface between the S12I128 peripheral modules SPI and Die-To-Die Interface module (D2DI) to the I/O pins of the MCU. All port A and port E pins support general purpose I/O functionality, if not in use with other functions. The PIM controls the signal prioritization and multiplexing on shared pins.

4.17.1.1 Overview

Figure 52 is a block diagram of the Port Integration Module.

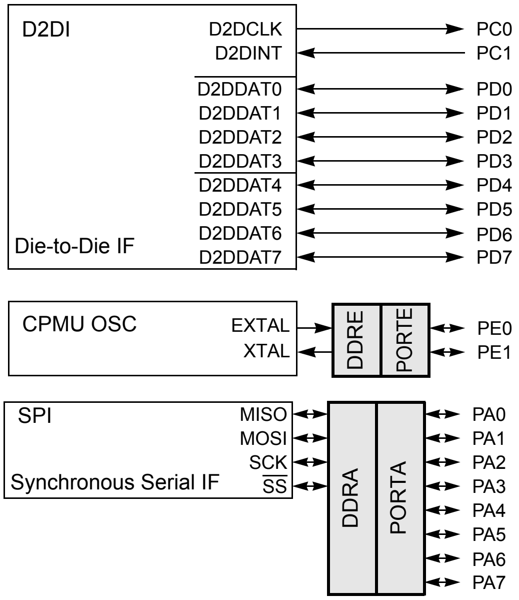


Figure 52. Port integration module - block diagram

4.17.1.2 Features

- 8-pin port A associated with the SPI module
- 2-pin port C used as D2DI clock output and D2DI interrupt input
- 8-pin port D used as 8 or 4 bit data I/O for the D2DI module
- 2-pin port E associated with the CPMU OSC module
- GPIO function shared on port A and E pins

- Pull-down devices on PC1 and PD7-0 if used as D2DI inputs
- Reduced drive capability on PC0 and PD7-0 on per pin basis

The Port Integration Module includes these distinctive registers:

- Data registers for ports A and E when used as general purpose I/O
- Data direction registers for ports A and E when used as general purpose I/O
- Port input register on ports A and E
- Reduced drive register on port C and D

4.17.2 External signal description

This section lists and describes the signals that do connect off-chip. Table 290 shows all the pins and their functions that are controlled by the Port Integration Module.

NOTE

If there is more than one function associated with a pin, the priority is indicated by the position in the table from top (highest priority) to bottom (lowest priority).

Table 290. Pin functions and priorities

Port	Pin name	Pin function & priority	I/O	Description	Pin function after reset
A	PA7	GPIO	I/O	General-purpose I/O	GPI
	PA6	GPIO	I/O	General-purpose I/O	
	PA5	GPIO	I/O	General-purpose I/O	
	PA4	GPIO	I/O	General-purpose I/O	
	PA3	SS	I/O	Serial Peripheral Interface 0 slave select output in master mode, input in slave or master mode	
		GPIO	I/O	General-purpose I/O	
	PA2	SCK	I/O	Serial Peripheral Interface 0 serial clock pin	
		GPIO	I/O	General-purpose I/O	
	PA1	MOSI	I/O	Serial Peripheral Interface 0 master out/slave in pin	
		GPIO	I/O	General-purpose I/O	
	PA0	MISO	I/O	Serial Peripheral Interface 0 master in/slave out pin	
		GPIO	I/O	General-purpose I/O	
E	PE1	XTAL	-	CPMU OSC XTAL pin	GPI
		GPIO	I/O	General-purpose I/O	
	PE0	EXTAL	-	CPMU OSC EXTAL pin	
		GPIO	I/O	General-purpose I/O	

4.17.3 Memory map and register definition

This section provides a detailed description of all Port Integration Module registers.

4.17.3.1 Memory map

Table 291. PIM register summary

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 PORTA	R	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	W								
0x0001 PORTE	R	0	0	0	0	0	0	PE1	PE0
	W								
0x0002 DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
	W								
0x0003 DDRE	R	0	0	0	0	0	0	DDRE1	DDRE0
	W								
0x0004- 0x0009 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x000C PUCR	R	0	BKPUE	0	0	0	0	PDPEE	0
	W								
0x000D RDRIV	R	0	0	0	0	RDPD	RDPC	0	0
	W								
0x0120 PTIA	R	PTIA7	PTIA6	PTIA5	PTIA4	PTIA3	PTIA2	PTIA1	PTIA0
	W								
0x0121 PTIE	R	0	0	0	0	0	0	PTIE1	PTIE0
	W								
0x0122- 0x017F Reserved	R	0	0	0	0	0	0	0	0
	W								

□ = Unimplemented or Reserved

4.17.3.2 Port A data register (PORTA)

Table 292. Port A data register (PORTA)

Address 0x0000 Access: User read/write

	7	6	5	4	3	2	1	0
R W	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
SPI Function	—	—	—	—	SS	SCK	MOSI	MISO
Reset	0	0	0	0	0	0	0	0

Read: Anytime.
Write: Anytime.

Table 293. PORTA register field descriptions

Field	Description
7–4 PA	**Port A general purpose input/output data**—Data Register In / output mode the register bit is driven to the pin. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered and synchronized pin input state is read.
3 PA	**Port A general purpose input/output data**—Data Register, SPI $\overline{SS}$ input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. The SPI function takes precedence over the general purpose I/O function if enabled.
2 PA	**Port A general purpose input/output data**—Data Register, SPI SCK input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. The SPI function takes precedence over the general purpose I/O function if enabled.
1 PA	**Port A general purpose input/output data**—Data Register, SPI MOSI input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. The SPI function takes precedence over the general purpose I/O function if enabled.
0 PA	**Port A general purpose input/output data**—Data Register, SPI MISO input/output When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. The SPI function takes precedence over the general purpose I/O function if enabled.

4.17.3.3 Port E data register (PORTE)

Table 294. Port E data register (PORTE)

Address 0x0001 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PE1	PE0
W								
CPMU OSC Function	—	—	—	—	—	—	XTAL	EXTAL
Reset	0	0	0	0	0	0	0	0

Read: Anytime.
Write: Anytime.

Table 295. PORTE register field descriptions

Field	Description
1 PE	**Port E general purpose input/output data**—Data Register, CPMU OSC XTAL signal When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. The CPMU OSC function takes precedence over the general purpose I/O function if enabled.
0 PE	**Port E general purpose input/output data**—Data Register, CPMU OSC EXTAL signal When not used with the alternative function, this pin can be used as general purpose I/O. In general purpose output mode the register bit is driven to the pin. If the associated data direction bit of this pin is set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read. The CPMU OSC function takes precedence over the general purpose I/O function if enabled.

4.17.3.4 Port A data direction register (DDRA)

Table 296. Port A data direction register (DDRA)

Address 0x0002 Access: User read/write

	7	6	5	4	3	2	1	0
R W	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
Reset	0	0	0	0	0	0	0	0

Read: Anytime.
Write: Anytime.

Table 297. DDRA register field descriptions

Field	Description
7–4 DDRA	**Port A Data Direction—** This bit determines whether the associated pin is an input or output. 1 Associated pin is configured as output. 0 Associated pin is configured as input.
3–0 DDRA	**Port A Data Direction—** This bit determines whether the associated pin is an input or output. Depending on the configuration of the enabled SPI the I/O state will be forced to input or output. In this case, the data direction bits will not change. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

4.17.3.5 Port E data direction register (DDRE)

Table 298. Port E data direction register (DDRE)

Address 0x0003 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	DDRE1	DDRE0
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime.
Write: Anytime.

Table 299. DDRE register field descriptions

Field	Description
1–0 DDRE	**Port E Data Direction—** This bit determines whether the associated pin is an input or output. The enabled CPMU OSC function connects the associated pins directly to the oscillator module. In this case, the data direction bits will not change. 1 Associated pin is configured as output. 0 Associated pin is configured as input.

4.17.3.6 Pull-up control register (PUCR)

Table 300. Pull control register (PUCR)

Address 0x000C

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	BKPUE	0	0	0	0	PDPEE	0
W								
Reset	0	1	0	0	0	0	1	0

Read: Anytime.
Write: Anytime.

Table 301. PUCR register field descriptions

Field	Description
6 BKPUE	BKGD pin pull-up Enable—Enable pull-up devices on BKGD pin This bit configures whether a pull-up device is activated, if the pin is used as input. This bit has no effect if the pin is used as output. Out of reset the pull-up device is enabled. 1 Pull-up device enabled. 0 Pull-up device disabled.
1 PDPEE	Pull-down Port E Enable—Enable pull-down devices on all Port E input pins This bit configures whether pull-down devices are activated, if the pins are used as inputs. This bit has no effect if the pins are used as outputs. Out of reset the pull-down devices are enabled. If the CPMU OSC function is active, the pull-down devices are disabled. In this case, the register bit will not change. 1 Pull-down devices enabled. 0 Pull-down devices disabled.

4.17.3.7 Reduced drive register (RDRIV)

Table 302. Reduced drive register (RDRIV)

Address 0x000D

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	RDPD	RDPC	0	0
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime.
Write: Anytime.

Table 303. RDRIV register field descriptions

Field	Description
3 RDPD	**Port D reduced drive—Select reduced drive for output pins** This bit configures the drive strength of output pins as either full or reduced. If a pin is used as input, this bit has no effect. 1 Reduced drive selected (1/5 of the full drive strength) 0 Full drive strength enabled
2 RDPC	**Port C reduced drive—Select reduced drive for D2DCLK output pin** This bit configures the drive strength of D2DCLK output pin as either full or reduced. 1 Reduced drive selected (1/5 of the full drive strength) 0 Full drive strength enabled

4.17.3.8 Port A input register (PTIA)

Table 304. Port A input register (PTIA)

Address 0x0120 Access: User read

	7	6	5	4	3	2	1	0
R	PTIA7	PTIA6	PTIA5	PTIA4	PTIA3	PTIA2	PTIA1	PTIA0
W								
Reset[241]	u	u	u	u	u	u	u	u

Notes

241. u = Unaffected by reset

Read: Anytime.
Write: Unimplemented. Writing to this register has no effect.

Table 305. PTIA register field descriptions

Field	Description
7–0 PTIA	**Port A input data—** A read always returns the buffered input state of the associated pin. It can be used to detect overload or short-circuit conditions on output pins.

4.17.3.9 Port E input register (PTIE)

Table 306. Port E input register (PTIE)

Address 0x0121 Access: User read

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	PTIE1	PTIE0
W								
Reset[242]	u	u	u	u	u	u	u	u

Notes

242. u = Unaffected by reset

Read: Anytime.
Write: Unimplemented. Writing to this register has no effect.

Table 307. PTIE register field descriptions

Field	Description
1–0 PTIE	**Port E input data—** A read always returns the buffered input state of the associated pin. It can be used to detect overload or short-circuit conditions on output pins.

4.17.4 Functional description

4.17.4.1 Registers

4.17.4.1.1 Data register (PORTx)

This register holds the value driven out to the pin, if the pin is used as a general purpose I/O.

Writing to this register has only an effect on the pin, if the pin is used as general purpose output. When reading this address, the buffered and synchronized state of the pin is returned, if the associated data direction register bit is set to “0”.

If the data direction register bits are set to logic level “1”, the contents of the data register is returned. This is independent of any other configuration (Figure 53).

4.17.4.1.2 Data direction register (DDRx)

This register defines whether the pin is used as an input or an output.

If a peripheral module controls the pin the contents of the data direction register is ignored (Figure 53).

4.17.4.1.3 Input register (PTIx)

This is a read-only register and always returns the buffered and synchronized state of the pin (Figure 53).

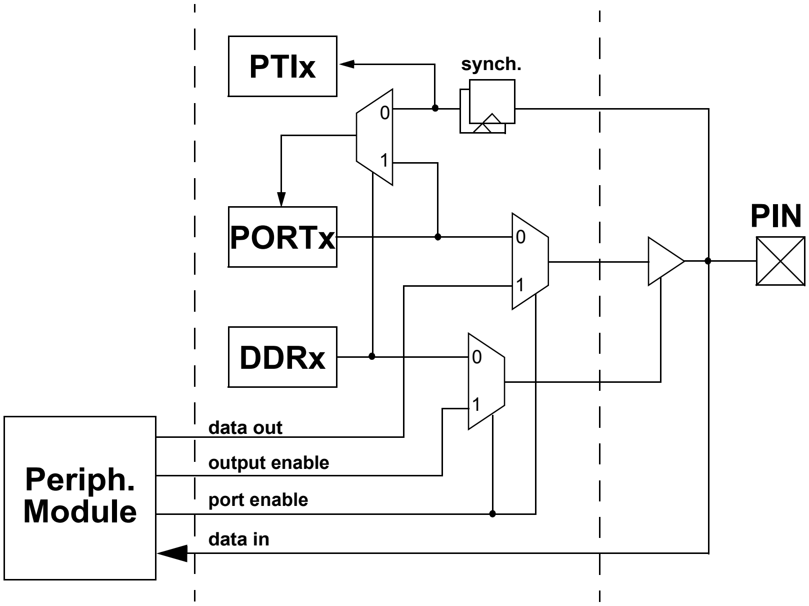


Figure 53. Illustration of I/O pin functionality

4.17.4.1.4 Reduced drive register (RDRIV)

If the pin is used as an output, this register allows the configuration of the drive strength.

4.17.4.1.5 Pull device enable register (PUCR)

This register turns on a pull-up or pull-down device. It becomes active only if the pin is used as an input.

4.17.4.2 Ports

4.17.4.2.1 Port A

This port is associated with the SPI. Port A pins PA7-0 can be used for general purpose I/O and PA3-0 also with the SPI subsystem.

4.17.4.2.2 Port E

This port is associated with the CPMU OSC. Port E pins PE1-0 can be used for general purpose or with the CPMU OSC module.

4.17.5 Initialization information

4.17.5.1 Port data and data direction register writes

Writing PTx and DDRx in a word access is not recommended. When changing the register pins from inputs to outputs, the data may have extra transitions during the write access. Initialize the port data register before enabling the outputs.

4.18 MCU - interrupt module (S12S9S12I128PIMV1V1)

4.18.1 Introduction

The 9S12I128PIMV1 module decodes the priority of all system exception requests and provides the applicable vector for processing the exception to the CPU. The 9S12I128PIMV1 module supports:

- I bit and X bit maskable interrupt requests
- A non-maskable unimplemented op-code trap
- A non-maskable software interrupt (SWI) or background debug mode request
- Three system reset vector requests
- A spurious interrupt vector

Each of the I bit maskable interrupt requests is assigned to a fixed priority level.

4.18.1.1 Glossary

Table 308 contains terms and abbreviations used in the document.

Table 308. Terminology

Term	Meaning
CCR	Condition Code Register (in the CPU)
ISR	Interrupt Service Routine
MCU	Micro-controller Unit

4.18.1.2 Features

- Interrupt vector base register (IVBR)
- One spurious interrupt vector (at address vector base[243] + 0x0080).
- 2–58 I bit maskable interrupt vector requests (at addresses vector base + 0x0082–0x00F2)
- I bit maskable interrupts can be nested
- One X bit maskable interrupt vector request (at address vector base + 0x00F4)
- One non-maskable software interrupt request (SWI) or background debug mode vector request (at address vector base + 0x00F6)
- One non-maskable unimplemented op-code trap (TRAP) vector (at address vector base + 0x00F8)
- Three system reset vectors (at addresses 0xFFFA–0xFFFE)
- Determines the highest priority interrupt vector requests, drives the vector to the bus on CPU request
- Wakes up the system from stop or wait mode when an appropriate interrupt request occurs

Notes

[243] The vector base is a 16-bit address which is accumulated from the contents of the interrupt vector base register (IVBR, used as upper byte) and 0x00 (used as lower byte).

4.18.1.3 Modes of operation

- Run mode
  This is the basic mode of operation.
- Wait mode
  In Wait mode, the clock to the 9S12I128PIMV1 module is disabled. The 9S12I128PIMV1 module is however capable of waking up the CPU from Wait mode, if an interrupt occurs. Refer to Section 4.18.5.3, “Wake-up from stop or wait mode"” for details
- Stop mode
  In Stop mode, the clock to the 9S12I128PIMV1 module is disabled. The 9S12I128PIMV1 module is however capable of waking up the CPU from Stop mode, if an interrupt occurs. Refer to Section 4.18.5.3, “Wake-up from stop or wait mode"” for details
- Freeze mode (BDM active)
  In Freeze mode (BDM active), the interrupt vector base register is overridden internally. Refer to Section 4.18.3.1.1, “Interrupt vector base register (IVBR)"” for details

4.18.1.4 Block diagram

Figure 54 shows a block diagram of the 9S12I128PIMV1 module.

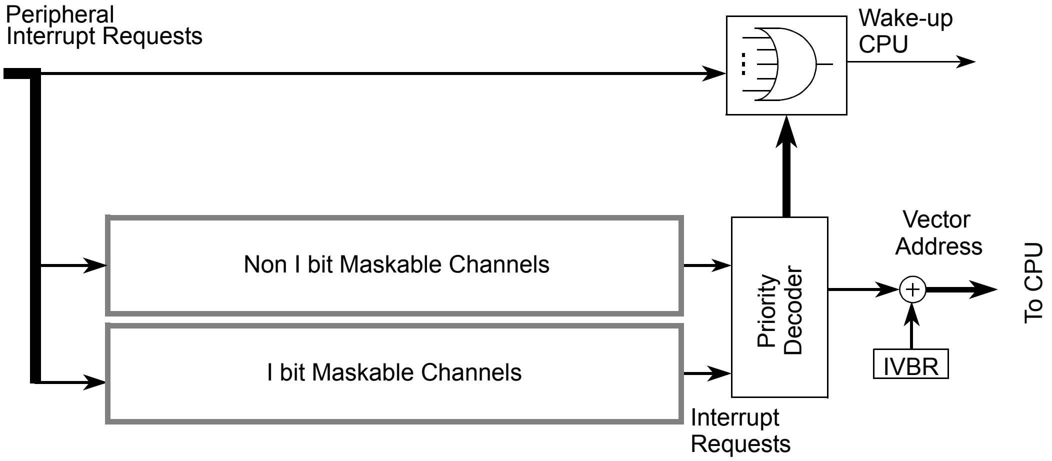


Figure 54. 9S12I128PIMV1 block diagram

4.18.2 External signal description

The 9S12I128PIMV1 module has no external signals.

4.18.3 Memory map and register definition

This section provides a detailed description of all registers accessible in the 9S12I128PIMV1 module.

4.18.3.1 Register descriptions

This section describes in address order all the 9S12I128PIMV1 registers and their individual bits.

4.18.3.1.1 Interrupt vector base register (IVBR)

Table 309. Interrupt vector base register (IVBR)

Address: 0x001F

	7	6	5	4	3	2	1	0
R W	IVB_ADDR[7:0]							
Reset	1	1	1	1	1	1	1	1

Read: Anytime.
Write: Anytime.

Table 310. IVBR field descriptions

Field	Description
7–0 IVB_ADDR[7:0]	**Interrupt Vector Base Address Bits** — These bits represent the upper byte of all vector addresses. Out of reset, these bits are set to 0xFF (i.e., vectors are located at 0xFF80–0xFFFE) to ensure compatibility to HCS12. **Note:** A system reset will initialize the interrupt vector base register with "0xFF" before it is used to determine the reset vector address. Therefore, changing the IVBR has no effect on the location of the three reset vectors (0xFFFA–0xFFFE). **Note:** If the BDM is active (i.e., the CPU is in the process of executing BDM firmware code), the contents of IVBR are ignored and the upper byte of the vector address is fixed as "0xFF". This is done to enable handling of all non-maskable interrupts in the BDM firmware.

4.18.4 Functional description

The 9S12I128PIMV1 module processes all exception requests to be serviced by the CPU module. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the following subsections.

4.18.4.1 S12S exception requests

The CPU handles both reset requests and interrupt requests. A priority decoder is used to evaluate the priority of pending interrupt requests.

4.18.4.2 Interrupt prioritization

The 9S12I128PIMV1 module contains a priority decoder to determine the priority for all interrupt requests pending for the CPU. If more than one interrupt request is pending, the interrupt request with the higher vector address wins the prioritization.

The following conditions must be met for an I bit maskable interrupt request to be processed.

1. The local interrupt enabled bit in the peripheral module must be set.
2. The I bit in the condition code register (CCR) of the CPU must be cleared.
3. There is no SWI, TRAP, or X bit maskable request pending.

NOTE

All non I bit maskable interrupt requests always have higher priority than the I bit maskable interrupt requests. If the X bit in the CCR is cleared, it is possible to interrupt an I bit maskable interrupt by an X bit maskable interrupt. It is possible to nest non maskable interrupt requests, e.g., by nesting SWI or TRAP calls.

Since an interrupt vector is only supplied at the time when the CPU requests it, it is possible that a higher priority interrupt request could override the original interrupt request that caused the CPU to request the vector. In this case, the CPU will receive the highest priority vector and the system will process this interrupt request first, before the original interrupt request is processed.

If the interrupt source is unknown (for example, in the case where an interrupt request becomes inactive after the interrupt has been recognized, but prior to the CPU vector request), the vector address supplied to the CPU will default to that of the spurious interrupt vector.

NOTE

Care must be taken to ensure that all interrupt requests remain active until the system begins execution of the applicable service routine; otherwise, the exception request may not get processed at all or the result may be a spurious interrupt request (vector at address (vector base + 0x0080)).

4.18.4.3 Reset exception requests

The 9S12I128PIMV1 module supports three system reset exception request types (Refer to the Clock and Reset generator module for details):

1. Pin reset, power-on reset or illegal address reset, low voltage reset (if applicable)
2. Clock monitor reset request
3. COP watchdog reset request

4.18.4.4 Exception priority

The priority (from highest to lowest) and address of all exception vectors issued by the 9S12I128PIMV1 module upon request by the CPU is shown in Table 311.

Table 311. Exception vector map and priority

Vector address[244]	Source
0xFFFE	Pin reset, power-on reset, illegal address reset, low voltage reset (if applicable)
0xFFFC	Clock monitor reset
0xFFFA	COP watchdog reset
(Vector base + 0x00F8)	Unimplemented opcode trap
(Vector base + 0x00F6)	Software interrupt instruction (SWI) or BDM vector request
(Vector base + 0x00F4)	X bit maskable interrupt request (XIRQ or D2D error interrupt)[245]
(Vector base + 0x00F2)	IRQ or D2D interrupt request[246]
(Vector base + 0x00F0–0x0082)	Device specific I bit maskable interrupt sources (priority determined by the low byte of the vector address, in descending order)
(Vector base + 0x0080)	Spurious interrupt

Notes

[244] 16-bit vector address based

[245] D2D error interrupt on MCUs featuring a D2D initiator module, otherwise XIRQ pin interrupt

[246] D2D interrupt on MCUs featuring a D2D initiator module, otherwise IRQ pin interrupt

4.18.5 Initialization/application information

4.18.5.1 Initialization

After a system reset, the software should:

1. Initialize the interrupt vector base register, if the interrupt vector table is not located at the default location (0xFF80–0xFFF9).
2. Enable I bit maskable interrupts by clearing the I bit in the CCR.
3. Enable the X bit maskable interrupt by clearing the X bit in the CCR.

4.18.5.2 Interrupt nesting

The interrupt request scheme makes it possible to nest I bit maskable interrupt requests handled by the CPU.

I bit maskable interrupt requests can be interrupted by an interrupt request with a higher priority.

I bit maskable interrupt requests cannot be interrupted by other I bit maskable interrupt requests per default. In order to make an interrupt service routine (ISR) interruptible, the ISR must explicitly clear the I bit in the CCR (CLI). After clearing the I bit, other I bit maskable interrupt requests can interrupt the current ISR.

An ISR of an interruptible I bit maskable interrupt request could basically look like this:

1. Service interrupt, e.g., clear interrupt flags, copy data, etc.
2. Clear I bit in the CCR by executing the instruction CLI (thus allowing other I bit maskable interrupt requests)
3. Process data
4. Return from interrupt by executing the instruction RTI

4.18.5.3 Wake-up from stop or wait mode

4.18.5.3.1 CPU wake-up from stop or wait mode

Every I bit maskable interrupt request is capable of waking the MCU from Stop or Wait mode. To determine whether an I bit maskable interrupts is qualified to wake-up the CPU, the same conditions as in normal run mode are applied during Stop or Wait mode:

If the I bit in the CCR is set, all I bit maskable interrupts are masked from waking up the MCU. Since there are no clocks running in Stop mode, only interrupts which can be asserted asynchronously can wake-up the MCU from Stop mode.

NOTE

> The only asynchronously asserted, I bit maskable interrupt for the MM912_637 would be the "D2D External Interrupt".

The X bit maskable interrupt request can wake-up the MCU from Stop or Wait mode at anytime, even if the X bit in CCR is set. If the X bit maskable interrupt request is used to wake-up the MCU with the X bit in the CCR set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAI or STOP instruction. This features works the same rules as with any interrupt request, i.e. care must be taken that the X interrupt request used for wake-up remains active at least until the system begins execution of the instruction following the WAI or STOP instruction; otherwise, wake-up may not occur.

NOTE

> The only X bit maskable interrupt for the MM912_637 would be the D2D Error Interrupt. As the D2D Initiator module is not active during STOP and WAIT mode, no X bit maskable interrupt source is existing for the MM912_637.

4.19 Memory map control (S12PMMCV1)

4.19.1 Introduction

The S12PMMC module controls the access to all internal memories and peripherals for the CPU12 and S12SBDM module. It regulates access priorities and determines the address mapping of the on-chip resources. Figure 55 shows a block diagram of the S12PMMC module.

4.19.1.1 Glossary

Table 312. Glossary of terms

Term	Definition
Local Address	Address within the CPU12's Local Address Map (Figure 60)
Global Address	Address within the Global Address Map (Figure 60)
Aligned Bus Access	Bus access to an even address.
Misaligned Bus Access	Bus access to an odd address.
NS	Normal Single-chip Mode
SS	Special Single-chip Mode
Unimplemented Address Ranges	Address ranges which are not mapped to any on-chip resource.
P-Flash	Program Flash
D-Plash	Data Flash

Table 312. Glossary of terms (continued)

Term	Definition
NVM	Non-volatile Memory; P-Flash or D-Flash
IFR	NVM Information Row. Refer to FTMRC Block Guide

4.19.1.2 Overview

The S12PMMC connects the CPU12's and the S12SBDM's bus interfaces to the MCU's on-chip resources (memories and peripherals). It arbitrates the bus accesses and determines all of the MCU's memory maps. Furthermore, the S12PMMC is responsible for constraining memory accesses on secured devices and for selecting the MCU's functional mode.

4.19.1.3 Features

The main features of this block are:

- Paging capability to support a global 256 kByte memory address space
- Bus arbitration between the masters CPU12, S12SBDM to different resources
- MCU operation mode control
- MCU security control
- Separate memory map schemes for each master CPU12, S12SBDM
- Generation of system reset when CPU12 accesses an unimplemented address (i.e., an address which does not belong to any of the on-chip modules) in single-chip modes

4.19.1.4 Modes of operation

The S12PMMC selects the MCU's functional mode. It also determines the devices behavior in secured and unsecured state.

4.19.1.4.1 Functional modes

Two functional modes are implemented on devices of the S12I product family:

- Normal Single Chip (NS)
  The mode used for running applications.
- Special Single Chip Mode (SS)
  A debug mode which causes the device to enter BDM Active Mode after each reset. Peripherals may also provide special debug features in this mode

4.19.1.4.2 Security

S12I devices can be secured to prohibit external access to the on-chip P-Flash. The S12PMMC module determines the access permissions to the on-chip memories in secured and unsecured state.

4.19.1.5 Block diagram

Figure 55 shows a block diagram of the S12PMMC.

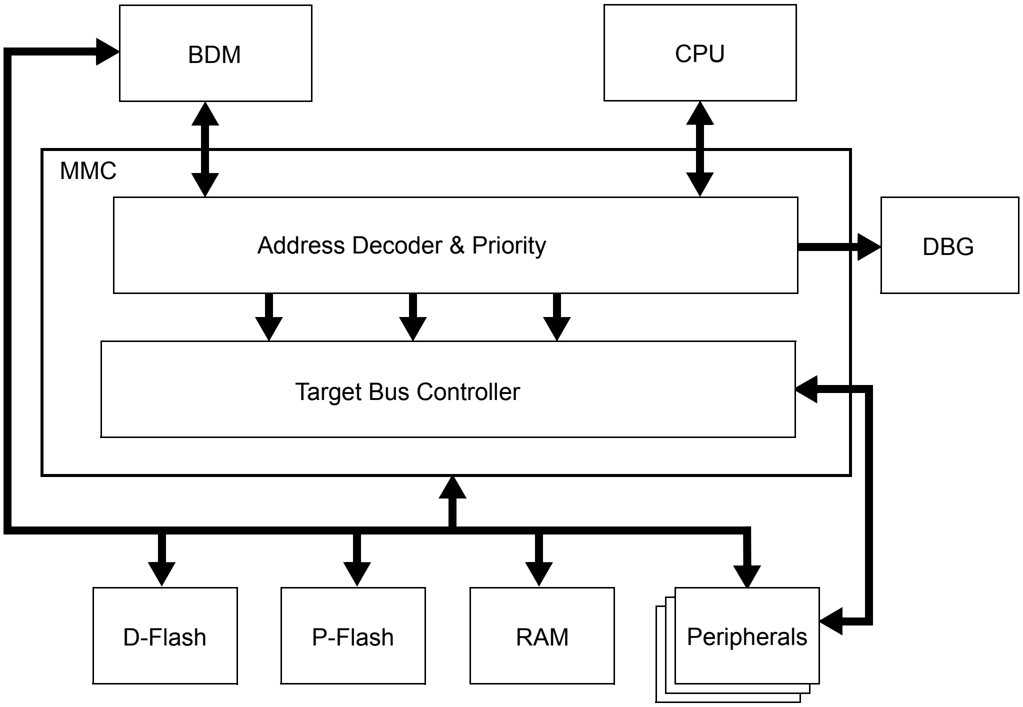


Figure 55. S12PMMC block diagram

4.19.2 External signal description

The S12PMMC uses two external pins to determine the devices operating mode: RESET and MODC (Table 313) See Device User Guide (DUG) for the mapping of these signals to device pins.

Table 313. External system pins associated With S12PMMC

Pin Name	Pin Functions	Description
$\overline{\text{RESET}}$ (See DUG)	$\overline{\text{RESET}}$	The $\overline{\text{RESET}}$ pin is used the select the MCU’s operating mode.
$\overline{\text{MODC}}$ (See DUG)	MODC	The MODC pin is captured at the rising edge of the $\overline{\text{RESET}}$ pin. The captured value determines the MCU’s operating mode.

4.19.3 Memory map and registers

4.19.3.1 Module memory map

A summary of the registers associated with the S12PMMC block is shown in Table 314. Detailed descriptions of the registers and bits are given in the subsections that follow.

Table 314. MMC register table

Address	Register name		Bit 7	6	5	4	3	2	1	Bit 0
0x000A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x000B	MODE	R	MODC	0	0	0	0	0	0	0
		W								

Table 314. MMC register table (continued)

0x0010	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0011	DIRECT	R	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
		W								
0x0012	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0013	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0014	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0015	PPAGE	R	0	0	0	0	PIX3	PIX2	PIX1	PIX0
		W								

□ = Unimplemented or Reserved

4.19.3.2 Register descriptions

This section consists of the S12PMMC control register descriptions in address order.

4.19.3.2.1 Mode register (MODE)

Table 315. Mode register (MODE)

Address: 0x000B

	7	6	5	4	3	2	1	0
R	MODC	0	0	0	0	0	0	0
W								
Reset	MODC[247]	0	0	0	0	0	0	0

□ = Unimplemented or Reserved

Notes

[247] External signal (see Table 313).

Read: Anytime.
Write: Only if a transition is allowed (see Figure 56).
The MODC bit of the MODE register is used to select the MCU's operating mode.

Table 316. MODE field descriptions

Field	Description
7 MODC	**Mode Select Bit** — This bit controls the current operating mode during $\overline{\text{RESET}}$ high (inactive). The external mode MODC pin determines the operating mode during $\overline{\text{RESET}}$ low (active). The state of the pin is registered into the respective register bit after the $\overline{\text{RESET}}$ signal goes inactive (see Figure 56). Write restrictions exist to disallow transitions between certain modes. Figure 56 illustrates all allowed mode changes. Attempting non authorized transitions will not change the MODE bit, but it will block further writes to the register bit except in special modes. Write accesses to the MODE register are blocked when the device is secured.

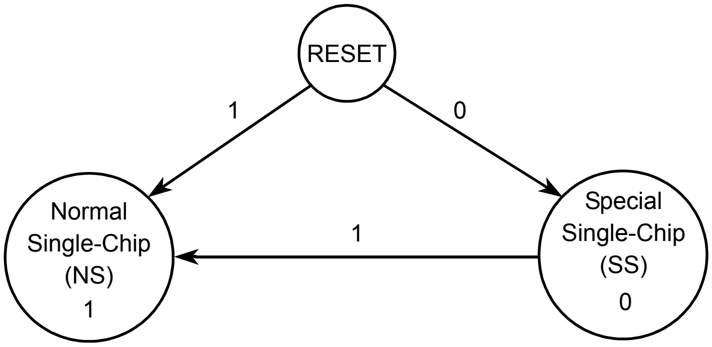


Figure 56. Mode transition diagram when MCU is unsecured

4.19.3.2.2 Direct page register (DIRECT)

Table 317. Direct register (DIRECT)

Address: 0x0011

	7	6	5	4	3	2	1	0
R W	DP15	DP14	DP13	DP12	DP11	DP10	DP9	DP8
Reset	0	0	0	0	0	0	0	0

Read: Anytime.
Write: anytime in special SS, write-one in NS.

This register determines the position of the 256 Byte direct page within the memory map.It is valid for both global and local mapping scheme.

Table 318. DIRECT field descriptions

Field	Description
7–0 DP[15:8]	**Direct Page Index Bits 15–8** — These bits are used by the CPU when performing accesses using the direct addressing mode. These register bits form bits [15:8] of the local address (see Figure 57).

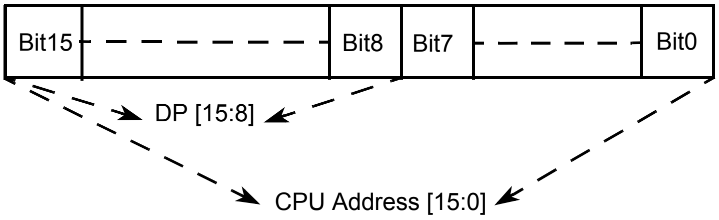


Figure 57. DIRECT address mapping

Example 1. This example demonstrates usage of the Direct Addressing Mode

```
MOVB    #$80,DIRECT         ;Set DIRECT register to 0x80. Write once only.
                            ;Global data accesses to the range 0xXX_80XX can be direct.
                            ;Logical data accesses to the range 0x80XX are direct.

LDY     <$00                ;Load the Y index register from 0x8000 (direct access).
                            ;< operator forces direct access on some assemblers but in
                            ;many cases assemblers are “direct page aware” and can
                            ;automatically select direct mode.
```

4.19.3.2.3 Program page index register (PPAGE)

Table 319. Program page index register (PPAGE)

Address: 0x0030

	7	6	5	4	3	2	1	0
R	0	0	0	0	PIX3	PIX2	PIX1	PIX0
W								
Reset	0	0	0	0	1	1	1	0

Read: Anytime.
Write: Anytime.

These four index bits are used to map 16 kB blocks into the Flash page window located in the local (CPU or BDM) memory map, from address 0x8000 to address 0xBFFF (see Figure 58). This supports accessing up to 256 kB of Flash (in the Global map) within the 64 kB Local map. The PPAGE index register is effectively used to construct paged Flash addresses in the Local map format. The CPU has special access to read and write this register directly during execution of CALL and RTC instructions.

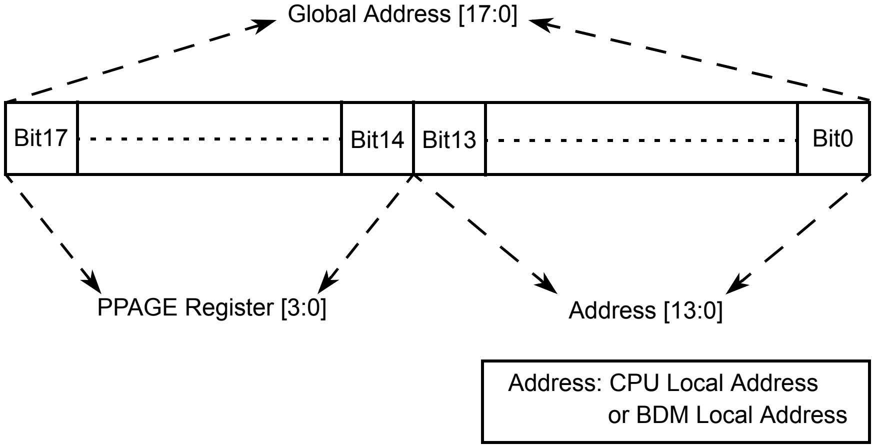


Figure 58. PPAGE address mapping

NOTE

Writes to this register using the special access of the CALL and RTC instructions will be complete before the end of the instruction execution.

Table 320. PPAGE field descriptions

Field	Description
3–0 PIX[3:0]	**Program Page Index Bits 3–0** — These page index bits are used to select which of the 256 P-Flash or ROM array pages is to be accessed in the Program Page Window.

The fixed 16 kB page from 0x0000 to 0x3FFF is the page number 0x0C. Parts of this page are covered by Registers, D-Flash and RAM space. See the SoC Guide for details.

The fixed 16 kB page from 0x4000–0x7FFF is the page number 0x0D. The reset value of 0x0E ensures that there is linear Flash space available between addresses 0x0000 and 0xFFFF out of reset. The fixed 16 kB page from 0xC000-0xFFFF is the page number 0x0F.

4.19.4 Functional description

The S12PMMC block performs several basic functions of the S12I sub-system operation: MCU operation modes, priority control, address mapping, select signal generation, and access limitations for the system. Each aspect is described in the following subsections.

4.19.4.1 MCU operating modes

- Normal single chip mode
  This is the operation mode for running application code. There is no external bus in this mode.
- Special single chip mode
  This mode is generally used for debugging operation, boot-strapping or security related operations. The active background debug mode is in control of the CPU code execution and the BDM firmware is waiting for serial commands sent through the BKGD pin.

4.19.4.2 Memory map scheme

4.19.4.2.1 CPU and BDM memory map scheme

The BDM firmware lookup tables and BDM register memory locations share addresses with other modules; however they are not visible in the memory map during user's code execution. The BDM memory resources are enabled only during the READ_BD and WRITE_BD access cycles to distinguish between accesses to the BDM memory area and accesses to the other modules. (Refer to the BDM Block Guide for further details).

When the MCU enters active BDM mode, the BDM firmware look-up tables and the BDM registers become visible in the local memory map in the range 0xFF00-0xFFFF (global address 0x3_FF00 - 0x3_FFFF) and the CPU begins execution of firmware commands or the BDM begins execution of hardware commands. The resources which share memory space with the BDM module will not be visible in the memory map during active BDM mode.

Note that after the MCU enters active BDM mode the BDM firmware look-up tables and the BDM registers will also be visible between addresses 0xBF00 and 0xBFFF if the PPAGE register contains value of 0x0F.

4.19.4.2.1.1 Expansion of the local address map

4.19.4.2.1.1.1 Expansion of the CPU local address map

The program page index register in S12PMMC allows accessing up to 256 kB of P-Flash in the global memory map by using the four index bits (PPAGE[3:0]) to page 16x16 kB blocks into the program page window, located from address 0x8000 to address 0xBFFF in the local CPU memory map.

The page value for the program page window is stored in the PPAGE register. The value of the PPAGE register can be read or written by normal memory accesses as well as by the CALL and RTC instructions (see Section 4.19.6.1, “CALL and RTC instructions").

Control registers, vector space and parts of the on-chip memories are located in unpaged portions of the 64 kB local CPU address space.

The starting address of an interrupt service routine must be located in unpaged memory unless the user is certain that the PPAGE register will be set to the appropriate value when the service routine is called. However an interrupt service routine can call other routines that are in paged memory. The upper 16 kB block of the local CPU memory space (0xC000–0xFFFF) is unpaged. It is recommended that all reset and interrupt vectors point to locations in this area or to the other unmapped pages sections of the local CPU memory map.

4.19.4.2.1.1.2 Expansion of the BDM local address map

PPAGE and BDMPPR register is also used for the expansion of the BDM local address to the global address. These registers can be read and written by the BDM. The BDM expansion scheme is the same as the CPU expansion scheme. The four BDMPPR Program Page index bits allow access to the full 256 kB address map that can be accessed with 18 address bits.

The BDM program page index register (BDMPPR) is used only when the feature is enabled in BDM and, in the case the CPU is executing a firmware command which uses CPU instructions, or by a BDM hardware commands. See the BDM Block Guide for further details. (see Figure 59).

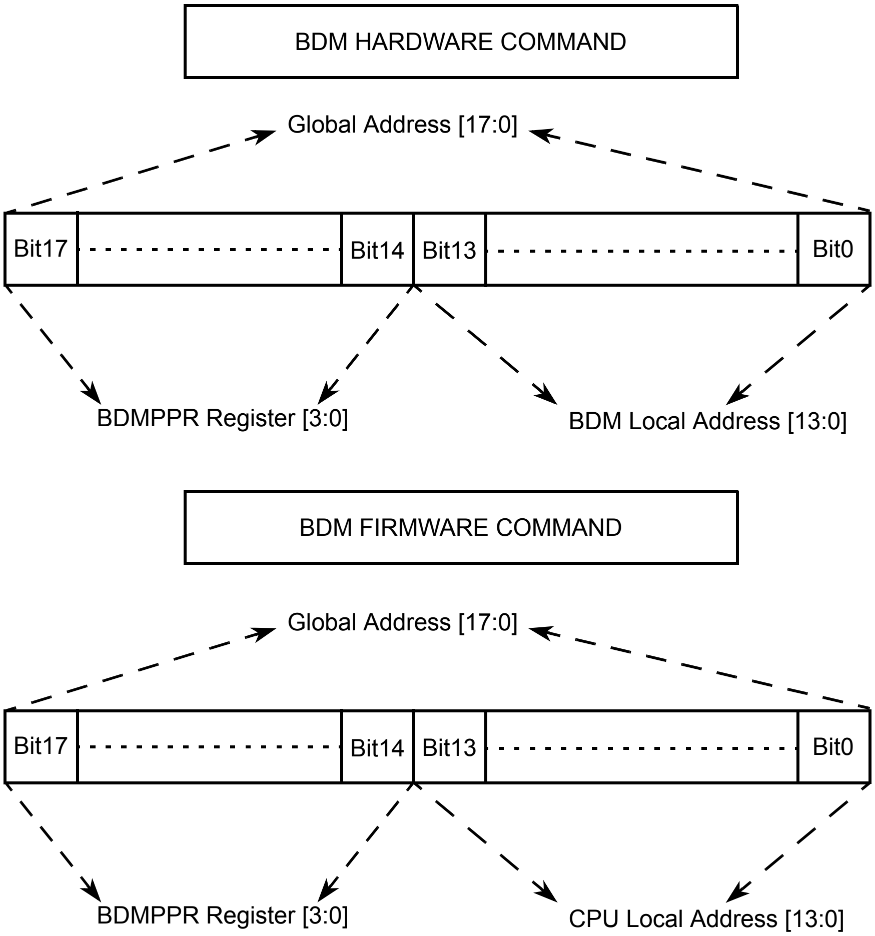


Figure 59. BDMPPR address mapping

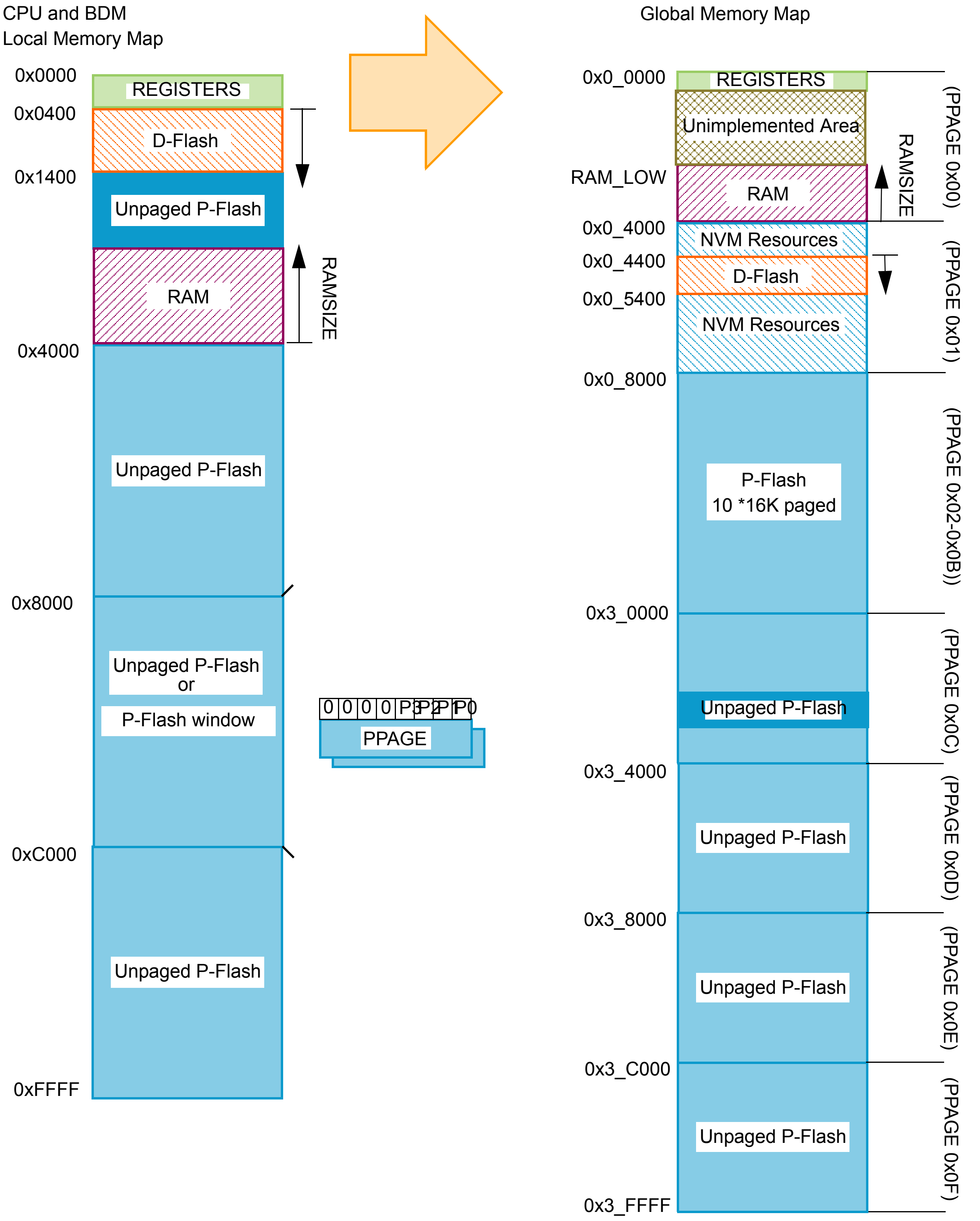


Figure 60. Local to global address mapping

4.19.5 Implemented memory in the system memory architecture

Each memory can be implemented in its maximum allowed size. But some devices have been defined for smaller sizes, which means less implemented pages. All non implemented pages are called unimplemented areas.

- Registers has a fixed size of 1.0 kB, accessible via xbus0
- SRAM has a maximum size of 11 kB, accessible via xbus0
- D-Flash has a fixed size of 4.0 kB accessible via xbus0
- P-Flash has a maximum size of 224 kB, accessible via xbus0
- NVM resources (IFR) including D-Flash have maximum size of 16 kB (PPAGE 0x01)

4.19.5.0.1 Implemented memory map

The global memory spaces reserved for the internal resources (RAM, D-Flash, and P-Flash) are not determined by the MMC module. Size of the individual internal resources are however fixed in the design of the device cannot be changed by the user. Refer to the SoC Guide for further details. Figure 61 and Table 321 show the memory spaces occupied by the on-chip resources. Note that the memory spaces have fixed top addresses.

Table 321. Global implemented memory space

Internal resource	Bottom address	Top address
Registers	0x0_0000	0x0_03FF
System RAM	RAM_LOW = 0x0_4000 minus RAMSIZE(248)	0x0_3FFF
D-Flash	0x0_4400	0x0_53FF
P-Flash	PF_LOW = 0x4_0000 minus FLASHSIZE(249)	0x3_FFFF

Notes

248. RAMSIZE is the hexadecimal value of RAM SIZE in bytes
249. FLASHSIZE is the hexadecimal value of FLASH SIZE in bytes

In single-chip modes accesses by the CPU12 (except for firmware commands) to any of the unimplemented areas (see Figure 61) will result in an illegal access reset (system reset). BDM accesses to the unimplemented areas are allowed but the data will be undefined.

No misaligned word access from the BDM module will occur; these accesses are blocked in the BDM module (Refer to BDM Block Guide).

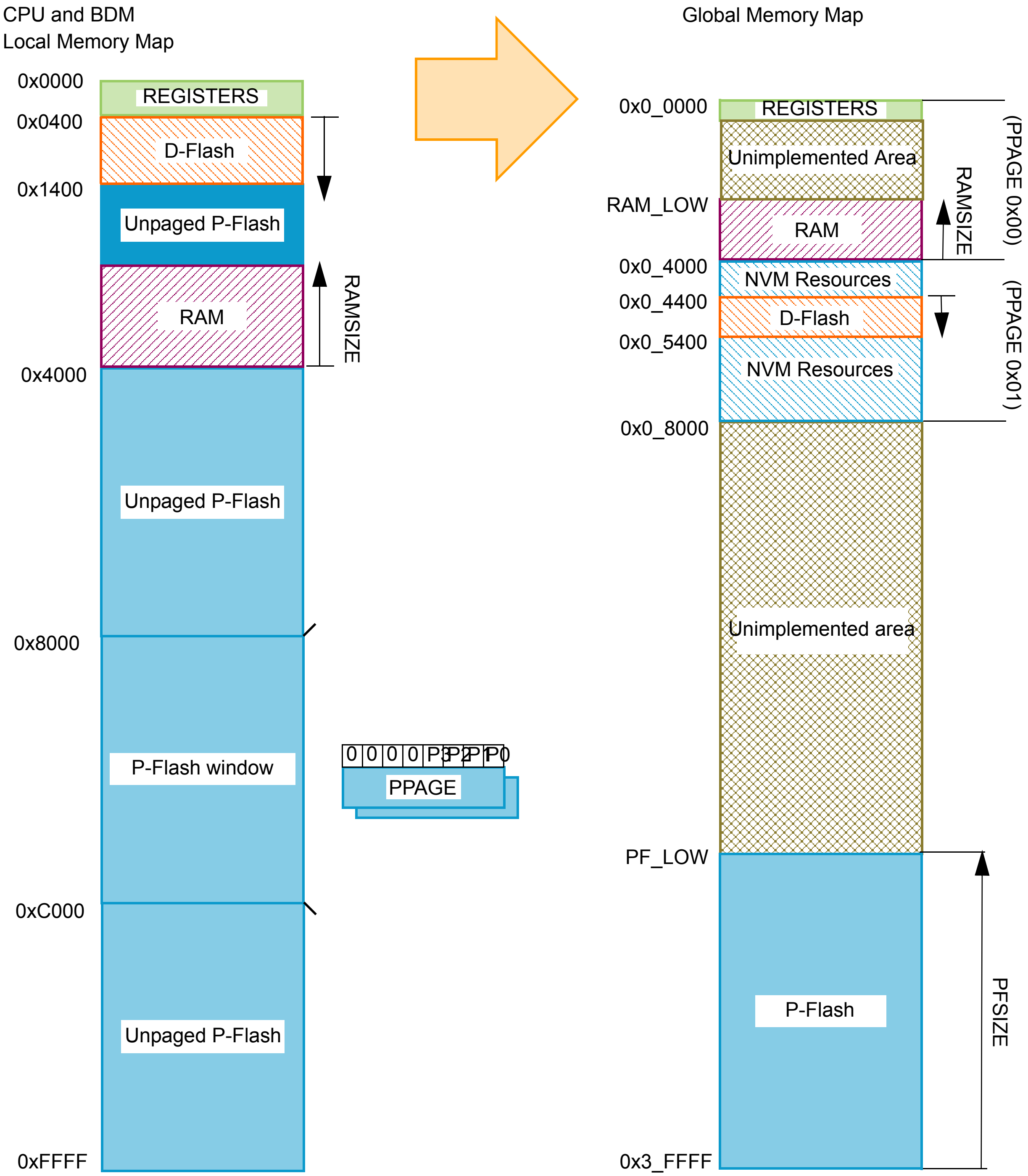


Figure 61. Implemented global address mapping

4.19.5.1 Chip bus control

The S12PMMC controls the address buses and the data buses that interface the bus masters (CPU12, S12SBDM) with the rest of the system (master buses). In addition, the MMC handles all CPU read data bus swapping operations. All internal resources are connected to specific target buses (see Figure 62).

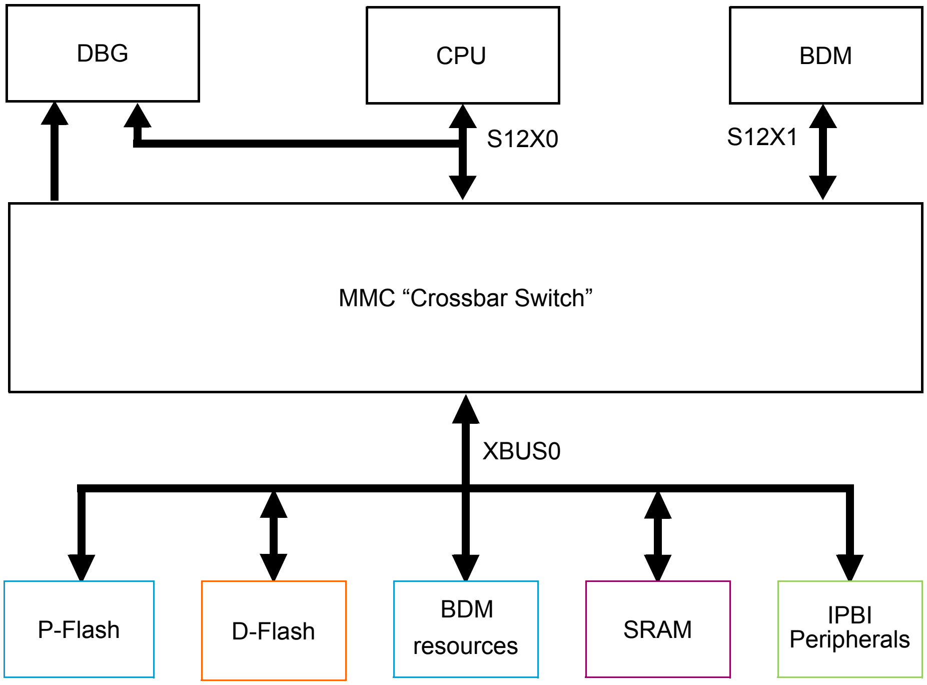


Figure 62. S12I platform

4.19.5.1.1 Master bus prioritization regarding access conflicts on target buses

The arbitration scheme allows only one master to be connected to a target at any given time. The following rules apply when prioritizing accesses from different masters to the same target bus:

- CPU12 always has priority over BDM.
- BDM has priority over CPU12 when its access is stalled for more than 128 cycles. In the later case the CPU will be stalled after finishing the current operation and the BDM will gain access to the bus.

4.19.5.2 Interrupts

The MMC does not generate any interrupts.

4.19.6 Initialization/application information

4.19.6.1 CALL and RTC instructions

CALL and RTC instructions are uninterruptable CPU instructions that automate page switching in the program page window. The CALL instruction is similar to the JSR instruction, but the subroutine that is called can be located anywhere in the local address space or in any Flash or ROM page visible through the program page window. The CALL instruction calculates and stacks a return address, stacks the current PPAGE value and writes a new instruction-supplied value to the PPAGE register. The PPAGE value controls which of the 256 possible pages is visible through the 16 kbyte program page window in the 64 kbyte local CPU memory map. Execution then begins at the address of the called subroutine.

During the execution of the CALL instruction, the CPU performs the following steps:

1. Writes the current PPAGE value into an internal temporary register and writes the new instruction supplied PPAGE value into the PPAGE register
2. Calculates the address of the next instruction after the CALL instruction (the return address) and pushes this 16-bit value onto the stack
3. Pushes the temporarily stored PPAGE value onto the stack
4. Calculates the effective address of the subroutine, refills the queue and begins execution at the new address

This sequence is uninterruptable. There is no need to inhibit interrupts during the CALL instruction execution. A CALL instruction can be performed from any address to any other address in the local CPU memory space.

The PPAGE value supplied by the instruction is part of the effective address of the CPU. For all addressing mode variations (except indexed-indirect modes), the new page value is provided by an immediate operand in the instruction. In indexed indirect variations of the CALL instruction, a pointer specifies memory locations where the new page value and the address of the called subroutine are stored. Using indirect addressing for both the new page value and the address within the page allows usage of values calculated at run time, rather than immediate values that must be known at the time of assembly.

The RTC instruction terminates subroutines invoked by a CALL instruction. The RTC instruction unstacks the PPAGE value and the return address and refills the queue. Execution resumes with the next instruction after the CALL instruction.

During the execution of an RTC instruction the CPU performs the following steps:

1. Pulls the previously stored PPAGE value from the stack
2. Pulls the 16-bit return address from the stack and loads it into the PC
3. Writes the PPAGE value into the PPAGE register
4. Refills the queue and resumes execution at the return address

This sequence is uninterruptable. The RTC can be executed from anywhere in the local CPU memory space.

The CALL and RTC instructions behave like JSR and RTS instruction. However they require more execution cycles. Usage of JSR/RTS instructions is therefore recommended when possible, and CALL/RTC instructions should only be used when needed. The JSR and RTS instructions can be used to access subroutines that are already present in the local CPU memory map (i.e. in the same page in the program memory page window for example). However calling a function located in a different page requires usage of the CALL instruction. The function must be terminated by the RTC instruction. Because the RTC instruction restores contents of the PPAGE register from the stack. Functions terminated with the RTC instruction must be called using the CALL instruction, even when the correct page is already present in the memory map. This is to make sure that the correct PPAGE value will be present on the stack at the time of the RTC instruction execution.

4.20 MCU - debug module (9S12I128PIMV1)

4.20.1 Introduction

The 9S12I128PIMV1 module provides an on-chip trace buffer with flexible triggering capability, to allow non-intrusive debug of application software. The 9S12I128PIMV1 module is optimized for S12SCPUdebugging.

Typically, the 9S12I128PIMV1 module is used in conjunction with the S12SBDM module, whereby the user configures the 9S12I128PIMV1 module for a debugging session over the BDM interface. Once configured the 9S12I128PIMV1 module is armed, the device leaves BDM returning control to the user program, which is then monitored by the 9S12I128PIMV1 module. Alternatively the 9S12I128PIMV1 module can be configured over a serial interface using SWI routines.

4.20.1.1 Glossary Of terms

COF: Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt.

BDM: Background Debug Mode

S12SBDM: Background Debug Module

DUG: Device User Guide, describing the features of the device into which the DBG is integrated.

WORD: 16-bit data entity

Data Line: 20-bit data entity

CPU: S12SCPU module

DBG: S12SDBG module

POR: Power On Reset

Tag: Tags can be attached to CPU opcodes as they enter the instruction pipe. If the tagged opcode reaches the execution stage a tag hit occurs.

4.20.1.2 Overview

The comparators monitor the bus activity of the CPU module. A match can initiate a state sequencer transition. On a transition to the Final state, bus tracing is triggered and/or a breakpoint can be generated.

Independent of comparator matches a transition to Final state with associated tracing and breakpoint can be triggered immediately by writing to the TRIG control bit. The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads. Tracing is disabled when the MCU system is secured.

