SIEMENS

# Microcomputer Components

## 16-Bit CMOS Single-Chip Microcontroller

# C167CR-4RM

Data Sheet 07.97 Preliminary

| C167CR-4RM<br>Revision History: | Version 07.97 Preliminary |
|---|---|
| Previous Releases: | Original Version 12.96 (Advance Information) |
| **Page** | **Subjects** |
| 2 | Ordering codes removed. |
| 11 | Description of bidirectional reset added. |
| 39, 40 | „Low Noise Driver“ note added. |
| 45...47 | Description of clock generation updated. |
| 50, 56 | $t_{22}$ changed. |
| 57 | $t_{28}$ definition changed. |
| 62 | $t_{36}$, $t_{59}$ changed. |
| | |
| --- | Granularity of timing specifications changed from 5 to 2. |

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

**Edition 07.97**

**Published by Siemens AG, Bereich Halbleiter, Marketing-Kommunikation**
**Balanstraße 73, D-81541 München.**

SIEMENS

## C16x-Family of
## High-Performance CMOS 16-Bit Microcontrollers

# C167CR-4RM

## Advance Information
## C167CR-4RM 16-Bit Microcontroller

- High Performance 16-bit CPU with 4-Stage Pipeline
- 100 ns Instruction Cycle Time at 20 MHz CPU Clock
- 500 ns Multiplication (16 × 16 bit), 1 µs 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
- Clock Generation via on-chip PLL or via direct clock input
- Up to 16 MBytes Linear Address Space for Code and Data
- 2 KBytes On-Chip Internal RAM (IRAM)
- 2 KBytes On-Chip Extension RAM (XRAM)
- 32 KBytes On-Chip ROM
- Programmable External Bus Characteristics for Different Address Ranges
- 8-Bit or 16-Bit External Data Bus
- Multiplexed or Demultiplexed External Address/Data Buses
- Five Programmable Chip-Select Signals
- Hold- and Hold-Acknowledge Bus Arbitration Support
- 1024 Bytes On-Chip Special Function Register Area
- Idle and Power Down Modes
- 8-Channel Interrupt-Driven Single-Cycle Data Transfer Facilities via Peripheral Event Controller (PEC)
- 16-Priority-Level Interrupt System with 56 Sources, Sample-Rate down to 50 ns
- 16-Channel 10-bit A/D Converter with 9.7µs Conversion Time
- Two 16-Channel Capture/Compare Units
- 4-Channel PWM Unit
- Two Multi-Functional General Purpose Timer Units with 5 Timers
- Two Serial Channels (Synchronous/Asynchronous and High-Speed-Synchronous)
- On-Chip CAN Interface with 15 Message Objects (Full-CAN/Basic-CAN)
- Programmable Watchdog Timer
- Up to 111 General Purpose I/O Lines, partly with Selectable Input Thresholds and Hysteresis
- Supported by a Wealth of Development Tools like C-Compilers, Macro-Assembler Packages, Emulators, Evaluation Boards, HLL-Debuggers, Simulators, Logic Analyzer Disassemblers, Programming Boards
- On-Chip Bootstrap Loader
- 144-Pin MQFP Package (EIAJ)

This document describes the **SAB-C167CR-4RM** and the **SAK-C167CR-4RM**. For simplicity all versions are referred to by the term **C167CR-4RM** throughout this document.

## Introduction

The C167CR-4RM is a new derivative of the Siemens C16x Family of full featured single-chip CMOS microcontrollers. It combines high CPU performance (up to 10 million instructions per second) with high peripheral functionality and enhanced IO-capabilities. It also provides on-chip ROM, on-chip high-speed RAM and clock generation via PLL.

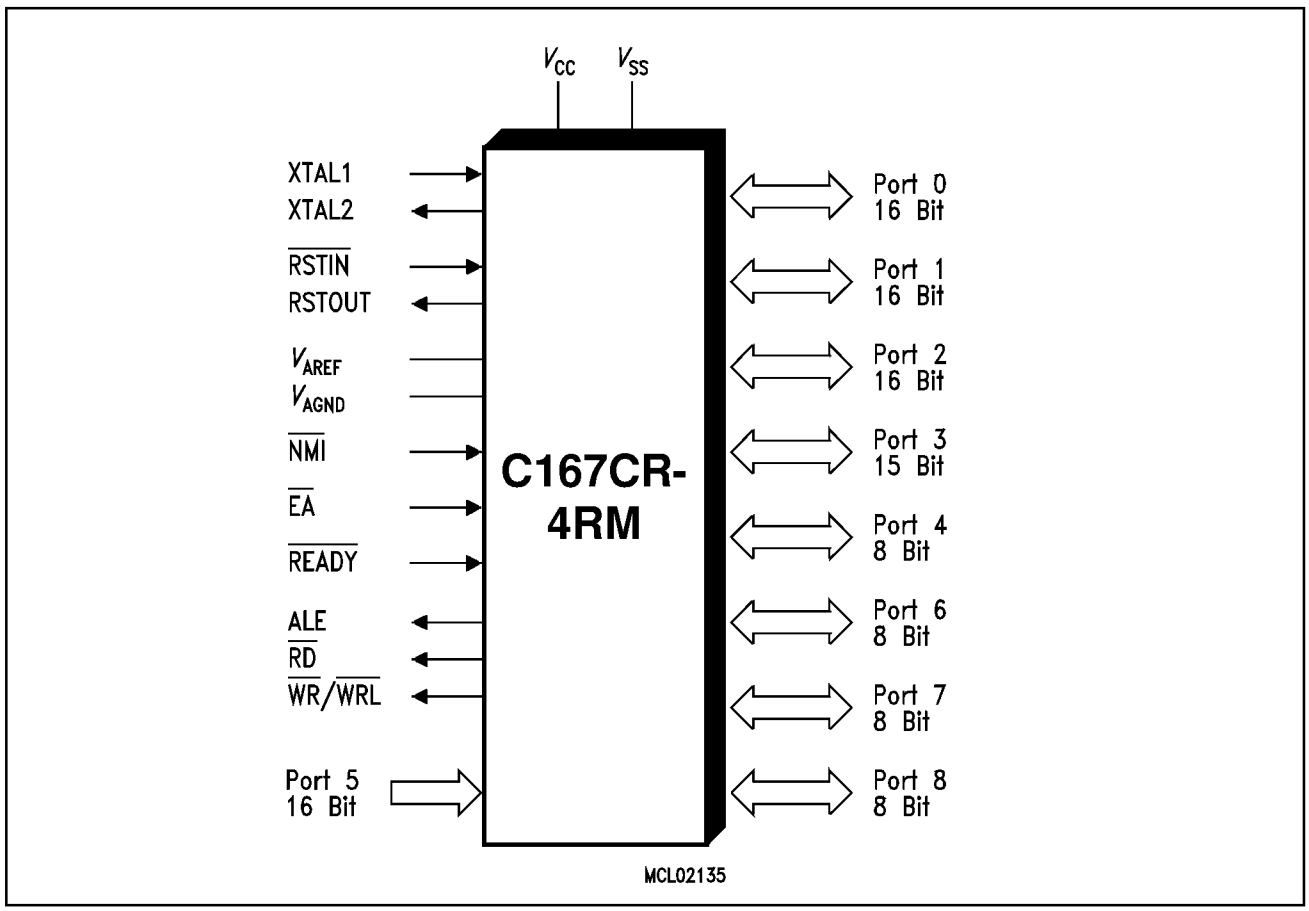



**Figure 1**
**Logic Symbol**

## Ordering Information

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

- the derivative itself, ie. its function set
- the specified temperature range
- the package
- the type of delivery.

For the available ordering codes for the C167CR-4RM please refer to the „**Product Information Microcontrollers**“, which summarizes all available microcontroller variants.

