Data Sheet, V3.0, Jan. 2001

# C161CS-32R/-L
# C161JC-32R/-L
# C161JI-32R/-L

## 16-Bit Single-Chip Microcontroller


Microcontrollers

Infineon technologies

Never stop thinking.

**Edition 2001-01**

**Published by Infineon Technologies AG,**
**St.-Martin-Strasse 53,**
**D-81541 München, Germany**



**Attention please!**

The information herein is given to describe certain components and shall not be considered as warranted characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Infineon Technologies is an approved CECC manufacturer.

**Information**

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office in Germany or our Infineon Technologies Representatives worldwide.

**Warnings**

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Data Sheet, V3.0, Jan. 2001

# C161CS-32R/-L
# C161JC-32R/-L
# C161JI-32R/-L

## 16-Bit Single-Chip Microcontroller

Microcontrollers

Infineon technologies

Never stop thinking.

**Revision History:** **2001-01** V3.0

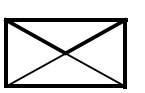
Previous Version: 2000-08 V2.0 (intermediate version)
1999-03 (Advance Information)

| Page | Subjects (major changes since last revision)[1)] |
|---|---|
| All | Converted to Infineon layout |
| 2 | Derivative Synopsis Table updated |
| 4, 6, 10, 18 | Programmable Interface Routing introduced |
| 27, 28 | GPT block diagrams updated |
| 29 | RTC description improved |
| 35 | OWD description improved |
| 39ff | RSTCON and SDLM registers added |
| 51 | Description of input/output voltage and hysteresis improved |
| 53 | Separate table for power consumption |
| 57 | Clock generation mode table updated |
| 60 | External clock drive specification improved |
| 62 | Reset calibration time specified, definition of $V_{AREF}$ improved |
| 63 | Programmable sample time introduced |
| 65ff | Timing tables updated to 25 MHz |

1) Changes refer to version 1999-03.

Controller Area Network (CAN): License of Robert Bosch GmbH

**We Listen to Your Comments**
Any information within this document that you feel is wrong, unclear or missing at all? Your feedback will help us to continuously improve the quality of this document. Please send your proposal (including a reference to this document) to:
**mcdocu.comments@infineon.com**



# C161CS/JC/JI

- High Performance 16-bit CPU with 4-Stage Pipeline
  - 80 ns Instruction Cycle Time at 25 MHz CPU Clock
  - 400 ns Multiplication (16 × 16 bit), 800 ns Division (32 / 16 bit)
  - Enhanced Boolean Bit Manipulation Facilities
  - Additional Instructions to Support HLL and Operating Systems
  - Register-Based Design with Multiple Variable Register Banks
  - Single-Cycle Context Switching Support
  - 16 MBytes Total Linear Address Space for Code and Data
  - 1024 Bytes On-Chip Special Function Register Area
- 16-Priority-Level Interrupt System with 59 Sources, Sample-Rate down to 40 ns
- 8-Channel Interrupt-Driven Single-Cycle Data Transfer Facilities via Peripheral Event Controller (PEC)
- Clock Generation via on-chip PLL (factors 1:1.5/2/2.5/3/4/5), via prescaler or via direct clock input
  - Additional 32 kHz Oscillator
- On-Chip Memory Modules
  - 2 KBytes On-Chip Internal RAM (IRAM)
  - 8 KBytes On-Chip Extension RAM (XRAM)
  - 256 KBytes On-Chip Mask ROM
- On-Chip Peripheral Modules
  - 12-Channel 10-bit A/D Converter with Programmable Conversion Time down to 7.8 µs
  - Two 16-Channel Capture/Compare Units (eight IO lines each)
  - Two Multi-Functional General Purpose Timer Units with 5 Timers
  - Two Asynchronous/Synchronous Serial Channels
  - High-Speed Synchronous Serial Channel (SPI)
  - On-Chip CAN Interface (Rev. 2.0B active, Full CAN / Basic CAN) with 15 Message Objects (**C161CS 2x, C161JC 1x**)
  - Serial Data Link Module (SDLM), compliant with J1850, supporting Class 2 (**C161JC/JI**)
  - IIC Bus Interface (10-bit Addressing, 400 kHz) with 2 Channels (multiplexed)
  - On-Chip Real Time Clock
- Up to 16 MBytes External Address Space for Code and Data
  - Programmable External Bus Characteristics for Different Address Ranges
  - Multiplexed or Demultiplexed External Address/Data Buses with 8-Bit or 16-Bit Data Bus Width
  - Five Programmable Chip-Select Signals
  - Hold- and Hold-Acknowledge Bus Arbitration Support

- Idle, Sleep, and Power Down Modes with Flexible Power Management
- Programmable Watchdog Timer and Oscillator Watchdog
- Up to 93 General Purpose I/O Lines,
  partly with Selectable Input Thresholds and Hysteresis
- Supported by a Large Range of Development Tools like C-Compilers,
  Macro-Assembler Packages, Emulators, Evaluation Boards, HLL-Debuggers,
  Simulators, Logic Analyzer Disassemblers, Programming Boards
- On-Chip Bootstrap Loader
- 128-Pin TQFP Package

This document describes several derivatives of the C161 group. **Table 1** enumerates these derivatives and summarizes the differences. As this document refers to all of these derivatives, some descriptions may not apply to a specific product.

**Table 1 C161CS/JC/JI Derivative Synopsis**

| Derivative | On-Chip Program Memory | Serial Bus Interface(s) | Maximum CPU Frequency |
|---|---|---|---|
| SAK-C161CS-32RF<br>SAB-C161CS-32RF | 256 KByte ROM | CAN1, CAN2 | 25 MHz |
| SAK-C161CS-LF<br>SAB-C161CS-LF | --- | CAN1, CAN2 | 25 MHz |
| SAK-C161JC-32RF<br>SAB-C161JC-32RF | 256 KByte ROM | CAN1, SDLM | 25 MHz |
| SAK-C161JC-LF<br>SAB-C161JC-LF | --- | CAN1, SDLM | 25 MHz |
| SAK-C161JI-32RF<br>SAB-C161JI-32RF | 256 KByte ROM | SDLM | 25 MHz |
| SAK-C161JI-LF<br>SAB-C161JI-LF | --- | SDLM | 25 MHz |

For simplicity all versions are referred to by the term **C161CS/JC/JI** throughout this document.

## Ordering Information

The ordering code for Infineon microcontrollers provides an exact reference to the required product. This ordering code identifies:

- the derivative itself, i.e. its function set, the temperature range, and the supply voltage
- the package and the type of delivery.

For the available ordering codes for the C161CS/JC/JI please refer to the **“Product Catalog Microcontrollers”**, which summarizes all available microcontroller variants.

*Note: The ordering codes for Mask-ROM versions are defined for each product after verification of the respective ROM code.*

## Introduction

The C161CS/JC/JI derivatives are high performance derivatives of the Infineon C166 Family of full featured single-chip CMOS microcontrollers. They combine high CPU performance (up to 12.5 million instructions per second) with high peripheral functionality and enhanced IO-capabilities. They also provide clock generation via PLL and various on-chip memory modules such as program ROM, internal RAM, and extension RAM.

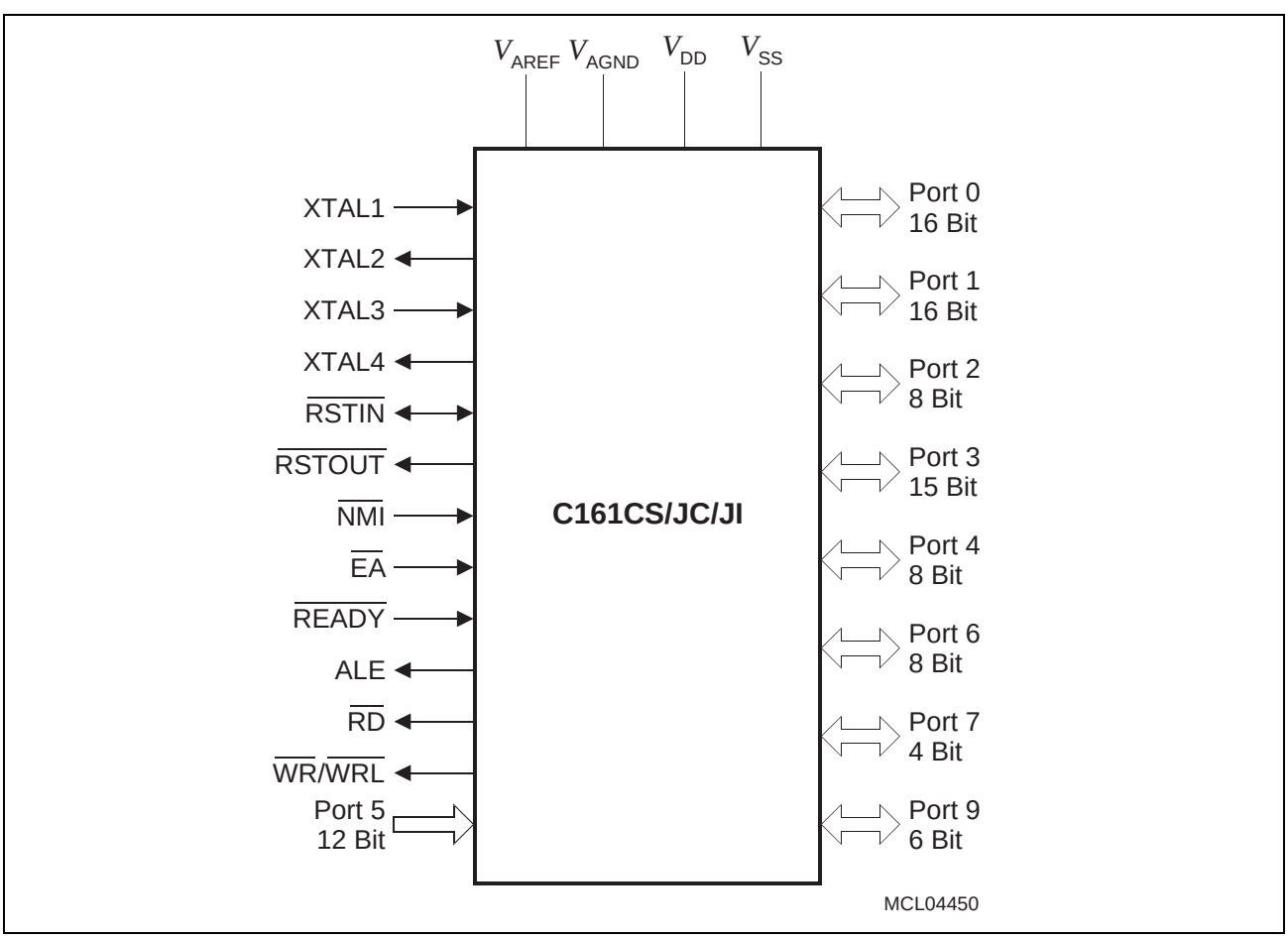



**Figure 1 Logic Symbol**

## Pin Configuration
(top view)

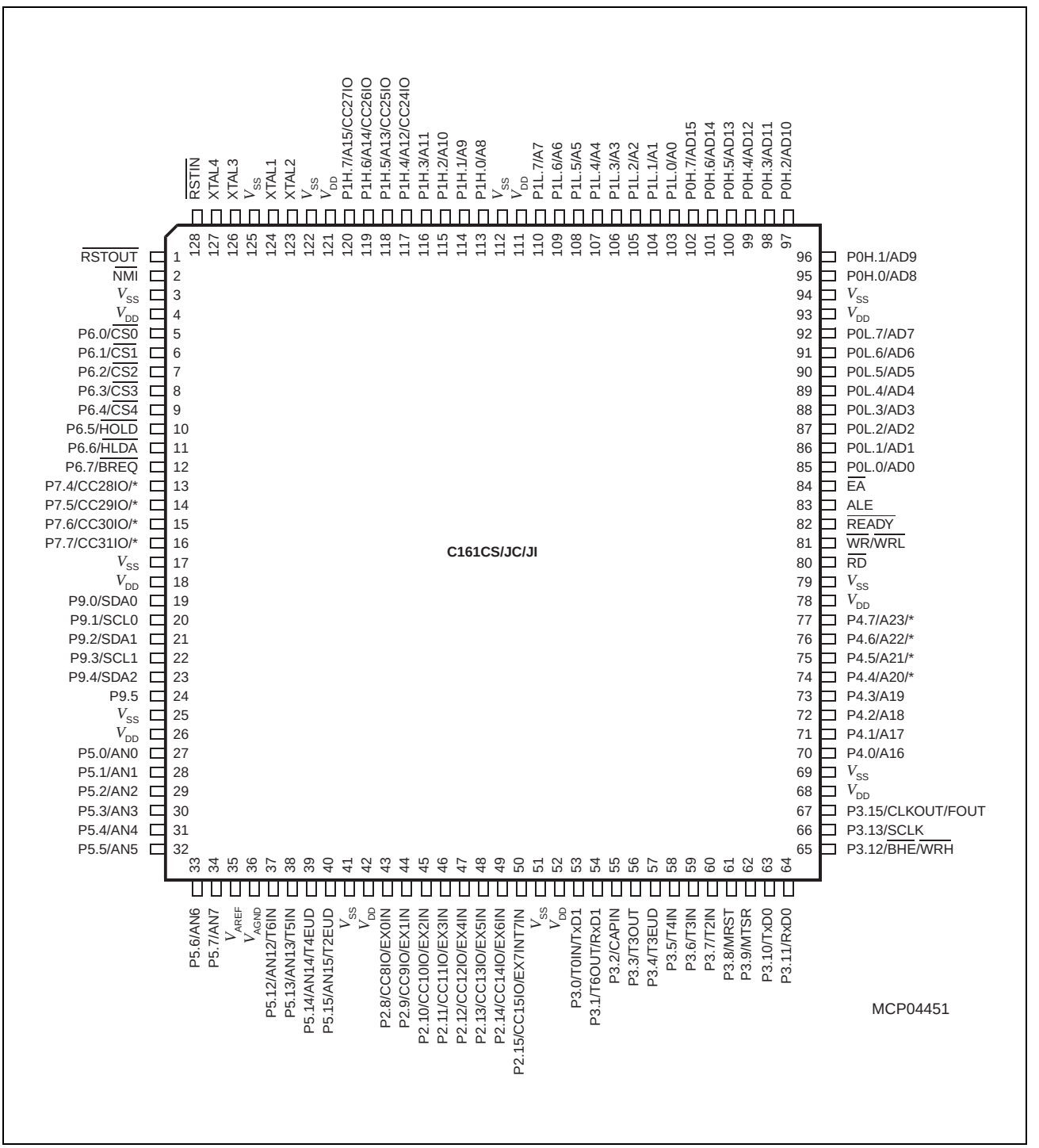



**Figure 2**

***)** The marked pins of Port 4 and Port 7 can have interface lines assigned to them (CAN interface in the **C161CS** and **C161JC**, SDLM interface in the **C161JC** and **C161JI**). **Table 2** on the pages below lists the possible assignments.

**Table 2 Pin Definitions and Functions**

| Symbol | Pin No. | Input Outp. | Function |
|---|---|---|---|
| $\overline{\text{RSTOUT}}$ | 1 | O | Internal Reset Indication Output. This pin is set to a low level when the part is executing either a hardware-, a software- or a watchdog timer reset. $\overline{\text{RSTOUT}}$ remains low until the EINIT (end of initialization) instruction is executed. |
| $\overline{\text{NMI}}$ | 2 | I | Non-Maskable Interrupt Input. A high to low transition at this pin causes the CPU to vector to the NMI trap routine. When the PWRDN (power down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the C161CS/JC/JI to go into power down mode. If $\overline{\text{NMI}}$ is high, when PWRDN is executed, the part will continue to run in normal mode. If not used, pin $\overline{\text{NMI}}$ should be pulled high externally. |
| **P6** | | IO | Port 6 is an 8-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. Port 6 outputs can be configured as push/pull or open drain drivers. The Port 6 pins also serve for alternate functions: |
| P6.0 | 5 | O | $\overline{\text{CS0}}$ Chip Select 0 Output |
| P6.1 | 6 | O | $\overline{\text{CS1}}$ Chip Select 1 Output |
| P6.2 | 7 | O | $\overline{\text{CS2}}$ Chip Select 2 Output |
| P6.3 | 8 | O | $\overline{\text{CS3}}$ Chip Select 3 Output |
| P6.4 | 9 | O | $\overline{\text{CS4}}$ Chip Select 4 Output |
| P6.5 | 10 | I | $\overline{\text{HOLD}}$ External Master Hold Request Input |
| P6.6 | 11 | I/O | $\overline{\text{HLDA}}$ Hold Acknowledge Output (master mode) or Input (slave mode) |
| P6.7 | 12 | O | $\overline{\text{BREQ}}$ Bus Request Output |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function |
|---|---|---|---|
| **P7** | | IO | Port 7 is a 4-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. Port 7 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 7 is selectable (TTL or special). Port 7 pins provide inputs/outputs for CAPCOM2 and serial interface lines.[1)] |
| P7.4 | 13 | I/O | CC28IO CAPCOM2: CC28 Capture Inp./Compare Outp., |
| | | I | CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**) |
| | | I | CAN2_RxD CAN 2 Receive Data Input, (**C161CS**) |
| | | O | SDL_TxD SDLM Transmit Data Output (**C161JC/JI**) |
| P7.5 | 14 | I/O | CC29IO CAPCOM2: CC29 Capture Inp./Compare Outp., |
| | | O | CAN1_TxD CAN 1 Transmit Data Output, (**C161CS/JC**) |
| | | O | CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**) |
| | | I | SDL_RxD SDLM Receive Data Input (**C161JC/JI**) |
| P7.6 | 15 | I/O | CC30IO CAPCOM2: CC30 Capture Inp./Compare Outp., |
| | | I | CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**) |
| | | I | CAN2_RxD CAN 2 Receive Data Input, (**C161CS**) |
| | | O | SDL_TxD SDLM Transmit Data Output (**C161JC/JI**) |
| P7.7 | 16 | I/O | CC31IO CAPCOM2: CC31 Capture Inp./Compare Outp., |
| | | O | CAN1_TxD CAN 1 Transmit Data Output, (**C161CS/JC**) |
| | | O | CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**) |
| | | I | SDL_RxD SDLM Receive Data Input (**C161JC/JI**) |
| **P9** | | IO | Port 9 is a 6-bit bidirectional open drain I/O port (provide external pullup resistors if required). It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. The following Port 9 pins also serve for alternate functions: |
| P9.0 | 19 | I/O | SDA0 IIC Bus Data Line 0 |
| P9.1 | 20 | I/O | SCL0 IIC Bus Clock Line 0 |
| P9.2 | 21 | I/O | SDA1 IIC Bus Data Line 1 |
| P9.3 | 22 | I/O | SCL1 IIC Bus Clock Line 1 |
| P9.4 | 23 | I/O | SDA2 IIC Bus Data Line 2 |
| P9.5 | 24 | – | – |
| | | | *Note: Port 9 pins can only tolerate positive overload currents (see **Table 9**).* |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function | | |
|---|---|---|---|---|---|
| **P5** | | I | Port 5 is a 12-bit input-only port with Schmitt-Trigger char. The pins of Port 5 also serve as analog input channels for the A/D converter, or they serve as timer inputs: | | |
| P5.0 | 27 | I | AN0 | | |
| P5.1 | 28 | I | AN1 | | |
| P5.2 | 29 | I | AN2 | | |
| P5.3 | 30 | I | AN3 | | |
| P5.4 | 31 | I | AN4 | | |
| P5.5 | 32 | I | AN5 | | |
| P5.6 | 33 | I | AN6 | | |
| P5.7 | 34 | I | AN7 | | |
| P5.12 | 37 | I | AN12, | T6IN | GPT2 Timer T6 Count Inp. |
| P5.13 | 38 | I | AN13, | T5IN | GPT2 Timer T5 Count Inp. |
| P5.14 | 39 | I | AN14, | T4EUD | GPT1 Timer T4 Ext. Up/Down Ctrl. Inp. |
| P5.15 | 40 | I | AN15, | T2EUD | GPT1 Timer T5 Ext. Up/Down Ctrl. Inp. |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function |
|---|---|---|---|
| **P2** | | IO | Port 2 is an 8-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. Port 2 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 2 is selectable (TTL or special).<br>The following Port 2 pins also serve for alternate functions: |
| P2.8 | 43 | I/O<br>I | CC8IO CAPCOM1: CC8 Capture Inp./Compare Output,<br>EX0IN Fast External Interrupt 0 Input |
| P2.9 | 44 | I/O<br>I | CC9IO CAPCOM1: CC9 Capture Inp./Compare Output,<br>EX1IN Fast External Interrupt 1 Input |
| P2.10 | 45 | I/O<br>I | CC10IO CAPCOM1: CC10 Capture Inp./Compare Outp.,<br>EX2IN Fast External Interrupt 2 Input |
| P2.11 | 46 | I/O<br>I | CC11IO CAPCOM1: CC11 Capture Inp./Compare Outp.,<br>EX3IN Fast External Interrupt 3 Input |
| P2.12 | 47 | I/O<br>I | CC12IO CAPCOM1: CC12 Capture Inp./Compare Outp.,<br>EX4IN Fast External Interrupt 4 Input |
| P2.13 | 48 | I/O<br>I | CC13IO CAPCOM1: CC13 Capture Inp./Compare Outp.,<br>EX5IN Fast External Interrupt 5 Input |
| P2.14 | 49 | I/O<br>I | CC14IO CAPCOM1: CC14 Capture Inp./Compare Outp.,<br>EX6IN Fast External Interrupt 6 Input |
| P2.15 | 50 | I/O<br>I<br>I | CC15IO CAPCOM1: CC15 Capture Inp./Compare Outp.,<br>EX7IN Fast External Interrupt 7 Input,<br>T7IN CAPCOM2: Timer T7 Count Input<br>*Note: During Sleep Mode a spike filter on the EXnIN interrupt inputs suppresses input pulses < 10 ns. Input pulses > 100 ns safely pass the filter.* |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function | |
|---|---|---|---|---|
| **P3** | | IO | Port 3 is a 15-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. Port 3 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 3 is selectable (TTL or special).<br>The following Port 3 pins also serve for alternate functions: | |
| P3.0 | 53 | I | T0IN | CAPCOM1 Timer T0 Count Input, |
| | | O | TxD1 | ASC1 Clock/Data Output (Async./Sync) |
| P3.1 | 54 | O | T6OUT | GPT2 Timer T6 Toggle Latch Output, |
| | | I/O | RxD1 | ASC1 Data Input (Async.) or Inp./Output (Sync.) |
| P3.2 | 55 | I | CAPIN | GPT2 Register CAPREL Capture Input |
| P3.3 | 56 | O | T3OUT | GPT1 Timer T3 Toggle Latch Output |
| P3.4 | 57 | I | T3EUD | GPT1 Timer T3 External Up/Down Control Input |
| P3.5 | 58 | I | T4IN | GPT1 Timer T4 Count/Gate/Reload/Capture Inp |
| P3.6 | 59 | I | T3IN | GPT1 Timer T3 Count/Gate Input |
| P3.7 | 60 | I | T2IN | GPT1 Timer T2 Count/Gate/Reload/Capture Inp |
| P3.8 | 61 | I/O | MRST | SSC Master-Receive/Slave-Transmit Inp./Outp. |
| P3.9 | 62 | I/O | MTSR | SSC Master-Transmit/Slave-Receive Outp./Inp. |
| P3.10 | 63 | O | TxD0 | ASC0 Clock/Data Output (Async./Sync.) |
| P3.11 | 64 | I/O | RxD0 | ASC0 Data Input (Async.) or Inp./Outp. (Sync.) |
| P3.12 | 65 | O | $\overline{\text{BHE}}$ | External Memory High Byte Enable Signal, |
| | | O | $\overline{\text{WRH}}$ | External Memory High Byte Write Strobe |
| P3.13 | 66 | I/O | SCLK | SSC Master Clock Output / Slave Clock Input. |
| P3.15 | 67 | O | CLKOUT | System Clock Output (= CPU Clock) |
| | | O | FOUT | Programmable Frequency Output |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function |
|---|---|---|---|
| **P4** | | IO | Port 4 is an 8-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. The Port 4 outputs can be configured as push/pull or open drain drivers. The input threshold of Port 4 is selectable (TTL or special).<br>Port 4 can be used to output the segment address lines and for serial interface lines:[1)] |
| P4.0 | 70 | O | A16 Least Significant Segment Address Line |
| P4.1 | 71 | O | A17 Segment Address Line |
| P4.2 | 72 | O | A18 Segment Address Line |
| P4.3 | 73 | O | A19 Segment Address Line |
| P4.4 | 74 | O | A20 Segment Address Line, |
| | | I | CAN2_RxD CAN 2 Receive Data Input, (**C161CS**) |
| | | I | SDL_RxD SDLM Receive Data Input (**C161JC/JI**) |
| P4.5 | 75 | O | A21 Segment Address Line, |
| | | I | CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**) |
| P4.6 | 76 | O | A22 Segment Address Line, |
| | | O | CAN1_TxD CAN 1 Transmit Data Output, (**C161CS/JC**) |
| | | O | CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**) |
| | | I | SDL_RxD SDLM Receive Data Input (**C161JC/JI**) |
| P4.7 | 77 | O | A23 Most Significant Segment Address Line, |
| | | I | CAN1_RxD CAN 1 Receive Data Input, (**C161CS/JC**) |
| | | O | CAN2_TxD CAN 2 Transmit Data Output, (**C161CS**) |
| | | I | CAN2_RxD CAN 2 Receive Data Input, (**C161CS**) |
| | | O | SDL_TxD SDLM Transmit Data Output (**C161JC/JI**) |
| $\overline{RD}$ | 80 | O | External Memory Read Strobe. $\overline{RD}$ is activated for every external instruction or data read access. |
| $\overline{WR}$/ $\overline{WRL}$ | 81 | O | External Memory Write Strobe. In $\overline{WR}$-mode this pin is activated for every external data write access. In $\overline{WRL}$-mode this pin is activated for low byte data write accesses on a 16-bit bus, and for every data write access on an 8-bit bus. See WRCFG in register SYSCON for mode selection. |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function |
|---|---|---|---|
| $\overline{\text{READY}}$ | 82 | I | Ready Input. When the Ready function is enabled, a high level at this pin during an external memory access will force the insertion of memory cycle time waitstates until the pin returns to a low level.<br>An internal pullup device will hold this pin high when nothing is driving it. |
| ALE | 83 | O | Address Latch Enable Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes. |
| $\overline{\text{EA}}$ | 84 | I | External Access Enable pin. A low level at this pin during and after Reset forces the C161CS/JC/JI to begin instruction execution out of external memory. A high level forces execution out of the internal program memory.<br>"ROMless" versions must have this pin tied to '0'. |
| **PORT0**<br>P0L.0-7<br><br>P0H.0-7 | <br>85-92<br><br>95-102 | IO | PORT0 consists of the two 8-bit bidirectional I/O ports P0L and P0H. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state.<br>In case of an external bus configuration, PORT0 serves as the address (A) and address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes.<br>**Demultiplexed bus modes:**<br>Data Path Width: 8-bit 16-bit<br>P0L.0 – P0L.7: D0 – D7 D0 - D7<br>P0H.0 – P0H.7: I/O D8 - D15<br>**Multiplexed bus modes:**<br>Data Path Width: 8-bit 16-bit<br>P0L.0 – P0L.7: AD0 – AD7 AD0 - AD7<br>P0H.0 – P0H.7: A8 - A15 AD8 - AD15<br>*Note: At the end of an external reset (EA = '0') PORT0 also inputs the configuration values.* |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function |
|---|---|---|---|
| **PORT1** | | IO | PORT1 consists of the two 8-bit bidirectional I/O ports P1L and P1H. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. PORT1 is used as the 16-bit address bus (A) in demultiplexed bus modes and also after switching from a demultiplexed bus mode to a multiplexed bus mode. |
| P1L.0-7 | 103-110 | | |
| P1H.0-7 | 113-120 | | |
| | | | The following PORT1 pins also serve for alternate functions: |
| P1H.4 | 117 | I/O | CC24IO CAPCOM2: CC24 Capture Inp./Compare Outp. |
| P1H.5 | 118 | I/O | CC25IO CAPCOM2: CC25 Capture Inp./Compare Outp. |
| P1H.6 | 119 | I/O | CC26IO CAPCOM2: CC26 Capture Inp./Compare Outp. |
| P1H.7 | 120 | I/O | CC27IO CAPCOM2: CC27 Capture Inp./Compare Outp. |
| XTAL2 | 123 | O | XTAL2: Output of the oscillator amplifier circuit. |
| XTAL1 | 124 | I | XTAL1: Input to the oscillator amplifier and input to the internal clock generator<br>To clock the device from an external source, drive XTAL1, while leaving XTAL2 unconnected. Minimum and maximum high/low and rise/fall times specified in the AC Characteristics must be observed. |
| XTAL3 | 126 | I | XTAL3: Input to the 32-kHz oscillator amplifier and input to the internal clock generator |
| XTAL4 | 127 | O | XTAL4: Output of the oscillator amplifier circuit.<br>To clock the device from an external source, drive XTAL3, while leaving XTAL4 unconnected. Minimum and maximum high/low and rise/fall times specified in the AC Characteristics must be observed. |

**Table 2 Pin Definitions and Functions** (cont'd)

| Symbol | Pin No. | Input Outp. | Function |
|---|---|---|---|
| $\overline{\text{RSTIN}}$ | 128 | I/O | Reset Input with Schmitt-Trigger characteristics. A low level at this pin while the oscillator is running resets the C161CS/JC/JI. An internal pullup resistor permits power-on reset using only a capacitor connected to $V_{SS}$.<br>A spike filter suppresses input pulses < 10 ns. Input pulses > 100 ns safely pass the filter. The minimum duration for a safe recognition should be 100 ns + 2 CPU clock cycles.<br>In bidirectional reset mode (enabled by setting bit BDRSTEN in register SYSCON) the $\overline{\text{RSTIN}}$ line is internally pulled low for the duration of the internal reset sequence upon any reset (HW, SW, WDT). See note below this table.<br>*Note: To let the reset configuration of PORT0 settle and to let the PLL lock a reset duration of ca. 1 ms is recommended.* |
| $V_{AREF}$ | 35 | – | Reference voltage for the A/D converter. |
| $V_{AGND}$ | 36 | – | Reference ground for the A/D converter. |
| $V_{DD}$ | 4, 18, 26[2], 42, 52, 68, 78, 93, 111, 121 | – | Digital Supply Voltage:<br>+5 V during normal operation and idle mode.<br>≥ 2.5 V during power down mode if RTC is off<br>≥ 2.7 V during power down mode if RTC is running |
| $V_{SS}$ | 3, 17, 25[2], 41, 51, 69, 79, 94, 112, 122, 125 | – | Digital Ground. |

1) The CAN and/or SDLM interface lines are assigned to ports P4 and P7 under software control. Within the CAN module or SDLM several assignments can be selected.