4.20.1.3 Features

- Three comparators (A, B and C)
  - Comparators A compares the full address bus and full 16-bit data bus
  - Comparator A features a data bus mask register
  - Comparators B and C compare the full address bus only
  - Each comparator features selection of read or write access cycles
  - Comparator B allows selection of byte or word access cycles
  - Comparator matches can initiate state sequencer transitions
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin ≤ Address ≤ Addmax
  - Outside address range match mode, Address < Admin or Address > Addmax
- Two types of matches
  - Tagged — This matches just before a specific instruction begins execution
  - Force — This is valid on the first instruction boundary after a match occurs
- Two types of breakpoints
  - CPU breakpoint entering BDM on breakpoint (BDM)
  - CPU breakpoint executing SWI on breakpoint (SWI)
- Trigger mode independent of comparators
  - TRIG Immediate software trigger
- Four trace modes
  - Normal: change of flow (COF) PC information is stored (see Section 4.20.4.5.2.1, “Normal mode") for change of flow definition.
  - Loop1: same as Normal but inhibits consecutive duplicate source address entries
  - Detail: address and data for all cycles except free cycles and opcode fetches are stored
  - Compressed Pure PC: all program counter addresses are stored
- 4-stage state sequencer for trace buffer control
  - Tracing session trigger linked to Final State of state sequencer
  - Begin and End alignment of tracing to trigger

4.20.1.4 Modes of operation

The DBG module can be used in all MCU functional modes. During BDM hardware accesses and while the BDM module is active, CPU monitoring is disabled. When the CPU enters active BDM Mode through a BACKGROUND command, the DBG module, if already armed, remains armed. The DBG module tracing is disabled if the MCU is secure, however, breakpoints can still be generated.

Table 322. Mode dependent restriction summary

BDM enable	BDM active	MCU secure	Comparator matches enabled	Breakpoints possible	Tagging possible	Tracing possible
x	x	1	Yes	Yes	Yes	No
0	0	0	Yes	Only SWI	Yes	Yes
0	1	0	Active BDM not possible when not enabled			
1	0	0	Yes	Yes	Yes	Yes
1	1	0	No	No	No	No

4.20.1.5 Block diagram

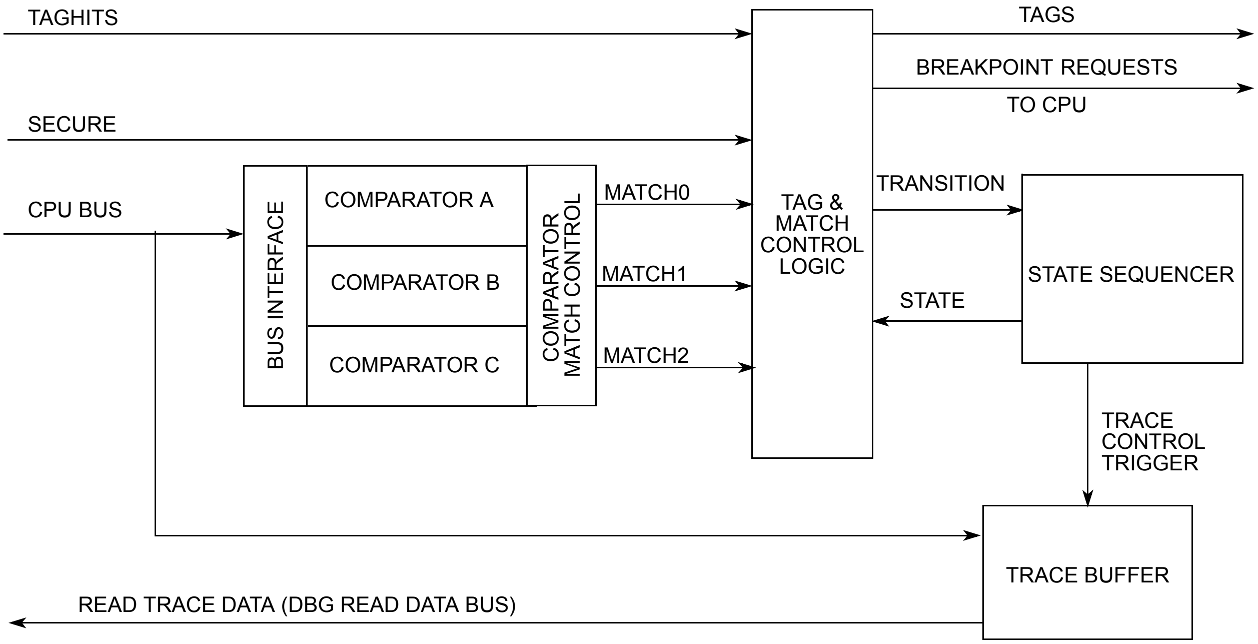


Figure 63. Debug module block diagram

4.20.2 External signal description

There are no external signals associated with this module.

4.20.3 Memory map and registers

4.20.3.1 Module memory map

A summary of the registers associated with the DBG sub-block is shown in Table 323. Detailed descriptions of the registers and bits are given in the subsections that follow.

Table 323. Quick reference to DBG registers

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0020	DBGC1	R	ARM	0	0	BDM	DBGBRK	0	COMRV	
		W		TRIG						
0x0021	DBGSR	R	[250]TBF	0	0	0	0	SSF2	SSF1	SSF0
		W								
0x0022	DBGTCR	R	0	TSOURCE	0	0	TRCMOD		0	TALIGN
		W								
0x0023	DBGC2	R	0	0	0	0	0	0	ABCM	
		W								
0x0024	DBGTBH	R	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
		W								

Table 323. Quick reference to DBG registers (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0025	DBGTBL	R	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
		W								
0x0026	DBGCNT	R	TBF	0	CNT					
		W								
0x0027	DBGSCRX	R	0	0	0	0	SC3	SC2	SC1	SC0
		W								
0x0027	DBGMFR	R	0	0	0	0	0	MC2	MC1	MC0
		W								
0x0028[251]	DBGACTL	R	SZE	SZ	TAG	BRK	RW	RWE	NDB	COMPE
		W								
0x0028[252]	DBGBCTL	R	SZE	SZ	TAG	BRK	RW	RWE	0	COMPE
		W								
0x0028[253]	DBGCCTL	R	0	0	TAG	BRK	RW	RWE	0	COMPE
		W								
0x0029	DBGXAH	R	0	0	0	0	0	0	Bit 17	Bit 16
		W								
0x002A	DBGXAM	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002B	DBGXAL	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x002C	DBGADH	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002D	DBGADL	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								
0x002E	DBGADHM	R	Bit 15	14	13	12	11	10	9	Bit 8
		W								
0x002F	DBGADLM	R	Bit 7	6	5	4	3	2	1	Bit 0
		W								

Notes

[250] This bit is visible at DBGCNT[7] and DBGSR[7]

[251] This represents the contents if the Comparator A control register is blended into this address

[252] This represents the contents if the Comparator B control register is blended into this address

[253] This represents the contents if the Comparator C control register is blended into this address

4.20.3.2 Register descriptions

This section consists of the DBG control and trace buffer register descriptions in address order. Each comparator has a bank of registers that are visible through an 8-byte window between 0x0028 and 0x002F in the DBG module register address map. When ARM is set in DBGC1, the only bits in the DBG module registers that can be written are ARM, TRIG, and COMRV[1:0]

4.20.3.2.1 Debug control register 1 (DBGC1)

Table 324. Debug control register (DBGC1)

Address: 0x0020

	7	6	5	4	3	2	1	0
R	ARM	0	0	BDM	DBGBRK	0	COMRV	
W		TRIG	▨			▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime
Write: Bits 7, 1, 0 anytime
Bit 6 can be written anytime but always reads back as 0.
Bits 4:3 anytime DBG is not armed.

NOTE

When disarming the DBG by clearing ARM with software, the contents of bits[4:3] are not affected by the write, since up until the write operation, ARM = 1 preventing these bits from being written. These bits must be cleared using a second write if required.

Table 325. DBGC1 field descriptions

Field	Description
7 ARM	**Arm Bit** — The ARM bit controls whether the DBG module is armed. This bit can be set and cleared by user software and is automatically cleared on completion of a debug session, or if a breakpoint is generated with tracing not enabled. On setting this bit the state sequencer enters State1. 0 Debugger disarmed 1 Debugger armed
6 TRIG	**Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate trigger independent of state sequencer status. When tracing is complete a forced breakpoint may be generated depending upon DBGBRK and BDM bit settings. This bit always reads back a 0. Writing a 0 to this bit has no effect. If the DBGTCR_TSOURCE bit is clear no tracing is carried out. If tracing has already commenced using BEGIN trigger alignment, it continues until the end of the tracing session as defined by the TALIGN bit, thus TRIG has no affect. In secure mode, tracing is disabled and writing to this bit cannot initiate a tracing session. The session is ended by setting TRIG and ARM simultaneously. 0 Do not trigger until the state sequencer enters the Final state. 1 Trigger immediately
4 BDM	**Background Debug Mode Enable** — This bit determines if a breakpoint causes the system to enter Background Debug mode (BDM) or initiate a Software Interrupt (SWI). If this bit is set but the BDM is not enabled by the ENBDM bit in the BDM module, then breakpoints default to SWI. 0 Breakpoint to Software Interrupt if BDM inactive. Otherwise no breakpoint. 1 Breakpoint to BDM, if BDM enabled. Otherwise breakpoint to SWI
3 DBGBRK	**9S12I128PIMV1 Breakpoint Enable Bit** — The DBGBRK bit controls whether the debugger will request a breakpoint on reaching the state sequencer Final State. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. 0 No Breakpoint generated 1 Breakpoint generated
1–0 COMRV	**Comparator Register Visibility Bits** — These bits determine which bank of comparator register is visible in the 8-byte window of the S12SDBG module address map, located between 0x0028 to 0x002F. Furthermore these bits determine which register is visible at the address 0x0027. See Table 326.

Table 326. COMRV encoding

COMRV	Visible comparator	Visible register at 0x0027
00	Comparator A	DBGSCR1
01	Comparator B	DBGSCR2
10	Comparator C	DBGSCR3
11	None	DBGMFR

4.20.3.2.2 Debug status register (DBGSR)

Table 327. Debug status register (DBGSR)

Address: 0x0021

	7	6	5	4	3	2	1	0
R	TBF	0	0	0	0	SSF2	SSF1	SSF0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	—	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime
Write: Never

Table 328. DBGSR field descriptions

Field	Description
7 TBF	**Trace Buffer Full** — The TBF bit indicates that the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines will be valid data, regardless of the value of DBGCNT bits. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit This bit is also visible at DBGCNT[7]
2–0 SSF[2:0]	**State Sequencer Flag Bits** — The SSF bits indicate in which state the State Sequencer is currently in. During a debug session on each transition to a new state these bits are updated. If the debug session is ended by software clearing the ARM bit, then these bits retain their value to reflect the last state of the state sequencer before disarming. If a debug session is ended by an internal event, then the state sequencer returns to state0 and these bits are cleared to indicate that state0 was entered during the session. On arming the module the state sequencer enters state1 and these bits are forced to SSF[2:0] = 001. See Table 329**.**

Table 329. SSF[2:0] — State sequence flag bit encoding

SSF[2:0]	Current state
000	State0 (disarmed)
001	State1
010	State2
011	State3
100	Final State
101,110,111	Reserved

4.20.3.2.3 Debug trace control register (DBGTCR)

Table 330. Debug trace control register (DBGTCR)

Address: 0x0022

<table>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td>0</td><td rowspan="2">TSOURCE</td><td>0</td><td>0</td><td rowspan="2" colspan="2">TRCMOD</td><td>0</td><td rowspan="2">TALIGN</td></tr>
<tr><td>W</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

Read: Anytime
Write: Bit 6 only when DBG is neither secure nor armed.Bits 3,2,0 anytime the module is disarmed.

Table 331. DBGTCR field descriptions

Field	Description
6 TSOURCE	**Trace Source Control Bit** — The TSOURCE bit enables a tracing session given a trigger condition. If the MCU system is secured, this bit cannot be set and tracing is inhibited. This bit must be set to read the trace buffer. 0 Debug session without tracing requested 1 Debug session with tracing requested
3–2 TRCMOD	**Trace Mode Bits** — See Section 4.20.4.5.2, “Trace modes" for detailed Trace mode descriptions. In Normal mode, change of flow information is stored. In Loop1 mode, change of flow information is stored but redundant entries into trace memory are inhibited. In Detail mode, address and data for all memory and register accesses is stored. In Compressed Pure PC mode the program counter value for each instruction executed is stored. See Table 332.
0 TALIGN	**Trigger Align Bit** — This bit controls whether the trigger is aligned to the beginning or end of a tracing session. 0 Trigger at end of stored data 1 Trigger before storing data

Table 332. TRCMOD trace mode bit encoding

TRCMOD	Description
00	Normal
01	Loop1
10	Detail
11	Compressed Pure PC

4.20.3.2.4 Debug control register2 (DBGC2)

Table 333. Debug control register2 (DBGC2)

Address: 0x0023

<table>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td rowspan="2" colspan="2">ABCM</td></tr>
<tr><td>W</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td><td>▨</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

▨ = Unimplemented or Reserved

Read: Anytime
Write: Anytime the module is disarmed

This register configures the comparators for range matching.

Table 334. DBGC2 field descriptions

Field	Description
1–0 ABCM[1:0]	**A and B Comparator Match Control** — These bits determine the A and B comparator match mapping as described in Table 335.

Table 335. ABCM encoding

ABCM	Description
00	Match0 mapped to comparator A match: Match1 mapped to comparator B match.
01	Match 0 mapped to comparator A/B inside range: Match1 disabled.
10	Match 0 mapped to comparator A/B outside range: Match1 disabled.
11	Reserved[254]

Notes
[254] Currently defaults to Comparator A, Comparator B disabled

4.20.3.2.5 Debug trace buffer register (DBGTBH:DBGTBL)

Table 336. Debug trace buffer register (DBGTB)

Address: 0x0024, 0x0025

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
POR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Other Resets	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Read: Only when unlocked AND unsecured AND not armed AND TSOURCE set
Write: Aligned word writes when disarmed unlock the trace buffer for reading but do not affect trace buffer contents

Table 337. DBGTB field descriptions

Field	Description
15–0 Bit[15:0]	**Trace Buffer Data Bits** — The Trace Buffer Register is a window through which the 20-bit wide data lines of the Trace Buffer may be read 16 bits at a time. Each valid read of DBGTB increments an internal trace buffer pointer which points to the next address to be read. When the ARM bit is set the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by writing to DBGTB with an aligned word write when the module is disarmed. The DBGTB register can be read only as an aligned word, any byte reads or misaligned access of these registers return a 0, and do not cause the trace buffer pointer to increment to the next trace buffer address. Similarly reads while the debugger is armed or with the TSOURCE bit clear, return a 0, and do not affect the trace buffer pointer. The POR state is undefined. Other resets do not affect the trace buffer contents.

4.20.3.2.6 Debug count register (DBGCNT)

Table 338. Debug count register (DBGCNT)

Address: 0x0026

	7	6	5	4	3	2	1	0
R	TBF	0	CNT					
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	—	—	—	—	—	—	—	—
POR	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime
Write: Never

Table 339. DBGCNT field descriptions

Field	Description
7 TBF	**Trace Buffer Full** — The TBF bit indicates that the trace buffer has stored 64 or more lines of data since it was last armed. If this bit is set, then all 64 lines will be valid data, regardless of the value of DBGCNT bits. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit This bit is also visible at DBGSR[7]
5–0 CNT[5:0]	**Count Value** — The CNT bits indicate the number of valid data 20-bit data lines stored in the Trace Buffer. Table 340 shows the correlation between the CNT bits and the number of valid data lines in the Trace Buffer. When the CNT rolls over to zero, the TBF bit in DBGSR is set and incrementing of CNT will continue in end-trigger mode. The DBGCNT register is cleared when ARM in DBGC1 is written to a one. The DBGCNT register is cleared by power-on-reset initialization but is not cleared by other system resets. Thus should a reset occur during a debug session, the DBGCNT register still indicates after the reset, the number of valid trace buffer entries stored before the reset occurred. The DBGCNT register is not decremented when reading from the trace buffer.

Table 340. CNT decoding table

TBF	CNT[5:0]	Description
0	000000	No data valid
0	000001 000010 000100 000110 ... 111111	1 line valid 2 lines valid 4 lines valid 6 lines valid ... 63 lines valid
1	000000	64 lines valid; if using Begin trigger alignment, ARM bit will be cleared and the tracing session ends.
1	000001 111110	64 lines valid, oldest data has been overwritten by most recent data

4.20.3.2.7 Debug state control registers

There is a dedicated control register for each of the state sequencer states 1 to 3, that determines if transitions from that state are allowed, depending upon comparator matches or tag hits, and define the next state for the state sequencer following a match. The three debug state control registers are located at the same address in the register address map (0x0027). Each register can be accessed using the COMRV bits in DBGC1 to blend in the required register. The COMRV = 11 value blends in the match flag register (DBGMFR).

Table 341. State control register access encoding

COMRV	Visible state control register
00	DBGSCR1
01	DBGSCR2
10	DBGSCR3
11	DBGMFR

4.20.3.2.7.1 Debug state control register 1 (DBGSCR1)

Table 342. Debug state control register 1 (DBGSCR1)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W	▨	▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: If COMRV[1:0] = 00
Write: If COMRV[1:0] = 00 and DBG is not armed

This register is visible at 0x0027 only with COMRV[1:0] = 00. The state control register 1 selects the targeted next state while in State1. The matches refer to the match channels of the comparator match control logic, as depicted in Figure 63 and described in Section 4.20.3.2.8.1, “Debug comparator control register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 343. DBGSCR1 field descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state while in State1, based upon the match event.

Table 344. State1 sequencer next state selection

SC[3:0]	Description (unspecified matches have no effect)
0000	Any match to Final State
0001	Match1 to State3
0010	Match2 to State2
0011	Match1 to State2
0100	Match0 to State2....... Match1 to State3
0101	Match1 to State3.........Match0 to Final State
0110	Match0 to State2....... Match2 to State3
0111	Either Match0 or Match1 to State2
1000	Reserved
1001	Match0 to State3
1010	Reserved
1011	Reserved
1100	Reserved

Table 344. State1 sequencer next state selection (continued)

SC[3:0]	Description (unspecified matches have no effect)
1101	Either Match0 or Match2 to Final State........Match1 to State2
1110	Reserved
1111	Reserved

The priorities described in Table 377 dictate that in the case of simultaneous matches, a match leading to final state has priority followed by the match on the lower channel number (0,1,2). Thus with SC[3:0]=1101, a simultaneous match0/match1 transitions to Final state.

4.20.3.2.7.2 Debug state control register 2 (DBGSCR2)

Table 345. Debug state control register 2 (DBGSCR2)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W	▨	▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: If COMRV[1:0] = 01
Write: If COMRV[1:0] = 01 and DBG is not armed

This register is visible at 0x0027 only with COMRV[1:0] = 01. The state control register 2 selects the targeted next state while in State 2. The matches refer to the match channels of the comparator match control logic, as depicted in Figure 63 and described in Section 4.20.3.2.8.1, “Debug comparator control register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 346. DBGSCR2 field descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state while in State 2, based upon the match event.

Table 347. State2 —sequencer next state selection

SC[3:0]	Description (unspecified matches have no effect)
0000	Match0 to State1....... Match2 to State3.
0001	Match1 to State3
0010	Match2 to State3
0011	Match1 to State3....... Match0 Final State
0100	Match1 to State1....... Match2 to State3.
0101	Match2 to Final State
0110	Match2 to State1..... Match0 to Final State
0111	Either Match0 or Match1 to Final State
1000	Reserved
1001	Reserved
1010	Reserved
1011	Reserved

Table 347. State2 —sequencer next state selection (continued)

SC[3:0]	Description (unspecified matches have no effect)
1100	Either Match0 or Match1 to Final State........Match2 to State3
1101	Reserved
1110	Reserved
1111	Either Match0 or Match1 to Final State........Match2 to State1

The priorities described in Table 377 dictate that in the case of simultaneous matches, a match leading to final state has priority followed by the match on the lower channel number (0,1,2).

4.20.3.2.7.3 Debug state control register 3 (DBGSCR3)

Table 348. Debug state control register 3 (DBGSCR3)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	SC3	SC2	SC1	SC0
W	▨	▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: If COMRV[1:0] = 10
Write: If COMRV[1:0] = 10 and DBG is not armed

This register is visible at 0x0027 only with COMRV[1:0] = 10. The state control register three selects the targeted next state while in State 3. The matches refer to the match channels of the comparator match control logic, as depicted in Figure 63 and described in Section 4.20.3.2.8.1, “Debug comparator control register (DBGXCTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 349. DBGSCR3 field descriptions

Field	Description
3–0 SC[3:0]	These bits select the targeted next state while in State 3, based upon the match event.

Table 350. State3 — sequencer next state selection

SC[3:0]	Description (Unspecified matches have no effect)
0000	Match0 to State1
0001	Match2 to State2........ Match1 to Final State
0010	Match0 to Final State....... Match1 to State1
0011	Match1 to Final State....... Match2 to State1
0100	Match1 to State2
0101	Match1 to Final State
0110	Match2 to State2........ Match0 to Final State
0111	Match0 to Final State
1000	Reserved
1001	Reserved
1010	Either Match1 or Match2 to State1....... Match0 to Final State

Table 350. State3 — sequencer next state selection (continued)

SC[3:0]	Description (Unspecified matches have no effect)
1011	Reserved
1100	Reserved
1101	Either Match1 or Match2 to Final State....... Match0 to State1
1110	Match0 to State2....... Match2 to Final State
1111	Reserved

The priorities described in Table 377 dictate that in the case of simultaneous matches, a match leading to final state has priority, followed by the match on the lower channel number (0,1,2).

4.20.3.2.7.4 Debug match flag register (DBGMFR)

Table 351. Debug match flag register (DBGMFR)

Address: 0x0027

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	MC2	MC1	MC0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: If COMRV[1:0] = 11
Write: Never

DBGMFR is visible at 0x0027 only with COMRV[1:0] = 11. It features 3 flag bits, each mapped directly to a channel. Should a match occur on the channel during the debug session, then the corresponding flag is set and remains set until the next time the module is armed, by writing to the ARM bit. Thus the contents are retained after a debug session for evaluation purposes. These flags cannot be cleared by software. They are cleared only when arming the module. A set flag does not inhibit the setting of other flags. Once a flag is set, further comparator matches on the same channel in the same session have no affect on that flag.

4.20.3.2.8 Comparator register descriptions

Each comparator has a bank of registers that are visible through an 8-byte window in the DBG module register address map. Comparator A consists of 8 register bytes (3 address bus compare registers, two data bus compare registers, two data bus mask registers, and a control register). Comparator B consists of four register bytes (three address bus compare registers and a control register). Comparator C consists of four register bytes (three address bus compare registers and a control register).

Each set of comparator registers can be accessed using the COMRV bits in the DBGC1 register. Unimplemented registers (e.g. Comparator B data bus and data bus masking) read as zero and cannot be written. The control register for comparator B differs from those of comparators A and C.

Table 352. Comparator register layout

0x0028	CONTROL	Read/Write	Comparators A,B and C
0x0029	ADDRESS HIGH	Read/Write	Comparators A,B and C
0x002A	ADDRESS MEDIUM	Read/Write	Comparators A,B and C
0x002B	ADDRESS LOW	Read/Write	Comparators A,B and C
0x002C	DATA HIGH COMPARATOR	Read/Write	Comparator A only
0x002D	DATA LOW COMPARATOR	Read/Write	Comparator A only
0x002E	DATA HIGH MASK	Read/Write	Comparator A only
0x002F	DATA LOW MASK	Read/Write	Comparator A only

4.20.3.2.8.1 Debug comparator control register (DBGXCTL)

The contents of register bits 7 and 6 differ, depending upon which comparator registers are visible in the 8-byte window of the DBG module register address map.

Table 353. Debug comparator control register DBGACTL (comparator A)

Address: 0x0028

	7	6	5	4	3	2	1	0
R	SZE	SZ	TAG	BRK	RW	RWE	NDB	COMPE
W								
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Table 354. Debug comparator control register DBGBCTL (comparator B)

Address: 0x0028

	7	6	5	4	3	2	1	0
R	SZE	SZ	TAG	BRK	RW	RWE	0	COMPE
W							▨	
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Table 355. Debug comparator control register DBGCCTL (comparator C)

Address: 0x0028

	7	6	5	4	3	2	1	0
R	0	0	TAG	BRK	RW	RWE	0	COMPE
W	▨	▨					▨	
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: DBGACTL if COMRV[1:0] = 00
DBGBCTL if COMRV[1:0] = 01
DBGCCTL if COMRV[1:0] = 10

Write: DBGACTL if COMRV[1:0] = 00 and DBG not armed
DBGBCTL if COMRV[1:0] = 01 and DBG not armed
DBGCCTL if COMRV[1:0] = 10 and DBG not armed

Table 356. DBGXCTL field descriptions

Field	Description
7 SZE (Comparators A and B)	**Size Comparator Enable Bit** — The SZE bit controls whether access size comparison is enabled for the associated comparator. This bit is ignored if the TAG bit in the same register is set. 0 Word/Byte access size is not used in comparison 1 Word/Byte access size is used in comparison
6 SZ (Comparators A and B)	**Size Comparator Value Bit** — The SZ bit selects either word or byte access size in comparison for the associated comparator. This bit is ignored if the SZE bit is cleared or if the TAG bit in the same register is set. 0 Word access size is compared 1 Byte access size is compared

Table 356. DBGXCTL field descriptions (continued)

Field	Description
5 TAG	**Tag Select**— This bit controls whether the comparator match has immediate effect, causing an immediate state sequencer transition or tag the opcode at the matched address. Tagged opcodes trigger only if they reach the execution stage of the instruction queue. 0 Allow state sequencer transition immediately on match 1 On match, tag the opcode. If the opcode is about to be executed allow a state sequencer transition
4 BRK	**Break**— This bit controls whether a comparator match terminates a debug session immediately, independent of state sequencer state. To generate an immediate breakpoint the module breakpoints must be enabled using the DBGC1 bit DBGBRK. 0 The debug session termination is dependent upon the state sequencer and trigger conditions. 1 A match on this channel terminates the debug session immediately; breakpoints if active are generated, tracing, if active, is terminated and the module disarmed.
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is not used if RWE = 0. This bit is ignored if the TAG bit in the same register is set. 0 Write cycle is matched1Read cycle is matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator.This bit is ignored if the TAG bit in the same register is set 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
1 NDB (Comparator A)	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. This bit is ignored if the TAG bit in the same register is set. This bit is only available for comparator A. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
0 COMPE	Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled

Table 357 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if the corresponding TAG bit is set, since the match occurs based on the tagged opcode reaching the execution stage of the instruction queue.

Table 357. Read or write comparison logic table

RWE bit	RW bit	RW signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write data bus
1	0	1	No match
1	1	0	No match
1	1	1	Read data bus

4.20.3.2.8.2 Debug comparator address high register (DBGXAH)

Table 358. Debug comparator address high register (DBGXAH)

Address: 0x0029

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	Bit 17	Bit 16
W	▨	▨	▨	▨	▨	▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

The DBGC1_COMRV bits determine which comparator address registers are visible in the 8-byte window from 0x0028 to 0x002F, as shown in Table 359

Table 359. Comparator address register visibility

COMRV	Visible comparator
00	DBGAAH, DBGAAM, DBGAAL
01	DBGBAH, DBGBAM, DBGBAL
10	DBGCAH, DBGCAM, DBGCAL
11	None

Notes

255. Read: Anytime. See **Table** for visible register encoding.
Write: If DBG not armed. See **Table** for visible register encoding.

Table 360. DBGXAH field descriptions

Field	Description
1–0 Bit[17:16]	**Comparator Address High Compare Bits** — The Comparator address high compare bits control whether the selected comparator compares the address bus bits [17:16] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

4.20.3.2.8.3 Debug comparator address mid register (DBGXAM)

Table 361. Debug comparator address mid register (DBGXAM)

Address: 0x002A

	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Read: Anytime. See Table 359 for visible register encoding.
Write: If DBG not armed. See Table 359 for visible register encoding.

Table 362. DBGXAM field descriptions

Field	Description
7–0 Bit[15:8]	**Comparator Address Mid Compare Bits** — The Comparator address mid compare bits control whether the selected comparator compares the address bus bits [15:8] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

4.20.3.2.8.4 Debug comparator address low register (DBGXAL)

Table 363. Debug comparator address low register (DBGXAL)

Address: 0x002B

	7	6	5	4	3	2	1	0
R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0

Read: Anytime. See **Table** for visible register encoding
Write: If DBG not armed. See **Table** for visible register encoding

Table 364. DBGXAL field descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Address Low Compare Bits** — The Comparator address low compare bits control whether the selected comparator compares the address bus bits [7:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

4.20.3.2.8.5 Debug comparator data high register (DBGADH)

Table 365. Debug comparator data high register (DBGADH)

Address: 0x002C

	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Read: If COMRV[1:0] = 00
Write: If COMRV[1:0] = 00 and DBG not armed

Table 366. DBGADH field descriptions

Field	Description
7–0 Bits[15:8]	**Comparator Data High Compare Bits**— The Comparator data high compare bits control whether the selected comparator compares the data bus bits [15:8] to a logic one or logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

4.20.3.2.8.6 Debug comparator data low register (DBGADL)

Table 367. Debug comparator data low register (DBGADL)

Address: 0x002D

	7	6	5	4	3	2	1	0
R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0

Read: If COMRV[1:0] = 00
Write: If COMRV[1:0] = 00 and DBG not armed

Table 368. DBGADL field descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Data Low Compare Bits** — The Comparator data low compare bits control, whether the selected comparator compares the data bus bits [7:0] to a logic one or a logic zero. The comparator data compare bits are only used in comparison if the corresponding data mask bit is a logic 1. This register is available only for comparator A. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

4.20.3.2.8.7 Debug comparator data high mask register (DBGADHM)

Table 369. Debug comparator data high mask register (DBGADHM)

Address: 0x002E

	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Reset	0	0	0	0	0	0	0	0

Read: If COMRV[1:0] = 00
Write: If COMRV[1:0] = 00 and DBG not armed

Table 370. DBGADHM field descriptions

Field	Description
7–0 Bits[15:8]	**Comparator Data High Mask Bits** — The Comparator data high mask bits control whether the selected comparator compares the data bus bits [15:8] to the corresponding comparator data compare bits. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear 0 Do not compare corresponding data bit Any value of corresponding data bit allows match. 1 Compare corresponding data bit

4.20.3.2.8.8 Debug comparator data low mask register (DBGADLM)

Table 371. Debug comparator data low mask register (DBGADLM)

Address: 0x002F

	7	6	5	4	3	2	1	0
R W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0

Read: If COMRV[1:0] = 00
Write: If COMRV[1:0] = 00 and DBG not armed

Table 372. DBGADLM field descriptions

Field	Description
7–0 Bits[7:0]	**Comparator Data Low Mask Bits** — The Comparator data low mask bits control whether the selected comparator compares the data bus bits [7:0] to the corresponding comparator data compare bits. Data bus comparisons are only performed if the TAG bit in DBGACTL is clear 0 Do not compare corresponding data bit. Any value of corresponding data bit allows match 1 Compare corresponding data bit

4.20.4 Functional description

This section provides a complete functional description of the DBG module. If the part is in secure mode, the DBG module can generate breakpoints, but tracing is not possible.

4.20.4.1 9S12I128PIMV1 operation

Arming the DBG module by setting ARM in DBGC1 allows triggering the state sequencer, storing of data in the trace buffer, and generation of breakpoints to the CPU. The DBG module is made up of four main blocks, the comparators, control logic, the state sequencer, and the trace buffer.

The comparators monitor the bus activity of the CPU. All comparators can be configured to monitor address bus activity. Comparator A can also be configured to monitor data bus activity and mask out individual data bus bits during a compare. Comparators can be configured to use R/W and word/byte access qualification in the comparison. A match with a comparator register value can initiate a state sequencer transition to another state (see Figure 65). Either forced or tagged matches are possible. Using a forced match, a state sequencer transition can occur immediately on a successful match of system busses and comparator registers. While tagging at a comparator match, the instruction opcode is tagged, and only if the instruction reaches the execution stage of the instruction queue, can a state sequencer transition occur. In the case of a transition to Final state, bus tracing is triggered, and/or a breakpoint can be generated.

A state sequencer transition to final state (with associated breakpoint, if enabled) can be initiated by writing to the TRIG bit in the DBGC1 control register. The trace buffer is visible through a 2-byte window in the register address map and must be read out using standard 16-bit word reads.

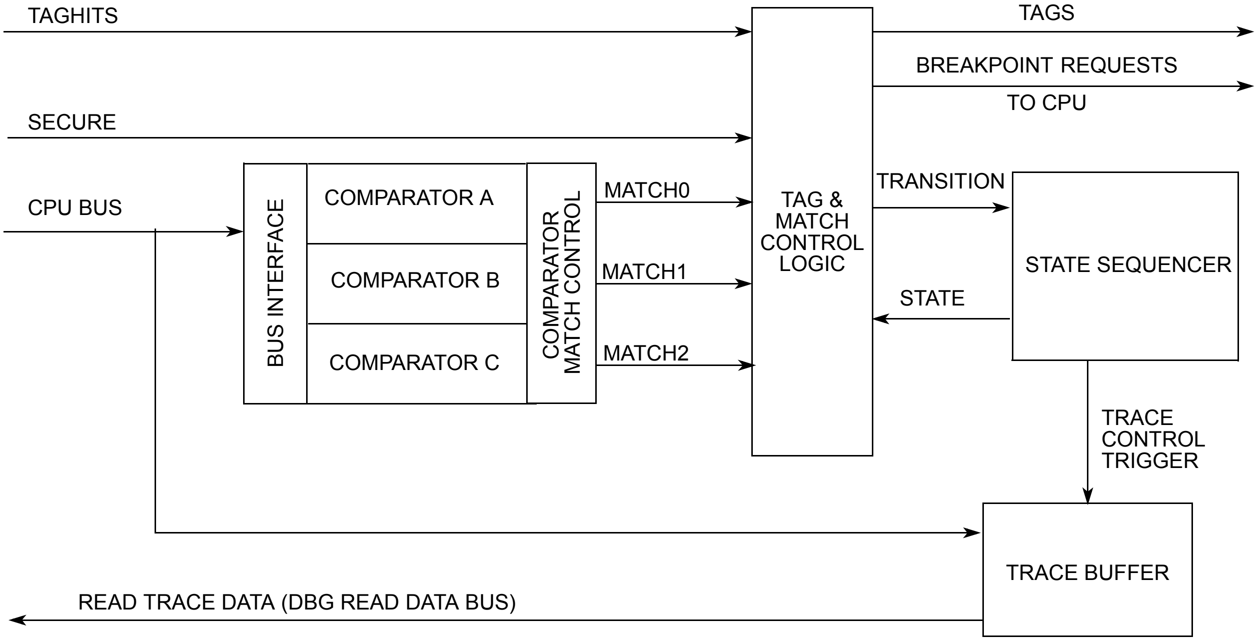


Figure 64. DBG overview

4.20.4.2 Comparator modes

The DBG contains three comparators, A, B and C. Each comparator compares the system address bus with the address stored in DBGXAH, DBGXAM, and DBGXAL. Furthermore, comparator A also compares the data buses to the data stored in DBGADH, DBGADL and allows masking of individual data bus bits.

All comparators are disabled in BDM and during BDM accesses. The comparator match control logic (see Figure 64) configures comparators to monitor the buses for an exact address or an address range, whereby either an access inside or outside the specified range generates a match condition. The comparator configuration is controlled by the control register contents and the range control by the DBGC2 contents.

A match can initiate a transition to another state sequencer state (see Section 4.20.4.4, “State sequence control"”). The comparator control register also allows the type of access to be included in the comparison through the use of the RWE, RW, SZE, and SZ bits. The RWE bit controls whether read or write comparison is enabled for the associated comparator and the RW bit selects either a read or write access for a valid match. Similarly the SZE and SZ bits allow the size of access (word or byte) to be considered in the compare. Only comparators A and B feature SZE and SZ.

The TAG bit in each comparator control register is used to determine the match condition. By setting TAG, the comparator qualifies a match with the output of opcode tracking logic, and a state sequencer transition occurs when the tagged instruction reaches the CPU execution stage. While tagging, the RW, RWE, SZE, and SZ bits and the comparator data registers are ignored; the comparator address register must be loaded with the exact opcode address.

If the TAG bit is clear (forced type match), a comparator match is generated when the selected address appears on the system address bus. If the selected address is an opcode address, the match is generated when the opcode is fetched from the memory, which precedes the instruction execution by an indefinite number of cycles due to instruction pipelining. For a comparator match of an opcode at an odd address when TAG = 0, the corresponding even address must be contained in the comparator register. Thus for an opcode at odd address (n), the comparator register must contain address (n–1).

Once a successful comparator match has occurred, the condition that caused the original match is not verified again on subsequent matches. If a particular data value is verified at a given address, this address may not still contain that data value when a subsequent match occurs.

Match[0, 1, 2] map directly to Comparators [A, B, C] respectively, except in range modes (see Section 4.20.3.2.4, “Debug control register2 (DBGC2)"). Comparator channel priority rules are described in the priority section (Section 4.20.4.3.4, “Channel priorities").

4.20.4.2.1 Single address comparator match

With range comparisons disabled, the match condition is an exact equivalence of address bus, with the value stored in the comparator address registers. Further qualification of the type of access (R/W, word/byte) and data bus contents is possible, depending on comparator channel.

4.20.4.2.1.1 Comparator C

Comparator C offers only address and direction (R/W) comparison. The exact address is compared, with the comparator address register loaded with address (n), a word access of address (n–1) also accesses (n) but does not cause a match.

Table 373. Comparator C access considerations

Condition for valid match	Comp C address	RWE	RW	Examples
Read and write accesses of ADDR[n]	ADDR[n][256]	0	X	LDAA ADDR[n] STAA #$BYTE ADDR[n]
Write accesses of ADDR[n]	ADDR[n]	1	0	STAA #$BYTE ADDR[n]
Read accesses of ADDR[n]	ADDR[n]	1	1	LDAA #$BYTE ADDR[n]

Notes

[256] A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match. The comparator address register must contain the exact address from the code.

4.20.4.2.1.2 Comparator B

Comparator B offers address, direction (R/W) and access size (word/byte) comparison. If the SZE bit is set, the access size (word or byte) is compared with the SZ bit value such that only the specified size of access causes a match. If configured for a byte access of a particular address, a word access covering the same address does not lead to match.

Assuming the access direction is not qualified (RWE=0), for simplicity, the size access considerations are shown in Table 374.

Table 374. Comparator B access size considerations

Condition for valid match	Comp B address	RWE	SZE	SZ8	Examples
Word and byte accesses of ADDR[n]	ADDR[n][257]	0	0	X	MOVB #$BYTE ADDR[n] MOVW #$WORD ADDR[n]
Word accesses of ADDR[n] only	ADDR[n]	0	1	0	MOVW #$WORD ADDR[n] LDD ADDR[n]
Byte accesses of ADDR[n] only	ADDR[n]	0	1	1	MOVB #$BYTE ADDR[n] LDAB ADDR[n]

Notes

[257] A word access of ADDR[n-1] also accesses ADDR[n] but does not generate a match. The comparator address register must contain the exact address from the code.

Access direction can also be used to qualify a match for Comparator B in the same way, as described for Comparator C in Table 373.

4.20.4.2.1.3 Comparator A

Comparator A offers address, direction (R/W), access size (word/byte), and data bus comparison. Table 375 lists access considerations with data bus comparison. On word accesses, the data byte of the lower address is mapped to DBGADH. Access direction can also be used to qualify a match for Comparator A in the same way as described for Comparator C in Table 373.

Table 375. Comparator A matches when accessing ADDR[n]

SZE	SZ	DBGADHM, DBGADLM	Access DH=DBGADH, DL=DBGADL	Comment
0	X	$0000	Byte Word	No databus comparison
0	X	$FF00	Byte, data(ADDR[n])=DH Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match data(ADDR[n])
0	X	$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match data(ADDR[n+1])
0	X	$00FF	Byte, data(ADDR[n])=X, data(ADDR[n+1])=DL	Possible unintended match
0	X	$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data(ADDR[n], ADDR[n+1])
0	X	$FFFF	Byte, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Possible unintended match
1	0	$0000	Word	No databus comparison
1	0	$00FF	Word, data(ADDR[n])=X, data(ADDR[n+1])=DL	Match only data at ADDR[n+1]
1	0	$FF00	Word, data(ADDR[n])=DH, data(ADDR[n+1])=X	Match only data at ADDR[n]
1	0	$FFFF	Word, data(ADDR[n])=DH, data(ADDR[n+1])=DL	Match data at ADDR[n] & ADDR[n+1]
1	1	$0000	Byte	No databus comparison
1	1	$FF00	Byte, data(ADDR[n])=DH	Match data at ADDR[n]

4.20.4.2.1.4 Comparator A data bus comparison NDB dependency

Comparator A features an NDB control bit, which allows data bus comparators to be configured to either trigger on equivalence or trigger on difference. This allows monitoring of a difference in the contents of an address location from an expected value.

When matching on an equivalence (NDB=0), each individual data bus bit position can be masked out by clearing the corresponding mask bit (DBGADHM/DBGADLM) so that it is ignored in the comparison. A match occurs when all data bus bits with corresponding mask bits set are equivalent. If all mask register bits are clear, then a match is based on the address bus only, the data bus is ignored.

When matching on a difference, mask bits can be cleared to ignore bit positions. A match occurs when any data bus bit with corresponding mask bit set is different. Clearing all mask bits causes all bits to be ignored and prevents a match because no difference can be detected. In this case, address bus equivalence does not cause a match.

Table 376. NDB and MASK bit dependency

NDB	DBGADHM[n] / DBGADLM[n]	Comment
0	0	Do not compare data bus bit.
0	1	Compare data bus bit. Match on equivalence.
1	0	Do not compare data bus bit.
1	1	Compare data bus bit. Match on difference.

4.20.4.2.2 Range comparisons

Using the AB comparator pair for a range comparison, the data bus can also be used for qualification by using the comparator A data registers. Furthermore, the DBGACTL RW and RWE bits can be used to qualify the range comparison on either a read or a write access. The corresponding DBGBCTL bits are ignored. The SZE and SZ control bits are ignored in range mode. The comparator A TAG bit is used to tag range comparisons. The comparator B TAG bit is ignored in range modes. For a range comparison using comparators A and B, both COMPEA and COMPEB must be set; to disable range comparisons both must be cleared. The comparator A BRK bit is used to for the AB range, the comparator B BRK bit is ignored in range mode.

When configured for range comparisons and tagging, the ranges are accurate only to word boundaries.

4.20.4.2.2.1 Inside range (CompA_Addr ≤ address ≤ CompB_Addr)

In the Inside Range comparator mode, comparator pair A and B can be configured for range comparisons. This configuration depends upon the control register (DBGC2). The match condition requires that a valid match for both comparators happens on the same bus cycle. A match condition on only one comparator is not valid. An aligned word access which straddles the range boundary is valid only if the aligned address is inside the range.

4.20.4.2.2.2 Outside range (address < CompA_Addr or address > CompB_Addr)

In the Outside Range comparator mode, comparator pair A and B can be configured for range comparisons. A single match condition on either of the comparators is recognized as valid. An aligned word access which straddles the range boundary is valid, only if the aligned address is outside the range.

Outside range mode in combination with tagging can be used to detect, if the opcode fetches are from an unexpected range. In forced match mode, the outside range match would typically be activated at any interrupt vector fetch or register access. This can be avoided by setting the upper range limit to $3FFFF or lower range limit to $00000 respectively.

4.20.4.3 Match modes (forced or tagged)

Match modes are used as qualifiers for a state sequencer change of state. The Comparator control register TAG bits select the match mode. The modes are described in the following sections.

4.20.4.3.1 Forced match

When configured for forced matching, a comparator channel match can immediately initiate a transition to the next state sequencer state, whereby the corresponding flags in DBGSR are set. The state control register for the current state determines the next state. Forced matches are typically generated 2-3 bus cycles after the final matching address bus cycle, independent of comparator RWE/RW settings. Furthermore, since opcode fetches occur several cycles before the opcode execution, a forced match of an opcode address typically precedes a tagged match at the same address.

4.20.4.3.2 Tagged match

If a CPU taghit occurs, a transition to another state sequencer state is initiated and the corresponding DBGSR flags are set. For a comparator related taghit to occur, the DBG must first attach tags to instructions as they are fetched from memory. When the tagged instruction reaches the execution stage of the instruction queue, a taghit is generated by the CPU. This can initiate a state sequencer transition.

4.20.4.3.3 Immediate trigger

Independent of comparator matches, it is possible to initiate a tracing session and/or breakpoint by writing to the TRIG bit in DBGC1. If configured for begin aligned tracing, this triggers the state sequencer into the Final state, if configured for end alignment, setting the TRIG bit disarms the module, ending the session and issues a forced breakpoint request to the CPU.

It is possible to set both TRIG and ARM simultaneously to generate an immediate trigger, independent of the current state of ARM.

4.20.4.3.4 Channel priorities

In case of simultaneous matches, the priority is resolved according to Table 377. The lower priority is suppressed. It is possible to miss a lower priority match, if it occurs simultaneously with a higher priority. The priorities described in Table 377 dictate that in the case of simultaneous matches, the match pointing to Final state has highest priority, followed by the lower channel number (0,1,2).

Table 377. Channel priorities

Priority	Source	Action
Highest	TRIG	Enter Final State
	Channel pointing to Final State	Transition to next state as defined by state control registers
	Match0 (force or tag hit)	Transition to next state as defined by state control registers
	Match1 (force or tag hit)	Transition to next state as defined by state control registers
Lowest	Match2 (force or tag hit)	Transition to next state as defined by state control registers

4.20.4.4 State sequence control

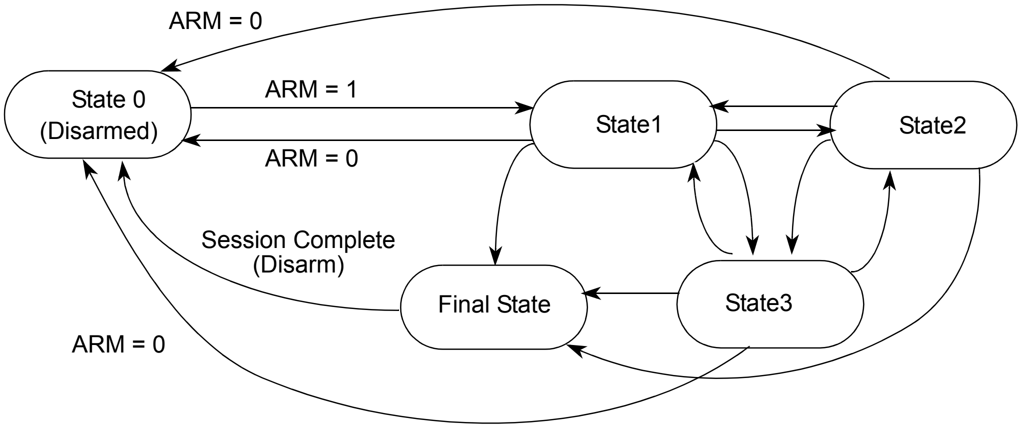


Figure 65. State sequencer diagram

The state sequencer allows a defined sequence of events to provide a trigger point for tracing of data in the trace buffer. Once the DBG module has been armed by setting the ARM bit in the DBGC1 register, the state 1 of the state sequencer is entered. Further transitions between the states are then controlled by the state control registers and channel matches. From Final state, the only permitted transition is back to the disarmed state 0. Transition between any of the states 1 to 3 is not restricted. Each transition updates the SSF[2:0] flags in DBGSR accordingly to indicate the current state. Alternatively, writing to the TRIG bit in DBGSC1, provides an immediate trigger independent of comparator matches.

Independent of the state sequencer, each comparator channel can be individually configured to generate an immediate breakpoint when a match occurs, through the use of the BRK bits in the DBGxCTL registers. It is possible to generate an immediate breakpoint on selected channels, while a state sequencer transition can be initiated by a match on other channels. If a debug session is ended by a match on a channel, the state sequencer transitions through Final state for a clock cycle to state 0. This is independent of tracing and breakpoint activity, and with tracing and breakpoints disabled, the state sequencer enters state 0 and the debug module is disarmed.

4.20.4.4.1 Final state

On entering Final state, a trigger may be issued to the trace buffer according to the trace alignment control, as defined by the TALIGN bit (see Section 4.20.3.2.3, “Debug trace control register (DBGTCR)"”). If the TSOURCE bit in DBGTCR is clear, then the trace buffer is disabled and the transition to Final state can only generate a breakpoint request. In this case or upon completion of a tracing session when tracing is enabled, the ARM bit in the DBGC1 register is cleared, returning the module to the disarmed state 0. If tracing is enabled, a breakpoint request can occur at the end of the tracing session. If neither tracing nor breakpoints are enabled, when the final state is reached, it returns automatically to state 0 and the debug module is disarmed.

4.20.4.5 Trace buffer operation

The trace buffer is a 64 lines deep by 20-bits wide RAM array. The DBG module stores trace information in the RAM array in a circular buffer format. The system accesses the RAM array through a register window (DBGTBH:DBGTBL) using 16-bit wide word accesses. After each complete 20-bit trace buffer line is read, an internal pointer into the RAM increments so that the next read receives fresh information. Data is stored in the format shown in Table 378 and Table 374. After each store the counter register DBGCNT is incremented. Tracing of CPU activity is disabled when the BDM is active. Reading the trace buffer while the DBG is armed, returns invalid data and the trace buffer pointer is not incremented.

4.20.4.5.1 Trace trigger alignment

Using the TALIGN bit (see Section 4.20.3.2.3, “Debug trace control register (DBGTCR)"), it is possible to align the trigger with the end or the beginning of a tracing session.

If End tracing is selected, tracing begins when the ARM bit in DBGC1 is set and State1 is entered; the transition to Final state signals the end of the tracing session. Tracing with Begin Trigger starts at the opcode of the trigger. Using End Trigger or when the tracing is initiated by writing to the TRIG bit while configured for Begin Trigger, tracing starts in the second cycle after the DBGC1 write cycle.

4.20.4.5.1.1 Storing with begin trigger

Storing with Begin Trigger, data is not stored in the Trace Buffer until the Final state is entered. Once the trigger condition is met, the DBG module remains armed until 64 lines are stored in the Trace Buffer. If the trigger is at the address of the change-of-flow instruction, the change of flow associated with the trigger is stored in the Trace Buffer. Using Begin Trigger together with tagging, if the tagged instruction is about to be executed, then the trace is started. Upon completion of the tracing session, the breakpoint is generated, thus the breakpoint does not occur at the tagged instruction boundary.

4.20.4.5.1.2 Storing with end trigger

Storing with End Trigger, data is stored in the Trace Buffer until the Final State is entered, at which point the DBG module becomes disarmed and no more data is stored. If the trigger is at the address of a change of flow instruction, the trigger event is not stored in the Trace Buffer.

4.20.4.5.2 Trace modes

Four trace modes are available. The mode is selected using the TRCMOD bits in the DBGTCR register. Tracing is enabled using the TSOURCE bit in the DBGTCR register. The modes are described in the following subsections.

4.20.4.5.2.1 Normal mode

In Normal mode, change of flow (COF) program counter (PC) addresses are stored.

COF addresses are defined as follows:

- Source address of taken conditional branches (long, short, bit-conditional, and loop primitives)
- Destination address of indexed JMP, JSR, and CALL instruction
- Destination address of RTI, RTS, and RTC instructions
- Vector address of interrupts, except for BDM vectors

LBRA, BRA, BSR, BGND, as well as non-indexed JMP, JSR, and CALL instructions are not classified as change of flow and are not stored in the trace buffer.

Stored information includes the full 18-bit address bus and information bits, which contains a source/destination bit to indicate whether the stored address was a source address or destination address.