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

## Pin Configuration

(top view)

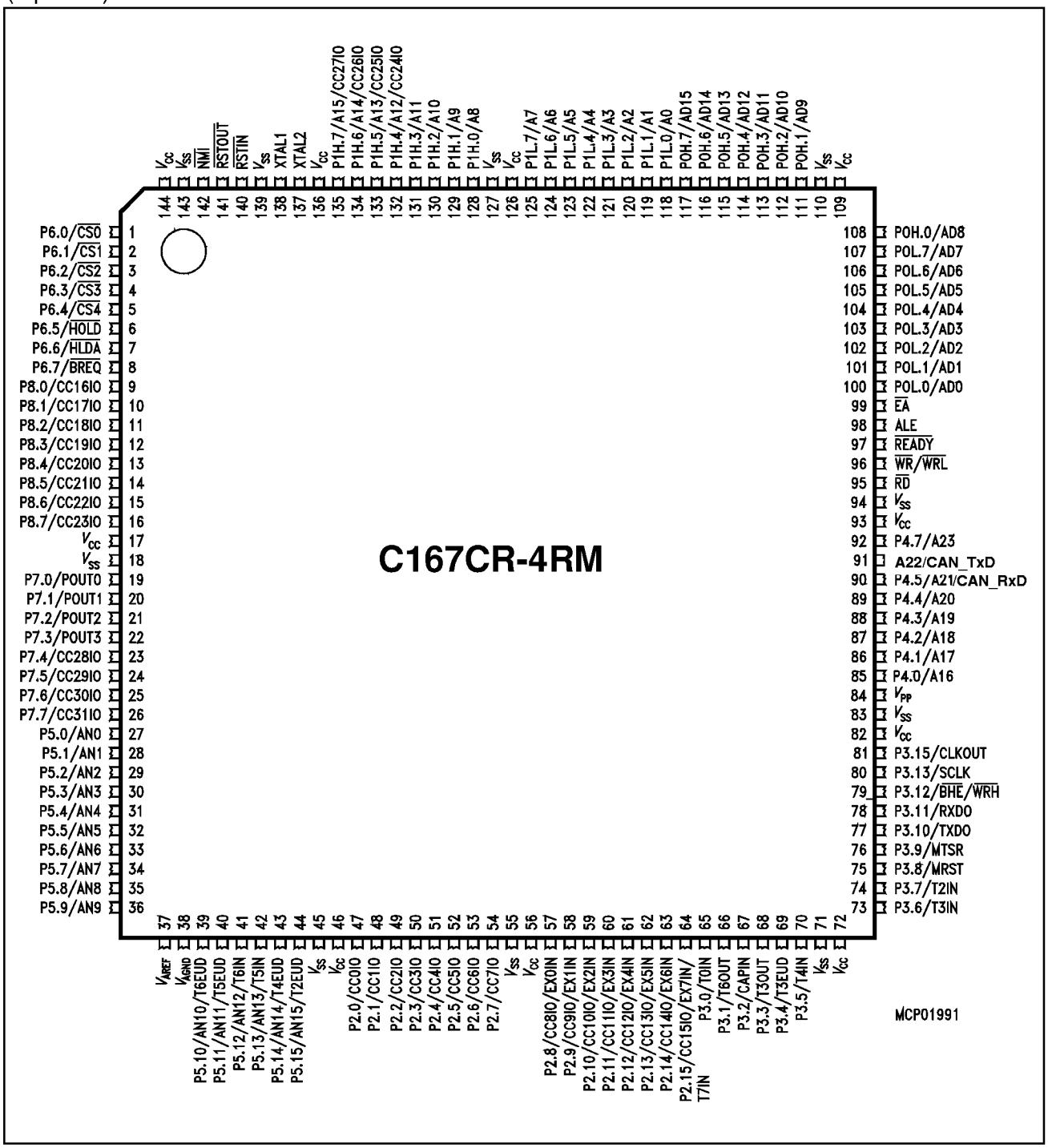



**Figure 2**

**Pin Definitions and Functions**

| Symbol | Pin Number | Input (I) Output (O) | Function | | |
|---|---|---|---|---|---|
| P6.0 – P6.7 | 1 - 8 | I/O | 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 following Port 6 pins also serve for alternate functions: | | |
| | 1 | O | P6.0 | $\overline{CS0}$ | Chip Select 0 Output |
| | ... | ... | ... | ... | ... |
| | 5 | O | P6.4 | $\overline{CS4}$ | Chip Select 4 Output |
| | 6 | I | P6.5 | $\overline{HOLD}$ | External Master Hold Request Input |
| | 7 | O | P6.6 | $\overline{HLDA}$ | Hold Acknowledge Output |
| | 8 | O | P6.7 | $\overline{BREQ}$ | Bus Request Output |
| P8.0 – P8.7 | 9 - 16 | I/O | Port 8 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 8 outputs can be configured as push/ pull or open drain drivers. The input threshold of Port 8 is selectable (TTL or special). The following Port 8 pins also serve for alternate functions: | | |
| | 9 | I/O | P8.0 | CC16IO | CAPCOM2: CC16 Cap.-In/Comp.Out |
| | ... | ... | ... | ... | ... |
| | 16 | I/O | P8.7 | CC23IO | CAPCOM2: CC23 Cap.-In/Comp.Out |
| P7.0 – P7.7 | 19 - 26 | I/O | Port 7 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 7 outputs can be configured as push/ pull or open drain drivers. The input threshold of Port 7 is selectable (TTL or special). The following Port 7 pins also serve for alternate functions: | | |
| | 19 | O | P7.0 | POUT0 | PWM Channel 0 Output |
| | ... | ... | ... | ... | ... |
| | 22 | O | P7.3 | POUT3 | PWM Channel 3 Output |
| | 23 | I/O | P7.4 | CC28IO | CAPCOM2: CC28 Cap.-In/Comp.Out |
| | ... | ... | ... | ... | ... |
| | 26 | I/O | P7.7 | CC31IO | CAPCOM2: CC31 Cap.-In/Comp.Out |

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

| Symbol | Pin Number | Input (I) Output (O) | Function | | |
|---|---|---|---|---|---|
| P5.0 – P5.15 | 27 – 36<br>39 – 44 | I<br>I | Port 5 is a 16-bit input-only port with Schmitt-Trigger characteristics. The pins of Port 5 also serve as the (up to 16) analog input channels for the A/D converter, where P5.x equals ANx (Analog input channel x), or they serve as timer inputs: | | |
| | 39 | I | P5.10 | T6EUD | GPT2 Timer T6 Ext.Up/Down Ctrl.Input |
| | 40 | I | P5.11 | T5EUD | GPT2 Timer T5 Ext.Up/Down Ctrl.Input |
| | 41 | I | P5.12 | T6IN | GPT2 Timer T6 Count Input |
| | 42 | I | P5.13 | T5IN | GPT2 Timer T5 Count Input |
| | 43 | I | P5.14 | T4EUD | GPT1 Timer T4 Ext.Up/Down Ctrl.Input |
| | 44 | I | P5.15 | T2EUD | GPT1 Timer T2 Ext.Up/Down Ctrl.Input |
| P2.0 – P2.15 | 47 – 54<br>57 - 64 | I/O | Port 2 is a 16-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: | | |
| | 47 | I/O | P2.0 | CC0IO | CAPCOM: CC0 Cap.-In/Comp.Out |
| | ... | ... | ... | ... | ... |
| | 54 | I/O | P2.7 | CC7IO | CAPCOM: CC7 Cap.-In/Comp.Out |
| | 57 | I/O | P2.8 | CC8IO | CAPCOM: CC8 Cap.-In/Comp.Out, |
| | | I | | EX0IN | Fast External Interrupt 0 Input |
| | ... | ... | ... | ... | ... |
| | 64 | I/O | P2.15 | CC15IO | CAPCOM: CC15 Cap.-In/Comp.Out, |
| | | I | | EX7IN | Fast External Interrupt 7 Input |
| | | I | | T7IN | CAPCOM2 Timer T7 Count Input |