2) Supply pins 25 and 26 feed the Analog/Digital Converter and should be decoupled separately.

*Note: The following behavioural differences must be observed when the bidirectional reset is active:*

- Bit BDRSTEN in register SYSCON cannot be changed after EINIT and is cleared automatically after a reset.
- The reset indication flags always indicate a long hardware reset.
- The PORT0 configuration is treated as if it were a hardware reset. In particular, the bootstrap loader may be activated when P0L.4 is low.
- Pin $\overline{\text{RSTIN}}$ may only be connected to external reset devices with an open drain output driver.
- A short hardware reset is extended to the duration of the internal reset sequence.

## Functional Description

The architecture of the C161CS/JC/JI combines advantages of both RISC and CISC processors and of advanced peripheral subsystems in a very well-balanced way. In addition the on-chip memory blocks allow the design of compact systems with maximum performance.
The following block diagram gives an overview of the different on-chip components and of the advanced, high bandwidth internal bus structure of the C161CS/JC/JI.

*Note: All time specifications refer to a CPU clock of 25 MHz (see definition in the AC Characteristics section).*

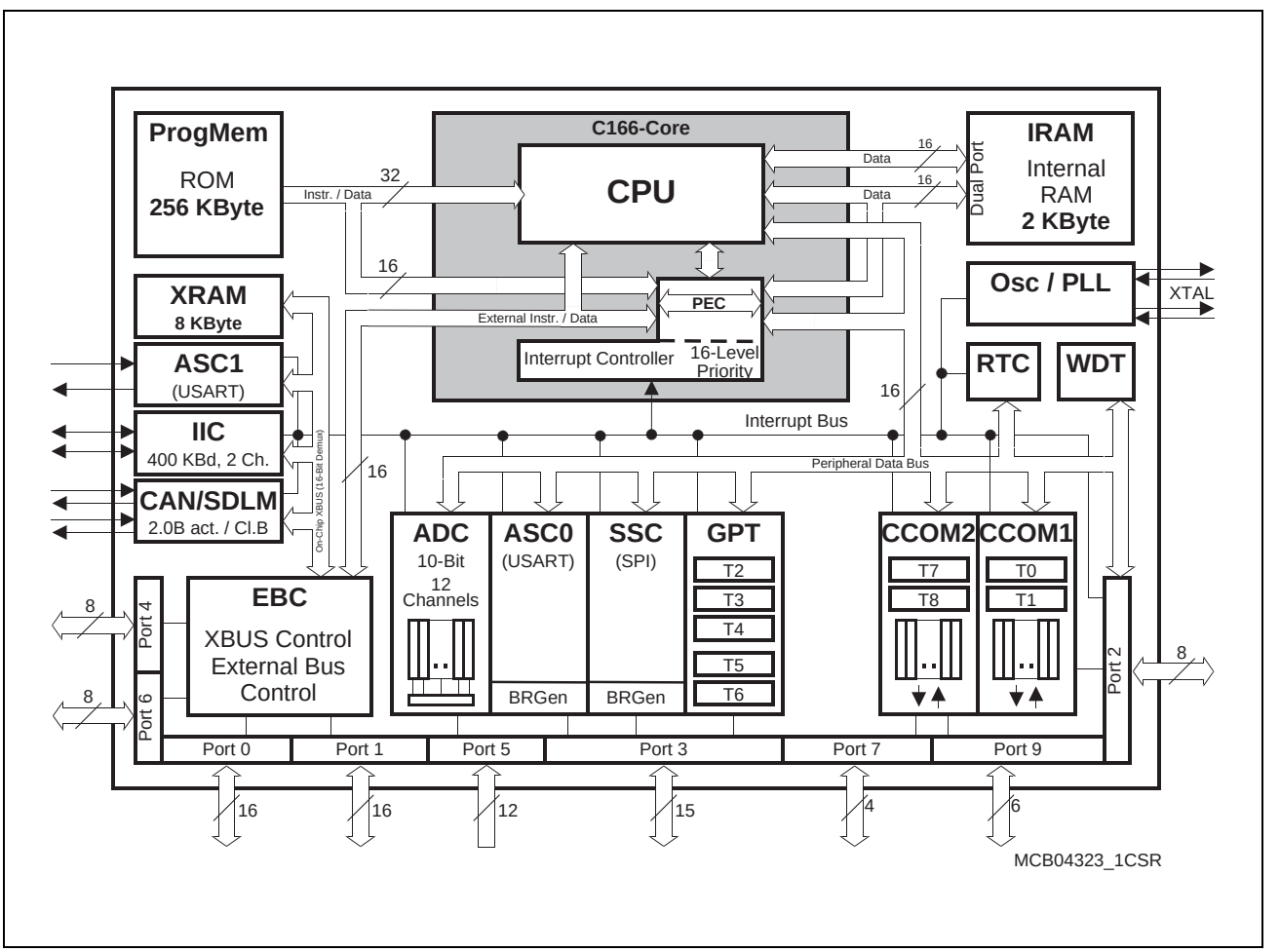



**Figure 3 Block Diagram**

The program memory, the internal RAM (IRAM) and the set of generic peripherals are connected to the CPU via separate buses. A fourth bus, the XBUS, connects external resources as well as additional on-chip resoures, the X-Peripherals (see **Figure 3**).

The XBUS resources (XRAM, CAN, SDLM, IIC, ASC1) of the C161CS/JC/JI can be enabled during initialization by setting the general X-Peripheral enable bit XPEN (SYSCON.2).
If the X-Peripherals remain disabled they consume neither address space nor port pins.

## Memory Organization

The memory space of the C161CS/JC/JI is configured in a Von Neumann architecture which means that code memory, data memory, registers and I/O ports are organized within the same linear address space which includes 16 MBytes. The entire memory space can be accessed bytewise or wordwise. Particular portions of the on-chip memory have additionally been made directly bitaddressable.

The C161CS/JC/JI incorporates 256 KBytes of on-chip mask-programmable ROM for code or constant data. The lower 32 KBytes of the on-chip ROM can be mapped either to segment 0 or segment 1.

2 KBytes of on-chip Internal RAM (IRAM) are provided as a storage for user defined variables, for the system stack, general purpose register banks and even for code. A register bank can consist of up to 16 wordwide (R0 to R15) and/or bytewide (RL0, RH0, …, RL7, RH7) so-called General Purpose Registers (GPRs).

1024 bytes (2 $\times$ 512 bytes) of the address space are reserved for the Special Function Register areas (SFR space and ESFR space). SFRs are wordwide registers which are used for controlling and monitoring functions of the different on-chip units. Unused SFR addresses are reserved for future members of the C166 Family.

8 KBytes of on-chip Extension RAM (XRAM) are provided to store user data, user stacks, or code. The XRAM is accessed like external memory and therefore cannot be used for the system stack or for register banks and is not bitaddressable. The XRAM permits 16-bit accesses with maximum speed.

In order to meet the needs of designs where more memory is required than is provided on chip, up to 16 MBytes of external RAM and/or ROM can be connected to the microcontroller.

## External Bus Controller

All of the external memory accesses are performed by a particular on-chip External Bus Controller (EBC). It can be programmed either to Single Chip Mode when no external memory is required, or to one of four different external memory access modes, which are as follows:

– 16-/18-/20-/24-bit Addresses, 16-bit Data, Demultiplexed
– 16-/18-/20-/24-bit Addresses, 16-bit Data, Multiplexed
– 16-/18-/20-/24-bit Addresses, 8-bit Data, Multiplexed
– 16-/18-/20-/24-bit Addresses, 8-bit Data, Demultiplexed

In the demultiplexed bus modes, addresses are output on PORT1 and data is input/ output on PORT0 or P0L, respectively. In the multiplexed bus modes both addresses and data use PORT0 for input/output.

Important timing characteristics of the external bus interface (Memory Cycle Time, Memory Tri-State Time, Length of ALE and Read Write Delay) have been made programmable to allow the user the adaption of a wide range of different types of memories and external peripherals.
In addition, up to 4 independent address windows may be defined (via register pairs ADDRSELx / BUSCONx) which control the access to different resources with different bus characteristics. These address windows are arranged hierarchically where BUSCON4 overrides BUSCON3 and BUSCON2 overrides BUSCON1. All accesses to locations not covered by these 4 address windows are controlled by BUSCON0.

Up to 5 external $\overline{CS}$ signals (4 windows plus default) can be generated in order to save external glue logic. The C161CS/JC/JI offers the possibility to switch the $\overline{CS}$ outputs to an unlatched mode. In this mode the internal filter logic is switched off and the $\overline{CS}$ signals are directly generated from the address. The unlatched $\overline{CS}$ mode is enabled by setting CSCFG (SYSCON.6).

Access to very slow memories or memories with varying access times is supported via a particular ‘Ready’ function.

A $\overline{HOLD}$/$\overline{HLDA}$ protocol is available for bus arbitration and allows to share external resources with other bus masters. The bus arbitration is enabled by setting bit HLDEN in register PSW. After setting HLDEN once, pins P6.7 … P6.5 ($\overline{BREQ}$, $\overline{HLDA}$, $\overline{HOLD}$) are automatically controlled by the EBC. In Master Mode (default after reset) the $\overline{HLDA}$ pin is an output. By setting bit DP6.7 to ‘1’ the Slave Mode is selected where pin $\overline{HLDA}$ is switched to input. This allows to directly connect the slave controller to another master controller without glue logic.

For applications which require less than 16 MBytes of external memory space, this address space can be restricted to 1 MByte, 256 KByte, or to 64 KByte. In this case Port 4 outputs four, two, or no address lines at all. It outputs all 8 address lines, if an address space of 16 MBytes is used.

*Note: When one or both of the on-chip CAN Modules or the SDLM are used with the interface lines assigned to Port 4, the interface lines override the segment address lines and the segment address output on Port 4 is therefore limited to 6/4 bits i.e. address lines A21/A19 … A16. $\overline{CS}$ lines can be used to increase the total amount of addressable external memory.*

### Central Processing Unit (CPU)

The main core of the CPU consists of a 4-stage instruction pipeline, a 16-bit arithmetic and logic unit (ALU) and dedicated SFRs. Additional hardware has been spent for a separate multiply and divide unit, a bit-mask generator and a barrel shifter.

Based on these hardware provisions, most of the C161CS/JC/JI's instructions can be executed in just one machine cycle which requires 80 ns at 25 MHz CPU clock. For example, shift and rotate instructions are always processed during one machine cycle independent of the number of bits to be shifted. All multiple-cycle instructions have been optimized so that they can be executed very fast as well: branches in 2 cycles, a 16 × 16 bit multiplication in 5 cycles and a 32-/16-bit division in 10 cycles. Another pipeline optimization, the so-called 'Jump Cache', allows reducing the execution time of repeatedly performed jumps in a loop from 2 cycles to 1 cycle.

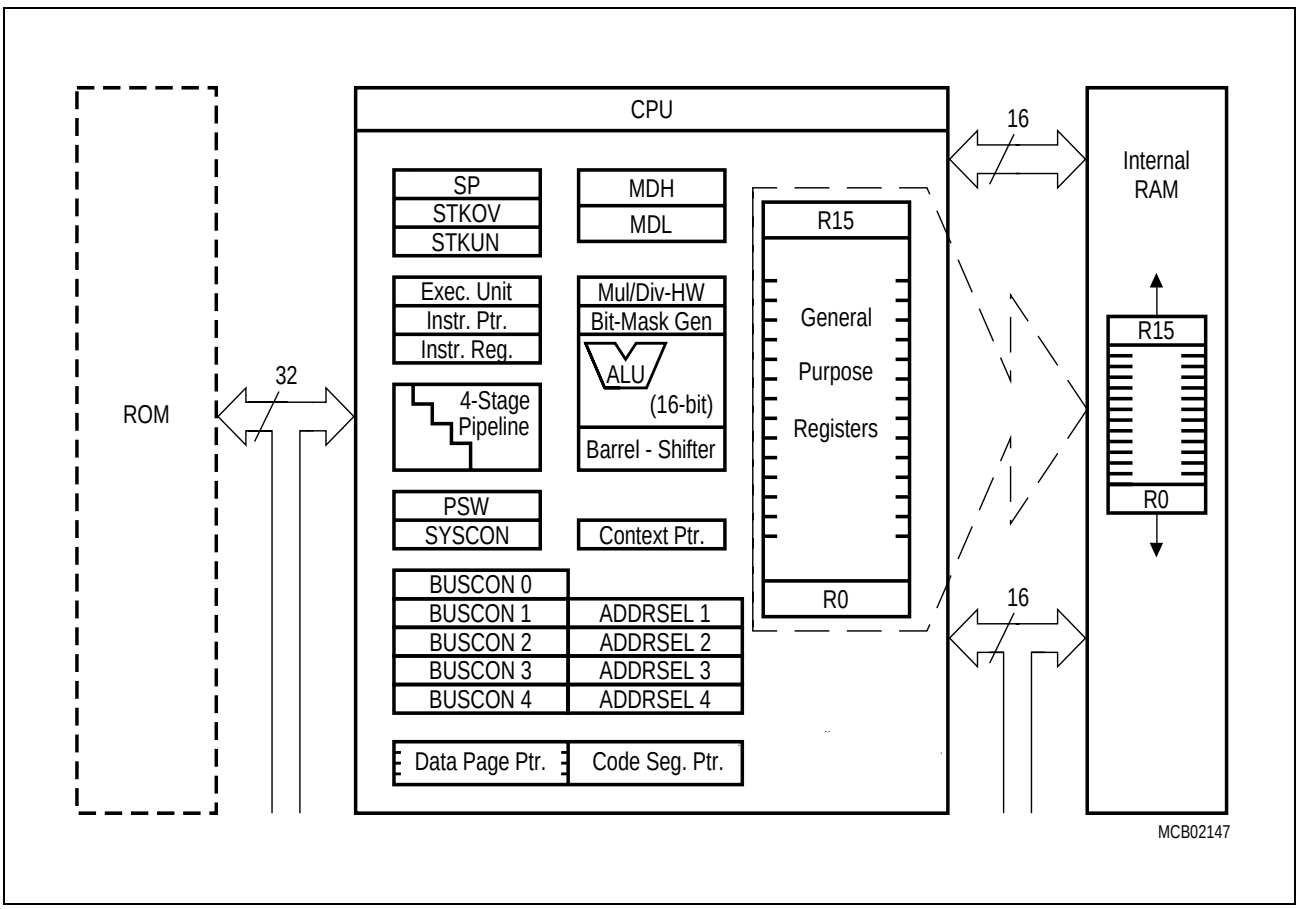



**Figure 4 CPU Block Diagram**

The CPU has a register context consisting of up to 16 wordwide GPRs at its disposal. These 16 GPRs are physically allocated within the on-chip RAM area. A Context Pointer (CP) register determines the base address of the active register bank to be accessed by the CPU at any time. The number of register banks is only restricted by the available internal RAM space. For easy parameter passing, a register bank may overlap others.

A system stack of up to 1024 words is provided as a storage for temporary data. The system stack is allocated in the on-chip RAM area, and it is accessed by the CPU via the stack pointer (SP) register. Two separate SFRs, STKOV and STKUN, are implicitly compared against the stack pointer value upon each stack access for the detection of a stack overflow or underflow.

The high performance offered by the hardware implementation of the CPU can efficiently be utilized by a programmer via the highly efficient C161CS/JC/JI instruction set which includes the following instruction classes:

– Arithmetic Instructions
– Logical Instructions
– Boolean Bit Manipulation Instructions
– Compare and Loop Control Instructions
– Shift and Rotate Instructions
– Prioritize Instruction
– Data Movement Instructions
– System Stack Instructions
– Jump and Call Instructions
– Return Instructions
– System Control Instructions
– Miscellaneous Instructions

The basic instruction length is either 2 or 4 bytes. Possible operand types are bits, bytes and words. A variety of direct, indirect or immediate addressing modes are provided to specify the required operands.

### Interrupt System

With an interrupt response time within a range from just 5 to 12 CPU clocks (in case of internal program execution), the C161CS/JC/JI is capable of reacting very fast to the occurrence of non-deterministic events.

The architecture of the C161CS/JC/JI supports several mechanisms for fast and flexible response to service requests that can be generated from various sources internal or external to the microcontroller. Any of these interrupt requests can be programmed to being serviced by the Interrupt Controller or by the Peripheral Event Controller (PEC).

In contrast to a standard interrupt service where the current program execution is suspended and a branch to the interrupt vector table is performed, just one cycle is ‘stolen’ from the current CPU activity to perform a PEC service. A PEC service implies a single byte or word data transfer between any two memory locations with an additional increment of either the PEC source or the destination pointer. An individual PEC transfer counter is implicity decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited, for example, for supporting the transmission or reception of blocks of data. The C161CS/JC/JI has 8 PEC channels each of which offers such fast interrupt-driven data transfer capabilities.

A separate control register which contains an interrupt request flag, an interrupt enable flag and an interrupt priority bitfield exists for each of the possible interrupt sources. Via its related register, each source can be programmed to one of sixteen interrupt priority levels. Once having been accepted by the CPU, an interrupt service can only be interrupted by a higher prioritized service request. For the standard interrupt processing, each of the possible interrupt sources has a dedicated vector location.

Fast external interrupt inputs are provided to service external interrupts with high precision requirements. These fast interrupt inputs feature programmable edge detection (rising edge, falling edge or both edges).

Software interrupts are supported by means of the ‘TRAP’ instruction in combination with an individual trap (interrupt) number.

**Table 3** shows all of the possible C161CS/JC/JI interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers.

*Note: Interrupt nodes which are not used by associated peripherals, may be used to generate software controlled interrupt requests by setting the respective interrupt request bit (xIR).*

**Table 3 C161CS/JC/JI Interrupt Nodes**

| Source of Interrupt or PEC Service Request | Request Flag | Enable Flag | Interrupt Vector | Vector Location | Trap Number |
|---|---|---|---|---|---|
| CAPCOM Register 0 | CC0IR | CC0IE | CC0INT | $00'0040_H$ | $10_H$ |
| CAPCOM Register 1 | CC1IR | CC1IE | CC1INT | $00'0044_H$ | $11_H$ |
| CAPCOM Register 2 | CC2IR | CC2IE | CC2INT | $00'0048_H$ | $12_H$ |
| CAPCOM Register 3 | CC3IR | CC3IE | CC3INT | $00'004C_H$ | $13_H$ |
| CAPCOM Register 4 | CC4IR | CC4IE | CC4INT | $00'0050_H$ | $14_H$ |
| CAPCOM Register 5 | CC5IR | CC5IE | CC5INT | $00'0054_H$ | $15_H$ |
| CAPCOM Register 6 | CC6IR | CC6IE | CC6INT | $00'0058_H$ | $16_H$ |
| CAPCOM Register 7 | CC7IR | CC7IE | CC7INT | $00'005C_H$ | $17_H$ |
| CAPCOM Register 8 | CC8IR | CC8IE | CC8INT | $00'0060_H$ | $18_H$ |
| CAPCOM Register 9 | CC9IR | CC9IE | CC9INT | $00'0064_H$ | $19_H$ |
| CAPCOM Register 10 | CC10IR | CC10IE | CC10INT | $00'0068_H$ | $1A_H$ |
| CAPCOM Register 11 | CC11IR | CC11IE | CC11INT | $00'006C_H$ | $1B_H$ |
| CAPCOM Register 12 | CC12IR | CC12IE | CC12INT | $00'0070_H$ | $1C_H$ |
| CAPCOM Register 13 | CC13IR | CC13IE | CC13INT | $00'0074_H$ | $1D_H$ |
| CAPCOM Register 14 | CC14IR | CC14IE | CC14INT | $00'0078_H$ | $1E_H$ |
| CAPCOM Register 15 | CC15IR | CC15IE | CC15INT | $00'007C_H$ | $1F_H$ |
| CAPCOM Register 16 | CC16IR | CC16IE | CC16INT | $00'00C0_H$ | $30_H$ |
| CAPCOM Register 17 | CC17IR | CC17IE | CC17INT | $00'00C4_H$ | $31_H$ |
| CAPCOM Register 18 | CC18IR | CC18IE | CC18INT | $00'00C8_H$ | $32_H$ |
| CAPCOM Register 19 | CC19IR | CC19IE | CC19INT | $00'00CC_H$ | $33_H$ |
| CAPCOM Register 20 | CC20IR | CC20IE | CC20INT | $00'00D0_H$ | $34_H$ |
| CAPCOM Register 21 | CC21IR | CC21IE | CC21INT | $00'00D4_H$ | $35_H$ |
| CAPCOM Register 22 | CC22IR | CC22IE | CC22INT | $00'00D8_H$ | $36_H$ |
| CAPCOM Register 23 | CC23IR | CC23IE | CC23INT | $00'00DC_H$ | $37_H$ |
| CAPCOM Register 24 | CC24IR | CC24IE | CC24INT | $00'00E0_H$ | $38_H$ |
| CAPCOM Register 25 | CC25IR | CC25IE | CC25INT | $00'00E4_H$ | $39_H$ |
| CAPCOM Register 26 | CC26IR | CC26IE | CC26INT | $00'00E8_H$ | $3A_H$ |
| CAPCOM Register 27 | CC27IR | CC27IE | CC27INT | $00'00EC_H$ | $3B_H$ |
| CAPCOM Register 28 | CC28IR | CC28IE | CC28INT | $00'00E0_H$ | $3C_H$ |
| CAPCOM Register 29 | CC29IR | CC29IE | CC29INT | $00'0110_H$ | $44_H$ |

**Table 3 C161CS/JC/JI Interrupt Nodes** (cont'd)

| Source of Interrupt or PEC Service Request | Request Flag | Enable Flag | Interrupt Vector | Vector Location | Trap Number |
|---|---|---|---|---|---|
| CAPCOM Register 30 | CC30IR | CC30IE | CC30INT | $00'0114_H$ | $45_H$ |
| CAPCOM Register 31 | CC31IR | CC31IE | CC31INT | $00'0118_H$ | $46_H$ |
| CAPCOM Timer 0 | T0IR | T0IE | T0INT | $00'0080_H$ | $20_H$ |
| CAPCOM Timer 1 | T1IR | T1IE | T1INT | $00'0084_H$ | $21_H$ |
| CAPCOM Timer 7 | T7IR | T7IE | T7INT | $00'00F4_H$ | $3D_H$ |
| CAPCOM Timer 8 | T8IR | T8IE | T8INT | $00'00F8_H$ | $3E_H$ |
| GPT1 Timer 2 | T2IR | T2IE | T2INT | $00'0088_H$ | $22_H$ |
| GPT1 Timer 3 | T3IR | T3IE | T3INT | $00'008C_H$ | $23_H$ |
| GPT1 Timer 4 | T4IR | T4IE | T4INT | $00'0090_H$ | $24_H$ |
| GPT2 Timer 5 | T5IR | T5IE | T5INT | $00'0094_H$ | $25_H$ |
| GPT2 Timer 6 | T6IR | T6IE | T6INT | $00'0098_H$ | $26_H$ |
| GPT2 CAPREL Reg. | CRIR | CRIE | CRINT | $00'009C_H$ | $27_H$ |
| A/D Conversion Compl. | ADCIR | ADCIE | ADCINT | $00'00A0_H$ | $28_H$ |
| A/D Overrun Error | ADEIR | ADEIE | ADEINT | $00'00A4_H$ | $29_H$ |
| ASC0 Transmit | S0TIR | S0TIE | S0TINT | $00'00A8_H$ | $2A_H$ |
| ASC0 Transmit Buffer | S0TBIR | S0TBIE | S0TBINT | $00'011C_H$ | $47_H$ |
| ASC0 Receive | S0RIR | S0RIE | S0RINT | $00'00AC_H$ | $2B_H$ |
| ASC0 Error | S0EIR | S0EIE | S0EINT | $00'00B0_H$ | $2C_H$ |
| SSC Transmit | SCTIR | SCTIE | SCTINT | $00'00B4_H$ | $2D_H$ |
| SSC Receive | SCRIR | SCRIE | SCRINT | $00'00B8_H$ | $2E_H$ |
| SSC Error | SCEIR | SCEIE | SCEINT | $00'00BC_H$ | $2F_H$ |
| IIC Data Transfer Event | XP0IR | XP0IE | XP0INT | $00'0100_H$ | $40_H$ |
| IIC Protocol Event | XP1IR | XP1IE | XP1INT | $00'0104_H$ | $41_H$ |
| CAN1 (**C161CS/JC**) | XP2IR | XP2IE | XP2INT | $00'0108_H$ | $42_H$ |
| PLL/OWD and RTC | XP3IR | XP3IE | XP3INT | $00'010C_H$ | $43_H$ |
| ASC1 Transmit | XP4IR | XP4IE | XP4INT | $00'0120_H$ | $48_H$ |
| ASC1 Receive | XP5IR | XP5IE | XP5INT | $00'0124_H$ | $49_H$ |
| ASC1 Error | XP6IR | XP6IE | XP6INT | $00'0128_H$ | $4A_H$ |
| CAN2 (**C161CS**) or SDLM (**C161JC/JI**) | XP7IR | XP7IE | XP7INT | $00'012C_H$ | $4B_H$ |

The C161CS/JC/JI also provides an excellent mechanism to identify and to process exceptions or error conditions that arise during run-time, so-called ‘Hardware Traps’. Hardware traps cause immediate non-maskable system reaction which is similar to a standard interrupt service (branching to a dedicated vector table location). The occurence of a hardware trap is additionally signified by an individual bit in the trap flag register (TFR). Except when another higher prioritized trap service is in progress, a hardware trap will interrupt any actual program execution. In turn, hardware trap services can normally not be interrupted by standard or PEC interrupts.

**Table 4** shows all of the possible exceptions or error conditions that can arise during run-time:

**Table 4 Hardware Trap Summary**

| Exception Condition | Trap Flag | Trap Vector | Vector Location | Trap Number | Trap Priority |
|---|---|---|---|---|---|
| Reset Functions: | – | | | | |
| Hardware Reset | | RESET | $00'0000_H$ | $00_H$ | III |
| Software Reset | | RESET | $00'0000_H$ | $00_H$ | III |
| W-dog Timer Overflow | | RESET | $00'0000_H$ | $00_H$ | III |
| Class A Hardware Traps: | | | | | |
| Non-Maskable Interrupt | NMI | NMITRAP | $00'0008_H$ | $02_H$ | II |
| Stack Overflow | STKOF | STOTRAP | $00'0010_H$ | $04_H$ | II |
| Stack Underflow | STKUF | STUTRAP | $00'0018_H$ | $06_H$ | II |
| Class B Hardware Traps: | | | | | |
| Undefined Opcode | UNDOPC | BTRAP | $00'0028_H$ | $0A_H$ | I |
| Protected Instruction Fault | PRTFLT | BTRAP | $00'0028_H$ | $0A_H$ | I |
| Illegal Word Operand Access | ILLOPA | BTRAP | $00'0028_H$ | $0A_H$ | I |
| Illegal Instruction Access | ILLINA | BTRAP | $00'0028_H$ | $0A_H$ | I |
| Illegal External Bus Access | ILLBUS | BTRAP | $00'0028_H$ | $0A_H$ | I |
| Reserved | – | – | $[2C_H – 3C_H]$ | $[0B_H – 0F_H]$ | – |
| Software Traps TRAP Instruction | – | – | Any $[00'0000_H – 00'01FC_H]$ in steps of $4_H$ | Any $[00_H – 7F_H]$ | Current CPU Priority |

### Capture/Compare (CAPCOM) Units

The CAPCOM units support generation and control of timing sequences on up to 32 channels with a maximum resolution of 16 TCL. The CAPCOM units are typically used to handle high speed I/O tasks such as pulse and waveform generation, pulse width modulation (PMW), Digital to Analog (D/A) conversion, software timing, or time recording relative to external events.

Four 16-bit timers (T0/T1, T7/T8) with reload registers provide two independent time bases for the capture/compare register array.

The input clock for the timers is programmable to several prescaled values of the internal system clock, or may be derived from an overflow/underflow of timer T6 in module GPT2. This provides a wide range of variation for the timer period and resolution and allows precise adjustments to the application specific requirements. In addition, external count inputs for CAPCOM timers T0 and T7 allow event scheduling for the capture/compare registers relative to external events.

Both of the two capture/compare register arrays contain 16 dual purpose capture/compare registers, each of which may be individually allocated to either CAPCOM timer T0 or T1 (T7 or T8, respectively), and programmed for capture or compare function. Eight registers of each module have one port pin associated with it which serves as an input pin for triggering the capture function, or as an output pin to indicate the occurrence of a compare event.

When a capture/compare register has been selected for capture mode, the current contents of the allocated timer will be latched (‘captured’) into the capture/compare register in response to an external event at the port pin which is associated with this register. In addition, a specific interrupt request for this capture/compare register is generated. Either a positive, a negative, or both a positive and a negative external signal transition at the pin can be selected as the triggering event.

The contents of all registers which have been selected for one of the five compare modes are continuously compared with the contents of the allocated timers.