NOTE

> When a COF instruction with destination address is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously, then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets executed after the interrupt service routine.
>
> In the following example, an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The BRN at the destination (SUB_1) is not executed until after the IRQ service routine, but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
        LDX     #SUB_1
MARK1   JMP     0,X                             ; IRQ interrupt occurs during execution of this
MARK2   NOP                                     ;

SUB_1   BRN     *                               ; JMP Destination address TRACE BUFFER ENTRY 1
                                                ; RTI Destination address TRACE BUFFER ENTRY 3
        NOP                                     ;
ADDR1   DBNE    A,PART5                         ; Source address TRACE BUFFER ENTRY 4

IRQ_ISR LDAB    #$F0                            ; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
        STAB    VAR_C1
        RTI                                     ;
```

The execution flow taking into account the IRQ is as follows

```
         LDX     #SUB_1
MARK1    JMP     0,X                     ;
IRQ_ISR  LDAB    #$F0                    ;
         STAB    VAR_C1
         RTI                             ;
SUB_1    BRN     *
         NOP                             ;
ADDR1    DBNE    A,PART5                 ;
```

4.20.4.5.2.2 Loop1 mode

Loop1 mode, similarly to Normal mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 mode is to prevent the Trace Buffer from being filled entirely with duplicate information from a looping construct, such as delays using the DBNE instruction or polling loops using BRSET/BRCLR instructions. Immediately after address information is placed in the Trace Buffer, the DBG module writes this value into a background register. This prevents consecutive duplicate address entries in the Trace Buffer resulting from repeated branches.

Loop1 mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these would most likely indicate a bug in the user’s code that the DBG module is designed to help find.

4.20.4.5.2.3 Detail mode

In Detail mode, address and data for all memory and register accesses is stored in the trace buffer. This mode is intended to supply additional information on indexed, indirect addressing modes where storing only the destination address would not provide all information required for a user to determine where the code is in error. This mode also features information bit storage to the trace buffer, for each address byte storage. The information bits indicate the size of access (word or byte) and the type of access (read or write).

When tracing in Detail mode, all cycles are traced except those when the CPU is either in a free or opcode fetch cycle.

4.20.4.5.2.4 Compressed pure PC mode

In Compressed Pure PC mode, the PC addresses of all executed opcodes, including where illegal opcodes are stored. A compressed storage format is used to increase the effective depth of the trace buffer. This is achieved by storing the lower order bits each time and using 2 information bits to indicate if a 64 byte boundary has been crossed, in which case the full PC is stored.

Each Trace Buffer row consists of 2 information bits and 18 PC address bits

NOTE:

When tracing is terminated using forced breakpoints, latency in breakpoint generation means that opcodes following the opcode causing the breakpoint can be stored to the trace buffer. The number of opcodes is dependent on program flow. This can be avoided by using tagged breakpoints.

4.20.4.5.3 Trace buffer organization (normal, loop1, detail modes)

ADRH, ADRM, ADRL denote address high, middle, and low byte respectively. The numerical suffix refers to the tracing count. The information format for Loop1 and Normal modes are identical. In Detail mode, the address and data for each entry are stored on consecutive lines, thus the maximum number of entries is 32. In this case, DBGCNT bits are incremented twice, once for the address line, and once for the data line, on each trace buffer entry. In Detail mode, CINF comprises of R/W and size access information (CRW and CSZ respectively).

Single byte data accesses in Detail mode are always stored to the low byte of the trace buffer (DATAL) and the high byte is cleared. When tracing word accesses, the byte at the lower address is always stored to trace buffer byte1 and the byte at the higher address is stored to byte0.

Table 378. Trace buffer organization (normal, loop1, detail modes)

Mode	Entry Number	4-bits	8-bits	8-bits
		Field 2	Field 1	Field 0
Detail Mode	Entry 1	CINF1,ADRH1	ADRM1	ADRL1
		0	DATAH1	DATAL1
	Entry 2	CINF2,ADRH2	ADRM2	ADRL2
		0	DATAH2	DATAL2
Normal/Loop1 Modes	Entry 1	PCH1	PCM1	PCL1
	Entry 2	PCH2	PCM2	PCL2

4.20.4.5.3.1 Information bit organization

The format of the bits is dependent upon the active trace mode as described by the following.

4.20.4.5.3.1.1 Field2 bits in detail mode

Table 379. Field2 bits in detail mode

Bit 3	Bit 2	Bit 1	Bit 0
CSZ	CRW	ADDR[17]	ADDR[16]

In Detail Mode, the CSZ and CRW bits indicate the type of access being made by the CPU.

Table 380. Field descriptions

Bit	Description
3 CSZ	**Access Type Indicator**— This bit indicates if the access was a byte or word size when tracing in Detail mode 0 Word Access 1 Byte Access
2 CRW	**Read Write Indicator** — This bit indicates if the corresponding stored address corresponds to a read or write access when tracing in Detail mode. 0 Write Access 1 Read Access
1 ADDR[17]	**Address Bus bit 17**— Corresponds to system address bus bit 17.
0 ADDR[16]	**Address Bus bit 16**— Corresponds to system address bus bit 16.

4.20.4.5.3.1.2 Field2 bits in normal and loop1 modes

Table 381. Information bits PCH

Bit 3	Bit 2	Bit 1	Bit 0
CSD	CVA	PC17	PC16

Table 382. PCH field descriptions

Bit	Description
3 CSD	**Source Destination Indicator** — In Normal and Loop1 mode, this bit indicates if the corresponding stored address is a source or destination address. This bit has no meaning in Compressed Pure PC mode. 0 Source Address 1 Destination Address
2 CVA	**Vector Indicator** — In Normal and Loop1 mode, this bit indicates if the corresponding stored address is a vector address. Vector addresses are destination addresses, thus if CVA is set, then the corresponding CSD is also set. This bit has no meaning in Compressed Pure PC mode. 0 Non-Vector destination address 1 Vector destination address
1 PC17	**Program Counter bit 17**— In Normal and Loop1 mode, this bit corresponds to program counter bit 17.
0 PC16	**Program Counter bit 16**— In Normal and Loop1 mode, this bit corresponds to program counter bit 16.

4.20.4.5.4 Trace buffer organization (compressed pure PC mode)

Table 383. Trace buffer organization example (compressed pure PC mode)

Mode	Line Number	2-bits	6-bits	6-bits	6-bits
		Field 3	Field 2	Field 1	Field 0
Compressed Pure PC Mode	Line 1	00	PC1 (Initial 18-bit PC Base Address)		
	Line 2	11	PC4	PC3	PC2
	Line 3	01	0	0	PC5
	Line 4	00	PC6 (New 18-bit PC Base Address)		
	Line 5	10	0	PC8	PC7
	Line 6	00	PC9 (New 18-bit PC Base Address)		

4.20.4.5.4.0.1 Field3 bits in compressed pure PC modes

Table 384. Compressed pure PC mode field 3 information bit encoding

INF1	INF0	TRACE BUFFER ROW CONTENT
0	0	Base PC address TB[17:0] contains a full PC[17:0] value
0	1	Trace Buffer[5:0] contain incremental PC relative to base address zero value
1	0	Trace Buffer[11:0] contain next 2 incremental PCs relative to base address zero value
1	1	Trace Buffer[17:0] contain next 3 incremental PCs relative to base address zero value

Each time PC[17:6] differs form the previous base PC[17:6], a new base address is stored. The base address zero value is the lowest address in the 64 address range.

The first line of the trace buffer always gets a base PC address; this applies also on rollover.

4.20.4.5.5 Reading data from trace buffer

The data stored in the Trace Buffer can be read provided the DBG module is not armed, is configured for tracing (TSOURCE bit is set), and the system not secured. When the ARM bit is written to 1 the trace buffer, it is locked to prevent reading. The trace buffer can only be unlocked for reading by a single aligned word write to DBGTB when the module is disarmed.

The Trace Buffer can only be read through the DBGTB register using aligned word reads. Any byte or misaligned reads return 0 and does not cause the trace buffer pointer to increment to the next trace buffer address. The Trace Buffer data is read out first-in first-out. By reading CNT in DBGCNT, the number of valid lines can be determined. DBGCNT does not decrement as data is read.

While reading, an internal pointer is used to determine the next line to be read. After a tracing session, the pointer points to the oldest data entry. If no overflow has occurred, the pointer points to line 0, otherwise it points to the line with the oldest entry. In compressed Pure PC mode on rollover, the line with the oldest data entry may also contain newer data entries in fields 0 and 1. If rollover is indicated by the TBF bit, the line status must be decoded using the INF bits in field3 of that line. If both INF bits are clear, the line contains only entries from before the last rollover.
If INF0=1, field 0 contains post rollover data but fields 1 and 2 contain pre rollover data.
If INF1=1, fields 0 and 1 contain post rollover data but field 2 contains pre rollover data.

The pointer is initialized by each aligned write to DBGTBH to point to the oldest data again. This enables an interrupted trace buffer read sequence to be easily restarted from the oldest data entry.

The least significant word of line is read out first. This corresponds to the fields 1 and 0 of Table 378. The next word read returns field 2 in the least significant bits [3:0] and “0” for bits [15:4].

Reading the Trace Buffer while the DBG module is armed, returns invalid data and no shifting of the RAM pointer occurs.

4.20.4.5.6 Trace buffer reset state

The Trace Buffer contents and DBGCNT bits are not initialized by a system reset. Should a system reset occur, immediately before the reset occurred, the trace session information can be read out and the number of valid lines in the trace buffer is indicated by DBGCNT. The internal pointer to the current trace buffer address is initialized by unlocking the trace buffer and points to the oldest valid data, even if a reset occurred during the tracing session. To read the trace buffer after a reset, TSOURCE must be set, otherwise the trace buffer reads as all zeroes. Generally, debugging occurrences of system resets are best handled using end trigger alignment, since the reset may occur before the trace trigger, which in the begin trigger alignment case, means no information would be stored in the trace buffer.

The Trace Buffer contents and DBGCNT bits are undefined following a POR.

NOTE

An external pin RESET that occurs simultaneous to a trace buffer entry can, in very few cases, lead to either that entry being corrupted, or the first entry of the session being corrupted. In such cases, the other contents of the trace buffer still contain valid tracing information. The case occurs when the reset assertion coincides with the trace buffer entry clock edge.

4.20.4.6 Tagging

A tag follows program information as it advances through the instruction queue. When a tagged instruction reaches the head of the queue, a tag hit occurs and can initiate a state sequencer transition.

Each comparator control register features a TAG bit, which controls whether the comparator match causes a state sequencer transition immediately or tags the opcode at the matched address. If a comparator is enabled for tagged comparisons, the address stored in the comparator match address registers must be an opcode address.

Using Begin Trigger together with tagging, if the tagged instruction is about to be executed, the transition to the next state sequencer state occurs. If the transition is to the Final state, tracing is started. Only upon completion of the tracing session can a breakpoint be generated. Using End alignment, when the tagged instruction is about to be executed and the next transition is to Final state, a breakpoint is generated immediately, before the tagged instruction is carried out.

R/W monitoring, access size (SZ) monitoring and data bus monitoring are not useful if tagging is selected, since the tag is attached to the opcode at the matched address and is not dependent on the data bus nor on ta type of access. These bits are ignored if tagging is selected.

When configured for range comparisons and tagging, the ranges are accurate only to word boundaries. Tagging is disabled when the BDM becomes active.

4.20.4.7 Breakpoints

It is possible to generate breakpoints from channel transitions to final state or use software to write to the TRIG bit in the DBGC1 register.

4.20.4.7.1 Breakpoints from comparator channels

Breakpoints can be generated when the state sequencer transitions to the Final state. If configured for tagging, the breakpoint is generated when the tagged opcode reaches the execution stage of the instruction queue.

If a tracing session is selected by the TSOURCE bit, breakpoints are requested when the tracing session has completed. If Begin aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace (see Table 385). If no tracing session is selected, breakpoints are requested immediately. If the BRK bit is set, the associated breakpoint is generated immediately independent of tracing trigger alignment.

Table 385. Breakpoint setup for CPU breakpoints

BRK	TALIGN	DBGBRK	Breakpoint Alignment
0	0	0	Fill Trace Buffer until trigger then disarm (no breakpoints)
0	0	1	Fill Trace Buffer until trigger, then breakpoint request occurs
0	1	0	Start Trace Buffer at trigger (no breakpoints)
0	1	1	Start Trace Buffer at trigger A breakpoint request occurs when Trace Buffer is full
1	x	1	Terminate tracing and generate breakpoint immediately on trigger
1	x	0	Terminate tracing immediately on trigger

4.20.4.7.2 Breakpoints generated via the TRIG bit

If a TRIG triggers occur, the Final state is entered, where the tracing trigger alignment is defined by the TALIGN bit. If a tracing session is selected by the TSOURCE bit, breakpoints are requested when the tracing session has completed. If Begin aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace (see Table 385). If no tracing session is selected, breakpoints are requested immediately. TRIG breakpoints are possible with a single write to DBGC1, setting ARM and TRIG simultaneously.

4.20.4.7.3 Breakpoint priorities

If a TRIG trigger occurs after Begin aligned tracing has already started, then the TRIG no longer has an effect. When the associated tracing session is complete, the breakpoint occurs. Similarly, if a TRIG is followed by a subsequent comparator channel match, it has no effect, since tracing has already started. If a forced SWI breakpoint coincides with a BGND in user code with BDM enabled, then the BDM is activated by the BGND and the breakpoint to SWI is suppressed.

4.20.4.7.3.1 DBG breakpoint priorities and BDM interfacing

Breakpoint operation is dependent on the state of the BDM module. If the BDM module is active, the CPU is executing out of BDM firmware. Comparator matches and associated breakpoints are disabled. In addition, while executing a BDM TRACE command, tagging into BDM is disabled. If BDM is not active, the breakpoint gives priority to BDM requests over SWI requests if the breakpoint happens to coincide with a SWI instruction in user code. On returning from BDM, the SWI from user code gets executed.

Table 386. Breakpoint mapping summary

DBGBRK	BDM bit (DBGC1[4])	BDM enabled	BDM active	Breakpoint mapping
0	X	X	X	No Breakpoint
1	0	X	0	Breakpoint to SWI
X	X	1	1	No Breakpoint
1	1	0	X	Breakpoint to SWI
1	1	1	0	Breakpoint to BDM

BDM cannot be entered from a breakpoint unless the ENABLE bit is set in the BDM. If entry to BDM via a BGND instruction is attempted and the ENABLE bit in the BDM is cleared, the CPU actually executes the BDM firmware code, checks the ENABLE, and returns if ENABLE is not set. If not serviced by the monitor, the breakpoint is re-asserted when the BDM returns to normal CPU flow.

If the comparator register contents coincide with the SWI/BDM vector address, an SWI in user code could coincide with a DBG breakpoint. The CPU ensures that BDM requests have a higher priority than SWI requests. Returning from the BDM/SWI service routine, care must be taken to avoid a repeated breakpoint at the same address.

Should a tagged or forced breakpoint coincide with a BGND in user code, the instruction that follows the BGND instruction is the first instruction executed when normal program execution resumes.

NOTE

When program control returns from a tagged breakpoint using an RTI or BDM GO command without program counter modification, it returns to the instruction whose tag generated the breakpoint. To avoid a repeated breakpoint at the same location, reconfigure the DBG module in the SWI routine, if configured for an SWI breakpoint, or over the BDM interface by executing a TRACE command before the GO, to increment the program flow past the tagged instruction.

4.20.5 Application information

4.20.5.1 State machine scenarios

Defining the state control registers as SCR1,SCR2, SCR3, and M0,M1,M2 as matches on channels 0, 1, 2 respectively. SCR encoding supported by S12SDBGV1 are shown in black. SCR encoding supported only in S12SDBGV2 are shown in red. For backwards compatibility the new scenarios use a 4th bit in each SCR register. Thus the existing encoding for SCRx[2:0] is not changed.

4.20.5.2 Scenario 1

A trigger is generated if a given sequence of 3 code events is executed.

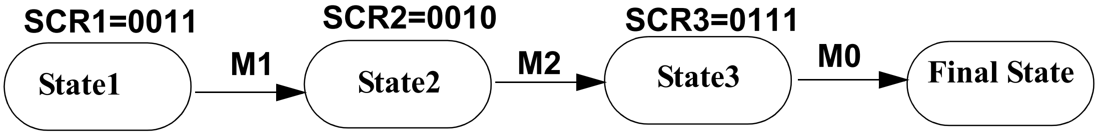


Figure 66. Scenario 1

Scenario 1 is possible with S12SDBGV1 SCR encoding.

4.20.5.3 Scenario 2

A trigger is generated if a given sequence of 2 code events is executed.

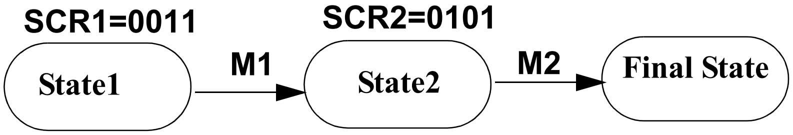


Figure 67. Scenario 2a

A trigger is generated if a given sequence of 2 code events is executed, whereby the first event is entry into a range (COMPA, COMPB configured for range mode). M1 is disabled in range modes.

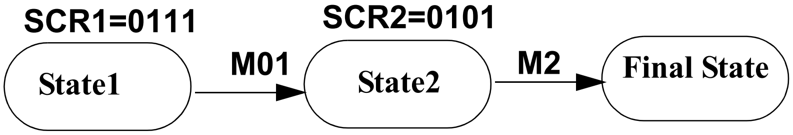


Figure 68. Scenario 2b

A trigger is generated if a given sequence of 2 code events is executed, whereby the second event is entry into a range (COMPA, COMPB configured for range mode).

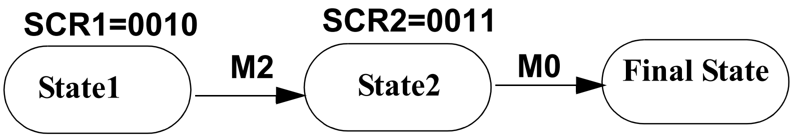


Figure 69. Scenario 2c

All 3 scenarios 2a, 2b, 2c are possible with the S12SDBGV1 SCR encoding.

4.20.5.4 Scenario 3

A trigger is generated immediately when one of up to 3 given events occurs.

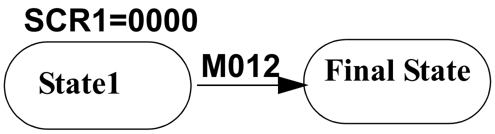


Figure 70. Scenario 3

Scenario 3 is possible with S12SDBGV1 SCR encoding.

4.20.5.5 Scenario 4

Trigger if a sequence of 2 events is carried out in an incorrect order. Event A must be followed by event B, and event B must be followed by event A. 2 consecutive occurrences of event A without an intermediate event B causes a trigger. Similarly 2 consecutive occurrences of event B without an intermediate event A causes a trigger. This is possible by using CompA and CompC to match on the same address as shown.

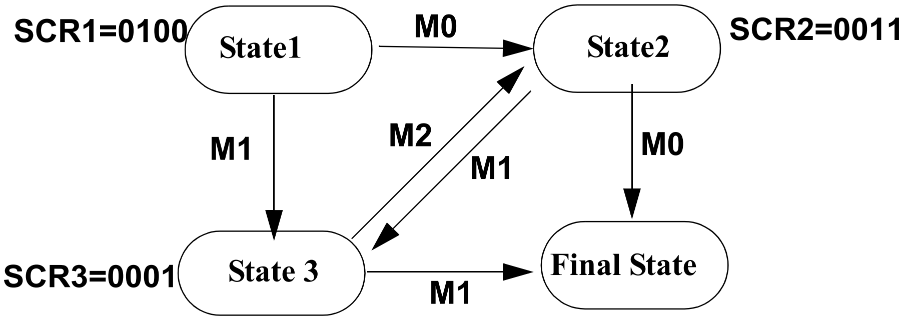


Figure 71. Scenario 4a

This scenario is currently not possible using 2 comparators only. S12SDBGV2 makes it possible with 2 comparators, State 3 allows a M0 to return to state 2, while a M2 leads to Final state, as shown in Figure 72.

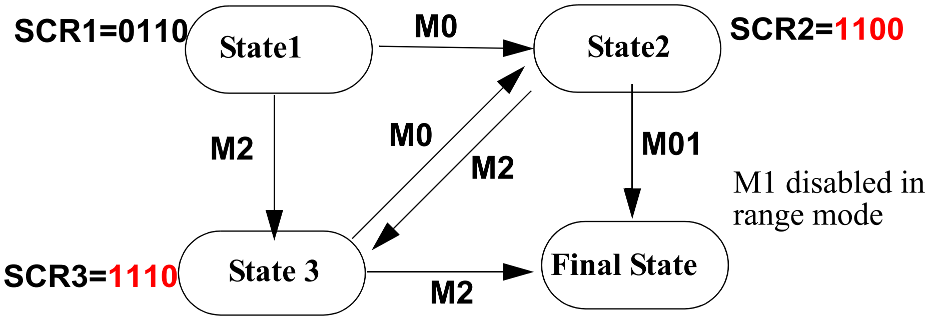


Figure 72. Scenario 4b (with 2 comparators)

The advantage of using only 2 channels is that range comparisons can now be included (channel0).

This however violates the S12SDBGV1 specification, which states that a match leading to final state always has priority, in case of a simultaneous match, while priority is also given to the lowest channel number. For the S12SDBG, the corresponding CPU priority decoder is removed to support this, such that on simultaneous taghits, taghits pointing to Final state have highest priority. If no taghit points to Final state, then the lowest channel number has priority. With the above encoding from State3, the CPU and DBG would break on a simultaneous M0/M2.

4.20.5.6 Scenario 5

Trigger if following event A, event C precedes event B. i.e... the expected execution flow is A->B->C.

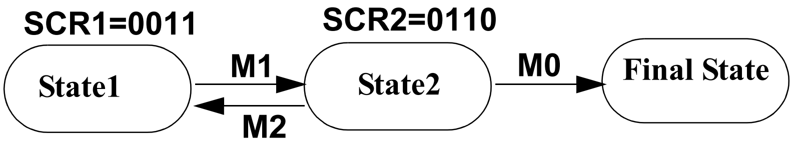


Figure 73. Scenario 5

Scenario 5 is possible with the S12SDBGV1 SCR encoding.

4.20.5.7 Scenario 6

Trigger if event A occurs twice in succession before any of 2 other events (BC) occur. This scenario is not possible using the S12SDBGV1 SCR encoding. S12SDBGV2 includes additions shown in red. The change in SCR1 encoding also has the advantage that a State1->State3 transition using M0 is now possible. This is advantageous because range and data bus comparisons use channel 0 only.

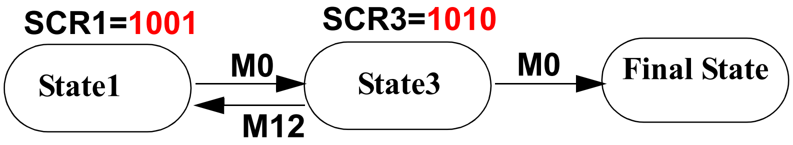


Figure 74. Scenario 6

4.20.5.8 Scenario 7

Trigger when a series of 3 events are executed out of order. Specifying the event order as M1, M2, M0 to run in loops (120120120). Any deviation from that order should trigger. This scenario is not possible using the S12SDBGV1 SCR encoding, because OR possibilities are very limited in the channel encoding. By adding OR forks as shown in red, this scenario is possible.

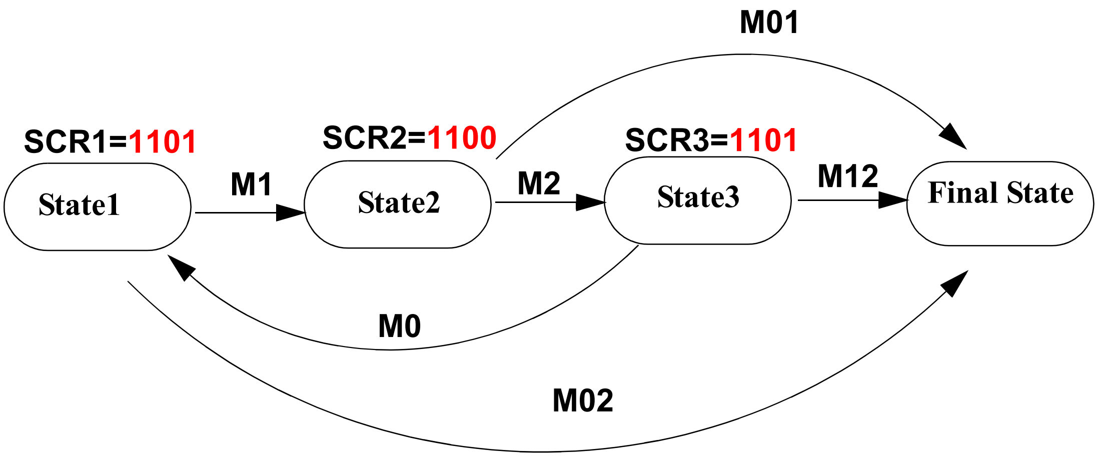


Figure 75. Scenario 7

On simultaneous matches the lowest channel number has priority, so with this configuration the forking from State1 has the peculiar effect that a simultaneous match0/match1 transitions to Final state, but a simultaneous match2/match1transitions to state2.

4.20.5.9 Scenario 8

Trigger when a routine/event at M2 follows either M1 or M0.

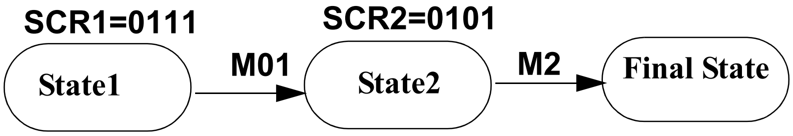


Figure 76. Scenario 8a

Trigger when an event M2 is followed by either an event M0 or event M1

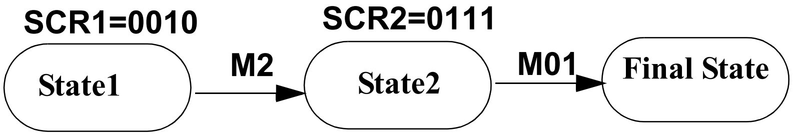


Figure 77. Scenario 8b

Scenario 8a and 8b are possible with the S12SDBGV1 and S12SDBGV2 SCR encoding.

4.20.5.10 Scenario 9

Trigger when a routine/event at A (M2) does not follow either B or C (M1 or M0) before they are executed again. This cannot be realized with the S12SDBGV1 SCR encoding, due to OR limitations. By changing the SCR2 encoding as shown in red this scenario becomes possible.

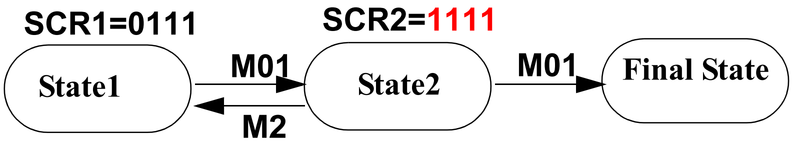


Figure 78. Scenario 9

4.20.5.11 Scenario 10

Trigger if an event M0 occurs following up to two successive M2 events, without the resetting event M1. As shown, up to 2 consecutive M2 events are allowed, whereby a reset to State1 is possible, after either one or two M2 events. If an event M0 occurs following the second M2, before M1 resets to State1, a trigger is generated. Configuring CompA and CompC the same, it is possible to generate a breakpoint on the third consecutive occurrence of event M0 without a reset on M1.

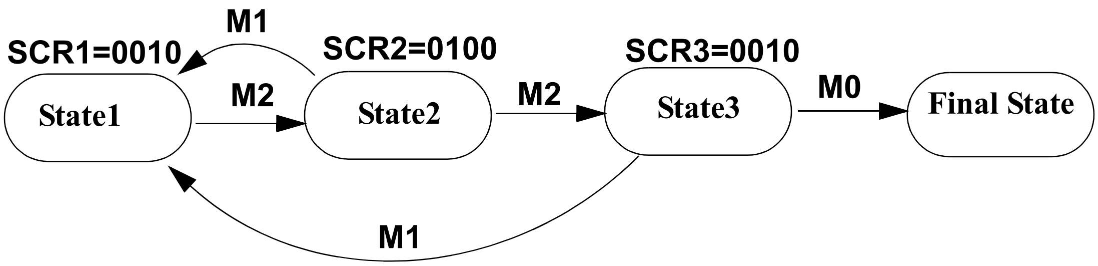


Figure 79. Scenario 10a

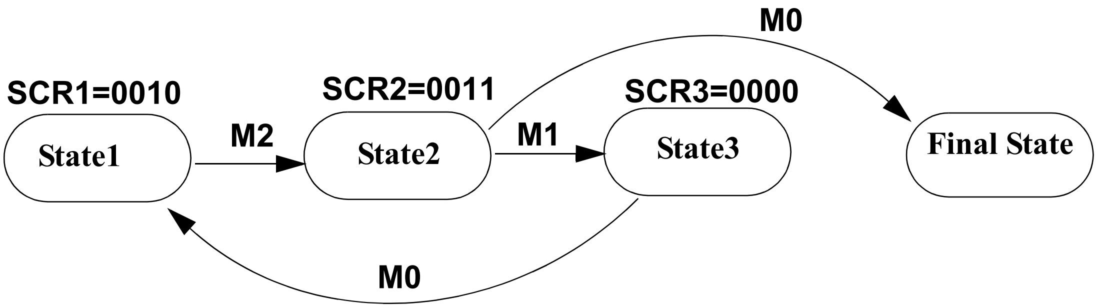


Figure 80. Scenario 10b

Scenario 10b shows the case that after M2, an M1 must occur before M0. Starting from a particular point in code, event M2 must always be followed by M1 before M0. If after any M2 event, M0 occurs before M1, then a trigger is generated.

4.21 MCU - security (S12XS9S12I128PIMV1V2)

4.21.1 Introduction

This specification describes the function of the security mechanism in the S12I chip family (9S12I128PIMV1).

NOTE

No security feature is absolutely secure. However, NXP’s strategy is to make reading or copying the FLASH and/or EEPROM difficult for unauthorized users.

4.21.1.1 Features

The user must be reminded that part of the security must lie with the application code. An extreme example would be application code that dumps the contents of the internal memory. This would defeat the purpose of security. At the same time, the user may also wish to put a backdoor in the application program. An example of this is the user downloads a security key through the SCI, which allows access to a programming routine that updates parameters stored in another section of the Flash memory.

The security features of the S12I chip family (in secure mode) are:

- Protect the content of non-volatile memories (Flash, EEPROM)
- Execution of NVM commands is restricted
- Disable access to internal memory via background debug module (BDM)

4.21.1.2 Modes of operation

Table 387 gives an overview over availability of security relevant features in unsecure and secure modes.

Table 387. Feature availability in unsecure and secure modes on S12XS

	Unsecure mode						Secure mode					
	NS	SS	NX	ES	EX	ST	NS	SS	NX	ES	EX	ST
Flash Array Access	4	4					4	4				
EEPROM Array Access	4	4					4	4				
NVM Commands	4[258]	4					4[258]	4[258]				
BDM	4	4					—	4[259]				
DBG Module Trace	4	4					—	—				

Notes

[258] Restricted NVM command set only. Refer to the NVM wrapper block guides for detailed information.

[259] BDM hardware commands restricted to peripheral registers only.

4.21.1.3 Securing the microcontroller

Once the user has programmed the Flash and EEPROM, the chip can be secured by programming the security bits located in the options/security byte in the Flash memory array. These non-volatile bits will keep the device secured through reset and power-down.

The options/security byte is located at address 0xFF0F (= global address 0x7F_FF0F) in the Flash memory array. This byte can be erased and programmed like any other Flash location. Two bits of this byte are used for security (SEC[1:0]). On devices which have a memory page window, the Flash options/security byte is also available at address 0xBF0F by selecting page 0x3F with the PPAGE register. The contents of this byte are copied into the Flash security register (FSEC) during a reset sequence.

Table 388. Flash options/security byte

	7	6	5	4	3	2	1	0
0xFF0F	KEYEN1	KEYEN0	NV5	NV4	NV3	NV2	SEC1	SEC0

The meaning of the bits KEYEN[1:0] is shown in Table 389. Refer to Section 4.21.1.5.1, “Unsecuring the MCU using the backdoor key access" for more information.

Table 389. Backdoor key access enable bits

KEYEN[1:0]	Backdoor key access enabled
00	0 (disabled)
01	0 (disabled)
10	1 (enabled)
11	0 (disabled)

The meaning of the security bits SEC[1:0] is shown in Table 390. For security reasons, the state of device security is controlled by two bits. To put the device in unsecured mode, these bits must be programmed to SEC[1:0] = ‘10’. All other combinations put the device in a secured mode. The recommended value to put the device in secured state is the inverse of the unsecured state, i.e. SEC[1:0] = ‘01’.

Table 390. Security bits

SEC[1:0]	Security state
00	1 (secured)
01	1 (secured)
10	0 (unsecured)
11	1 (secured)

NOTE

Refer to the Flash block guide for actual security configuration (in section “Flash Module Security”).

4.21.1.4 Operation of the secured microcontroller

By securing the device, unauthorized access to the EEPROM and Flash memory contents can be prevented. However, it must be understood that the security of the EEPROM and Flash memory contents also depends on the design of the application program. For example, if the application has the capability of downloading code through a serial port and then executing that code (e.g. an application containing bootloader code), then this capability could potentially be used to read the EEPROM and Flash memory contents, even when the microcontroller is in the secure state. In this example, the security of the application could be enhanced by requiring a challenge/response authentication before any code can be downloaded.

Secured operation has the following effects on the microcontroller:

4.21.1.4.1 Normal single chip mode (NS)

- Background debug module (BDM) operation is completely disabled.
- Execution of Flash and EEPROM commands is restricted. Refer to the NVM block guide for details.
- Tracing code execution using the DBG module is disabled.

4.21.1.4.2 Special single chip mode (SS)

- BDM firmware commands are disabled.
- BDM hardware commands are restricted to the register space.
- Execution of Flash and EEPROM commands is restricted. Refer to the NVM block guide for details.
- Tracing code execution using the DBG module is disabled.

Special single chip mode means BDM is active after reset. The availability of BDM firmware commands depends on the security state of the device. The BDM secure firmware first performs a blank check of both the Flash memory and the EEPROM. If the blank check succeeds, security will be temporarily turned off and the state of the security bits in the appropriate Flash memory location can be changed.

If the blank check fails, security will remain active, only the BDM hardware commands will be enabled, and the accessible memory space is restricted to the peripheral register area. This will allow the BDM to be used to erase the EEPROM and Flash memory without giving access to their contents. After erasing both Flash memory and EEPROM, another reset into special single chip mode will cause the blank check to succeed and the options/security byte can be programmed to “unsecured” state via BDM.

While the BDM is executing the blank check, the BDM interface is completely blocked, which means that all BDM commands are temporarily blocked.

4.21.1.5 Unsecuring the microcontroller

Unsecuring the microcontroller can be done by three different methods:

1. Backdoor key access
2. Reprogramming the security bits
3. Complete memory erase (special modes)

4.21.1.5.1 Unsecuring the MCU using the backdoor key access

In Normal modes (single chip and expanded), security can be temporarily disabled using the backdoor key access method. This method requires that:

- The backdoor key at 0xFF00–0xFF07 (= global addresses 0x7F_FF00–0x7F_FF07) has been programmed to a valid value.
- The KEYEN[1:0] bits within the Flash options/security byte select ‘enabled’.
- In single chip mode, the application program programmed into the microcontroller must be designed to have the capability to write to the backdoor key locations.

The backdoor key values themselves would not normally be stored within the application data, which means the application program would have to be designed to receive the backdoor key values from an external source (e.g. through a serial port).

The backdoor key access method allows debugging of a secured microcontroller without having to erase the Flash. This is particularly useful for failure analysis.

NOTE

No word of the backdoor key is allowed to have the value 0x0000 or 0xFFFF.

4.21.1.6 Reprogramming the security bits

In normal single chip mode (NS), security can also be disabled by erasing and reprogramming the security bits within Flash options/security byte to the unsecured value. Because the erase operation will erase the entire sector from 0xFE00–0xFFFF (0x7F_FE00–0x7F_FFFF), the backdoor key and the interrupt vectors will also be erased; this method is not recommended for normal single chip mode. The application software can only erase and program the Flash options/security byte if the Flash sector containing the Flash options/security byte is not protected (see Flash protection). Flash protection is a useful means of preventing this method. The microcontroller will enter the unsecured state after the next reset following the programming of the security bits to the unsecured value.

This method requires that:

- The application software previously programmed into the microcontroller has been designed to have the capability to erase and program the Flash options/security byte, or security is first disabled using the backdoor key method, allowing BDM to be used to issue commands to erase and program the Flash options/security byte.
- The Flash sector containing the Flash options/security byte is not protected.

4.21.1.7 Complete memory erase (special modes)

The microcontroller can be unsecured in special modes by erasing the entire EEPROM and Flash memory contents.

When a secure microcontroller is reset into special single chip mode (SS), the BDM firmware verifies whether the EEPROM and Flash memory are erased. If any EEPROM or Flash memory address is not erased, only BDM hardware commands are enabled. BDM hardware commands can then be used to write to the EEPROM and Flash registers to mass erase the EEPROM and all Flash memory blocks.

When next resetting into special single chip mode, the BDM firmware will again verify whether all EEPROM and Flash memory are erased. This being the case, it will enable all BDM commands, allowing the Flash options/security byte to be programmed to the unsecured value. The security bits SEC[1:0] in the Flash security register will indicate the unsecure state following the next reset.

4.22 Background debug module (9S12I128PIMV1)

4.22.1 Introduction

This section describes the functionality of the background debug module (BDM) sub-block of the HCS12S core platform.

The background debug module (BDM) sub-block is a single-wire, background debug system, implemented in on-chip hardware for minimal CPU intervention. All interfacing with the BDM is done via the BKGD pin.

The BDM has enhanced capability for maintaining synchronization between the target and host while allowing more flexibility in clock rates. This includes a sync signal to determine the communication rate and a handshake signal to indicate when an operation is complete. The system is backwards compatible to the BDM of the S12 family with the following exceptions:

- TAGGO command not supported by S12SBDM
- External instruction tagging feature is part of the DBG module
- S12SBDM register map and register content modified
- Family ID readable from BDM ROM at global address 0x3_FF0F in active BDM
(value for devices with HCS12S core is 0xC2)
- Clock switch removed from BDM (CLKSW bit removed from BDMSTS register)

4.22.1.1 Features

The BDM includes these distinctive features:

- Single-wire communication with host development system
- Enhanced capability for allowing more flexibility in clock rates
- SYNC command to determine communication rate
- GO_UNTIL[266] command
- Hardware handshake protocol to increase the performance of the serial communication
- Active out of reset in special single chip mode
- Nine hardware commands using free cycles, if available, for minimal CPU intervention
- Hardware commands not requiring active BDM
- 14 firmware commands execute from the standard BDM firmware lookup table
- Software control of BDM operation during Wait mode
- When secured, hardware commands are allowed to access the register space in special single chip mode, if the Flash erase tests fail
- Family ID readable from BDM ROM at global address 0x3_FF0F in active BDM
(value for devices with HCS12S core is 0xC2)
- BDM hardware commands are operational until system Stop mode is entered

4.22.1.2 Modes of operation

BDM is available in all operating modes, but must be enabled before firmware commands are executed. Some systems may have a control bit that allows suspending the function during background debug mode.

4.22.1.2.1 Regular run modes

All of these operations refer to the part in run mode and not being secured. The BDM does not provide controls to conserve power during run mode.

- Normal modes - General operation of the BDM is available and operates the same in all normal modes
- Special single chip mode - In special single chip mode, background operation is enabled and active out of reset. This allows programming a system with blank memory

4.22.1.2.2 Secure mode operation

If the device is in secure mode, the operation of the BDM is reduced to a small subset of its regular run mode operation. Secure operation prevents access to Flash other than allowing erasure. For more information, see Section 4.22.4.1, “Security".

4.22.1.2.3 Low-power modes

The BDM can be used until Stop mode is entered. When CPU is in Wait mode, all BDM firmware commands as well as the hardware BACKGROUND command cannot be used and are ignored. In this case, the CPU can not enter BDM active mode, and only hardware read and write commands are available. Also, the CPU can not enter a Low Power mode (stop or wait) during BDM active mode.

In Stop mode, the BDM clocks are stopped. When BDM clocks are disabled and Stop mode is exited, the BDM clocks will restart and BDM will have a soft reset (clearing the instruction register, any command in progress and disable the ACK function). The BDM is now ready to receive a new command.

4.22.1.3 Block diagram

A block diagram of the BDM is shown in Figure 81.

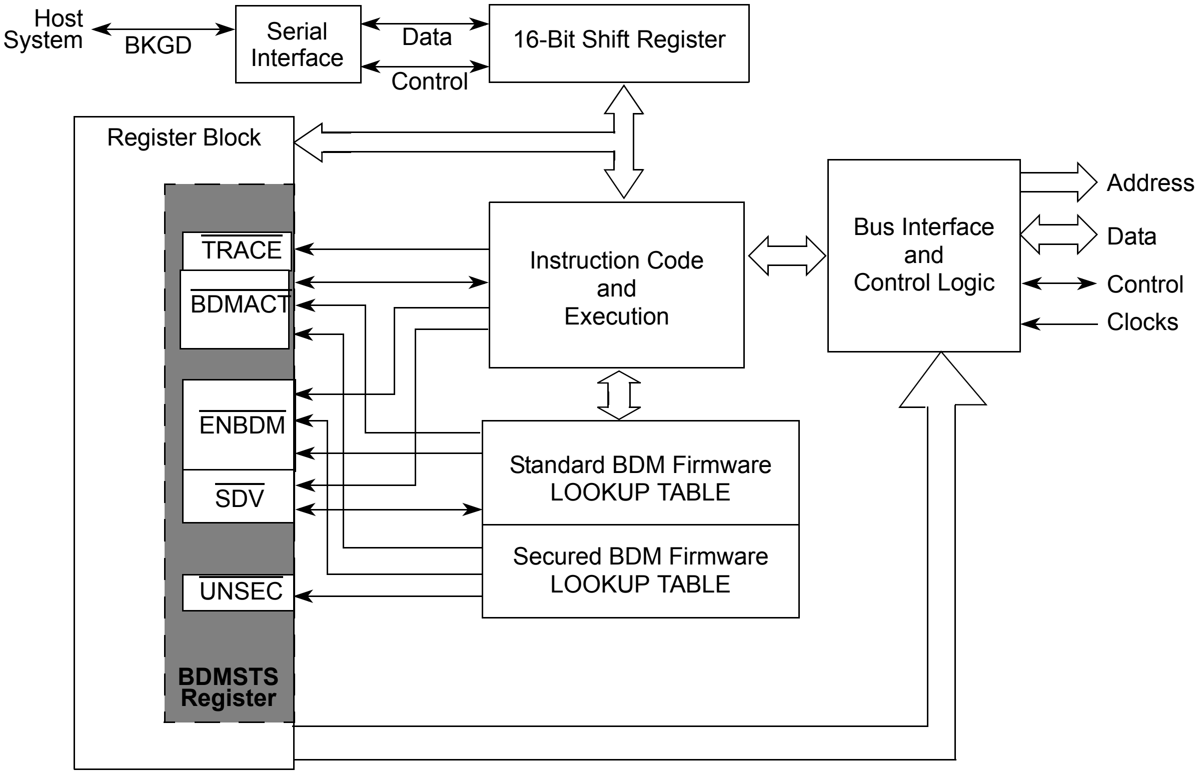


Figure 81. BDM block diagram

4.22.2 External signal description

A single-wire interface pin called the background debug interface (BKGD) pin is used to communicate with the BDM system. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the background debug mode. The communication rate of this pin is based on the settings for the VCO clock (CPMUSYNR). The BDM clock frequency is always VCO clock frequency divided by 8. After reset, the BDM clock is based on the reset values of the CPMUSYNR register (4.0 MHz). When modifying the VCO clock, make sure that the communication rate is adapted accordingly, and a communication timeout (BDM soft reset) has occurred.

4.22.3 Memory map and register definition

4.22.3.1 Module memory map

Table 48 shows the BDM memory map when BDM is active.

Table 391. BDM memory map

Global Address	Module	Size (bytes)
0x3_FF00–0x3_FF0B	BDM registers	12
0x3_FF0C–0x3_FF0E	BDM firmware ROM	3
0x3_FF0F	Family ID (part of BDM firmware ROM)	1
0x3_FF10–0x3_FFFF	BDM firmware ROM	240

4.22.3.2 Register descriptions

A summary of the registers associated with the BDM is shown in Table 392. Registers are accessed by host-driven communications to the BDM hardware using READ_BD and WRITE_BD commands.

Table 392. BDM register summary

Global address	Register name		Bit 7	6	5	4	3	2	1	Bit 0
0x3_FF00	Reserved	R	X	X	X	X	X	X	0	0
		W								
0x3_FF01	BDMSTS	R	ENBDM	BDMACT	0	SDV	TRACE	0	UNSEC	0
		W				Z	Z			
0x3_FF02	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF03	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF04	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF05	Reserved	R	X	X	X	X	X	X	X	X
		W								
0x3_FF06	BDMCCR	R	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
		W								

▨	= Unimplemented, Reserved	Z	= Implemented (do not alter)
X	= Indeterminate	0	= Always read zero

Table 392. BDM register summary (continued)

Global address	Register name		Bit 7	6	5	4	3	2	1	Bit 0
0x3_FF07	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF08	BDMPPR	R	BPAE	0	0	0	BPP3	BPP2	BPP1	BPP0
		W								
0x3_FF09	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF0A	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3_FF0B	Reserved	R	0	0	0	0	0	0	0	0
		W								

▨	= Unimplemented, Reserved	Z	= Implemented (do not alter)
X	= Indeterminate	0	= Always read zero

4.22.3.2.1 BDM status register (BDMSTS)

Table 393. BDM status register (BDMSTS)

Register Global Address 0x3_FF01	7	6	5	4	3	2	1	0
R	ENBDM	BDMACT	0	SDV	TRACE	0	UNSEC	0
W								
Reset								
Special Single-Chip Mode	0[260]	1	0	0	0	0	0[261]	0
All Other Modes	0	0	0	0	0	0	0	0

▨	= Unimplemented, Reserved	Z	= Implemented (do not alter)
0	= Always read zero		

Notes

[260] ENBDM is read as a 1 by a debugging environment in special single chip mode, when the device is either secured or not secured, but fully erased (Flash). This is because the ENBDM bit is set by the standard BDM firmware before a BDM command can be fully transmitted and executed.

[261] UNSEC is read as a 1 by a debugging environment in special single chip mode when the device is secured and fully erased, else it is 0 and can only be read if not secure (see also bit description).

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured, but subject to the following:

- ENBDM should only be set via a BDM hardware command if the BDM firmware commands are needed. (This does not apply in special single chip mode)
- BDMACT can only be set by BDM hardware upon entry into BDM. It can only be cleared by the standard BDM firmware lookup table upon exit from BDM active mode
- All other bits, while writable via BDM hardware or standard BDM firmware write commands, should only be altered by the BDM hardware or the standard firmware lookup table, as part of BDM command execution

Table 394. BDMSTS field descriptions

Field	Description
7 ENBDM	**Enable BDM** — This bit controls whether the BDM is enabled or disabled. When enabled, BDM can be made active to allow firmware commands to be executed. When disabled, BDM cannot be made active but BDM hardware commands are still allowed. 0 BDM disabled 1 BDM enabled **Note:** ENBDM is set out of reset in special single chip mode. In special single chip mode with the device secured, this bit will not be set until after the Flash erase verify tests are complete.
6 BDMACT	**BDM Active Status** — This bit becomes set upon entering BDM. The standard BDM firmware lookup table is then enabled and put into the memory map. BDMACT is cleared by a carefully timed store instruction in the standard BDM firmware, as part of the exit sequence to return to user code and remove the BDM memory from the map. 0 BDM not active 1 BDM active
4 SDV	**Shift Data Valid** — This bit is set and cleared by the BDM hardware. It is set after data has been transmitted as part of a BDM firmware or hardware read command, or after data has been received as part of a BDM firmware or hardware write command. It is cleared when the next BDM command has been received or BDM is exited. SDV is used by the standard BDM firmware to control program flow execution. 0 Data phase of command not complete 1 Data phase of command is complete
3 TRACE	**TRACE1 BDM Firmware Command is Being Executed** — This bit gets set when a BDM TRACE1 firmware command is first recognized. It will stay set until BDM firmware is exited by one of the following BDM commands: GO or GO_UNTIL[266]. 0 TRACE1 command is not being executed 1 TRACE1 command is being executed
1 UNSEC	**Unsecure** — If the device is secured this bit is only writable in special single chip mode from the BDM secure firmware. It is in a zero state as secure mode is entered so that the secure BDM firmware lookup table is enabled and put into the memory map, overlapping the standard BDM firmware lookup table. The secure BDM firmware lookup table verifies that the on-chip Flash is erased. This being the case, the UNSEC bit is set and the BDM program jumps to the start of the standard BDM firmware lookup table, and the secure BDM firmware lookup table is turned off. If the erase test fails, the UNSEC bit will not be asserted. 0 System is in a secured mode. 1 System is in a unsecured mode. **Note:** When UNSEC is set, security is off and the user can change the state of the secure bits in the on-chip Flash EEPROM. Note that if the user does not change the state of the bits to “unsecured” mode, the system will be secured again when it is next taken out of reset. After reset, this bit has no meaning or effect when the security byte in the Flash EEPROM is configured for unsecure mode.

Table 395. BDM CCR holding register (BDMCCR)

Register Global Address 0x3_FF06	7	6	5	4	3	2	1	0
R W	CCR7	CCR6	CCR5	CCR4	CCR3	CCR2	CCR1	CCR0
Reset								
Special Single-Chip Mode	1	1	0	1	1	0	0	0
All Other Modes	0	0	0	0	0	0	0	0

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

NOTE

When BDM is made active, the CPU stores the content of its CCR register in the BDMCCR register. However, out of special single-chip reset, the BDMCCR is set to 0xD8 and not 0xD0 which is the reset value of the CCR register in this CPU mode. Out of reset in all other modes the BDMCCR register is read zero.

When entering background debug mode, the BDM CCR holding register is used to save the condition code register of the user’s program. It is also used for temporary storage in the standard BDM firmware mode. The BDM CCR holding register can be written to modify the CCR value.

4.22.3.2.2 BDM program page index register (BDMPPR)

Table 396. BDM program page register (BDMPPR)

Register Global Address 0x3_FF08	7	6	5	4	3	2	1	0
R	BPAE	0	0	0	BPP3	BPP2	BPP1	BPP0
W		▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented, Reserved

Read: All modes through BDM operation when not secured

Write: All modes through BDM operation when not secured

Table 397. BDMPPR field descriptions

Field	Description
7 BPAE	**BDM Program Page Access Enable Bit** — BPAE enables program page access for BDM hardware and firmware read/write instructions The BDM hardware commands used to access the BDM registers (READ_BD and WRITE_BD) can not be used for global accesses even if the BGAE bit is set. 0 BDM Program Paging disabled 1 BDM Program Paging enabled
3–0 BPP[3:0]	**BDM Program Page Index Bits 3–0** — These bits define the selected program page. For more detailed information regarding the program page window scheme, refer to the S12S_MMC Block Guide.

4.22.3.3 Family ID assignment

The family ID is an 8-bit value located in the BDM ROM in active BDM (at global address: 0x3_FF0F). The read-only value is a unique family ID which is 0xC2 for devices with an HCS12S core.

4.22.4 Functional description

The BDM receives and executes commands from a host via a single wire serial interface. There are two types of BDM commands: hardware and firmware commands.

Hardware commands are used to read and write target system memory locations and to enter active background debug mode. See Section 4.22.4.3, “BDM hardware commands". Target system memory includes all memory that is accessible by the CPU.

Firmware commands are used to read and write CPU resources and to exit from active background debug mode. See Section 4.22.4.4, “Standard BDM firmware commands". The CPU resources referred to are the accumulator (D), X index register (X), Y index register (Y), stack pointer (SP), and program counter (PC).

Hardware commands can be executed at any time and in any mode, excluding a few exceptions as highlighted (see Section 4.22.4.3, “BDM hardware commands") and in secure mode (see Section 4.22.4.1, “Security"). BDM firmware commands can only be executed when the system is not secure and is in active background debug mode (BDM).

4.22.4.1 Security

If the user resets into special single chip mode with the system secured, a secured mode BDM firmware lookup table is brought into the map overlapping a portion of the standard BDM firmware lookup table. The secure BDM firmware verifies that the on-chip Flash EEPROM is erased. This being the case, the UNSEC and ENBDM bits will get set. The BDM program jumps to the start of the standard BDM firmware, the secured mode BDM firmware is turned off, and all BDM commands are allowed. If the Flash does not verify as erased, the BDM firmware sets the ENBDM bit, without asserting UNSEC, and the firmware enters a loop. This causes the BDM hardware commands to become enabled, but does not enable the firmware commands. This allows the BDM hardware to be used to erase the Flash.

BDM operation is not possible in any other mode than special single chip mode when the device is secured. The device can only be unsecured via the BDM serial interface in special single chip mode. For more information regarding security, see the S12S_9SEC Block Guide.

4.22.4.2 Enabling and activating BDM

The system must be in active BDM to execute standard BDM firmware commands. BDM can be activated only after being enabled. BDM is enabled by setting the ENBDM bit in the BDM status (BDMSTS) register. The ENBDM bit is set by writing to the BDM status (BDMSTS) register, via the single-wire interface, using a hardware command such as WRITE_BD_BYTE.

After being enabled, BDM is activated by one of the following [262]:

- Hardware BACKGROUND command
- CPU BGND instruction
- Breakpoint force or tag mechanism[263]

Notes

[262] BDM is enabled and active immediately out of special single-chip reset.

[263] This method is provided by the S12S_DBG module.

When BDM is activated, the CPU finishes executing the current instruction and then begins executing the firmware in the standard BDM firmware lookup table. When BDM is activated by a breakpoint, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

NOTE

> If an attempt is made to activate BDM before being enabled, the CPU resumes normal instruction execution after a brief delay. If BDM is not enabled, any hardware BACKGROUND commands issued are ignored by the BDM and the CPU is not delayed.

In active BDM, the BDM registers and standard BDM firmware lookup table are mapped to addresses 0x3_FF00 to 0x3_FFFF. BDM registers are mapped to addresses 0x3_FF00 to 0x3_FF0B. The BDM uses these registers which are readable anytime by the BDM. However, these registers are not readable by user programs.

When BDM is activated, while CPU executes code overlapping with the BDM firmware space, the saved program counter (PC) will be auto incremented by one from the BDM firmware, regardless of what caused the entry into BDM active mode (BGND instruction, BACKGROUND command or breakpoints). In such cases, the PC must be set to the next valid address via a WRITE_PC command, before executing the GO command.

4.22.4.3 BDM hardware commands

Hardware commands are used to read and write target system memory locations and to enter active background debug mode. Target system memory includes all memory that is accessible by the CPU such as on-chip RAM, Flash, I/O and control registers.

Hardware commands are executed with minimal or no CPU intervention, and do not require the system to be in active BDM for execution, although, they can still be executed in this mode. When executing a hardware command, the BDM sub-block waits for a free bus cycle, so the background access does not disturb the running application program. If a free cycle is not found within 128 clock cycles, the CPU is momentarily frozen so the BDM can steal a cycle. When the BDM finds a free cycle, the operation does not intrude on normal CPU operation, provided it can be completed in a single cycle. However, if an operation requires multiple cycles, the CPU is frozen until the operation is complete, even though the BDM found a free cycle.

The BDM hardware commands are listed in Table 398. The READ_BD and WRITE_BD commands allow access to the BDM register locations. These locations are not normally in the system memory map, but share addresses with the application in memory. To distinguish between physical memory locations that share the same address, BDM memory resources are enabled just for the READ_BD and WRITE_BD access cycle. This allows the BDM to access BDM locations unobtrusively, even if the addresses conflict with the application memory map.