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

| Symbol | Pin Number | Input (I) Output (O) | Function | | |
|---|---|---|---|---|---|
| P3.0 – P3.13, P3.15 | 65 – 70, 73 – 80, 81 | I/O I/O I/O | Port 3 is a 15-bit (P3.14 is missing) 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). The following Port 3 pins also serve for alternate functions: | | |
| | 65 | I | P3.0 | T0IN | CAPCOM Timer T0 Count Input |
| | 66 | O | P3.1 | T6OUT | GPT2 Timer T6 Toggle Latch Output |
| | 67 | I | P3.2 | CAPIN | GPT2 Register CAPREL Capture Input |
| | 68 | O | P3.3 | T3OUT | GPT1 Timer T3 Toggle Latch Output |
| | 69 | I | P3.4 | T3EUD | GPT1 Timer T3 Ext.Up/Down Ctrl.Input |
| | 70 | I | P3.5 | T4IN | GPT1 Timer T4 Input for Count/Gate/Reload/Capture |
| | 73 | I | P3.6 | T3IN | GPT1 Timer T3 Count/Gate Input |
| | 74 | I | P3.7 | T2IN | GPT1 Timer T2 Input for Count/Gate/Reload/Capture |
| | 75 | I/O | P3.8 | MRST | SSC Master-Rec./Slave-Transmit I/O |
| | 76 | I/O | P3.9 | MTSR | SSC Master-Transmit/Slave-Rec. O/I |
| | 77 | O | P3.10 | T×D0 | ASC0 Clock/Data Output (Asyn./Syn.) |
| | 78 | I/O | P3.11 | R×D0 | ASC0 Data Input (Asyn.) or I/O (Syn.) |
| | 79 | O | P3.12 | $\overline{\text{BHE}}$ | Ext. Memory High Byte Enable Signal, |
| | | O | | $\overline{\text{WRH}}$ | Ext. Memory High Byte Write Strobe |
| | 80 | I/O | P3.13 | SCLK | SSC Master Clock Outp./Slave Cl. Inp. |
| | 81 | O | P3.15 | CLKOUT | System Clock Output (=CPU Clock) |
| P4.0 – P4.7 | 85 - 92 | I/O | 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. In case of an external bus configuration, Port 4 can be used to output the segment address lines: | | |
| | 85 | O | P4.0 | A16 | Least Significant Segment Addr. Line |
| | ... | ... | ... | ... | ... |
| | 89 | O | P4.4 | A20 | Segment Address Line |
| | 90 | O | P4.5 | A21 | Segment Address Line, |
| | | I | | CAN_RxD | CAN Receive Data Input |
| | 91 | O | P4.6 | A22 | Segment Address Line, |
| | | O | | CAN_TxD | CAN Transmit Data Output |
| | 92 | O | P4.7 | A23 | Most Significant Segment Addr. Line |
| $\overline{\text{RD}}$ | 95 | O | External Memory Read Strobe. $\overline{\text{RD}}$ is activated for every external instruction or data read access. | | |

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

| Symbol | Pin Number | Input (I) Output (O) | Function |
|---|---|---|---|
| $\overline{WR}$/ $\overline{WRL}$ | 96 | 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. |
| $\overline{READY}$ | 97 | 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. |
| ALE | 98 | O | Address Latch Enable Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes. |
| $\overline{EA}$ | 99 | I | External Access Enable pin. A low level at this pin during and after Reset forces the C167CR-4RM to begin instruction execution out of external memory. A high level forces execution out of the internal ROM. ROMless versions must have this pin tied to '0'. |
| PORT0:<br>P0L.0 –<br>P0L.7,<br>P0H.0 -<br>P0H.7 | <br>100 –<br>107<br>108,<br>111-117 | I/O | 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 |

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

| Symbol | Pin Number | Input (I) Output (O) | Function |
|---|---|---|---|
| PORT1: P1L.0 – P1L.7, P1H.0 - P1H.7 | 118 – 125 128 – 135 | I/O | 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.<br>The following PORT1 pins also serve for alternate functions:<br>P1H.4 CC24IO CAPCOM2: CC24 Capture Input<br>P1H.5 CC25IO CAPCOM2: CC25 Capture Input<br>P1H.6 CC26IO CAPCOM2: CC26 Capture Input<br>P1H.7 CC27IO CAPCOM2: CC27 Capture Input |
| | 132 | I | |
| | 133 | I | |
| | 134 | I | |
| | 135 | I | |
| XTAL1 | 138 | I | XTAL1: Input to the oscillator amplifier and input to the internal clock generator |
| XTAL2 | 137 | O | XTAL2: Output of the oscillator amplifier circuit.<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. |
| $\overline{\text{RSTIN}}$ | 140 | I/O | Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the C167CR-4RM. An internal pullup resistor permits power-on reset using only a capacitor connected to $V_{SS}$.<br>In bidirectional reset mode (enabled by setting bit BDRSTEN in register SYSCON) the $\overline{\text{RSTIN}}$ line is pulled low for the duration of the internal reset sequence upon a software or WDT reset. [1] |
| $\overline{\text{RSTOUT}}$ | 141 | 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. |
| $\overline{\text{NMI}}$ | 142 | 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 C167CR-4RM 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.<br>If not used, pin $\overline{\text{NMI}}$ should be pulled high externally. |
| $V_{AREF}$ | 37 | - | Reference voltage for the A/D converter. |
| $V_{AGND}$ | 38 | - | Reference ground for the A/D converter. |

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

| Symbol | Pin Number | Input (I) Output (O) | Function |
|---|---|---|---|
| $V_{PP}$ / OWE | 84 | - | Flash programming voltage / Oscillator Watchdog Enable. This pin accepts the programming voltage for C167 derivatives with on-chip flash memory. During normal operation this pin must be connected to $V_{CC}$.<br>It also enables the oscillator watchdog when high or disables it when low eg. for testing purposes. An internal pullup device holds this input high if nothing is driving it (eg. on non-flash devices). |
| $V_{CC}$ | 17, 46, 56, 72, 82, 93, 109, 126, 136, 144 | - | Digital Supply Voltage:<br>+ 5 V during normal operation and idle mode.<br>≥ 2.5 V during power down mode. |
| $V_{SS}$ | 18, 45, 55, 71, 83, 94, 110, 127, 139, 143 | - | Digital Ground. |

[1] The following behaviour 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.
- Bit WDTR will always be '0', even after a watchdog timer reset.
- The PORT0 configuration is treated like on a hardware reset. Especially 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.

## Functional Description

The architecture of the C167CR-4RM combines advantages of both RISC and CISC processors and of advanced peripheral subsystems in a very well-balanced way. The following block diagram gives an overview of the different on-chip components and of the advanced, high bandwidth internal bus structure of the C167CR-4RM.

**Note**: All time specifications refer to a CPU clock of 20 MHz
(see definition in the AC Characteristics section).

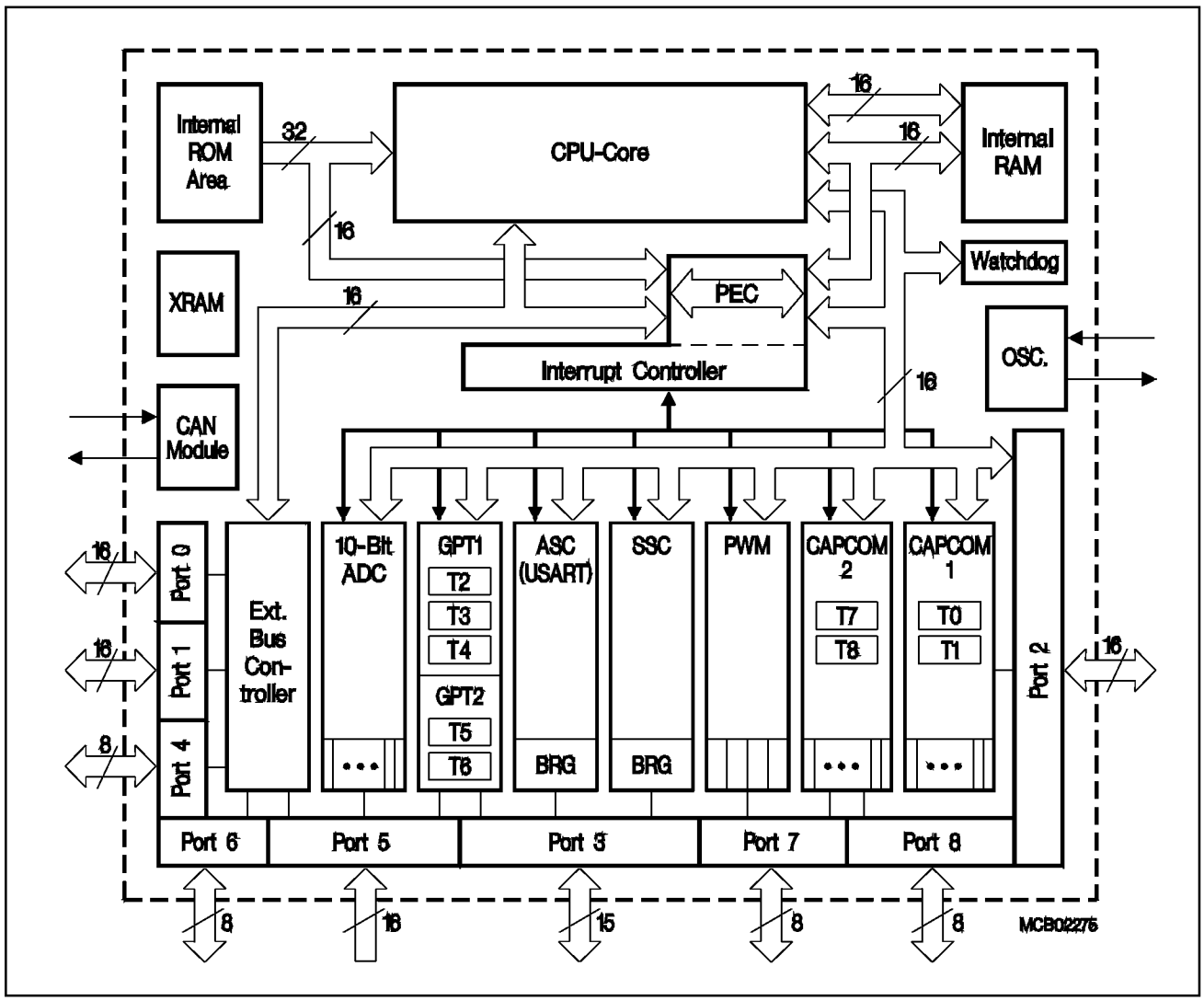



**Figure 3**
**Block Diagram**

**Bidirectional Reset**

In a special mode (Bidirectional reset) the C167CR-4RM's line $\overline{\text{RSTIN}}$ (normally an input) may be be driven active by the chip logic eg. in order to support external equipment which is required for startup (eg. flash memory).