**Table 5 Compare Modes (CAPCOM)**

| Compare Modes | Function |
|---|---|
| Mode 0 | Interrupt-only compare mode;<br>several compare interrupts per timer period are possible |
| Mode 1 | Pin toggles on each compare match;<br>several compare events per timer period are possible |
| Mode 2 | Interrupt-only compare mode;<br>only one compare interrupt per timer period is generated |
| Mode 3 | Pin set ‘1’ on match; pin reset ‘0’ on compare time overflow;<br>only one compare event per timer period is generated |

When a match occurs between the timer value and the value in a capture/compare register, specific actions will be taken based on the selected compare mode.

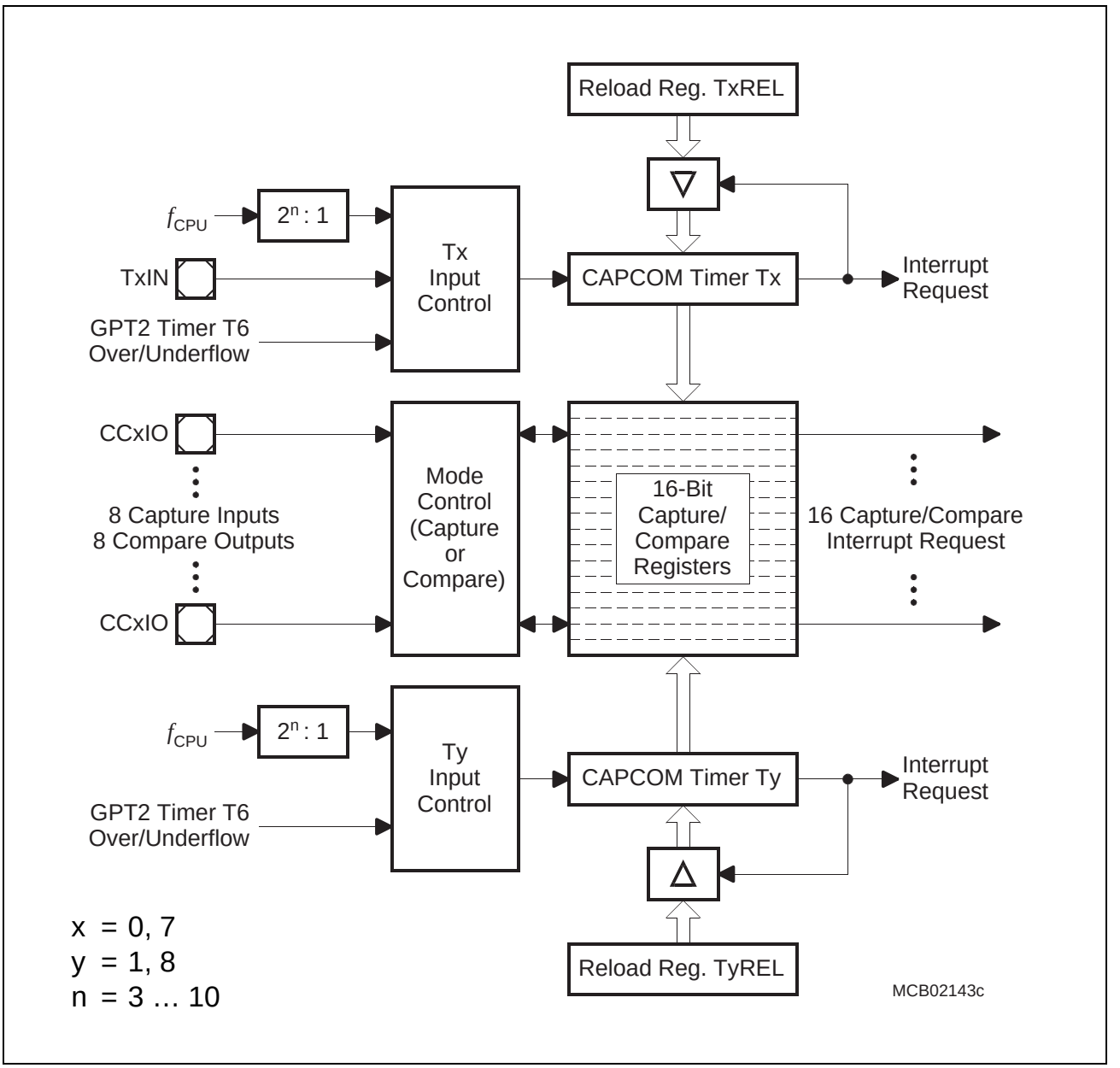



**Figure 5 CAPCOM Unit Block Diagram**

## General Purpose Timer (GPT) Unit

The GPT unit represents a very flexible multifunctional timer/counter structure which may be used for many different time related tasks such as event timing and counting, pulse width and duty cycle measurements, pulse generation, or pulse multiplication.

The GPT unit incorporates five 16-bit timers which are organized in two separate modules, GPT1 and GPT2. Each timer in each module may operate independently in a number of different modes, or may be concatenated with another timer of the same module.

Each of the three timers T2, T3, T4 of **module GPT1** can be configured individually for one of four basic modes of operation, which are Timer, Gated Timer, Counter, and Incremental Interface Mode. In Timer Mode, the input clock for a timer is derived from the CPU clock, divided by a programmable prescaler, while Counter Mode allows a timer to be clocked in reference to external events.
Pulse width or duty cycle measurement is supported in Gated Timer Mode, where the operation of a timer is controlled by the ‘gate’ level on an external input pin. For these purposes, each timer has one associated port pin (TxIN) which serves as gate or clock input. The maximum resolution of the timers in module GPT1 is 16 TCL.

The count direction (up/down) for each timer is programmable by software or may additionally be altered dynamically by an external signal on a port pin (TxEUD) to facilitate e.g. position tracking.

In Incremental Interface Mode the GPT1 timers (T2, T3, T4) can be directly connected to the incremental position sensor signals A and B via their respective inputs TxIN and TxEUD. Direction and count signals are internally derived from these two input signals, so the contents of the respective timer Tx corresponds to the sensor position. The third position sensor signal TOP0 can be connected to an interrupt input.

Timer T3 has an output toggle latch (T3OTL) which changes its state on each timer over-flow/underflow. The state of this latch may be output on pin T3OUT e.g. for time out monitoring of external hardware components, or may be used internally to clock timers T2 and T4 for measuring long time periods with high resolution.

In addition to their basic operating modes, timers T2 and T4 may be configured as reload or capture registers for timer T3. When used as capture or reload registers, timers T2 and T4 are stopped. The contents of timer T3 is captured into T2 or T4 in response to a signal at their associated input pins (TxIN). Timer T3 is reloaded with the contents of T2 or T4 triggered either by an external signal or by a selectable state transition of its toggle latch T3OTL. When both T2 and T4 are configured to alternately reload T3 on opposite state transitions of T3OTL with the low and high times of a PWM signal, this signal can be constantly generated without software intervention.

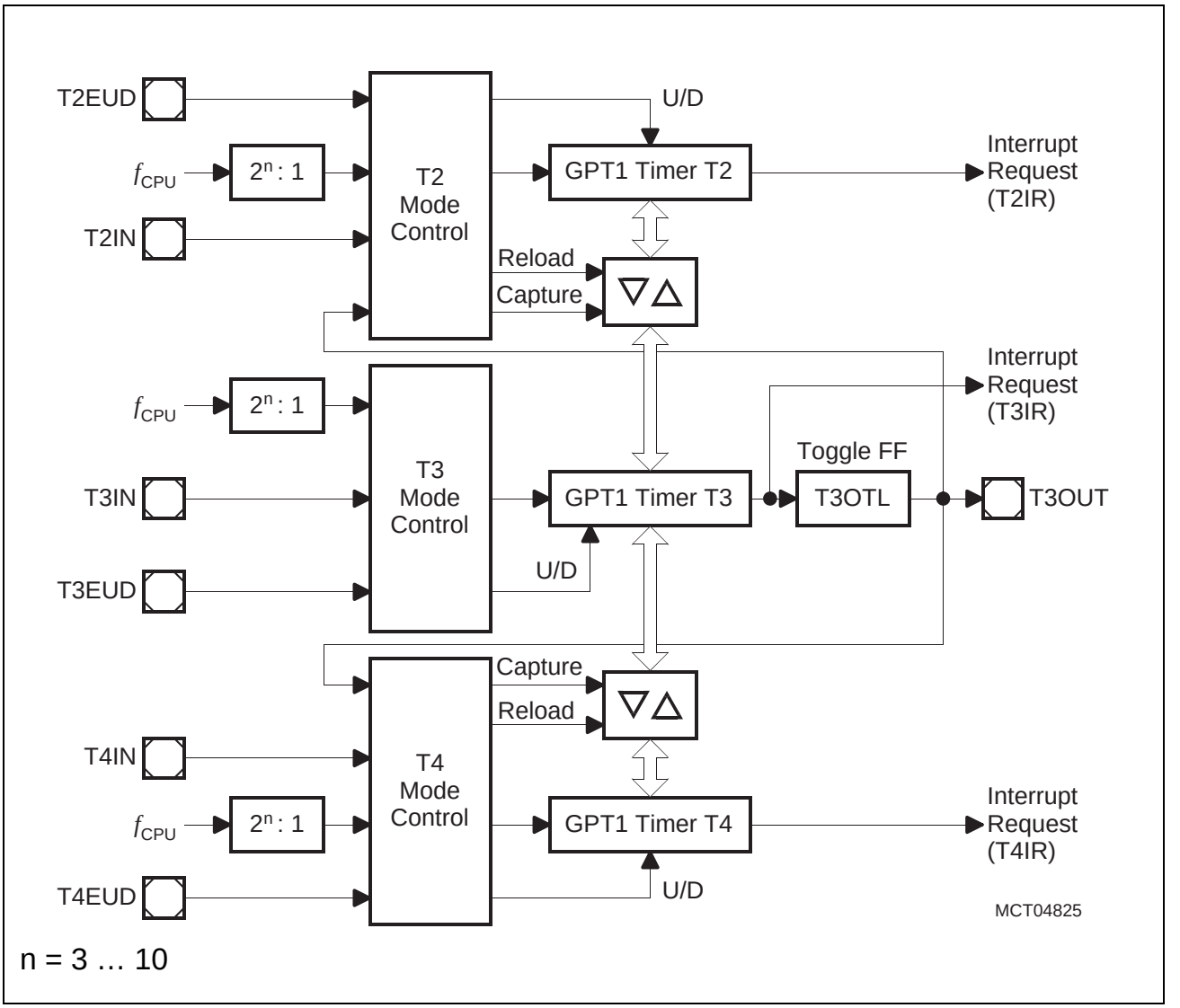



**Figure 6 Block Diagram of GPT1**

With its maximum resolution of 8 TCL, the **GPT2 module** provides precise event control and time measurement. It includes two timers (T5, T6) and a capture/reload register (CAPREL). Both timers can be clocked with an input clock which is derived from the CPU clock via a programmable prescaler or with external signals. The count direction (up/down) for each timer is programmable by software or may additionally be altered dynamically by an external signal on a port pin (TxEUD). Concatenation of the timers is supported via the output toggle latch (T6OTL) of timer T6, which changes its state on each timer overflow/underflow.

The state of this latch may be used to clock timer T5, and/or it may be output on pin T6OUT. The overflows/underflows of timer T6 can additionally be used to clock the CAPCOM timers T0 or T1, and to cause a reload from the CAPREL register. The CAPREL register may capture the contents of timer T5 based on an external signal transition on the corresponding port pin (CAPIN), and timer T5 may optionally be cleared

after the capture procedure. This allows the C161CS/JC/JI to measure absolute time differences or to perform pulse multiplication without software overhead.

The capture trigger (timer T5 to CAPREL) may also be generated upon transitions of GPT1 timer T3's inputs T3IN and/or T3EUD. This is especially advantageous when T3 operates in Incremental Interface Mode.

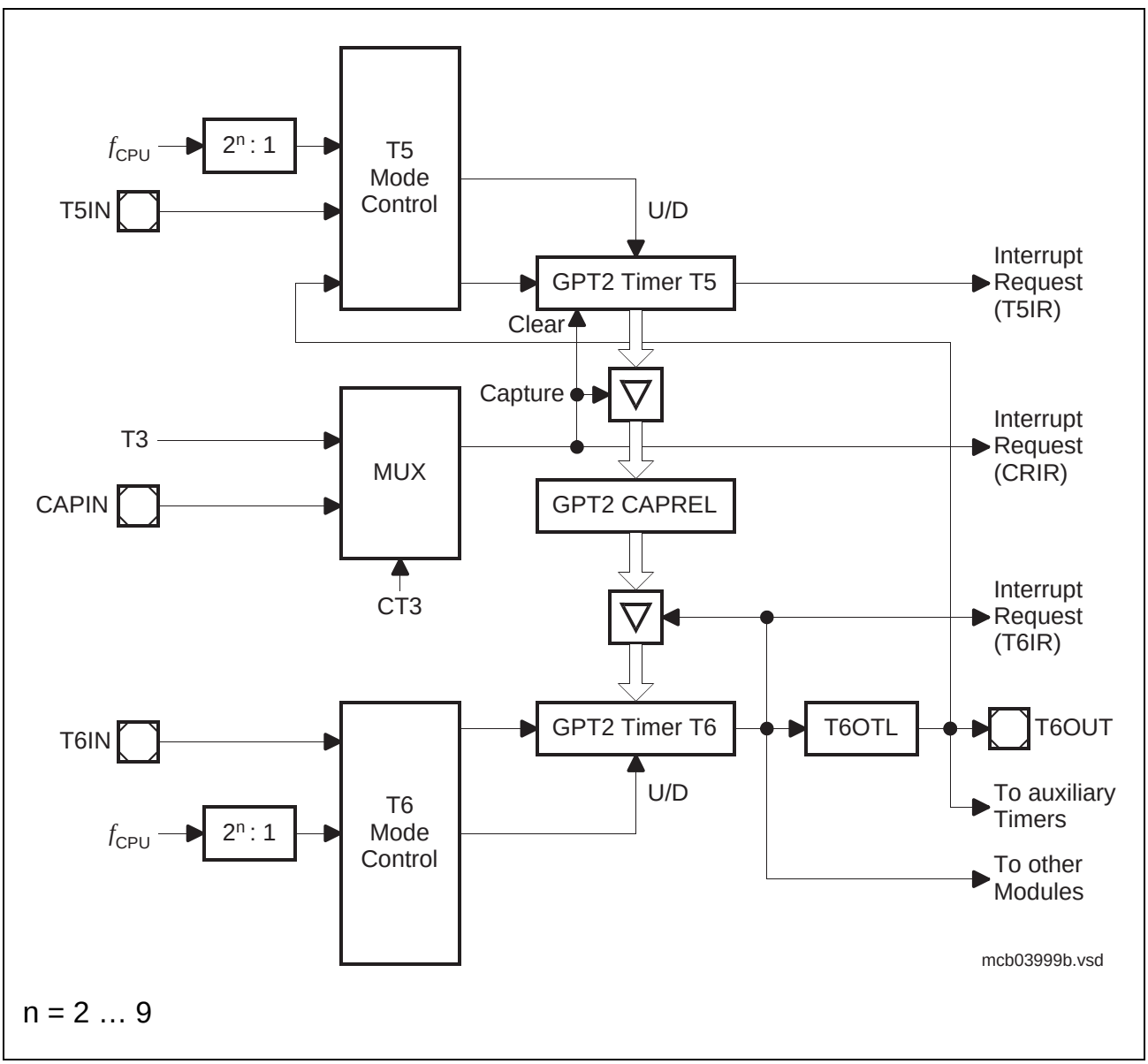



**Figure 7 Block Diagram of GPT2**

## Real Time Clock

The Real Time Clock (RTC) module of the C161CS/JC/JI consists of a chain of 3 divider blocks, a fixed 8:1 divider, the reloadable 16-bit timer T14, and the 32-bit RTC timer (accessible via registers RTCH and RTCL). The RTC module is directly clocked via a separate clock driver with the on-chip main oscillator frequency divided by 32 ($f_{RTC} = f_{OSCm} / 32$) or with the on-chip auxiliary oscillator frequency ($f_{RTC} = f_{OSCa}$). It is therefore independent from the selected clock generation mode of the C161CS/JC/JI. All timers count up.

The RTC module can be used for different purposes:

- System clock to determine the current time and date
- Cyclic time based interrupt
- 48-bit timer for long term measurements

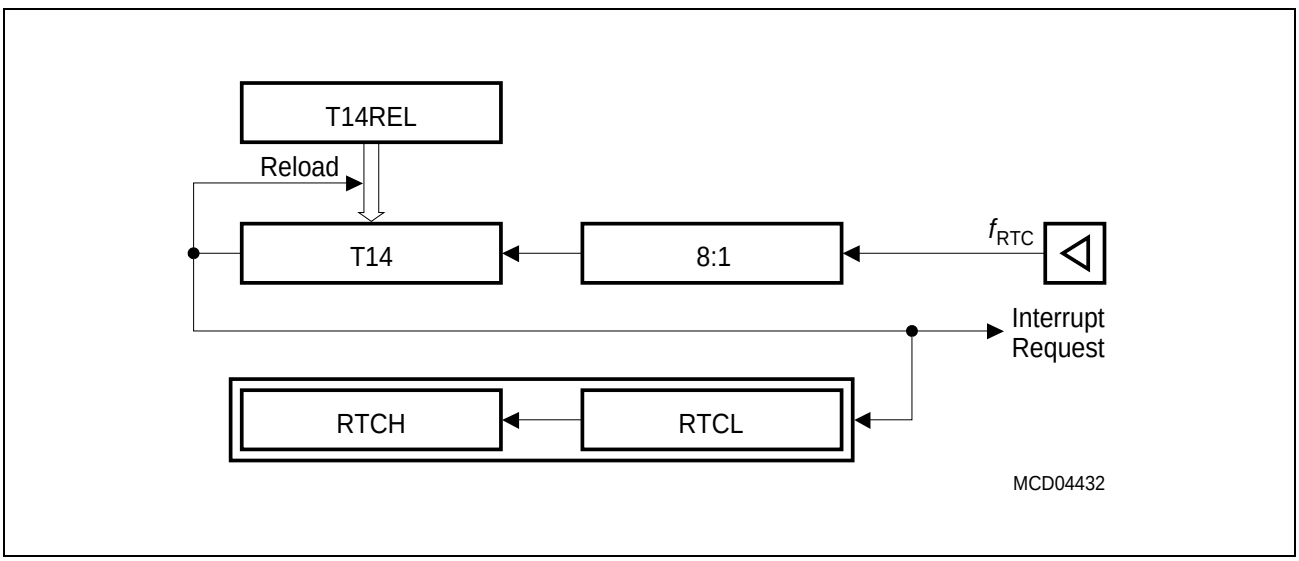



**Figure 8 RTC Block Diagram**

*Note: The registers associated with the RTC are not affected by a reset in order to maintain the correct system time even when intermediate resets are executed.*

## A/D Converter

For analog signal measurement, a 10-bit A/D converter with 12 multiplexed input channels and a sample and hold circuit has been integrated on-chip. It uses the method of successive approximation. The sample time (for loading the capacitors) and the conversion time is programmable and can so be adjusted to the external circuitry.

Overrun error detection/protection is provided for the conversion result register (ADDAT): either an interrupt request will be generated when the result of a previous conversion has not been read from the result register at the time the next conversion is complete, or the next conversion is suspended in such a case until the previous result has been read.

For applications which require less than 12 analog input channels, the remaining channel inputs can be used as digital input port pins.

The A/D converter of the C161CS/JC/JI supports four different conversion modes. In the standard Single Channel conversion mode, the analog level on a specified channel is sampled once and converted to a digital result. In the Single Channel Continuous mode, the analog level on a specified channel is repeatedly sampled and converted without software intervention. In the Auto Scan mode, the analog levels on a prespecified number of channels (standard or extension) are sequentially sampled and converted. In the Auto Scan Continuous mode, the number of prespecified channels is repeatedly sampled and converted. In addition, the conversion of a specific channel can be inserted (injected) into a running sequence without disturbing this sequence. This is called Channel Injection Mode.

The Peripheral Event Controller (PEC) may be used to automatically store the conversion results into a table in memory for later evaluation, without requiring the overhead of entering and exiting interrupt routines for each data transfer.

After each reset and also during normal operation the ADC automatically performs calibration cycles. This automatic self-calibration constantly adjusts the converter to changing operating conditions (e.g. temperature) and compensates process variations.

These calibration cycles are part of the conversion cycle, so they do not affect the normal operation of the A/D converter.

In order to decouple analog inputs from digital noise and to avoid input trigger noise those pins used for analog input can be disconnected from the digital IO or input stages under software control. This can be selected for each pin separately via register P5DIDIS (Port 5 Digital Input Disable).

## Serial Channels

Serial communication with other microcontrollers, processors, terminals or external peripheral components is provided by three serial interfaces with different functionality, two Asynchronous/Synchronous Serial Channels (**ASC0**/**ASC1**) and a High-Speed Synchronous Serial Channel (**SSC**).

**The ASC0** is upward compatible with the serial ports of the Infineon 8-bit microcontroller families and supports full-duplex asynchronous communication at up to 781 kBaud and half-duplex synchronous communication at up to 3.1 MBaud (@ 25 MHz CPU clock).
A dedicated baud rate generator allows to set up all standard baud rates without oscillator tuning. For transmission, reception and error handling 4 separate interrupt vectors are provided. In asynchronous mode, 8- or 9-bit data frames are transmitted or received, preceded by a start bit and terminated by one or two stop bits. For multiprocessor communication, a mechanism to distinguish address from data bytes has been included (8-bit data plus wake up bit mode).
In synchronous mode, the ASC0 transmits or receives bytes (8 bits) synchronously to a shift clock which is generated by the ASC0. The ASC0 always shifts the LSB first. A loop back option is available for testing purposes.
A number of optional hardware error detection capabilities has been included to increase the reliability of data transfers. A parity bit can automatically be generated on transmission or be checked on reception. Framing error detection allows to recognize data frames with missing stop bits. An overrun error will be generated, if the last character received has not been read out of the receive buffer register at the time the reception of a new character is complete.

**The ASC1** is function compatible with the ASC0, except that its registers are not bit-addressable (XBUS peripheral) and it provides only three interrupt vectors.

**The SSC** supports full-duplex synchronous communication at up to 6.25 MBaud (@ 25 MHz CPU clock). It may be configured so it interfaces with serially linked peripheral components. A dedicated baud rate generator allows to set up all standard baud rates without oscillator tuning. For transmission, reception and error handling three separate interrupt vectors are provided.
The SSC transmits or receives characters of 2 … 16 bits length synchronously to a shift clock which can be generated by the SSC (master mode) or by an external master (slave mode). The SSC can start shifting with the LSB or with the MSB and allows the selection of shifting and latching clock edges as well as the clock polarity.
A number of optional hardware error detection capabilities has been included to increase the reliability of data transfers. Transmit and receive error supervise the correct handling of the data buffer. Phase and baudrate error detect incorrect serial data.

### Serial Data Link Module (SDLM)

The Serial Data Link Module (SDLM) provides serial communication via a J1850 type multiplexed serial bus via an external J1850 bus transceiver. The module conforms to the SAE Class B J1850 specification for variable pulse width modulation (VPW). The SDLM is integrated as an on-chip peripheral and is connected to the CPU via the XBUS.

### General SDLM Features:

- Compliant to the SAE Class B J1850 specification (VPW)
- Class 2 protocol fully supported
- Variable Pulse Width (VPW) operation at 10.4 kBaud
- High Speed 4X operation at 41.6 kBaud
- Programmable Normalization Bit
- Programmable Delay for transceiver interface
- Digital Noise Filter
- Power Down mode with automatic wakeup support upon bus activity
- Single Byte Header and Consolidated Header supported
- CRC generation and checking
- Receive and transmit Block Mode

### Data Link Operation Features:

- 11 Byte Transmit Buffer
- Double buffered 11 Byte receive buffer (optional overwrite enable)
- Support for In Frame Response (IFR) types 1, 2 and 3
- Transmit and Receiver Message Buffers configurable for either FIFO or Byte mode
- Advanced Interrupt Handling with 8 separately enabled sources:
  Error, format or bus shorted
  CRC error
  Lost Arbitration
  Break received
  In-Frame-Response request
  Header received
  Complete message received
  Transmit successful
- Automatic IFR transmission (Types 1 and 2) for 3-Byte consolidated headers
- User configurable clock divider
- Bus status flags (IDLE, EOF, EOD, SOF, Tx and Rx in progress)

*Note: When the SDLM is used with the interface lines assigned to Port 4, the interface lines override the segment address lines and the segment address output on Port 4 is therefore limited to 6/4 bits i.e. address lines A21/A19 … A16. $\overline{CS}$ lines can be used to increase the total amount of addressable external memory.*

### CAN-Modules

The integrated CAN-Modules handle the completely autonomous transmission and reception of CAN frames in accordance with the CAN specification V2.0 part B (active), i.e. the on-chip CAN-Modules can receive and transmit standard frames with 11-bit identifiers as well as extended frames with 29-bit identifiers.

The modules provide Full CAN functionality on up to 15 message objects each. Message object 15 may be configured for Basic CAN functionality. Both modes provide separate masks for acceptance filtering which allows to accept a number of identifiers in Full CAN mode and also allows to disregard a number of identifiers in Basic CAN mode. All message objects can be updated independent from the other objects and are equipped for the maximum message length of 8 bytes.

The bit timing is derived from the XCLK and is programmable up to a data rate of 1 MBaud. Each CAN-Module uses two pins of Port 4 or Port 8 to interface to an external bus transceiver. The interface pins are assigned via software.

**Module CAN2** (C161CS only) is identical with the first one, except that it uses a separate address area and a separate interrupt node.

The two CAN modules can be internally coupled by assigning their interface pins to the same two port pins, or they can interface to separate CAN buses.

*Note: When one or both of the on-chip CAN Modules are used with the interface lines assigned to Port 4, the interface lines override the segment address lines and the segment address output on Port 4 is therefore limited to 6/4 bits i.e. address lines A21/A19 … A16. $\overline{CS}$ lines can be used to increase the total amount of addressable external memory.*

### IIC Module

The integrated IIC Bus Module handles the transmission and reception of frames over the two-line IIC bus in accordance with the IIC Bus specification. The on-chip IIC Module can receive and transmit data using 7-bit or 10-bit addressing and it can operate in slave mode, in master mode or in multi-master mode.

Several physical interfaces (port pins) can be established under software control. Data can be transferred at speeds up to 400 kbit/sec.

Two interrupt nodes dedicated to the IIC module allow efficient interrupt service and also support operation via PEC transfers.

*Note: The port pins associated with the IIC interfaces feature open drain drivers only, as required by the IIC specification.*

## Parallel Ports

The C161CS/JC/JI provides up to 93 I/O lines which are organized into eight input/output ports and one input port. All port lines are bit-addressable, and all input/output lines are individually (bit-wise) programmable as inputs or outputs via direction registers. The I/O ports are true bidirectional ports which are switched to high impedance state when configured as inputs. The output drivers of five I/O ports can be configured (pin by pin) for push/pull operation or open-drain operation via control registers, Port 9 provides open-drain-only drivers. During the internal reset, all port pins are configured as inputs.

The input threshold of Port 2, Port 3, Port 4, Port 6, and Port 7 is selectable (TTL or CMOS like), where the special CMOS like input threshold reduces noise sensitivity due to the input hysteresis. The input threshold may be selected individually for each byte of the respective ports.

All port lines have programmable alternate input or output functions associated with them. All port lines that are not used for these alternate functions may be used as general purpose IO lines.

PORT0 and PORT1 may be used as address and data lines when accessing external memory, while Port 4 outputs the additional segment address bits A23/19/17 … A16 in systems where segmentation is enabled to access more than 64 KBytes of memory.
Port 2, Port 7, and parts of PORT1 are associated with the capture inputs or compare outputs of the CAPCOM units.
Port 6 provides optional bus arbitration signals ($\overline{\text{BREQ}}$, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$) and chip select signals.
Port 3 includes alternate functions of timers, serial interfaces, the optional bus control signal $\overline{\text{BHE}}$, and the system clock output CLKOUT (or the programmable frequency output FOUT).
Port 5 is used for the analog input channels to the A/D converter or timer control signals.

The edge characteristics (transition time) and driver characteristics (output current) of the C161CS/JC/JI's port drivers can be selected via the Port Output Control registers (POCONx).

### Watchdog Timer

The Watchdog Timer represents one of the fail-safe mechanisms which have been implemented to prevent the controller from malfunctioning for longer periods of time.

The Watchdog Timer is always enabled after a reset of the chip, and can only be disabled in the time interval until the EINIT (end of initialization) instruction has been executed. Thus, the chip's start-up procedure is always monitored. The software has to be designed to service the Watchdog Timer before it overflows. If, due to hardware or software related failures, the software fails to do so, the Watchdog Timer overflows and generates an internal hardware reset and pulls the $\overline{\text{RSTOUT}}$ pin low in order to allow external hardware components to be reset.

The Watchdog Timer is a 16-bit timer, clocked with the system clock divided by 2/4/128/256. The high byte of the Watchdog Timer register can be set to a prespecified reload value (stored in WDTREL) in order to allow further variation of the monitored time interval. Each time it is serviced by the application software, the high byte of the Watchdog Timer is reloaded. Thus, time intervals between 20 µs and 671 ms can be monitored (@ 25 MHz).
The default Watchdog Timer interval after reset is 5.24 ms (@ 25 MHz).

### Oscillator Watchdog

The Oscillator Watchdog (OWD) monitors the clock signal generated by the on-chip oscillator (either with a crystal or via external clock drive). For this operation the PLL provides a clock signal which is used to supervise transitions on the oscillator clock. This PLL clock is independent from the XTAL1 clock. When the expected oscillator clock transitions are missing the OWD activates the PLL Unlock / OWD interrupt node and supplies the CPU with the PLL clock signal. Under these circumstances the PLL will oscillate with its basic frequency.

In direct drive mode the PLL base frequency is used directly ($f_{CPU}$ = 2 … 5 MHz).
In prescaler mode the PLL base frequency is divided by 2 ($f_{CPU}$ = 1 … 2.5 MHz).