Table 398. Hardware commands

Command	Opcode (hex)	Data	Description
BACKGROUND	90	None	Enter background mode if BDM is enabled. If enabled, an ACK will be issued when the part enters active background mode.
ACK_ENABLE	D5	None	Enable Handshake. Issues an ACK pulse after the command is executed.
ACK_DISABLE	D6	None	Disable Handshake. This command does not issue an ACK pulse.
READ_BD_BYTE	E4	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte.
READ_BD_WORD	EC	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table in map. Must be aligned access.

Table 398. Hardware commands (continued)

Command	Opcode (hex)	Data	Description
READ_BYTE	E0	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte.
READ_WORD	E8	16-bit address 16-bit data out	Read from memory with standard BDM firmware lookup table out of map. Must be aligned access.
WRITE_BD_BYTE	C4	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Odd address data on low byte; even address data on high byte.
WRITE_BD_WORD	CC	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table in map. Must be aligned access.
WRITE_BYTE	C0	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table out of map. Odd address data on low byte; even address data on high byte.
WRITE_WORD	C8	16-bit address 16-bit data in	Write to memory with standard BDM firmware lookup table out of map. Must be aligned access.

If enabled, ACK will occur when data is ready for transmission for all BDM READ commands, and will occur after the write is complete for all BDM WRITE commands.

4.22.4.4 Standard BDM firmware commands

BDM firmware commands are used to access and manipulate CPU resources. The system must be in active BDM to execute standard BDM firmware commands. See Section 4.22.4.2, “Enabling and activating BDM". Normal instruction execution is suspended while the CPU executes the firmware located in the standard BDM firmware lookup table. The hardware command BACKGROUND is the usual way to activate BDM.

As the system enters active BDM, the standard BDM firmware lookup table, BDM registers become visible in the on-chip memory map at 0x3_FF00–0x3_FFFF, and the CPU begins executing the standard BDM firmware. The standard BDM firmware watches for serial commands and executes them as they are received.

The firmware commands are shown in Table 399.

Table 399. Firmware commands

Command[264]	Opcode (hex)	Data	Description
READ_NEXT[265]	62	16-bit data out	Increment X index register by 2 (X = X + 2), then read word X points to.
READ_PC	63	16-bit data out	Read program counter.
READ_D	64	16-bit data out	Read D accumulator.
READ_X	65	16-bit data out	Read X index register.
READ_Y	66	16-bit data out	Read Y index register.
READ_SP	67	16-bit data out	Read stack pointer.
WRITE_NEXT	42	16-bit data in	Increment X index register by 2 (X = X + 2), then write word to location pointed to by X.
WRITE_PC	43	16-bit data in	Write program counter.
WRITE_D	44	16-bit data in	Write D accumulator.
WRITE_X	45	16-bit data in	Write X index register.
WRITE_Y	46	16-bit data in	Write Y index register.
WRITE_SP	47	16-bit data in	Write stack pointer.
GO	08	none	Go to user program. If enabled, ACK will occur when leaving active background mode.
GO_UNTIL[266]	0C	none	Go to user program. If enabled, ACK will occur upon returning to active background mode.
TRACE1	10	none	Execute one user instruction then return to active BDM. If enabled, ACK will occur upon returning to active background mode.

Table 399. Firmware commands (continued)

Command[264]	Opcode (hex)	Data	Description
TAGGO -> GO	18	none	(Previous enable tagging and go to user program.) This command will be deprecated and should not be used anymore. Opcode will be executed as a GO command.

Notes

[264] If enabled, ACK will occur when data is ready for transmission for all BDM READ commands, and will occur after the write is complete for all BDM WRITE commands.

[265] When the firmware command READ_NEXT or WRITE_NEXT is used to access the BDM address space, the BDM resources are accessed, rather than user code. Writing BDM firmware is not possible.

[266] System stop disables the ACK function and ignored commands will have no ACK-pulse (e.g., CPU in stop or wait mode). The GO_UNTIL command will not get an Acknowledge, if the CPU executes the wait or stop instruction before the “UNTIL” condition (BDM active again) is reached (see Section 4.22.4.7, “Serial interface hardware handshake protocol" last note).

4.22.4.5 BDM command structure

Hardware and firmware BDM commands start with an 8-bit opcode followed by a 16-bit address and/or a 16-bit data word, depending on the command. All the read commands return 16 bits of data despite the byte or word implication in the command name.

> 8-bit reads return 16-bits of data, only one byte of which contains valid data. If reading an even address, the valid data will appear in the MSB. If reading an odd address, the valid data will appear in the LSB.
>
> 16-bit misaligned reads and writes are generally not allowed. If attempted by BDM hardware command, the BDM ignores the least significant bit of the address and assumes an even address from the remaining bits.

For hardware data read commands, the external host must wait at least 150 bus clock cycles after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDM shift register, ready to be shifted out. For hardware write commands, the external host must wait 150 bus clock cycles after sending the data to be written, before attempting to send a new command. This is to avoid disturbing the BDM shift register before the write has been completed. The 150 bus clock cycle delay, in both cases, includes the maximum 128 cycle delay that can be incurred, as the BDM waits for a free cycle before stealing a cycle.

The external host should wait at least 48 bus clock cycles after sending the command opcode and before attempting to obtain the read data for BDM firmware read commands. The 48 cycle wait allows enough time for the requested data to be made available in the BDM shift register, ready to be shifted out.

The external host must wait 36 bus clock cycles after sending the data to be written, before attempting to send a new command for BDM firmware write commands. This is to avoid disturbing the BDM shift register before the write has been completed.

The external host should wait for at least for 76 bus clock cycles, after a TRACE1 or GO command and before starting any new serial command. This is to allow the CPU to exit gracefully from the standard BDM firmware lookup table and resume execution of the user code. Disturbing the BDM shift register prematurely may adversely affect the exit from the standard BDM firmware lookup table.

NOTE

> If the bus rate of the target processor is unknown or could be changing, it is recommended that the ACK (acknowledge function) is used to indicate when an operation is complete. When using ACK, the delay times are automated.

Figure 82 represents the BDM command structure. The command blocks illustrate a series of eight bit times, starting with a falling edge. The bar across the top of the blocks indicates that the BKGD line idles in the high state. The time for an 8-bit command is 8 × 16 target clock cycles.[267]

Notes

[267] Target clock cycles are cycles measured using the target MCU’s serial clock rate. See Section 4.22.4.6, “BDM serial interface" and Section 4.22.3.2.1, “BDM status register (BDMSTS)" for information on how serial clock rate is selected.

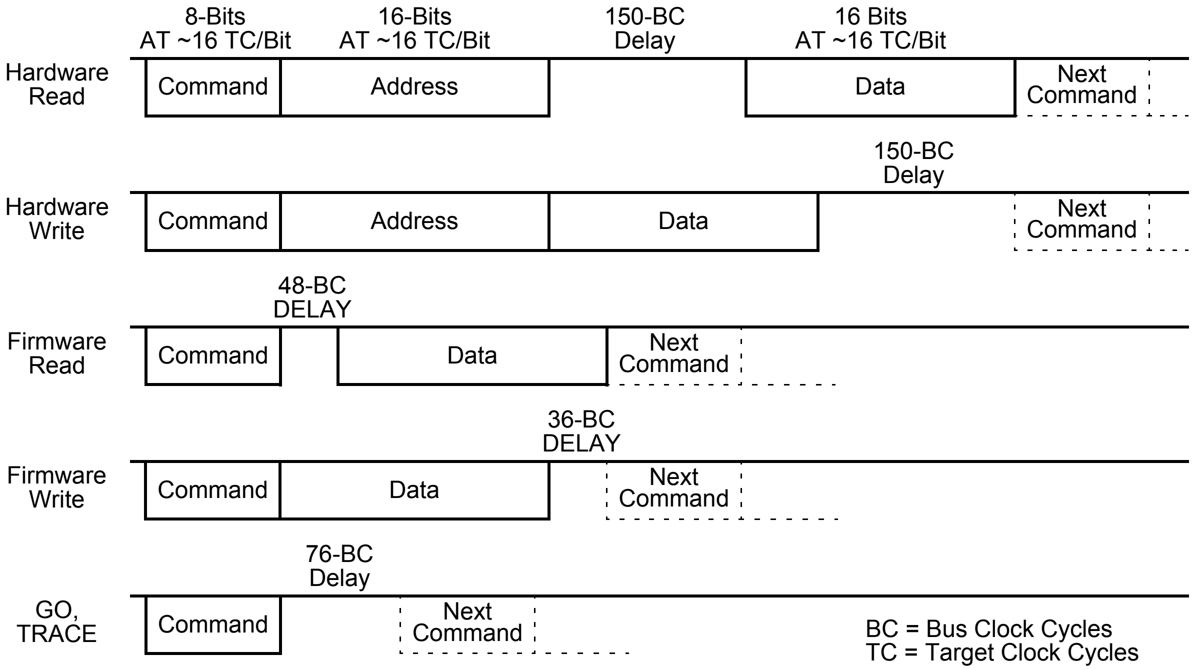


Figure 82. BDM command structure

4.22.4.6 BDM serial interface

The BDM communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDM.

The BDM serial interface is timed, based on the VCO clock (refer to the CPMU Block Guide for more details), which gets divided by 8. This clock will be referred to as the target clock in the following explanation.

The BDM serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred, most significant bit (MSB) first, at 16 target clock cycles per bit. The interface times out if 512 clock cycles occur between falling edges from the host.

The BKGD pin is a pseudo open-drain pin and has an weak on-chip active pull-up that is enabled at all times. It is assumed that there is an external pull-up and drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief driven-high (speedup) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for host-to-target is shown in Figure 83, and that of target-to-host in Figure 84 and Figure 85. All four cases begin when the host drives the BKGD pin low to generate a falling edge. Since the host and target are operating from separate clocks, it can take the target system up to one full clock cycle to recognize this edge. The target measures delays from this perceived start of the bit time, while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 83 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires the pin be driven high no later that eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

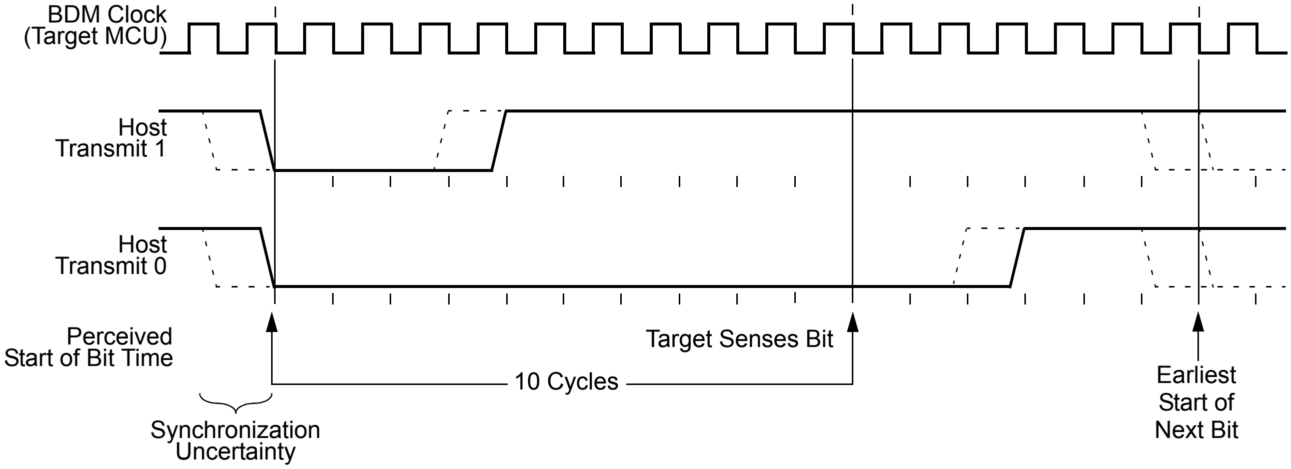


Figure 83. BDM host-to-target serial bit timing

The receive cases are more complicated. Figure 84 shows the host receiving a logic 1 from the target system. Since the host is asynchronous to the target, there is up to one clock cycle delay from the host-generated falling edge on BKGD to the perceived start of the bit time in the target. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low drive before the target drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

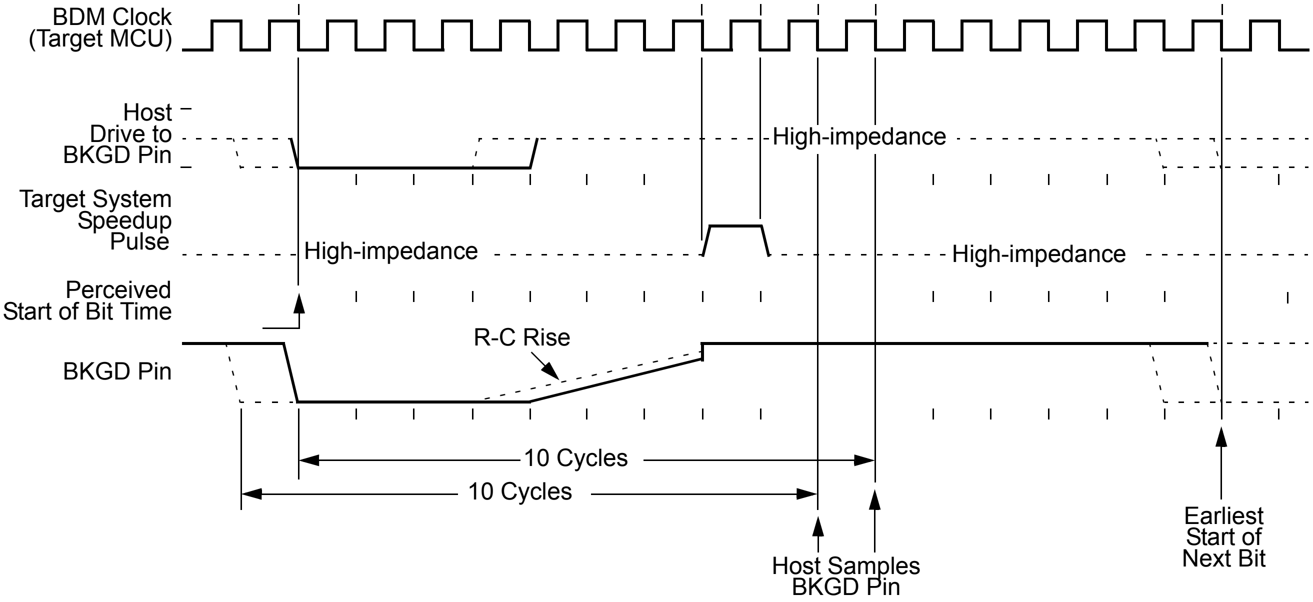


Figure 84. BDM target-to-host serial bit timing (logic 1)

Figure 85 shows the host receiving a logic 0 from the target. Since the host is asynchronous to the target, there is up to a one clock cycle delay from the host-generated falling edge on BKGD to the start of the bit time as perceived by the target. The host initiates the bit time but the target finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

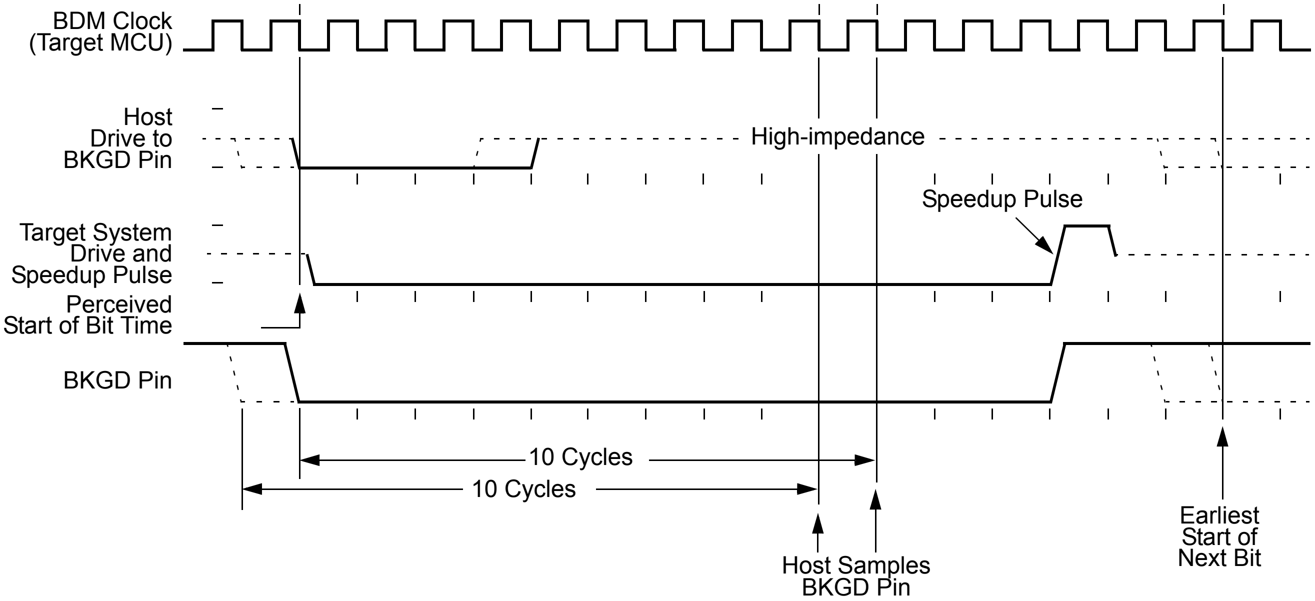


Figure 85. BDM target-to-host serial bit timing (logic 0)

4.22.4.7 Serial interface hardware handshake protocol

BDM commands that require CPU execution are ultimately treated at the MCU bus rate. Since the BDM clock source can be modified when changing the settings for the VCO frequency (CPMUSYNR), it is very helpful to provide a handshake protocol in which the host could determine when an issued command is executed by the CPU. The BDM clock frequency is always VCO frequency divided by 8. The alternative is to always wait the amount of time equal to the appropriate number of cycles at the slowest possible rate the clock could be running. This sub-section will describe the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when an issued command was successfully executed by the target. This protocol is implemented by a 16 serial clock cycle low pulse followed by a brief speedup pulse in the BKGD pin. This pulse is generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 86). This pulse is referred to as the ACK pulse. After the ACK pulse has finished: the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command (BACKGROUND, GO, GO_UNTIL[266] or TRACE1). The ACK pulse is not issued earlier than 32 serial clock cycles after the BDM command was issued. The end of the BDM command is assumed to be the 16th tick of the last bit. This minimum delay assures enough time for the host to perceive the ACK pulse. Note also that, there is no upper limit for the delay between the command and the related ACK pulse, since the command execution depends upon the CPU bus, which in some cases could be very slow due to long accesses taking place.This protocol allows a great flexibility for the POD designers, since it does not rely on any accurate time measurement or short response time to any event in the serial communication.

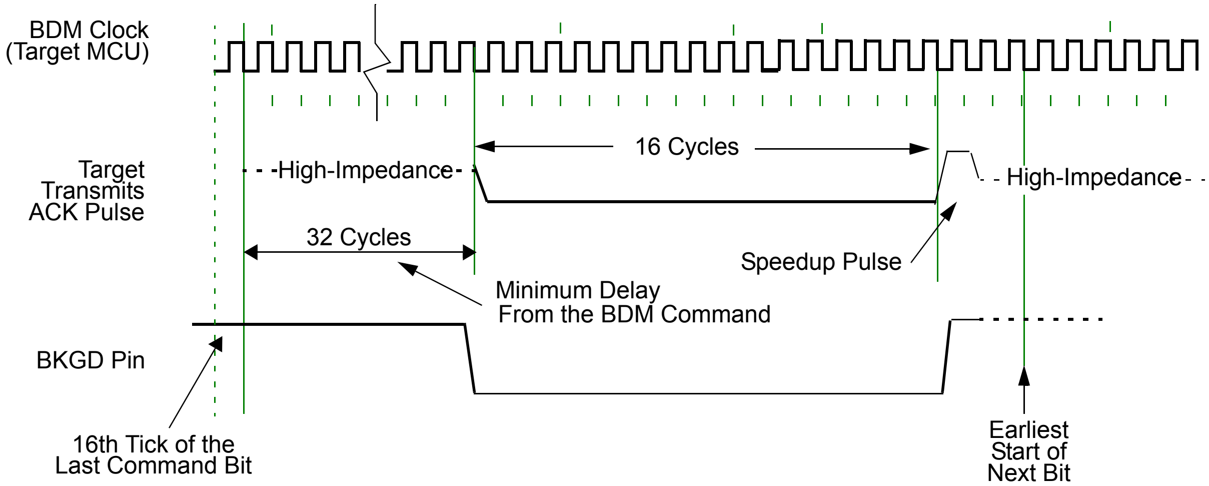


Figure 86. Target acknowledge pulse (ACK)

NOTE

If the ACK pulse was issued by the target, the host assumes the previous command was executed. If the CPU enters wait or stop prior to executing a hardware command, the ACK pulse will not be issued meaning that the BDM command was not executed. After entering wait or stop mode, the BDM command is no longer pending.

Figure 87 shows the ACK handshake protocol in a command level timing diagram. The READ_BYTE instruction is used as an example. First, the 8-bit instruction opcode is sent by the host, followed by the address of the memory location to be read. The target BDM decodes the instruction. A bus cycle is grabbed (free or stolen) by the BDM and it executes the READ_BYTE operation. Having retrieved the data, the BDM issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the byte retrieval process. Note that data is sent in the form of a word, and the host needs to determine which is the appropriate byte, based on whether the address was odd or even.

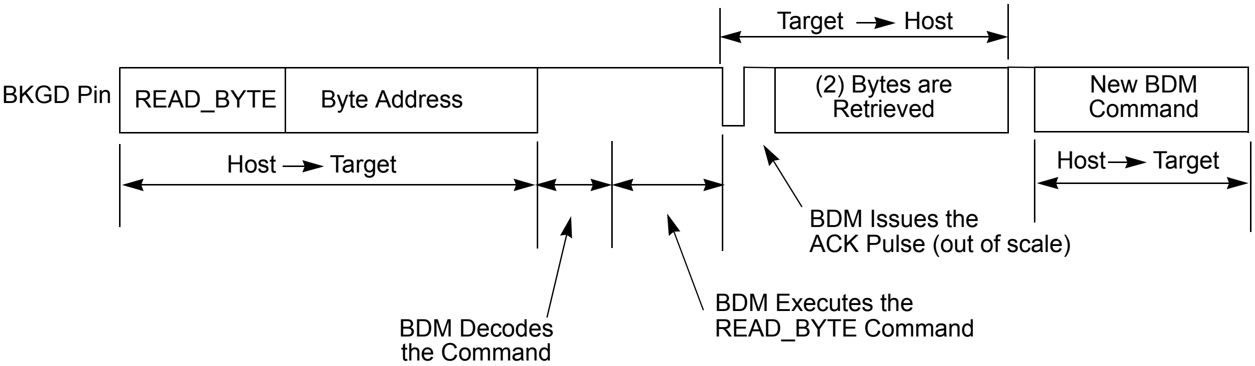


Figure 87. Handshake protocol at command level

Differently from the normal bit transfer (where the host initiates the transmission), the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge on the BKGD pin. The hardware handshake protocol in Figure 86 specifies the timing when the BKGD pin is being driven, so the host should follow this timing constraint to avoid the risk of an electrical conflict on the BKGD pin.

NOTE

The only place the BKGD pin can have an electrical conflict is when one side is driving low and the other side is issuing a speedup pulse (high). Other “highs” are pulled rather than driven. The time of the speedup pulse can become lengthy at low rates, and so the potential conflict time becomes longer as well.

The ACK handshake protocol does not support nested ACK pulses. If a BDM command is not acknowledge by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDM command. When the CPU enters wait or stop while the host issues a hardware command (e.g., WRITE_BYTE), the target discards the incoming command due to the wait or stop being detected. Therefore, the command is not acknowledged by the target, which means that the ACK pulse will not be issued. After a certain time the host (not aware of stop or wait) should decide to abort any possible pending ACK pulse, to be sure a new command can be issued. The protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

NOTE

The ACK pulse does not provide a timeout. This means for the GO_UNTIL[266] command, it cannot be distinguished if a stop or wait has been executed (command discarded and ACK not issued), or if the “UNTIL” condition (BDM active) is just not reached yet. Therefore, where the ACK pulse of a command is not issued, the possible pending command should be aborted before issuing a new command. See the handshake abort procedure described in Section 4.22.4.8, “Hardware handshake abort procedure".

4.22.4.8 Hardware handshake abort procedure

The abort procedure is based on the SYNC command. To abort a command which had not issued the corresponding ACK pulse, the host controller should generate a low pulse on the BKGD pin by driving it low for at least 128 serial clock cycles, and then driving it high for one serial clock cycle, providing a speedup pulse. By detecting this long low pulse on the BKGD pin, the target executes the SYNC protocol, see Section 4.22.4.9, "SYNC — request timed reference pulse", and assumes that the pending command, and therefore the related ACK pulse, are being aborted. Therefore, after the SYNC protocol has been completed, the host is free to issue new BDM commands. For BDM firmware READ or WRITE commands, it can not be guaranteed that the pending command is aborted, when issuing a SYNC before the corresponding ACK pulse. There is a short latency time from the time the READ or WRITE access begins until it is finished and the corresponding ACK pulse is issued. The latency time depends on the firmware READ or WRITE command that is issued and on the selected bus clock rate. When the SYNC command starts during this latency time, the READ or WRITE command will not be aborted, but the corresponding ACK pulse will be aborted. A pending GO, TRACE1 or GO_UNTIL[266] command can not be aborted. Only the corresponding ACK pulse can be aborted by the SYNC command.

Although it is not recommended, the host could abort a pending BDM command by issuing a low pulse on the BKGD pin, shorter than 128 serial clock cycles, which will not be interpreted as the SYNC command. The ACK is actually aborted when a negative edge is perceived by the target in the BKGD pin. The short abort pulse should have at least 4 clock cycles keeping the BKGD pin low, to allow the negative edge to be detected by the target. In this case, the target will not execute the SYNC protocol, but the pending command will be aborted along with the ACK pulse. The potential problem with this abort procedure is when there is a conflict between the ACK pulse and the short abort pulse, where the target may not perceive the abort pulse. The worst case is when the pending command is a read command (i.e., READ_BYTE). If the abort pulse is not perceived by the target, the host will attempt to send a new command after the abort pulse was issued, while the target expects the host to retrieve the accessed memory byte. In this case, host and target will run out of synchronism. However, if the command to be aborted is not a read command, the short abort pulse could be used. After a command is aborted, the target assumes the next negative edge, after the abort pulse, is the first bit of a new BDM command.

NOTE

The details about the short abort pulse are being provided only as a reference for the reader to better understand the BDM internal behavior. It is not recommended that this procedure be used in a real application.

Since the host knows the target serial clock frequency, the SYNC command (used to abort a command) does not need to consider the lower possible target frequency. The host could issue a SYNC very close to the 128 serial clock cycles length, providing a small overhead on the pulse length, to assure the SYNC pulse will not be misinterpreted by the target. See Section 4.22.4.9, "SYNC — request timed reference pulse".

Figure 88 shows a SYNC command being issued after a READ_BYTE, which aborts the READ_BYTE command. Note that, after the command is aborted, a new command could be issued by the host computer.

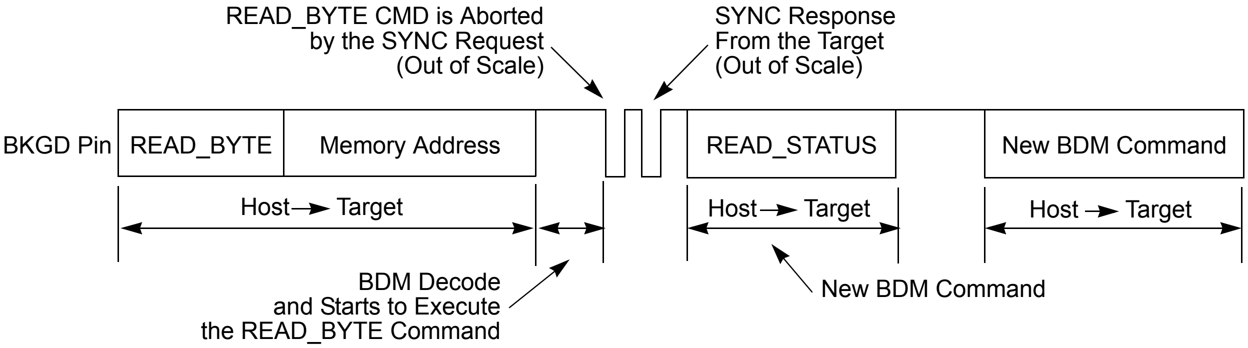


Figure 88. ACK abort procedure at the command level

NOTE

Figure 88 does not represent the signals in a true timing scale.

Figure 89 shows a conflict between the ACK pulse and the SYNC request pulse. This conflict could occur if a POD device is connected to the target BKGD pin and the target is already in debug active mode. Consider that the target CPU is executing a pending BDM command at the exact moment the POD is being connected to the BKGD pin. In this case, an ACK pulse is issued along with the SYNC command. In this case, there is an electrical conflict between the ACK speedup pulse and the SYNC pulse. Since this is not a probable situation, the protocol does not prevent this conflict from happening.

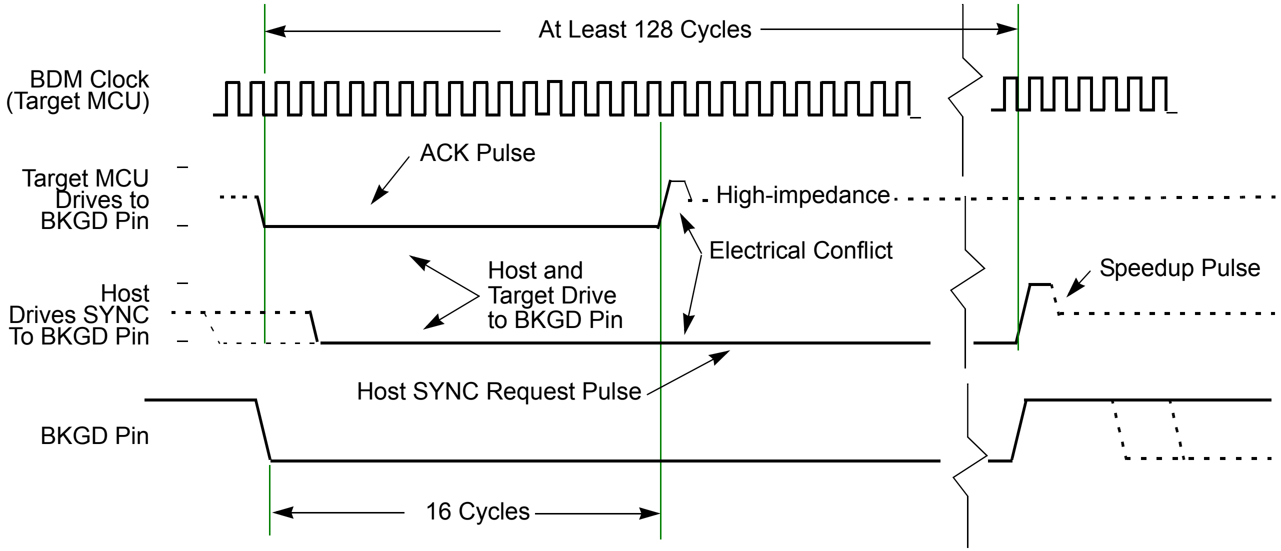


Figure 89. ACK pulse and sync request conflict

NOTE

> This information is being provided so that the MCU integrator will be aware that such a conflict could occur.

The hardware handshake protocol is enabled by the ACK_ENABLE and disabled by the ACK_DISABLE BDM commands. This provides backwards compatibility with the existing POD devices, which are not able to execute the hardware handshake protocol. It also allows for new POD devices supporting the hardware handshake protocol, to freely communicate with the target device. If desired, without the need for waiting for the ACK pulse.

The commands are described as follows:

- ACK_ENABLE — enables the hardware handshake protocol. The target will issue the ACK pulse when a CPU command is executed by the CPU. The ACK_ENABLE command itself also has the ACK pulse as a response.
- ACK_DISABLE — disables the ACK pulse protocol The host needs to use the worst case delay time at the appropriate places in the protocol.

The default state of the BDM after reset is hardware handshake protocol disabled.

All the read commands will ACK (if enabled) when the data bus cycle has completed and the data is then ready for reading out by the BKGD serial pin. All the write commands will ACK (if enabled) after the data has been received by the BDM through the BKGD serial pin, and when the data bus cycle is complete. See Section 4.22.4.3, “BDM hardware commands" and Section 4.22.4.4, “Standard BDM firmware commands" for more information on the BDM commands.

The ACK_ENABLE sends an ACK pulse when the command has been completed. This feature could be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol. If the target does not support the hardware handshake protocol the ACK pulse is not issued. In this case, the ACK_ENABLE command is ignored by the target since it is not recognized as a valid command.

The BACKGROUND command issues an ACK pulse when the CPU changes from normal to background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO command issues an ACK pulse when the CPU exits from background mode. The ACK pulse related to this command could be aborted using the SYNC command.

The GO_UNTIL(266) command is equivalent to a GO command with exception that the ACK pulse, in this case, is issued when the CPU enters into background mode. This command is an alternative to the GO command and should be used when the host wants to trace if a breakpoint match occurs and causes the CPU to enter active background mode. Note that the ACK is issued whenever the CPU enters BDM, which could be caused by a breakpoint match or by a BGND instruction being executed. The ACK pulse related to this command could be aborted using the SYNC command.

The TRACE1 command has the related ACK pulse issued when the CPU enters background active mode after one instruction of the application program is executed. The ACK pulse related to this command could be aborted using the SYNC command.

4.22.4.9 SYNC — request timed reference pulse

The SYNC command is unlike other BDM commands, because the host does not necessarily know the correct communication speed to use for BDM communications until after it has analyzed the response to the SYNC command. To issue a SYNC command, the host should perform the following steps:

1. Drive the BKGD pin low for at least 128 cycles at the lowest possible BDM serial communication frequency (The lowest serial communication frequency is determined by the settings for the VCO clock (CPMUSYNR). The BDM clock frequency is always VCO clock frequency divided by 8.)
2. Drive BKGD high for a brief speedup pulse to get a fast rise time (this speedup pulse is typically one cycle of the host clock.)
3. Remove all drive to the BKGD pin so it reverts to high-impedance.
4. Listen to the BKGD pin for the sync response pulse.

Upon detecting the SYNC request from the host, the target performs the following steps:

1. Discards any incomplete command received or bit retrieved.
2. Waits for BKGD to return to a logic one.
3. Delays 16 cycles to allow the host to stop driving the high speedup pulse.
4. Drives BKGD low for 128 cycles at the current BDM serial communication frequency.
5. Drives a one-cycle high speedup pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high-impedance.

The host measures the low time of this 128 cycle SYNC response pulse and determines the correct speed for subsequent BDM communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed, and the communication protocol can easily tolerate speed errors of several percent.

As soon as the SYNC request is detected by the target, any partially received command or bit retrieved is discarded. This is referred to as a soft-reset, equivalent to a time-out in the serial communication. After the SYNC response, the target will consider the next negative edge (issued by the host) as the start of a new BDM command or the start of new SYNC request.

Another use of the SYNC command pulse is to abort a pending ACK pulse. The behavior is exactly the same as in a regular SYNC command. Note that one of the possible causes for a command to not be acknowledged by the target is a host-target synchronization problem. In this case, the command may not have been understood by the target, so an ACK response pulse will not be issued.

4.22.4.10 Instruction tracing

When a TRACE1 command is issued to the BDM in active BDM, the CPU exits the standard BDM firmware and executes a single instruction in the user code. Once this has occurred, the CPU is forced to return to the standard BDM firmware, the BDM is active, and ready to receive a new command. If the TRACE1 command is issued again, the next user instruction will be executed. This facilitates stepping or tracing through the user code one instruction at a time.

If an interrupt is pending when a TRACE1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. Once back in standard BDM firmware execution, the program counter points to the first instruction in the interrupt service routine.

Be aware when tracing through the user code that the execution of the user code is done step by step, but peripherals are free running. Hence possible timing relations between CPU code execution and occurrence of events of other peripherals no longer exist.

Do not trace the CPU instruction BGND used for soft breakpoints. Tracing over the BGND instruction will result in a return address pointing to BDM firmware address space.

When tracing through user code which contains stop or wait instructions the following will happen when the stop or wait instruction is traced:

> The CPU enters stop or wait mode and the TRACE1 command can not be finished before leaving the low power mode. This is the case because BDM active mode can not be entered after the CPU executed the stop instruction. However, all BDM hardware commands except the BACKGROUND command are operational after tracing a stop or wait instruction, and still being in stop or wait mode. If system stop mode is entered (all bus masters are in stop mode), no BDM command is operational.
>
> As soon as stop or wait mode is exited, the CPU enters BDM active mode and the saved PC value points to the entry of the corresponding interrupt service routine.
>
> If the handshake feature is enabled, the corresponding ACK pulse of the TRACE1 command will be discarded when tracing a stop or wait instruction. Hence, there is no ACK pulse when BDM active mode is entered as part of the TRACE1 command, after CPU exited from stop or wait mode. All valid commands sent during CPU being in stop or wait mode or after CPU exited from stop or wait mode will have an ACK pulse. The handshake feature becomes disabled only when system stop mode has been reached. After a system stop mode, the handshake feature must be enabled again by sending the ACK_ENABLE command.

4.22.4.11 Serial communication timeout

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target will keep waiting for a rising edge on BKGD, to answer the SYNC request pulse. If the rising edge is not detected, the target will keep waiting forever without any timeout limit.

Consider now the case where the host returns BKGD to a logic one before 128 cycles. This is interpreted as a valid bit transmission, and not as a SYNC request. The target will keep waiting for another falling edge, marking the start of a new bit. If, a new falling edge is not detected by the target within 512 clock cycles, since the last falling edge, a timeout occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset.

If a read command is issued, but the data is not retrieved within 512 serial clock cycles, a soft-reset will occur causing the command to be disregarded. The data is not available for retrieval after the timeout has occurred. This is expected behavior if the handshake protocol is not enabled. To allow the data to be retrieved, even with a large clock frequency mismatch (between BDM and CPU) when the hardware handshake protocol is enabled, the timeout between a read command and the data retrieval is disabled. Therefore, the host could wait for more then 512 serial clock cycles, and still be able to retrieve the data from an issued read command. However, once the handshake pulse (ACK pulse) is issued, the timeout feature is re-activated, meaning that the target will timeout after 512 clock cycles. The host needs to retrieve the data within a 512 serial clock cycles time frame after the ACK pulse had been issued. After that period, the read command is discarded and the data is no longer available for retrieval. Any negative edge in the BKGD pin after the timeout period is considered to be a new command or a SYNC request.

Note that whenever a partially issued command, or partially retrieved data has occurred, the timeout in the serial communication is active. This means that if a time frame higher than 512 serial clock cycles is observed between two consecutive negative edges and the command being issued or data being retrieved is not complete, a soft-reset will occur causing the partially received command or data retrieved to be disregarded. The next negative edge in the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDM command, or the start of a SYNC request pulse.

4.23 S12 clock, reset, and power management unit (9S12I128PIMV1)

4.23.1 Introduction

This specification describes the function of the Clock, Reset, and Power Management Unit (9S12I128PIMV1).

- The Pierce oscillator (OSCLCP) provides a robust, low noise and low power external clock source. It is designed for optimal start-up margin with typical crystal oscillators
- The voltage regulator (IVREG) operates from the range 3.13 to 5.5 V. It provides all the required chip internal voltages and voltage monitors
- The Phase Locked Loop (PLL) provides a highly accurate frequency multiplier with internal filter
- The Internal Reference Clock (IRC1M) provides a1.0 MHz clock

4.23.1.1 Features

The Pierce Oscillator (OSCLCP) contains circuitry to dynamically control current gain in the output amplitude. This ensures a signal with low harmonic distortion, low power, and good noise immunity.

- Supports crystals or resonators from 4.0 to 16 MHz
- High noise immunity due to input hysteresis and spike filtering
- Low RF emissions with peak-to-peak swing limited dynamically
- Transconductance (gm) sized for optimum start-up margin for typical crystals
- Dynamic gain control eliminates the need for external current limiting resistor
- Integrated resistor eliminates the need for external bias resistor
- Low power consumption: Operates from an internal 1.8 V (nominal) supply, amplitude control limits power

The Voltage Regulator (IVREG) has the following features:

- Input voltage range from 3.13 to 5.5 V
- Low voltage detect (LVD) with low voltage interrupt (LVI)
- Power-on reset (POR)
- Low voltage reset (LVR)

The Phase Locked Loop (PLL) has the following features:

- Highly accurate and phase locked frequency multiplier
- Configurable internal filter for best stability and lock time
- Frequency modulation for defined jitter and reduced emission
- Automatic frequency lock detector
- Interrupt request on entry or exit from locked condition
- Reference clock either external (crystal) or internal square wave (1.0 MHz IRC1M) based
- PLL stability is sufficient for LIN communication, even if using IRC1M as reference clock

The Internal Reference Clock (IRC1M) has the following features:

- Trimmable in frequency
- Factory trimmed value for 1.0 MHz in Flash memory, can be overwritten by application if required

Other features of the 9S12I128PIMV1 include

- Clock monitor to detect loss of crystal
- Bus Clock Generator
  - Clock switch to select either PLLCLK or external crystal/resonator based bus clock
  - PLLCLK divider to adjust system speed
- System Reset generation from the following possible sources:
  - Power-on reset (POR)
  - Low voltage reset (LVR)
  - Illegal address access
  - COP timeout
  - Loss of oscillation (clock monitor fail)
  - External pin $\overline{\text{RESET}}$

4.23.1.2 Modes of operation

This subsection lists and briefly describes all operating modes supported by the 9S12I128PIMV1.

4.23.1.2.1 Run mode

The voltage regulator is in Full Performance mode (FPM).

The Phase Locked Loop (PLL) is on.

The Internal Reference Clock (IRC1M) is on.

- PLL Engaged Internal (PEI)
  - This is the default mode after system reset and power-on reset.
  - The bus clock is based on the PLLCLK.
  - After reset the PLL is configured for 64 MHz VCOCLK operation. Post divider is 0x03, so PLLCLK is VCOCLK divided by 4, that is 16 MHz and bus clock is 8.0 MHz. The PLL can be re-configured for other bus frequencies.
  - The reference clock for the PLL (REFCLK) is based on internal reference clock IRC1M
- PLL Engaged External (PEE)
  - The bus clock is based on the PLLCLK.
  - This mode can be entered from default mode PEI by performing the following steps:
    - Configure the PLL for desired bus frequency.
    - Program the reference divider (REFDIV[3:0] bits) to divide down oscillator frequency if necessary.
    - Enable the external oscillator (OSCE bit)
- PLL Bypassed External (PBE)
  - The bus clock is based on the oscillator clock (OSCCLK).
  - This mode can be entered from default mode PEI by performing the following steps:
    - Enable the external oscillator (OSCE bit)
    - Wait for oscillator to start up (UPOSC=1)
    - Select the oscillator clock (OSCCLK) as bus clock (PLLSEL=0).
  - The PLLCLK is still on to filter possible spikes of the external oscillator clock.

4.23.1.2.2 Wait mode

For 9S12I128PIMV1 Wait mode is the same as Run mode.

4.23.1.2.3 Stop mode

This mode is entered by executing the CPU STOP instruction.

The voltage regulator is in Reduced Power mode (RPM).

The Phase Locked Loop (PLL) is off.

The internal reference clock (IRC1M) is off.

Core clock, bus clock and BDM clock are stopped.

Depending on the setting of the PSTP and the OSCE bit, Stop mode can be differentiated between Full Stop mode (PSTP = 0 or OSCE=0) and Pseudo Stop mode (PSTP = 1 and OSCE=1).

- **Full Stop mode (PSTP = 0 or OSCE=0)**
  The external oscillator (OSCLCP) is disabled.
  After wake-up from Full Stop mode the core clock and bus clock are running on PLLCLK (PLLSEL=1). After wake-up from Full Stop mode the COP and RTI are running on IRCCLK (COPOSCSEL=0, RTIOSCSEL=0).
- **Pseudo Stop Mode (PSTP = 1 and OSCE=1)**
  The external oscillator (OSCLCP) continues to run. If the respective enable bits are set the COP and RTI will continue to run. The clock configuration bits PLLSEL, COPOSCSEL, RTIOSCSEL are unchanged.

NOTE

When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop mode with OSCE bit already 1), the software must wait for a minimum time equivalent to the startup-time of the external oscillator t_{UPOSC} before entering Pseudo Stop mode.

4.23.1.3 9S12I128PIMV1 block diagram

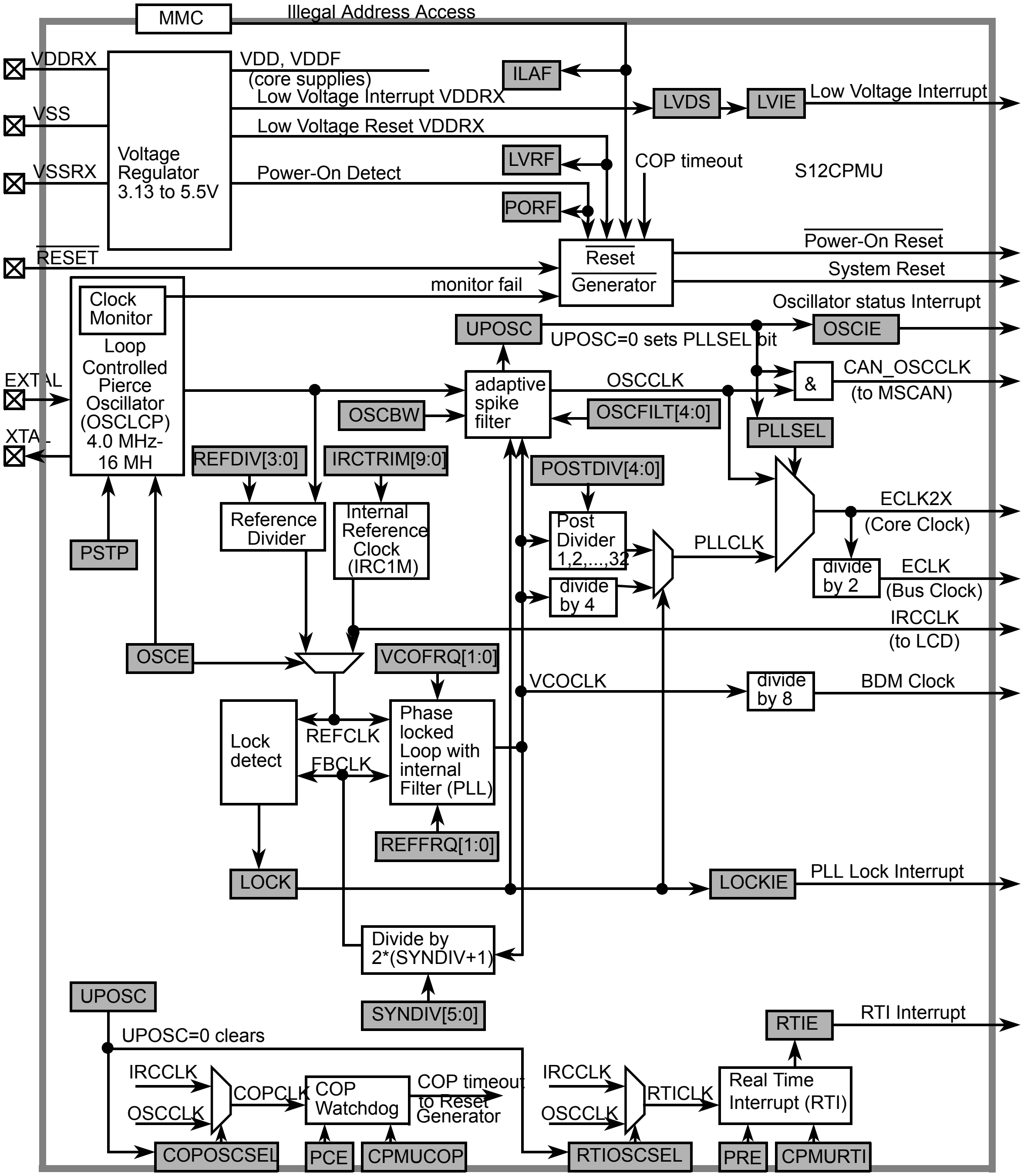


Figure 90. Block diagram of 9S12I128PIMV1

Figure 91 shows a block diagram of the OSCLCP.

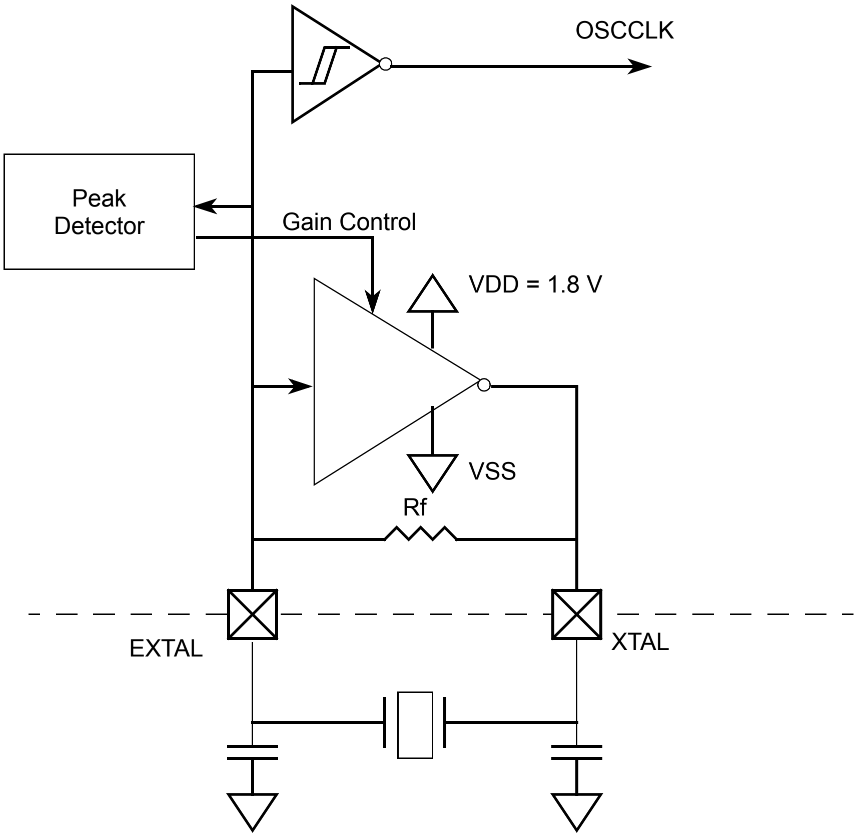


Figure 91. OSCLCP block diagram

4.23.2 Signal description

This section lists and describes the signals that connect off chip.

4.23.2.1 $\overline{\text{RESET}}$

Pin $\overline{\text{RESET}}$ is an active-low bidirectional pin. As an input, it initializes the MCU asynchronously to a known start-up state. As an open-drain output, it indicates that an MCU-internal reset has been triggered.

4.23.2.2 EXTAL and XTAL

These pins provide the interface for a crystal to control the internal clock generator circuitry. EXTAL is the external clock input or the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. The MCU internal OSCCLK is derived from the EXTAL input frequency. If OSCE=0, the EXTAL pin is pulled down by an internal resistor of approximately 200 kΩ, and the XTAL pin is pulled down by an internal resistor of approximately 700 kΩ.

NOTE

NXP recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier. Loop controlled circuit is not suited for overtone resonators and crystals.

4.23.2.3 VSS — ground pin

VSS must be grounded.

4.23.2.4 VDDRX, VSSRX— regulator power input pin and pad supply pins

VDDRX is the power input of IVREG and the PAD positive supply pin. All currents sourced into the regulator loads flow through this pin.The VDDRX/VSSX supply domain is monitored by the low voltage reset circuit. An off-chip decoupling capacitor (100 nF...220 nF, X7R ceramic) between VDDRX and VSSX can further improve the quality of this supply.

4.23.2.5 VDD — internal regulator output supply (core logic)

Node VDD is a device internal supply output of the voltage regulator that provides the power supply for the core logic. This supply domain is monitored by the low voltage reset circuit.

4.23.2.6 VDDF — internal regulator output supply (NVM logic)

Node VDDF is a device internal supply output of the voltage regulator that provides the power supply for the NVM logic. This supply domain is monitored by the low voltage reset circuit

4.23.3 Memory map and registers

This section provides a detailed description of all registers accessible in the 9S12I128PIMV1.

4.23.3.1 Module memory map

The 9S12I128PIMV1 registers are shown in Table 400.

Table 400. CPMU register summary

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0034	CPMU SYNR	R	VCOFRQ[1:0]		SYNDIV[5:0]					
		W								
0x0035	CPMU REFDIV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
		W			▨	▨				
0x0036	CPMU POSTDIV	R	0	0	0	POSTDIV[4:0]				
		W	▨	▨	▨					
0x0037	CPMUFLG	R	RTIF	PORF	LVRF	LOCKIF	LOCK	ILAF	OSCIF	UPOSC
		W					▨			▨
0x0038	CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
		W		▨	▨		▨	▨		▨
0x0039	CPMUCLKS	R	PLLSEL	PSTP	0	0	PRE	PCE	RTI OSCSEL	COP OSCSEL
		W			▨	▨				
0x003A	CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
		W	▨	▨			▨	▨	▨	▨
0x003B	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								
0x003C	CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		W			WRTMASK	▨	▨			
0x003D	RESERVEDC	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

Table 400. CPMU register summary (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x003E	RESERVEDC	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x003F	CPMU ARMCOP	R	0	0	0	0	0	0	0	0
		W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x02F0	RESERVED	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x02F1	CPMU LVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W	▨	▨	▨	▨	▨	▨		
0x02F2	RESERVED	R		0	0					
		W		▨	▨					
0x02F3	RESERVED	R							0	0
		W							▨	▨
0x02F4	RESERVED	R								
		W								
0x02F5	RESERVED	R								
		W								
0x02F6	RESERVEDC	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x02F7	RESERVED	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x02F8	CPMU IRCTRIMH	R	TCTRIM[4:0]					0	IRCTRIM[9:8]	
		W						▨		
0x02F9	CPMU IRCTRIML	R	IRCTRIM[7:0]							
		W								
0x02FA	CPMUOSC	R	OSCE	OSCBW	OSCPINS_EN	OSCFILT[4:0]				
		W			▨					
0x02FB	CPMUPROT	R	0	0	0	0	0	0	0	PROT
		W	▨	▨	▨	▨	▨	▨	▨	
0x02FC	RESERVEDC	R	0	0	0	0	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

4.23.3.2 Register descriptions

This section describes all the 9S12I128PIMV1 registers and their individual bits. Address order is as listed in Table 400.

4.23.3.2.1 9S12I128PIMV1 synthesizer register (CPMUSYNR)

The CPMUSYNR register controls the multiplication factor of the PLL and selects the VCO frequency range.

Table 401. 9S12I128PIMV1 synthesizer register (CPMUSYNR)

0x0034

	7	6	5	4	3	2	1	0
R W	VCOFRQ[1:0]		SYNDIV[5:0]					
Reset	0	1	0	1	1	1	1	1

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register), Else write has no effect.

NOTE

Writing to this register clears the LOCK and UPOSC status bits.

If PLL has locked (LOCK=1) $f_{VCO} = 2 \times f_{REF} \times (SYNDIV + 1)$

NOTE

f_{VCO} must be within the specified VCO frequency lock range. Bus frequency f_{BUS} must not exceed the specified maximum.

The VCOFRQ[1:0] bits are used to configure the VCO gain for optimal stability and lock time. For correct PLL operation, the VCOFRQ[1:0] bits have to be selected according to the actual target VCOCLK frequency, as shown in Table 402. Setting the VCOFRQ[1:0] bits incorrectly can result in a non functional PLL (no locking and/or insufficient stability).

Table 402. VCO clock frequency selection

VCOCLK Frequency Ranges	VCOFRQ[1:0]
32 MHz <= f_{VCO}<= 48 MHz	00
48 MHz < f_{VCO}<= 64 MHz	01
Reserved	10
Reserved	11

4.23.3.2.2 9S12I128PIMV1 reference divider register (CPMUREFDIV)

The CPMUREFDIV register provides a finer granularity for the PLL multiplier steps when using the external oscillator as reference.

Table 403. 9S12I128PIMV1 reference divider register (CPMUREFDIV)

0x0035

	7	6	5	4	3	2	1	0
R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
W			▨	▨				
Reset	0	0	0	0	1	1	1	1

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register), else write has no effect.

NOTE

Write to this register clears the LOCK and UPOSC status bits.

If OSCLCP is enabled (OSCE=1) $f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$

If OSCLCP is disabled (OSCE=0) $f_{REF} = f_{IRC1M}$

The REFFRQ[1:0] bits are used to configure the internal PLL filter for optimal stability and lock time. For correct PLL operation, the REFFRQ[1:0] bits have to be selected according to the actual REFCLK frequency as shown in Table 404.

If IRC1M is selected as REFCLK (OSCE=0) the PLL filter is fixed configured for the 1.0 MHz <= f_{REF} <= 2.0 MHz range. The bits can still be written but will have no effect on the PLL filter configuration.

For OSCE=1, setting the REFFRQ[1:0] bits incorrectly can result in a non functional PLL (no locking and/or insufficient stability).

Table 404. Reference clock frequency selection if OSC_LCP Is enabled

REFCLK Frequency Ranges (OSCE=1)	REFFRQ[1:0]
1.0 MHz <= f_{REF} <= 2.0 MHz	00
2.0 MHz < f_{REF} <= 6.0 MHz	01
6.0 MHz < f_{REF} <= 12.0 MHz	10
f_{REF} >12.0 MHz	11

4.23.3.2.3 9S12I128PIMV1 post divider register (CPMUPOSTDIV)

The POSTDIV register controls the frequency ratio between the VCOCLK and the PLLCLK.

Table 405. 9S12I128PIMV1 post divider register (CPMUPOSTDIV)

0x0036

	7	6	5	4	3	2	1	0
R	0	0	0	POSTDIV[4:0]				
W	▨	▨	▨					
Reset	0	0	0	0	0	0	1	1

▨ = Unimplemented or Reserved

Read: Anytime

Write: Anytime if PLLSEL=1. Else write has no effect.

If PLL is locked (LOCK=1) $$f_{PLL} = \frac{f_{VCO}}{(POSTDIV + 1)}$$

If PLL is not locked (LOCK=0) $$f_{PLL} = \frac{f_{VCO}}{4}$$

If PLL is selected (PLLSEL=1) $$f_{bus} = \frac{f_{PLL}}{2}$$

4.23.3.2.4 9S12I128PIMV1 flags register (CPMUFLG)

This register provides 9S12I128PIMV1 status bits and flags.

Table 406. 9S12I128PIMV1 flags register (CPMUFLG)

0x0037

	7	6	5	4	3	2	1	0
R	RTIF	PORF	LVRF	LOCKIF	LOCK	ILAF	OSCIF	UPOSC
W					▨			▨
Reset	0	[268]	[269]	0	0	[270]	0	0

▨ = Unimplemented or Reserved

Notes

[268] 1. PORF is set to 1 when a power on reset occurs. Unaffected by System Reset.

[269] 2. LVRF is set to 1 when a low voltage reset occurs. Unaffected by System Reset. Set by power on reset.

[270] 3. ILAF is set to 1 when an illegal address reset occurs. Unaffected by System Reset. Cleared by power on reset.

Read: Anytime

Write: Refer to each bit for individual write conditions

Table 407. CPMUFLG field descriptions

Field	Description
7 RTIF	**Real Time Interrupt Flag** — RTIF is set to 1 at the end of the RTI period. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (RTIE=1), RTIF causes an interrupt request. 0 RTI timeout has not yet occurred. 1 RTI timeout has occurred.
6 PORF	**Power on Reset Flag** — PORF is set to 1 when a power on reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Power on reset has not occurred. 1 Power on reset has occurred.
5 LVRF	**Low Voltage Reset Flag** — LVRF is set to 1 when a low voltage reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Low voltage reset has not occurred. 1 Low voltage reset has occurred.
4 LOCKIF	**PLL Lock Interrupt Flag** — LOCKIF is set to 1 when LOCK status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect.If enabled (LOCKIE=1), LOCKIF causes an interrupt request. 0 No change in LOCK bit. 1 LOCK bit has changed.
3 LOCK	**Lock Status Bit** — LOCK reflects the current state of PLL lock condition. Writes have no effect. **While PLL is unlocked (LOCK=0) f_{PLL} is f_{VCO} / 4 to protect the system from high core clock frequencies during the PLL stabilization time t_{lock}.** 0 VCOCLK is not within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/4$. 1 VCOCLK is within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/(POSTDIV+1)$.
2 ILAF	**Illegal Address Reset Flag** — ILAF is set to 1 when an illegal address reset occurs. Refer to MMC chapter for details. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Illegal address reset has not occurred. 1 Illegal address reset has occurred.
1 OSCIF	**Oscillator Interrupt Flag** — OSCIF is set to 1 when UPOSC status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect.If enabled (OSCIE=1), OSCIF causes an interrupt request. 0 No change in UPOSC bit. 1 UPOSC bit has changed.
0 UPOSC	**Oscillator Status Bit** — UPOSC reflects the status of the oscillator. Writes have no effect. While UPOSC=0 the OSCCLK going to the MSCAN module is off. Entering Full Stop Mode UPOSC is cleared. 0 The oscillator is off or oscillation is not qualified by the PLL. 1 The oscillator is qualified by the PLL.

NOTE

The adaptive oscillator filter uses the VCO clock as a reference to continuously qualify the external oscillator clock. As a result, the PLL is always active and a valid PLL configuration is required for the system to work properly. Furthermore, the adaptive oscillator filter is used to determine the status of the external oscillator (reflected in the UPOSC bit). Since this function also relies on the VCO clock, loosing PLL lock status (LOCK=0, except for entering Pseudo Stop mode) means loosing the oscillator status information as well (UPOSC=0).

4.23.3.2.5 9S12I128PIMV1 interrupt enable register (CPMUINT)

This register enables 9S12I128PIMV1 interrupt requests.

Table 408. 9S12I128PIMV1 interrupt enable register (CPMUINT)

0x0038

	7	6	5	4	3	2	1	0
R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
W		▨	▨		▨	▨		▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Anytime

Table 409. CRGINT field descriptions

Field	Description
7 RTIE	Real Time Interrupt Enable Bit 0 Interrupt requests from RTI are disabled. 1 Interrupt will be requested whenever RTIF is set.
4 LOCKIE	PLL Lock Interrupt Enable Bit 0 PLL LOCK interrupt requests are disabled. 1 Interrupt will be requested whenever LOCKIF is set.
1 OSCIE	Oscillator Corrupt Interrupt Enable Bit 0 Oscillator Corrupt interrupt requests are disabled. 1 Interrupt will be requested whenever OSCIF is set.

4.23.3.2.6 9S12I128PIMV1 clock select register (CPMUCLKS)

This register controls 9S12I128PIMV1 clock selection.

Table 410. 9S12I128PIMV1 clock select register (CPMUCLKS)

0x0039

	7	6	5	4	3	2	1	0
R	PLLSEL	PSTP	0	0	PRE	PCE	RTI OSCSEL	COP OSCSEL
W			▨	▨				
Reset	1	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write:

1. Only possible if PROT=0 (CPMUPROT register) in all MCU modes (Normal and Special mode).
2. All bits in Special mode (if PROT=0).

3. PLLSEL, PSTP, PRE, PCE, RTIOSCSEL: In Normal mode (if PROT=0).
4. COPOSCSEL: In Normal mode (if PROT=0) until CPMUCOP write once is taken. If COPOSCSEL was cleared by UPOSC=0 (entering Full Stop mode with COPOSCSEL=1 or insufficient OSCCLK quality), then COPOSCSEL can be set again once.

NOTE

After writing CPMUCLKS register, it is strongly recommended to read back CPMUCLKS register to make sure that write of PLLSEL, RTIOSCSEL and COPOSCSEL was successful.

Table 411. CPMUCLKS descriptions

Field	Description
7 PLLSEL	PLL Select Bit This bit selects the PLLCLK as source of the system clocks (core clock and bus clock). PLLSEL can only be set to 0, if UPOSC=1. UPOSC= 0 sets the PLLSEL bit. Entering Full Stop mode sets the PLLSEL bit. 0 System clocks are derived from OSCCLK if oscillator is up (UPOSC=1, $f_{BUS} = f_{OSC} / 2$. 1 System clocks are derived from PLLCLK, $f_{BUS} = f_{PLL} / 2$.
6 PSTP	Pseudo Stop Bit This bit controls the functionality of the oscillator during Stop mode. 0 Oscillator is disabled in Stop mode (Full Stop mode). 1 Oscillator continues to run in Stop mode (Pseudo Stop mode), option to run RTI and COP. **Note:** Pseudo Stop mode allows for faster STOP recovery and reduces the mechanical stress and aging of the resonator in case of frequent STOP conditions at the expense of a slightly increased power consumption. **Note:** When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop mode with OSCE bit is already 1) the software must wait for a minimum time equivalent to the startup time of the external oscillator t_{UPOSC} before entering Pseudo Stop mode.
3 PRE	**RTI Enable During Pseudo Stop Bit** — PRE enables the RTI during Pseudo Stop mode. 0 RTI stops running during Pseudo Stop mode. 1 RTI continues running during Pseudo Stop mode if RTIOSCSEL=1. **Note:** If PRE=0 or RTIOSCSEL=0 then the RTI will go static while Stop mode is active. The RTI counter will not be reset.
2 PCE	**COP Enable During Pseudo Stop Bit** — PCE enables the COP during Pseudo Stop mode. 0 COP stops running during Pseudo Stop mode 1 COP continues running during Pseudo Stop mode if COPOSCSEL=1 **Note:** If PCE=0 or COPOSCSEL=0 then the COP will go static while Stop mode is active. The COP counter will not be reset.
1 RTIOSCSEL	**RTI Clock Select**— RTIOSCSEL selects the clock source to the RTI. Either IRCCLK or OSCCLK. Changing the RTIOSCSEL bit re-starts the RTI timeout period. RTIOSCSEL can only be set to 1, if UPOSC=1. UPOSC= 0 clears the RTIOSCSEL bit. 0 RTI clock source is IRCCLK. 1 RTI clock source is OSCCLK.
0 COPOSCSEL	**COP Clock Select**— COPOSCSEL selects the clock source to the COP. Either IRCCLK or OSCCLK. Changing the COPOSCSEL bit re-starts the COP timeout period. COPOSCSEL can only be set to 1, if UPOSC=1. UPOSC= 0 clears the COPOSCSEL bit. 0 COP clock source is IRCCLK. 1 COP clock source is OSCCLK

4.23.3.2.7 9S12I128PIMV1 PLL control register (CPMUPLL)

This register controls the PLL functionality.

Table 412. 9S12I128PIMV1 PLL control register (CPMUPLL)

0x003A

	7	6	5	4	3	2	1	0
R	0	0	FM1	FM0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

NOTE

Write to this register clears the LOCK and UPOSC status bits.

NOTE

Care should be taken to ensure that the bus frequency does not exceed the specified maximum when frequency modulation is enabled.

NOTE

The frequency modulation (FM1 and FM0) can not be used if the Adaptive Oscillator Filter is enabled.