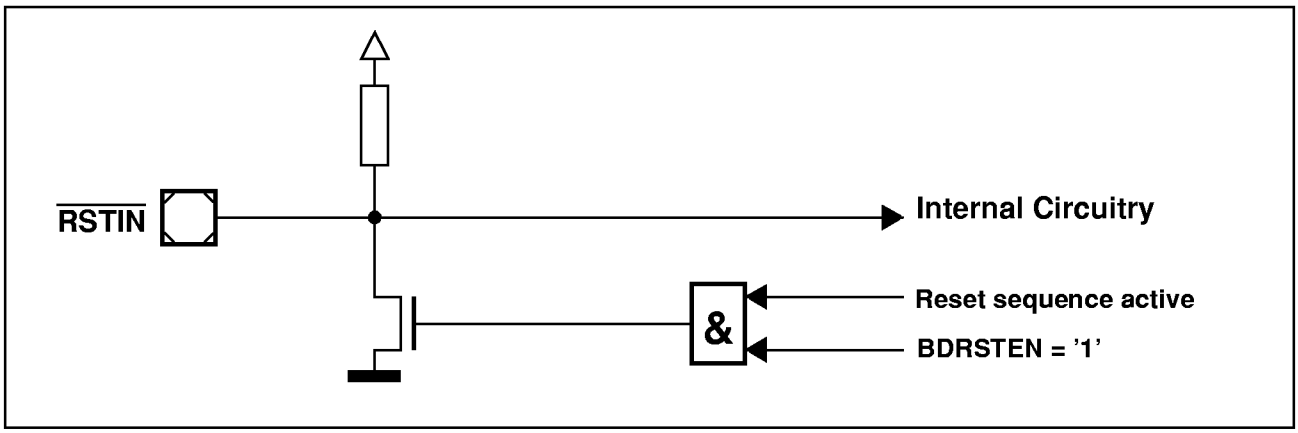



**Figure 4**
**Bidirectional Reset Operation**

Bidirectional reset reflects internal reset sources (software, watchdog) also to the $\overline{\text{RSTIN}}$ pin and converts short hardware reset pulses to a minimum duration of the internal reset sequence. Bidirectional reset is enabled by setting bit BDRSTEN (=SYSCON.3) and changes $\overline{\text{RSTIN}}$ from a pure input to an open drain IO line. When an internal reset is triggered by the SRST instruction or by a watchdog timer overflow or a low level is applied to the $\overline{\text{RSTIN}}$ line, an internal driver pulls it low for the duration of the internal reset sequence. After that it is released and is then controlled by the external circuitry alone.

The Bidirectional reset function is useful in applications where external devices require a defined reset signal but cannot be connected to the C167CR-4RM's $\overline{\text{RSTOUT}}$ signal, eg. an external flash memory which must come out of reset and deliver code well before $\overline{\text{RSTOUT}}$ can be deactivated via EINIT.

Please note the behaviour differences listed below the pin description.

### Memory Organization

The memory space of the C167CR-4RM 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 C167CR-4RM contains 32 KBytes of on-chip mask-programmable ROM for code or constant data. The on-chip ROM can be mapped either to segment 0 or segment 1.

2 KBytes of on-chip Internal RAM 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 * 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 C16x family.

2 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 bitadressable. The XRAM allows 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 allow to access 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. Access to very slow memories 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 SYSCON. 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 the on-chip CAN Module is to be used the segment address output on Port 4 must be limited to 4 bits (ie. A19...A16) in order to enable the alternate function of the CAN interface pins.

### 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 C167CR-4RM's instructions can be executed in just one machine cycle which requires 100 ns at 20-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.

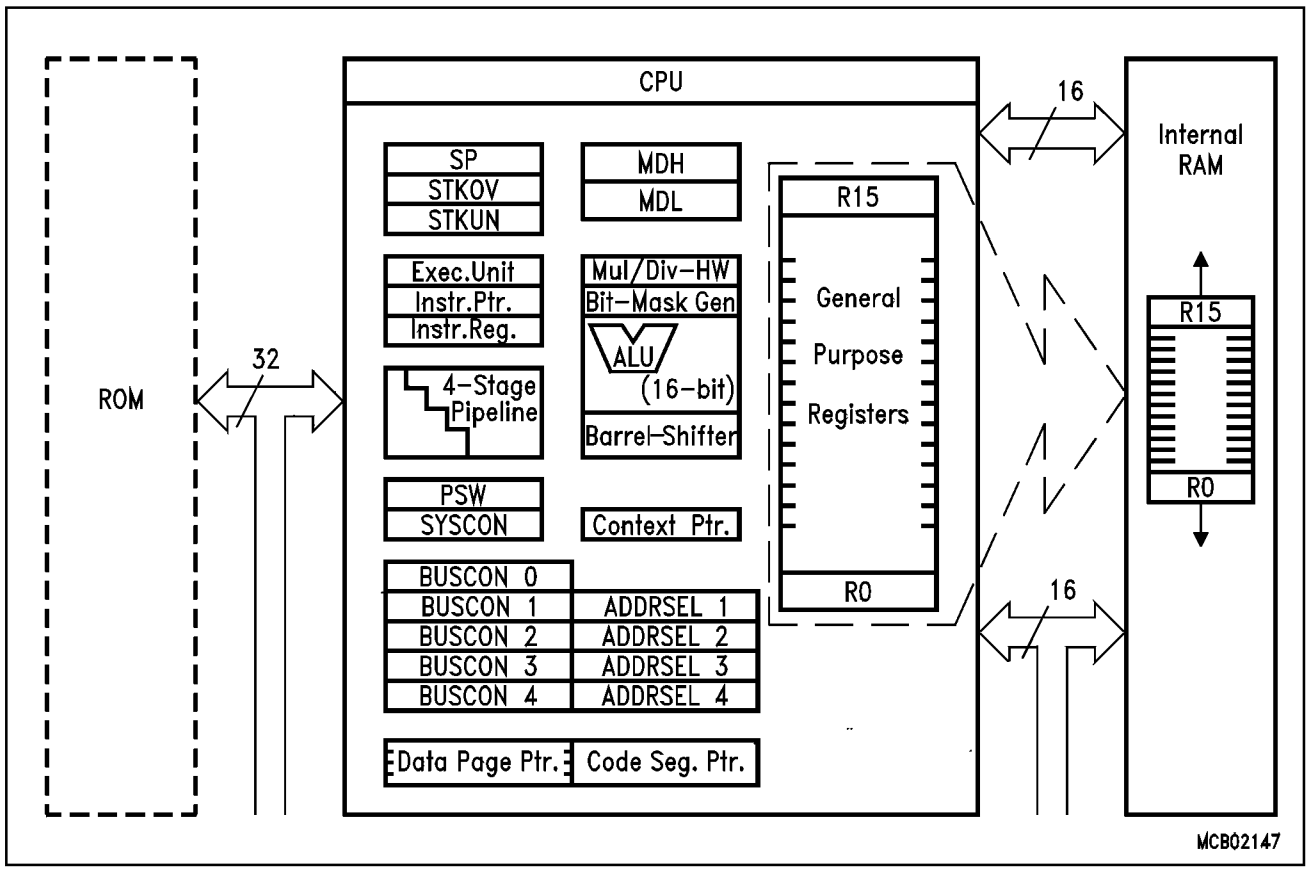



**Figure 5**
**CPU Block Diagram**

The CPU disposes of an actual register context consisting of up to 16 wordwide GPRs which 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 a 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 2048 bytes 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 C167CR-4RM 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 250 ns to 600 ns (in case of internal program execution), the C167CR-4RM is capable of reacting very fast to the occurence of non-deterministic events.