*Note: The CPU clock source is only switched back to the oscillator clock after a hardware reset.*

**The oscillator watchdog can be disabled** by setting bit OWDDIS in register SYSCON. In this case (OWDDIS = '1') the PLL remains idle and provides no clock signal, while the CPU clock signal is derived directly from the oscillator clock or via prescaler or SDD. Also no interrupt request will be generated in case of a missing oscillator clock.

*Note: At the end of an external reset ($\overline{EA}$ = '0') bit OWDDIS reflects the inverted level of pin $\overline{RD}$ at that time. Thus the oscillator watchdog may also be disabled via hardware by (externally) pulling the $\overline{RD}$ line low upon a reset, similar to the standard reset configuration via PORT0. At the end of an internal reset ($\overline{EA}$ = '1') bit OWDDIS is cleared.*

### Power Management

The C161CS/JC/JI provides several means to control the power it consumes either at a given time or averaged over a certain timespan. Three mechanisms can be used (partly in parallel):

- **Power Saving Modes** switch the C161CS/JC/JI into a special operating mode (control via instructions).
  Idle Mode stops the CPU while the peripherals can continue to operate.
  Sleep Mode and Power Down Mode stop all clock signals and all operation (RTC may optionally continue running). Sleep Mode can be terminated by external interrupt signals.
- **Clock Generation Management** controls the distribution and the frequency of internal and external clock signals (control via register SYSCON2).
  Slow Down Mode lets the C161CS/JC/JI run at a CPU clock frequency of $f_{OSC}$ / 1 … 32 (half for prescaler operation) which drastically reduces the consumed power. The PLL can be optionally disabled while operating in Slow Down Mode.
  External circuitry can be controlled via the programmable frequency output FOUT.
- **Peripheral Management** permits temporary disabling of peripheral modules (control via register SYSCON3).
  Each peripheral can separately be disabled/enabled. A group control option disables a major part of the peripheral set by setting one single bit.

The on-chip RTC supports intermittend operation of the C161CS/JC/JI by generating cyclic wakeup signals. This offers full performance to quickly react on action requests while the intermittend sleep phases greatly reduce the average power consumption of the system.

## Instruction Set Summary

Table 6 lists the instructions of the C161CS/JC/JI in a condensed way.
The various addressing modes that can be used with a specific instruction, the operation of the instructions, parameters for conditional execution of instructions, and the opcodes for each instruction can be found in the **“C166 Family Instruction Set Manual”**.

This document also provides a detailled description of each instruction.

**Table 6 Instruction Set Summary**

| Mnemonic | Description | Bytes |
|---|---|---|
| ADD(B) | Add word (byte) operands | 2 / 4 |
| ADDC(B) | Add word (byte) operands with Carry | 2 / 4 |
| SUB(B) | Subtract word (byte) operands | 2 / 4 |
| SUBC(B) | Subtract word (byte) operands with Carry | 2 / 4 |
| MUL(U) | (Un)Signed multiply direct GPR by direct GPR (16-16-bit) | 2 |
| DIV(U) | (Un)Signed divide register MDL by direct GPR (16-/16-bit) | 2 |
| DIVL(U) | (Un)Signed long divide reg. MD by direct GPR (32-/16-bit) | 2 |
| CPL(B) | Complement direct word (byte) GPR | 2 |
| NEG(B) | Negate direct word (byte) GPR | 2 |
| AND(B) | Bitwise AND, (word/byte operands) | 2 / 4 |
| OR(B) | Bitwise OR, (word/byte operands) | 2 / 4 |
| XOR(B) | Bitwise XOR, (word/byte operands) | 2 / 4 |
| BCLR | Clear direct bit | 2 |
| BSET | Set direct bit | 2 |
| BMOV(N) | Move (negated) direct bit to direct bit | 4 |
| BAND, BOR, BXOR | AND/OR/XOR direct bit with direct bit | 4 |
| BCMP | Compare direct bit to direct bit | 4 |
| BFLDH/L | Bitwise modify masked high/low byte of bit-addressable direct word memory with immediate data | 4 |
| CMP(B) | Compare word (byte) operands | 2 / 4 |
| CMPD1/2 | Compare word data to GPR and decrement GPR by 1/2 | 2 / 4 |
| CMPI1/2 | Compare word data to GPR and increment GPR by 1/2 | 2 / 4 |
| PRIOR | Determine number of shift cycles to normalize direct word GPR and store result in direct word GPR | 2 |
| SHL / SHR | Shift left/right direct word GPR | 2 |
| ROL / ROR | Rotate left/right direct word GPR | 2 |
| ASHR | Arithmetic (sign bit) shift right direct word GPR | 2 |

**Table 6 Instruction Set Summary** (cont'd)

| Mnemonic | Description | Bytes |
|---|---|---|
| MOV(B) | Move word (byte) data | 2 / 4 |
| MOVBS | Move byte operand to word operand with sign extension | 2 / 4 |
| MOVBZ | Move byte operand to word operand. with zero extension | 2 / 4 |
| JMPA, JMPI, JMPR | Jump absolute/indirect/relative if condition is met | 4 |
| JMPS | Jump absolute to a code segment | 4 |
| J(N)B | Jump relative if direct bit is (not) set | 4 |
| JBC | Jump relative and clear bit if direct bit is set | 4 |
| JNBS | Jump relative and set bit if direct bit is not set | 4 |
| CALLA, CALLI, CALLR | Call absolute/indirect/relative subroutine if condition is met | 4 |
| CALLS | Call absolute subroutine in any code segment | 4 |
| PCALL | Push direct word register onto system stack and call absolute subroutine | 4 |
| TRAP | Call interrupt service routine via immediate trap number | 2 |
| PUSH, POP | Push/pop direct word register onto/from system stack | 2 |
| SCXT | Push direct word register onto system stack and update register with word operand | 4 |
| RET | Return from intra-segment subroutine | 2 |
| RETS | Return from inter-segment subroutine | 2 |
| RETP | Return from intra-segment subroutine and pop direct word register from system stack | 2 |
| RETI | Return from interrupt service subroutine | 2 |
| SRST | Software Reset | 4 |
| IDLE | Enter Idle Mode | 4 |
| PWRDN | Enter Power Down Mode (supposes $\overline{\text{NMI}}$-pin being low) | 4 |
| SRVWDT | Service Watchdog Timer | 4 |
| DISWDT | Disable Watchdog Timer | 4 |
| EINIT | Signify End-of-Initialization on RSTOUT-pin | 4 |
| ATOMIC | Begin ATOMIC sequence | 2 |
| EXTR | Begin EXTended Register sequence | 2 |
| EXTP(R) | Begin EXTended Page (and Register) sequence | 2 / 4 |
| EXTS(R) | Begin EXTended Segment (and Register) sequence | 2 / 4 |
| NOP | Null operation | 2 |

## Special Function Registers Overview

Table 7 lists all SFRs which are implemented in the C161CS/JC/JI in alphabetical order. **Bit-addressable** SFRs are marked with the letter "**b**" in column "Name". SFRs within the **Extended SFR-Space** (ESFRs) are marked with the letter "**E**" in column "Physical Address". Registers within on-chip X-peripherals are marked with the letter "**X**" in column "Physical Address".

An SFR can be specified via its individual mnemonic name. Depending on the selected addressing mode, an SFR can be accessed via its physical address (using the Data Page Pointers), or via its short 8-bit address (without using the Data Page Pointers).

*Note: Registers within device specific interface modules (CAN, SDLM) are only present in the corresponding device, of course.*

**Table 7 C161CS/JC/JI Registers, Ordered by Name**

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **ADCIC b** | $FF98_H$ | $CC_H$ | A/D Converter End of Conversion Interrupt Control Register | $0000_H$ |
| **ADCON b** | $FFA0_H$ | $D0_H$ | A/D Converter Control Register | $0000_H$ |
| **ADDAT** | $FEA0_H$ | $50_H$ | A/D Converter Result Register | $0000_H$ |
| **ADDAT2** | $F0A0_H$ **E** | $50_H$ | A/D Converter 2 Result Register | $0000_H$ |
| **ADDRSEL1** | $FE18_H$ | $0C_H$ | Address Select Register 1 | $0000_H$ |
| **ADDRSEL2** | $FE1A_H$ | $0D_H$ | Address Select Register 2 | $0000_H$ |
| **ADDRSEL3** | $FE1C_H$ | $0E_H$ | Address Select Register 3 | $0000_H$ |
| **ADDRSEL4** | $FE1E_H$ | $0F_H$ | Address Select Register 4 | $0000_H$ |
| **ADEIC b** | $FF9A_H$ | $CD_H$ | A/D Converter Overrun Error Interrupt Control Register | $0000_H$ |
| **BUFFCON** | $EB24_H$ **X** | --- | SDLM Buffer Control Register | $0000_H$ |
| **BUFFSTAT** | $EB1C_H$ **X** | --- | SDLM Buffer Status Register | $0000_H$ |
| **BUSCON0 b** | $FF0C_H$ | $86_H$ | Bus Configuration Register 0 | $0000_H$ |
| **BUSCON1 b** | $FF14_H$ | $8A_H$ | Bus Configuration Register 1 | $0000_H$ |
| **BUSCON2 b** | $FF16_H$ | $8B_H$ | Bus Configuration Register 2 | $0000_H$ |
| **BUSCON3 b** | $FF18_H$ | $8C_H$ | Bus Configuration Register 3 | $0000_H$ |
| **BUSCON4 b** | $FF1A_H$ | $8D_H$ | Bus Configuration Register 4 | $0000_H$ |
| **BUSSTAT** | $EB20_H$ **X** | --- | SDLM Bus Status Register | $0000_H$ |
| **C1BTR** | $EF04_H$ **X** | --- | CAN1 Bit Timing Register | $UUUU_H$ |
| **C1CSR** | $EF00_H$ **X** | --- | CAN1 Control / Status Register | $XX01_H$ |
| **C1GMS** | $EF06_H$ **X** | --- | CAN1 Global Mask Short | $UFUU_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **C1PCIR** | $EF02_H$ **X** | --- | CAN1 Port Control / Interrupt Register | $XXXX_H$ |
| **C1LARn** | $EFn4_H$ **X** | --- | CAN1 Lower Arbitration Reg. (msg. **n**) | $UUUU_H$ |
| **C1LGML** | $EF0A_H$ **X** | --- | CAN1 Lower Global Mask Long | $UUUU_H$ |
| **C1LMLM** | $EF0E_H$ **X** | --- | CAN1 Lower Mask of Last Message | $UUUU_H$ |
| **C1MCFGn** | $EFn6_H$ **X** | --- | CAN1 Message Config. Reg. (msg. **n**) | $UU_H$ |
| **C1MCRn** | $EFn0_H$ **X** | --- | CAN1 Message Control Reg. (msg. **n**) | $UUUU_H$ |
| **C1UARn** | $EFn2_H$ **X** | --- | CAN1 Upper Arbitration Reg. (msg. **n**) | $UUUU_H$ |
| **C1UGML** | $EF08_H$ **X** | --- | CAN1 Upper Global Mask Long | $UUUU_H$ |
| **C1UMLM** | $EF0C_H$ **X** | --- | CAN1 Upper Mask of Last Message | $UUUU_H$ |
| **C2BTR** | $EE04_H$ **X** | --- | CAN2 Bit Timing Register | $UUUU_H$ |
| **C2CSR** | $EE00_H$ **X** | --- | CAN2 Control / Status Register | $XX01_H$ |
| **C2GMS** | $EE06_H$ **X** | --- | CAN2 Global Mask Short | $UFUU_H$ |
| **C2PCIR** | $EE02_H$ **X** | --- | CAN2 Port Control / Interrupt Register | $XXXX_H$ |
| **C2LARn** | $EEn4_H$ **X** | --- | CAN2 Lower Arbitration Reg. (msg. **n**) | $UUUU_H$ |
| **C2LGML** | $EE0A_H$ **X** | --- | CAN2 Lower Global Mask Long | $UUUU_H$ |
| **C2LMLM** | $EE0E_H$ **X** | --- | CAN2 Lower Mask of Last Message | $UUUU_H$ |
| **C2MCFGn** | $EEn6_H$ **X** | --- | CAN2 Message Config. Reg. (msg. **n**) | $UU_H$ |
| **C2MCRn** | $EEn0_H$ **X** | --- | CAN2 Message Control Reg. (msg. **n**) | $UUUU_H$ |
| **C2UARn** | $EEn2_H$ **X** | --- | CAN2 Upper Arbitration Reg. (msg. **n**) | $UUUU_H$ |
| **C2UGML** | $EE08_H$ **X** | --- | CAN2 Upper Global Mask Long | $UUUU_H$ |
| **C2UMLM** | $EE0C_H$ **X** | --- | CAN2 Upper Mask of Last Message | $UUUU_H$ |
| **CAPREL** | $FE4A_H$ | $25_H$ | GPT2 Capture/Reload Register | $0000_H$ |
| **CC0** | $FE80_H$ | $40_H$ | CAPCOM Register 0 | $0000_H$ |
| **CC0IC** **b** | $FF78_H$ | $BC_H$ | CAPCOM Register 0 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC1** | $FE82_H$ | $41_H$ | CAPCOM Register 1 | $0000_H$ |
| **CC10** | $FE94_H$ | $4A_H$ | CAPCOM Register 10 | $0000_H$ |
| **CC10IC** **b** | $FF8C_H$ | $C6_H$ | CAPCOM Reg. 10 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC11** | $FE96_H$ | $4B_H$ | CAPCOM Register 11 | $0000_H$ |
| **CC11IC** **b** | $FF8E_H$ | $C7_H$ | CAPCOM Reg. 11 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC12** | $FE98_H$ | $4C_H$ | CAPCOM Register 12 | $0000_H$ |
| **CC12IC** **b** | $FF90_H$ | $C8_H$ | CAPCOM Reg. 12 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC13** | $FE9A_H$ | $4D_H$ | CAPCOM Register 13 | $0000_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **CC13IC b** | $FF92_H$ | $C9_H$ | CAPCOM Reg. 13 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC14** | $FE9C_H$ | $4E_H$ | CAPCOM Register 14 | $0000_H$ |
| **CC14IC b** | $FF94_H$ | $CA_H$ | CAPCOM Reg. 14 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC15** | $FE9E_H$ | $4F_H$ | CAPCOM Register 15 | $0000_H$ |
| **CC15IC b** | $FF96_H$ | $CB_H$ | CAPCOM Reg. 15 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC16** | $FE60_H$ | $30_H$ | CAPCOM Register 16 | $0000_H$ |
| **CC16IC b** | $F160_H$ **E** | $B0_H$ | CAPCOM Reg.16 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC17** | $FE62_H$ | $31_H$ | CAPCOM Register 17 | $0000_H$ |
| **CC17IC b** | $F162_H$ **E** | $B1_H$ | CAPCOM Reg. 17 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC18** | $FE64_H$ | $32_H$ | CAPCOM Register 18 | $0000_H$ |
| **CC18IC b** | $F164_H$ **E** | $B2_H$ | CAPCOM Reg. 18 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC19** | $FE66_H$ | $33_H$ | CAPCOM Register 19 | $0000_H$ |
| **CC19IC b** | $F166_H$ **E** | $B3_H$ | CAPCOM Reg. 19 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC1IC b** | $FF7A_H$ | $BD_H$ | CAPCOM Reg. 1 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC2** | $FE84_H$ | $42_H$ | CAPCOM Register 2 | $0000_H$ |
| **CC20** | $FE68_H$ | $34_H$ | CAPCOM Register 20 | $0000_H$ |
| **CC20IC b** | $F168_H$ **E** | $B4_H$ | CAPCOM Reg. 20 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC21** | $FE6A_H$ | $35_H$ | CAPCOM Register 21 | $0000_H$ |
| **CC21IC b** | $F16A_H$ **E** | $B5_H$ | CAPCOM Reg. 21 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC22** | $FE6C_H$ | $36_H$ | CAPCOM Register 22 | $0000_H$ |
| **CC22IC b** | $F16C_H$ **E** | $B6_H$ | CAPCOM Reg. 22 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC23** | $FE6E_H$ | $37_H$ | CAPCOM Register 23 | $0000_H$ |
| **CC23IC b** | $F16E_H$ **E** | $B7_H$ | CAPCOM Reg. 23 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC24** | $FE70_H$ | $38_H$ | CAPCOM Register 24 | $0000_H$ |
| **CC24IC b** | $F170_H$ **E** | $B8_H$ | CAPCOM Reg. 24 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC25** | $FE72_H$ | $39_H$ | CAPCOM Register 25 | $0000_H$ |
| **CC25IC b** | $F172_H$ **E** | $B9_H$ | CAPCOM Reg. 25 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC26** | $FE74_H$ | $3A_H$ | CAPCOM Register 26 | $0000_H$ |
| **CC26IC b** | $F174_H$ **E** | $BA_H$ | CAPCOM Reg. 26 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC27** | $FE76_H$ | $3B_H$ | CAPCOM Register 27 | $0000_H$ |
| **CC27IC b** | $F176_H$ **E** | $BB_H$ | CAPCOM Reg. 27 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC28** | $FE78_H$ | $3C_H$ | CAPCOM Register 28 | $0000_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **CC28IC b** | $F178_H$ **E** | $BC_H$ | CAPCOM Reg. 28 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC29** | $FE7A_H$ | $3D_H$ | CAPCOM Register 29 | $0000_H$ |
| **CC29IC b** | $F184_H$ **E** | $C2_H$ | CAPCOM Reg. 29 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC2IC b** | $FF7C_H$ | $BE_H$ | CAPCOM Reg. 2 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC3** | $FE86_H$ | $43_H$ | CAPCOM Register 3 | $0000_H$ |
| **CC30** | $FE7C_H$ | $3E_H$ | CAPCOM Register 30 | $0000_H$ |
| **CC30IC b** | $F18C_H$ **E** | $C6_H$ | CAPCOM Reg. 30 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC31** | $FE7E_H$ | $3F_H$ | CAPCOM Register 31 | $0000_H$ |
| **CC31IC b** | $F194_H$ **E** | $CA_H$ | CAPCOM Reg. 31 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC3IC b** | $FF7E_H$ | $BF_H$ | CAPCOM Reg. 3 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC4** | $FE88_H$ | $44_H$ | CAPCOM Register 4 | $0000_H$ |
| **CC4IC b** | $FF80_H$ | $C0_H$ | CAPCOM Reg. 4 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC5** | $FE8A_H$ | $45_H$ | CAPCOM Register 5 | $0000_H$ |
| **CC5IC b** | $FF82_H$ | $C1_H$ | CAPCOM Reg. 5 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC6** | $FE8C_H$ | $46_H$ | CAPCOM Register 6 | $0000_H$ |
| **CC6IC b** | $FF84_H$ | $C2_H$ | CAPCOM Reg. 6 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC7** | $FE8E_H$ | $47_H$ | CAPCOM Register 7 | $0000_H$ |
| **CC7IC b** | $FF86_H$ | $C3_H$ | CAPCOM Reg. 7 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC8** | $FE90_H$ | $48_H$ | CAPCOM Register 8 | $0000_H$ |
| **CC8IC b** | $FF88_H$ | $C4_H$ | CAPCOM Reg. 8 Interrupt Ctrl. Reg. | $0000_H$ |
| **CC9** | $FE92_H$ | $49_H$ | CAPCOM Register 9 | $0000_H$ |
| **CC9IC b** | $FF8A_H$ | $C5_H$ | CAPCOM Reg. 9 Interrupt Ctrl. Reg. | $0000_H$ |
| **CCM0 b** | $FF52_H$ | $A9_H$ | CAPCOM Mode Control Register 0 | $0000_H$ |
| **CCM1 b** | $FF54_H$ | $AA_H$ | CAPCOM Mode Control Register 1 | $0000_H$ |
| **CCM2 b** | $FF56_H$ | $AB_H$ | CAPCOM Mode Control Register 2 | $0000_H$ |
| **CCM3 b** | $FF58_H$ | $AC_H$ | CAPCOM Mode Control Register 3 | $0000_H$ |
| **CCM4 b** | $FF22_H$ | $91_H$ | CAPCOM Mode Control Register 4 | $0000_H$ |
| **CCM5 b** | $FF24_H$ | $92_H$ | CAPCOM Mode Control Register 5 | $0000_H$ |
| **CCM6 b** | $FF26_H$ | $93_H$ | CAPCOM Mode Control Register 6 | $0000_H$ |
| **CCM7 b** | $FF28_H$ | $94_H$ | CAPCOM Mode Control Register 7 | $0000_H$ |
| **CLKDIV** | $EB14_H$ **X** | --- | SDLM Clock Divider Register | $0000_H$ |
| **CP** | $FE10_H$ | $08_H$ | CPU Context Pointer Register | $FC00_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | | Physical Address | | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|---|---|
| **CRIC** | **b** | $FF6A_H$ | | $B5_H$ | GPT2 CAPREL Interrupt Ctrl. Reg. | $0000_H$ |
| **CSP** | | $FE08_H$ | | $04_H$ | CPU Code Segment Pointer Register (8 bits, not directly writeable) | $0000_H$ |
| **DP0H** | **b** | $F102_H$ | **E** | $81_H$ | P0H Direction Control Register | $00_H$ |
| **DP0L** | **b** | $F100_H$ | **E** | $80_H$ | P0L Direction Control Register | $00_H$ |
| **DP1H** | **b** | $F106_H$ | **E** | $83_H$ | P1H Direction Control Register | $00_H$ |
| **DP1L** | **b** | $F104_H$ | **E** | $82_H$ | P1L Direction Control Register | $00_H$ |
| **DP2** | **b** | $FFC2_H$ | | $E1_H$ | Port 2 Direction Control Register | $0000_H$ |
| **DP3** | **b** | $FFC6_H$ | | $E3_H$ | Port 3 Direction Control Register | $0000_H$ |
| **DP4** | **b** | $FFCA_H$ | | $E5_H$ | Port 4 Direction Control Register | $00_H$ |
| **DP6** | **b** | $FFCE_H$ | | $E7_H$ | Port 6 Direction Control Register | $00_H$ |
| **DP7** | **b** | $FFD2_H$ | | $E9_H$ | Port 7 Direction Control Register | $00_H$ |
| **DP9** | **b** | $FFDA_H$ | | $ED_H$ | Port 9 Direction Control Register | $00_H$ |
| **DPP0** | | $FE00_H$ | | $00_H$ | CPU Data Page Pointer 0 Reg. (10 bits) | $0000_H$ |
| **DPP1** | | $FE02_H$ | | $01_H$ | CPU Data Page Pointer 1 Reg. (10 bits) | $0001_H$ |
| **DPP2** | | $FE04_H$ | | $02_H$ | CPU Data Page Pointer 2 Reg. (10 bits) | $0002_H$ |
| **DPP3** | | $FE06_H$ | | $03_H$ | CPU Data Page Pointer 3 Reg. (10 bits) | $0003_H$ |
| **ERRSTAT** | | $EB22_H$ | **X** | --- | SDLM Error Status Register | $0000_H$ |
| **EXICON** | **b** | $F1C0_H$ | **E** | $E0_H$ | External Interrupt Control Register | $0000_H$ |
| **EXISEL** | **b** | $F1DA_H$ | **E** | $ED_H$ | External Interrupt Source Select Register | $0000_H$ |
| **FLAGRST** | | $EB28_H$ | **X** | --- | SDLM Flag Reset Register | $0000_H$ |
| **FOCON** | **b** | $FFAA_H$ | | $D5_H$ | Frequency Output Control Register | $0000_H$ |
| **GLOBCON** | | $EB10_H$ | **X** | --- | SDLM Global Control Register | $0000_H$ |
| **ICADR** | | $ED06_H$ | **X** | --- | IIC Address Register | $0XXX_H$ |
| **ICCFG** | | $ED00_H$ | **X** | --- | IIC Configuration Register | $XX00_H$ |
| **ICCON** | | $ED02_H$ | **X** | --- | IIC Control Register | $0000_H$ |
| **ICRTB** | | $ED08_H$ | **X** | --- | IIC Receive/Transmit Buffer | $XX_H$ |
| **ICST** | | $ED04_H$ | **X** | --- | IIC Status Register | $0000_H$ |
| **IDCHIP** | | $F07C_H$ | **E** | $3E_H$ | Identifier | $1XXX_H$ |
| **IDMANUF** | | $F07E_H$ | **E** | $3F_H$ | Identifier | $1820_H$ |
| **IDMEM** | | $F07A_H$ | **E** | $3D_H$ | Identifier | $X040_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **IDPROG** | $F078_H$ **E** | $3C_H$ | Identifier | $XXXX_H$ |
| **IFR** | $EB18_H$ **X** | --- | SDLM In-Frame Response Register | $0000_H$ |
| **INTCON** | $EB2C_H$ **X** | --- | SDLM Interrupt Control Register | $0000_H$ |
| **IPCR** | $EB04_H$ **X** | --- | SDLM Interface Port Connect Register | $0007_H$ |
| **ISNC** | $F1DE_H$ **E** | $EF_H$ | Interrupt Subnode Control Register | $0000_H$ |
| **MDC b** | $FF0E_H$ | $87_H$ | CPU Multiply Divide Control Register | $0000_H$ |
| **MDH** | $FE0C_H$ | $06_H$ | CPU Multiply Divide Reg. – High Word | $0000_H$ |
| **MDL** | $FE0E_H$ | $07_H$ | CPU Multiply Divide Reg. – Low Word | $0000_H$ |
| **ODP2 b** | $F1C2_H$ **E** | $E1_H$ | Port 2 Open Drain Control Register | $0000_H$ |
| **ODP3 b** | $F1C6_H$ **E** | $E3_H$ | Port 3 Open Drain Control Register | $0000_H$ |
| **ODP4 b** | $F1CA_H$ **E** | $E5_H$ | Port 4 Open Drain Control Register | $00_H$ |
| **ODP6 b** | $F1CE_H$ **E** | $E7_H$ | Port 6 Open Drain Control Register | $00_H$ |
| **ODP7 b** | $F1D2_H$ **E** | $E9_H$ | Port 7 Open Drain Control Register | $00_H$ |
| **ONES b** | $FF1E_H$ | $8F_H$ | Constant Value 1's Register (read only) | $FFFF_H$ |
| **P0H b** | $FF02_H$ | $81_H$ | Port 0 High Reg. (Upper half of PORT0) | $00_H$ |
| **P0L b** | $FF00_H$ | $80_H$ | Port 0 Low Reg. (Lower half of PORT0) | $00_H$ |
| **P1H b** | $FF06_H$ | $83_H$ | Port 1 High Reg. (Upper half of PORT1) | $00_H$ |
| **P1L b** | $FF04_H$ | $82_H$ | Port 1 Low Reg. (Lower half of PORT1) | $00_H$ |
| **P2 b** | $FFC0_H$ | $E0_H$ | Port 2 Register | $0000_H$ |
| **P3 b** | $FFC4_H$ | $E2_H$ | Port 3 Register | $0000_H$ |
| **P4 b** | $FFC8_H$ | $E4_H$ | Port 4 Register (7 bits) | $00_H$ |
| **P5 b** | $FFA2_H$ | $D1_H$ | Port 5 Register (read only) | $XXXX_H$ |
| **P6 b** | $FFCC_H$ | $E6_H$ | Port 6 Register (8 bits) | $00_H$ |
| **P7 b** | $FFD0_H$ | $E8_H$ | Port 7 Register (8 bits) | $00_H$ |
| **P9 b** | $FFD8_H$ | $EC_H$ | Port 9 Register (8 bits) | $00_H$ |
| **PECC0** | $FEC0_H$ | $60_H$ | PEC Channel 0 Control Register | $0000_H$ |
| **PECC1** | $FEC2_H$ | $61_H$ | PEC Channel 1 Control Register | $0000_H$ |
| **PECC2** | $FEC4_H$ | $62_H$ | PEC Channel 2 Control Register | $0000_H$ |
| **PECC3** | $FEC6_H$ | $63_H$ | PEC Channel 3 Control Register | $0000_H$ |
| **PECC4** | $FEC8_H$ | $64_H$ | PEC Channel 4 Control Register | $0000_H$ |
| **PECC5** | $FECA_H$ | $65_H$ | PEC Channel 5 Control Register | $0000_H$ |
| **PECC6** | $FECC_H$ | $66_H$ | PEC Channel 6 Control Register | $0000_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **PECC7** | $FECE_H$ | $67_H$ | PEC Channel 7 Control Register | $0000_H$ |
| **PICON b** | $F1C4_H$ **E** | $E2_H$ | Port Input Threshold Control Register | $0000_H$ |
| **POCON0H** | $F082_H$ **E** | $41_H$ | P0L Output Control Register | $0000_H$ |
| **POCON0L** | $F080_H$ **E** | $40_H$ | P0H Output Control Register | $0000_H$ |
| **POCON1H** | $F086_H$ **E** | $43_H$ | P1L Output Control Register | $0000_H$ |
| **POCON1L** | $F084_H$ **E** | $42_H$ | P1H Output Control Register | $0000_H$ |
| **POCON2** | $F088_H$ **E** | $44_H$ | Port 2 Output Control Register | $0000_H$ |
| **POCON20** | $F0AA_H$ **E** | $55_H$ | Dedicated Pins Output Control Register | $0000_H$ |
| **POCON3** | $F08A_H$ **E** | $45_H$ | Port 3 Output Control Register | $0000_H$ |
| **POCON4** | $F08C_H$ **E** | $46_H$ | Port 4 Output Control Register | $0000_H$ |
| **POCON6** | $F08E_H$ **E** | $47_H$ | Port 6 Output Control Register | $0000_H$ |
| **POCON7** | $F090_H$ **E** | $48_H$ | Port 7 Output Control Register | $0000_H$ |
| **PSW b** | $FF10_H$ | $88_H$ | CPU Program Status Word | $0000_H$ |
| **RP0H b** | $F108_H$ **E** | $84_H$ | System Startup Configuration Register (Rd. only) | $XX_H$ |
| **RSTCON b** | $F1E0_H$ **m** | --- | Reset Control Register | $00XX_H$ |
| **RTCH** | $F0D6_H$ **E** | $6B_H$ | RTC High Register | no |
| **RTCL** | $F0D4_H$ **E** | $6A_H$ | RTC Low Register | no |
| **RXCNT** | $EB4C_H$ **X** | --- | SDLM Bus Receive Byte Counter (CPU) | $0000_H$ |
| **RXCNTB** | $EB4A_H$ **X** | --- | SDLM Bus Receive Byte Counter (bus) | $0000_H$ |
| **RXCPU** | $EB4E_H$ **X** | --- | SDLM CPU Receive Byte Counter Reg. | $0000_H$ |
| **RXD00** | $EB40_H$ **X** | --- | SDLM Receive Data Register 00 (CPU) | $0000_H$ |
| **RXD010** | $EB4A_H$ **X** | --- | SDLM Receive Data Register 010 (CPU) | $0000_H$ |
| **RXD02** | $EB42_H$ **X** | --- | SDLM Receive Data Register 02 (CPU) | $0000_H$ |
| **RXD04** | $EB44_H$ **X** | --- | SDLM Receive Data Register 04 (CPU) | $0000_H$ |
| **RXD06** | $EB46_H$ **X** | --- | SDLM Receive Data Register 06 (CPU) | $0000_H$ |
| **RXD08** | $EB48_H$ **X** | --- | SDLM Receive Data Register 08 (CPU) | $0000_H$ |
| **RXD10** | $EB50_H$ **X** | --- | SDLM Receive Data Register 10 (bus) | $0000_H$ |
| **RXD110** | $EB5A_H$ **X** | --- | SDLM Receive Data Register 110 (bus) | $0000_H$ |
| **RXD12** | $EB52_H$ **X** | --- | SDLM Receive Data Register 12 (bus) | $0000_H$ |
| **RXD14** | $EB54_H$ **X** | --- | SDLM Receive Data Register 14 (bus) | $0000_H$ |
| **RXD16** | $EB56_H$ **X** | --- | SDLM Receive Data Register 16 (bus) | $0000_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **RXD18** | $EB58_H$ **X** | --- | SDLM Receive Data Register 18 (bus) | $0000_H$ |
| **S0BG** | $FEB4_H$ | $5A_H$ | Serial Channel 0 Baud Rate Generator Reload Register | $0000_H$ |
| **S0CON** **b** | $FFB0_H$ | $D8_H$ | Serial Channel 0 Control Register | $0000_H$ |
| **S0EIC** **b** | $FF70_H$ | $B8_H$ | Serial Channel 0 Error Interrupt Ctrl. Reg. | $0000_H$ |
| **S0RBUF** | $FEB2_H$ | $59_H$ | Serial Channel 0 Receive Buffer Register (read only) | $XXXX_H$ |
| **S0RIC** **b** | $FF6E_H$ | $B7_H$ | Serial Channel 0 Receive Interrupt Control Register | $0000_H$ |
| **S0TBIC** **b** | $F19C_H$ **E** | $CE_H$ | Serial Channel 0 Transmit Buffer Interrupt Control Register | $0000_H$ |
| **S0TBUF** | $FEB0_H$ | $58_H$ | Serial Channel 0 Transmit Buffer Register | $0000_H$ |
| **S0TIC** **b** | $FF6C_H$ | $B6_H$ | Serial Channel 0 Transmit Interrupt Control Register | $0000_H$ |
| **S1BG** | $EDA4_H$ **X** | --- | Serial Channel 1 Baud Rate Generator Reload Register | $0000_H$ |
| **S1CON** | $EDA6_H$ **X** | --- | Serial Channel 1 Control Register | $0000_H$ |
| **S1RBUF** | $EDA2_H$ **X** | --- | Serial Channel 1 Receive Buffer Register (read only) | $XXXX_H$ |
| **S1TBUF** | $EDA0_H$ **X** | --- | Serial Channel 1 Transmit Buffer Register | $0000_H$ |
| **SOFPTR** | $EB60_H$ **X** | --- | SDLM Start-of-Frame Pointer Register | $0000_H$ |
| **SP** | $FE12_H$ | $09_H$ | CPU System Stack Pointer Register | $FC00_H$ |
| **SSCBR** | $F0B4_H$ **E** | $5A_H$ | SSC Baudrate Register | $0000_H$ |
| **SSCCON** **b** | $FFB2_H$ | $D9_H$ | SSC Control Register | $0000_H$ |
| **SSCEIC** **b** | $FF76_H$ | $BB_H$ | SSC Error Interrupt Control Register | $0000_H$ |
| **SSCRB** | $F0B2_H$ **E** | $59_H$ | SSC Receive Buffer (read only) | $XXXX_H$ |
| **SSCRIC** **b** | $FF74_H$ | $BA_H$ | SSC Receive Interrupt Control Register | $0000_H$ |
| **SSCTB** | $F0B0_H$ **E** | $58_H$ | SSC Transmit Buffer (write only) | $0000_H$ |
| **SSCTIC** **b** | $FF72_H$ | $B9_H$ | SSC Transmit Interrupt Control Register | $0000_H$ |
| **STKOV** | $FE14_H$ | $0A_H$ | CPU Stack Overflow Pointer Register | $FA00_H$ |
| **STKUN** | $FE16_H$ | $0B_H$ | CPU Stack Underflow Pointer Register | $FC00_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **SYSCON b** | $FF12_H$ | $89_H$ | CPU System Configuration Register | [1]$0XX0_H$ |
| **SYSCON1 b** | $F1DC_H$ **E** | $EE_H$ | CPU System Configuration Register 1 | $0000_H$ |
| **SYSCON2 b** | $F1D0_H$ **E** | $E8_H$ | CPU System Configuration Register 2 | $0000_H$ |
| **SYSCON3 b** | $F1D4_H$ **E** | $EA_H$ | CPU System Configuration Register 3 | $0X00_H$ |
| **T0** | $FE50_H$ | $28_H$ | CAPCOM Timer 0 Register | $0000_H$ |
| **T01CON b** | $FF50_H$ | $A8_H$ | CAPCOM Timer 0 and Timer 1 Ctrl. Reg. | $0000_H$ |
| **T0IC b** | $FF9C_H$ | $CE_H$ | CAPCOM Timer 0 Interrupt Ctrl. Reg. | $0000_H$ |
| **T0REL** | $FE54_H$ | $2A_H$ | CAPCOM Timer 0 Reload Register | $0000_H$ |
| **T1** | $FE52_H$ | $29_H$ | CAPCOM Timer 1 Register | $0000_H$ |
| **T14** | $F0D2_H$ **E** | $69_H$ | RTC Timer 14 Register | no |
| **T14REL** | $F0D0_H$ **E** | $68_H$ | RTC Timer 14 Reload Register | no |
| **T1IC b** | $FF9E_H$ | $CF_H$ | CAPCOM Timer 1 Interrupt Ctrl. Reg. | $0000_H$ |
| **T1REL** | $FE56_H$ | $2B_H$ | CAPCOM Timer 1 Reload Register | $0000_H$ |
| **T2** | $FE40_H$ | $20_H$ | GPT1 Timer 2 Register | $0000_H$ |
| **T2CON b** | $FF40_H$ | $A0_H$ | GPT1 Timer 2 Control Register | $0000_H$ |
| **T2IC b** | $FF60_H$ | $B0_H$ | GPT1 Timer 2 Interrupt Control Register | $0000_H$ |
| **T3** | $FE42_H$ | $21_H$ | GPT1 Timer 3 Register | $0000_H$ |
| **T3CON b** | $FF42_H$ | $A1_H$ | GPT1 Timer 3 Control Register | $0000_H$ |
| **T3IC b** | $FF62_H$ | $B1_H$ | GPT1 Timer 3 Interrupt Control Register | $0000_H$ |
| **T4** | $FE44_H$ | $22_H$ | GPT1 Timer 4 Register | $0000_H$ |
| **T4CON b** | $FF44_H$ | $A2_H$ | GPT1 Timer 4 Control Register | $0000_H$ |
| **T4IC b** | $FF64_H$ | $B2_H$ | GPT1 Timer 4 Interrupt Control Register | $0000_H$ |
| **T5** | $FE46_H$ | $23_H$ | GPT2 Timer 5 Register | $0000_H$ |
| **T5CON b** | $FF46_H$ | $A3_H$ | GPT2 Timer 5 Control Register | $0000_H$ |
| **T5IC b** | $FF66_H$ | $B3_H$ | GPT2 Timer 5 Interrupt Control Register | $0000_H$ |
| **T6** | $FE48_H$ | $24_H$ | GPT2 Timer 6 Register | $0000_H$ |
| **T6CON b** | $FF48_H$ | $A4_H$ | GPT2 Timer 6 Control Register | $0000_H$ |
| **T6IC b** | $FF68_H$ | $B4_H$ | GPT2 Timer 6 Interrupt Control Register | $0000_H$ |
| **T7** | $F050_H$ **E** | $28_H$ | CAPCOM Timer 7 Register | $0000_H$ |
| **T78CON b** | $FF20_H$ | $90_H$ | CAPCOM Timer 7 and 8 Ctrl. Reg. | $0000_H$ |
| **T7IC b** | $F17A_H$ **E** | $BD_H$ | CAPCOM Timer 7 Interrupt Ctrl. Reg. | $0000_H$ |
| **T7REL** | $F054_H$ **E** | $2A_H$ | CAPCOM Timer 7 Reload Register | $0000_H$ |