Table 413. CPMUPLL field descriptions

Field	Description
5, 4 FM1, FM0	PLL **Frequency Modulation Enable Bits** — FM1 and FM0 enable frequency modulation on the VCOCLK. This is to reduce noise emission. The modulation frequency is f_{REF} divided by 16. See Table 414 for coding.

Table 414. FM Amplitude selection

FM1	FM0	FM amplitude /f_{VCO} variation
0	0	FM off
0	1	±1%
1	0	±2%
1	1	±4%

4.23.3.2.8 9S12I128PIMV1 RTI control register (CPMURTI)

This register selects the timeout period for the Real Time Interrupt. The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop mode with PSTP=1 (Pseudo Stop mode) and RTIOSCSEL=1 the RTI continues to run, else the RTI counter halts in Stop mode.

Table 415. 9S12I128PIMV1 RTI control register (CPMURTI)

0x003B

	7	6	5	4	3	2	1	0
R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime

Write: Anytime

NOTE

A write to this register starts the RTI timeout period. A change of the RTIOSCSEL bit (writing a different value or loosing UPOSC status) re-starts the RTI timeout period.

Table 416. CPMURTI field descriptions

Field	Description
7 RTDEC	**Decimal or Binary Divider Select Bit** — RTDEC selects decimal or binary based prescaler values. 0 Binary based divider value. See Table 417 1 Decimal based divider value. See Table 418
6–4 RTR[6:4]	**Real Time Interrupt Prescale Rate Select Bits** — These bits select the prescale rate for the RTI. See Table 417 and Table 418.
3–0 RTR[3:0]	**Real Time Interrupt Modulus Counter Select Bits** — These bits select the modulus counter target value to provide additional granularity.Table 417 and Table 418 show all possible divide values selectable by the CPMURTI register**.**

Table 417. RTI frequency divide rates for RTDEC = 0

RTR[3:0]	RTR[6:4] =							
	000 **(OFF)**	**001** **(2^{10})**	**010** **(2^{11})**	**011** **(2^{12})**	**100** **(2^{13})**	**101** **(2^{14})**	**110** **(2^{15})**	**111** **(2^{16})**
0000 (÷1)	OFF[271]	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
0001 (÷2)	OFF	$2x2^{10}$	$2x2^{11}$	$2x2^{12}$	$2x2^{13}$	$2x2^{14}$	$2x2^{15}$	$2x2^{16}$
0010 (÷3)	OFF	$3x2^{10}$	$3x2^{11}$	$3x2^{12}$	$3x2^{13}$	$3x2^{14}$	$3x2^{15}$	$3x2^{16}$
0011 (÷4)	OFF	$4x2^{10}$	$4x2^{11}$	$4x2^{12}$	$4x2^{13}$	$4x2^{14}$	$4x2^{15}$	$4x2^{16}$
0100 (÷5)	OFF	$5x2^{10}$	$5x2^{11}$	$5x2^{12}$	$5x2^{13}$	$5x2^{14}$	$5x2^{15}$	$5x2^{16}$
0101 (÷6)	OFF	$6x2^{10}$	$6x2^{11}$	$6x2^{12}$	$6x2^{13}$	$6x2^{14}$	$6x2^{15}$	$6x2^{16}$
0110 (÷7)	OFF	$7x2^{10}$	$7x2^{11}$	$7x2^{12}$	$7x2^{13}$	$7x2^{14}$	$7x2^{15}$	$7x2^{16}$
0111 (÷8)	OFF	$8x2^{10}$	$8x2^{11}$	$8x2^{12}$	$8x2^{13}$	$8x2^{14}$	$8x2^{15}$	$8x2^{16}$
1000 (÷9)	OFF	$9x2^{10}$	$9x2^{11}$	$9x2^{12}$	$9x2^{13}$	$9x2^{14}$	$9x2^{15}$	$9x2^{16}$
1001 (÷10)	OFF	$10x2^{10}$	$10x2^{11}$	$10x2^{12}$	$10x2^{13}$	$10x2^{14}$	$10x2^{15}$	$10x2^{16}$
1010 (÷11)	OFF	$11x2^{10}$	$11x2^{11}$	$11x2^{12}$	$11x2^{13}$	$11x2^{14}$	$11x2^{15}$	$11x2^{16}$
1011 (÷12)	OFF	$12x2^{10}$	$12x2^{11}$	$12x2^{12}$	$12x2^{13}$	$12x2^{14}$	$12x2^{15}$	$12x2^{16}$
1100 (÷13)	OFF	$13x2^{10}$	$13x2^{11}$	$13x2^{12}$	$13x2^{13}$	$13x2^{14}$	$13x2^{15}$	$13x2^{16}$
1101 (÷14)	OFF	$14x2^{10}$	$14x2^{11}$	$14x2^{12}$	$14x2^{13}$	$14x2^{14}$	$14x2^{15}$	$14x2^{16}$
1110 (÷15)	OFF	$15x2^{10}$	$15x2^{11}$	$15x2^{12}$	$15x2^{13}$	$15x2^{14}$	$15x2^{15}$	$15x2^{16}$
1111 (÷16)	OFF	$16x2^{10}$	$16x2^{11}$	$16x2^{12}$	$16x2^{13}$	$16x2^{14}$	$16x2^{15}$	$16x2^{16}$

Notes

[271] Denotes the default value out of reset.This value should be used to disable the RTI to ensure future backwards compatibility.

Table 418. RTI frequency divide rates for RTDEC=1

RTR[3:0]	RTR[6:4] =							
	000 **(1x10^3)**	**001** **(2x10^3)**	**010** **(5x10^3)**	**011** **(10x10^3)**	**100** **(20x10^3)**	**101** **(50x10^3)**	**110** **(100x10^3)**	**111** **(200x10^3)**
0000 (÷1)	1x10^3	2x10^3	5x10^3	10x10^3	20x10^3	50x10^3	100x10^3	200x10^3
0001 (÷2)	2x10^3	4x10^3	10x10^3	20x10^3	40x10^3	100x10^3	200x10^3	400x10^3
0010 (÷3)	3x10^3	6x10^3	15x10^3	30x10^3	60x10^3	150x10^3	300x10^3	600x10^3
0011 (÷4)	4x10^3	8x10^3	20x10^3	40x10^3	80x10^3	200x10^3	400x10^3	800x10^3
0100 (÷5)	5x10^3	10x10^3	25x10^3	50x10^3	100x10^3	250x10^3	500x10^3	1x10^6
0101 (÷6)	6x10^3	12x10^3	30x10^3	60x10^3	120x10^3	300x10^3	600x10^3	1.2x10^6
0110 (÷7)	7x10^3	14x10^3	35x10^3	70x10^3	140x10^3	350x10^3	700x10^3	1.4x10^6
0111 (÷8)	8x10^3	16x10^3	40x10^3	80x10^3	160x10^3	400x10^3	800x10^3	1.6x10^6
1000 (÷9)	9x10^3	18x10^3	45x10^3	90x10^3	180x10^3	450x10^3	900x10^3	1.8x10^6
1001 (÷10)	10 x10^3	20x10^3	50x10^3	100x10^3	200x10^3	500x10^3	1x10^6	2x10^6
1010 (÷11)	11 x10^3	22x10^3	55x10^3	110x10^3	220x10^3	550x10^3	1.1x10^6	2.2x10^6
1011 (÷12)	12x10^3	24x10^3	60x10^3	120x10^3	240x10^3	600x10^3	1.2x10^6	2.4x10^6
1100 (÷13)	13x10^3	26x10^3	65x10^3	130x10^3	260x10^3	650x10^3	1.3x10^6	2.6x10^6
1101 (÷14)	14x10^3	28x10^3	70x10^3	140x10^3	280x10^3	700x10^3	1.4x10^6	2.8x10^6
1110 (÷15)	15x10^3	30x10^3	75x10^3	150x10^3	300x10^3	750x10^3	1.5x10^6	3x10^6
1111 (÷16)	16x10^3	32x10^3	80x10^3	160x10^3	320x10^3	800x10^3	1.6x10^6	3.2x10^6

4.23.3.2.9 9S12I128PIMV1 COP control register (CPMUCOP)

This register controls the COP (Computer Operating Properly) watchdog. The clock source for the COP is either IRCCLK or OSCCLK depending on the setting of the COPOSCSEL bit. In Stop mode with PSTP=1(Pseudo Stop mode), COPOSCSEL=1 and PCE=1 the COP continues to run, else the COP counter halts in Stop mode.

Table 419. 9S12I128PIMV1 COP control register (CPMUCOP)

0x003C

	7	6	5	4	3	2	1	0
R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
W			WRTMASK	▨	▨			
Reset	F	0	0	0	0	F	F	F

After de-assert of System Reset the values are automatically loaded from the Flash memory. See Device specification for details.

▨ = Unimplemented or Reserved

Read: Anytime

Write:

1. RSBCK: anytime in Special Mode; write to “1” but not to “0” in Normal mode
2. WCOP, CR2, CR1, CR0:
   - — Anytime in Special mode, when WRTMASK is 0, otherwise it has no effect
   - — Write once in Normal mode, when WRTMASK is 0, otherwise it has no effect.
     - – Writing CR[2:0] to “000” has no effect, but counts for the “write once” condition.
     - – Writing WCOP to “0” has no effect, but counts for the “write once” condition.

When a non-zero value is loaded from Flash to CR[2:0] the COP timeout period is started.

A change of the COPOSCSEL bit (writing a different value or loosing UPOSC status) re-starts the COP timeout period.

In Normal mode the COP timeout period is restarted if either of these conditions is true:

1. Writing a non-zero value to CR[2:0] (anytime in Special mode, once in Normal mode) with WRTMASK = 0.
2. Writing WCOP bit (anytime in Special mode, once in Normal mode) with WRTMASK = 0.
3. Changing RSBCK bit from “0” to “1”.

In Special mode, any write access to CPMUCOP register restarts the COP timeout period.

Table 420. CPMUCOP field descriptions

Field	Description
7 WCOP	**Window COP Mode Bit** — When set, a write to the CPMUARMCOP register must occur in the last 25% of the selected period. A write during the first 75% of the selected period generates a COP reset. As long as all writes occur during this window, $55 can be written as often as desired. Once $AA is written after the $55, the timeout logic restarts and the user must wait until the next window before writing to CPMUARMCOP. Table 421 shows the duration of this window for the seven available COP rates. 0 Normal COP operation 1 Window COP operation
6 RSBCK	COP and RTI Stop in Active BDM Mode Bit 0 Allows the COP and RTI to keep running in Active BDM mode. 1 Stops the COP and RTI counters whenever the part is in Active BDM mode.
5 WRTMASK	**Write Mask for WCOP and CR[2:0] Bit** — This write-only bit serves as a mask for the WCOP and CR[2:0] bits while writing the CPMUCOP register. It is intended for BDM writing the RSBCK without changing the content of WCOP and CR[2:0]. 0 Write of WCOP and CR[2:0] has an effect with this write of CPMUCOP 1 Write of WCOP and CR[2:0] has no effect with this write of CPMUCOP. (Does not count for “write once”.)
2–0 CR[2:0]	**COP Watchdog Timer Rate Select** — These bits select the COP timeout rate (see Table 421). Writing a nonzero value to CR[2:0] enables the COP counter and starts the timeout period. A COP counter timeout causes a System Reset. This can be avoided by periodically (before timeout) initializing the COP counter via the CPMUARMCOP register. While all of the following four conditions are true the CR[2:0], WCOP bits are ignored and the COP operates at highest timeout period (2^{24} cycles) in normal COP mode (Window COP mode disabled): 1) COP is enabled (CR[2:0] is not 000) 2) BDM mode active 3) RSBCK = 0 4) Operation in Special mode

Table 421. COP watchdog rates

CR2	CR1	CR0	COPCLK Cycles to Time-out (COPCLK is either IRCCLK or OSCCLK depending on the COPOSCSEL bit)
0	0	0	COP disabled
0	0	1	2^{14}
0	1	0	2^{16}
0	1	1	2^{18}
1	0	0	2^{20}
1	0	1	2^{22}
1	1	0	2^{23}
1	1	1	2^{24}

4.23.3.2.10 Reserved register CPMUTEST0

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special mode can alter the 9S12I128PIMV1's functionality.

Table 422. Reserved register (CPMUTEST0)

0x003D

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Only in Special mode

4.23.3.2.11 Reserved register CPMUTEST1

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special mode can alter the 9S12I128PIMV1's functionality.

Table 423. Reserved register (CPMUTEST1)

0x003E

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Only in Special Mode

4.23.3.2.12 9S12I128PIMV1 COP Timer Arm/reset Register (CPMUARMCOP)

This register is used to restart the COP timeout period.

Table 424. 9S12I128PIMV1 CPMUARMCOP register

0x003F

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0

Read: Always reads $00

Write: Anytime

When the COP is disabled (CR[2:0] = "000") writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than $55 or $AA causes a COP reset. To restart the COP timeout period write $55 followed by a write of $AA. These writes do not need to occur back-to-back, but the sequence ($55, $AA) must be completed prior to COP end of timeout period to avoid a COP reset. Sequences of $55 writes are allowed. When the WCOP bit is set, $55 and $AA writes must be done in the last 25% of the selected timeout period; writing any value in the first 75% of the selected period will cause a COP reset.

4.23.3.2.13 Low-voltage control register (CPMULVCTL)

The CPMULVCTL register allows the configuration of the low-voltage detect features.

Table 425. Low-voltage control register (CPMULVCTL)

0x02F1

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LVDS	LVIE	LVIF
W	▨	▨	▨	▨	▨	▨		
Reset	0	0	0	0	0	U	0	U

The Reset state of LVDS and LVIF depends on the external supplied VDDXR level

▨ = Unimplemented or Reserved

Read: Anytime

Write: LVIE and LVIF are write anytime, LVDS is read only

Table 426. CPMULVCTL field descriptions

Field	Description
2 LVDS	**Low-voltage Detect Status Bit** — This read-only status bit reflects the voltage level on VDDXR. Writes have no effect. 0 Input voltage VDDXR is above level V_{LVID} or RPM. 1 Input voltage VDDRX is below level V_{LVIA} and FPM.
1 LVIE	Low-voltage Interrupt Enable Bit 0 Interrupt request is disabled. 1 Interrupt will be requested whenever LVIF is set.
0 LVIF	**Low-voltage Interrupt Flag** — LVIF is set to 1 when LVDS status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LVIE = 1), LVIF causes an interrupt request. 0 No change in LVDS bit. 1 LVDS bit has changed.

4.23.3.2.14 Reserved register CPMUTEST3

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special mode can alter the 9S12I128PIMV1's functionality.

Table 427. Reserved register (CPMUTEST3)

0x02F6

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Only in Special mode

4.23.3.2.15 9S12I128PIMV1 IRC1M trim registers (CPMUIRCTRIMH / CPMUIRCTRIML)

Table 429. 9S12I128PIMV1 IRC1M trim high register (CPMUIRCTRIMH)

0x02F8

	15	14	13	12	11	10	9	8
R	TCTRIM[4:0]					0	IRCTRIM[9:8]	
W								
Reset	F	F	F	F	0	0	F	F

After de-assert of System Reset a factory programmed trim value is automatically loaded from the Flash memory to provide trimmed Internal Reference Frequency f_{IRC1M_TRIM}.

Table 430. 9S12I128PIMV1 IRC1M trim low register (CPMUIRCTRIML)

0x02F9

	7	6	5	4	3	2	1	0
R	IRCTRIM[7:0]							
W								
Reset	F	F	F	F	F	F	F	F

After de-assert of System Reset a factory programmed trim value is automatically loaded from the Flash memory to provide trimmed Internal Reference Frequency f_{IRC1M_TRIM}.

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register). Else write has no effect

NOTE

Writes to these registers while PLLSEL=1 clears the LOCK and UPOSC status bits.

Table 431. CPMUIRCTRIMH/L field descriptions

Field	Description
15-11 TCTRIM	IRC1M temperature coefficient Trim Bits Trim bits for the Temperature Coefficient (TC) of the IRC1M frequency. Figure 93 shows the influence of the bits TCTRIM4:0] on the relationship between frequency and temperature. Figure 93 shows an approximate TC variation, relative to the nominal TC of the IRC1M (i.e. for TCTRIM[4:0]=0x00000 or 0x10000).
9-0 IRCTRIM	**IRC1M Frequency Trim Bits** — Trim bits for Internal Reference Clock After System Reset the factory programmed trim value is automatically loaded into these registers, resulting in an Internal Reference Frequency f_{IRC1M_TRIM}. See device electrical characteristics for value of f_{IRC1M_TRIM}. The frequency trimming consists of two different trimming methods: A rough trimming controlled by bits IRCTRIM[9:6] can be done with frequency leaps of about 6% in average. A fine trimming controlled by the bits IRCTRIM[5:0] can be done with frequency leaps of about 0.3% (this trimming determines the precision of the frequency setting of 0.15%, i.e. 0.3% is the distance between two trimming values). Figure 92 shows the relationship between the trim bits and the resulting IRC1M frequency.

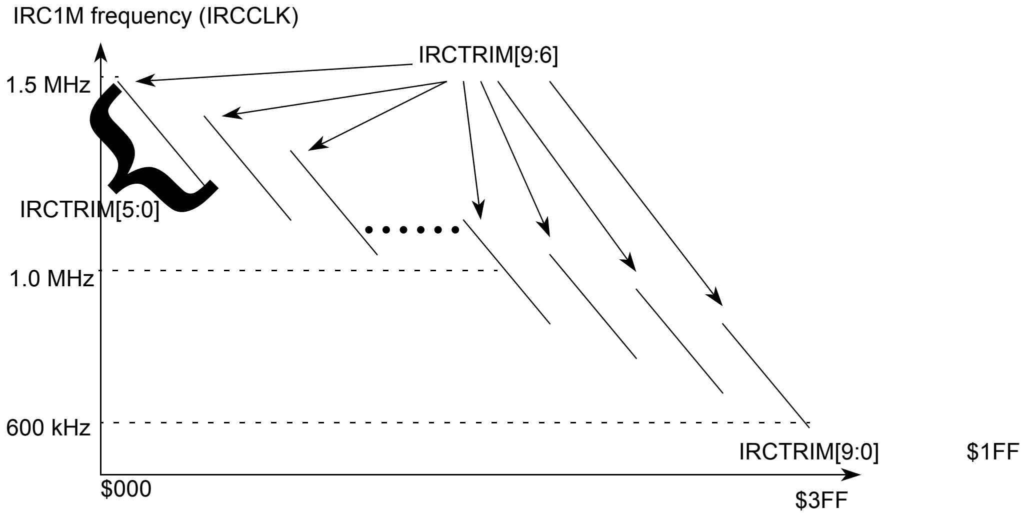


Figure 92. IRC1M frequency trimming diagram

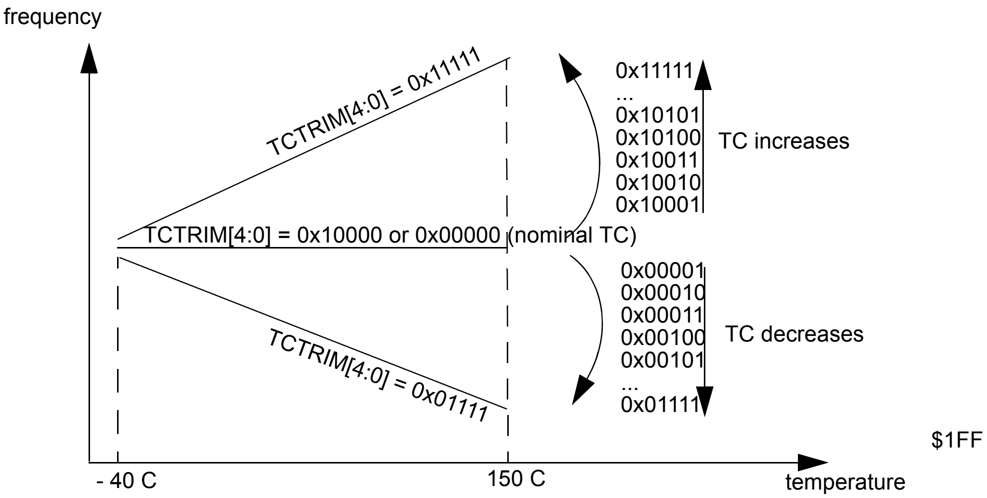


Figure 93. Influence of TCTRIM[4:0] on the temperature coefficient

NOTE

The frequency is not necessarily linear with the temperature (in most cases it will not be). The above diagram is meant only to give the direction (positive or negative) of the variation of the TC, relative to the nominal TC.

Setting TCTRIM[4:0] at 0x00000 or 0x10000 does not mean that the temperature coefficient will be zero. These two combinations basically switch off the TC compensation module, which result in the nominal TC of the IRC1M.

Table 432. TC Trimming of the Frequency of the IRC1M

TCTRIM[4:0]	IRC1M indicative relative TC variation	IRC1M indicative frequency drift for relative TC variation
00000	0 (nominal TC of the IRC)	0%
00001	-0.27%	-0.5%
00010	-0.54%	-0.9%
00011	-0.81%	-1.3%
00100	-1.08%	-1.7%

Table 432. TC Trimming of the Frequency of the IRC1M (continued)

TCTRIM[4:0]	IRC1M indicative relative TC variation	IRC1M indicative frequency drift for relative TC variation
00101	-1.35%	-2.0%
00110	-1.63%	-2.2%
00111	-1.9%	-2.5%
01000	-2.20%	-3.0%
01001	-2.47%	-3.4%
01010	-2.77%	-3.9%
01011	-3.04	-4.3%
01100	-3.33%	-4.7%
01101	-3.6%	-5.1%
01110	-3.91%	-5.6%
01111	-4.18%	-5.9%
10000	0 (nominal TC of the IRC)	0%
10001	+0.27%	+0.5%
10010	+0.54%	+0.9%
10011	+0.81%	+1.3%
10100	+1.07%	+1.7%
10101	+1.34%	+2.0%
10110	+1.59%	+2.2%
10111	+1.86%	+2.5%
11000	+2.11%	+3.0%
11001	+2.38%	+3.4%
11010	+2.62%	+3.9%
11011	+2.89%	+4.3%
11100	+3.12%	+4.7%
11101	+3.39%	+5.1%
11110	+3.62%	+5.6%
11111	+3.89%	+5.9%

NOTE

Since the IRC1M frequency is not a linear function of the temperature, but more like a parabola, the above relative variation is only an indication and should be considered with care.

Be aware that the output frequency varies with the TC trimming. A frequency trimming correction is therefore necessary. The values provided in Table 432 are typical values at ambient temperature which can vary from device to device.

4.23.3.2.16 9S12I128PIMV1 oscillator register (CPMUOSC)

This register configures the external oscillator (OSCLCP).

Table 433. 9S12I128PIMV1 oscillator register (CPMUOSC)

0x02FA

	7	6	5	4	3	2	1	0
R	OSCE	OSCBW	OSCPINS_EN	OSCFILT[4:0]				
W								
Reset	0	0	0	0	0	0	0	0

Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register), else write has no effect.

NOTE

Write to this register clears the LOCK and UPOSC status bits.

NOTE

If the chosen VCOCLK-to-OSCCLK ratio divided by two ((f_{VCO} / f_{OSC})/2) is not an integer number, the filter can not be used and the OSCFILT[4:0] bits must be set to 0.

NOTE

The frequency modulation (FM1 and FM0) can not be used if the Adaptive Oscillator Filter is enabled.

Table 434. CPMUOSC field descriptions

Field	Description
7 OSCE	**Oscillator Enable Bit** — This bit enables the external oscillator (OSCLCP). The UPOSC status bit in the CPMUFLG register indicates when the oscillation is stable and OSCCLK can be selected as bus clock or source of the COP or RTI. A loss of oscillation will lead to a clock monitor reset. 0 External oscillator is disabled. REFCLK for PLL is IRCCLK. 1 External oscillator is enabled.Clock monitor is enabled. REFCLK for PLL is external oscillator clock divided by REFDIV. **Note:** When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the startup time of the external oscillator t_{UPOSC} before entering Pseudo Stop mode.
6 OSCBW	**Oscillator Filter Bandwidth Bit** — If the VCOCLK frequency exceeds 25 MHz wide bandwidth must be selected. The Oscillator Filter is described in more detail in Section 4.23.4.5.2, "The adaptive oscillator filter" 0 Oscillator filter bandwidth is narrow (window for expected OSCCLK edge is one VCOCLK cycle). 1 Oscillator filter bandwidth is wide (window for expected OSCCLK edge is three VCOCLK cycles).
5 OSCPINS_EN	**Oscillator Pins EXTAL and XTAL Enable Bit** If OSCE=1 this read-only bit is set. It can only be cleared with the next reset. Enabling the external oscillator reserves the EXTAL and XTAL pins exclusively for oscillator application. 0 EXTAL and XTAL pins are not reserved for oscillator. 1 EXTAL and XTAL pins exclusively reserved for oscillator.
4-0 OSCFILT	**Oscillator Filter Bits** — When using the oscillator a noise filter can be enabled, which filters noise from the incoming external oscillator clock and detects if the external oscillator clock is qualified or not (quality status shown by bit UPOSC). The VCOCLK-to-OSCCLK ratio divided by two ((f_{VCO} / f_{OSC})/2) must be an integer value. This value must be written to the OSCFILT[4:0] bits to enable the Adaptive Oscillator Filter. 0x0000 Adaptive Oscillator Filter disabled, else Adaptive Oscillator Filter enabled]

4.23.3.2.17 9S12I128PIMV1 protection register (CPMUPROT)

This register protects the following clock configuration registers from accidental overwrite:

CPMUSYNR, CPMUREFDIV, CPMUCLKS, CPMUPLL, CPMUIRCTRIMH/L, and CPMUOSC

Table 435. 9S12I128PIMV1 protection register (CPMUPROT)

0x02FB

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	PROT
W	▨	▨	▨	▨	▨	▨	▨	
Reset	0	0	0	0	0	0	0	0

Read: Anytime

Write: Anytime

Table 436. CPMUPROT field description

Field	Description
0	**Clock Configuration Registers Protection Bit** — This bit protects the clock configuration registers from accidental overwrite (see list of protected registers above). Writing 0x26 to the CPMUPROT register clears the PROT bit, other write accesses set the PROT bit. 0 Protection of clock configuration registers is disabled. 1 Protection of clock configuration registers is enabled. (see list of protected registers above)

4.23.3.2.18 Reserved register CPMUTEST2

NOTE

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special mode can alter the 9S12I128PIMV1's functionality.

Table 437. Reserved register CPMUTEST2

0x02FC

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Only in Special mode

4.23.4 Functional description

4.23.4.1 Phase locked loop with internal filter (PLL)

The PLL is used to generate a high speed PLLCLK based on a low frequency REFCLK. The REFCLK is by default the IRCCLK which is trimmed to f_{IRC1M_TRIM}=1.0 MHz.

If using the oscillator (OSCE=1) REFCLK will be based on OSCCLK. For increased flexibility, OSCCLK can be divided in a range of 1 to 16 to generate the reference frequency REFCLK using the REFDIV[3:0] bits. Based on the SYNDIV[5:0] bits, the PLL generates the VCOCLK by multiplying the reference clock by a 2, 4, 6,... 126, 128. Based on the POSTDIV[4:0] bits, the VCOCLK can be divided in a range of 1,2, 3, 4, 5, 6,... to 32 to generate the PLLCLK.

.

If Oscillator is enabled (OSCE=1) $$f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$$

If Oscillator is disabled (OSCE=0) $$f_{REF} = f_{IRC1M}$$

$$f_{VCO} = 2 \times f_{REF} \times (SYNDIV + 1)$$

If PLL is locked (LOCK=1) $$f_{PLL} = \frac{f_{VCO}}{(POSTDIV + 1)}$$

If PLL is not locked (LOCK=0) $$f_{PLL} = \frac{f_{VCO}}{4}$$

If PLL is selected (PLLSEL=1) $$f_{bus} = \frac{f_{PLL}}{2}$$

NOTE

> Although it is possible to set the dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU.

Several examples of PLL divider settings are shown in Table 438. The following rules help to achieve optimum stability and shortest lock time:

- Use lowest possible f_{VCO} / f_{REF} ratio (SYNDIV value).
- Use highest possible REFCLK frequency f_{REF}.

Table 438. Examples of PLL divider settings

f_{osc}	REFDIV[3:0]	f_{REF}	REFFRQ[1:0]	SYNDIV[5:0]	f_{VCO}	VCOFRQ[1:0]	POSTDIV[4:0]	f_{PLL}	f_{bus}
off	$00	1.0 MHz	00	$1F	64 MHz	01	$03	16 MHz	8.0 MHz
off	$00	1.0 MHz	00	$1F	64 MHz	01	$00	64 MHz	32 MHz
off	$00	1.0 MHz	00	$0F	32 MHz	00	$00	32 MHz	16 MHz
4.0 MHz	$00	4.0 MHz	01	$03	32 MHz	01	$00	32 MHz	16 MHz

The phase detector inside the PLL compares the feedback clock (FBCLK = VCOCLK/(SYNDIV+1) with the reference clock (REFCLK = (IRC1M or OSCCLK)/(REFDIV+1)). Correction pulses are generated based on the phase difference between the two signals. The loop filter alters the DC voltage on the internal filter capacitor, based on the width and direction of the correction pulse, which leads to a higher or lower VCO frequency.

The user must select the range of the REFCLK frequency (REFFRQ[1:0] bits) and the range of the VCOCLK frequency (VCOFRQ[1:0] bits) to ensure that the correct PLL loop bandwidth is set.

The lock detector compares the frequencies of the FBCLK and the REFCLK. Therefore, the speed of the lock detector is directly proportional to the reference clock frequency. The circuit determines the lock condition based on this comparison.

If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and for instance check the LOCK bit. If interrupt requests are disabled, software can poll the LOCK bit continuously (during PLL start-up) or at periodic intervals. In either case, only when the LOCK bit is set, the VCOCLK will have stabilized to the programmed frequency.

- The LOCK bit is a read-only indicator of the locked state of the PLL.
- The LOCK bit is set when the VCO frequency is within the tolerance, Δ_{LOCK}, and is cleared when the VCO frequency is out of the tolerance, Δ_{UNI}.
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

4.23.4.2 Startup from reset

An example of startup of clock system from Reset is given in Figure 94.

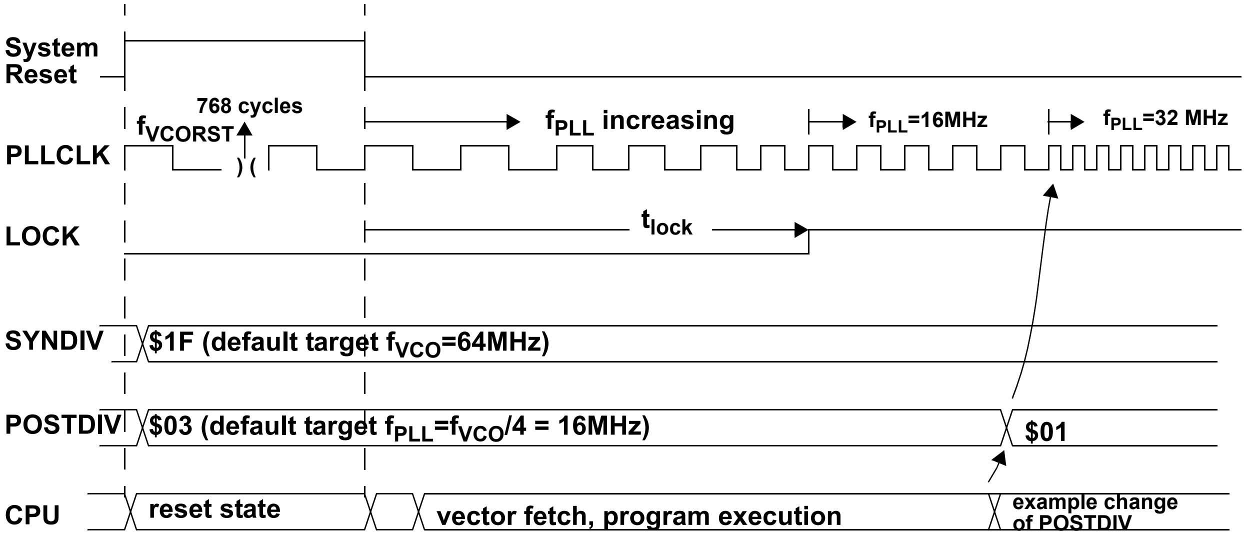

System Reset
768 cycles
fVCORST
PLLCLK
fPLL increasing
fPLL=16MHz
fPLL=32 MHz
LOCK
tlock
SYNDIV
$1F (default target fVCO=64MHz)
POSTDIV
$03 (default target fPLL=fVCO/4 = 16MHz)
$01
CPU
reset state
vector fetch, program execution
example change of POSTDIV

Figure 94. Startup of clock system after reset

4.23.4.3 Stop mode using PLLCLK as bus clock

An example of what happens going into Stop mode and exiting Stop mode after an interrupt is shown in Figure 95. Disable PLL Lock interrupt (LOCKIE=0) before going into Stop mode.

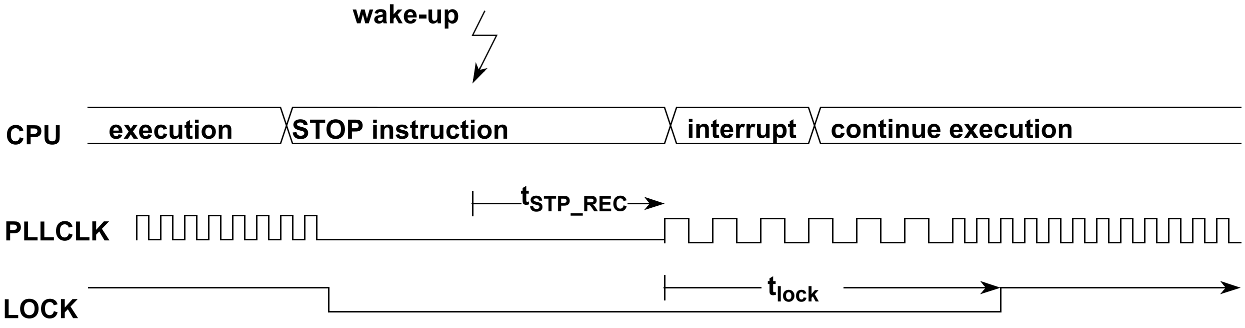


Figure 95. Stop mode using PLLCLK as bus clock

4.23.4.4 Full stop mode using oscillator clock as bus clock

An example of what happens going into Full Stop mode and exiting Full Stop mode after an interrupt is shown in Figure 96. Disable PLL Lock interrupt (LOCKIE=0) and oscillator status change interrupt (OSCIE=0) before going into Full Stop mode.

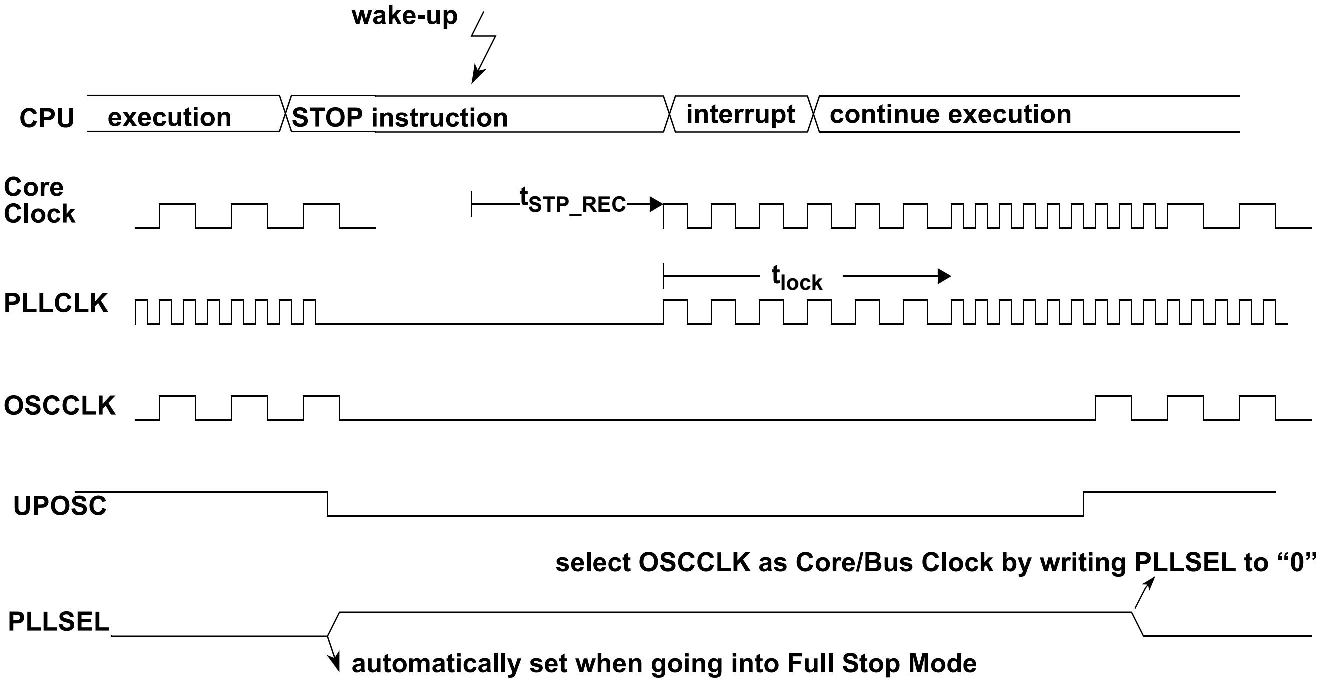


Figure 96. Full stop mode using oscillator clock as bus clock

4.23.4.5 External oscillator

4.23.4.5.1 Enabling the external oscillator

An example of how to use the oscillator as Bus Clock is shown in Figure 97.

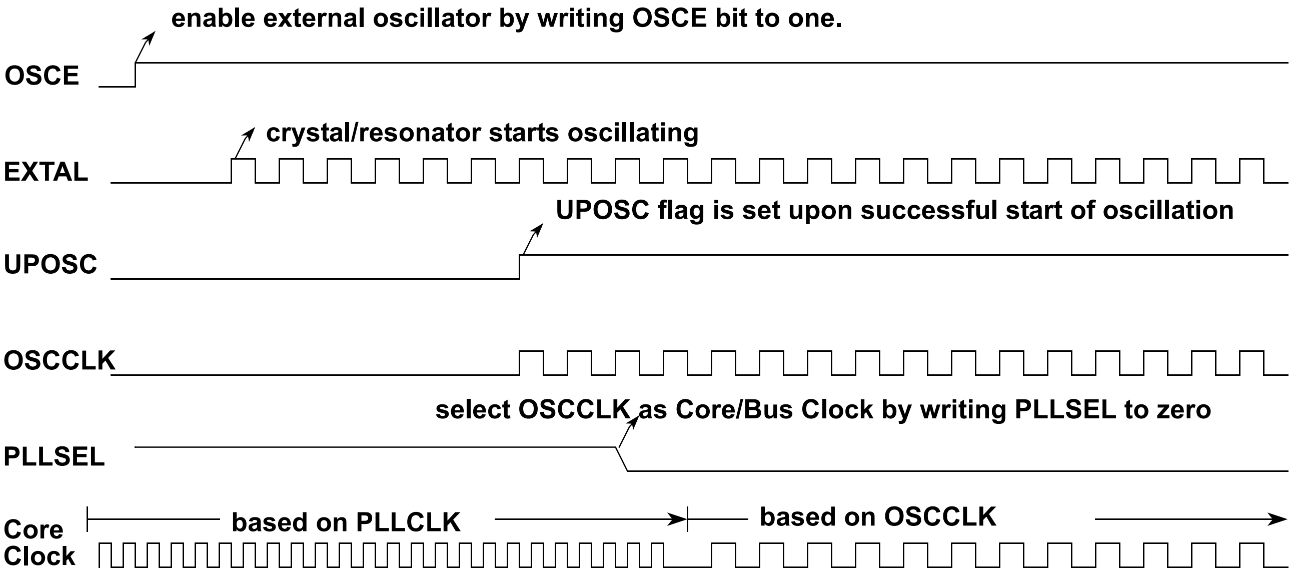

enable external oscillator by writing OSCE bit to one.
OSCE
crystal/resonator starts oscillating
EXTAL
UPOSC flag is set upon successful start of oscillation
UPOSC
OSCCLK
select OSCCLK as Core/Bus Clock by writing PLLSEL to zero
PLLSEL
Core Clock
based on PLLCLK
based on OSCCLK

Figure 97. Enabling the external oscillator

4.23.4.5.2 The adaptive oscillator filter

A spike in the oscillator clock can disturb the function of the modules driven by this clock. The Adaptive Oscillator Filter includes two features:

1. Filter noise (spikes) from the incoming external oscillator clock. The filter feature is illustrated in Figure 98.

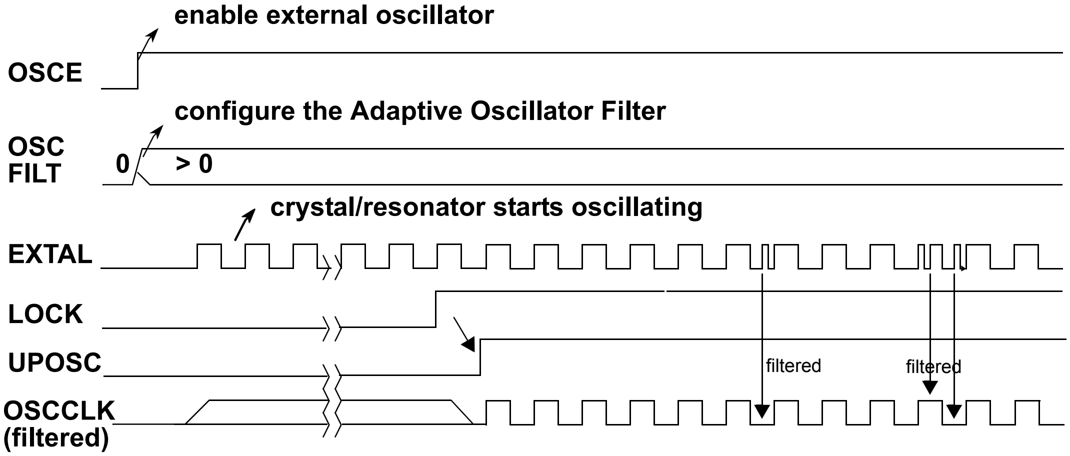

enable external oscillator
OSCE
configure the Adaptive Oscillator Filter
OSC FILT
0
> 0
crystal/resonator starts oscillating
EXTAL
LOCK
UPOSC
filtered
filtered
OSCCLK (filtered)

Figure 98. Noise filtered by the adaptive oscillator filter

2. Detect severe noise disturbance on external oscillator clock which can not be filtered and indicate the critical situation to the software by clearing the UPOSC and LOCK status bit and setting the OSCIF and LOCKIF flag. An example for the detection of critical noise is illustrated in Figure 99.

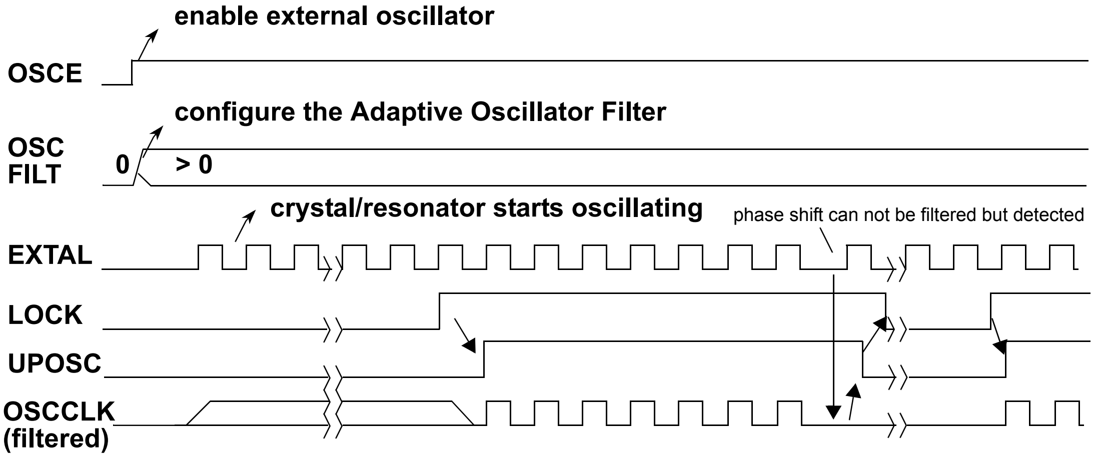

enable external oscillator
OSCE
configure the Adaptive Oscillator Filter
OSC FILT
0
> 0
crystal/resonator starts oscillating
phase shift can not be filtered but detected
EXTAL
LOCK
UPOSC
OSCCLK (filtered)

Figure 99. Critical noise detected by the adaptive oscillator filter

NOTE

> If the LOCK bit is clear due to severe noise disturbance on the external oscillator clock, the PLLCLK is derived from the VCO clock (with its actual frequency) divided by four (see Section 4.23.3.2.3, “9S12I128PIMV1 post divider register (CPMUPOSTDIV)").

The use of the filter function is only possible if the VCOCLK-to-OSCCLK ratio divided by two ((f_{VCO} / f_{OSC})/2) is an integer number. This integer value must be written to the OSCFILT[4:0] bits. If enabled, the Adaptive Oscillator Filter samples the incoming external oscillator clock signal (EXTAL) with the VCOCLK frequency.

Using VCOCLK, a time window is defined of which an edge of the OSCCLK is expected. In case of OSCBW = 1, the width of this window is three VCOCLK cycles, if the OSCBW = 0 it is one VCOCLK cycle. The noise detection is active for certain combinations of OSCFILT[4:0] and OSCBW bit settings, as shown in Table 439.

Table 439. Noise detection settings

OSCFILT[4:0]	OSCBW	Detection	Filter
0	x	disabled	disabled
1	x	disabled	active
2 or 3	0	active	active
	1	disabled	active
>=4	x	active	active

NOTE

If the VCOCLK frequency is higher than 25 MHz the wide bandwidth must be selected (OSCBW = 1).

4.23.4.6 System clock configurations

4.23.4.6.1 PLL engaged internal mode (PEI)

This mode is the default mode after System Reset or Power-on Reset. The Bus Clock is based on the PLLCLK, the reference clock for the PLL is internally generated (IRC1M). The PLL is configured to 64 MHz VCOCLK with POSTDIV set to 0x03. If locked (LOCK=1) this results in a PLLCLK of 16 MHz and a Bus Clock of 8.0 MHz. The PLL can be re-configured to other bus frequencies. The clock sources for COP and RTI are based on the internal reference clock generator (IRC1M).

4.23.4.6.2 PLL engaged external mode (PEE)

In this mode, the Bus Clock is based on the PLLCLK as well (like PEI). The reference clock for the PLL is based on the external oscillator. The adaptive spike filter and detection logic which uses the VCOCLK to filter and qualify the external oscillator clock can be enabled.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock.

This mode can be entered from default mode PEI by performing the following steps:

1. Configure the PLL for desired bus frequency.
2. Optionally the adaptive spike filter and detection logic can be enabled by calculating the integer value for the OSCFIL[4:0] bits and setting the bandwidth (OSCBW) accordingly.
3. Enable the external Oscillator (OSCE bit).
4. Wait for the PLL being locked (LOCK = 1) and the oscillator to start-up and additionally being qualified if the Adaptive Oscillator Filter is enabled (UPOSC =1).
5. Clear all flags in the CPMUFLG register to be able to detect any future status bit change.
6. Optionally status interrupts can be enabled (CPMUINT register).

Since the Adaptive Oscillator Filter (adaptive spike filter and detection logic) uses the VCOCLK to continuously filter and qualify the external oscillator clock, losing PLL lock status (LOCK=0), means losing the oscillator status information as well (UPOSC=0).

The impact of losing the oscillator status in PEE mode is as follows:

- The PLLCLK is derived from the VCO clock (with its actual frequency) divided by four until the PLL locks again.

Application software needs to be prepared to deal with the impact of losing the oscillator status at any time.

4.23.4.6.3 PLL bypassed external mode (PBE)

In this mode, the Bus Clock is based on the external oscillator clock. The reference clock for the PLL is based on the external oscillator. The adaptive spike filter and detection logic can be enabled which uses the VCOCLK to filter and qualify the external oscillator clock.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock.

This mode can be entered from default mode PEI by performing the following steps:

1. Make sure the PLL configuration is valid.
2. Optionally, the adaptive spike filter and detection logic can be enabled by calculating the integer value for the OSCFIL[4:0] bits and setting the bandwidth (OSCBW) accordingly.
3. Enable the external Oscillator (OSCE bit)

4. Wait for the PLL being locked (LOCK = 1) and the oscillator to start-up, and additionally being qualified if the Adaptive Oscillator Filter is enabled (UPOSC=1).
5. Clear all flags in the CPMUFLG register to be able to detect any status bit change.
6. Optionally status interrupts can be enabled (CPMUINT register).
7. Select the Oscillator Clock (OSCCLK) as Bus Clock (PLLSEL=0)

Since the Adaptive Oscillator Filter (adaptive spike filter and detection logic) uses the VCOCLK to continuously filter and qualify the external oscillator clock, losing PLL lock status (LOCK=0) means losing the oscillator status information as well (UPOSC=0).

The impact of losing the oscillator status in PBE mode is as follows:

- PLLSEL is set automatically and the Bus Clock is switched back to the PLLCLK.
- The PLLCLK is derived from the VCO clock (with its actual frequency) divided by four until the PLL locks again.

Application software needs to be prepared to deal with the impact of loosing the oscillator status at any time.

In the PBE mode, not every noise disturbance can be indicated by bits LOCK and UPOSC (both bits are based on the Bus Clock domain). There are clock disturbances possible, after which UPOSC and LOCK both stay asserted, while occasional pauses on the filtered OSCCLK and resulting Bus Clock occur. The adaptive spike filter is still functional and protects the Bus Clock from frequency overshoot due to spikes on the external oscillator clock. The filtered OSCCLK and resulting Bus Clock will pause until the PLL has stabilized again.

4.23.5 Resets

4.23.5.1 General

All reset sources are listed in Table 440. Refer to MCU specification for related vector addresses and priorities.

Table 440. Reset summary

Reset source	Local enable
Power-On Reset (POR)	None
Low Voltage Reset (LVR)	None
External pin $\overline{\text{RESET}}$	None
Illegal Address Reset	None
Clock Monitor Reset	OSCE Bit in CPMUOSC register
COP Reset	CR[2:0] in CPMUCOP register

4.23.5.2 Description of reset operation

Upon detection of any reset in Table 440, an internal circuit drives the $\overline{\text{RESET}}$ pin low for 512 PLLCLK cycles. After 512 PLLCLK cycles, the $\overline{\text{RESET}}$ pin is released. The reset generator of the 9S12I128PIMV1 waits for additional 256 PLLCLK cycles and then samples the $\overline{\text{RESET}}$ pin to determine the originating source. Table 441 shows which vector will be fetched.

Table 441. Reset vector selection

Sampled $\overline{\text{RESET}}$ Pin (256 cycles after release)	Oscillator monitor fail pending	COP timeout pending	Vector fetch
1	0	0	POR LVR Illegal Address Reset External pin $\overline{\text{RESET}}$
1	1	X	Clock Monitor Reset
1	0	1	COP Reset
0	X	X	POR LVR Illegal Address Reset External pin $\overline{\text{RESET}}$

NOTE

While System Reset is asserted, the PLLCLK runs with the frequency f_{VCORST}.

The internal reset of the MCU remains asserted while the reset generator completes the 768 PLLCLK cycles long reset sequence. In case the $\overline{\text{RESET}}$ pin is externally driven low for more than these 768 PLLCLK cycles (External Reset), the internal reset remains asserted longer.

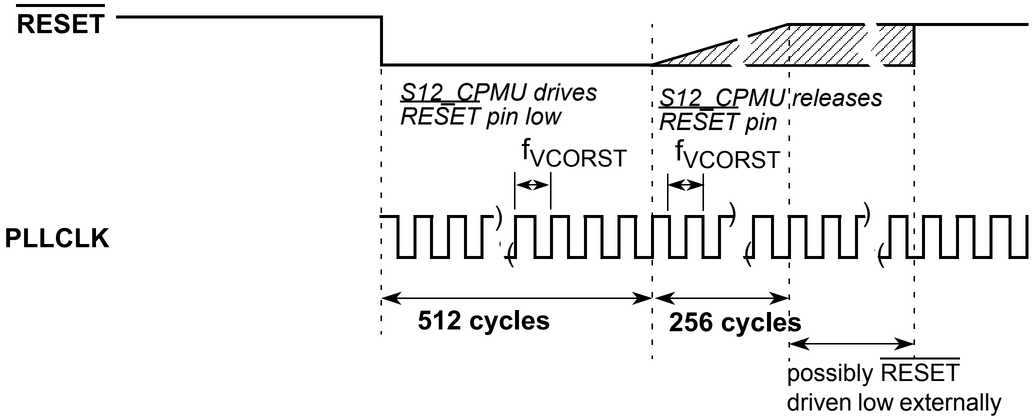


Figure 100. RESET timing

4.23.5.2.1 Clock monitor reset

When the external oscillator is enabled (OSCE=1), in case of a loss of oscillation or the oscillator frequency is below the failure assert frequency f_{CMFA} (see device electrical characteristics for values), the 9S12I128PIMV1 generates a clock monitor reset. In Full Stop mode the external oscillator and the clock monitor are disabled.

4.23.5.2.2 Computer operating properly watchdog (COP) reset

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out, it is an indication that the software is no longer being executed in the intended sequence, and a COP reset is generated.

The clock source for the COP is either IRCCLK or OSCCLK, depending on the setting of the COPOSCSEL bit. In Stop mode with PSTP=1 (Pseudo Stop mode), COPOSCSEL=1 and PCE=1 the COP continues to run, else the COP counter halts in Stop mode. Three control bits in the CPMUCOP register allow selection of seven COP timeout periods.

When COP is enabled, the program must write $55 and $AA (in this order) to the CPMUARMCOP register during the selected timeout period. Once this is done, the COP timeout period is restarted. If the program fails to do this and the COP times out, a COP reset is generated. Also, if any value other than $55 or $AA is written, a COP reset is generated.

Windowed COP operation is enabled by setting WCOP in the CPMUCOP register. In this mode, writes to the CPMUARMCOP register to clear the COP timer must occur in the last 25% of the selected timeout period. A premature write will immediately reset the part.

4.23.5.3 Power-on Reset (POR)

The on-chip POR circuitry detects when the internal supply VDD drops below an appropriate voltage level. The POR is deasserted if the internal supply VDD exceeds an appropriate voltage level (voltage levels are not specified in this document, because this internal supply is not visible on device pins).

4.23.5.4 Low-voltage Reset (LVR)

The on-chip LVR circuitry detects when one of the supply voltages VDD, VDDF, or VDDX, drops below an appropriate voltage level. If LVR is deasserted, the MCU is fully operational at the specified maximum speed. The LVR assert and deassert levels for the supply voltage VDDX are V_{LVRXA} and V_{LVRXD}, and are specified in the device reference manual.

4.23.6 Interrupts

The interrupt/reset vectors requested by the 9S12I128PIMV1 are listed in Table 442. Refer to MCU specification for related vector addresses and priorities.

Table 442. 9S12I128PIMV1 Interrupt Vectors

Interrupt source	CCR mask	Local enable
RTI timeout interrupt	I bit	CPMUINT (RTIE)
PLL lock interrupt	I bit	CPMUINT (LOCKIE)
Oscillator status interrupt	I bit	CPMUINT (OSCIE)
Low voltage interrupt	I bit	CPMULVCTL (LVIE)

4.23.6.1 Description of interrupt operation

4.23.6.1.1 Real time interrupt (RTI)

The clock source for the RTI is either IRCCLK or OSCCLK, depending on the setting of the RTIOSCSEL bit. In Stop mode with PSTP=1 (Pseudo Stop mode), RTIOSCSEL=1 and PRE=1 the RTI continues to run, else the RTI counter halts in Stop mode.

The RTI can be used to generate hardware interrupts at a fixed periodic rate. If enabled (by setting RTIE=1), this interrupt will occur at the rate selected by the CPMURTI register. At the end of the RTI timeout period, the RTIF flag is set to one and a new RTI timeout period starts immediately.

A write to the CPMURTI register restarts the RTI timeout period.

4.23.6.1.2 PLL lock interrupt

The 9S12I128PIMV1 generates a PLL Lock interrupt when the lock condition (LOCK status bit) of the PLL changes, either from a locked state to an unlocked state, or vice versa. Lock interrupts are locally disabled by setting the LOCKIE bit to zero. The PLL Lock interrupt flag (LOCKIF) is set to 1 when the lock condition has changed, and is cleared to 0 by writing a 1 to the LOCKIF bit.

4.23.6.1.3 Oscillator status interrupt

The Adaptive Oscillator filter contains two different features:

1. Filters spikes of the external oscillator clock.

2. Qualify the external oscillator clock.

When the OSCE bit is 0, then UPOSC stays 0. When OSCE=1 and OSCFILT = 0, then the filter is transparent and no spikes are filtered. The UPOSC bit is then set after the LOCK bit is set.

Upon detection of a status change (UPOSC), where an unqualified oscillation becomes qualified or vice versa, the OSCIF flag is set. Going into Full Stop mode or disabling the oscillator can also cause a status change of UPOSC.

Since the Adaptive Oscillator Filter is based on the PLLCLK, any change in PLL configuration or any other event which causes the PLL lock status to be cleared, leads to a loss of the oscillator status information as well (UPOSC=0).

Oscillator status change interrupts are locally enabled with the OSCIE bit.

NOTE

Losing the oscillator status (UPOSC=0) affects the clock configuration of the system[272]. This needs to be addressed in application software.

Notes

[272] For details refer to "Section 4.23.4.6, "System clock configurations""

4.23.6.1.4 Low-voltage interrupt (LVI)

In FPM, the input voltage VDDXR is monitored. Whenever VDDXR drops below level V_{LVIA}, the status bit LVDS is set to 1. When VDDXR rises above level V_{LVID}, the status bit LVDS is cleared to 0. An interrupt, indicated by flag LVIF = 1, is triggered by any change of the status bit LVDS if interrupt enable bit LVIE = 1.

4.23.7 Initialization/application information

4.23.7.1 General initialization information

Usually applications run in MCU Normal mode. It is recommended to write the CPMUCOP register from the application program initialization routine after reset, regardless if the COP is used in the application, even if a configuration is loaded via the flash memory after reset. By doing a “controlled” write access in MCU Normal mode (with the right value for the application), the write once for the COP configuration bits (WCOP,CR[2:0]) takes place, which protects these bits from further accidental change. If there is a program sequencing issue (code runaway), the COP configuration cannot be accidentally modified.

4.24 MCU - serial peripheral interface (S129S12I128PIMV1V5)

4.24.1 Introduction

The SPI module allows a duplex, synchronous, serial communication, between the MCU and peripheral devices. Software can poll the SPI status flags or the SPI operation can be interrupt driven.

4.24.1.1 Glossary of terms

Table 443. Term Definition

SPI	Serial Peripheral Interface
SS	Slave Select
SCK	Serial Clock
MOSI	Master Output, Slave Input
MISO	Master Input, Slave Output
MOMI	Master Output, Master Input
SISO	Slave Input, Slave Output

4.24.1.2 Features

The 9S12I128PIMV1 includes these distinctive features:

- Master mode and slave mode
- Selectable 8 or 16-bit transfer width
- Bidirectional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase
- Control of SPI operation during Wait mode

4.24.1.3 Modes of operation

The SPI functions in three modes: run, wait, and stop.

- Run mode
  This is the basic mode of operation.
- Wait mode
  SPI operation in Wait mode is a configurable low power mode, controlled by the SPISWAI bit located in the SPICR2 register. In Wait mode, if the SPISWAI bit is clear, the SPI operates like in Run mode. If the SPISWAI bit is set, the SPI goes into a power conservative state, with the SPI clock generation turned off. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into Run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.

- Stop mode
  The SPI is inactive in stop mode for reduced power consumption. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into Run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.

For a detailed description of operating modes, refer to Section 4.24.4.7, “Low power mode options".

4.24.1.4 Block diagram

Figure 101 gives an overview on the SPI architecture. The main parts of the SPI are status, control and data registers, shifter logic, baud rate generator, master/slave control logic, and port control logic.

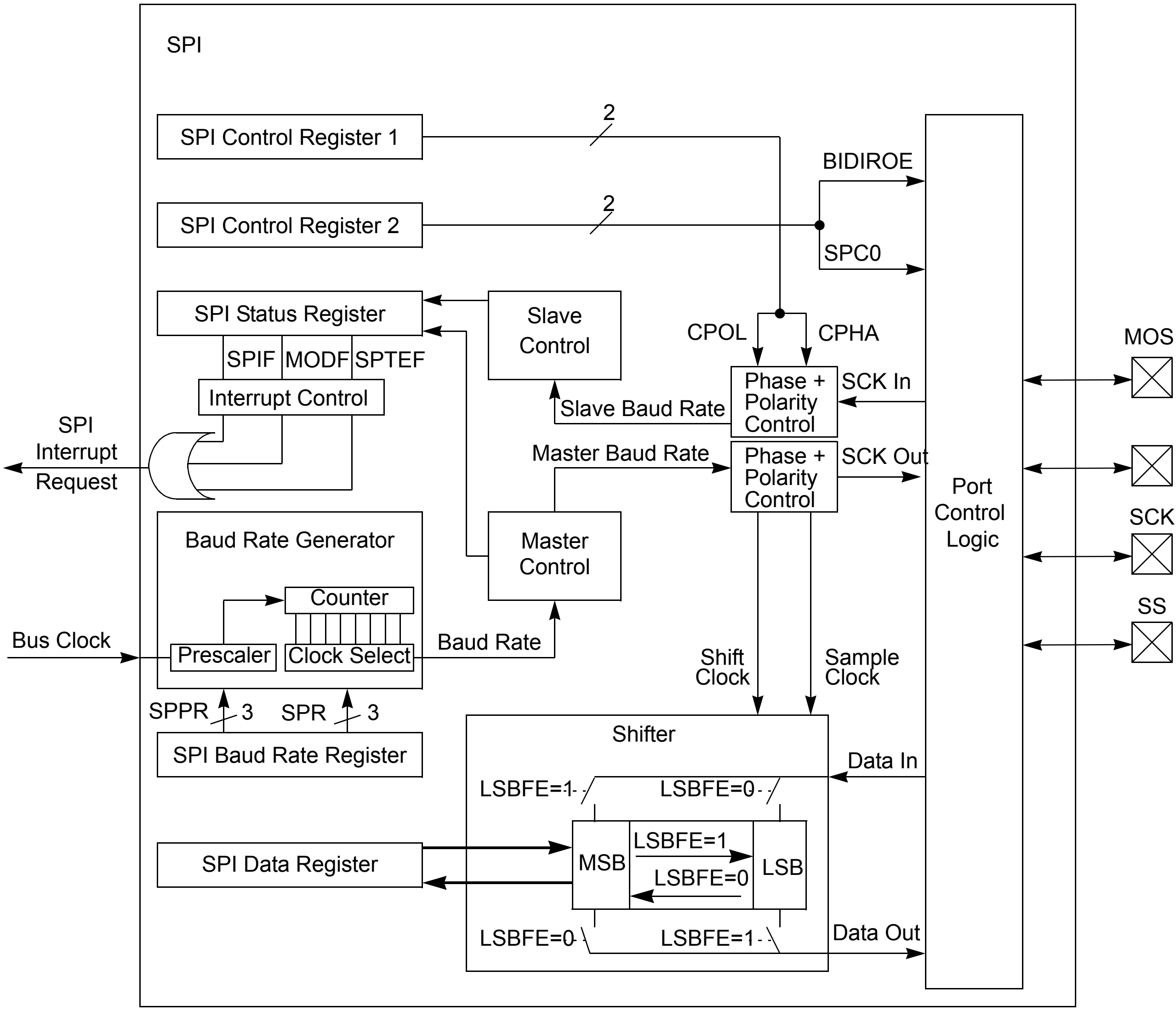


Figure 101. SPI block diagram

4.24.2 External signal description

This section lists the name and description of all ports including inputs and outputs that do, or may, connect off chip. The 9S12I128PIMV1 module has a total of four external pins.

4.24.2.1 MOSI — master out/slave in pin

This pin is used to transmit data out of the SPI module when it is configured as a master and receive data when it is configured as slave.

4.24.2.2 MISO — master in/slave out pin

This pin is used to transmit data out of the SPI module when configured as a slave and receive data when configured as master.

4.24.2.3 $\overline{\text{SS}}$ — slave select pin

This pin is used to output the select signal from the SPI module to another peripheral, with which a data transfer is to take place when it is configured as a master, and is used as an input to receive the slave select signal when the SPI is configured as a slave.

4.24.2.4 SCK — serial clock pin

In master mode, this is the synchronous output clock. In slave mode, this is the synchronous input clock.

4.24.3 Memory map and register definition

This section provides a detailed description of address space and registers used by the SPI.

4.24.3.1 Module memory map

The memory map for the 9S12I128PIMV1 is given in Table 444. The address listed for each register is the sum of a base address and an address offset. The base address is defined at the SoC level and the address offset is defined at the module level. Reads from the reserved bits return zeros and writes to the reserved bits have no effect.

Table 444. SPI register summary

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x00E8 SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
	W								
0x00E9 SPICR2	R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
	W	▨		▨			▨		
0x00EA SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
	W	▨				▨			
0x00EB SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x00EC SPIDRH	R	R15	R14	R13	R12	R11	R10	R9	R8
	W	T15	T14	T13	T12	T11	T10	T9	T8
0x00ED SPIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0
0x00EE Reserved	R	▨	▨	▨	▨	▨	▨	▨	▨
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x00EF Reserved	R	▨	▨	▨	▨	▨	▨	▨	▨
	W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

4.24.3.2 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

4.24.3.2.1 SPI control register 1 (SPICR1)

Table 445. SPI control register 1 (SPICR1)

0x00E8

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Read: Anytime

Write: Anytime

Table 446. SPICR1 field descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set. 0 SPI interrupts disabled. 1 SPI interrupts enabled.
6 SPE	**SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset. 0 SPI disabled (lower power consumption). 1 SPI enabled, port pins are dedicated to SPI functions.
5 SPTIE	**SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set. 0 SPTEF interrupt disabled. 1 SPTEF interrupt enabled.
4 MSTR	**SPI Master/Slave Mode Select Bit** — This bit selects whether the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state. 0 SPI is in slave mode. 1 SPI is in master mode.
3 CPOL	**SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Active-high clocks selected. In idle state SCK is low. 1 Active-low clocks selected. In idle state SCK is high.
2 CPHA	**SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Sampling of data occurs at odd edges (1,3,5,...) of the SCK clock. 1 Sampling of data occurs at even edges (2,4,6,...) of the SCK clock.
1 SSOE	**Slave Select Output Enable** — The $\overline{\text{SS}}$ output feature is enabled only in master mode, if MODFEN is set, by asserting the SSOE as shown in Table 447. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.
0 LSBFE	**LSB-First Enable** — This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in the highest bit position. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Data is transferred most significant bit first. 1 Data is transferred least significant bit first.

Table 447. $\overline{SS}$ input/output selection

MODFEN	SSOE	Master mode	Slave mode
0	0	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
0	1	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
1	0	$\overline{SS}$ input with MODF feature	$\overline{SS}$ input
1	1	$\overline{SS}$ is slave select output	$\overline{SS}$ input

4.24.3.2.2 SPI control register 2 (SPICR2)

Table 448. SPI control register 2 (SPICR2)

0x00E9

	7	6	5	4	3	2	1	0
R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W	▨		▨			▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

Table 449. SPICR2 field descriptions

Field	Description
6 XFRW	**Transfer Width —** This bit is used for selecting the data transfer width. If 8-bit transfer width is selected, SPIDRL becomes the dedicated data register and SPIDRH is unused. If 16-bit transfer width is selected, SPIDRH and SPIDRL form a 16-bit data register. Refer to Section 4.24.3.2.4, "SPI status register (SPISR)" for information about transmit/receive data handling and the interrupt flag clearing mechanism. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 8-bit Transfer Width (n = 8)[273] 1 16-bit Transfer Width (n = 16)[273]
4 MODFEN	**Mode Fault Enable Bit** — This bit allows the MODF failure to be detected. If the SPI is in master mode and MODFEN is cleared, then the $\overline{SS}$ port pin is not used by the SPI. In slave mode, the $\overline{SS}$ is available only as an input regardless of the value of MODFEN. For an overview on the impact of the MODFEN bit on the $\overline{SS}$ port pin configuration, refer to Table 447. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 $\overline{SS}$ port pin is not used by the SPI. 1 $\overline{SS}$ port pin with MODF feature.
3 BIDIROE	**Output Enable in the Bidirectional Mode of Operation** — This bit controls the MOSI and MISO output buffer of the SPI, when in bidirectional mode of operation (SPC0 is set). In master mode, this bit controls the output buffer of the MOSI port, in slave mode it controls the output buffer of the MISO port. In master mode, with SPC0 set, a change of this bit will abort a transmission in progress and force the SPI into idle state. 0 Output buffer disabled. 1 Output buffer enabled.
1 SPISWAI	**SPI Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode. 0 SPI clock operates normally in wait mode. 1 Stop SPI clock generation when in wait mode.
0 SPC0	**Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 450. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.

Notes

[273] n is used later in this document as a placeholder for the selected transfer width.

Table 450. Bidirectional pin configurations

Pin Mode	SPC0	BIDIROE	MISO	MOSI
Master Mode of Operation				
Normal	0	X	Master In	Master Out
Bidirectional	1	0	MISO not used by SPI	Master In
		1		Master I/O
Slave Mode of Operation				
Normal	0	X	Slave Out	Slave In
Bidirectional	1	0	Slave In	MOSI not used by SPI
		1	Slave I/O	

4.24.3.2.3 SPI baud rate register (SPIBR)

Table 451. SPI baud rate register (SPIBR)

0x00EA

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W	▨				▨			
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Anytime; writes to the reserved bits have no effect

Table 452. SPIBR field descriptions

Field	Description
6–4 SPPR[2:0]	**SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 453. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.
2–0 SPR[2:0]	**SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 453. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.

The baud rate divisor equation is as follows:

$$\mathbf{BaudRateDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}} \qquad \textit{Eqn. 4}$$

The baud rate can be calculated with the following equation:

$$\mathbf{Baud\ Rate = BusClock / BaudRateDivisor} \qquad \textit{Eqn. 5}$$

NOTE

For maximum allowed baud rates, refer to Section 3.6.2.5, “SPI timing" of this data sheet.

Table 453. Example SPI baud rate selection (25 MHz bus clock)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud rate divisor	Baud rate
0	0	0	0	0	0	2	12.5 Mbit/s
0	0	0	0	0	1	4	6.25 Mbit/s
0	0	0	0	1	0	8	3.125 Mbit/s

Table 453. Example SPI baud rate selection (25 MHz bus clock) (continued)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud rate divisor	Baud rate
0	0	0	0	1	1	16	1.5625 Mbit/s
0	0	0	1	0	0	32	781.25 kbit/s
0	0	0	1	0	1	64	390.63 kbit/s
0	0	0	1	1	0	128	195.31 kbit/s
0	0	0	1	1	1	256	97.66 kbit/s
0	0	1	0	0	0	4	6.25 Mbit/s
0	0	1	0	0	1	8	3.125 Mbit/s
0	0	1	0	1	0	16	1.5625 Mbit/s
0	0	1	0	1	1	32	781.25 kbit/s
0	0	1	1	0	0	64	390.63 kbit/s
0	0	1	1	0	1	128	195.31 kbit/s
0	0	1	1	1	0	256	97.66 kbit/s
0	0	1	1	1	1	512	48.83 kbit/s
0	1	0	0	0	0	6	4.16667 Mbit/s
0	1	0	0	0	1	12	2.08333 Mbit/s
0	1	0	0	1	0	24	1.04167 Mbit/s
0	1	0	0	1	1	48	520.83 kbit/s
0	1	0	1	0	0	96	260.42 kbit/s
0	1	0	1	0	1	192	130.21 kbit/s
0	1	0	1	1	0	384	65.10 kbit/s
0	1	0	1	1	1	768	32.55 kbit/s
0	1	1	0	0	0	8	3.125 Mbit/s
0	1	1	0	0	1	16	1.5625 Mbit/s
0	1	1	0	1	0	32	781.25 kbit/s
0	1	1	0	1	1	64	390.63 kbit/s
0	1	1	1	0	0	128	195.31 kbit/s
0	1	1	1	0	1	256	97.66 kbit/s
0	1	1	1	1	0	512	48.83 kbit/s
0	1	1	1	1	1	1024	24.41 kbit/s
1	0	0	0	0	0	10	2.5 Mbit/s
1	0	0	0	0	1	20	1.25 Mbit/s
1	0	0	0	1	0	40	625 kbit/s
1	0	0	0	1	1	80	312.5 kbit/s
1	0	0	1	0	0	160	156.25 kbit/s
1	0	0	1	0	1	320	78.13 kbit/s
1	0	0	1	1	0	640	39.06 kbit/s
1	0	0	1	1	1	1280	19.53 kbit/s

Table 453. Example SPI baud rate selection (25 MHz bus clock) (continued)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud rate divisor	Baud rate
1	0	1	0	0	0	12	2.08333 Mbit/s
1	0	1	0	0	1	24	1.04167 Mbit/s
1	0	1	0	1	0	48	520.83 kbit/s