The architecture of the C167CR-4RM 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 C167CR-4RM 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.

The following table shows all of the possible C167CR-4RM interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers:

**Note:** Three nodes in the table (X-Peripheral nodes) are prepared to accept interrupt requests from integrated X-Bus peripherals. Nodes, where no X-Peripherals are connected, may be used to generate software controlled interrupt requests by setting the respective XPnIR bit.

| 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 | $00'0110_H$ | $44_H$ |
| 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 | $00'0080_H$ | $20_H$ |

| Source of Interrupt or PEC Service Request | Request Flag | Enable Flag | Interrupt Vector | Vector Location | Trap Number |
|---|---|---|---|---|---|
| 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 Register | CRIR | CRIE | CRINT | $00'009C_H$ | $27_H$ |
| A/D Conversion Complete | 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 | $00'00BC_H$ | $2F_H$ |
| PWM Channel 0...3 | PWMIR | PWMIE | PWMINT | $00'00FC_H$ | $3F_H$ |
| CAN Interface | XP0IR | XP0IE | XP0INT | $00'0100_H$ | $40_H$ |
| X-Peripheral Node | XP1IR | XP1IE | XP1INT | $00'0104_H$ | $41_H$ |
| X-Peripheral Node | XP2IR | XP2IE | XP2INT | $00'0108_H$ | $42_H$ |
| PLL Unlock | XP3IR | XP3IE | XP3INT | $00'010C_H$ | $43_H$ |

The C167CR-4RM 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.

The following table shows all of the possible exceptions or error conditions that can arise during run-time:

| 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 |
| Watchdog 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<br>TRAP Instruction | | | Any<br>$[00'0000_H –$<br>$00'01FC_H]$<br>in steps<br>of $4_H$ | Any<br>$[00_H – 7F_H]$ | 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 400 ns (at 20-MHz system clock). 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. Each register has one port pin associated with it which serves as an input pin for triggering the capture function, or as an output pin (except for CC24...CC27) to indicate the occurence 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 (‘capture’d) 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. 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.

| 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 |
| Double Register Mode | Two registers operate on one pin; pin toggles on each compare match;<br>several compare events per timer period are possible. |

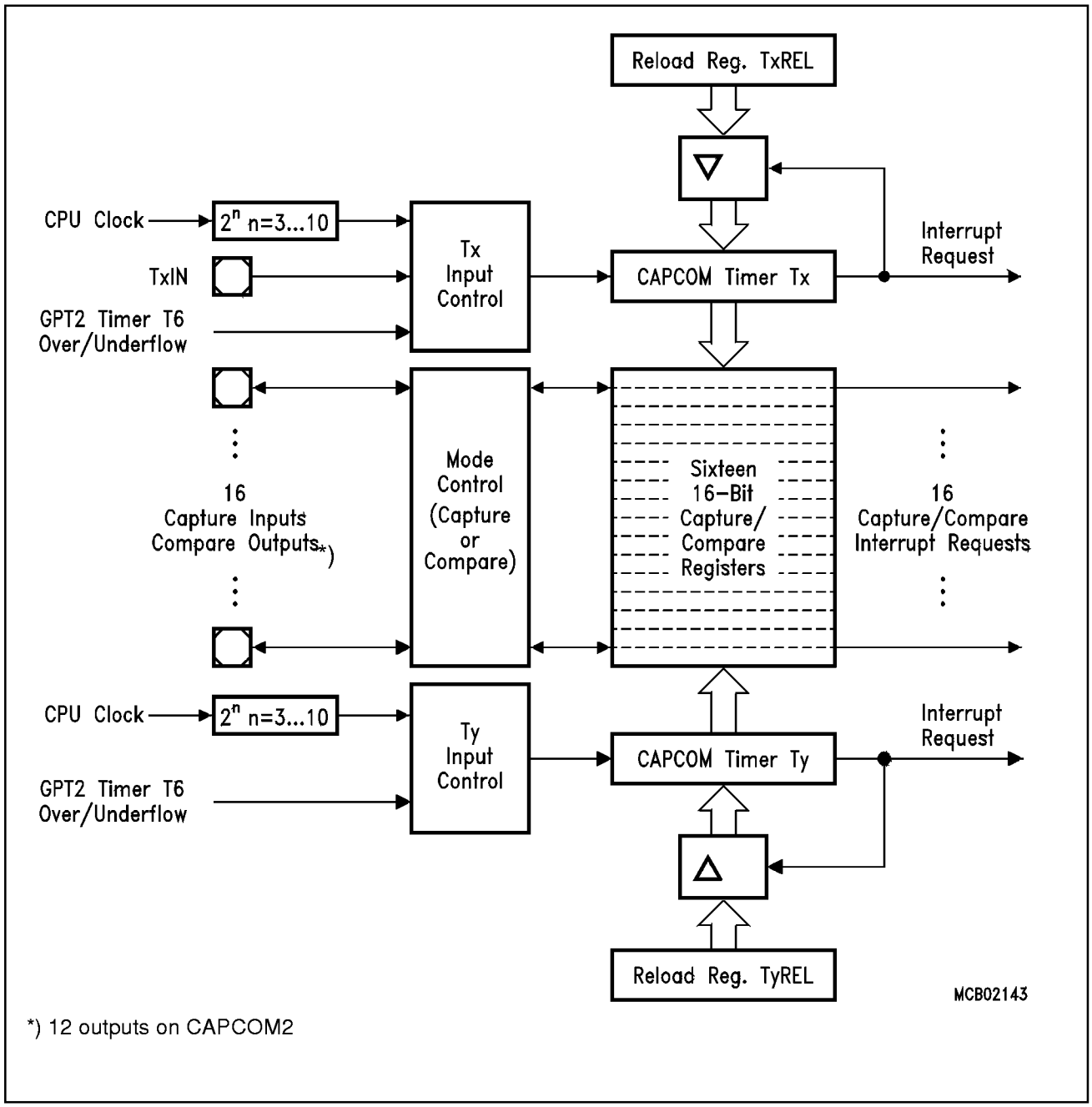



**Figure 6**
**CAPCOM Unit Block Diagram**

### PWM Module

The Pulse Width Modulation Module can generate up to four PWM output signals using edge-aligned or center-aligned PWM. In addition the PWM module can generate PWM burst signals and single shot outputs. The frequency range of the PWM signals covers 4.8 Hz to 1 MHz (referred to a CPU clock of 20 MHz), depending on the resolution of the PWM output signal. The level of the output signals is selectable and the PWM module can generate interrupt requests.

### 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 400 ns (@ 20-MHz CPU clock).

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 eg. 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.

Timers T3 and T4 have output toggle latches (TxOTL) which change their state on each timer over-flow/underflow. The state of these latches may be output on port pins (TxOUT) eg. 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.

With its maximum resolution of 200 ns (@ 20 MHz), 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 a port 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 absolute time differences to be measured or pulse multiplication to be performed 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.