**Table 7 C161CS/JC/JI Registers, Ordered by Name** (cont'd)

| Name | Physical Address | 8-Bit Addr. | Description | Reset Value |
|---|---|---|---|---|
| **T8** | $F052_H$ **E** | $29_H$ | CAPCOM Timer 8 Register | $0000_H$ |
| **T8IC b** | $F17C_H$ **E** | $BE_H$ | CAPCOM Timer 8 Interrupt Ctrl. Reg. | $0000_H$ |
| **T8REL** | $F056_H$ **E** | $2B_H$ | CAPCOM Timer 8 Reload Register | $0000_H$ |
| **TFR b** | $FFAC_H$ | $D6_H$ | Trap Flag Register | $0000_H$ |
| **TRANSSTAT** | $EB1E_H$ **X** | --- | SDLM Transmission Status Register | $0000_H$ |
| **TXCNT** | $EB3C_H$ **X** | --- | SDLM Bus Transmit Byte Counter Reg. | $0000_H$ |
| **TXCPU** | $EB3E_H$ **X** | --- | SDLM CPU Transmit Byte Counter Reg. | $0000_H$ |
| **TXD0** | $EB30_H$ **X** | --- | SDLM Transmit Data Register 0 | $0000_H$ |
| **TXD10** | $EB3A_H$ **X** | --- | SDLM Transmit Data Register 10 | $0000_H$ |
| **TXD2** | $EB32_H$ **X** | --- | SDLM Transmit Data Register 2 | $0000_H$ |
| **TXD4** | $EB34_H$ **X** | --- | SDLM Transmit Data Register 4 | $0000_H$ |
| **TXD6** | $EB36_H$ **X** | --- | SDLM Transmit Data Register 6 | $0000_H$ |
| **TXD8** | $EB38_H$ **X** | --- | SDLM Transmit Data Register 8 | $0000_H$ |
| **TxDELAY** | $EB16_H$ **X** | --- | SDLM Transceiver Delay Register | $0014_H$ |
| **WDT** | $FEAE_H$ | $57_H$ | Watchdog Timer Register (read only) | $0000_H$ |
| **WDTCON b** | $FFAE_H$ | $D7_H$ | Watchdog Timer Control Register | [2] $00XX_H$ |
| **XP0IC b** | $F186_H$ **E** | $C3_H$ | IIC Data Interrupt Control Register | $0000_H$ |
| **XP1IC b** | $F18E_H$ **E** | $C7_H$ | IIC Protocol Interrupt Control Register | $0000_H$ |
| **XP2IC b** | $F196_H$ **E** | $CB_H$ | CAN1 Interrupt Control Register | $0000_H$ |
| **XP3IC b** | $F19E_H$ **E** | $CF_H$ | PLL/RTC Interrupt Control Register | $0000_H$ |
| **XP4IC b** | $F182_H$ **E** | $C1_H$ | ASC1 Transmit Interrupt Ctrl. Reg. | $0000_H$ |
| **XP5IC b** | $F18A_H$ **E** | $C5_H$ | ASC1 Receive Interrupt Control Register | $0000_H$ |
| **XP6IC b** | $F192_H$ **E** | $C9_H$ | ASC1 Error Interrupt Control Register | $0000_H$ |
| **XP7IC b** | $F19A_H$ **E** | $CD_H$ | CAN2/SDLM Interrupt Control Register | $0000_H$ |
| **ZEROS b** | $FF1C_H$ | $8E_H$ | Constant Value 0's Register (read only) | $0000_H$ |

1) The system configuration is selected during reset.

2) The reset value depends on the indicated reset source.

## Absolute Maximum Ratings

**Table 8 Absolute Maximum Rating Parameters**

| Parameter | Symbol | Limit Values | | Unit | Notes |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Storage temperature | $T_{ST}$ | -65 | 150 | °C | – |
| Junction temperature | $T_J$ | -40 | 150 | °C | under bias |
| Voltage on $V_{DD}$ pins with respect to ground ($V_{SS}$) | $V_{DD}$ | -0.5 | 6.5 | V | – |
| Voltage on any pin with respect to ground ($V_{SS}$) | $V_{IN}$ | -0.5 | $V_{DD}$ + 0.5 | V | – |
| Input current on any pin during overload condition | – | -10 | 10 | mA | – |
| Absolute sum of all input currents during overload condition | – | – | \|100\| | mA | – |
| Power dissipation | $P_{DISS}$ | – | 1.5 | W | – |

*Note: Stresses above those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*
*During absolute maximum rating overload conditions ($V_{IN} > V_{DD}$ or $V_{IN} < V_{SS}$) the voltage on $V_{DD}$ pins with respect to ground ($V_{SS}$) must not exceed the values defined by the absolute maximum ratings.*

## Operating Conditions

The following operating conditions must not be exceeded in order to ensure correct operation of the C161CS/JC/JI. All parameters specified in the following sections refer to these operating conditions, unless otherwise noticed.

**Table 9 Operating Condition Parameters**

| Parameter | Symbol | Limit Values | | Unit | Notes |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Digital supply voltage | $V_{DD}$ | 4.5 | 5.5 | V | Active mode, $f_{CPUmax}$ = 25 MHz |
| | | 2.5[1)] | 5.5 | V | PowerDown mode |
| Digital ground voltage | $V_{SS}$ | 0 | | V | Reference voltage |
| Overload current | $I_{OV}$ | – | ±5 | mA | Per pin[2)3)4)] |
| Absolute sum of overload currents | $\Sigma\|I_{OV}\|$ | – | 50 | mA | [3)] |
| External Load Capacitance | $C_L$ | – | 100 | pF | Pin drivers in **fast edge** mode[5)] |
| Ambient temperature | $T_A$ | 0 | 70 | °C | SAB-C161CS/JC/JI … |
| | | -40 | 85 | °C | SAF-C161CS/JC/JI … |
| | | -40 | 125 | °C | SAK-C161CS/JC/JI … |

1) Output voltages and output currents will be reduced when $V_{DD}$ leaves the range defined for active mode.

2) Overload conditions occur if the standard operatings conditions are exceeded, i.e. the voltage on any pin exceeds the specified range (i.e. $V_{OV} > V_{DD}$ + 0.5 V or $V_{OV} < V_{SS}$ - 0.5 V). The absolute sum of input overload currents on all pins may not exceed **50 mA**. The supply voltage must remain within the specified limits. Proper operation is not guaranteed if overload conditions occur on functional pins line XTAL1, $\overline{RD}$, $\overline{WR}$, etc.

3) Not 100% tested, guaranteed by design and characterization.

4) Due to the different port structure of Port 9 (required by the IIC bus specification) the pins of Port 9 can only tolerate positive overload current, i.e. $V_{OV} > V_{SS}$ - 0.5 V.

5) The timing is valid for pin drivers in high current or dynamic current mode. The reduced static output current in dynamic current mode must be respected when designing the system.

### Parameter Interpretation

The parameters listed in the following partly represent the characteristics of the C161CS/JC/JI and partly its demands on the system. To aid in interpreting the parameters right, when evaluating them for a design, they are marked in column “Symbol”:

**CC** (**C**ontroller **C**haracteristics):
The logic of the C161CS/JC/JI will provide signals with the respective timing characteristics.

**SR** (**S**ystem **R**equirement):
The external system must provide signals with the respective timing characteristics to the C161CS/JC/JI.

### DC Characteristics

(Operating Conditions apply)[1)]

| Parameter | Symbol | | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Input low voltage (TTL, all except XTAL1, XTAL3, Port 9) | $V_{IL}$ | SR | -0.5 | 0.2 $V_{DD}$ - 0.1 | V | – |
| Input low voltage XTAL1, XTAL3, Port 9 | $V_{IL2}$ | SR | -0.5 | 0.3 $V_{DD}$ | V | – |
| Input low voltage (Special Threshold) | $V_{ILS}$ | SR | -0.5 | 2.0 | V | – |
| Input high voltage (TTL, all except $\overline{\text{RSTIN}}$, XTAL1, XTAL3, Port 9) | $V_{IH}$ | SR | 0.2 $V_{DD}$ + 0.9 | $V_{DD}$ + 0.5 | V | – |
| Input high voltage $\overline{\text{RSTIN}}$ (when operated as input) | $V_{IH1}$ | SR | 0.6 $V_{DD}$ | $V_{DD}$ + 0.5 | V | – |
| Input high voltage XTAL1, XTAL3, Port 9 | $V_{IH2}$ | SR | 0.7 $V_{DD}$ | $V_{DD}$ + 0.5 | V | – |
| Input high voltage (Special Threshold) | $V_{IHS}$ | SR | 0.8 $V_{DD}$ - 0.2 | $V_{DD}$ + 0.5 | V | – |
| Input Hysteresis (Special Threshold) | HYS | | 400 | – | mV | Series resistance = 0 Ω |
| Output low voltage (PORT0, PORT1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$, $\overline{\text{RSTIN}}$[2)]) | $V_{OL}$ | CC | – | 0.45 | V | $I_{OL}$ = 2.4 mA[3)]<br>$I_{OL}$ = 0.5 mA[4)] |
| Output low voltage (Port 9) | $V_{OL9}$ | CC | – | 0.4 | V | $I_{OL}$ = 3.0 mA |

**DC Characteristics** (cont'd)
(Operating Conditions apply)[1]

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Output low voltage (all other outputs) | $V_{OL1}$ CC | – | 0.45 | V | $I_{OL}$ = 1.6 mA[3]<br>$I_{OL}$ = 1.6 mA[4] |
| Output high voltage[5] (PORT0, PORT1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT, $\overline{RSTOUT}$) | $V_{OH}$ CC | 2.4 | – | V | $I_{OH}$ = -2.4 mA[3]<br>$I_{OH}$ = -0.5 mA[4] |
| | | 0.9 $V_{DD}$ | – | V | $I_{OH}$ = -0.5 mA[3] |
| Output high voltage[5] (all other outputs) | $V_{OH1}$ CC | 2.4 | – | V | $I_{OH}$ = -1.6 mA[3]<br>$I_{OH}$ = -0.5 mA[4] |
| | | 0.9 $V_{DD}$ | – | V | $I_{OH}$ = -0.5 mA[3] |
| Input leakage current (Port 5) | $I_{OZ1}$ CC | – | ±200 | nA | 0 V < $V_{IN}$ < $V_{DD}$ |
| Input leakage current (all other) | $I_{OZ2}$ CC | – | ±500 | nA | 0.45 V < $V_{IN}$ < $V_{DD}$ |
| $\overline{RSTIN}$ inactive current[6] | $I_{RSTH}$[7] | – | -10 | µA | $V_{IN}$ = $V_{IH1}$ |
| $\overline{RSTIN}$ active current[6] | $I_{RSTL}$[8] | -100 | – | µA | $V_{IN}$ = $V_{IL}$ |
| $\overline{READY}$/$\overline{RD}$/$\overline{WR}$ inact. current[9] | $I_{RWH}$[7] | – | -40 | µA | $V_{OUT}$ = 2.4 V |
| $\overline{READY}$/$\overline{RD}$/$\overline{WR}$ active current[9] | $I_{RWL}$[8] | -500 | – | µA | $V_{OUT}$ = $V_{OLmax}$ |
| ALE inactive current[9] | $I_{ALEL}$[7] | – | 40 | µA | $V_{OUT}$ = $V_{OLmax}$ |
| ALE active current[9] | $I_{ALEH}$[8] | 500 | – | µA | $V_{OUT}$ = 2.4 V |
| Port 6 inactive current[9] | $I_{P6H}$[7] | – | -40 | µA | $V_{OUT}$ = 2.4 V |
| Port 6 active current[9] | $I_{P6L}$[8] | -500 | – | µA | $V_{OUT}$ = $V_{OL1max}$ |
| PORT0 configuration current[10] | $I_{P0H}$[7] | – | -10 | µA | $V_{IN}$ = $V_{IHmin}$ |
| | $I_{P0L}$[8] | -100 | – | µA | $V_{IN}$ = $V_{ILmax}$ |
| XTAL1 input current | $I_{IL}$ CC | – | ±20 | µA | 0 V < $V_{IN}$ < $V_{DD}$ |
| Pin capacitance[11] (digital inputs/outputs) | $C_{IO}$ CC | – | 10 | pF | $f$ = 1 MHz<br>$T_A$ = 25 °C |

1) Keeping signal levels within the levels specified in this table, ensures operation without overload conditions. For signal levels outside these specifications also refer to the specification of the overload current $I_{OV}$.

2) Valid in bidirectional reset mode only.

3) This output current may be drawn from (output) pins operating in High Current mode.

4) This output current may be drawn from (output) pins operating in Low Current mode.

5) This specification is not valid for outputs which are switched to open drain mode. In this case the respective output will float and the voltage results from the external circuitry.

6) These parameters describe the $\overline{\text{RSTIN}}$ pullup, which equals a resistance of ca. 50 to 250 kΩ.

7) The maximum current may be drawn while the respective signal line remains inactive.

8) The minimum current must be drawn in order to drive the respective signal line active.

9) This specification is valid during Reset and during Hold-mode or Adapt-mode. During Hold-mode Port 6 pins are only affected, if they are used (configured) for $\overline{\text{CS}}$ output and the open drain function is not enabled. The $\overline{\text{READY}}$-pullup is always active, except for Powerdown mode.

10) This specification is valid during Reset and during Adapt-mode.

11) Not 100% tested, guaranteed by design and characterization.

## Power Consumption C161CS/JC/JI

(Operating Conditions apply)

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Power supply current (active) with all peripherals active | $I_{DD}$ | – | 15 + 2.5 × $f_{CPU}$ | mA | $\overline{\text{RSTIN}}$ = $V_{IL}$ $f_{CPU}$ in [MHz][1] |
| Idle mode supply current with all peripherals active | $I_{IDX}$ | – | 5 + 1.5 × $f_{CPU}$ | mA | $\overline{\text{RSTIN}}$ = $V_{IH1}$ $f_{CPU}$ in [MHz][1] |
| Idle mode supply curr., Main osc, with all peripherals deactivated, PLL off, SDD factor = 32 | $I_{IDOM}$[2] | – | 500 + 50 × $f_{OSC}$ | µA | $\overline{\text{RSTIN}}$ = $V_{IH1}$ $f_{OSC}$ in [MHz][1] |
| Idle mode supply curr., Aux. osc, with all peripherals deactivated, PLL off, SDD factor = 32 | $I_{IDOA}$[2] | – | 100 | µA | $V_{DD}$ = $V_{DDmax}$ $f_{OSC}$ = 32 kHz[3] |
| Sleep and Power-down mode supply current with RTC running on main oscillator | $I_{PDRM}$[2] | – | 200 + 25 × $f_{OSC}$ | µA | $V_{DD}$ = $V_{DDmax}$ $f_{OSC}$ in [MHz][3] |
| Sleep and Power-down mode supply current with RTC disabled | $I_{PDO}$ | – | 50 | µA | $V_{DD}$ = $V_{DDmax}$[3] |

1) The supply current is a function of the operating frequency. This dependency is illustrated in **Figure 10**. These parameters are tested at $V_{DDmax}$ and maximum CPU clock with all outputs disconnected and all inputs at $V_{IL}$ or $V_{IH}$.

2) This parameter is determined mainly by the current consumed by the oscillator (see **Figure 9**). This current, however, is influenced by the external oscillator circuitry (crystal, capacitors). The values given refer to a typical circuitry and may change in case of a not optimized external oscillator circuitry.

3) This parameter is tested including leakage currents. All inputs (including pins configured as inputs) at 0 V to 0.1 V or at $V_{DD}$ - 0.1 V to $V_{DD}$, all outputs (including pins configured as outputs) disconnected.

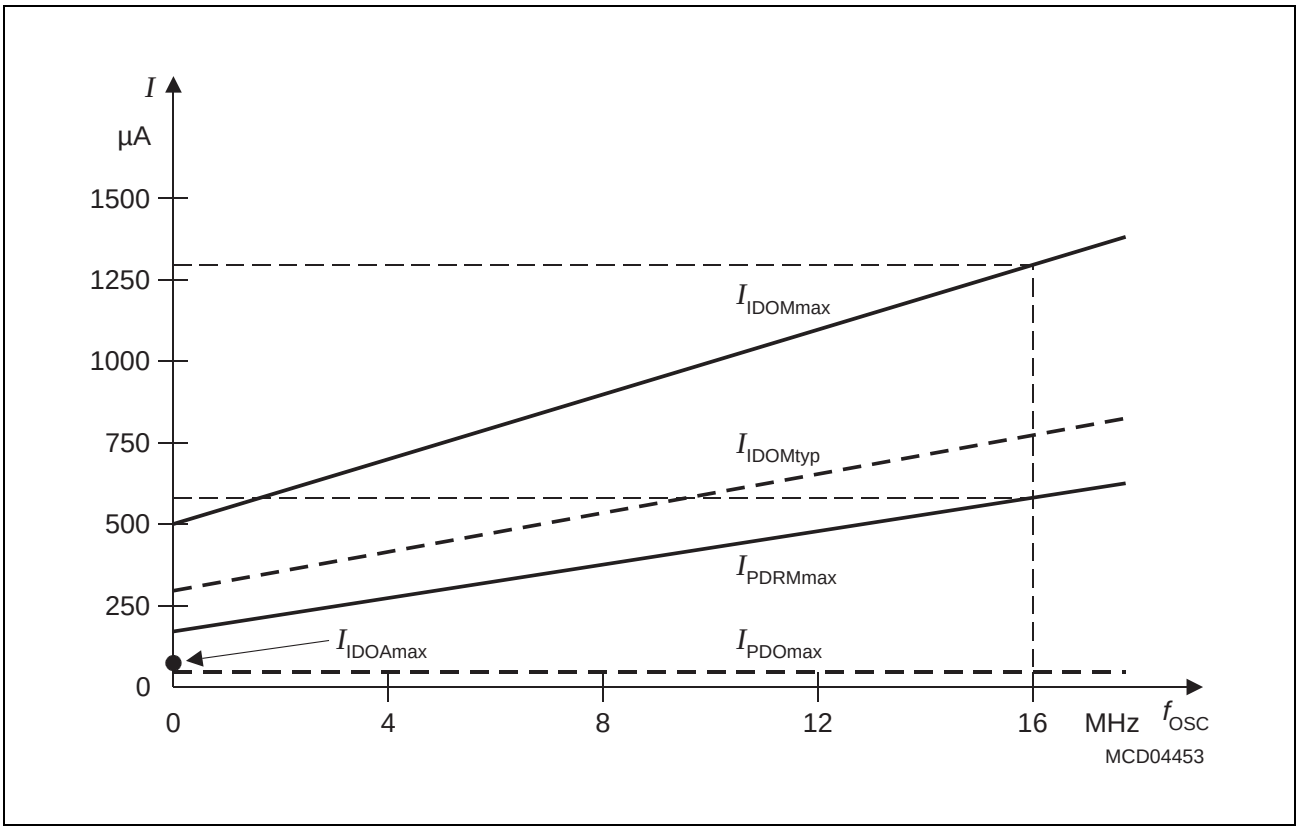



**Figure 9** **Idle and Power Down Supply Current as a Function of Oscillator Frequency**

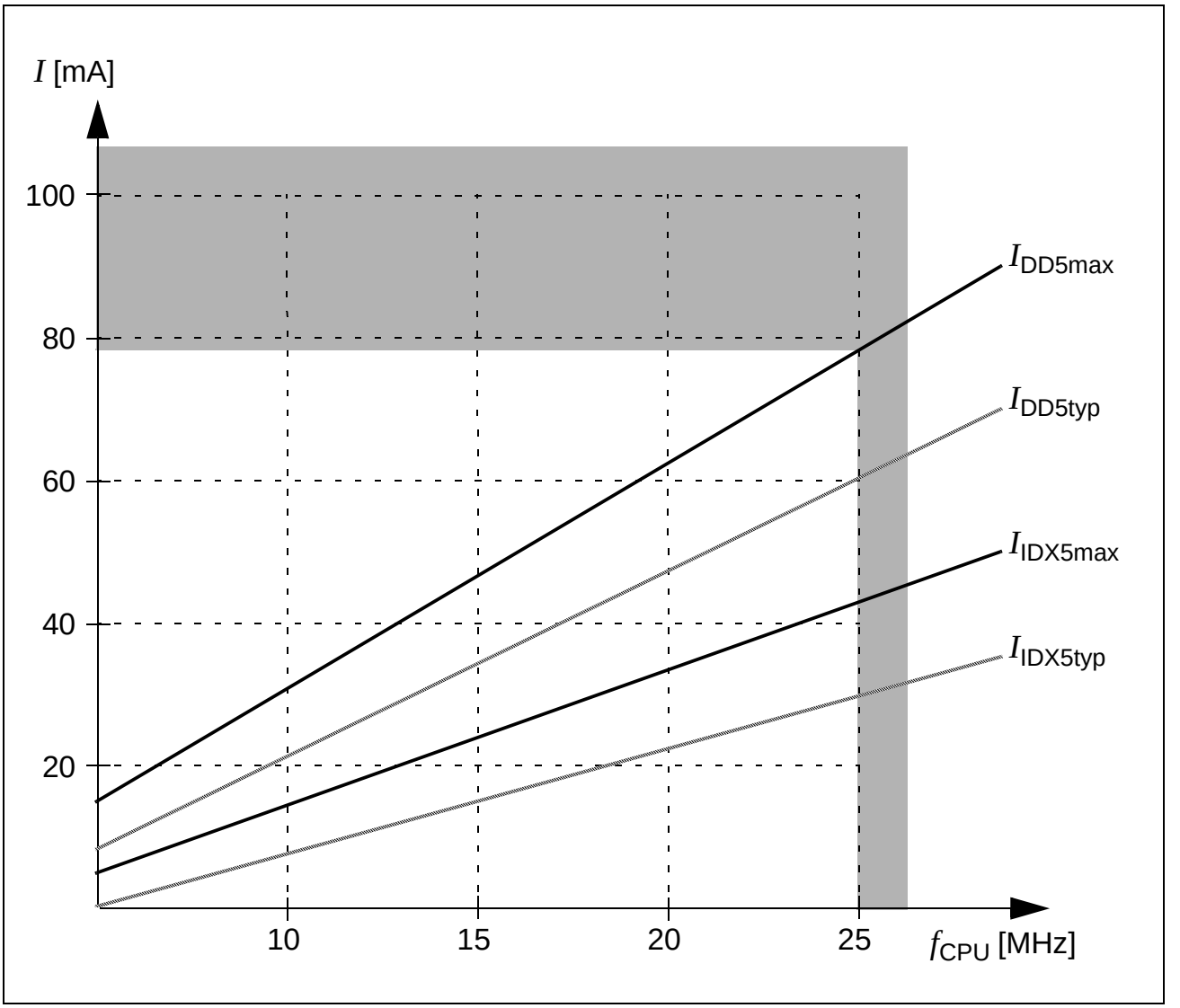



**Figure 10 Supply/Idle Current as a Function of Operating Frequency**

## AC Characteristics
## Definition of Internal Timing

The internal operation of the C161CS/JC/JI is controlled by the internal CPU clock $f_{CPU}$. Both edges of the CPU clock can trigger internal (e.g. pipeline) or external (e.g. bus cycles) operations.