1	0	1	0	1	1	96	260.42 kbit/s
1	0	1	1	0	0	192	130.21 kbit/s
1	0	1	1	0	1	384	65.10 kbit/s
1	0	1	1	1	0	768	32.55 kbit/s
1	0	1	1	1	1	1536	16.28 kbit/s
1	1	0	0	0	0	14	1.78571 Mbit/s
1	1	0	0	0	1	28	892.86 kbit/s
1	1	0	0	1	0	56	446.43 kbit/s
1	1	0	0	1	1	112	223.21 kbit/s
1	1	0	1	0	0	224	111.61 kbit/s
1	1	0	1	0	1	448	55.80 kbit/s
1	1	0	1	1	0	896	27.90 kbit/s
1	1	0	1	1	1	1792	13.95 kbit/s
1	1	1	0	0	0	16	1.5625 Mbit/s
1	1	1	0	0	1	32	781.25 kbit/s
1	1	1	0	1	0	64	390.63 kbit/s
1	1	1	0	1	1	128	195.31 kbit/s
1	1	1	1	0	0	256	97.66 kbit/s
1	1	1	1	0	1	512	48.83 kbit/s
1	1	1	1	1	0	1024	24.41 kbit/s
1	1	1	1	1	1	2048	12.21 kbit/s

4.24.3.2.4 SPI status register (SPISR)

Table 454. SPI status register (SPISR)

0x00EB

	7	6	5	4	3	2	1	0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	1	0	0	0	0	0

▨ = Unimplemented or Reserved

Read: Anytime

Write: Has no effect

Table 455. SPISR field descriptions

Field	Description
7 SPIF	**SPIF Interrupt Flag** — This bit is set after received data has been transferred into the SPI data register. For information about clearing SPIF Flag, refer to Table 456. 0 Transfer not yet complete. 1 New data copied to SPIDR.
5 SPTEF	**SPI Transmit Empty Interrupt Flag** — If set, this bit indicates that the transmit data register is empty. For information about clearing this bit and placing data into the transmit data register, refer to Table 457. 0 SPI data register not empty. 1 SPI data register empty.
4 MODF	**Mode Fault Flag** — This bit is set if the $\overline{SS}$ input becomes low while the SPI is configured as a master and mode fault detection is enabled, MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in Section 4.24.3.2.2, “SPI control register 2 (SPICR2)"”. The flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to the SPI control register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

Table 456. SPIF interrupt flag clearing sequence

XFRW bit	SPIF interrupt flag clearing sequence			
0	Read SPISR with SPIF == 1	then	Read SPIDRL	
1	Read SPISR with SPIF == 1	then	Byte Read SPIDRL [274]	
			or	
			Byte Read SPIDRH [275]	Byte Read SPIDRL
			or	
			Word Read (SPIDRH:SPIDRL)	

Notes

[274] Data in SPIDRH is lost, in this case.

[275] SPIDRH can be read repeatedly without any effect on SPIF. SPIF Flag is cleared only by the read of SPIDRL after reading SPISR with SPIF == 1.

Table 457. SPTEF interrupt flag clearing sequence

XFRW bit	SPTEF interrupt flag clearing sequence			
0	Read SPISR with SPTEF == 1	then	Write to SPIDRL [276]	
1	Read SPISR with SPTEF == 1	then	Byte Write to SPIDRL [276] [277]	
			or	
			Byte Write to SPIDRH [276] [278]	Byte Write to SPIDRL [276]
			or	
			Word Write to (SPIDRH:SPIDRL) [276]	

Notes

[276] Any write to SPIDRH or SPIDRL with SPTEF == 0 is effectively ignored.

[277] Data in SPIDRH is undefined in this case.

[278] SPIDRH can be written repeatedly without any effect on SPTEF. SPTEF Flag is cleared only by writing to SPIDRL after reading SPISR with SPTEF == 1.

4.24.3.2.5 SPI data register (SPIDR = SPIDRH:SPIDRL)

Table 458. SPI data register high (SPIDRH)

0x00EC

	7	6	5	4	3	2	1	0
R	R15	R14	R13	R12	R11	R10	R9	R8
W	T15	T14	T13	T12	T11	T10	T9	T8
Reset	0	0	0	0	0	0	0	0

Table 459. SPI data register low (SPIDRL)

0x00ED

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Read: Anytime; read data only valid when SPIF is set
Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows data to be queued and transmitted. For an SPI configured as a master, queued data is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data. Received data in the SPIDR is valid when SPIF is set.

If SPIF is cleared and data has been received, the received data is transferred from the receive shift register to the SPIDR and SPIF is set.

If SPIF is set and not serviced, and a second data value has been received, the second received data is kept as valid data in the receive shift register until the start of another transmission. The data in the SPIDR does not change.

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced before the start of a third transmission, the data in the receive shift register is transferred into the SPIDR and SPIF remains set (see Figure 102).

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced after the start of a third transmission, the data in the receive shift register has become invalid and is not transferred into the SPIDR (see Figure 103).

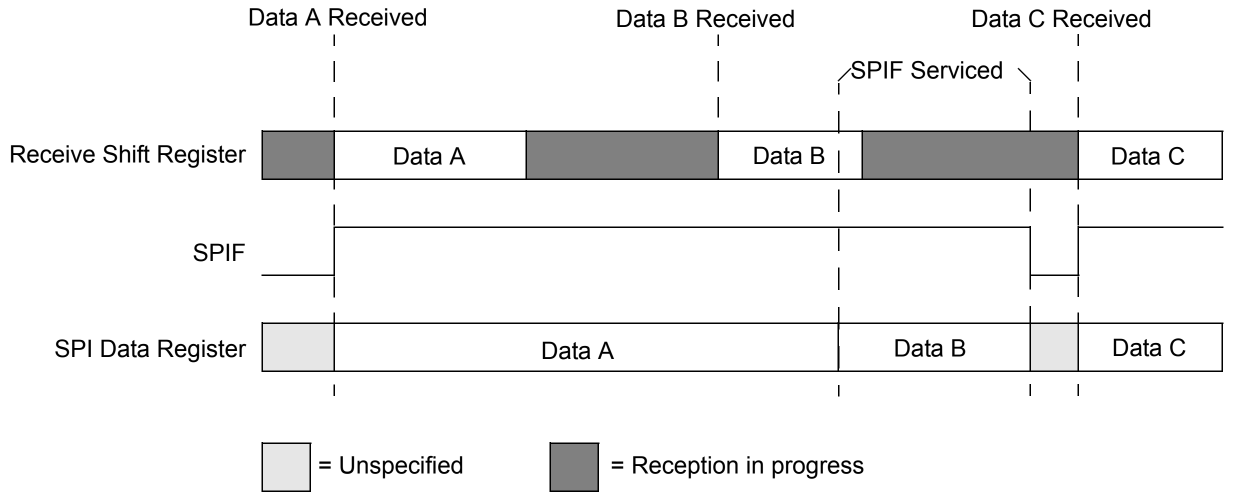


Figure 102. Reception with SPIF serviced in time

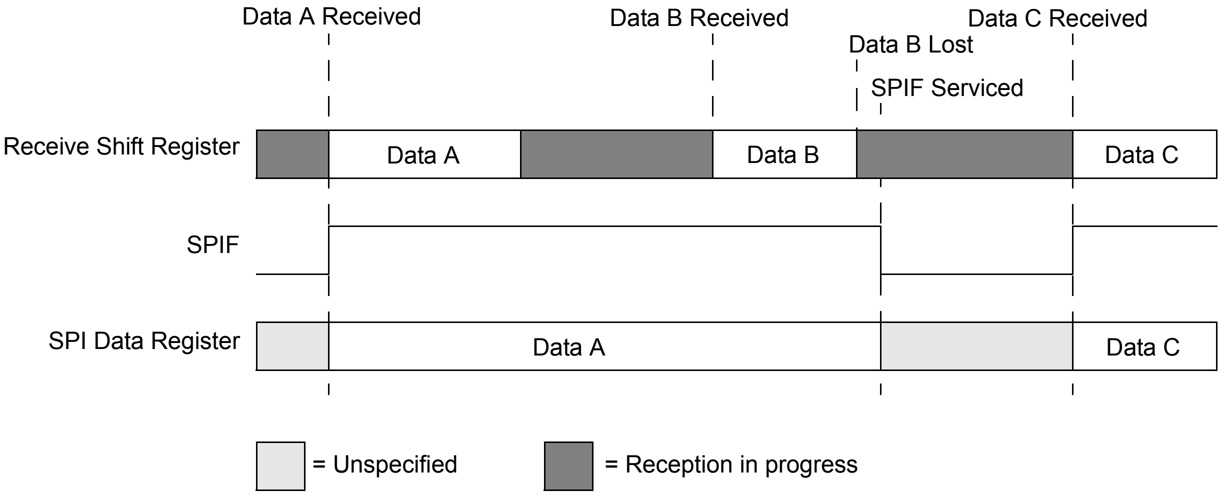


Figure 103. Reception with SPIF serviced too late

4.24.4 Functional description

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or SPI operation can be interrupt driven.

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{SS}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

The main element of the SPI system is the SPI data register. The n-bit[(279)] data register in the master and the n-bit[(279)] data register in the slave are linked by the MOSI and MISO pins to form a distributed 2n-bit[(279)] register. When a data transfer operation is performed, this 2n-bit[(279)] register is serially shifted n[(279)] bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

Notes

279. n depends on the selected transfer width, refer to Section 4.24.3.2.2, “SPI control register 2 (SPICR2)"

A read of SPISR with SPTEF = 1 followed by a write to SPIDR, puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A common SPI data register address is shared for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Section 4.24.4.3, “Transmission formats").

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

NOTE

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register will destroy the received byte and must be avoided.

4.24.4.1 Master mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, data immediately transfers to the shift register. Data begins shifting out on the MOSI pin under the control of the serial clock.

- Serial clock

  The SPR2, SPR1, and SPR0 baud rate selection bits, in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register, control the baud rate generator and determine the speed of the transmission. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.
- MOSI, MISO pin

  In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.
- $\overline{\text{SS}}$ pin

  If MODFEN and SSOE are set, the $\overline{\text{SS}}$ pin is configured as slave select output. The $\overline{\text{SS}}$ output becomes low during each transmission and is high when the SPI is in idle state.

  If MODFEN is set and SSOE is cleared, the $\overline{\text{SS}}$ pin is configured as input for detecting mode fault error. If the $\overline{\text{SS}}$ input becomes low, this indicates a mode fault error where another master tries to drive the MOSI and SCK lines. In this case, the SPI immediately switches to slave mode, by clearing the MSTR bit and also disables the slave output buffer MISO (or SISO in bidirectional mode). The result is that all outputs are disabled and SCK, MOSI, and MISO are inputs. If a transmission is in progress when the mode fault occurs, the transmission is aborted and the SPI is forced into idle state.

  This mode fault error also sets the mode fault (MODF) flag in the SPI status register (SPISR). If the SPI interrupt enable bit (SPIE) is set when the MODF flag becomes set, then an SPI interrupt sequence is also requested.

  When a write to the SPI data register in the master occurs, there is a half SCK-cycle delay. After the delay, SCK is started within the master. The rest of the transfer operation differs slightly, depending on the clock format specified by the SPI clock phase bit, CPHA, in SPI control register 1 (see Section 4.24.4.3, “Transmission formats").

NOTE

> A change of the bits CPOL, CPHA, SSOE, LSBFE, XFRW, MODFEN, SPC0, or BIDIROE with SPC0 set, SPPR2-SPPR0, and SPR2-SPR0 in master mode, will abort a transmission in progress and force the SPI into idle state. The remote slave cannot detect this, therefore the master must ensure that the remote slave is returned to idle state.

4.24.4.2 Slave mode

The SPI operates in slave mode when the MSTR bit in SPI control register 1 is clear.

- Serial clock

  In slave mode, SCK is the SPI clock input from the master.
- MISO, MOSI pins

  In slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit and BIDIROE bit in SPI control register 2.
- $\overline{\text{SS}}$ pin

  The $\overline{\text{SS}}$ pin is the slave select input. Before a data transmission occurs, the $\overline{\text{SS}}$ pin of the slave SPI must be low. $\overline{\text{SS}}$ must remain low until the transmission is complete. If $\overline{\text{SS}}$ goes high, the SPI is forced into idle state.

  The $\overline{\text{SS}}$ input also controls the serial data output pin. If $\overline{\text{SS}}$ is high (not selected), the serial data output pin is high impedance, and, if $\overline{\text{SS}}$ is low, the first bit in the SPI data register is driven out of the serial data output pin. Also, if the slave is not selected ($\overline{\text{SS}}$ is high), then the SCK input is ignored and no internal shifting of the SPI shift register occurs.

  Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a slave mode. For these simpler devices, there is no serial data out pin.

NOTE

> When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave’s serial data output line.

As long as no more than one slave device drives the system slave’s serial data output line, it is possible for several slaves to receive the same transmission from a master, although the master would not receive return information from all of the receiving slaves.

If the CPHA bit in SPI control register 1 is clear, odd numbered edges on the SCK input cause the data at the serial data input pin to be latched. Even numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

If the CPHA bit is set, even numbered edges on the SCK input cause the data at the serial data input pin to be latched. Odd numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

When CPHA is set, the first edge is used to get the first data bit onto the serial data output pin. When CPHA is clear and the $\overline{SS}$ input is low (slave selected), the first bit of the SPI data is driven out of the serial data output pin. After the nth[280] shift, the transfer is considered complete and the received data is transferred into the SPI data register. To indicate transfer is complete, the SPIF flag in the SPI status register is set.

Notes

[280] n depends on the selected transfer width, refer to Section 4.24.3.2.2, “SPI control register 2 (SPICR2)"

NOTE

> A change of the bits CPOL, CPHA, SSOE, LSBFE, MODFEN, SPC0, or BIDIROE with SPC0 set in slave mode, will corrupt a transmission in progress and must be avoided.

4.24.4.3 Transmission formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

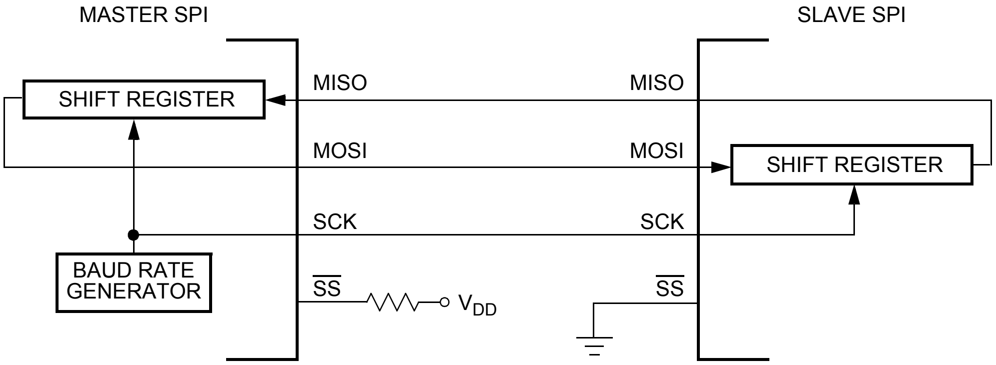


Figure 104. Master/slave transfer block diagram

4.24.4.3.1 Clock phase and polarity controls

Using two bits in the SPI control register 1, software selects one of four combinations of serial clock phase and polarity. The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format. The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

4.24.4.3.2 CPHA = 0 transfer format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave’s data is available at the slave’s data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

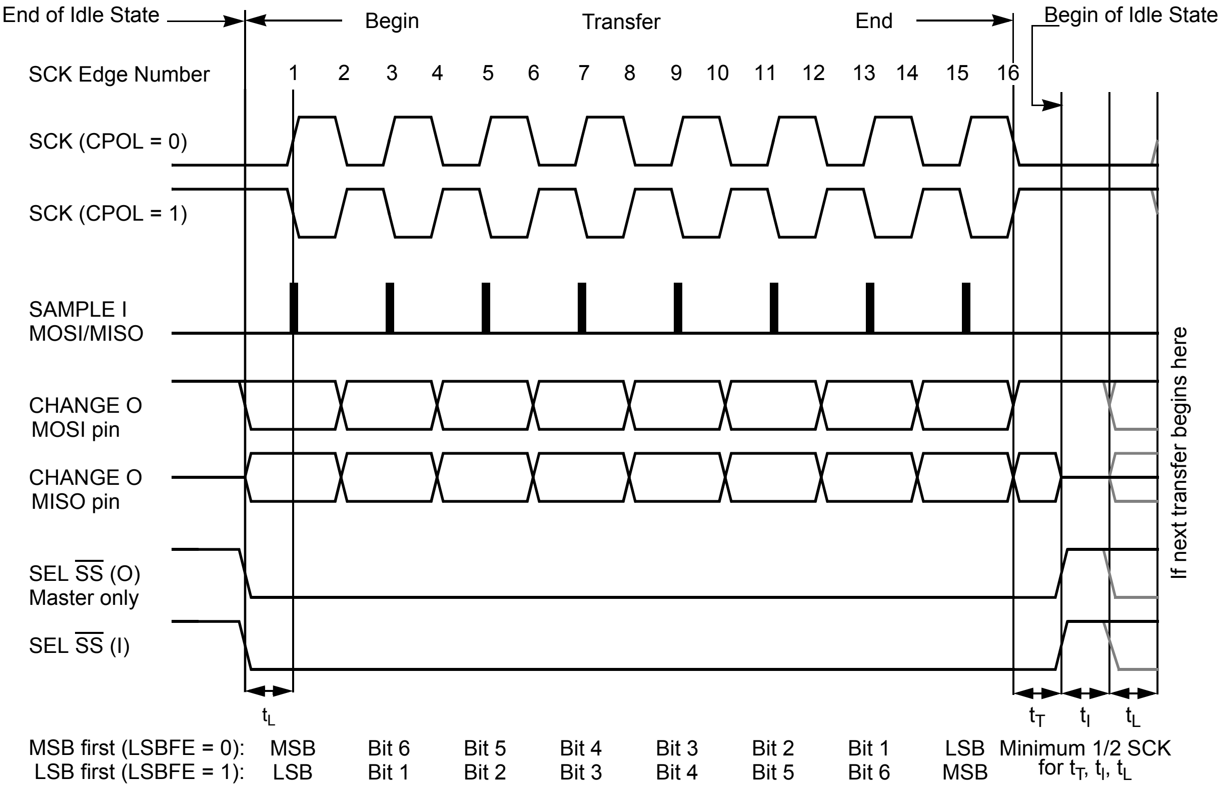
After 2n[281] (last) SCK edges:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating that the transfer is complete.

Notes

281. n depends on the selected transfer width, refer to Section 4.24.3.2.2, “SPI control register 2 (SPICR2)"

Figure 105 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general purpose output not affecting the SPI.



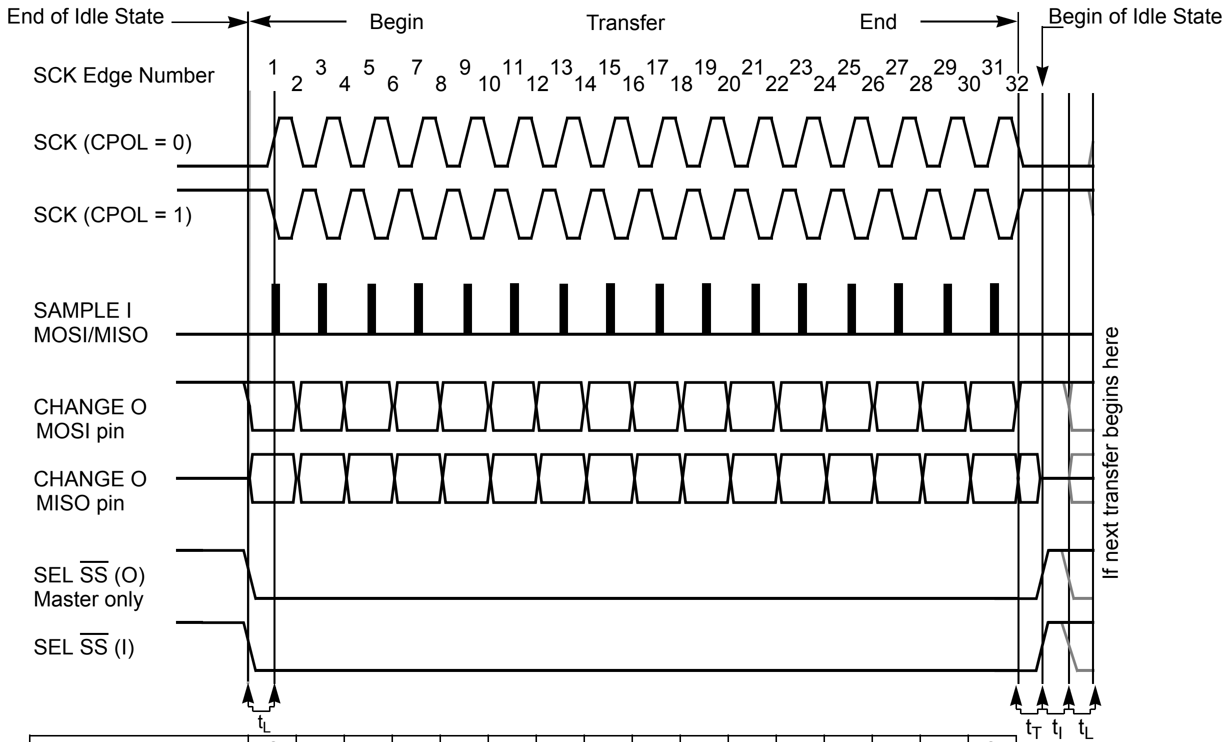
t_L = Minimum leading time before the first SCK edge

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)

t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 105. SPI Clock format 0 (CPHA = 0), with 8-bit transfer width selected (XFRW = 0)



MSB first (LSBFE = 0)	MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	LSB
LSB first (LSBFE = 1)	LSB	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Bit 9	Bit 10	Bit 11	Bit 12	Bit 13	Bit 14	MSB

Minimum 1/2 SCK
for t_T, t_I, t_L

t_L = Minimum leading time before the first SCK edge
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)
t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 106. SPI clock format 0 (CPHA = 0), with 16-Bit transfer width selected (XFRW = 1)

In slave mode, if the $\overline{SS}$ line is not deasserted between the successive transmissions, then the content of the SPI data register is not transmitted; instead the last received data is transmitted. If the $\overline{SS}$ line is deasserted for at least minimum idle time (half SCK cycle) between successive transmissions, then the content of the SPI data register is transmitted.

In master mode, with slave select output enabled, the $\overline{SS}$ line is always deasserted and reasserted between successive transfers for at least minimum idle time.

4.24.4.3.3 CPHA = 1 transfer format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the n[282]-cycle transfer operation.

Notes
[282] n depends on the selected transfer width, refer to Section 4.24.3.2.2, “SPI control register 2 (SPICR2)"

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

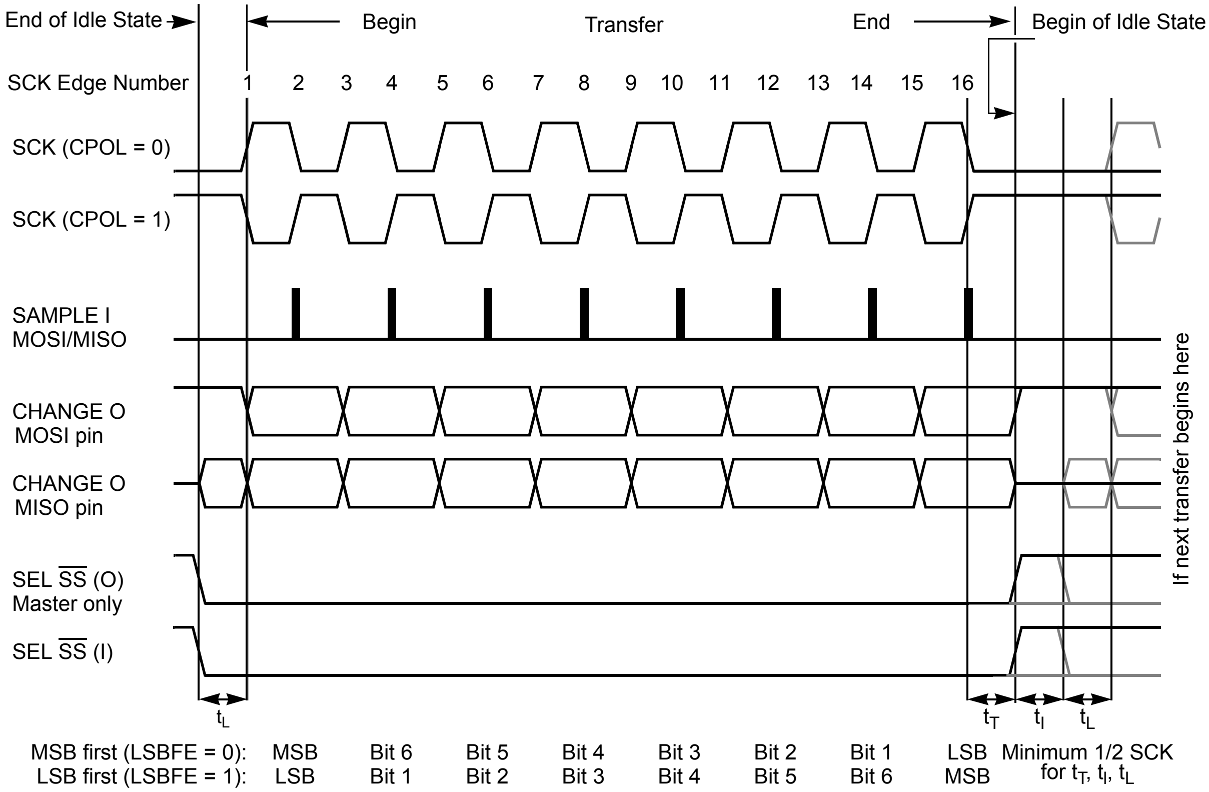
This process continues for a total of n^4 edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After 2n[4] SCK edges:

- Data that was previously in the SPI data register of the master is now in the data register of the slave, and data that was in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating that the transfer is complete.

Figure 107 shows two clocking variations for CPHA = 1. The diagram may be interpreted as a master or slave timing diagram, because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general purpose output not affecting the SPI.

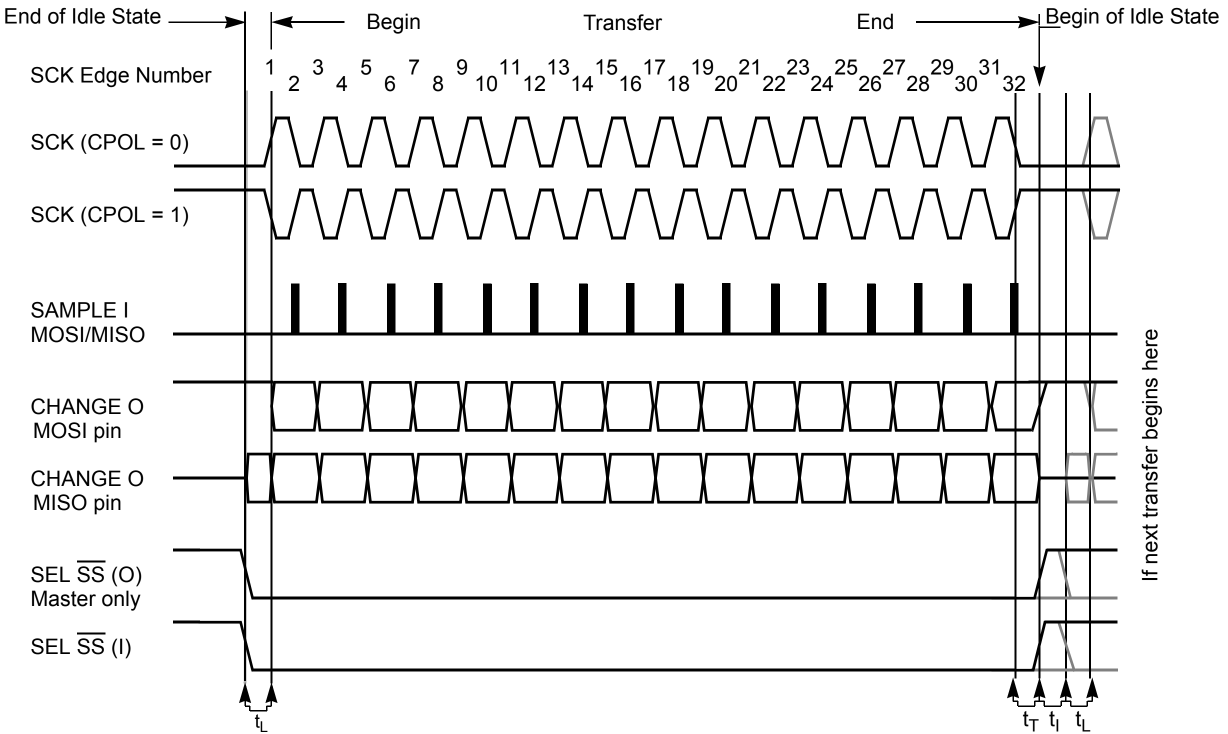


t_L = Minimum leading time before the first SCK edge, not required for back-to-back transfers

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

Figure 107. SPI clock format 1 (CPHA = 1), with 8-Bit transfer width selected (XFRW = 0)



MSB first (LSBFE = 0)	MSB	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	LSB
LSB first (LSBFE = 1)	LSB	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	Bit 9	Bit 10	Bit 11	Bit 12	Bit 13	Bit 14	MSB

Minimum 1/2 SCK for t_T, t_I, t_L

t_L = Minimum leading time before the first SCK edge, not required for back-to-back transfers

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

Figure 108. SPI clock format 1 (CPHA = 1), with 16-Bit transfer width selected (XFRW = 1)

The $\overline{SS}$ line can remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave that drive the MISO data line.

- Back-to-back transfers in master mode

  In master mode, if a transmission has completed and new data is available in the SPI data register, this data is sent out immediately without a trailing and minimum idle time.

The SPI interrupt request flag (SPIF) is common to both the master and slave modes. SPIF gets set one half SCK cycle after the last SCK edge.

4.24.4.4 SPI baud rate generation

Baud rate generation consists of a series of divider stages. Six bits in the SPI baud rate register (SPPR2, SPPR1, SPPR0, SPR2, SPR1, and SPR0) determine the divisor to the SPI module clock which results in the SPI baud rate.

The SPI clock rate is determined by the product of the value in the baud rate preselection bits (SPPR2–SPPR0) and the value in the baud rate selection bits (SPR2–SPR0). The module clock divisor equation is shown in Equation 6.

$$\mathbf{BaudRateDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}} \qquad \textit{Eqn. 6}$$

When all bits are clear (the default condition), the SPI module clock is divided by 2. When the selection bits (SPR2–SPR0) are 001 and the preselection bits (SPPR2–SPPR0) are 000, the module clock divisor becomes 4. When the selection bits are 010, the module clock divisor becomes 8, etc.

When the preselection bits are 001, the divisor determined by the selection bits is multiplied by 2. When the preselection bits are 010, the divisor is multiplied by 3, etc. See Table 453 for baud rate calculations for all bit conditions, based on a 25 MHz bus clock. The two sets of selects allows the clock to be divided by a non-power of two to achieve other baud rates such as divide by 6, divide by 10, etc.

The baud rate generator is activated only when the SPI is in master mode and a serial transfer is taking place. In the other cases, the divider is disabled to decrease I_{DD} current.

NOTE

For maximum allowed baud rates, refer to Section 3.6.2.5, “SPI timing" of this data sheet.

4.24.4.5 Special features

4.24.4.5.1 $\overline{\text{SS}}$ output

The $\overline{\text{SS}}$ output feature automatically drives the $\overline{\text{SS}}$ pin low during transmission, to select external devices and drives it high during idle to deselect external devices. When $\overline{\text{SS}}$ output is selected, the $\overline{\text{SS}}$ output pin is connected to the $\overline{\text{SS}}$ input pin of the external device.

The $\overline{\text{SS}}$ output is available only in master mode during normal SPI operation by asserting SSOE and MODFEN bit as shown in Table 447.

The mode fault feature is disabled while $\overline{\text{SS}}$ output is enabled.

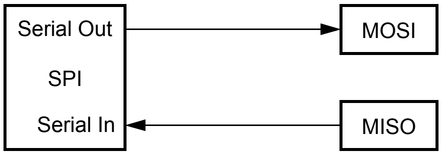
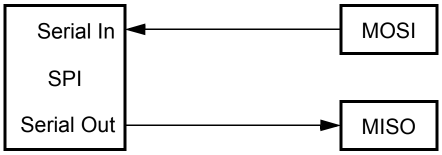
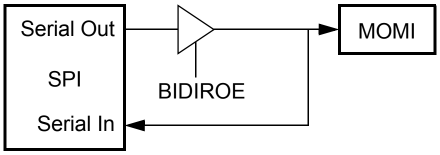
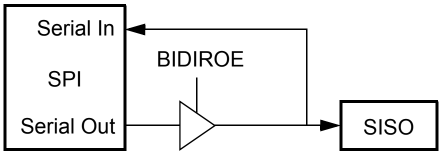
NOTE

Care must be taken when using the $\overline{\text{SS}}$ output feature in a multimaster system because the mode fault feature is not available for detecting system errors between masters.

4.24.4.5.2 Bidirectional mode (MOMI or SISO)

The bidirectional mode is selected when the SPC0 bit is set in SPI control register 2 (see Table 460). In this mode, the SPI uses only one serial data pin for the interface with external device(s). The MSTR bit decides which pin to use. The MOSI pin becomes the serial data I/O (MOMI) pin for the master mode, and the MISO pin becomes serial data I/O (SISO) pin for the slave mode. The MISO pin in master mode and MOSI pin in slave mode are not used by the SPI.

Table 460. Normal mode and bidirectional mode

When SPE = 1	Master mode MSTR = 1	Slave mode MSTR = 0
Normal mode **SPC0 = 0**		
Bidirectional mode **SPC0 = 1**		

The direction of each serial I/O pin depends on the BIDIROE bit. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

- The SCK is output for the master mode and input for the slave mode.
- The $\overline{\text{SS}}$ is the input or output for the master mode, and it is always the input for the slave mode.
- The bidirectional mode does not affect SCK and $\overline{\text{SS}}$ functions.

NOTE

In bidirectional master mode, with mode fault enabled, both data pins MISO and MOSI can be occupied by the SPI, though MOSI is normally used for transmissions in bidirectional mode and MISO is not used by the SPI. If a mode fault occurs, the SPI is automatically switched to slave mode. In this case, MISO becomes occupied by the SPI and MOSI is not used. This must be considered, if the MISO pin is used for another purpose.

4.24.4.6 Error conditions

The SPI has one error condition: Mode fault error

4.24.4.6.1 Mode fault error

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation, the MODF bit in the SPI status register is set automatically, provided the MODFEN bit is set.

In the special case where the SPI is in master mode and MODFEN bit is cleared, the $\overline{SS}$ pin is not used by the SPI. In this special case, the mode fault error function is inhibited and MODF remains cleared. In case the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in slave mode.

If a mode fault error occurs, the SPI is switched to slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO, and MOSI pins are forced to be high-impedance inputs to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to SPI control register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

NOTE

If a mode fault error occurs and a received data byte is pending in the receive shift register, this data byte will be lost.

4.24.4.7 Low power mode options

4.24.4.7.1 SPI in run mode

In run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

4.24.4.7.2 SPI in wait mode

SPI operation in wait mode depends upon the state of the SPISWAI bit in SPI control register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in wait mode.
  - — If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at wait mode entry. The transmission and reception resumes when the SPI exits wait mode.
  - — If SPISWAI is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SCK continues to be driven from the master. This keeps the slave synchronized to the master and the SCK.
    If the master transmits several bytes while the slave is in wait mode, the slave will continue to send out bytes consistent with the operation mode at the start of wait mode (i.e., if the slave is currently sending its SPIDR to the master, it will continue to send the same byte. Else if the slave is currently sending the last received byte from the master, it will continue to send each previous master byte).

NOTE

Care must be taken when expecting data from a master while the slave is in wait or stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e., a SPIF interrupt will **not** be generated until exiting stop or wait mode). Also, the byte from the shift register will not be copied into the SPIDR register until after the slave SPI has exited wait or stop mode. In slave mode, a received byte pending in the receive shift register will be lost when entering wait or stop mode. An SPIF flag and SPIDR copy is generated only if wait mode is entered or exited during a transmission. If the slave enters wait mode in idle mode and exits wait mode in idle mode, neither a SPIF nor a SPIDR copy will occur.

4.24.4.7.3 SPI in stop mode

Stop mode is dependent on the system. The SPI enters stop mode when the module clock is disabled (held high or low). If the SPI is in master mode and exchanging data when the CPU enters stop mode, the transmission is frozen until the CPU exits stop mode. After stop, data to and from the external SPI is exchanged correctly. In slave mode, the SPI will stay synchronized with the master.

The stop mode is not dependent on the SPISWAI bit.

4.24.4.7.4 Reset

The reset values of registers and signals are described in Section 4.18.3, “Memory map and register definition", which details the registers and their bit fields.

- If a data transmission occurs in slave mode after reset without a write to SPIDR, it will transmit garbage, or the data last received from the master before the reset.
- Reading from the SPIDR after reset will always read zeros.

4.24.4.7.5 Interrupts

The 9S12I128PIMV1 only originates interrupt requests when the SPI is enabled (SPE bit in SPICR1 set). The following is a description of how the 9S12I128PIMV1 makes a request and how the MCU should acknowledge that request. The interrupt vector offset and interrupt priority are chip dependent.

The interrupt flags MODF, SPIF, and SPTEF are logically ORed to generate an interrupt request.

4.24.4.7.5.1 MODF

MODF occurs when the master detects an error on the $\overline{\text{SS}}$ pin. The master SPI must be configured for the MODF feature (see Table 447). After MODF is set, the current transfer is aborted and the following bit is changed: MSTR = 0, The master bit in SPICR1 resets.

The MODF interrupt is reflected in the status register MODF flag. Clearing the flag will also clear the interrupt. This interrupt will stay active while the MODF flag is set. MODF has an automatic clearing process which is described in Section 4.24.3.2.4, “SPI status register (SPISR)".

4.24.4.7.5.2 SPIF

SPIF occurs when new data has been received and copied to the SPI data register. After SPIF is set, it does not clear until it is serviced. SPIF has an automatic clearing process, which is described in Section 4.24.3.2.4, “SPI status register (SPISR)".

4.24.4.7.5.3 SPTEF

SPTEF occurs when the SPI data register is ready to accept new data. After SPTEF is set, it does not clear until it is serviced. SPTEF has an automatic clearing process, which is described in Section 4.24.3.2.4, “SPI status register (SPISR)".

4.25 128 kByte flash module (S12FTMRC128K1V1)

4.25.1 Introduction

The FTMRC128K1 module implements the following:

- 128 kbytes of P-Flash (Program Flash) memory
- 4.0 kbytes of D-Flash (Data Flash) memory

The Flash memory is ideal for single-supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller that executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

The Flash memory may be read as bytes, aligned words, or misaligned words. Read access time is one bus cycle for bytes and aligned words, and two bus cycles for misaligned words. For Flash memory, an erased bit reads 1 and a programmed bit reads 0.

It is possible to read from P-Flash memory while some commands are executing on D-Flash memory. It is not possible to read from D-Flash memory while a command is executing on P-Flash memory. Simultaneous P-Flash and D-Flash operations are discussed in Section 4.25.4.4, “Allowed simultaneous P-flash and D-flash operations".

Both P-Flash and D-Flash memories are implemented with Error Correction Codes (ECC) that can resolve single bit faults and detect double bit faults. For P-Flash memory, the ECC implementation requires that programming be done on an aligned 8-byte basis (a Flash phrase). Since P-Flash memory is always read by half-phrase, only one single bit fault in an aligned 4-byte half-phrase containing the byte or word accessed will be corrected.

4.25.1.1 Glossary

Command Write Sequence — An MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.

D-Flash Memory — The D-Flash memory constitutes the nonvolatile memory store for data.

D-Flash Sector — The D-Flash sector is the smallest portion of the D-Flash memory that can be erased. The D-Flash sector consists of four 64-byte rows for a total of 256-bytes.

NVM Command Mode — An NVM mode using the CPU to setup the FCCOB register to pass parameters required for Flash command execution.

Phrase — An aligned group of four 16-bit words within the P-Flash memory. Each phrase includes two sets of aligned double words with each set, including 7 ECC bits for single bit fault correction and double bit fault detection within each double word.

P-Flash Memory — The P-Flash memory constitutes the main nonvolatile memory store for applications.

P-Flash Sector — The P-Flash sector is the smallest portion of the P-Flash memory that can be erased. Each P-Flash sector contains 512-bytes.

Program IFR — Nonvolatile information register located in the P-Flash block that contains the Device ID, Version ID, and the Program Once field.

4.25.1.2 Features

4.25.1.2.1 P-flash features

- 128 kbytes of P-Flash memory composed of one 128 kbyte Flash block divided into 256 sectors of 512-bytes
- Single bit fault correction and double bit fault detection within a 32-bit double word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Ability to read the P-Flash memory while programming a word in the D-Flash memory
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

4.25.1.2.2 D-flash features

- 4.0 kbytes of D-Flash memory composed of one 4.0 kbyte Flash block divided into 16 sectors of 256-bytes
- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Protection scheme to prevent accidental program or erase of D-Flash memory
- Ability to program up to four words in a burst sequence

4.25.1.2.3 Other flash module features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

4.25.1.3 Block diagram

The block diagram of the Flash module is shown in Figure 109.

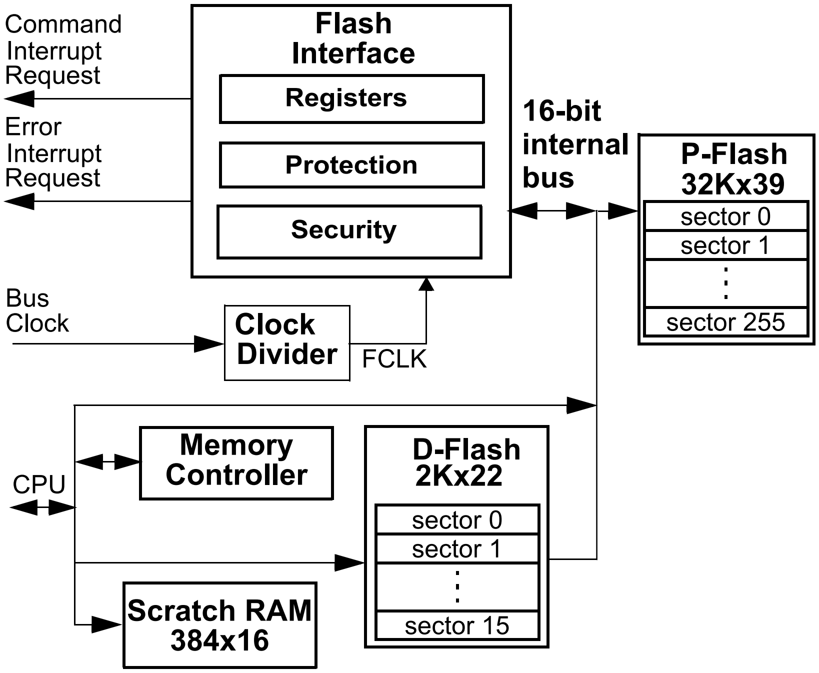


Figure 109. FTMRC128K1 block diagram

4.25.2 External signal description

The Flash module contains no signals that connect off-chip.

4.25.3 Memory map and registers

This section describes the memory map and registers for the Flash module. Read data from unimplemented memory space in the Flash module is undefined. Write access to unimplemented or reserved memory space in the Flash module will be ignored by the Flash module.

4.25.3.1 Module memory map

The S12 architecture places the P-Flash memory between global addresses 0x2_0000 and 0x3_FFFF, as shown in Table 461.The P-Flash memory map is shown in Figure 110.

Table 461. P-flash memory addressing

Global address	Size (bytes)	Description
0x2_0000 – 0x3_FFFF	128 k	P-Flash Block Contains Flash Configuration Field (see Table 462)

The FPROT register, described in Section 4.25.3.2.9, “P-flash protection register (FPROT)", can be set to protect regions in the Flash memory from accidental program or erase. Three separate memory regions, one growing upward from global address 0x3_8000 in the Flash memory (called the lower region), one growing downward from global address 0x3_FFFF in the Flash memory (called the higher region), and the remaining addresses in the Flash memory, can be activated for protection. The Flash memory addresses covered by these protectable regions are shown in the P-Flash memory map. The higher address region is mainly targeted to hold the boot loader code since it covers the vector space. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 462.

Table 462. Flash configuration field

Global address	Size (bytes)	Description
0x3_FF00-0x3_FF07	8	Backdoor Comparison Key Refer to Section 4.25.4.5.11, “Verify backdoor access key command", and Section 4.25.5.1, “Unsecuring the MCU using backdoor key access".
0x3_FF08-0x3_FF0B(283)	4	Reserved
0x3_FF0C(283)	1	P-Flash Protection byte. Refer to Section 4.25.3.2.9, “P-flash protection register (FPROT)".
0x3_FF0D(283)	1	D-Flash Protection byte. Refer to Section 4.25.3.2.10, “D-flash protection register (DFPROT)".
0x3_FF0E(283)	1	Flash Nonvolatile byte Refer to Section 4.25.3.2.16, “Flash option register (FOPT)".
0x3_FF0F(283)	1	Flash Security byte Refer to Section 4.25.3.2.2, “Flash security register (FSEC)".

Notes

283. 0x3FF08-0x3_FF0F form a Flash phrase and must be programmed in a single command write sequence. Each byte in the 0x3_FF08 - 0x3_FF0B reserved field should be programmed to 0xFF.

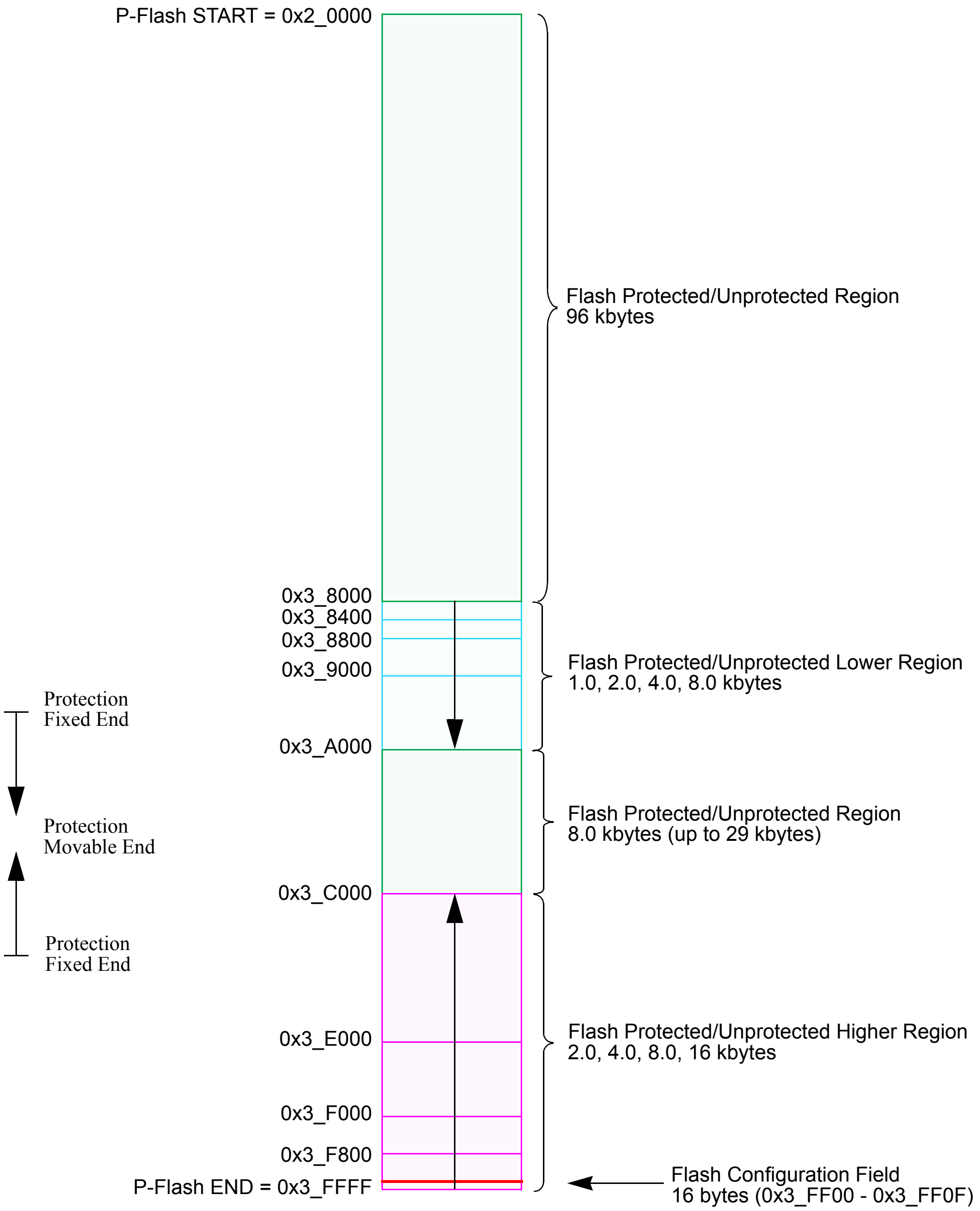

P-Flash START = 0x2_0000
Flash Protected/Unprotected Region
96 kbytes
0x3_8000
0x3_8400
0x3_8800
0x3_9000
Flash Protected/Unprotected Lower Region
1.0, 2.0, 4.0, 8.0 kbytes
Protection
Fixed End
0x3_A000
Flash Protected/Unprotected Region
8.0 kbytes (up to 29 kbytes)
Protection
Movable End
0x3_C000
Protection
Fixed End
0x3_E000
Flash Protected/Unprotected Higher Region
2.0, 4.0, 8.0, 16 kbytes
0x3_F000
0x3_F800
P-Flash END = 0x3_FFFF
Flash Configuration Field
16 bytes (0x3_FF00 - 0x3_FF0F)

Figure 110. P-flash memory map

Table 463. Program IFR fields

Global Address	Size (Bytes)	Field description
0x01_8000 - 0x01_8007	8	Reserved
0x01_8008 - 0x01_80B5	174	Reserved
0x01_80B6 - 0x01_80B7	2	Version ID[284]
0x01_80B8 - 0x01_80BF	8	Reserved
0x01_80C0 - 0x01_80FF	64	Program Once Field. Refer to Section 4.25.4.5.6, “Program once command".

Notes

[284] Used to track firmware patch versions, see Section 4.25.4.2, “IFR version ID word".

4.25.3.2 Register descriptions

The Flash module contains a set of 20 control and status registers located between 0x0100 and 0x0113. A summary of the Flash module registers is given in Table 464 with detailed descriptions in the following subsections.

CAUTION

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and adversely affect Memory Controller behavior.

Table 464. FTMRC128K1 register summary

Address & Name		7	6	5	4	3	2	1	0
0x0100 FCLKDIV	R	FDIVLD	FDIVLCK	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0101 FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
	W								
0x0102 FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
	W								
0x0103 FRSV0	R	0	0	0	0	0	0	0	0
	W								
0x0104 FCNFG	R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
	W								
0x0105 FERCNFG	R	0	0	0	0	0	0	DFDIE	SFDIE
	W								
0x0106 FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
	W								
0x0107 FERSTAT	R	0	0	0	0	0	0	DFDIF	SFDIF
	W								
0x0108 FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0109 DFPROT	R	DPOPEN	0	0	0	DPS3	DPS2	DPS1	DPS0
	W								
0x010A FCCOBHI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x010B FCCOBLO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x010C FRSV1	R	0	0	0	0	0	0	0	0
	W								
0x010D FRSV2	R	0	0	0	0	0	0	0	0
	W								
0x010E FRSV3	R	0	0	0	0	0	0	0	0
	W								

Table 464. FTMRC128K1 register summary (continued)

Address & Name		7	6	5	4	3	2	1	0
0x010F FRSV4	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0110 FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0111 FRSV5	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x0112 FRSV6	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨
0x01103 FRSV7	R	0	0	0	0	0	0	0	0
	W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

4.25.3.2.1 Flash clock divider register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Table 465. Flash clock divider register (FCLKDIV)

Address: 0x0100

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIVLCK	FDIV[5:0]					
W	▨							
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-hi and controls the writability of the FDIV field.

CAUTION

The FCLKDIV register must never be written to while a Flash command is executing (CCIF=0). The FCLKDIV register is writable during the Flash reset sequence even though CCIF is clear.

Table 466. FCLKDIV field descriptions

Field	Description
7 FDIVLD	Clock Divider Loaded 0 FCLKDIV register has not been written since the last reset 1 FCLKDIV register has been written since the last reset
6 FDIVLCK	Clock Divider Locked 0 FDIV field is open for writing 1 FDIV value is locked and cannot be changed. Once the lock bit is set high, only reset can clear this bit and restore writability to the FDIV field.
5–0 FDIV[5:0]	**Clock Divider Bits** — FDIV[5:0] must be set to effectively divide BUSCLK down to 1.0 MHz to control timed events during Flash program and erase algorithms. Table 467 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Refer to Section 4.25.4.3, “Flash command operations", for more information.

Table 467. FDIV values for various BUSCLK frequencies

BUSCLK frequency (MHz)		FDIV[5:0]	BUSCLK frequency (MHz)		FDIV[5:0]
MIN. [285]	MAX. [286]		MIN. [285]	MAX. [286]	
1.0	1.6	0x00	16.6	17.6	0x10
1.6	2.6	0x01	17.6	18.6	0x11
2.6	3.6	0x02	18.6	19.6	0x12
3.6	4.6	0x03	19.6	20.6	0x13
4.6	5.6	0x04	20.6	21.6	0x14
5.6	6.6	0x05	21.6	22.6	0x15
6.6	7.6	0x06	22.6	23.6	0x16
7.6	8.6	0x07	23.6	24.6	0x17
8.6	9.6	0x08	24.6	25.6	0x18
9.6	10.6	0x09	25.6	26.6	0x19
10.6	11.6	0x0A	26.6	27.6	0x1A
11.6	12.6	0x0B	27.6	28.6	0x1B
12.6	13.6	0x0C	28.6	29.6	0x1C
13.6	14.6	0x0D	29.6	30.6	0x1D
14.6	15.6	0x0E	30.6	31.6	0x1E
15.6	16.6	0x0F	31.6	32.6	0x1F

Notes

[285] BUSCLK is Greater Than this value.

[286] BUSCLK is Less Than or Equal to this value.

4.25.3.2.2 Flash security register (FSEC)

The FSEC register holds all bits associated with the security of the MCU and Flash module.

Table 468. Flash security register (FSEC)

Address: 0x0101

	7	6	5	4	3	2	1	0
R	KEYEN[1:0]		RNV[5:2]				SEC[1:0]	
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	F	F	F	F	F	F	F	F

▨ = Unimplemented or Reserved

All bits in the FSEC register are readable, but not writable.

During the reset sequence, the FSEC register is loaded with the contents of the Flash security byte in the Flash configuration field at global address 0x3_FF0F located in P-Flash memory (see Table 462), as indicated by reset condition F in Figure 468. If a double bit fault is detected, while reading the P-Flash phrase containing the Flash security byte during the reset sequence, all bits in the FSEC register will be set to leave the Flash module in a secured state with backdoor key access disabled.

Table 469. FSEC field descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 470.
5–2 RNV[5:2}	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 471. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 470. Flash KEYEN states

KEYEN[1:0]	Status of backdoor key access
00	DISABLED
01	DISABLED(287)
10	ENABLED
11	DISABLED

Notes

287. Preferred KEYEN state to disable backdoor key access.

Table 471. Flash Security States

SEC[1:0]	Status of security
00	SECURED
01	SECURED(288)
10	UNSECURED
11	SECURED

Notes

288. Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Section 4.25.5, “Security".

4.25.3.2.3 Flash CCOB index register (FCCOBIX)

The FCCOBIX register is used to index the FCCOB register for Flash memory operations.

Table 472. FCCOB index register (FCCOBIX)

Address: 0x0102

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCOBIX[2:0]		
W	▨	▨	▨	▨	▨			
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 473. FCCOBIX field descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to select to which word of the FCCOB register array is being read or written. See Section 4.25.3.2.11, “Flash common command object register (FCCOB)" for more details.

4.25.3.2.4 Flash reserved0 register (FRSV0)

This Flash register is reserved for factory testing.

Table 474. Flash reserved0 register (FRSV0)

Address: 0x0103

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV0 register read 0 and are not writable.

4.25.3.2.5 Flash configuration register (FCNFG)

The FCNFG register enables the Flash command complete interrupt and forces ECC faults on Flash array read access from the CPU.

Table 475. Flash configuration register (FCNFG)

Address: 0x0104

	7	6	5	4	3	2	1	0
R	CCIE	0	0	IGNSF	0	0	FDFD	FSFD
W		▨	▨		▨	▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable while remaining bits read 0 and are not writable.

Table 476. FCNFG field descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Section 4.25.3.2.7, “Flash status register (FSTAT)")
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Section 4.25.3.2.8, “Flash error status register (FERSTAT)"). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated

Table 476. FCNFG field descriptions (continued)

Field	Description
1 FDFD	**Force Double Bit Fault Detect** — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations, and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. The FECCR registers will not be updated during the Flash array read operation with FDFD set unless an actual double bit fault is detected. 0 Flash array read operations will set the DFDIF flag in the FERSTAT register only if a double bit fault is detected 1 Any Flash array read operation will force the DFDIF flag in the FERSTAT register to be set (see Section 4.25.3.2.7, “Flash status register (FSTAT)") and an interrupt will be generated, as long as the DFDIE interrupt enable in the FERCNFG register is set (see Section 4.25.3.2.6, “Flash error configuration register (FERCNFG)")
0 FSFD	**Force Single Bit Fault Detect** — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations, and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. The FECCR registers will not be updated during the Flash array read operation with FSFD set unless an actual single bit fault is detected. 0 Flash array read operations will set the SFDIF flag in the FERSTAT register only if a single bit fault is detected 1 Flash array read operation will force the SFDIF flag in the FERSTAT register to be set (see Section 4.25.3.2.7, “Flash status register (FSTAT)"), and an interrupt will be generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Section 4.25.3.2.6, “Flash error configuration register (FERCNFG)")

4.25.3.2.6 Flash error configuration register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

Table 477. Flash error configuration register (FERCNFG)

Address: 0x0105

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	DFDIE	SFDIE
W	▨	▨	▨	▨	▨	▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All assigned bits in the FERCNFG register are readable and writable.

Table 478. FERCNFG field descriptions

Field	Description
1 DFDIE	**Double Bit Fault Detect Interrupt Enable** — The DFDIE bit controls interrupt generation when a double bit fault is detected during a Flash block read operation. 0 DFDIF interrupt disabled 1 An interrupt will be requested whenever the DFDIF flag is set (see Section 4.25.3.2.8, “Flash error status register (FERSTAT)")
0 SFDIE	**Single Bit Fault Detect Interrupt Enable** — The SFDIE bit controls interrupt generation when a single bit fault is detected during a Flash block read operation. 0 SFDIF interrupt disabled whenever the SFDIF flag is set (see Section 4.25.3.2.8, “Flash error status register (FERSTAT)") 1 An interrupt will be requested whenever the SFDIF flag is set (see Section 4.25.3.2.8, “Flash error status register (FERSTAT)")

4.25.3.2.7 Flash status register (FSTAT)

The FSTAT register reports the operational status of the Flash module.

Table 479. Flash status register (FSTAT)

Address: 0x0106

	7	6	5	4	3	2	1	0
R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT[1:0]	
W		▨			▨	▨	▨	▨
Reset	1	0	0	0	0	0	0[289]	0[289]

▨ = Unimplemented or Reserved

Notes

[289] Reset value can deviate from the value shown if a double bit fault is detected during the reset sequence (see Section 4.25.6, “Initialization").

CCIF, ACCERR, and FPVIOL bits are readable and writable, MGBUSY and MGSTAT bits are readable but not writable, while remaining bits read 0 and are not writable.

Table 480. FSTAT field descriptions

Field	Description
7 CCIF	**Command Complete Interrupt Flag** — The CCIF flag indicates that a Flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command and CCIF will stay low until command completion or command violation. 0 Flash command in progress 1 Flash command has completed
5 ACCERR	**Flash Access Error Flag** — The ACCERR bit indicates an illegal access has occurred to the Flash memory caused by either a violation of the command write sequence (see Section 4.25.4.3.2, “Command write sequence") or issuing an illegal Flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR bit has no effect on ACCERR. 0 No access error detected 1 Access error detected
4 FPVIOL	**Flash Protection Violation Flag** —The FPVIOL bit indicates an attempt was made to program or erase an address in a protected area of P-Flash or D-Flash memory during a command write sequence. The FPVIOL bit is cleared by writing a 1 to FPVIOL. Writing a 0 to the FPVIOL bit has no effect on FPVIOL. While FPVIOL is set, it is not possible to launch a command or start a command write sequence. 0 No protection violation detected 1 Protection violation detected
3 MGBUSY	**Memory Controller Busy Flag** — The MGBUSY flag reflects the active state of the Memory Controller**.** 0 Memory Controller is idle 1 Memory Controller is busy executing a Flash command (CCIF = 0)
2 RSVD	**Reserved Bit** — This bit is reserved and always reads 0**.**
1–0 MGSTAT[1:0]	**Memory Controller Command Completion Status Flag** — One or more MGSTAT flag bits are set if an error is detected during execution of a Flash command or during the Flash reset sequence. See Section 4.25.4.5, “Flash command description", and Section 4.25.6, “Initialization", for details.

4.25.3.2.8 Flash error status register (FERSTAT)

The FERSTAT register reflects the error status of internal Flash operations.

Table 481. Flash error status register (FERSTAT)

Address: 0x0107

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	DFDIF	SFDIF
W	▨	▨	▨	▨	▨	▨		
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All flags in the FERSTAT register are readable and only writable to clear the flag.

Table 482. FERSTAT field descriptions

Field	Description
1 DFDIF	**Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation, or that a Flash array read operation was attempted on a Flash block that was under a Flash command operation.(290) The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF. 0 No double bit fault detected 1 Double bit fault detected or an invalid Flash array read operation attempted
0 SFDIF	**Single Bit Fault Detect Interrupt Flag** — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation, or that a Flash array read operation was attempted on a Flash block that was under a Flash command operation.(290) The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF. 0 No single bit fault detected 1 Single bit fault detected and corrected or an invalid Flash array read operation attempted

Notes

290. The single bit fault and double bit fault flags are mutually exclusive for parity errors (an ECC fault occurrence can be either single fault or double fault but never both). A simultaneous access collision (read attempted while command running) is indicated when both SFDIF and DFDIF flags are high.

4.25.3.2.9 P-flash protection register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

Table 483. Flash protection register (FPROT)

Address: 0x0108

	7	6	5	4	3	2	1	0
R	FPOPEN	RNV6	FPHDIS	FPHS[1:0]		FPLDIS	FPLS[1:0]	
W		▨						
Reset	F	F	F	F	F	F	F	F

▨ = Unimplemented or Reserved

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see Section 4.25.3.2.9.1, “P-flash protection restrictions", and Table 488).

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0x3_FF0C located in P-Flash memory (see Table 462), as indicated by reset condition ‘F’ in Figure 483. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

Table 484. FPROT field descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 485, for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0x3_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 486. The FPHS bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0x3_8000. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 487. The FPLS bits can only be written to while the FPLDIS bit is set.

Table 485. P-flash protection function

FPOPEN	FPHDIS	FPLDIS	Function[291]
1	1	1	No P-Flash Protection
1	1	0	Protected Low Range
1	0	1	Protected High Range
1	0	0	Protected High and Low Ranges
0	1	1	Full P-Flash Memory Protected
0	1	0	Unprotected Low Range
0	0	1	Unprotected High Range
0	0	0	Unprotected High and Low Ranges

Notes

291. For range sizes, refer to Table 486 and Table 487.

Table 486. P-flash protection higher address range

FPHS[1:0]	Global address range	Protected size
00	0x3_F800–0x3_FFFF	2.0 kbytes
01	0x3_F000–0x3_FFFF	4.0 kbytes
10	0x3_E000–0x3_FFFF	8.0 kbytes
11	0x3_C000–0x3_FFFF	16 kbytes

Table 487. P-Flash protection lower address range

FPLS[1:0]	Global address range	Protected size
00	0x3_8000–0x3_83FF	1.0 kbyte
01	0x3_8000–0x3_87FF	2.0 kbytes
10	0x3_8000–0x3_8FFF	4.0 kbytes
11	0x3_8000–0x3_9FFF	8.0 kbytes

All possible P-Flash protection scenarios are shown in Figure 111. Although the protection scheme is loaded from the Flash memory at global address 0x3_FF0C during the reset sequence, it can be changed by the user. The P-Flash protection scheme can be used by applications requiring reprogramming in single chip mode while providing as much protection as possible, if reprogramming is not required.

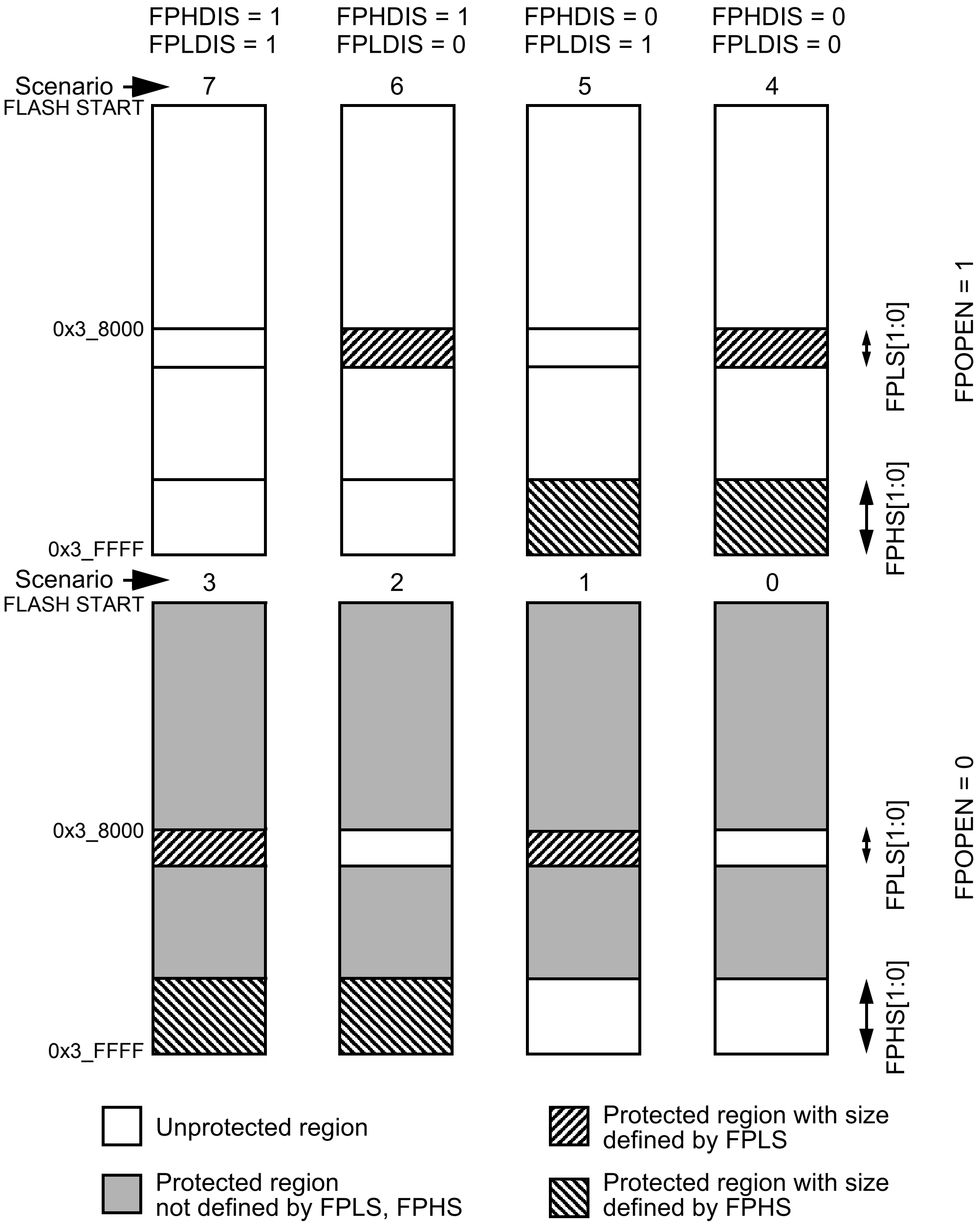


Figure 111. P-flash protection scenarios

4.25.3.2.9.1 P-flash protection restrictions

The general guideline is that P-Flash protection can only be added and not removed. Table 488 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

Table 488. P-flash protection scenario transitions

From protection scenario	To Protection Scenario[292]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	
7	X	X	X	X	X	X	X	X

Notes

[292] Allowed transitions marked with X, see Figure 111 for a definition of the scenarios.

4.25.3.2.10 D-flash protection register (DFPROT)

The DFPROT register defines which D-Flash sectors are protected against program and erase operations.

Table 489. D-flash protection register (DFPROT)

Address: 0x0109

	7	6	5	4	3	2	1	0
R	DPOPEN	0	0	0	DPS[3:0]			
W		▨	▨	▨				
Reset	F	0	0	0	F	F	F	F

▨ = Unimplemented or Reserved

The (unreserved) bits of the DFPROT register are writable with the restriction that protection can be added but not removed. Writes must increase the DPS value and the DPOPEN bit can only be written from a 1 (protection disabled) to a 0 (protection enabled). If the DPOPEN bit is set, the state of the DPS bits is irrelevant.

During the reset sequence, the DFPROT register is loaded with the contents of the D-Flash protection byte in the Flash configuration field at global address 0x3_FF0D located in P-Flash memory (see Table 462) as indicated by reset condition F in Figure 489. To change the D-Flash protection that will be loaded during the reset sequence, the P-Flash sector containing the D-Flash protection byte must be unprotected, then the D-Flash protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the D-Flash protection byte during the reset sequence, the DPOPEN bit will be cleared and DPS bits will be set to leave the D-Flash memory fully protected.

Trying to alter data in any protected area in the D-Flash memory will result in a protection violation error, and the FPVIOL bit will be set in the FSTAT register. Block erase of the D-Flash memory is not possible if any of the D-Flash sectors are protected.

Table 490. DFPROT field descriptions

Field	Description
7 DPOPEN	D-Flash Protection Control 0 Enables D-Flash memory protection from program and erase with protected address range defined by DPS bits 1 Disables D-Flash memory protection from program and erase
3–0 DPS[3:0]	**D-Flash Protection Size** — The DPS[3:0] bits determine the size of the protected area in the D-Flash memory as shown in Table 491.

Table 491. D-flash protection address range

DPS[3:0]	Global address range	Protected size
0000	0x0_4400 – 0x0_44FF	256 bytes
0001	0x0_4400 – 0x0_45FF	512 bytes
0010	0x0_4400 – 0x0_46FF	768 bytes
0011	0x0_4400 – 0x0_47FF	1024 bytes
0100	0x0_4400 – 0x0_48FF	1280 bytes
0101	0x0_4400 – 0x0_49FF	1536 bytes
0110	0x0_4400 – 0x0_4AFF	1792 bytes
0111	0x0_4400 – 0x0_4BFF	2048 bytes
1000	0x0_4400 – 0x0_4CFF	2304 bytes
1001	0x0_4400 – 0x0_4DFF	2560 bytes
1010	0x0_4400 – 0x0_4EFF	2816 bytes
1011	0x0_4400 – 0x0_4FFF	3072 bytes
1100	0x0_4400 – 0x0_50FF	3328 bytes
1101	0x0_4400 – 0x0_51FF	3584 bytes
1110	0x0_4400 – 0x0_52FF	3840 bytes
1111	0x0_4400 – 0x0_53FF	4096 bytes

4.25.3.2.11 Flash common command object register (FCCOB)

The FCCOB is an array of six words addressed via the CCOBIX index found in the FCCOBIX register. Byte wide reads and writes are allowed to the FCCOB register.

Table 492. Flash common command object high register (FCCOBHI)

Address: 0x010A