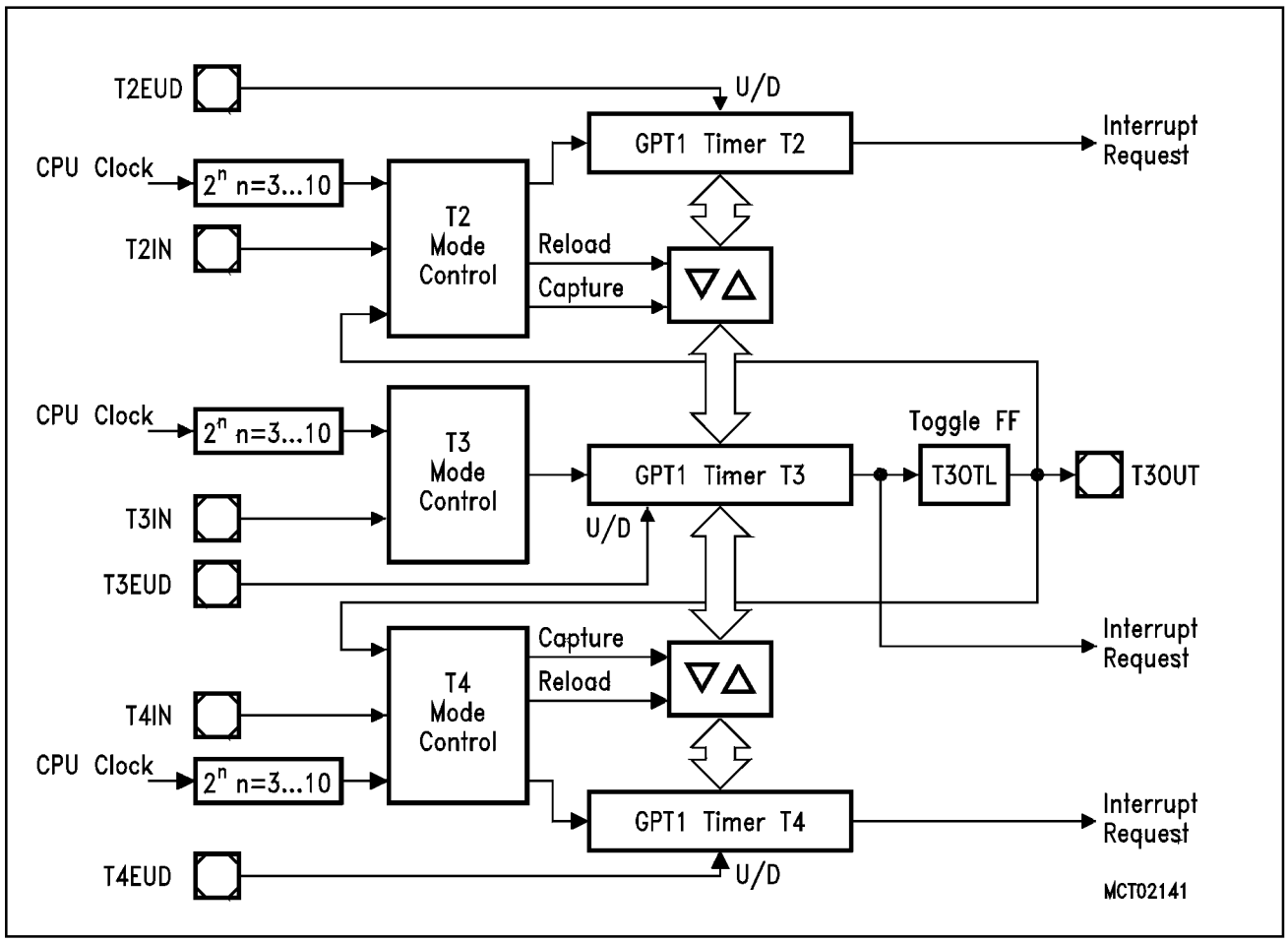



**Figure 7**
**Block Diagram of GPT1**

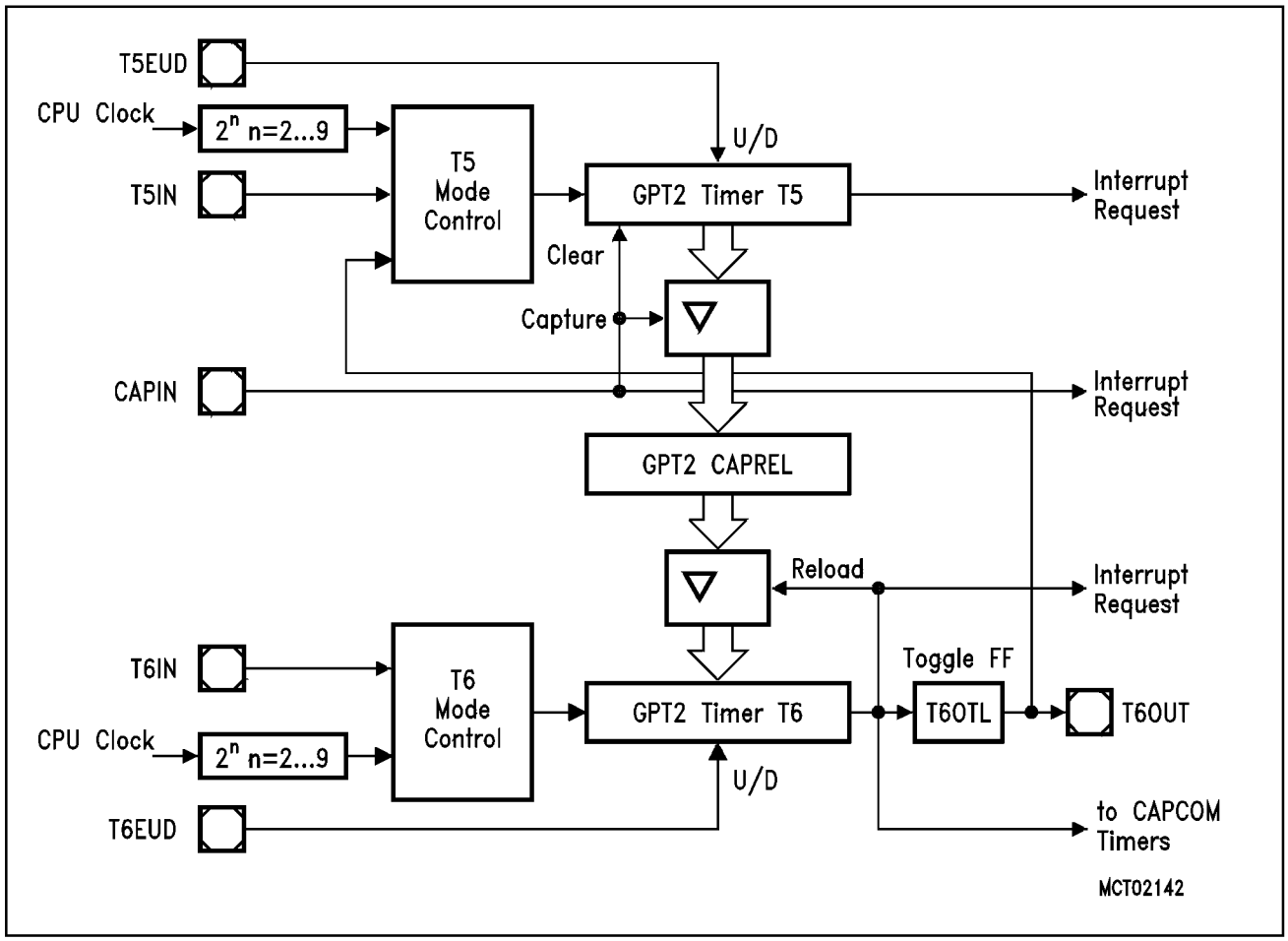



**Figure 8**
**Block Diagram of GPT2**

### 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 either by 2 or by 128. 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 25 µs and 420 ms can be monitored (@ 20 MHz). The default Watchdog Timer interval after reset is 6.55 ms (@ 20 MHz).

### A/D Converter

For analog signal measurement, a 10-bit A/D converter with 16 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 16 analog input channels, the remaining channel inputs can be used as digital input port pins.

The A/D converter of the C167CR-4RM 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 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 (eg. 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.

### Serial Channels

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

**The ASC0** is upward compatible with the serial ports of the Siemens 8-bit microcontroller families and supports full-duplex asynchronous communication at up to 625 KBaud and half-duplex synchronous communication at up to 2.5 MBaud @ 20 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 SSC** supports full-duplex synchronous communication at up to 5 Mbaud @ 20 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 3 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.

### CAN-Module

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

The module provides Full CAN functionality on up to 15 message objects. 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. The CAN-Module uses two pins of Port 4 to interface to a bus transceiver.

**Note:** When the CAN interface is to be used the segment address output on Port 4 must be limited to 4 bits, ie. A19...A16. This is necessary to enable the alternate function of the CAN interface pins.

### Parallel Ports

The C167CR-4RM provides up to 111 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. During the internal reset, all port pins are configured as inputs.

The input threshold of Port 2, Port 3, Port 7 and Port 8 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.
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 8 and Port 7 are associated with the capture inputs or compare outputs of the CAPCOM units and/or with the outputs of the PWM module.
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).
Port 5 is used for the analog input channels to the A/D converter or timer control signals.

All port lines that are not used for these alternate functions may be used as general purpose IO lines.

### Instruction Set Summary

The table below lists the instructions of the C167CR-4RM 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 **"C16x Family Instruction Set Manual"**.

This document also provides a detailled description of each instruction.

### 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 |

**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 und 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

The following table lists all SFRs which are implemented in the C167CR-4RM 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".