The specification of the external timing (AC Characteristics) therefore depends on the time between two consecutive edges of the CPU clock, called “TCL” (see **Figure 11**).

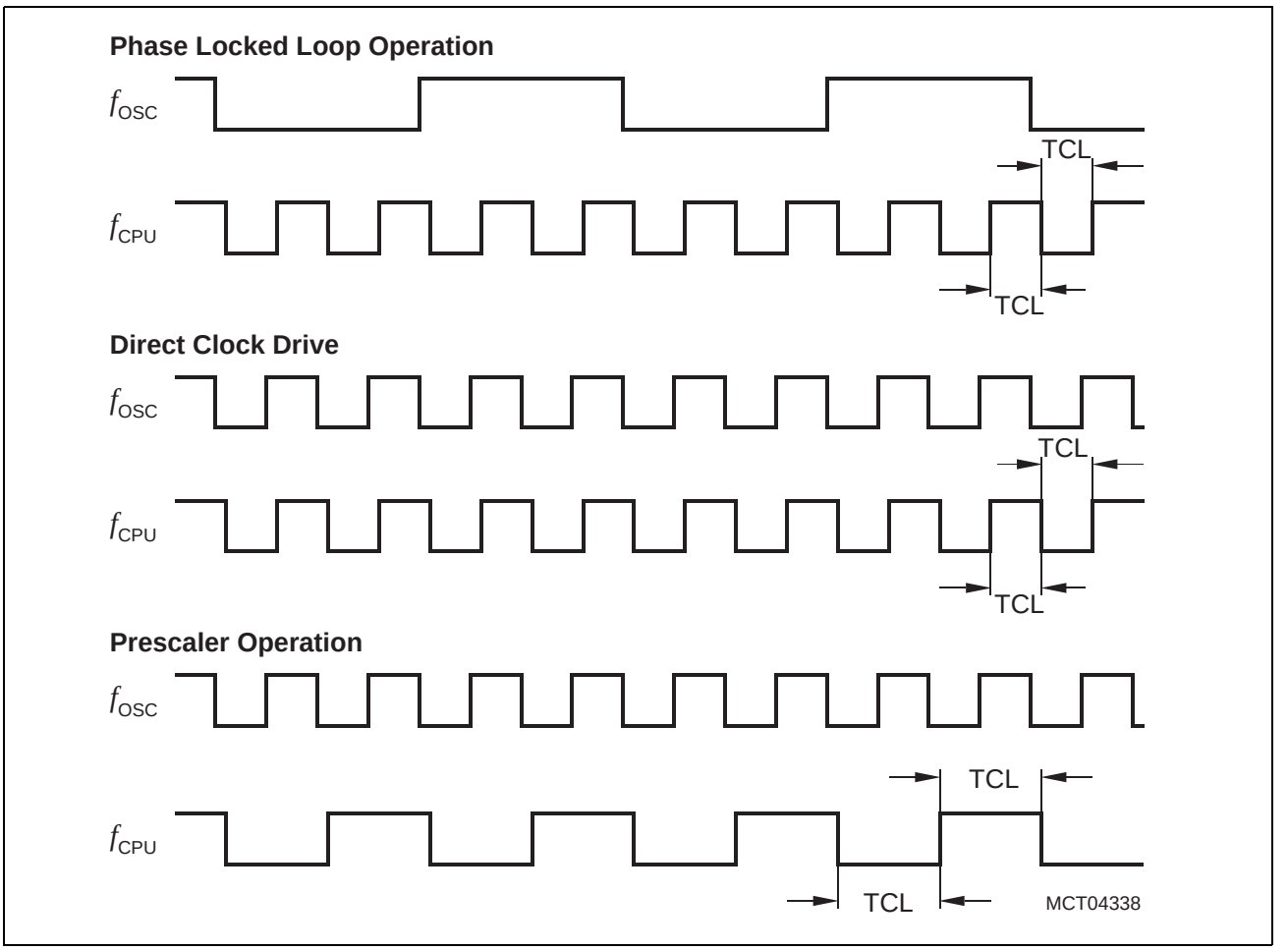



**Figure 11 Generation Mechanisms for the CPU Clock**

The CPU clock signal $f_{CPU}$ can be generated from the oscillator clock signal $f_{OSC}$ via different mechanisms. The duration of TCLs and their variation (and also the derived external timing) depends on the used mechanism to generate $f_{CPU}$. This influence must be regarded when calculating the timings for the C161CS/JC/JI.

*Note: The example for PLL operation shown in the fig. above refers to a PLL factor of 4.*

The used mechanism to generate the basic CPU clock is selected by bitfield CLKCFG in register RP0H.7-5.
Upon a long hardware reset register RP0H is loaded with the logic levels present on the upper half of PORT0 (P0H), i.e. bitfield CLKCFG represents the logic levels on pins

P0.15-13 (P0H.7-5). Register RP0H can be loaded from the upper half of register RSTCON under software control.

**Table 10** associates the combinations of these three bits with the respective clock generation mode.

**Table 10 C161CS/JC/JI Clock Generation Modes**

| CLKCFG (P0H.7-5) | CPU Frequency $f_{CPU} = f_{OSC} \times F$ | External Clock Input Range[1)] | Notes |
|---|---|---|---|
| 1 1 1 | $f_{OSC} \times 4$ | 2.5 to 6.25 MHz | Default configuration |
| 1 1 0 | $f_{OSC} \times 3$ | 3.33 to 8.33 MHz | – |
| 1 0 1 | $f_{OSC} \times 2$ | 5 to 12.5 MHz | – |
| 1 0 0 | $f_{OSC} \times 5$ | 2 to 5 MHz | – |
| 0 1 1 | $f_{OSC} \times 1$ | 1 to 25 MHz | Direct drive[2)] |
| 0 1 0 | $f_{OSC} \times 1.5$ | 6.66 to 16.6 MHz | – |
| 0 0 1 | $f_{OSC} / 2$ | 2 to 50 MHz | CPU clock via prescaler |
| 0 0 0 | $f_{OSC} \times 2.5$ | 4 to 10 MHz | – |

1) The external clock input range refers to a CPU clock range of 10 … 25 MHz.

2) The maximum frequency depends on the duty cycle of the external clock signal.

### Prescaler Operation

When prescaler operation is configured (CLKCFG = $001_B$) the CPU clock is derived from the internal oscillator (input clock signal) by a 2:1 prescaler.
The frequency of $f_{CPU}$ is half the frequency of $f_{OSC}$ and the high and low time of $f_{CPU}$ (i.e. the duration of an individual TCL) is defined by the period of the input clock $f_{OSC}$.

The timings listed in the AC Characteristics that refer to TCLs therefore can be calculated using the period of $f_{OSC}$ for any TCL.

### Phase Locked Loop

When PLL operation is configured (via CLKCFG) the on-chip phase locked loop is enabled and provides the CPU clock (see table above). The PLL multiplies the input frequency by the factor **F** which is selected via the combination of pins P0.15-13 (i.e. $f_{CPU} = f_{OSC} \times$ **F**). With every **F**'th transition of $f_{OSC}$ the PLL circuit synchronizes the CPU clock to the input clock. This synchronization is done smoothly, i.e. the CPU clock frequency does not change abruptly.

Due to this adaptation to the input clock the frequency of $f_{CPU}$ is constantly adjusted so it is locked to $f_{OSC}$. The slight variation causes a jitter of $f_{CPU}$ which also effects the duration of individual TCLs.

The timings listed in the AC Characteristics that refer to TCLs therefore must be calculated using the minimum TCL that is possible under the respective circumstances.

The actual minimum value for TCL depends on the jitter of the PLL. As the PLL is constantly adjusting its output frequency so it corresponds to the applied input frequency (crystal or oscillator) the relative deviation for periods of more than one TCL is lower than for one single TCL (see formula and **Figure 12**).
For a period of $N \times$ TCL the minimum value is computed using the corresponding deviation $D_N$:

$$(N \times TCL)_{min} = N \times TCL_{NOM} - D_N \quad D_N\,[ns] = \pm(13.3 + N \times 6.3) / f_{CPU}\,[MHz],$$

where $N$ = number of consecutive TCLs and $1 \le N \le 40$.

So for a period of 3 TCLs @ 25 MHz (i.e. $N = 3$): $D_3 = (13.3 + 3 \times 6.3) / 25 = 1.288$ ns, and $(3TCL)_{min} = 3TCL_{NOM} - 1.288$ ns = 58.7 ns (@ $f_{CPU}$ = 25 MHz).

This is especially important for bus cycles using waitstates and e.g. for the operation of timers, serial interfaces, etc. For all slower operations and longer periods (e.g. pulse train generation or measurement, lower baudrates, etc.) the deviation caused by the PLL jitter is neglectible.

*Note: For all periods longer than 40 TCL the N = 40 value can be used (see **Figure 12**).*

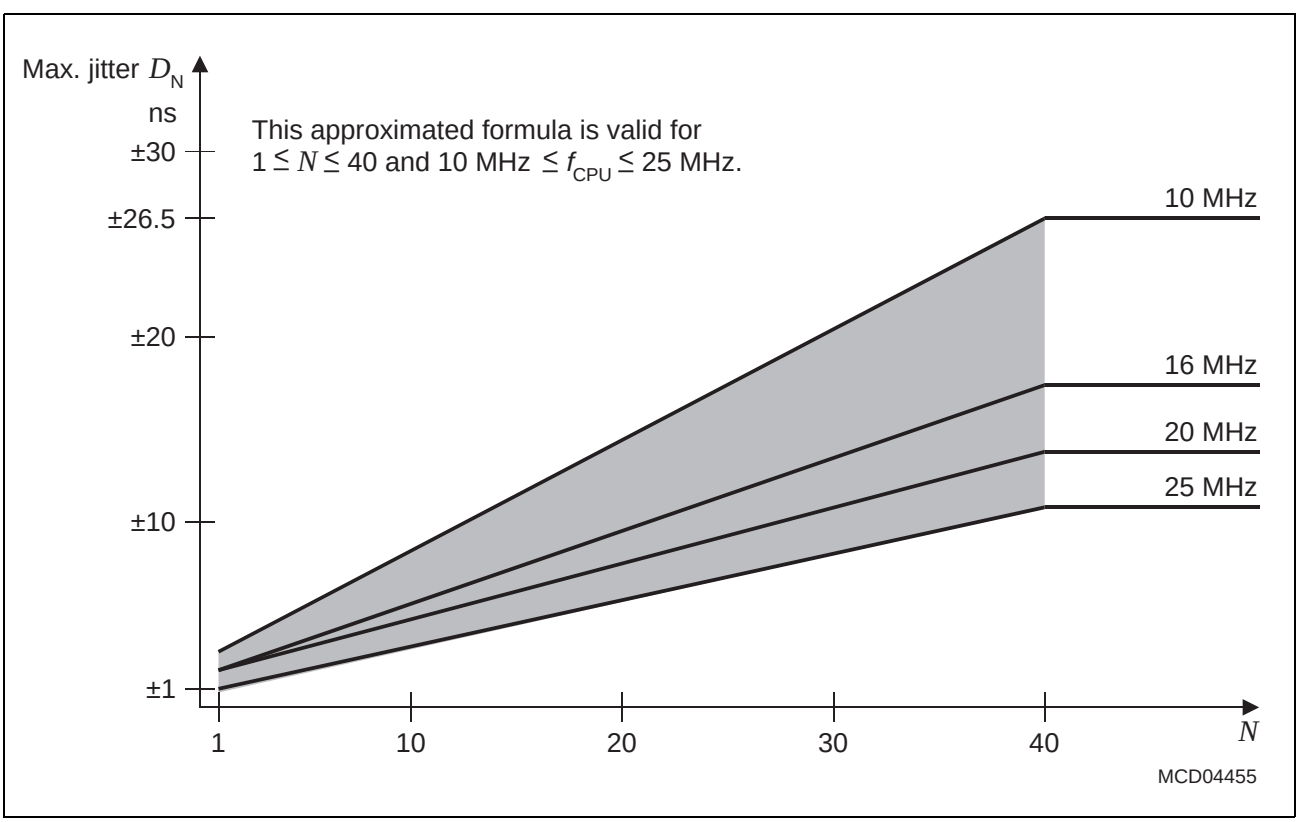



**Figure 12 Approximated Maximum Accumulated PLL Jitter**

**Direct Drive**

When direct drive is configured (CLKCFG = $011_B$) the on-chip phase locked loop is disabled and the CPU clock is directly driven from the internal oscillator with the input clock signal.
The frequency of $f_{CPU}$ directly follows the frequency of $f_{OSC}$ so the high and low time of $f_{CPU}$ (i.e. the duration of an individual TCL) is defined by the duty cycle of the input clock $f_{OSC}$.
The timings listed below that refer to TCLs therefore must be calculated using the minimum TCL that is possible under the respective circumstances. This minimum value can be calculated via the following formula:

$TCL_{min} = 1/f_{OSC} \times DC_{min}$ (DC = duty cycle)

For two consecutive TCLs the deviation caused by the duty cycle of $f_{OSC}$ is compensated so the duration of 2TCL is always $1/f_{OSC}$. The minimum value $TCL_{min}$ therefore has to be used only once for timings that require an odd number of TCLs (1, 3, …). Timings that require an even number of TCLs (2, 4, …) may use the formula $2TCL = 1/f_{OSC}$.

*Note: The address float timings in Multiplexed bus mode ($t_{11}$ and $t_{45}$) use the maximum duration of TCL ($TCL_{max} = 1/f_{OSC} \times DC_{max}$) instead of $TCL_{min}$.*

## AC Characteristics

### External Clock Drive XTAL1 (Main Oscillator)

(Operating Conditions apply)

**Table 11 External Clock Drive Characteristics**

| Parameter | Symbol | | Direct Drive 1:1 | | Prescaler 2:1 | | PLL 1:N | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | min. | max. | |
| Oscillator period | $t_{OSCM}$ | SR | 40 | – | 20 | – | 60[1] | 500[1] | ns |
| High time[2] | $t_1$ | SR | 20[3] | – | 6 | – | 10 | – | ns |
| Low time[2] | $t_2$ | SR | 20[3] | – | 6 | – | 10 | – | ns |
| Rise time[2] | $t_3$ | SR | – | 10 | – | 6 | – | 10 | ns |
| Fall time[2] | $t_4$ | SR | – | 10 | – | 6 | – | 10 | ns |

1) The minimum and maximum oscillator periods for PLL operation depend on the selected CPU clock generation mode. Please see respective table above.

2) The clock input signal must reach the defined levels $V_{IL2}$ and $V_{IH2}$.

3) The minimum high and low time refers to a duty cycle of 50%. The maximum operating frequency ($f_{CPU}$) in direct drive mode depends on the duty cycle of the clock input signal.

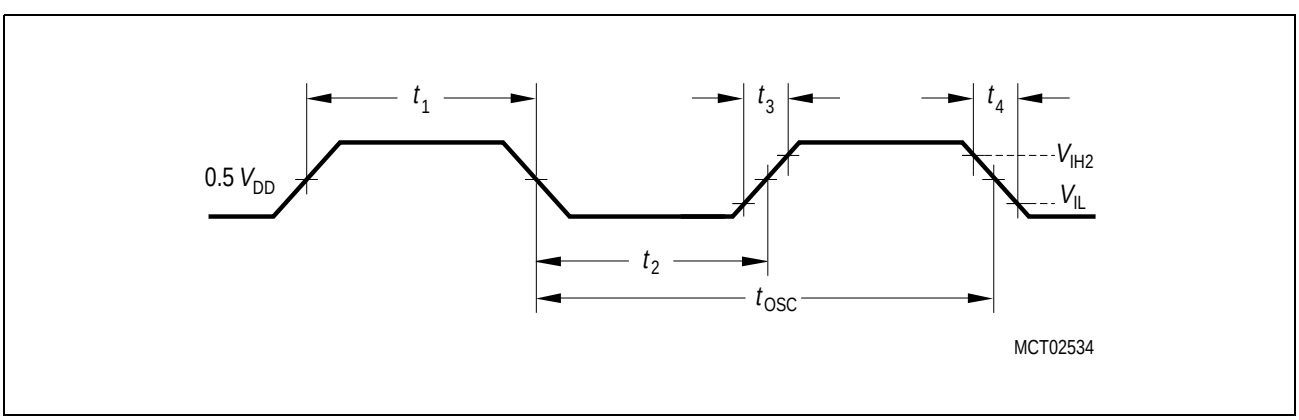



**Figure 13 External Clock Drive XTAL1**

*Note: If the on-chip oscillator is used together with a crystal, the oscillator frequency is limited to a range of 4 MHz to 16 MHz.*
*It is strongly recommended to measure the oscillation allowance (or margin) in the final target system (layout) to determine the optimum parameters for the oscillator operation. Please refer to the limits specified by the crystal supplier.*
*When driven by an external clock signal it will accept the specified frequency range. Operation at lower input frequencies is possible but is guaranteed by design only (not 100% tested).*

## AC Characteristics

**External Clock Drive XTAL3** (Auxiliary Oscillator)

(Operating Conditions apply)

**Table 12 AC Characteristics**

| Parameter | Symbol | | Optimum Input Clock = 32 kHz | | Variable Input Clock 1 / $t_{OSCA}$ = 10 to 50 kHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| Oscillator period | $t_{OSCA}$ | SR | 31 | 31 | 20 | 100 | µs |
| High time | $t_1$ | SR | 6[1)] | – | $0.2 \times t_{OSCA}$[1)] | – | µs |
| Low time | $t_2$ | SR | 6[1)] | – | $0.2 \times t_{OSCA}$[1)] | – | µs |
| Rise time | $t_3$ | SR | – | 12 | – | $0.4 \times t_{OSCA}$ | µs |
| Fall time | $t_4$ | SR | – | 12 | – | $0.4 \times t_{OSCA}$ | µs |

1) The clock input signal must reach the defined levels $V_{IL}$ and $V_{IH2}$.

*Note: The auxiliary oscillator is optimized for oscillation with a crystal at a frequency of 32 kHz. When driven by an external clock signal it will accept the specified frequency range.*
*Operation at lower input frequencies is possible but is guaranteed by design only (not 100% tested).*

## A/D Converter Characteristics

(Operating Conditions apply)

**Table 13 A/D Converter Characteristics**

| Parameter | Symbol | | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Analog reference supply | $V_{AREF}$ | SR | 4.0 | $V_{DD}$ + 0.1 | V | 1) |
| Analog reference ground | $V_{AGND}$ | SR | $V_{SS}$ - 0.1 | $V_{SS}$ + 0.2 | V | |
| Analog input voltage range | $V_{AIN}$ | SR | $V_{AGND}$ | $V_{AREF}$ | V | 2) |
| Basic clock frequency | $f_{BC}$ | | 0.5 | 6.25 | MHz | 3) |
| Conversion time | $t_C$ | CC | – | 40 $t_{BC}$ + $t_S$ + 2$t_{CPU}$ | – | 4) $t_{CPU} = 1 / f_{CPU}$ |
| Calibration time after reset | $t_{CAL}$ | CC | – | 3328 $t_{BC}$ | – | 5) |
| Total unadjusted error | TUE | CC | – | ±2 | LSB | 1) |
| Internal resistance of reference voltage source | $R_{AREF}$ | SR | – | $t_{BC}$ / 60 - 0.25 | kΩ | $t_{BC}$ in [ns]6)7) |
| Internal resistance of analog source | $R_{ASRC}$ | SR | – | $t_S$ / 450 - 0.25 | kΩ | $t_S$ in [ns]7)8) |
| ADC input capacitance | $C_{AIN}$ | CC | – | 33 | pF | 7) |

1) TUE is tested at $V_{AREF}$ = 5.0 V, $V_{AGND}$ = 0 V, $V_{DD}$ = 4.9 V. It is guaranteed by design for all other voltages within the defined voltage range.
If the analog reference supply voltage exceeds the power supply voltage by up to 0.2 V (i.e. $V_{AREF}$ = $V_{DD}$ = +0.2 V) the maximum TUE is increased to ±3 LSB. This range is not 100% tested.
The specified TUE is guaranteed only if the absolute sum of input overload currents on Port 5 pins (see $I_{OV}$ specification) does not exceed 10 mA.
During the reset calibration sequence the maximum TUE may be ±4 LSB.

2) $V_{AIN}$ may exceed $V_{AGND}$ or $V_{AREF}$ up to the absolute maximum ratings. However, the conversion result in these cases will be $X000_H$ or $X3FF_H$, respectively.

3) The limit values for $f_{BC}$ must not be exceeded when selecting the CPU frequency and the ADCTC setting.

4) This parameter includes the sample time $t_S$, the time for determining the digital result and the time to load the result register with the conversion result.
Values for the basic clock $t_{BC}$ depend on programming and can be taken from **Table 14**.
This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.

5) During the reset calibration conversions can be executed (with the current accuracy). The time required for these conversions is added to the total reset calibration time.

6) During the conversion the ADC's capacitance must be repeatedly charged or discharged. The internal resistance of the reference voltage source must allow the capacitance to reach its respective voltage level within each conversion step. The maximum internal resistance results from the programmed conversion timing.

7) Not 100% tested, guaranteed by design and characterization.

8) During the sample time the input capacitance $C_{AIN}$ can be charged/discharged by the external source. The internal resistance of the analog source must allow the capacitance to reach its final voltage level within $t_S$. After the end of the sample time $t_S$, changes of the analog input voltage have no effect on the conversion result.
Values for the sample time $t_S$ depend on programming and can be taken from **Table 14**.

Sample time and conversion time of the C161CS/JC/JI's A/D Converter are programmable. **Table 14** should be used to calculate the above timings.
The limit values for $f_{BC}$ must not be exceeded when selecting ADCTC.

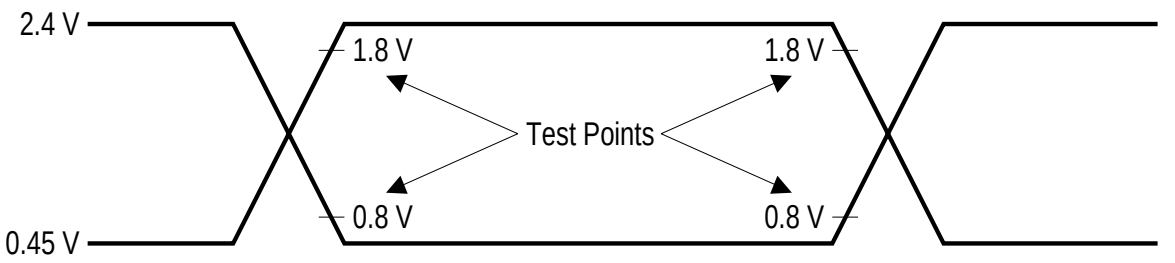
**Table 14 A/D Converter Computation Table**

| ADCON.15\|14 (ADCTC) | A/D Converter Basic Clock $f_{BC}$ | ADCON.13\|12 (ADSTC) | Sample time $t_S$ |
|---|---|---|---|
| 00 | $f_{CPU}$ / 4 | 00 | $t_{BC} \times 8$ |
| 01 | $f_{CPU}$ / 2 | 01 | $t_{BC} \times 16$ |
| 10 | $f_{CPU}$ / 16 | 10 | $t_{BC} \times 32$ |
| 11 | $f_{CPU}$ / 8 | 11 | $t_{BC} \times 64$ |

Converter Timing Example:

| | | |
|---|---|---|
| Assumptions: | $f_{CPU}$ | = 25 MHz (i.e. $t_{CPU}$ = 40 ns), ADCTC = '00', ADSTC = '00'. |
| Basic clock | $f_{BC}$ | = $f_{CPU}$ / 4 = 6.25 MHz, i.e. $t_{BC}$ = 160 ns. |
| Sample time | $t_S$ | = $t_{BC} \times 8$ = 1280 ns. |
| Conversion time | $t_C$ | = $t_S$ + 40 $t_{BC}$ + 2 $t_{CPU}$ = (1280 + 6400 + 80) ns = 7.8 µs. |

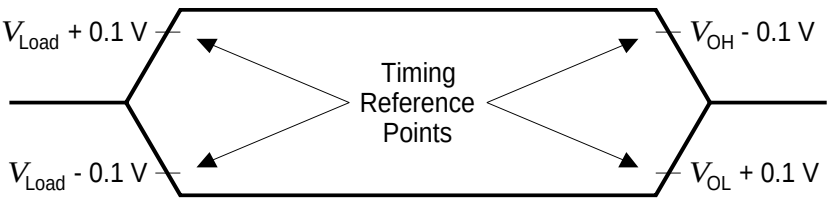
## Testing Waveforms




AC inputs during testing are driven at 2.4 V for a logic ‘1’ and 0.45 V for a logic ‘0’.
Timing measurements are made at $V_{IH}$ min for a logic ‘1’ and $V_{IL}$ max for a logic ‘0’.

MCA04414

**Figure 14 Input Output Waveforms**




For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, but begins to float when a 100 mV change from the loaded $V_{OH}$ / $V_{OL}$ level occurs ($I_{OH}$ / $I_{OL}$ = 20 mA).

MCA00763

**Figure 15 Float Waveforms**

### Memory Cycle Variables

The timing tables below use three variables which are derived from the BUSCONx registers and represent the special characteristics of the programmed memory cycle. The following table describes, how these variables are to be computed.