	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 493. Flash common command object low register (FCCOBLO)

Address: 0x010B

	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

4.25.3.2.11.1 FCCOB - NVM command mode

NVM command mode uses the indexed FCCOB register to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 494. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. Writes to the unimplemented parameter fields (CCOBIX = 110 and CCOBIX = 111) are ignored with reads from these fields returning 0x0000.

Table 494 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Section 4.25.4.5, "Flash command description".

Table 494. FCCOB - NVM command mode (typical usage)

CCOBIX[2:0]	Byte	FCCOB parameter fields (NVM command mode)
000	HI	FCMD[7:0] defining Flash command
	LO	6'h0, Global address [17:16]
001	HI	Global address [15:8]
	LO	Global address [7:0]
010	HI	Data 0 [15:8]
	LO	Data 0 [7:0]
011	HI	Data 1 [15:8]
	LO	Data 1 [7:0]
100	HI	Data 2 [15:8]
	LO	Data 2 [7:0]
101	HI	Data 3 [15:8]
	LO	Data 3 [7:0]

4.25.3.2.12 Flash reserved1 register (FRSV1)

This Flash register is reserved for factory testing.

Table 495. Flash reserved1 register (FRSV1)

Address: 0x010C

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV1 register read 0 and are not writable.

4.25.3.2.13 Flash reserved2 register (FRSV2)

This Flash register is reserved for factory testing.

Table 496. Flash reserved2 register (FRSV2)

Address: 0x010D

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV2 register read 0 and are not writable.

4.25.3.2.14 Flash reserved3 register (FRSV3)

This Flash register is reserved for factory testing.

Table 497. Flash reserved3 register (FRSV3)

Address: 0x010E

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV3 register read 0 and are not writable.

4.25.3.2.15 Flash reserved4 register (FRSV4)

This Flash register is reserved for factory testing.

Table 498. Flash reserved4 register (FRSV4)

Address: 0x010F

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV4 register read 0 and are not writable.

4.25.3.2.16 Flash option register (FOPT)

The FOPT register is the Flash option register.

Table 499. Flash option register (FOPT)

Address: 0x0110

	7	6	5	4	3	2	1	0
R	NV[7:0]							
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	F	F	F	F	F	F	F	F

▨ = Unimplemented or Reserved

All bits in the FOPT register are readable but are not writable.

During the reset sequence, the FOPT register is loaded from the Flash nonvolatile byte in the Flash configuration field, at global address 0x3_FF0E located in P-Flash memory (see Table 462), as indicated by reset condition F in Figure 499. If a double bit fault is detected while reading the P-Flash phrase containing the Flash nonvolatile byte during the reset sequence, all bits in the FOPT register will be set.

Table 500. FOPT field descriptions

Field	Description
7–0 NV[7:0]	**Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the device user guide for proper use of the NV bits.

4.25.3.2.17 Flash reserved5 register (FRSV5)

This Flash register is reserved for factory testing.

Table 501. Flash reserved5 register (FRSV5)

Address: 0x0111

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV5 register read 0 and are not writable.

4.25.3.2.18 Flash reserved6 register (FRSV6)

This Flash register is reserved for factory testing.

Table 502. Flash reserved6 register (FRSV6)

Address: 0x0112

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV6 register read 0 and are not writable.

4.25.3.2.19 Flash reserved7 register (FRSV7)

This Flash register is reserved for factory testing.

Table 503. Flash reserved7 register (FRSV7)

Address: 0x0113

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

▨ = Unimplemented or Reserved

All bits in the FRSV7 register read 0 and are not writable.

4.25.4 Functional description

4.25.4.1 Modes of operation

The FTMRC128K1 module provides the modes of operation, as shown in Table 504. The operating mode is determined by module-level inputs and affects the FCLKDIV, FCNFG, and DFPROT registers, Scratch RAM writes, and the command set availability (see Table 506).

Table 504. Modes and mode control inputs

Operating mode	FTMRC input
	mmc_mode_ss_t2
Normal:	0
Special:	1

4.25.4.2 IFR version ID word

The version ID word is stored in the IFR at address 0x01_80B6. The contents of the word are defined in Table 505.

Table 505. IFR version ID fields

[15:4]	[3:0]
Reserved	VERNUM

VERNUM: Version number. The first version is number 0b_0001 with both 0b_0000 and 0b_1111 meaning ‘none’.

4.25.4.3 Flash command operations

Flash command operations are used to modify Flash memory contents.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from BUSCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution

4.25.4.3.1 Writing the FCLKDIV register

Prior to issuing any Flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide BUSCLK down to a target FCLK of 1.0 MHz. Table 467 shows recommended values for the FDIV field based on BUSCLK frequency.

NOTE

Programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 0.8 MHz. Setting FDIV too high can destroy the Flash memory due to overstress. Setting FDIV too low can result in incomplete programming or erasure of the Flash memory cells.

When the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written, any Flash program or erase command loaded during a command write sequence will not execute and the ACCERR bit in the FSTAT register will set.

4.25.4.3.2 Command write sequence

The Memory Controller will launch all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see Section 4.25.3.2.7, “Flash status register (FSTAT)") and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active and a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

CAUTION

Writes to any Flash register must be avoided while a Flash command is active (CCIF=0) to prevent corruption of Flash register contents and Memory Controller behavior.

4.25.4.3.2.1 Define FCCOB contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. Access to the FCCOB parameter fields is controlled via the CCOBIX bits in the FCCOBIX register (see Section 4.25.3.2.3, “Flash CCOB index register (FCCOBIX)").

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag will remain clear until the Flash command has completed. Upon completion, the Memory Controller will return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 112.

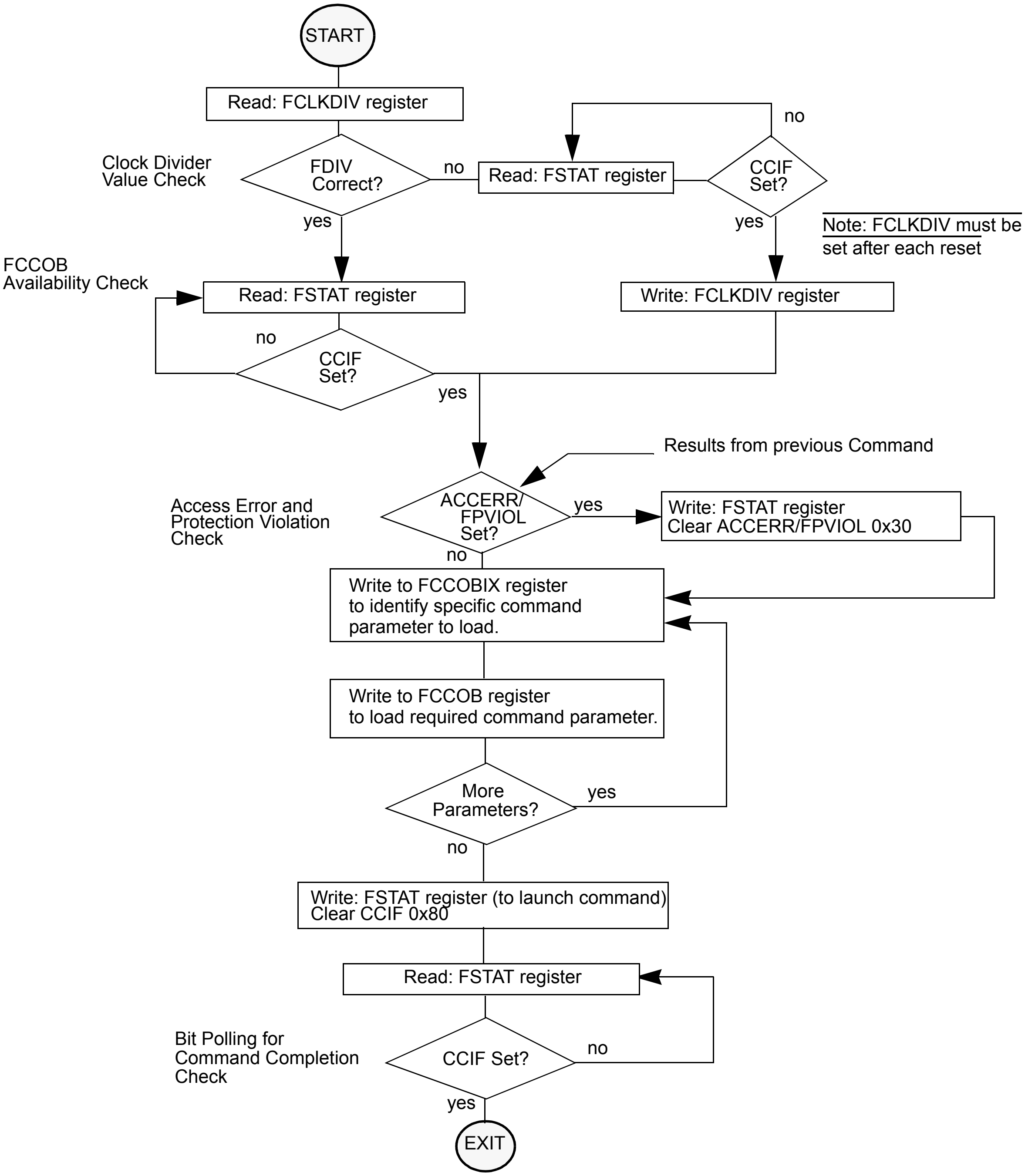


Figure 112. Generic flash command write sequence flowchart

4.25.4.3.3 Valid flash module commands

Table 506. Flash commands by mode

FCMD	Command	Unsecured		Secured	
		NS[293]	SS[294]	NS[295]	SS[296]
0x01	Erase Verify All Blocks	*	*	*	*
0x02	Erase Verify Block	*	*	*	*
0x03	Erase Verify P-Flash Section	*	*	*	
0x04	Read Once	*	*	*	
0x06	Program P-Flash	*	*	*	
0x07	Program Once	*	*	*	
0x08	Erase All Blocks		*		*
0x09	Erase Flash Block	*	*	*	
0x0A	Erase P-Flash Sector	*	*	*	
0x0B	Unsecure Flash		*		*
0x0C	Verify Backdoor Access Key	*		*	
0x0D	Set User Margin Level	*	*	*	
0x0E	Set Field Margin Level		*		
0x10	Erase Verify D-Flash Section	*	*	*	
0x11	Program D-Flash	*	*	*	
0x12	Erase D-Flash Sector	*	*	*	

Notes

[293] Unsecured Normal Single Chip mode.

[294] Unsecured Special Single Chip mode.

[295] Secured Normal Single Chip mode.

[296] Secured Special Single Chip mode.

4.25.4.3.4 P-flash commands

Table 507 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

Table 507. P-flash commands

FCMD	Command	Function on P-flash memory
0x01	Erase Verify All Blocks	Verify that all P-Flash (and D-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that a P-Flash block is erased.
0x03	Erase Verify P-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x04	Read Once	Read a dedicated 64-byte field in the nonvolatile information register in P-Flash block that was previously programmed using the Program Once command.
0x06	Program P-Flash	Program a phrase in a P-Flash block.
0x07	Program Once	Program a dedicated 64-byte field in the nonvolatile information register in P-Flash block that is allowed to be programmed only once.
0x08	Erase All Blocks	Erase all P-Flash (and D-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.

Table 507. P-flash commands (continued)

FCMD	Command	Function on P-flash memory
0x09	Erase Flash Block	Erase a P-Flash (or D-Flash) block. An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command.
0x0A	Erase P-Flash Sector	Erase all bytes in a P-Flash sector.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all P-Flash (and D-Flash) blocks and verifying that all P-Flash (and D-Flash) blocks are erased.
0x0C	Verify Backdoor Access Key	Supports a method of releasing MCU security by verifying a set of security keys.
0x0D	Set User Margin Level	Specifies a user margin read level for all P-Flash blocks.
0x0E	Set Field Margin Level	Specifies a field margin read level for all P-Flash blocks (special modes only).

4.25.4.3.5 D-flash commands

Table 508 summarizes the valid D-Flash commands along with the effects of the commands on the D-Flash block.

Table 508. D-flash commands

FCMD	Command	Function on D-flash memory
0x01	Erase Verify All Blocks	Verify that all D-Flash (and P-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that the D-Flash block is erased.
0x08	Erase All Blocks	Erase all D-Flash (and P-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a D-Flash (or P-Flash) block. An erase of the full D-Flash block is only possible when DPOPEN bit in the DFPROT register is set prior to launching the command.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all D-Flash (and P-Flash) blocks and verifying that all D-Flash (and P-Flash) blocks are erased.
0x0D	Set User Margin Level	Specifies a user margin read level for the D-Flash block.
0x0E	Set Field Margin Level	Specifies a field margin read level for the D-Flash block (special modes only).
0x10	Erase Verify D-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x11	Program D-Flash	Program up to four words in the D-Flash block.
0x12	Erase D-Flash Sector	Erase all bytes in a sector of the D-Flash block.

4.25.4.4 Allowed simultaneous P-flash and D-flash operations

Only the operations marked ‘OK’ in Table 509 are permitted to be run simultaneously on the Program Flash and Data Flash blocks. Some operations cannot be executed simultaneously because certain hardware resources are shared by the two memories. The priority has been placed on permitting Program Flash reads while program and erase operations execute on the Data Flash, providing read (P-Flash) while write (D-Flash) functionality.

Table 509. Allowed P-flash and D-flash simultaneous operations

	Data flash				
Program flash	**Read**	**Margin read** [297]	**Program**	**Sector erase**	**Mass erase** [299]
Read		OK	OK	OK	
Margin read [297]		OK[298]			
Program					
Sector erase				OK	
Mass erase [299]					OK

Notes

[297] A ‘Margin Read’ is any read after executing the margin setting commands ‘Set User Margin Level’, or ‘Set Field Margin Level’ with anything but the ‘normal’ level specified.

[298] See the Note on margin settings in Section 4.25.4.5.12, “Set user margin level command" and Section 4.25.4.5.13, “Set field margin level command".

[299] The ‘Mass Erase’ operations are commands ‘Erase All Blocks’ and ‘Erase Flash Block’.

4.25.4.5 Flash command description

This section provides details of all available Flash commands launched by a command write sequence. The ACCERR bit in the FSTAT register will be set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the Memory Controller:

- Starting any command write sequence that programs or erases Flash memory before initializing the FCLKDIV register
- Writing an invalid command as part of the command write sequence
- For additional possible errors, refer to the error handling table provided for each command

If a Flash block is read during execution of an algorithm (CCIF = 0) on that same block, the read operation will return invalid data. If the SFDIF or DFDIF flags were not previously set when the invalid read operation occurred, both the SFDIF and DFDIF flags will be set.

If the ACCERR or FPVIOL bits are set in the FSTAT register, the user must clear these bits before starting any command write sequence (see Section 4.25.3.2.7, “Flash status register (FSTAT)").

CAUTION

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

4.25.4.5.1 Erase verify all blocks command

The Erase Verify All Blocks command will verify that all P-Flash and D-Flash blocks have been erased.

Table 510. Erase verify all blocks command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x01	Not required

Upon clearing CCIF to launch the Erase Verify All Blocks command, the Memory Controller will verify that the entire Flash memory space is erased. The CCIF flag will set after the Erase Verify All Blocks operation has completed.

Table 511. Erase verify all blocks command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 000 at command launch
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

4.25.4.5.2 Erase verify block command

The Erase Verify Block command allows the user to verify that an entire P-Flash or D-Flash block has been erased. The FCCOB upper global address bits determine which block must be verified.

Table 512. Erase verify block command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x02	Global address [17:16] of the Flash block to be verified.

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or D-Flash block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.

Table 513. Erase verify block command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 000 at command launch
		Set if an invalid global address [17:16] is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

4.25.4.5.3 Erase verify P-flash section command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

Table 514. Erase verify P-flash section command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x03	Global address [17:16] of a P-Flash block
001	Global address [15:0] of the first phrase to be verified	
010	Number of phrases to be verified	

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed.

Table 515. Erase verify P-flash section command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 010 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0]!= 000)
		Set if the requested section crosses a 128 kbyte boundary
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

4.25.4.5.4 Read once command

The Read Once command provides read access to a reserved 64-byte field (8 phrases) located in the nonvolatile information register of P-Flash. The Read Once field is programmed using the Program Once command described in Section 4.25.4.5.6, “Program once command". The Read Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 516. Read once command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x04	Not Required
001	Read Once phrase index (0x0000 - 0x0007)	
010	Read Once word 0 value	
011	Read Once word 1 value	
100	Read Once word 2 value	
101	Read Once word 3 value	

Upon clearing CCIF to launch the Read Once command, a Read Once phrase is fetched and stored in the FCCOB indexed register. The CCIF flag will set after the Read Once operation has completed. Valid phrase index values for the Read Once command range from 0x0000 to 0x0007. During execution of the Read Once command, any attempt to read addresses within P-Flash block will return invalid data.

Table 517. Read once command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid phrase index is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

4.25.4.5.5 Program P-flash command

The Program P-Flash operation will program a previously erased phrase in the P-Flash memory using an embedded algorithm.

CAUTION

A P-Flash phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash phrase is not allowed.

Table 518. Program P-flash command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x06	Global address [17:16] to identify P-Flash block
001	Global address [15:0] of phrase location to be programmed[300]	
010	Word 0 program value	
011	Word 1 program value	
100	Word 2 program value	
101	Word 3 program value	

Notes

[300] Global address [2:0] must be 000

Upon clearing CCIF to launch the Program P-Flash command, the Memory Controller will program the data words to the supplied global address and will then proceed to verify the data words read back as expected. The CCIF flag will set after the Program P-Flash operation has completed.

Table 519. Program P-flash command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 101 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0]!= 000)
	FPVIOL	Set if the global address [17:0] points to a protected area
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

4.25.4.5.6 Program once command

The Program Once command restricts programming to a reserved 64-byte field (8 phrases) in the nonvolatile information register located in P-Flash. The Program Once reserved field can be read using the Read Once command as described in Section 4.25.4.5.4, “Read once command". The Program Once command must only be issued once, since the nonvolatile information register in P-Flash cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 520. Program once command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x07	Not Required
001	Program Once phrase index (0x0000 - 0x0007)	
010	Program Once word 0 value	
011	Program Once word 1 value	
100	Program Once word 2 value	
101	Program Once word 3 value	

Upon clearing CCIF to launch the Program Once command, the Memory Controller first verifies that the selected phrase is erased. If erased, then the selected phrase will be programmed and then verified with read back. The CCIF flag will remain clear, setting only after the Program Once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash will return invalid data.

Table 521. Program once command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 101 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid phrase index is supplied
		Set if the requested phrase has already been programmed[301]
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

Notes

[301] If a Program Once phrase is initially programmed to 0xFFFF_FFFF_FFFF_FFFF, the Program Once command will be allowed to execute again on that same phrase.

4.25.4.5.7 Erase all blocks command

The Erase All Blocks operation will erase the entire P-Flash and D-Flash memory space.

Table 522. Erase all blocks command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x08	Not required

Upon clearing CCIF to launch the Erase All Blocks command, the Memory Controller will erase the entire Flash memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag will set after the Erase All Blocks operation has completed.

Table 523. Erase all blocks command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 000 at command launch
		Set if command not available in current mode (see Table 506)
	FPVIOL	Set if any area of the P-Flash or D-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

4.25.4.5.8 Erase flash block command

The Erase Flash Block operation will erase all addresses in a P-Flash or D-Flash block.

Table 524. Erase flash block command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x09	Global address [17:16] to identify Flash block
001	Global address [15:0] in Flash block to be erased	

Upon clearing CCIF to launch the Erase Flash Block command, the Memory Controller will erase the selected Flash block and verify that it is erased. The CCIF flag will set after the Erase Flash Block operation has completed.

Table 525. Erase flash block command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:16] is supplied
		Set if the supplied P-Flash address is not phrase-aligned or if the D-Flash address is not word-aligned
	FPVIOL	Set if an area of the selected Flash block is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

4.25.4.5.9 Erase P-flash sector command

The Erase P-Flash Sector operation will erase all addresses in a P-Flash sector.

Table 526. Erase P-flash sector command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0A	Global address [17:16] to identify P-Flash block to be erased
001	Global address [15:0] anywhere within the sector to be erased. Refer to Section 4.25.1.2.1, “P-flash features" for the P-Flash sector size.	

Upon clearing CCIF to launch the Erase P-Flash Sector command, the Memory Controller will erase the selected Flash sector and then verify that it is erased. The CCIF flag will be set after the Erase P-Flash Sector operation has completed.

Table 527. Erase P-flash sector command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:16] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0]!= 000)
	FPVIOL	Set if the selected P-Flash sector is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

4.25.4.5.10 Unsecure flash command

The Unsecure Flash command will erase the entire P-Flash and D-Flash memory space and, if the erase is successful, will release security.

Table 528. Unsecure flash command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and D-Flash memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 529. Unsecure flash command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 000 at command launch
		Set if command not available in current mode (see Table 506)
	FPVIOL	Set if any area of the P-Flash or D-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

4.25.4.5.11 Verify backdoor access key command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 470). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see Table 462). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

Table 530. Verify backdoor access key command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x0C	Not required
001	Key 0	
010	Key 1	
011	Key 2	
100	Key 3	

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0x3_FF00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 531. Verify backdoor access key command Error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0]!= 10, see Section 4.25.3.2.2)
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

4.25.4.5.12 Set user margin level command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or D-Flash block.

Table 532. Set user margin level command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0D	Global address [17:16] to identify the Flash block
001	Margin level setting	

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

NOTE

When the D-Flash block is targeted, the D-Flash user margin levels are applied only to the D-Flash reads. However, when the P-Flash block is targeted, the P-Flash user margin levels are applied to both P-Flash and D-Flash reads. It is not possible to apply user margin levels to the P-Flash block only.

Valid margin level settings for the Set User Margin Level command are defined in Table 533.

Table 533. Valid set user margin level settings

CCOB (CCOBIX=001)	Level description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level[302]
0x0002	User Margin-0 Level[303]

Notes

[302] Read margin to the erased state

[303] Read margin to the programmed state

Table 534. Set user margin level command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:16] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

NOTE

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

4.25.4.5.13 Set field margin level command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of the P-Flash or D-Flash block.

Table 535. Set field margin level command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x0E	Global address [17:16] to identify the Flash block
001	Margin level setting	

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the field margin level for the targeted block and then set the CCIF flag.

NOTE

When the D-Flash block is targeted, the D-Flash field margin levels are applied only to the D-Flash reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and D-Flash reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 536.

Table 536. Valid set field margin level settings

CCOB (CCOBIX=001)	Level description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level(304)
0x0002	User Margin-0 Level(305)
0x0003	Field Margin-1 Level(304)
0x0004	Field Margin-0 Level(305)

Notes

304. Read margin to the erased state

305. Read margin to the programmed state

Table 537. Set field margin level command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:16] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

CAUTION

Field margin levels must only be used during verify of the initial factory programming.

NOTE

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

4.25.4.5.14 Erase verify D-flash section command

The Erase Verify D-Flash Section command will verify that a section of code in the D-Flash is erased. The Erase Verify D-Flash Section command defines the starting point of the data to be verified and the number of words.

Table 538. Erase verify D-flash section command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x10	Global address [17:16] to identify the D-Flash block
001	Global address [15:0] of the first word to be verified	
010	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify D-Flash Section command, the Memory Controller will verify the selected section of D-Flash memory is erased. The CCIF flag will set after the Erase Verify D-Flash Section operation has completed.

Table 539. Erase verify D-flash section command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 010 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned word address is supplied (global address [0]!= 0)
		Set if the requested section breaches the end of the D-Flash block
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

4.25.4.5.15 Program D-flash command

The Program D-Flash operation programs one to four previously erased words in the D-Flash block. The Program D-Flash operation will confirm that the targeted location(s) were successfully programmed upon completion.

CAUTION

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

Table 540. Program D-flash command FCCOB requirements

CCOBIX[2:0]	FCCOB parameters	
000	0x11	Global address [17:16] to identify the D-Flash block
001	Global address [15:0] of word to be programmed	
010	Word 0 program value	
011	Word 1 program value, if desired	
100	Word 2 program value, if desired	
101	Word 3 program value, if desired	

Upon clearing CCIF to launch the Program D-Flash command, the user-supplied words will be transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program D-Flash command launch determines how many words will be programmed in the D-Flash block. The CCIF flag is set when the operation has completed.

Table 541. Program D-flash command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0] < 010 at command launch
		Set if CCOBIX[2:0] > 101 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned word address is supplied (global address [0]!= 0)
		Set if the requested group of words breaches the end of the D-Flash block
	FPVIOL	Set if the selected area of the D-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

4.25.4.5.16 Erase D-flash sector command

The Erase D-Flash Sector operation will erase all addresses in a sector of the D-Flash block.

Table 542. Erase D-flash sector command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
000	0x12	Global address [17:16] to identify D-Flash block
001	Global address [15:0] anywhere within the sector to be erased. See Section 4.25.1.2.2, “D-flash features" for D-Flash sector size.	

Upon clearing CCIF to launch the Erase D-Flash Sector command, the Memory Controller will erase the selected Flash sector and verify that it is erased. The CCIF flag will set after the Erase D-Flash Sector operation has completed.

Table 543. Erase D-flash sector command error handling

Register	Error bit	Error condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 506)
		Set if an invalid global address [17:0] is supplied
		Set if a misaligned word address is supplied (global address [0]!= 0)
	FPVIOL	Set if the selected area of the D-Flash memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

4.25.4.6 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed, or when a Flash command operation has detected an ECC fault.

Table 544. Flash interrupt sources

Interrupt source	Interrupt flag	Local enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

NOTE

Vector addresses and their relative interrupt priority are determined at the MCU level.

4.25.4.6.1 Description of flash interrupt operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Section 4.25.3.2.5, “Flash configuration register (FCNFG)", Section 4.25.3.2.6, “Flash error configuration register (FERCNFG)", Section 4.25.3.2.7, “Flash status register (FSTAT)", and Section 4.25.3.2.8, “Flash error status register (FERSTAT)".

The logic used for generating the Flash module interrupts is shown in Figure 113.

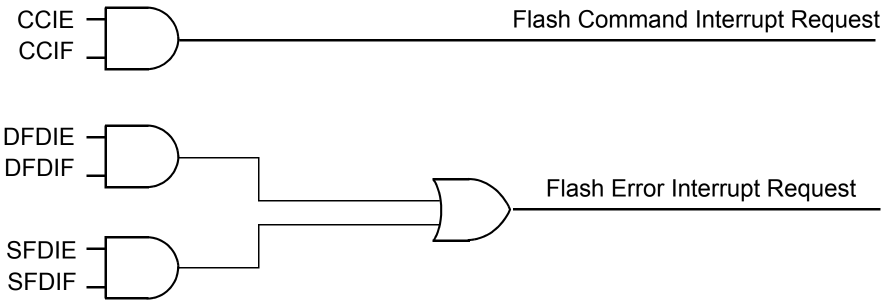

CCIE
CCIF
Flash Command Interrupt Request
DFDIE
DFDIF
Flash Error Interrupt Request
SFDIE
SFDIF

Figure 113. Flash module interrupts implementation

4.25.4.7 Wait mode

The Flash module is not affected if the MCU enters wait mode. The Flash module can recover the MCU from wait via the CCIF interrupt (see Section 4.25.4.6, “Interrupts").

4.25.4.8 Stop mode

If a Flash command is active (CCIF = 0) when the MCU requests stop mode, the current Flash operation will be completed before the CPU is allowed to enter stop mode.

4.25.5 Security

The Flash module provides security information to the MCU. The Flash security state is defined by the SEC bits of the FSEC register (see Table 471). During reset, the Flash module initializes the FSEC register using data read from the security byte of the Flash configuration field at global address 0x3_FF0F. The security state out of reset can be permanently changed by programming the security byte, assuming that the MCU is starting from a mode where the necessary P-Flash erase and program commands are available, and that the upper region of the P-Flash is unprotected. If the Flash security byte is successfully programmed, its new value will take affect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using backdoor key access
- Unsecuring the MCU in special single chip mode using BDM
- Mode and security effects on flash command availability

4.25.5.1 Unsecuring the MCU using backdoor key access

The MCU may be unsecured by using the backdoor key access feature, which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0x3_FF00-0x3_FF07). If the KEYEN[1:0] bits are in the enabled state (see Section 4.25.3.2.2, “Flash security register (FSEC)"), the Verify Backdoor Access Key command (see Section 4.25.4.5.11, “Verify backdoor access key command") allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 471) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash memory, and D-Flash memory will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Section 4.25.3.2.2, “Flash security register (FSEC)"), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Section 4.25.4.5.11, “Verify backdoor access key command"
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command. The security as defined in the Flash security byte (0x3_FF0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0x3_FF00-0x3_FF07 are unaffected by the Verify Backdoor Access Key command sequence. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0x3_FF00-0x3_FF07 in the Flash configuration field.

4.25.5.2 Unsecuring the MCU in special single chip mode using BDM

A secured MCU can be unsecured in special single chip mode by using the following method to erase the P-Flash and D-Flash memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and D-Flash memories are erased
3. Send BDM commands to disable protection in the P-Flash and D-Flash memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and D-Flash memory
5. After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify that the P-Flash and D-Flash memory are erased

If the P-Flash and D-Flash memory are verified as erased, the MCU will be unsecured. All BDM commands will now be enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state
8. Reset the MCU

4.25.5.3 Mode and security effects on flash command availability

The availability of Flash module commands depends on the MCU operating mode and security state as shown in Table 506.

4.25.6 Initialization

On each system reset the Flash module executes a reset sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and DFPROT protection registers, and the FOPT and FSEC registers. The Flash module reverts to using built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register will be set.

CCIF remains clear throughout the reset sequence. The Flash module holds off all CPU access for the initial portion of the reset sequence. While Flash memory reads and access to most Flash registers are possible when the hold is removed, writes to the FCCOBIX, FCCOBHI, and FCCOBLO registers are ignored. Completion of the reset sequence is marked by setting CCIF high which enables writes to the FCCOBIX, FCCOBHI, and FCCOBLO registers to launch any available Flash command.

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

4.26 MCU - die-to-die initiator (9S12I128PIMV1)

4.26.0.1 Acronyms and abbreviations

Table 545 contains sample acronyms and abbreviations used in this document.

Table 545. Acronyms and abbreviated terms

Term	Meaning
D2D	Die-to-Die

4.26.0.1.1 Glossary

Table 312 shows a glossary of the major terms used in this document.

Table 546. Glossary

Term	Definition
Active low	The signal is asserted when it changes to logic-level zero.
Active high	The signal is asserted when it changes to logic-level one.
Asserted	Discrete signal is in active logic state.
Customer	The end user of an SoC design or device.
EOT	End of Transaction
Negated	A discrete signal is in inactive logic state.
Pin	External physical connection.
Revision	Revised or new version of a document. Revisions produce versions; there can be no ‘Rev 0.0.’
Signal	Electronic construct whose state or change in state conveys information.
Transfer	A read or write on the CPU bus following the IP-Bus protocol.
Transaction	Command, address and if required data sent on the D2D interface. A transaction is finished by the EOT acknowledge cycle.
Version	Particular form or variation of an earlier or original document.

4.26.1 Introduction

This section describes the functionality of the die-to-die (9S12I128PIMV1) initiator block especially designed for low cost connections between a microcontroller die (Interface Initiator) and an analog die (Interface Target) located in the same package.

The D2DI block

- realizes the initiator part of the D2D interface, including supervision and error interrupt generation
- generates the clock for this interface
- disables/enables the interrupt from the D2D interface

4.26.1.1 Overview

The D2DI is the initiator for a data transfer to and from a target, typically located on another die in the same package. It provides a set of configuration registers and two memory mapped 256 Byte address windows. When writing to a window a transaction is initiated, sending a write command followed, by an 8-bit address and the data byte or word to the target. When reading from a window, a transaction is initiated, sending a read command, followed by an 8-bit address to the target. The target then responds with the data. The basic idea is that a peripheral located on another die, can be addressed like an on-chip peripheral, except for a small transaction delay.

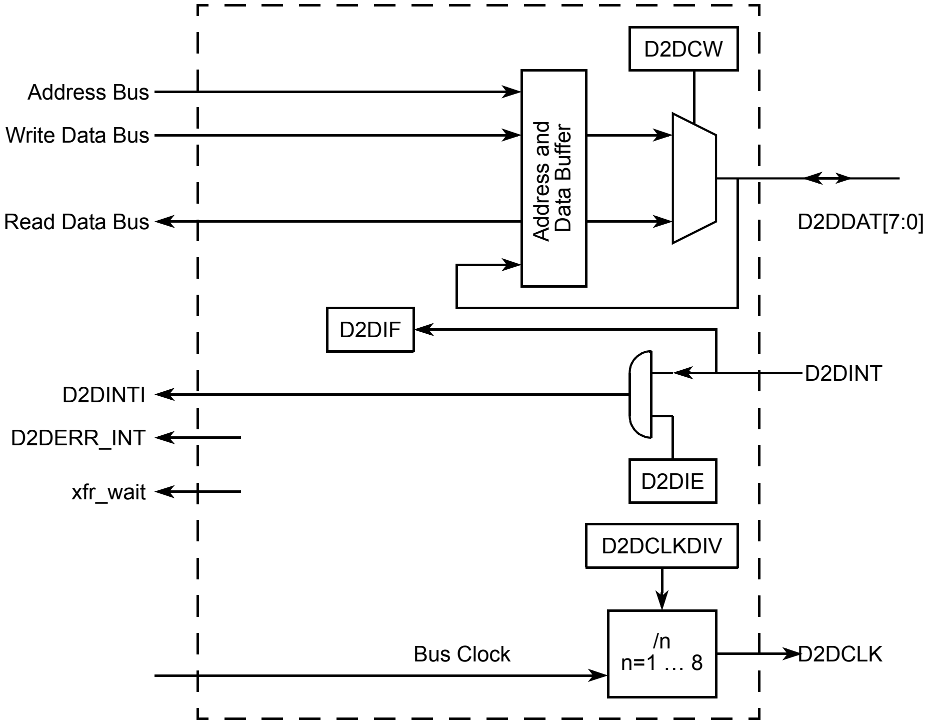


Figure 114. Die-to-die initiator (D2DI) block diagram

4.26.1.2 Features

The main features of this block are

- Software transparent, memory mapped access to peripherals on target die
  - 256 Byte address window
  - Supports blocking read or write as well as non-blocking write transactions
- Scalable interface clock divide by 1, 2, 3 and 4 of bus clock
- Clock halt on system STOP
- Configurable for 4- or 8-bit wide transfers
- Configurable timeout period
- Non-maskable interrupt on transaction errors
- Transaction Status and Error Flags
- Interrupt enable for receiving interrupt (from D2D target)

4.26.1.3 Modes of operation

4.26.1.3.1 D2DI in STOP/WAIT mode

The D2DI stops working in STOP/WAIT mode. The D2DCLK signal as well as the data signals used are driven low (only after the end of the current high phase, as defined by D2DCLKDIV).

Waking from STOP/WAIT mode, the D2DCLK line starts clocking again and the data lines will be driven low until the first transaction starts.

STOP and WAIT mode are entered by different CPU instructions. In the WAIT mode the behavior of the D2DI can be configured (D2DSWAI). Every (enabled) interrupt can be used to leave the STOP and WAIT mode.

4.26.1.3.2 D2DI in special modes

The MCU can enter a special mode (used for test and debugging purposes as well as programming the FLASH). In the D2DI, the “write-once” feature is disabled. See the MCU description for details.

4.26.2 External signal description

The D2DI optionally uses 6 or 10 port pins. The functions of those pins depends on the settings in the D2DCTL0 register, when the D2DI module is enabled.

4.26.2.1 D2DCLK

When the D2DI is enabled, this pin is the clock output. This signal is low if the initiator is disabled, in STOP mode or in WAIT mode (with D2DSWAI asserted), otherwise it is a continuos clock. This pin may be shared with general purpose functionality if the D2DI is disabled.

4.26.2.2 D2DDAT[7:4]

When the D2DI is enabled and the interface connection width D2DCW is set to be 8-bit wide, those lines carry the data bits 7:4 acting as outputs or inputs. When they act as inputs pull-down elements are enabled. If the D2DI is disabled or if the interface connection width is set as 4-bit wide, the pins may be shared with general purpose pin functionality.

4.26.2.3 D2DDAT[3:0]

When the D2DI is enabled those lines carry the data bits 3:0 acting as outputs or inputs. When they act as inputs pull-down elements are enabled. If the D2DI is disabled the pins and may be shared with general purpose pin functionality.

4.26.2.4 D2DINT

The D2DINT is an active input interrupt input driven by the target device. The pin has an active pull-down device. If the D2DI is disabled, the pin may be shared with general purpose pin functionality.

Table 547. Signal properties

Name	Primary (D2DEN=1)	I/O	Secondary (D2DEN=0)	Reset	Comment	Pull down
D2DDAT[7:0]	Bidirectional Data Lines	I/O	GPIO	0	driven low if in STOP mode	Active[306]
D2DCLK	Interface Clock Signal	O	GPIO	0	low if in STOP mode	—
$\overline{\text{D2DINT}}$	Active High Interrupt	I	GPIO	—	—	Active[307]

Notes

[306] Active if in input state, only if D2DEN=1

[307] only if D2DEN=1

See the port interface module (PIM) guide for details of the GPIO function.

4.26.3 Memory map and register definition

4.26.3.1 Memory map

The D2DI memory map is split into three sections.

1. An eight byte set of control registers
2. A 256 byte window for blocking transactions
3. A 256 byte window for non-blocking transactions

See the chapter “Device Memory Map” for the register layout (distribution of these sections).

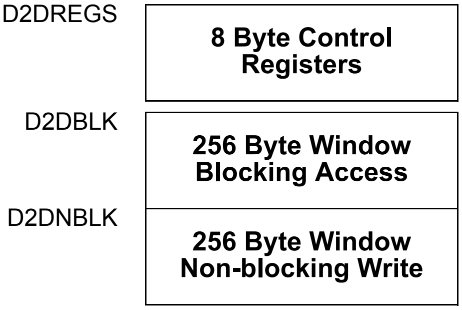

D2DREGS
8 Byte Control Registers
D2DBLK
256 Byte Window Blocking Access
D2DNBLK
256 Byte Window Non-blocking Write

Figure 115. D2DI top level memory map

A summary of the registers associated with the D2DI block is shown in Table 548. Detailed descriptions of the registers and bits are given in the subsections that follow.

Table 548. D2DI register summary

Offset	Register name		Bit 7	6	5	4	3	2	1	Bit 0
0x0	D2DCTL0	R	D2DEN	D2DCW	D2DSWAI	0	0	0	D2DCLKDIV[1:0]	
		W				▨	▨	▨		
0x1	D2DCTL1		D2DIE	0	0	0	TIMEOUT[3:0]			
				▨	▨	▨				
0x2	D2DSTAT0	R	ERRIF	ACKERF	CNCLF	TIMEF	TERRF	PARF	PAR1	PAR0
		W		▨	▨	▨	▨	▨	▨	▨
0x3	D2DSTAT1		D2DIF	D2DBSY	0	0	0	0	0	0
				▨	▨	▨	▨	▨	▨	▨
0x4	D2DADRHI	R	RWB	SZ8	0	NBLK	0	0	0	0
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x5	D2DADRLO	R	ADR[7:0]							
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x6	D2DDATAHI	R	DATA[15:8]							
		W	▨	▨	▨	▨	▨	▨	▨	▨
0x7	D2DDATALO	R	DATA[7:0]							
		W	▨	▨	▨	▨	▨	▨	▨	▨

▨ = Unimplemented or Reserved

4.26.3.2 Register definition

4.26.3.3 D2DI control register 0 (D2DCTL0)

This register is used to enable and configure the interface width, the wait behavior and the frequency of the interface clock.

Table 549. D2DI control register 0 (D2DCTL0)

Offset 0x0 Access: User read/write

	7	6	5	4	3	2	1	0
R	D2DEN	D2DCW	D2DSWAI	0	0	0	D2DCLKDIV[1:0]	
W								
Reset	0	0	0	0	0	0	0	0

Table 550. D2DCTL0 register field descriptions

Field	Description
7 D2DEN	**D2DI Enable** — Enables the D2DI module. This bit is write-once in normal mode and can always be written in special modes. 0 D2DI initiator is disabled. No lines are not used, the pins have their GPIO (secondary) function. 1 D2DI initiator is enabled. After setting D2DEN=1 the D2DDAT[7:0] (or [3:0], see D2DCW) lines are driven low with the IDLE command; the D2DCLK is driven by the divided bus clock.
6 D2DCW	**D2D Connection Width** — Sets the number of data lines used by the interface. This bit is write-once in normal modes and can always be written in special modes. 0 Lines D2DDAT[3:0] are used for four line data transfer. D2DDAT[7:4] are unused. 1 All eight interface lines D2DDAT[7:0] are used for data transfer.
5 D2DSWAI	**D2D Stop In Wait** — Controls the WAIT behavior. This bit can be written at any time. 0 Interface clock continues to run if the CPU enters WAIT mode 1 Interface clock stops if the CPU enters WAIT mode.
4:2	Reserved, should be written to 0 to ensure compatibility with future versions of this interface.
1:0 D2DCLKDIV	**Interface Clock Divider** — Determines the frequency of the interface clock. These bits are write-once in normal modes and can be always written in special modes. See Figure 116 for details on the clock waveforms 00 Encoding 0. Bus clock divide by 1. 01 Encoding 1. Bus clock divide by 2. 10 Encoding 2. Bus clock divide by 3. 11 Encoding 3. Bus clock divide by 4.

The Clock Divider will provide the waveforms as shown in Figure 116. The duty cycle of the clock is not always 50%, the high cycle is shorter than 50% or equal but never longer, since this is beneficial for the transaction speed.

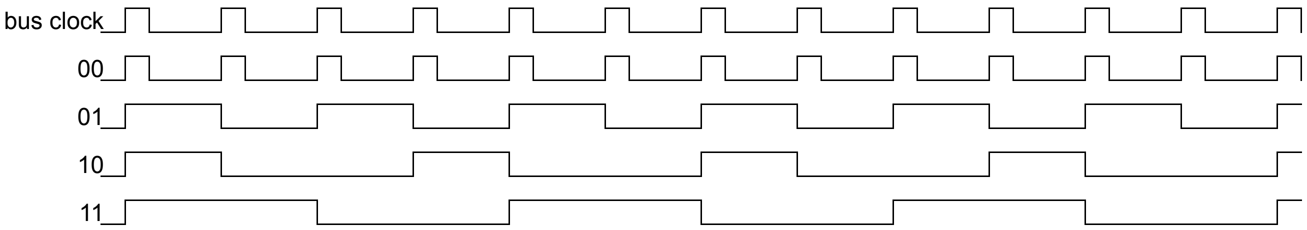


Figure 116. Interface clock waveforms for various D2DCLKDIV encoding

4.26.3.4 D2DI control register 1 (D2DCTL1)

This register is used to enable the D2DI interrupt and set number of D2DCLK cycles before a timeout error is asserted.

Table 551. D2DI control register 1 (D2DCTL1)

Offset 0x1 Access: User read/write

	7	6	5	4	3	2	1	0
R	D2DIE	0	0	0	TIMOUT[3:0]			
W								
Reset	0	0	0	0	0	0	0	0

Table 552. D2DCTL1 register field descriptions

Field	Description
7 D2DIE	**D2D Interrupt Enable** — Enables the external interrupt 0 External Interrupt is disabled 1 External Interrupt is enabled
6:4	Reserved, should be written to 0 to ensure compatibility with future versions of this interface.
3:0 TIMOUT	**Time-out Setting** — Defines the number of D2DCLK cycles to wait after the last transaction cycle until a timeout is asserted. In case of a timeout the TIMEF flag in the D2DSTAT0 register will be set. These bits are write-once in normal modes and can always be written in special modes. 0000 The acknowledge is expected directly after the last transfer, i.e. the target must not insert a wait cycle. 0001 - 1111: The target may insert up to TIMOUT wait states before acknowledging a transaction until a timeout is asserted

NOTE

"Write-once" means that after writing D2DCNTL0.D2DEN=1 the write accesses to these bits have no effect.

4.26.3.5 D2DI status register 0 (D2DSTAT0)

This register reflects the status of the D2DI transactions.

Table 553. D2DI status register 0 (D2DSTAT0)

Offset 0x2 Access: User read/write

	7	6	5	4	3	2	1	0
R	ERRIF	ACKERF	CNCLF	TIMEF	TERRF	PARF	PAR1	PAR0
W								
Reset	0	0	0	0	0	0	0	0

Table 554. D2DI status register 0 field descriptions

Field	Description
7 ERRIF	**D2DI error interrupt flag** — This status bit indicates that the D2D initiator has detected an error condition (summary of the following five flags).This interrupt is not locally maskable. Write a 1 to clear the flag. Writing a 0 has no effect. 0 D2DI has not detected an error during a transaction. 1 D2DI has detected an error during a transaction.
6 ACKERF	**Acknowledge Error Flag**— This read-only flag indicates that in the acknowledge cycle not all data inputs are sampled high, indicating a potential broken wire. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
5 CNCLF	**CNCLF** — This read-only flag indicates the initiator has canceled a transaction and replaced it by an IDLE command due to a pending error flag (ERRIF). This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
4 TIMEF	**Time Out Error Flag** — This read-only flag indicates the initiator has detected a time-out error. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.

Table 554. D2DI status register 0 field descriptions (continued)

Field	Description
3 TERRF	**Transaction Error Flag** — This read-only flag indicates the initiator has detected the error signal during the acknowledge cycle of the transaction. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
2 PARF	**Parity Error Flag** — This read-only flag indicates the initiator has detected a parity error. Parity bits[1:0] contain further information. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
1 PAR1	**Parity Bit** — P[1] as received by the D2DI
0 PAR0	**Parity Bit** — P[0] as received by the D2DI

4.26.3.6 D2DI status register 1 (D2DSTAT1)

This register holds the status of the external interrupt pin and an indicator about the D2DI transaction status.

Table 555. D2DI status register 1 (D2DSTAT1)

Offset 0x3 Access: User read

	7	6	5	4	3	2	1	0
R	D2DIF	D2DBSY	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Table 556. D2DSTAT1 register field descriptions

Field	Description
7 D2DIF	**D2D Interrupt Flag** — This read-only flag reflects the status of the D2DINT Pin. The D2D interrupt flag can only be cleared by a target specific interrupt acknowledge sequence. 0 External Interrupt is negated 1 External Interrupt is asserted
6 D2DBSY	**D2D Initiator Busy** — This read-only status bit indicates that a D2D transaction is ongoing. 0 D2D initiator idle. 1 D2D initiator transaction ongoing.
5:0	Reserved, should be masked to ensure compatibility with future versions of this interface.

4.26.3.7 D2DI address buffer register (D2DADR)

This read-only register contains information about the ongoing D2D interface transaction. The register content will be updated when a new transaction starts. In error cases the user can track back, which transaction failed.

Table 557. D2DI address buffer register (D2DADR)

Offset 0x4/0x5 Access: User read

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RWB	SZ8	0	NBLK	0	0	0	0	ADR[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 558. D2DI address buffer register bit descriptions

Field	Description
15 RWB	**Transaction Read-Write Direction** — This read-only bit reflects the direction of the transaction 0 Write Transaction 1 Read Transaction
14 SZ8	**Transaction Size** — This read-only bit reflects the data size of the transaction 0 16-bit transaction. 1 8-bit transaction.
13	Reserved, should be masked to ensure compatibility with future versions of this interface.
12 NBLK	**Transaction Mode** — This read-only bit reflects the mode of the transaction 0 Blocking transaction. 1 Non-blocking transaction.
11:8	Reserved, should be masked to ensure compatibility with future versions of this interface.
7:0 ADR[7:0]	**Transaction Address** — Those read-only bits contain the address of the transaction

4.26.3.8 D2DI data buffer register (D2DDATA)

This read-only register contains information about the ongoing D2D interface transaction. For a write transaction, the data becomes valid at the begin of the transaction. For a read transaction, the data will be updated during the transaction and is finalized when the transaction is acknowledged by the target. In error cases, the user can track back what has happened.

Table 559. D2DI data buffer register (D2DDATA)

Offset 0x6/0x7

Access: User read

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA15:0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 560. D2DI data buffer register bit descriptions

Field	Description
15:0 DATA	**Transaction Data** — Those read-only bits contain the data of the transaction

Both D2DDATA and D2DADR can be read with byte accesses.

4.26.4 Functional description

4.26.4.1 Initialization

Out of reset the interface is disabled. The interface must be initialized by setting the interface clock speed, the timeout value, the transfer width and finally enabling the interface. This should be done using a 16-bit write or if using 8-bit write D2DCTL1 must be written before D2D2CTL0.D2DEN=1 is written. Once it is enabled in normal modes, only a reset can disable it again (write-once feature).

4.26.4.2 Transactions

A transaction on the D2D Interface is triggered by writing to either the 256 byte address window or reading from the address window (see STAA/LDAA 0/1 in the next figure). Depending on which address window is used, a blocking or a non-blocking transaction is performed. The address for the transaction is the 8-bit wide window relative address. The data width of the CPU read or write instructions determines if 8-bit or 16-bit wide data are transferred. There is always only one transaction active. Figure 117 shows the various types of transactions explained in more detail below.

For all 16-bit read/write accesses of the CPU the addresses are assigned according the big-endian model:

word [15:8]: addr word[7:0]: addr+1

addr: byte-address (8 bit wide) inside the blocking or non-blocking window, as provided by the CPU and transferred to the D2D target
word: CPU data, to be transferred from/to the D2D target

The application must care for the stretched CPU cycles (limited by the TIMOUT value, caused by blocking or consecutive accesses), which could affect time limits, including COP (computer operates properly) supervision. The stretched CPU cycles cause the “CPU halted” phases (see Figure 117).

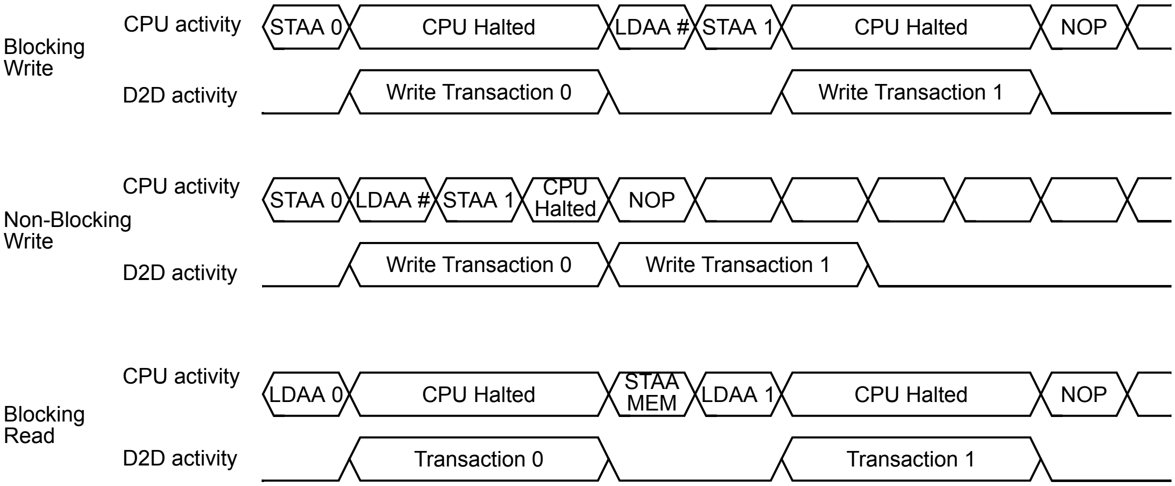


Figure 117. Blocking and non-blocking transfers.

4.26.4.2.1 Blocking writes

When writing to the address window associated with blocking transactions, the CPU is held until the transaction is completed, before completing the instruction. Figure 117 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAA    BLK_WINDOW+OFFS0 ; WRITE0 8-bit as a blocking transaction
LDAA    #BYTE1
STAA    BLK_WINDOW+OFFS1 ; WRITE1 is executed after WRITE0 transaction is completed
NOP
```