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).

### Special Function Registers Overview

| Name | Physical Address | 8-Bit Address | 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 | $0000_H$ |
| **BUSCON0 b** | $FF0C_H$ | $86_H$ | Bus Configuration Register 0 | $0XX0_H$ |
| **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 | $0000_H$ |
| **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 Control Register | $0000_H$ |
| **CC1** | $FE82_H$ | $41_H$ | CAPCOM Register 1 | $0000_H$ |
| **CC1IC b** | $FF7A_H$ | $BD_H$ | CAPCOM Register 1 Interrupt Control Register | $0000_H$ |
| **CC2** | $FE84_H$ | $42_H$ | CAPCOM Register 2 | $0000_H$ |
| **CC2IC b** | $FF7C_H$ | $BE_H$ | CAPCOM Register 2 Interrupt Control Register | $0000_H$ |

**Special Function Registers Overview** (cont'd)

| Name | Physical Address | 8-Bit Address | Description | Reset Value |
|---|---|---|---|---|
| **CC3** | $FE86_H$ | $43_H$ | CAPCOM Register 3 | $0000_H$ |
| **CC3IC** b | $FF7E_H$ | $BF_H$ | CAPCOM Register 3 Interrupt Control Register | $0000_H$ |
| **CC4** | $FE88_H$ | $44_H$ | CAPCOM Register 4 | $0000_H$ |
| **CC4IC** b | $FF80_H$ | $C0_H$ | CAPCOM Register 4 Interrupt Control Register | $0000_H$ |
| **CC5** | $FE8A_H$ | $45_H$ | CAPCOM Register 5 | $0000_H$ |
| **CC5IC** b | $FF82_H$ | $C1_H$ | CAPCOM Register 5 Interrupt Control Register | $0000_H$ |
| **CC6** | $FE8C_H$ | $46_H$ | CAPCOM Register 6 | $0000_H$ |
| **CC6IC** b | $FF84_H$ | $C2_H$ | CAPCOM Register 6 Interrupt Control Register | $0000_H$ |
| **CC7** | $FE8E_H$ | $47_H$ | CAPCOM Register 7 | $0000_H$ |
| **CC7IC** b | $FF86_H$ | $C3_H$ | CAPCOM Register 7 Interrupt Control Register | $0000_H$ |
| **CC8** | $FE90_H$ | $48_H$ | CAPCOM Register 8 | $0000_H$ |
| **CC8IC** b | $FF88_H$ | $C4_H$ | CAPCOM Register 8 Interrupt Control Register | $0000_H$ |
| **CC9** | $FE92_H$ | $49_H$ | CAPCOM Register 9 | $0000_H$ |
| **CC9IC** b | $FF8A_H$ | $C5_H$ | CAPCOM Register 9 Interrupt Control Register | $0000_H$ |
| **CC10** | $FE94_H$ | $4A_H$ | CAPCOM Register 10 | $0000_H$ |
| **CC10IC** b | $FF8C_H$ | $C6_H$ | CAPCOM Register 10 Interrupt Control Register | $0000_H$ |
| **CC11** | $FE96_H$ | $4B_H$ | CAPCOM Register 11 | $0000_H$ |
| **CC11IC** b | $FF8E_H$ | $C7_H$ | CAPCOM Register 11 Interrupt Control Register | $0000_H$ |
| **CC12** | $FE98_H$ | $4C_H$ | CAPCOM Register 12 | $0000_H$ |
| **CC12IC** b | $FF90_H$ | $C8_H$ | CAPCOM Register 12 Interrupt Control Register | $0000_H$ |
| **CC13** | $FE9A_H$ | $4D_H$ | CAPCOM Register 13 | $0000_H$ |
| **CC13IC** b | $FF92_H$ | $C9_H$ | CAPCOM Register 13 Interrupt Control Register | $0000_H$ |
| **CC14** | $FE9C_H$ | $4E_H$ | CAPCOM Register 14 | $0000_H$ |
| **CC14IC** b | $FF94_H$ | $CA_H$ | CAPCOM Register 14 Interrupt Control Register | $0000_H$ |
| **CC15** | $FE9E_H$ | $4F_H$ | CAPCOM Register 15 | $0000_H$ |
| **CC15IC** b | $FF96_H$ | $CB_H$ | CAPCOM Register 15 Interrupt Control Register | $0000_H$ |
| **CC16** | $FE60_H$ | $30_H$ | CAPCOM Register 16 | $0000_H$ |
| **CC16IC** b | $F160_H$ **E** | $B0_H$ | CAPCOM Register 16 Interrupt Control Register | $0000_H$ |
| **CC17** | $FE62_H$ | $31_H$ | CAPCOM Register 17 | $0000_H$ |

**Special Function Registers Overview** (cont'd)

| Name | Physical Address | 8-Bit Address | Description | Reset Value |
|---|---|---|---|---|
| **CC17IC b** | $F162_H$ **E** | $B1_H$ | CAPCOM Register 17 Interrupt Control Register | $0000_H$ |
| **CC18** | $FE64_H$ | $32_H$ | CAPCOM Register 18 | $0000_H$ |
| **CC18IC b** | $F164_H$ **E** | $B2_H$ | CAPCOM Register 18 Interrupt Control Register | $0000_H$ |
| **CC19** | $FE66_H$ | $33_H$ | CAPCOM Register 19 | $0000_H$ |
| **CC19IC b** | $F166_H$ **E** | $B3_H$ | CAPCOM Register 19 Interrupt Control Register | $0000_H$ |
| **CC20** | $FE68_H$ | $34_H$ | CAPCOM Register 20 | $0000_H$ |
| **CC20IC b** | $F168_H$ **E** | $B4_H$ | CAPCOM Register 20 Interrupt Control Register | $0000_H$ |
| **CC21** | $FE6A_H$ | $35_H$ | CAPCOM Register 21 | $0000_H$ |
| **CC21IC b** | $F16A_H$ **E** | $B5_H$ | CAPCOM Register 21 Interrupt Control Register | $0000_H$ |
| **CC22** | $FE6C_H$ | $36_H$ | CAPCOM Register 22 | $0000_H$ |
| **CC22IC b** | $F16C_H$ **E** | $B6_H$ | CAPCOM Register 22 Interrupt Control Register | $0000_H$ |
| **CC23** | $FE6E_H$ | $37_H$ | CAPCOM Register 23 | $0000_H$ |
| **CC23IC b** | $F16E_H$ **E** | $B7_H$ | CAPCOM Register 23 Interrupt Control Register | $0000_H$ |
| **CC24** | $FE70_H$ | $38_H$ | CAPCOM Register 24 | $0000_H$ |
| **CC24IC b** | $F170_H$ **E** | $B8_H$ | CAPCOM Register 24 Interrupt Control Register | $0000_H$ |
| **CC25** | $FE72_H$ | $39_H$ | CAPCOM Register 25 | $0000_H$ |
| **CC25IC b** | $F172_H$ **E** | $B9_H$ | CAPCOM Register 25 Interrupt Control Register | $0000_H$ |
| **CC26** | $FE74_H$ | $3A_H$ | CAPCOM Register 26 | $0000_H$ |
| **CC26IC b** | $F174_H$ **E** | $BA_H$ | CAPCOM Register 26 Interrupt Control Register | $0000_H$ |
| **CC27** | $FE76_H$ | $3B_H$ | CAPCOM Register 27 | $0000_H$ |
| **CC27IC b** | $F176_H$ **E** | $BB_H$ | CAPCOM Register 27 Interrupt Control Register | $0000_H$ |
| **CC28** | $FE78_H$ | $3C_H$ | CAPCOM Register 28 | $0000_H$ |
| **CC28IC b** | $F178_H$ **E** | $BC_H$ | CAPCOM Register 28 Interrupt Control Register | $0000_H$ |
| **CC29** | $FE7A_H$ | $3D_H$ | CAPCOM Register 29 | $0000_H$ |
| **CC29IC b** | $F184_H$ **E** | $C2_H$ | CAPCOM Register 29 Interrupt Control Register | $0000_H$ |
| **CC30** | $FE7C_H$ | $3E_H$ | CAPCOM Register 30 | $0000_H$ |
| **CC30IC b** | $F18C_H$ **E** | $C6_H$ | CAPCOM Register 30 Interrupt Control Register | $0000_H$ |
| **CC31** | $FE7E_H$ | $3F_H$ | CAPCOM Register 31 | $0000_H$ |
| **CC31IC b** | $F194_H$ **E** | $CA_H$ | CAPCOM Register 31 Interrupt Control Register | $0000_H$ |