**Table 15 Memory Cycle Variables**

| Description | Symbol | Values |
|---|---|---|
| ALE Extension | $t_A$ | TCL × <ALECTL> |
| Memory Cycle Time Waitstates | $t_C$ | 2TCL × (15 - <MCTC>) |
| Memory Tristate Time | $t_F$ | 2TCL × (1 - <MTTC>) |

*Note: Please respect the maximum operating frequency of the respective derivative.*

### AC Characteristics

### Multiplexed Bus

(Operating Conditions apply)

ALE cycle time = 6 TCL + 2$t_A$ + $t_C$ + $t_F$ (120 ns at 25 MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| ALE high time | $t_5$ | CC | 10 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| Address setup to ALE | $t_6$ | CC | 4 + $t_A$ | – | TCL - 16 + $t_A$ | – | ns |
| Address hold after ALE | $t_7$ | CC | 10 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay) | $t_8$ | CC | 10 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay) | $t_9$ | CC | -10 + $t_A$ | – | -10 + $t_A$ | – | ns |
| Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay) | $t_{10}$ | CC | – | 6 | – | 6 | ns |
| Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay) | $t_{11}$ | CC | – | 26 | – | TCL + 6 | ns |
| $\overline{RD}$, $\overline{WR}$ low time (with RW-delay) | $t_{12}$ | CC | 30 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |

**Multiplexed Bus** (cont'd)
(Operating Conditions apply)
ALE cycle time = 6 TCL + $2t_A$ + $t_C$ + $t_F$ (120 ns at 25 MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| $\overline{RD}$, $\overline{WR}$ low time (no RW-delay) | $t_{13}$ | CC | 50 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |
| $\overline{RD}$ to valid data in (with RW-delay) | $t_{14}$ | SR | – | 20 + $t_C$ | – | 2TCL - 20 + $t_C$ | ns |
| $\overline{RD}$ to valid data in (no RW-delay) | $t_{15}$ | SR | – | 40 + $t_C$ | – | 3TCL - 20 + $t_C$ | ns |
| ALE low to valid data in | $t_{16}$ | SR | – | 40 + $t_A$ + $t_C$ | – | 3TCL - 20 + $t_A$ + $t_C$ | ns |
| Address to valid data in | $t_{17}$ | SR | – | 50 + $2t_A$ + $t_C$ | – | 4TCL - 30 + $2t_A$ + $t_C$ | ns |
| Data hold after $\overline{RD}$ rising edge | $t_{18}$ | SR | 0 | – | 0 | – | ns |
| Data float after $\overline{RD}$ | $t_{19}$ | SR | – | 26 + $t_F$ | – | 2TCL - 14 + $t_F$ | ns |
| Data valid to $\overline{WR}$ | $t_{22}$ | CC | 20 + $t_C$ | – | 2TCL - 20 + $t_C$ | – | ns |
| Data hold after $\overline{WR}$ | $t_{23}$ | CC | 26 + $t_F$ | – | 2TCL - 14 + $t_F$ | – | ns |
| ALE rising edge after $\overline{RD}$, $\overline{WR}$ | $t_{25}$ | CC | 26 + $t_F$ | – | 2TCL - 14 + $t_F$ | – | ns |
| Address hold after $\overline{RD}$, $\overline{WR}$ | $t_{27}$ | CC | 26 + $t_F$ | – | 2TCL - 14 + $t_F$ | – | ns |
| ALE falling edge to $\overline{CS}$[1)] | $t_{38}$ | CC | -4 - $t_A$ | 10 - $t_A$ | -4 - $t_A$ | 10 - $t_A$ | ns |
| $\overline{CS}$ low to Valid Data In[1)] | $t_{39}$ | SR | – | 40 + $t_C$ + $2t_A$ | – | 3TCL - 20 + $t_C$ + $2t_A$ | ns |
| $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$[1)] | $t_{40}$ | CC | 46 + $t_F$ | – | 3TCL - 14 + $t_F$ | – | ns |
| ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay) | $t_{42}$ | CC | 16 + $t_A$ | – | TCL - 4 + $t_A$ | – | ns |

**Multiplexed Bus** (cont'd)
(Operating Conditions apply)
ALE cycle time = 6 TCL + $2t_A$ + $t_C$ + $t_F$ (120 ns at 25 MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| ALE fall. edge to $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ (no RW delay) | $t_{43}$ | CC | -4 + $t_A$ | – | -4 + $t_A$ | – | ns |
| Address float after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ (with RW delay) | $t_{44}$ | CC | – | 0 | – | 0 | ns |
| Address float after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ (no RW delay) | $t_{45}$ | CC | – | 20 | – | TCL | ns |
| $\overline{\text{RdCS}}$ to Valid Data In (with RW delay) | $t_{46}$ | SR | – | 16 + $t_C$ | – | 2TCL - 24 + $t_C$ | ns |
| $\overline{\text{RdCS}}$ to Valid Data In (no RW delay) | $t_{47}$ | SR | – | 36 + $t_C$ | – | 3TCL - 24 + $t_C$ | ns |
| $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ Low Time (with RW delay) | $t_{48}$ | CC | 30 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |
| $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ Low Time (no RW delay) | $t_{49}$ | CC | 50 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |
| Data valid to $\overline{\text{WrCS}}$ | $t_{50}$ | CC | 26 + $t_C$ | – | 2TCL - 14 + $t_C$ | – | ns |
| Data hold after $\overline{\text{RdCS}}$ | $t_{51}$ | SR | 0 | – | 0 | – | ns |
| Data float after $\overline{\text{RdCS}}$ | $t_{52}$ | SR | – | 20 + $t_F$ | – | 2TCL - 20 + $t_F$ | ns |
| Address hold after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ | $t_{54}$ | CC | 20 + $t_F$ | – | 2TCL - 20 + $t_F$ | – | ns |
| Data hold after $\overline{\text{WrCS}}$ | $t_{56}$ | CC | 20 + $t_F$ | – | 2TCL - 20 + $t_F$ | – | ns |

[1] These parameters refer to the latched chip select signals ($\overline{\text{CSxL}}$). The early chip select signals ($\overline{\text{CSxE}}$) are specified together with the address and signal $\overline{\text{BHE}}$ (see figures below).

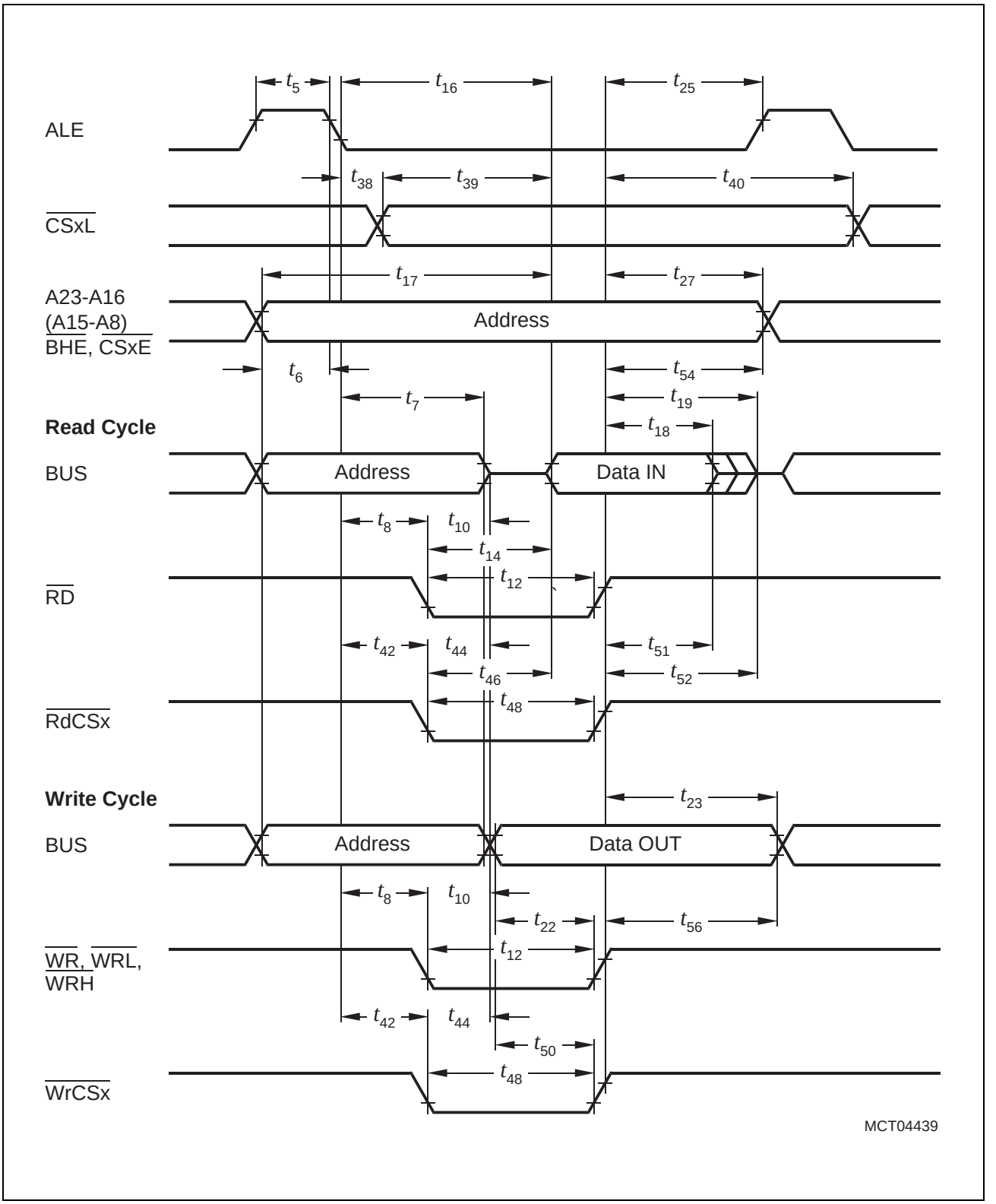



**Figure 16 External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE**

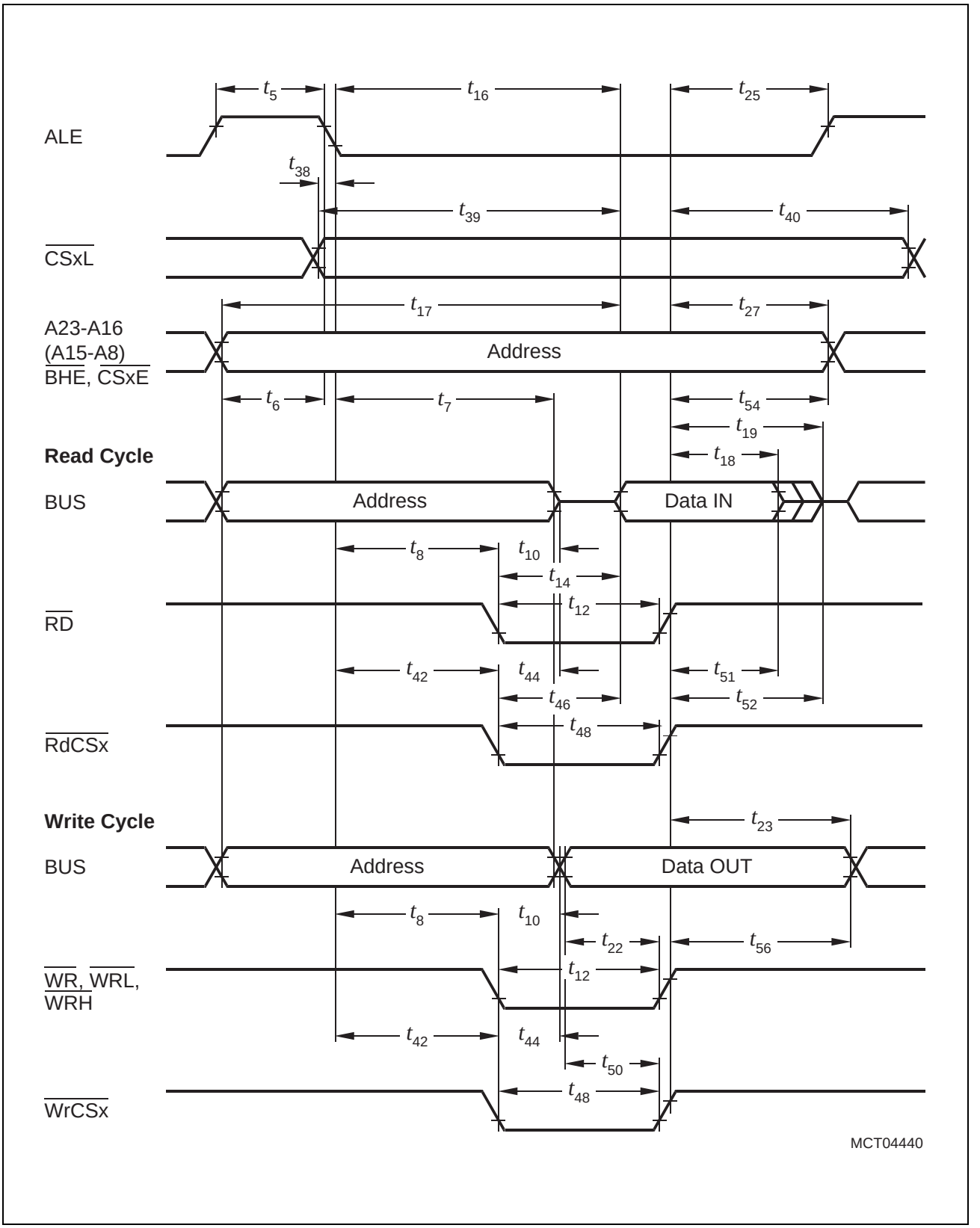



**Figure 17 External Memory Cycle:
Multiplexed Bus, With Read/Write Delay, Extended ALE**

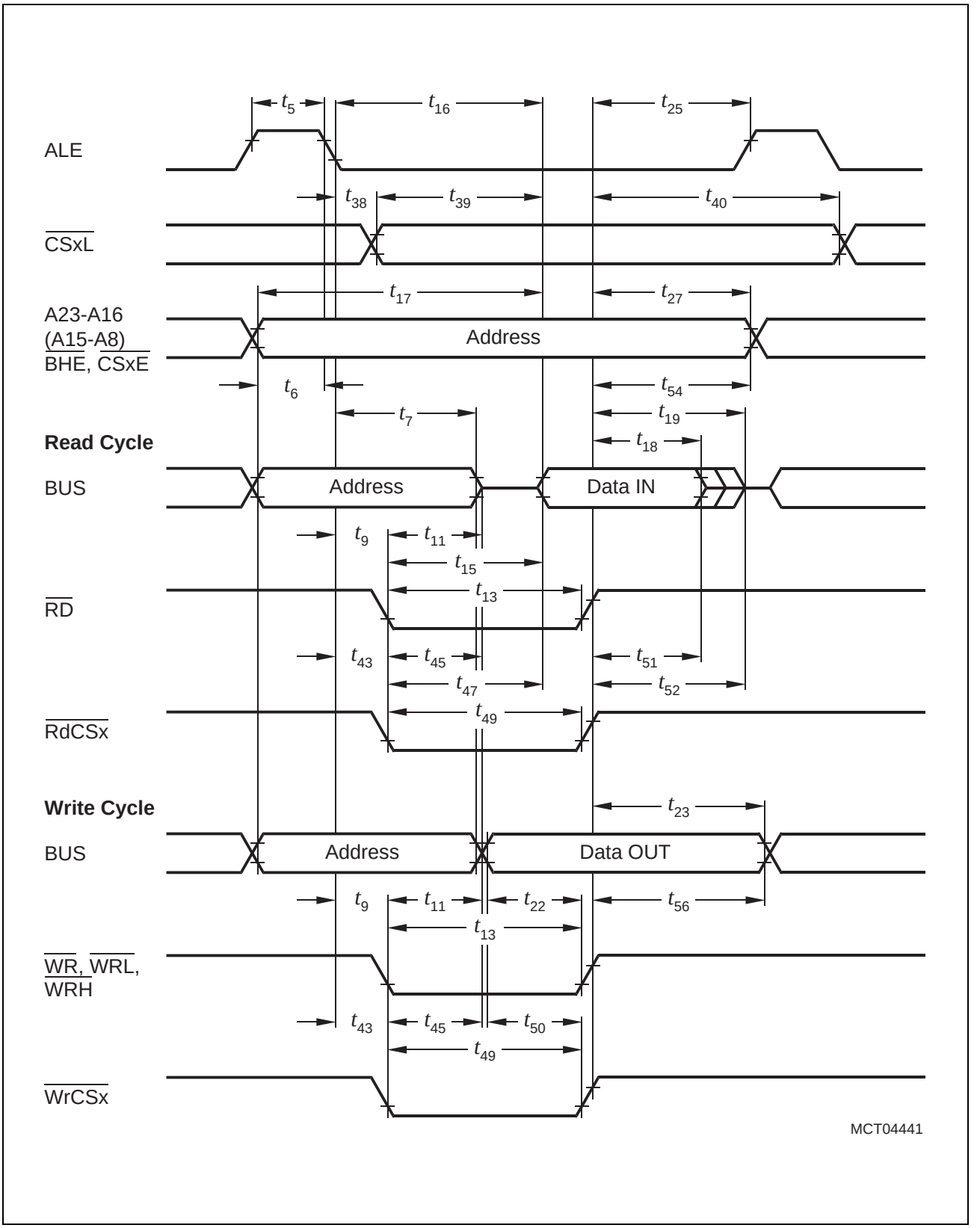



**Figure 18 External Memory Cycle:
Multiplexed Bus, No Read/Write Delay, Normal ALE**

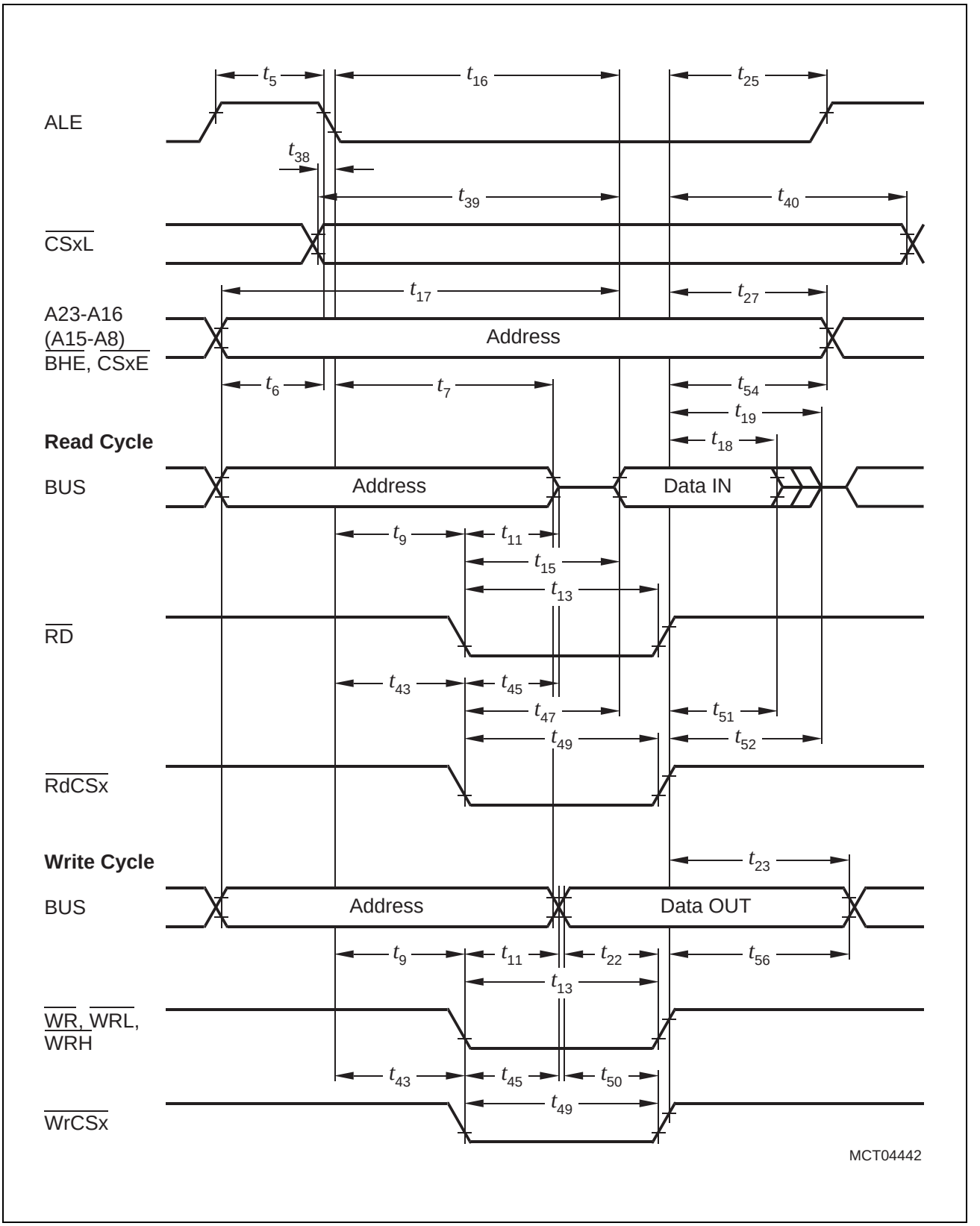



**Figure 19 External Memory Cycle:
Multiplexed Bus, No Read/Write Delay, Extended ALE**

## AC Characteristics

### Demultiplexed Bus

(Operating Conditions apply)
ALE cycle time = 4 TCL + 2$t_A$ + $t_C$ + $t_F$ (80 ns at 25 MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| ALE high time | $t_5$ | CC | 10 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| Address setup to ALE | $t_6$ | CC | 4 + $t_A$ | – | TCL - 16 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay) | $t_8$ | CC | 10 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay) | $t_9$ | CC | -10 + $t_A$ | – | -10 + $t_A$ | – | ns |
| $\overline{RD}$, $\overline{WR}$ low time (with RW-delay) | $t_{12}$ | CC | 30 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |
| $\overline{RD}$, $\overline{WR}$ low time (no RW-delay) | $t_{13}$ | CC | 50 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |
| $\overline{RD}$ to valid data in (with RW-delay) | $t_{14}$ | SR | – | 20 + $t_C$ | – | 2TCL - 20 + $t_C$ | ns |
| $\overline{RD}$ to valid data in (no RW-delay) | $t_{15}$ | SR | – | 40 + $t_C$ | – | 3TCL - 20 + $t_C$ | ns |
| ALE low to valid data in | $t_{16}$ | SR | – | 40 + $t_A$ + $t_C$ | – | 3TCL - 20 + $t_A$ + $t_C$ | ns |
| Address to valid data in | $t_{17}$ | SR | – | 50 + 2$t_A$ + $t_C$ | – | 4TCL - 30 + 2$t_A$ + $t_C$ | ns |
| Data hold after $\overline{RD}$ rising edge | $t_{18}$ | SR | 0 | – | 0 | – | ns |
| Data float after $\overline{RD}$ rising edge (with RW-delay[1)]) | $t_{20}$ | SR | – | 26 + 2$t_A$ + $t_F$[1)] | – | 2TCL - 14 + 22$t_A$ + $t_F$[1)] | ns |
| Data float after $\overline{RD}$ rising edge (no RW-delay[1)]) | $t_{21}$ | SR | – | 10 + 2$t_A$ + $t_F$[1)] | – | TCL - 10 + 22$t_A$ + $t_F$[1)] | ns |

**Demultiplexed Bus** (cont'd)
(Operating Conditions apply)
ALE cycle time = 4 TCL + $2t_A$ + $t_C$ + $t_F$ (80 ns at 25 MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| Data valid to $\overline{WR}$ | $t_{22}$ | CC | 20 + $t_C$ | – | 2TCL - 20 + $t_C$ | – | ns |
| Data hold after $\overline{WR}$ | $t_{24}$ | CC | 10 + $t_F$ | – | TCL - 10 + $t_F$ | – | ns |
| ALE rising edge after $\overline{RD}$, $\overline{WR}$ | $t_{26}$ | CC | -10 + $t_F$ | – | -10 + $t_F$ | – | ns |
| Address hold after $\overline{WR}$[2] | $t_{28}$ | CC | 0 + $t_F$ | – | 0 + $t_F$ | – | ns |
| ALE falling edge to $\overline{CS}$[3] | $t_{38}$ | CC | -4 - $t_A$ | 10 - $t_A$ | -4 - $t_A$ | 10 - $t_A$ | ns |
| $\overline{CS}$ low to Valid Data In[3] | $t_{39}$ | SR | – | 40 + $t_C$ + $2t_A$ | – | 3TCL - 20 + $t_C$ + $2t_A$ | ns |
| $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$[3] | $t_{41}$ | CC | 6 + $t_F$ | – | TCL - 14 + $t_F$ | – | ns |
| ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW-delay) | $t_{42}$ | CC | 16 + $t_A$ | – | TCL - 4 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW-delay) | $t_{43}$ | CC | -4 + $t_A$ | – | -4 + $t_A$ | – | ns |
| $\overline{RdCS}$ to Valid Data In (with RW-delay) | $t_{46}$ | SR | – | 16 + $t_C$ | – | 2TCL - 24 + $t_C$ | ns |
| $\overline{RdCS}$ to Valid Data In (no RW-delay) | $t_{47}$ | SR | – | 36 + $t_C$ | – | 3TCL - 24 + $t_C$ | ns |
| $\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW-delay) | $t_{48}$ | CC | 30 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |
| $\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW-delay) | $t_{49}$ | CC | 50 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |
| Data valid to $\overline{WrCS}$ | $t_{50}$ | CC | 26 + $t_C$ | – | 2TCL - 14 + $t_C$ | – | ns |
| Data hold after $\overline{RdCS}$ | $t_{51}$ | SR | 0 | – | 0 | – | ns |
| Data float after $\overline{RdCS}$ (with RW-delay)[1] | $t_{53}$ | SR | – | 20 + $t_F$ | – | 2TCL - 20 + $2t_A$ + $t_F$[1] | ns |

**Demultiplexed Bus** (cont'd)
(Operating Conditions apply)
ALE cycle time = 4 TCL + 2$t_A$ + $t_C$ + $t_F$ (80 ns at 25 MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| Data float after $\overline{\text{RdCS}}$ (no RW-delay)[1] | $t_{68}$ | SR | – | 0 + $t_F$ | – | TCL - 20 + 2$t_A$ + $t_F$[1] | ns |
| Address hold after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ | $t_{55}$ | CC | -6 + $t_F$ | – | -6 + $t_F$ | – | ns |
| Data hold after $\overline{\text{WrCS}}$ | $t_{57}$ | CC | 6 + $t_F$ | – | TCL - 14 + $t_F$ | – | ns |

1) RW-delay and $t_A$ refer to the next following bus cycle (including an access to an on-chip X-Peripheral).

2) Read data are latched with the same clock edge that triggers the address change and the rising $\overline{\text{RD}}$ edge. Therefore address changes before the end of $\overline{\text{RD}}$ have no impact on read cycles.

3) These parameters refer to the latched chip select signals ($\overline{\text{CSxL}}$). The early chip select signals ($\overline{\text{CSxE}}$) are specified together with the address and signal $\overline{\text{BHE}}$ (see figures below).