Blocking writes should be used when clearing interrupt flags, located in the target or other writes which require that the operation at the target, is completed before proceeding with the CPU instruction stream.

4.26.4.3 Non-blocking writes

When writing to the address window associated with non-blocking transactions, the CPU can continue before the transaction is completed. However, if there was an ongoing transaction when doing the 2nd write, the CPU is held until the first one is completed, and before executing the 2nd one. Figure 117 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAA    NONBLK_WINDOW+OFFS0; write 8-bit as a blocking transaction
LDAA    #BYTE1                 ; load next byte
STAA    NONBLK_WINDOW+OFFS1; executed right after the first
NOP
```

As the figure illustrates, non-blocking writes have a performance advantage, but care must be taken that the following instructions are not affected by the change in the target caused by the previous transaction.

4.26.4.4 Blocking read

When reading from the address window associated with blocking transactions, the CPU is held until the data is returned from the target, before completing the instruction. Figure 117 shows the behavior of the CPU for a blocking read transaction shown in the following example.

```
LDAA    BLK_WINDOW+OFFS0 ; Read 8-bit as a blocking transaction
STAA    MEM              ; Store result to local Memory
LDAA    BLK_WINDOW+OFFS1 ; Read 8-bit as a blocking transaction
```

4.26.4.5 Non-blocking read

Read access to the non-blocking window is reserved for future use. When reading from the address window associated with non-blocking writes, the read returns an all 0s data byte or word. This behavior can change in future revisions.

4.26.4.6 Transfer width

8-bit wide writes or reads are translated into 8-bit wide interface transactions. 16-bit wide, aligned writes or reads are translated into a16-bit wide interface transactions. 16-bit wide, misaligned writes or reads are split up into two consecutive 8-bit transactions with the transaction on the odd address first followed by the transaction on the next higher even address. Due to the much more complex error handling (by the MCU), misaligned 16-bit transfers should be avoided.

4.26.4.7 Error conditions and handling faults

Since the S12 CPU (as well as the S08) does not provide a method to abort a transfer once started, the D2DI asserts a D2DERRINT. The ERRIF Flag is set in the D2DSTAT0 register. Depending on the error condition, further error flags will be set, as described below. The content of the address and data buffers are frozen and all transactions will be replaced by an IDLE command, until the error flag is cleared. If an error is detected during the read transaction of a read-modify-write instruction, or a non-blocking write transaction was followed by another write or read transaction, the second transaction is cancelled. The CNCLF is set in the D2DSTAT0 register to indicate that a transaction has been cancelled. The D2DERRINT handler can read the address and data buffer register to assess the error situation. Any further transaction will be replaced by IDLE until the ERRIF is cleared.

4.26.4.7.1 Missing acknowledge

If the target detects a wrong command, it will not send back an acknowledge. The same situation occurs if the acknowledge is corrupted. The D2DI detects this missing acknowledge after the timeout period configured in the TIMOUT parameter of the D2DCTL1 register. In case of a timeout, the ERRIF and the TIMEF flags in the D2DSTAT0 register will be set.

4.26.4.7.2 Parity error

In the final acknowledge cycle of a transaction, the target sends two parity bits. If this parity does not match the parity calculated by the initiator, the ERRIF and the PARF flags in the D2DSTAT0 register will be set. The PAR[1:0] bits contain the parity value received by the D2DI.

4.26.4.7.3 Error signal

During the acknowledge cycle the target can signal a target specific error condition. If the D2DI finds the error signal asserted during a transaction, the ERRIF and the TERRF flags in the D2DSTAT0 register will be set.

4.26.4.8 Low power mode options

4.26.4.8.1 D2DI in run mode

In run mode, with the D2D Interface enable (D2DEN) bit in the D2D control register 0 clear, the D2DI system is in a low-power, disabled state. D2D registers remain accessible, but clocks to the core of this module are disabled. On D2D lines the GPIO function is activated.

4.26.4.8.2 D2DI in wait mode

D2DI operation in wait mode depends upon the state of the D2DSWAI bit in D2D control register 0.

- If D2DSWAI is clear, the D2DI operates normally when the CPU is in the wait mode
- If D2DSWAI is set and the CPU enters the wait mode, any pending transmission is completed. When the D2DCLK output is driven low then the clock generation is stopped, all internal clocks to the D2DI module are stopped as well and the module enters a power saving state.

4.26.4.8.3 D2DI in stop mode

If the CPU enters the STOP mode, the D2DI shows the same behavior as for the wait mode with an activated D2DSWAI bit.

4.26.4.8.4 Reset

In case of reset any transaction is immediately stopped and the D2DI module is disabled.

4.26.4.8.5 Interrupts

The D2DI only originates interrupt requests when D2DI is enabled (D2DIE bit in D2DCTL0 set). There are two different interrupt requests from the D2D module. The interrupt vector offset and interrupt priority are chip dependent.

4.26.4.8.5.1 D2D external interrupt

This is a level sensitive active high external interrupt driven by the D2DINT input. This interrupt is enabled if the D2DIE bit in the D2DCTL1 register is set. The interrupt must be cleared using an target specific clearing sequence. The status of the D2D input pin can be observed by reading the D2DIF bit in the D2DSTAT1 register.

The D2DINIT signal is also asserted in the wait and stop mode; it can be used to leave these modes.

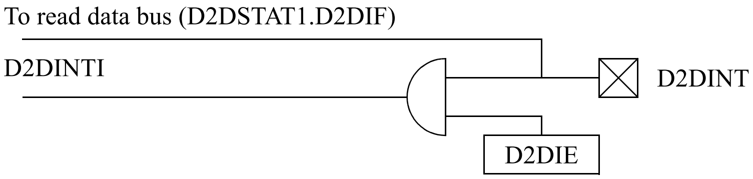


Figure 118. D2D external interrupt scheme

4.26.4.8.5.2 D2D error interrupt

Those D2D interface specific interrupts are level sensitive and are all cleared by writing a 1 to the ERRIF flag in the D2DSTAT0 register. This interrupt is not locally maskable and should be tied to the highest possible interrupt level in the system, on an S12 architecture to the XIRQ. See the chapter “Vectors” of the MCU description for details.

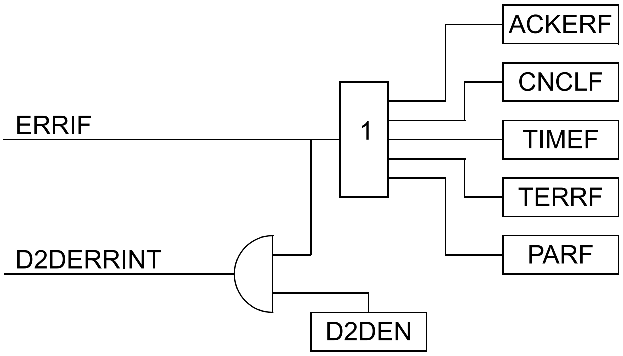


Figure 119. D2D internal interrupts

4.26.5 Initialization information

During initialization, the transfer width, clock divider, and timeout value must be set according to the capabilities of the target device before starting any transaction. See the D2D Target specification for details.

4.26.6 Application information

4.26.6.1 Entering low-power mode

The D2DI module is typically used on a microcontroller along with an analog companion device containing the D2D target interface and supplying the power. Interface specification does not provide special wires for signalling low power modes to the target device. The CPU should determine when it is time to enter one of the above power modes.The basic flow is as follows:

1. CPU determines there is no more work pending.
2. CPU writes a byte to a register on the analog die using blocking write configuring which mode to enter.
3. Analog die acknowledges that write sending back an acknowledge symbol on the interface.
4. CPU executes WAIT or STOP command.
5. Analog die can enter low power mode - (S12 needs some more cycles to stack data!)

```
; Example shows S12 code
SEI        ; disable interrupts during test
; check is there is work pending?
; if yes, branch off and re-enable interrupt
; else
LDAA       #STOP_ENTRY
STAA       MODE_REG         ; store to the analog die mode reg (use blocking write here)
CLI        ; re-enable right before the STOP instruction
STOP       ; stack and turn off all clocks inc. interface clock
```

For wake-up from STOP the basic flow is as follows:

1. Analog die detects a wake-up condition e.g. on a switch input or start bit of a LIN message.
2. Analog die exits Voltage Regulator low power mode.
3. Analog die asserts the interrupt signal D2DINT.
4. CPU starts clock generation.
5. CPU enters interrupt handler routine.
6. CPU services interrupt and acknowledges the source on the analog die.

NOTE

Entering STOP mode or WAIT mode with D2DSWAI asserted, the clock will complete the high duty cycle portion and settle at a low level.

5 MM912_637 - trimming

5.1 Introduction

To ensure the high precision requirements over a wide temperature and lifetime range, the MM912_637 uses several trimming and calibration techniques. Due to the advantage of the FLASH technology available in the microcontroller die, several factory trimmed values can be used to increase the overall device accuracy.

Trimming will use factory measured and calculated values stored in the microcontroller IFR (Information Register) to be loaded into specific registers in the MCU and analog die at system power up.

Calibration would be done during operation of the system using internal references or specific measurement procedures. As calibration is an essential part of the signal acquisition, see Section 4.8.5, “Calibration" as part of Section 4.8, “Channel acquisition".

NOTE

The MM912_637 trimming is primarily used to achieve the specified analog die parameters. The only valid trimming of the MCU die, the Internal Oscillator Trimming (ICG) will be automatically stored into the MCU trimming register during power up. See Section 4.23.3.2.15, “9S12I128PIMV1 IRC1M trim registers (CPMUIRCTRIMH / CPMUIRCTRIML)".

5.2 IFR trimming content and location

All device trimming information are stored in the MCU Information Register (IFR) located at the following address. See also Section 4.25, “128 kByte flash module (S12FTMRC128K1V1)".

Table 561. IFR location

Global address	Size (bytes)	Field description
0x01_8000 - 0x01_8007	8	Unique Device ID
0x01_8008 - 0x01_80B5	174	Reserved
0x01_80B6 - 0x01_80B7	2	Version ID[308]
0x01_80B8 - 0x01_80BF	8	Reserved
0x01_80C0 - 0x01_80FF	64	Analog Die Trimming Information (Program Once Field)

Notes

[308] Used to track firmware patch versions, see Section 4.25.4.2, “IFR version ID word".

NOTE

The Program Once reserved field can be read using the Read Once command as described in Section 4.25.4.5.4, “Read once command".

5.2.1 IFR - trimming content for analog die functionality

The following table shows the details of the 64 byte (0x01_80C0 - 0x01_80FF) Program Once Field Content used to store the Analog Die Trimming Information. Refer to Section 4.25.4.5.4, “Read once command", for access instructions.

Table 562. Analog die trimming information

Global address	OFFSET		Byte description								Target register	
	HEX	DEC	7	6	5	4	3	2	1	0	Name	Offset
0x01_80C0	00	00							IGC4[9:8]		COMP_IG4 (hi)	0xB0
0x01_80C1	01	01	IGC4[7:0]								COMP_IG4 (lo)	
0x01_80C2	02	02							IGC8[9:8]		COMP_IG8 (hi)	0xB2
0x01_80C3	03	03	IGC8[7:0]								COMP_IG8 (lo)	

Table 562. Analog die trimming information (continued)

Global address	OFFSET		Byte description								Target register	
	HEX	DEC	7	6	5	4	3	2	1	0	Name	Offset
0x01_80C4	04	04							IGC16[9:8]		COMP_IG16 (hi)	0xB4
0x01_80C5	05	05	IGC16[7:0]								COMP_IG16 (lo)	
0x01_80C6	06	06							IGC32[9:8]		COMP_IG32 (hi)	0xB6
0x01_80C7	07	07	IGC32[7:0]								COMP_IG32 (lo)	
0x01_80C8	08	08							IGC64[9:8]		COMP_IG64 (hi)	0xB8
0x01_80C9	09	09	IGC64[7:0]								COMP_IG64 (lo)	
0x01_80CA	0A	10							IGC128[9:8]		COMP_IG128 (hi)	0xBA
0x01_80CB	0B	11	IGC128[7:0]								COMP_IG128 (lo)	
0x01_80CC	0C	12							IGC256[9:8]		COMP_IG256 (hi)	0xBC
0x01_80CD	0D	13	IGC256[7:0]								COMP_IG256 (lo)	
0x01_80CE	0E	14							IGC512[9:8]		COMP_IG512 (hi)	0xBE
0x01_80CF	0F	15	IGC512[7:0]								COMP_IG512 (lo)	
0x01_80D0	10	16			TCIBG2[2:0]			SLPBG[2:0]			TRIM_BG0 (hi)	0xE0
0x01_80D1	11	17			IBG2[2:0]			IBG1[2:0]			TRIM_BG0 (lo)	0xE1
0x01_80D2	12	18	UBG3	DBG3	TCBG2[2:0]			TCBG1[2:0]			TRIM_BG1 (hi)	0xE2
0x01_80D3	13	19						SLPBG[2:0]			TRIM_BG1 (lo)	0xE3
0x01_80D4	14	20	V1P2BG2[3:0]				V1P2BG1[3:0]				TRIM_BG2 (hi)	0xE4
0x01_80D5	15	21	V2P5BG2[3:0]				V2P5BG1[3:0]				TRIM_BG2 (lo)	0xE5
0x01_80D6	16	22								LIN	TRIM_LIN	0xE6
0x01_80D7	17	23								LVT	TRIM_LVT	0xE7
0x01_80D8	18	24				LPOSC[12:8]					TRIM_OSC (hi)	0xE8
0x01_80D9	19	25	LPOSC[7:0]								TRIM_OSC (lo)	0xE9
0x01_80DA	1A	26	VOC_S[7:0]								COMP_VO	0xAA[309]
0x01_80DB	1B	27	VOC_O[7:0]								COMP_VO	0xAA[309]
0x01_80DC	1C	28	VOC_S_CHOP[7:0] (Chopper Mode)								COMP_VO	0xAA[309]
0x01_80DD	1D	29	VOC_O_CHOP[7:0] (Chopper Mode)								COMP_VO	0xAA[309]
0x01_80DE	1E	30							VSGC[9:8]		COMP_VSG (hi)	0xAC[309]
0x01_80DF	1F	31	VSGC[7:0]								COMP_VSG (lo)	
0x01_80E0	20	32							VOGC[9:8]		COMP_VOG (hi)	0xAC[309]
0x01_80E1	21	33	VOGC[7:0]								COMP_VOG (lo)	
0x01_80E2	22	34	ITO[7:0]								COMP_ITO	0xD0
0x01_80E3	23	35	ITG[7:0]								COMP_ITG	0xD1
0x01_80E4	24	36	BG3 diag measurement from Vsense channel after cal at room								GAIN_CAL_VSENSE_ROOM (hi)	n.a.
0x01_80E5	25	37									GAIN_CAL_VSENSE_ROOM (lo)	n.a.
0x01_80E6	26	38	BG3 diag measurement from Vopt channel after cal at room								GAIN_CAL_VOPT_ROOM (hi)	n.a.
0x01_80E7	27	39									GAIN_CAL_VOPT_ROOM (lo)	n.a.

Table 562. Analog die trimming information (continued)

Global address	OFFSET		Byte description								Target register	
	HEX	DEC	7	6	5	4	3	2	1	0	Name	Offset
0x01_80E8	28	40	BG3 diag measurement from I channel (gain4) at room								GAIN_CAL_IG4_ROOM (hi)	n.a.
0x01_80E9	29	41									GAIN_CAL_IG4_ROOM (med)	n.a.
0x01_80EA	2A	42									GAIN_CAL_IG4_ROOM (lo)	n.a.
0x01_80EB	2B	43	Reserved									
0x01_80EC	2C	44	COMP_VSG_COLD[7:0]								VSENSE Channel Gain Compensation - COLD Temp[310]	n.a.
0x01_80ED	2D	45	COMP_VSG_HOT[7:0]								VSENSE Channel Gain Compensation - HOT Temp[310]	n.a.
0x01_80EE	2E	46	COMP_VOG_COLD[7:0]								VOPT Channel Gain Compensation - COLD Temp[310]	n.a.
0x01_80EF	2F	47	COMP_VOG_HOT[7:0]								VOPT Channel Gain Compensation - HOT Temp[310]	n.a.
0x01_80F0	30	48	IGC4_COLD[7:0]								Current Channel Gain (4) Compensation - COLD Temp[310]	n.a.
0x01_80F1	31	49	IGC4_HOT[7:0]								Current Channel Gain (4) Compensation - HOT Temp[310]	n.a.
0x01_80F2	32	50	IGC8_COLD[7:0]								Current Channel Gain (8) Compensation - COLD Temp[310]	n.a.
0x01_80F3	33	51	IGC8_HOT[7:0]								Current Channel Gain (8) Compensation - HOT Temp[310]	n.a.
0x01_80F4	34	52	IGC16_COLD[7:0]								Current Channel Gain (16) Compensation - COLD Temp[310]	n.a.
0x01_80F5	35	53	IGC16_HOT[7:0]								Current Channel Gain (16) Compensation - HOT Temp[310]	n.a.
0x01_80F6	36	54	IGC32_COLD[7:0]								Current Channel Gain (32) Compensation - COLD Temp[310]	n.a.
0x01_80F7	37	55	IGC32_HOT[7:0]								Current Channel Gain (32) Compensation - HOT Temp[310]	n.a.
0x01_80F8	38	56	IGC64_COLD[7:0]								Current Channel Gain (64) Compensation - COLD Temp[310]	n.a.
0x01_80F9	39	57	IGC64_HOT[7:0]								Current Channel Gain (64) Compensation - HOT Temp[310]	n.a.
0x01_80FA	3A	58	IGC128_COLD[7:0]								Current Channel Gain (128) Compensation - COLD Temp[310]	n.a.
0x01_80FB	3B	59	IGC128_HOT[7:0]								Current Channel Gain (128) Compensation - HOT Temp[310]	n.a.
0x01_80FC	3C	60	IGC256_COLD[7:0]								Current Channel Gain (256) Compensation - COLD Temp[310]	n.a.
0x01_80FD	3D	61	IGC256_HOT[7:0]								Current Channel Gain (256) Compensation - HOT Temp[310]	n.a.
0x01_80FE	3E	62	IGC512_COLD[7:0]								Current Channel Gain (512) Compensation - COLD Temp[310]	n.a.

Table 562. Analog die trimming information (continued)

Global address	OFFSET		Byte description								Target register	
	HEX	DEC	7	6	5	4	3	2	1	0	Name	Offset
0x01_80FF	3F	63	IGC512_HOT[7:0]								Current Channel Gain (512) Compensation - HOT Temp(310)	n.a.

Notes

309. Based on the selection of the voltage measurement source (VSENSE or VOPT) and the activation of chopper mode.

310. 7 Bit character with bit 7 (MSB) as sign (0 = “+”; 1 = “-”) with the difference to the corresponding room temperature value (e.g. 10000010 = “-2”).

5.2.2 Analog die trimming overview

5.2.2.1 Current channel gain compensation trim (COMP_IG4-COMP_IG512)

To achieve the specified accuracy of the current acquisition, the optimum trim value is calculated during final test and stored into the MCU FLASH memory. On device every power up, the corresponding trim value needs to be copied into the corresponding analog register via D2D interface. See Section 4.8, “Channel acquisition" for additional details.

5.2.2.2 Bandgap reference trimming (TRIM_BG0-TRIM_BG2)

To achieve the specified accuracy of the integrated voltage regulators on the analog die, the optimum trim value is calculated during final test and stored into the MCU FLASH memory. On device every power up, the corresponding trim value needs to be copied into the desired analog register via D2D interface.

5.2.2.3 LIN slope control trimming (TRIM_LIN)

To achieve the specified slope of the LIN output signal, the optimum trim information is determined during final test and stored into the IFR register block of the MCU FLASH memory. On device every power up, the corresponding trim value needs to be copied into the desired analog register via D2D interface.

5.2.2.4 Low voltage threshold trim (TRIM_LVT)

To achieve the specified low voltage behavior, on device every power up, the corresponding trim value (LVR) needs to be copied into the corresponding analog trim register via D2D interface.

5.2.2.5 Low power oscillator trimming (TRIM_OSC)

To achieve the specified accuracy of the analog low power reference frequency (f_{TOL_A}), the optimum trim value is calculated during final test and stored into the IFR register block of the MCU FLASH memory. On device every power up, the corresponding trim value needs to be copied into the desired analog register via D2D interface.

5.2.2.6 Voltage channel compensation (COMP_VOx, COMP_VSG, COMP_VOG)

To achieve the specified accuracy of the voltage channels, gain and offset compensation are trimmed during final test and stored into the IFR register block of the MCU FLASH memory. The information is used during the calibration procedure described in Section 4.8.5, “Calibration".

5.2.2.7 Temperature sense module trimming (COMP_ITO, COMP_ITG)

To achieve the specified accuracy of the internal temperature sense module, the optimum trim information is determined during final test at hot / cold temperature and stored into the IFR register block of the MCU FLASH memory. On device every power up, the corresponding trim value needs to be copied into the desired analog register via D2D interface.

5.2.2.8 Band gap reference - diagnostic measurements (GAIN_CAL_X_X)

To achieve the specified accuracy of the voltage and current channels, reference measurements are performed during final test and stored for different temperatures into the IFR register block of the MCU FLASH memory. The information is used during the calibration procedure described in Section 4.8.5, “Calibration".

5.2.2.9 Hot/cold gain compensation data (0x01_80EC - 0x01_80FF)

To achieve the specified accuracy of the voltage and current channels, reference measurements are performed during final test and stored for different temperatures into the IFR register block of the MCU FLASH memory. The information is used during the calibration procedure described in Section 4.8.5, “Calibration".

5.3 Memory map and registers

5.3.1 Overview

This section provides a detailed description of the memory map and registers for the analog die trimming excluding registers used for calibration located from offset 0xE0 to 0xEF. Refer to Section 4.8.5, “Calibration" for details on Current channel gain compensation trim (COMP_IG4-COMP_IG512), Voltage channel compensation (COMP_VOx, COMP_VSG, COMP_VOG), Temperature sense module trimming (COMP_ITO, COMP_ITG), Band gap reference - diagnostic measurements (GAIN_CAL_X_X) and Hot/cold gain compensation data (0x01_80EC - 0x01_80FF).

5.3.2 Module memory map

The memory map for the Compensation module is given below in Table 62.

Table 563. Module memory map

Offset[311]	Name		7	6	5	4	3	2	1	0
0xE0	TRIM_BG0 (hi)	R	0	0	TCIBG2[2:0]			TCIBG1[2:0]		
	Trim bandgap 0	W								
0xE1	TRIM_BG0 (lo)	R	0	0	IBG2[2:0]			IBG1[2:0]		
	Trim bandgap 0	W								
0xE2	TRIM_BG1 (hi)	R	UBG3	DBG3	TCBG2[2:0]			TCBG1[2:0]		
	Trim bandgap 1	W								
0xE3	TRIM_BG1 (lo)	R	0	0	0	0	0	SLPBG[2:0]		
	Trim bandgap 1	W								
0xE4	TRIM_BG2 (hi)	R	V1P2BG2[3:0]				V1P2BG1[3:0]			
	Trim bandgap 2	W								
0xE5	TRIM_BG2 (lo)	R	V2P5BG2[3:0]				V2P5BG1[3:0]			
	Trim bandgap 2	W								
0xE6	TRIM_LIN	R	0	0	0	0	0	0	0	LIN
	Trim LIN	W								
0xE7	TRIM_LVT	R	0	0	0	0	0	0	0	LVT
	Trim low voltage threshold	W								

Table 563. Module memory map (continued)

Offset[311]	Name		7	6	5	4	3	2	1	0
0xE8	TRIM_OSC (hi)	R								
	Trim LP oscillator	W				LPOSC[12:0]				
0xE9	TRIM_OSC (lo)	R								
	Trim LP oscillator	W								
0xEA- 0xEF	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes

[311] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

5.3.2.1 Trim bandgap 0 (TRIM_BG0 (hi))

Table 564. Trim bandgap 0 (TRIM_BG0 (hi))

Offset[312] 0xE0 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	TCIBG2[2:0]			TCIBG1[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

Notes

[312] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 565. Trim bandgap 0 (TRIM_BG0 (hi)) - register field descriptions

Field	Description
5-3 TCIBG2[2:0]	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, “IFR - trimming content for analog die functionality" for location information.
2-0 TCIBG1[2:0]	

5.3.2.2 Trim bandgap 0 (TRIM_BG0 (lo))

Table 566. Trim bandgap 0 (TRIM_BG0 (lo))

Offset[313] 0xE1 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	IBG2[2:0]			IBG1[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

Notes

[313] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 567. Trim bandgap 0 (TRIM_BG0 (lo)) - register field descriptions

Field	Description
5-3 IBG2[2:0]	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, “IFR - trimming content for analog die functionality" for location information.
2-0 IBG1[2:0]	

5.3.2.3 Trim bandgap 1 (TRIM_BG1 (hi))

Table 568. Trim bandgap 1 (TRIM_BG1 (hi))

Offset[(314)] 0xE2 Access: User read/write

	7	6	5	4	3	2	1	0
R W	UBG3	DBG3	TCBG2[2:0]			TCBG1[2:0]		
Reset	0	0	0	0	0	0	0	0

Notes

314. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 569. Trim bandgap 1 (TRIM_BG1 (hi)) - register field descriptions

Field	Description
7 UBG3	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, “IFR - trimming content for analog die functionality" for location information.
6 DBG3	
5-3 TCBG2[2:0]	
2-1 TCBG1[2:0]	

5.3.2.4 Trim bandgap 1 (TRIM_BG1 (lo))

Table 570. Trim bandgap 1 (TRIM_BG1 (lo))

Offset[(315)] 0xE3 Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	SLPBG[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

Notes

315. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 571. Trim bandgap 1 (TRIM_BG1 (lo)) - register field descriptions

Field	Description
2-0 SLPBG[2:0]	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, “IFR - trimming content for analog die functionality" for location information.

5.3.2.5 Trim bandgap 2 (TRIM_BG2 (hi))

Table 572. Trim bandgap 2 (TRIM_BG2 (hi))

Offset[316] 0xE4 Access: User read/write

	7	6	5	4	3	2	1	0
R W	V1P2BG2[3:0]				V1P2BG1[3:0]			
Reset	0	0	0	0	0	0	0	0

Notes

[316] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 573. Trim bandgap 2 (TRIM_BG2 (hi)) - register field descriptions

Field	Description
7-4 V1P2BG2[3:0	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, "IFR - trimming content for analog die functionality" for location information.
3-0 V1P2BG1[3:0	

5.3.2.6 Trim Bandgap 2 (TRIM_BG2 (lo))

Table 574. Trim bandgap 2 (TRIM_BG2 (hi))

Offset[317] 0xE5 Access: User read/write

	7	6	5	4	3	2	1	0
R W	V2P5BG2[3:0]				V2P5BG1[3:0]			
Reset	0	0	0	0	0	0	0	0

Notes

[317] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 575. Trim bandgap 2 (TRIM_BG2 (hi)) - register field descriptions

Field	Description
7-4 V2P5BG2[3:0	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, "IFR - trimming content for analog die functionality" for location information.
3-0 V2P5BG1[3:0	

5.3.2.7 trim LIN (TRIM_LIN)

Table 576. Trim LIN (TRIM_LIN)

Offset[318] 0xE6

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	LIN
W								
Reset	0	0	0	0	0	0	0	0

Notes

318. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 577. Trim LIN (TRIM_LIN) - register field descriptions

Field	Description
0 LIN	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, “IFR - trimming content for analog die functionality" for location information.

5.3.2.8 Trim low voltage threshold (TRIM_LVT)

Table 578. Trim low-voltage threshold (TRIM_LVT)

Offset[319] 0xE7

Access: User read/write

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	LVT
W								
Reset	0	0	0	0	0	0	0	0

Notes

319. Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

Table 579. Trim low-voltage threshold (TRIM_LVT) - register field descriptions

Field	Description
0 LVT	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, “IFR - trimming content for analog die functionality" for location information.

5.3.2.9 Trim LP oscillator (TRIM_OSC (hi), TRIM_OSC (lo))

Table 580. Trim LP oscillator (TRIM_OSC (hi), TRIM_OSC (lo))

Offset[320] 0xE8

Access: User read/write

	7	6	5	4	3	2	1	0
R				LPOSC[12:8]				
W								
Reset	0	0	0	0	0	0	0	0
R	LPOSC[7:0]							
W								
Reset	0	0	1	1	1	1	1	1

Notes

[320] Offset related to 0x0200 for blocking access and 0x300 for non blocking access within the global address space.

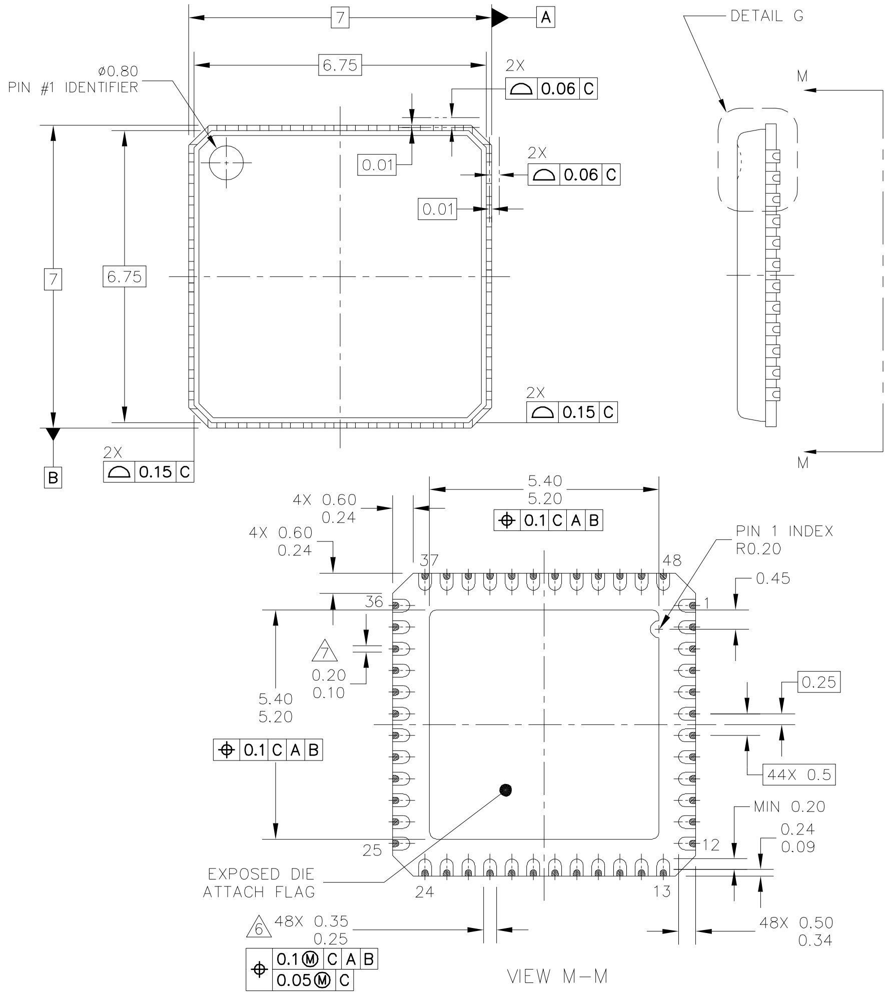
Table 581. Trim LP Oscillator (TRIM_OSC (hi), TRIM_OSC (lo)) - Register Field Descriptions

Field	Description
12-0 LPOSC[12:0]	The optimal content of this register is determined during final test and stored in the microcontroller IFR. For proper operation of the MM912_637, the content has to be copied to this location. See Section 5.2.1, “IFR - trimming content for analog die functionality" for location information.

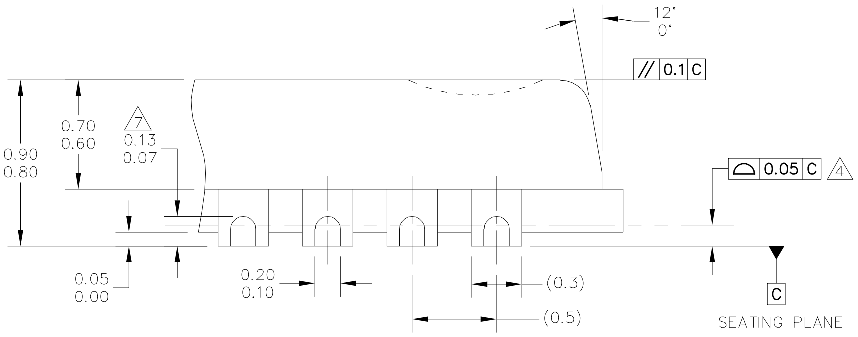
6 Packaging

6.1 Package dimensions

For the most current package revision, visit www.nxp.com and perform a keyword search using the “98A” listed below.

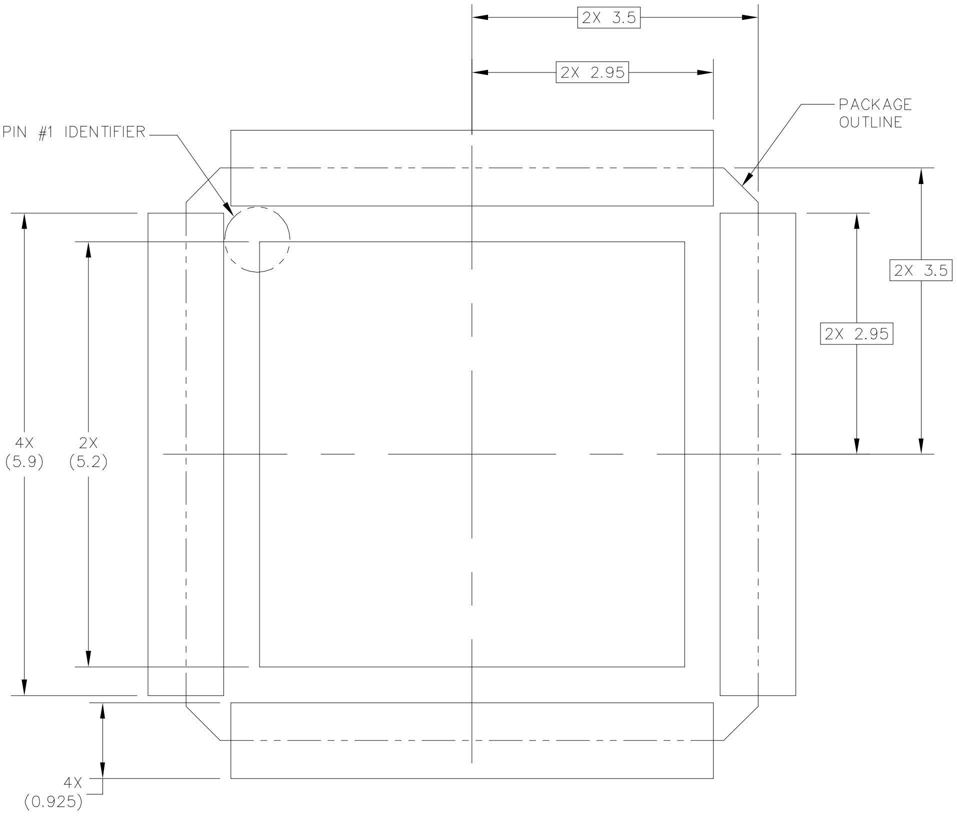


© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON–LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D	REV: B
	STANDARD: NON–JEDEC	
	SOT619–16	07 JAN 2016



DETAIL G
VIEW ROTATED 90° CW

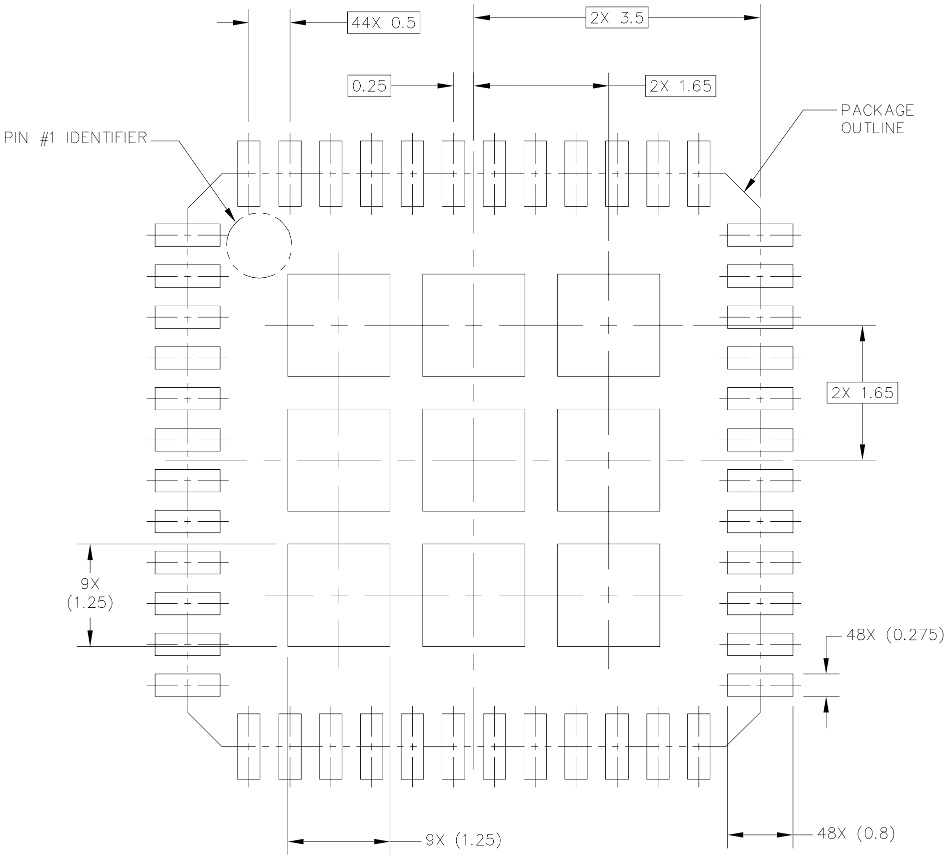
© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON–LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D REV: B STANDARD: NON–JEDEC SOT619–16 07 JAN 2016	



PCB DESIGN GUIDELINES – SOLDER MASK OPENINGS

THIS SHEET SERVES ONLY AS A GUIDELINE TO HELP DEVELOP A USER SPECIFIC SOLUTION. DEVELOPMENT EFFORT WILL STILL BE REQUIRED BY END USERS TO OPTIMIZE PCB MOUNTING PROCESSES AND BOARD DESIGN IN ORDER TO MEET INDIVIDUAL / SPECIFIC REQUIREMENTS.

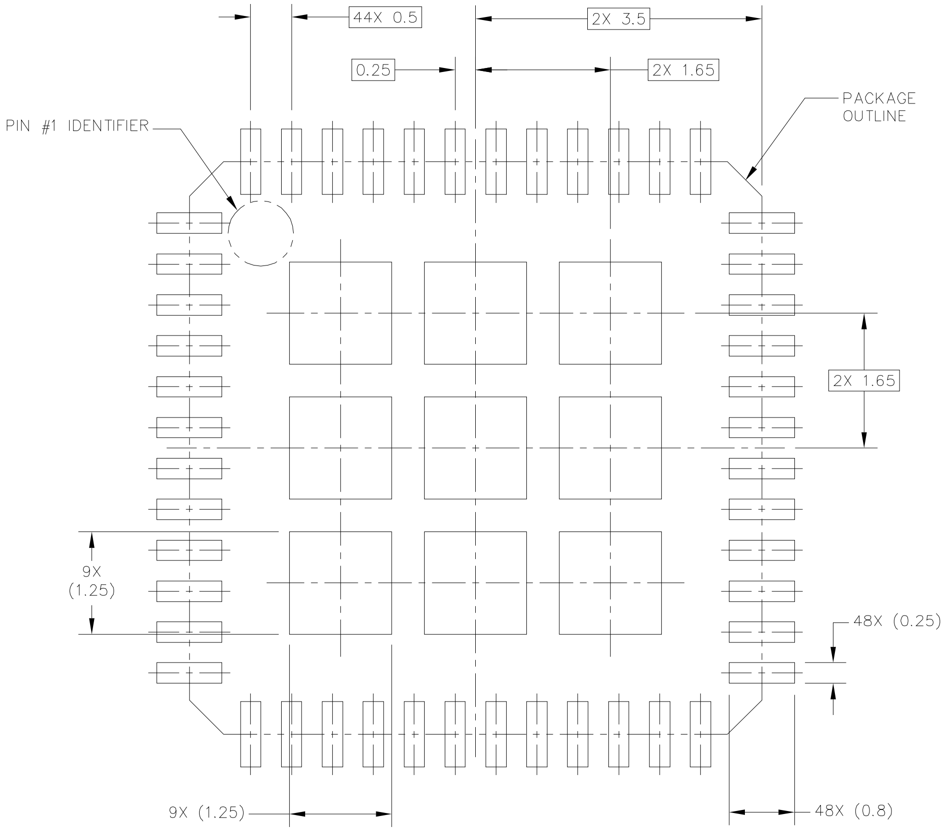
© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON–LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D REV: B STANDARD: NON–JEDEC SOT619–16 07 JAN 2016	



PCB CU GUIDELINES – I/O PADS & SOLDERABLE AREAS

THIS SHEET SERVES ONLY AS A GUIDELINE TO HELP DEVELOP A USER SPECIFIC SOLUTION. DEVELOPMENT EFFORT WILL STILL BE REQUIRED BY END USERS TO OPTIMIZE PCB MOUNTING PROCESSES AND BOARD DESIGN IN ORDER TO MEET INDIVIDUAL / SPECIFIC REQUIREMENTS.

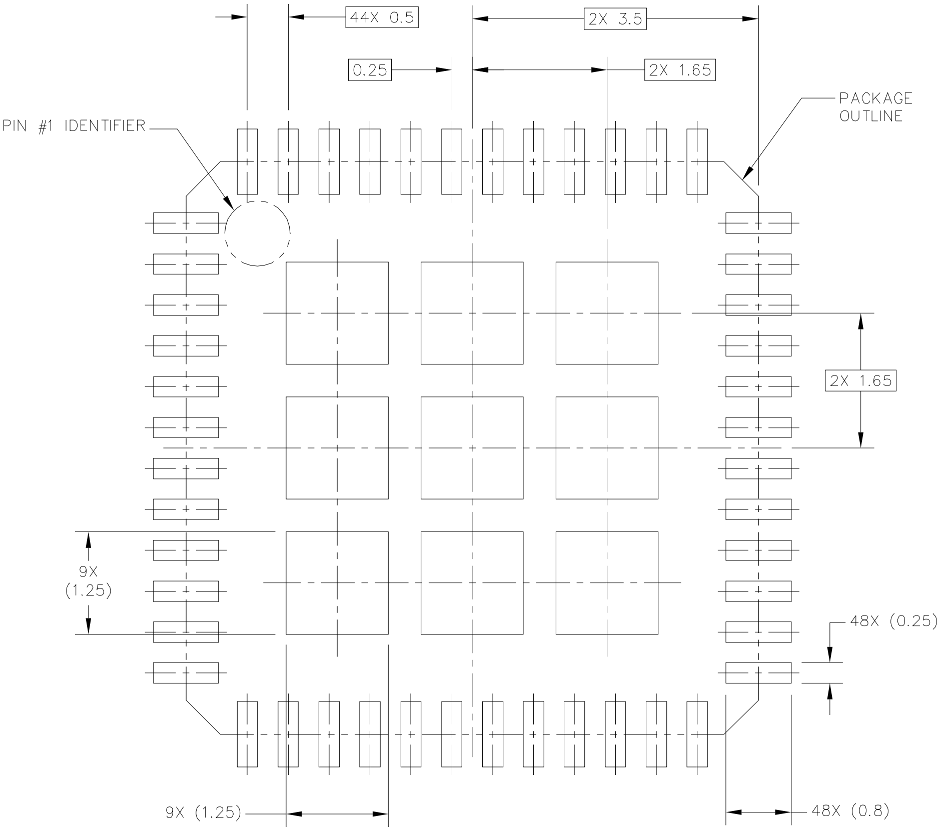
© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON-LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D	REV: B
	STANDARD: NON-JEDEC	
	SOT619-16	07 JAN 2016



SOLDER PASTE STENCIL DESIGN GUIDELINES

THIS SHEET SERVES ONLY AS A GUIDELINE TO HELP DEVELOP A USER SPECIFIC SOLUTION. DEVELOPMENT EFFORT WILL STILL BE REQUIRED BY END USERS TO OPTIMIZE PCB MOUNTING PROCESSES AND BOARD DESIGN IN ORDER TO MEET INDIVIDUAL / SPECIFIC REQUIREMENTS.

© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON-LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D	REV: B
	STANDARD: NON-JEDEC	
	SOT619-16	07 JAN 2016



SOLDER PASTE STENCIL DESIGN GUIDELINES

THIS SHEET SERVES ONLY AS A GUIDELINE TO HELP DEVELOP A USER SPECIFIC SOLUTION. DEVELOPMENT EFFORT WILL STILL BE REQUIRED BY END USERS TO OPTIMIZE PCB MOUNTING PROCESSES AND BOARD DESIGN IN ORDER TO MEET INDIVIDUAL / SPECIFIC REQUIREMENTS.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON-LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D	REV: A
	CASE NUMBER: 2095-01	13 JUL 2011
	STANDARD: NON-JEDEC	

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.

2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M−1994.

3. THIS IS NON JEDEC REGISTERED PACKAGE.

4. COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH FLAG.

5. MIN METAL GAP BETWEEN TERMINAL AND DIE PADDLE SHALL BE 0.25MM.

6. THIS DIMENSION APPLIES TO PLATED TERMINAL AND IS MEASURED BETWEEN 0.20 AND 0.25MM FROM TERMINAL TIP.

7. THIS DIMENSION APPLIES TO PLATED TERMINAL AND IS MEASURED BETWEEN 0.02MM AND PACKAGE CUTTING LINE.

© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON−LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D	REV: B
	STANDARD: NON−JEDEC	
	SOT619−16	07 JAN 2016

7 Revision history

Revision	Date	Description of Changes
1.0	4/2011	• Initial release
2.0	8/2011	• Minor changes throughout the document
3.0	1/2012	• Minor description changes and logo to align this data sheet to the Xtrinsic product platform. No content was altered.
4.0	10/2013	• Added targeted applications • Normal mode current max = 25 mA • MM912_637 stop/sleep/pseudo stop consumption combined for the product • Max operating conditions on Isense set to +/-0.3 V • Low voltage resetH lower limit = 1.9 V • Undervoltage interrupt assert lower limit = 4.55 V • Undervoltage interrupt deassert lower limit = 4.7 V • Description updated to fit current implementation on current ampere hour counter: The accumulator could be reset by writing 1 into AHCR register. The Ampere Hour Counter is counting after wake-up. In Normal mode, the accumulator register ACQ_AHC can be read out any time. • Added RESET to RESETA connection to Figure 4. Required / Recommended External Components. • Added connection information to VDDX/VDDRX, VDDH/VDDD2D and RESET/RESETA pins into pin description. • Repeated description of the Life Time Counter • Clarified to wait for PLL Lock after wake-up • Repeated description of Low Power Current trigger threshold
	3/2014	• Removed Xtrinsic logo. No other change to the document • Changed document status to Technical Data.
5.0	1/2015	• Remove option detailing reduced accuracy for voltage and current measurement if the temperature based compensation is not used (single temperature calibration). All known customers are using temperature based compensation for best accuracy. • Updated tables 62 and 182, and updated description corresponding descriiption tables 27 and 184. • Updated per PB #16604.
	8/2016	• Updated to NXP document form and style

How to Reach Us:

Home Page:
NXP.com

Web Support:
http://www.nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no expressed or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation, consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by the customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:

http://www.nxp.com/terms-of-use.html.

Document Number: MM912_637D1
Rev. 5.0
8/2016

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

NXP:

KIT912J637EVME MM912I637AM2EP MM912I637AM2EPR2 MM912I637AV1EP MM912I637AV1EPR2 MM912J637AM2EP MM912J637AM2EPR2 MM912J637AV1EP MM912J637AV1EPR2