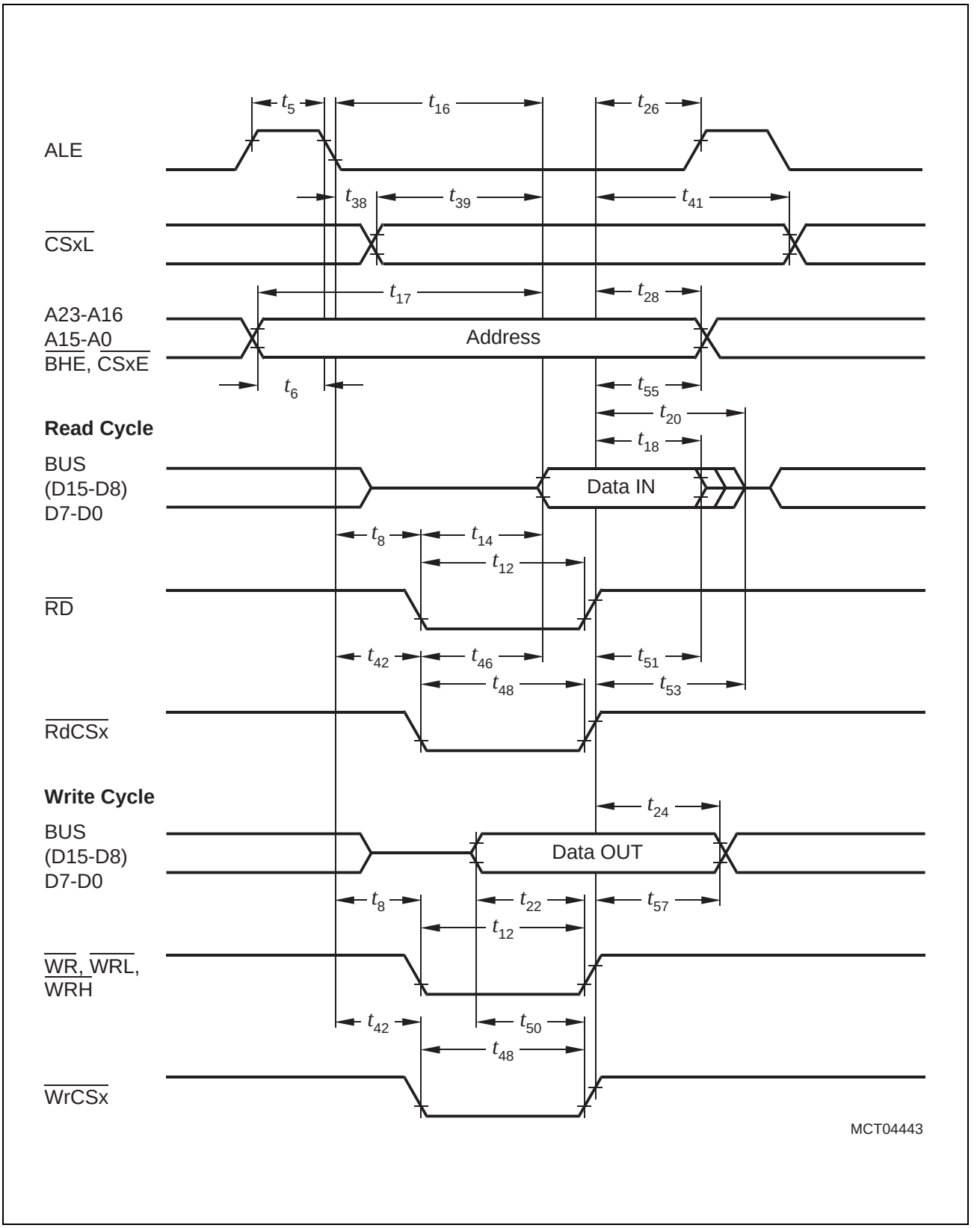



**Figure 20 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Normal ALE**

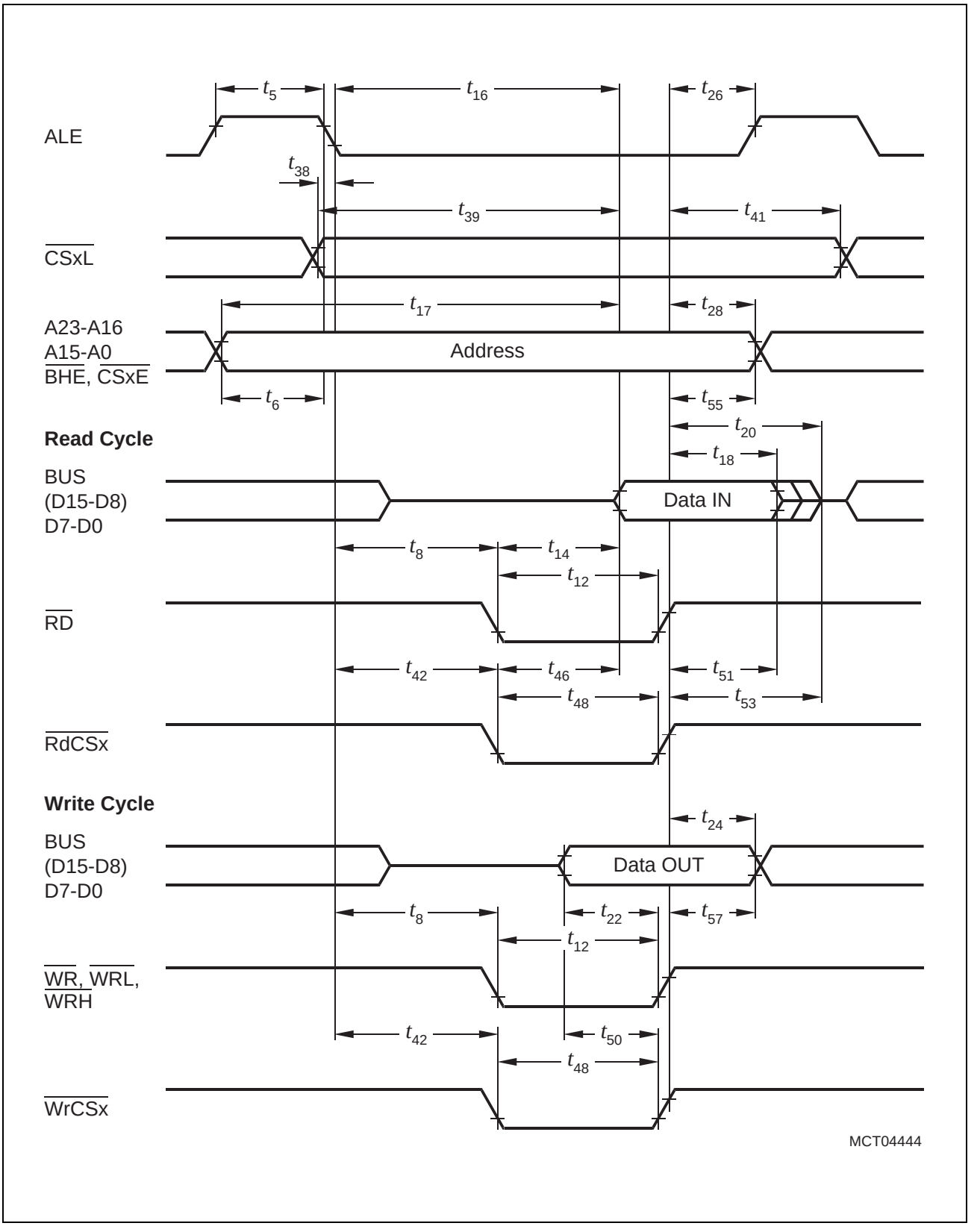



**Figure 21 External Memory Cycle:
Demultiplexed Bus, With Read/Write Delay, Extended ALE**

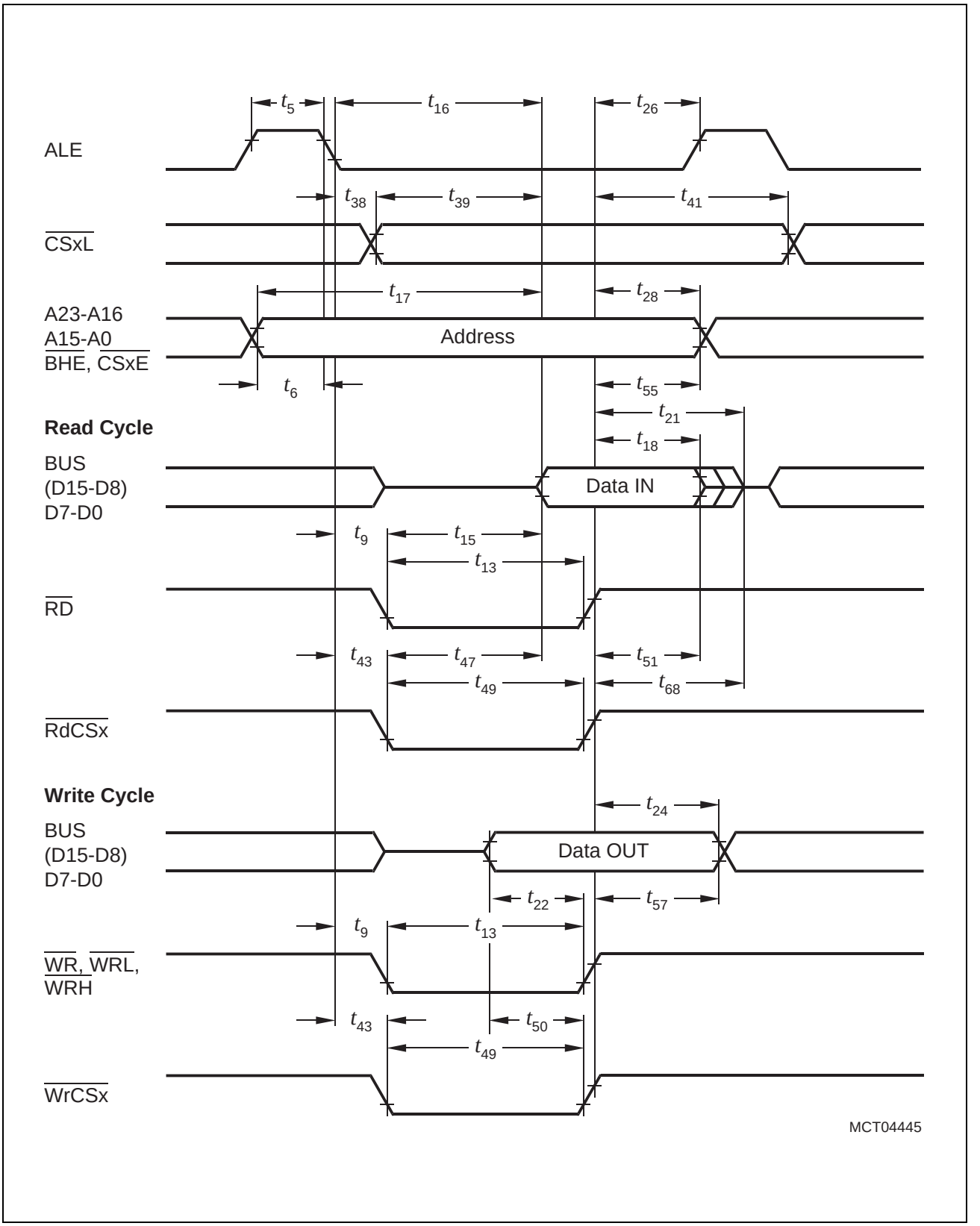



**Figure 22 External Memory Cycle:
Demultiplexed Bus, No Read/Write Delay, Normal ALE**

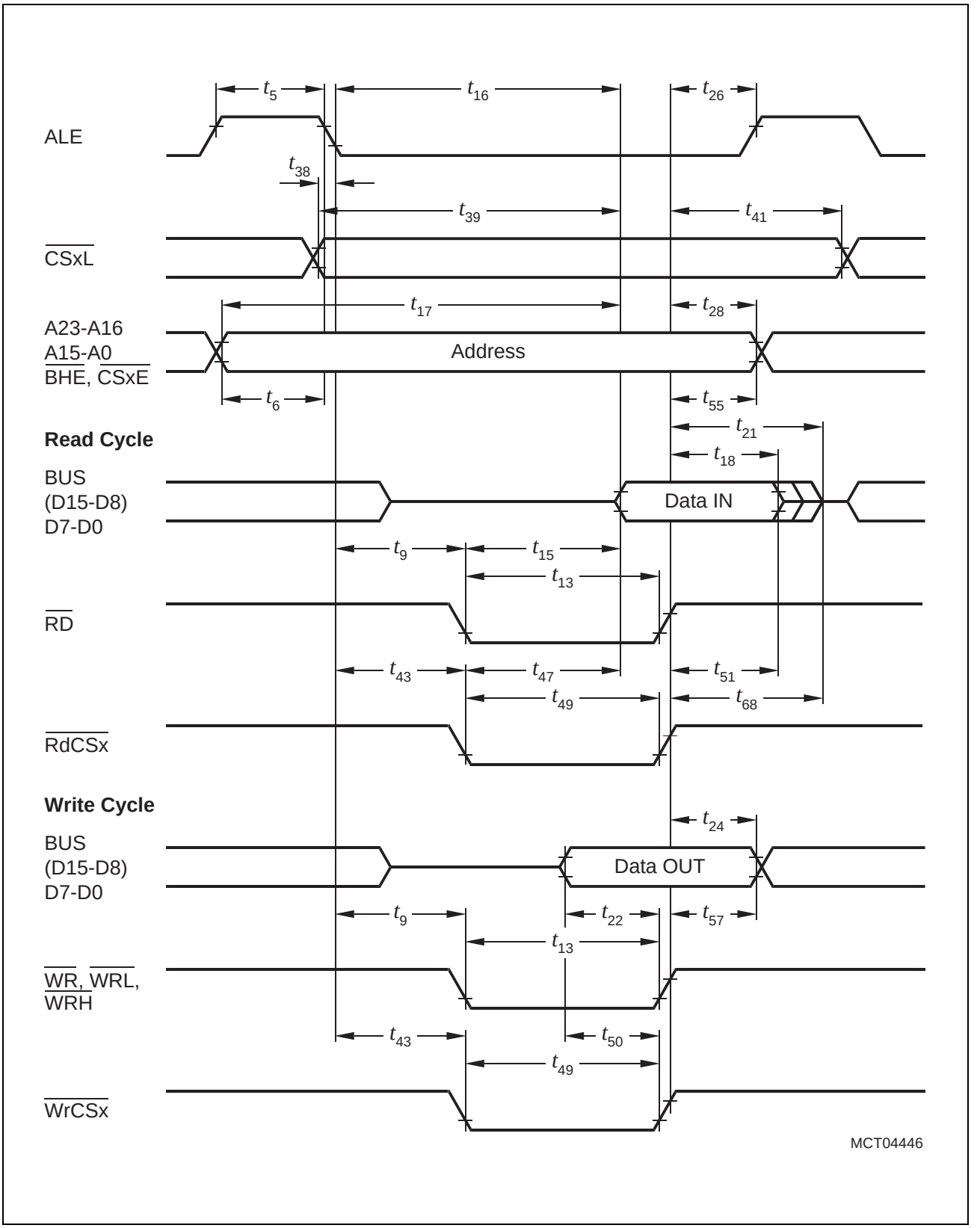



**Figure 23 External Memory Cycle:
Demultiplexed Bus, No Read/Write Delay, Extended ALE**

## AC Characteristics

### CLKOUT and $\overline{\text{READY}}$

(Operating Conditions apply)

| Parameter | Symbol | | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| CLKOUT cycle time | $t_{29}$ | CC | 40 | 40 | 2TCL | 2TCL | ns |
| CLKOUT high time | $t_{30}$ | CC | 14 | – | TCL - 6 | – | ns |
| CLKOUT low time | $t_{31}$ | CC | 10 | – | TCL - 10 | – | ns |
| CLKOUT rise time | $t_{32}$ | CC | – | 4 | – | 4 | ns |
| CLKOUT fall time | $t_{33}$ | CC | – | 4 | – | 4 | ns |
| CLKOUT rising edge to ALE falling edge | $t_{34}$ | CC | 0 + $t_A$ | 10 + $t_A$ | 0 + $t_A$ | 10 + $t_A$ | ns |
| Synchronous $\overline{\text{READY}}$ setup time to CLKOUT | $t_{35}$ | SR | 14 | – | 14 | – | ns |
| Synchronous $\overline{\text{READY}}$ hold time after CLKOUT | $t_{36}$ | SR | 4 | – | 4 | – | ns |
| Asynchronous $\overline{\text{READY}}$ low time | $t_{37}$ | SR | 54 | – | 2TCL + $t_{58}$ | – | ns |
| Asynchronous $\overline{\text{READY}}$ setup time[1)] | $t_{58}$ | SR | 14 | – | 14 | – | ns |
| Asynchronous $\overline{\text{READY}}$ hold time[1)] | $t_{59}$ | SR | 4 | – | 4 | – | ns |
| Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus)[2)] | $t_{60}$ | SR | 0 | 0 + $2t_A$ + $t_C$ + $t_F$[2)] | 0 | TCL - 20 + $2t_A$ + $t_C$ + $t_F$[2)] | ns |

1) These timings are given for test purposes only, in order to assure recognition at a specific clock edge.

2) Demultiplexed bus is the worst case. For multiplexed bus 2TCL are to be added to the maximum values. This adds even more time for deactivating $\overline{\text{READY}}$.
The $2t_A$ and $t_C$ refer to the next following bus cycle, $t_F$ refers to the current bus cycle.
The maximum limit for $t_{60}$ must be fulfilled if the next following bus cycle is $\overline{\text{READY}}$ controlled.

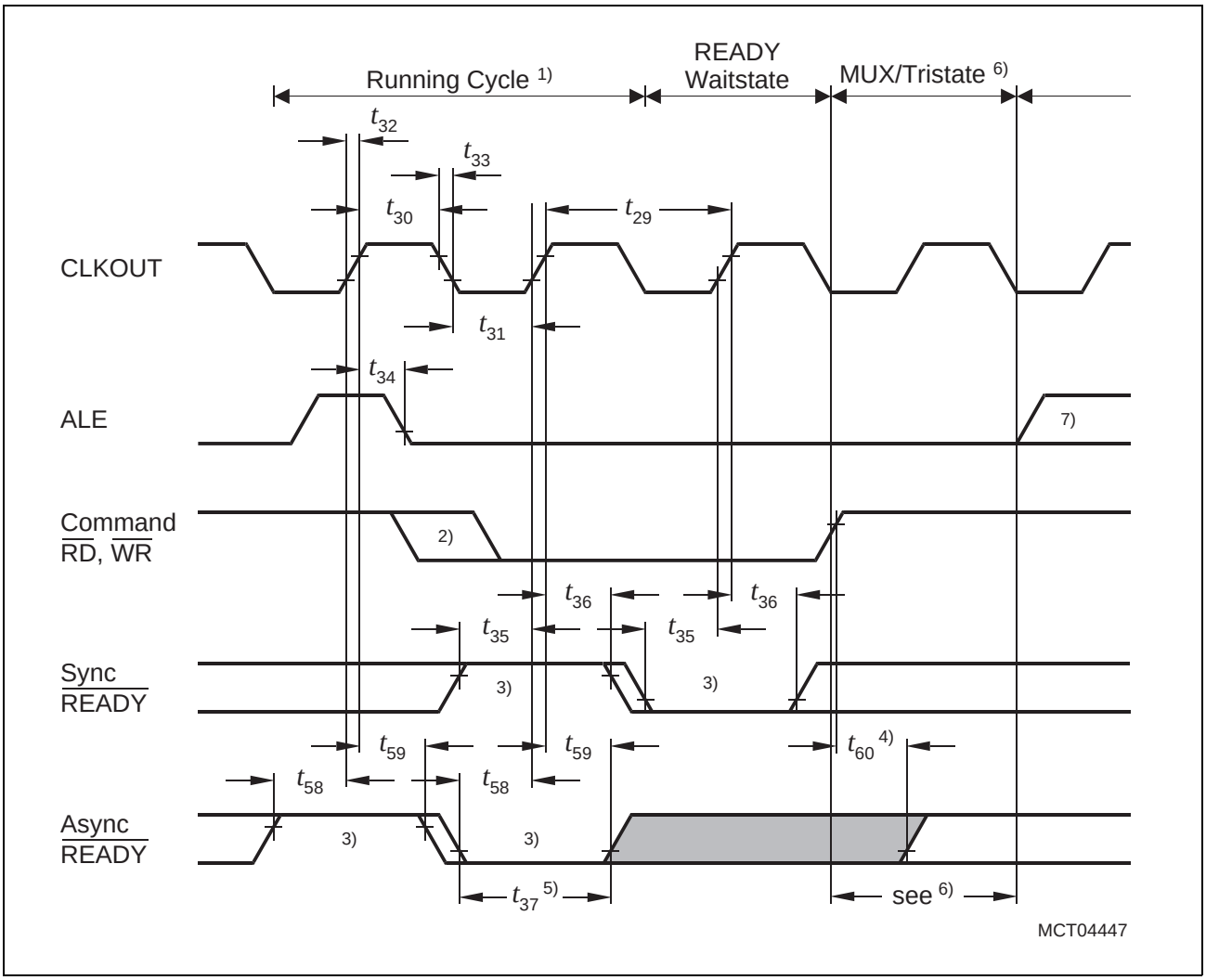



**Figure 24 CLKOUT and READY**

**Notes**

1) Cycle as programmed, including MCTC waitstates (Example shows 0 MCTC WS).
2) The leading edge of the respective command depends on RW-delay.
3) READY sampled HIGH at this sampling point generates a READY controlled waitstate,
READY sampled LOW at this sampling point terminates the currently running bus cycle.
4) READY may be deactivated in response to the trailing (rising) edge of the corresponding command (RD or WR).
5) If the Asynchronous READY signal does not fulfill the indicated setup and hold times with respect to CLKOUT (e.g. because CLKOUT is not enabled), it must fulfill $t_{37}$ in order to be safely synchronized. This is guaranteed, if READY is removed in reponse to the command (see Note 4)).
6) Multiplexed bus modes have a MUX waitstate added after a bus cycle, and an additional MTTC waitstate may be inserted here.
For a multiplexed bus with MTTC waitstate this delay is 2 CLKOUT cycles, for a demultiplexed bus without MTTC waitstate this delay is zero.
7) The next external bus cycle may start here.

## AC Characteristics

### External Bus Arbitration

(Operating Conditions apply)

| Parameter | Symbol | Max. CPU Clock = 25 MHz | | Variable CPU Clock 1 / 2TCL = 1 to 25 MHz | | Unit |
|---|---|---|---|---|---|---|
| | | min. | max. | min. | max. | |
| $\overline{\text{HOLD}}$ input setup time to CLKOUT | $t_{61}$ SR | 20 | – | 20 | – | ns |
| CLKOUT to $\overline{\text{HLDA}}$ high or $\overline{\text{BREQ}}$ low delay | $t_{62}$ CC | – | 20 | – | 20 | ns |
| CLKOUT to $\overline{\text{HLDA}}$ low or $\overline{\text{BREQ}}$ high delay | $t_{63}$ CC | – | 20 | – | 20 | ns |
| $\overline{\text{CSx}}$ release | $t_{64}$ CC | – | 20 | – | 20 | ns |
| $\overline{\text{CSx}}$ drive | $t_{65}$ CC | -4 | 24 | -4 | 24 | ns |
| Other signals release | $t_{66}$ CC | – | 20 | – | 20 | ns |
| Other signals drive | $t_{67}$ CC | – 4 | 24 | – 4 | 24 | ns |

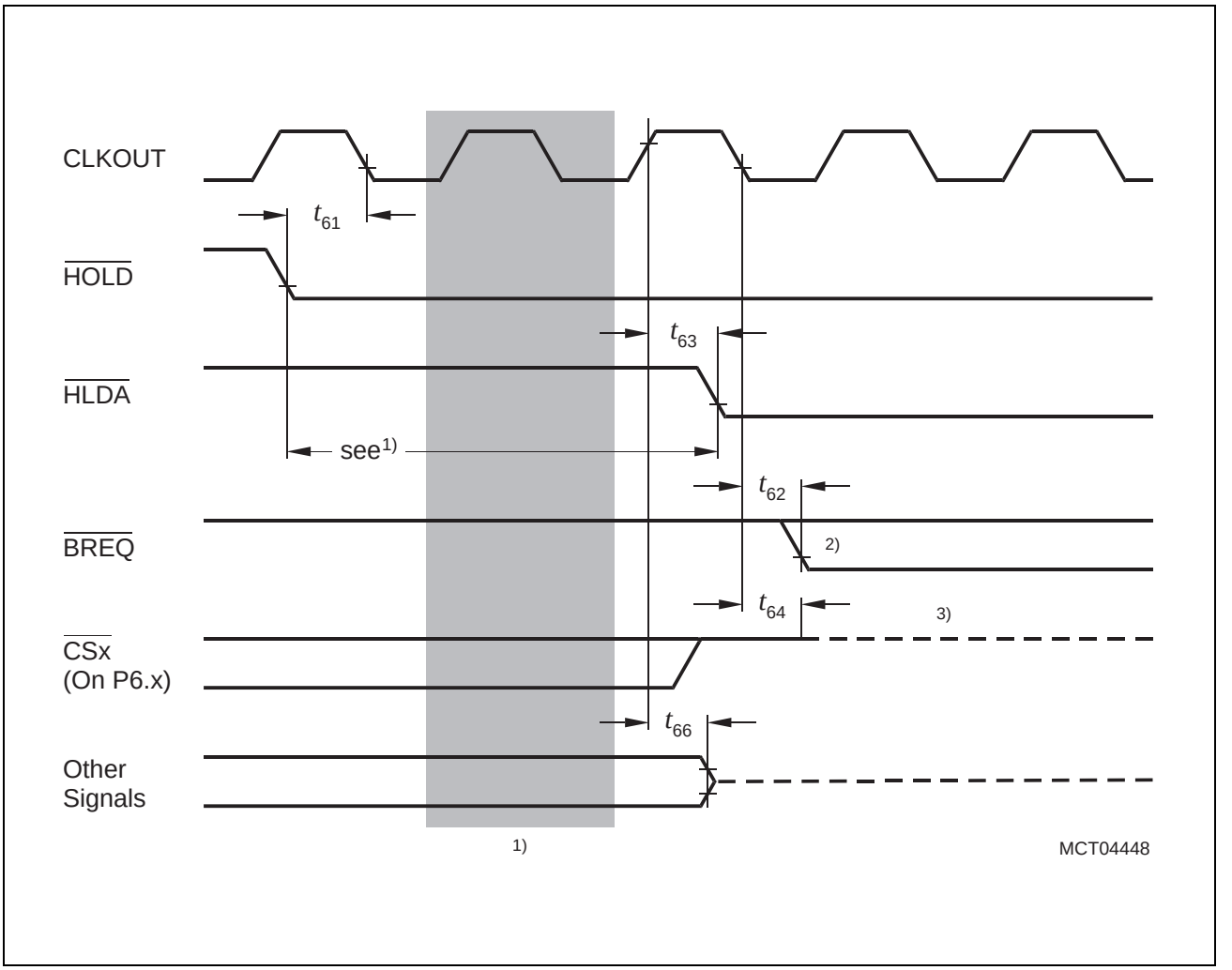



**Figure 25 External Bus Arbitration, Releasing the Bus**

**Notes**
1) The C161CS/JC/JI will complete the currently running bus cycle before granting bus access.
2) This is the first possibility for $\overline{\text{BREQ}}$ to get active.
3) The $\overline{\text{CS}}$ outputs will be resistive high (pullup) after $t_{64}$.

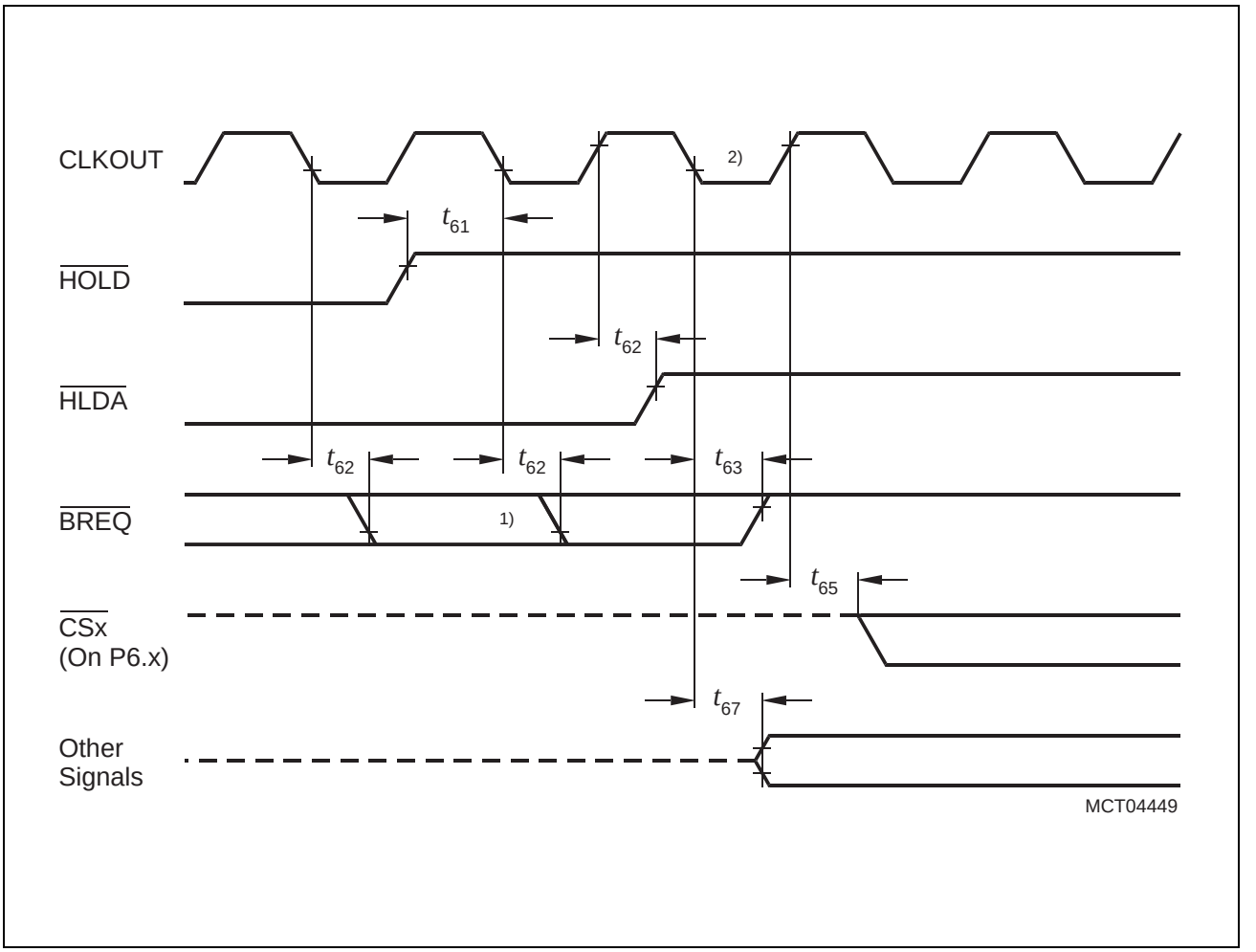



**Figure 26 External Bus Arbitration, (Regaining the Bus)**

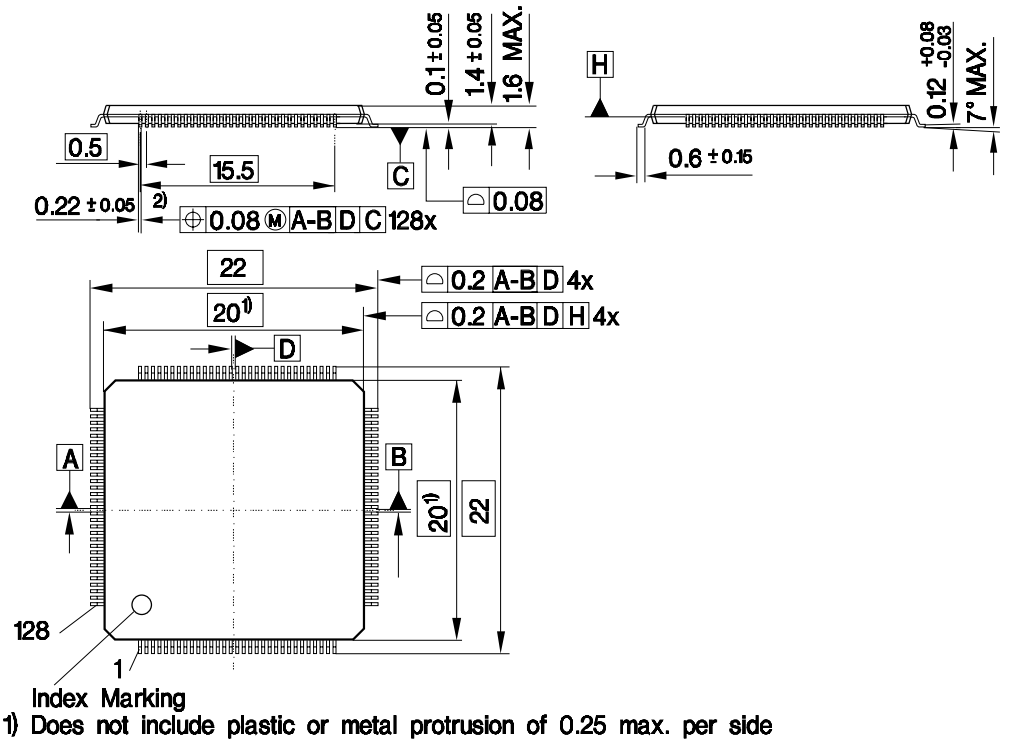
**Notes**

1) This is the last chance for BREQ to trigger the indicated regain-sequence.
Even if BREQ is activated earlier, the regain-sequence is initiated by HOLD going high.
Please note that HOLD may also be deactivated without the C161CS/JC/JI requesting the bus.

2) The next C161CS/JC/JI driven bus cycle may start here.

## Package Outline

**P-TQFP-128-2**
(Plastic Thin Metric Quad Flat Package)




GPP09028

**Sorts of Packing**
Package outlines for tubes, trays etc. are contained in our Data Book “Package Information”.

SMD = Surface Mounted Device

Dimensions in mm

# Infineon goes for Business Excellence

“Business excellence means intelligent approaches and clearly defined processes, which are both constantly under review and ultimately lead to good operating results.
Better operating results and business excellence mean less idleness and wastefulness for all of us, more professional success, more accurate information, a better overview and, thereby, less frustration and more satisfaction.”

Dr. Ulrich Schumacher

http://www.infineon.com

Published by Infineon Technologies AG

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Infineon:

SAK-C161JC-LF CA SAB-C161JC-LF CA SAB-C161JI-LF CA SAK-C161CS-LF CA SAB-C161CS-LF CA SAF-C161CS-LF CA B161CSLFCAXP B161JCLFCAXP