**Special Function Registers Overview** (cont'd)

| Name | | Physical Address | | 8-Bit Address | Description | Reset Value |
|---|---|---|---|---|---|---|
| **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 | $0000_H$ |
| **CP** | | $FE10_H$ | | $08_H$ | CPU Context Pointer Register | $FC00_H$ |
| **CRIC** | **b** | $FF6A_H$ | | $B5_H$ | GPT2 CAPREL Interrupt Control Register | $0000_H$ |
| **CSP** | | $FE08_H$ | | $04_H$ | CPU Code Segment Pointer Register (read only) | $0000_H$ |
| **DP0L** | **b** | $F100_H$ | **E** | $80_H$ | P0L Direction Control Register | $00_H$ |
| **DP0H** | **b** | $F102_H$ | **E** | $81_H$ | P0H Direction Control Register | $00_H$ |
| **DP1L** | **b** | $F104_H$ | **E** | $82_H$ | P1L Direction Control Register | $00_H$ |
| **DP1H** | **b** | $F106_H$ | **E** | $83_H$ | P1H 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 | $00_H$ |
| **DP8** | **b** | $FFD6_H$ | | $EB_H$ | Port 8 Direction Control Register | $00_H$ |
| **DPP0** | | $FE00_H$ | | $00_H$ | CPU Data Page Pointer 0 Register (10 bits) | $0000_H$ |
| **DPP1** | | $FE02_H$ | | $01_H$ | CPU Data Page Pointer 1 Register (10 bits) | $0001_H$ |
| **DPP2** | | $FE04_H$ | | $02_H$ | CPU Data Page Pointer 2 Register (10 bits) | $0002_H$ |
| **DPP3** | | $FE06_H$ | | $03_H$ | CPU Data Page Pointer 3 Register (10 bits) | $0003_H$ |
| **EXICON** | **b** | $F1C0_H$ | **E** | $E0_H$ | External Interrupt 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 Register – High Word | $0000_H$ |
| **MDL** | | $FE0E_H$ | | $07_H$ | CPU Multiply Divide Register – Low Word | $0000_H$ |

**Special Function Registers Overview** (cont'd)

| Name | | Physical Address | | 8-Bit Address | Description | Reset Value |
|---|---|---|---|---|---|---|
| **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 | $0000_H$ |
| **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 | $00_H$ |
| **ODP8** | **b** | $F1D6_H$ | **E** | $EB_H$ | Port 8 Open Drain Control Register | $00_H$ |
| **ONES** | | $FF1E_H$ | | $8F_H$ | Constant Value 1's Register (read only) | $FFFF_H$ |
| **P0L** | **b** | $FF00_H$ | | $80_H$ | Port 0 Low Register (Lower half of PORT0) | $00_H$ |
| **P0H** | **b** | $FF02_H$ | | $81_H$ | Port 0 High Register (Upper half of PORT0) | $00_H$ |
| **P1L** | **b** | $FF04_H$ | | $82_H$ | Port 1 Low Register (Lower half of PORT1) | $00_H$ |
| **P1H** | **b** | $FF06_H$ | | $83_H$ | Port 1 High Register (Upper 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 (8 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) | $00_H$ |
| **P8** | **b** | $FFD4_H$ | | $EA_H$ | Port 8 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 | $0000_H$ |
| **PECC7** | | $FECE_H$ | | $67_H$ | PEC Channel 7 Control Register | $0000_H$ |
| **PICON** | | $F1C4_H$ | **E** | $E2_H$ | Port Input Threshold Control Register | $0000_H$ |
| **PP0** | | $F038_H$ | **E** | $1C_H$ | PWM Module Period Register 0 | $0000_H$ |
| **PP1** | | $F03A_H$ | **E** | $1D_H$ | PWM Module Period Register 1 | $0000_H$ |
| **PP2** | | $F03C_H$ | **E** | $1E_H$ | PWM Module Period Register 2 | $0000_H$ |

**Special Function Registers Overview** (cont'd)

| Name | Physical Address | 8-Bit Address | Description | Reset Value |
|---|---|---|---|---|
| **PP3** | $F03E_H$ **E** | $1F_H$ | PWM Module Period Register 3 | $0000_H$ |
| **PSW b** | $FF10_H$ | $88_H$ | CPU Program Status Word | $0000_H$ |
| **PT0** | $F030_H$ **E** | $18_H$ | PWM Module Up/Down Counter 0 | $0000_H$ |
| **PT1** | $F032_H$ **E** | $19_H$ | PWM Module Up/Down Counter 1 | $0000_H$ |
| **PT2** | $F034_H$ **E** | $1A_H$ | PWM Module Up/Down Counter 2 | $0000_H$ |
| **PT3** | $F036_H$ **E** | $1B_H$ | PWM Module Up/Down Counter 3 | $0000_H$ |
| **PW0** | $FE30_H$ | $18_H$ | PWM Module Pulse Width Register 0 | $0000_H$ |
| **PW1** | $FE32_H$ | $19_H$ | PWM Module Pulse Width Register 1 | $0000_H$ |
| **PW2** | $FE34_H$ | $1A_H$ | PWM Module Pulse Width Register 2 | $0000_H$ |
| **PW3** | $FE36_H$ | $1B_H$ | PWM Module Pulse Width Register 3 | $0000_H$ |
| **PWMCON0 b** | $FF30_H$ | $98_H$ | PWM Module Control Register 0 | $0000_H$ |
| **PWMCON1 b** | $FF32_H$ | $99_H$ | PWM Module Control Register 1 | $0000_H$ |
| **PWMIC b** | $F17E_H$ **E** | $BF_H$ | PWM Module Interrupt Control Register | $0000_H$ |
| **RP0H b** | $F108_H$ **E** | $84_H$ | System Startup Configuration Register (Rd. only) | $XX_H$ |
| **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 Control Register | $0000_H$ |
| **S0RBUF** | $FEB2_H$ | $59_H$ | Serial Channel 0 Receive Buffer Register (read only) | $XX_H$ |
| **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 (write only) | $00_H$ |
| **S0TIC b** | $FF6C_H$ | $B6_H$ | Serial Channel 0 Transmit Interrupt Control 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 | $0000_H$ |

**Special Function Registers Overview** (cont'd)

| Name | Physical Address | 8-Bit Address | Description | Reset Value |
|---|---|---|---|---|
| **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 | $FC00_H$ |
| **SYSCON b** | $FF12_H$ | $89_H$ | CPU System Configuration Register | $0xx0_H$ [1] |
| **T0** | $FE50_H$ | $28_H$ | CAPCOM Timer 0 Register | $0000_H$ |
| **T01CON b** | $FF50_H$ | $A8_H$ | CAPCOM Timer 0 and Timer 1 Control Register | $0000_H$ |
| **T0IC b** | $FF9C_H$ | $CE_H$ | CAPCOM Timer 0 Interrupt Control Register | $0000_H$ |
| **T0REL** | $FE54_H$ | $2A_H$ | CAPCOM Timer 0 Reload Register | $0000_H$ |
| **T1** | $FE52_H$ | $29_H$ | CAPCOM Timer 1 Register | $0000_H$ |
| **T1IC b** | $FF9E_H$ | $CF_H$ | CAPCOM Timer 1 Interrupt Control Register | $0000_H$ |
| **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 | $0000_H$ |

**Special Function Registers Overview** (cont'd)

| Name | | Physical Address | | 8-Bit Address | Description | Reset Value |
|---|---|---|---|---|---|---|
| **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 Control Register | $0000_H$ |
| **T7IC** | **b** | $F17A_H$ | **E** | $BE_H$ | CAPCOM Timer 7 Interrupt Control Register | $0000_H$ |
| **T7REL** | | $F054_H$ | **E** | $2A_H$ | CAPCOM Timer 7 Reload Register | $0000_H$ |
| **T8** | | $F052_H$ | **E** | $29_H$ | CAPCOM Timer 8 Register | $0000_H$ |
| **T8IC** | **b** | $F17C_H$ | **E** | $BF_H$ | CAPCOM Timer 8 Interrupt Control Register | $0000_H$ |
| **T8REL** | | $F056_H$ | **E** | $2B_H$ | CAPCOM Timer 8 Reload Register | $0000_H$ |
| **TFR** | **b** | $FFAC_H$ | | $D6_H$ | Trap Flag Register | $0000_H$ |
| **WDT** | | $FEAE_H$ | | $57_H$ | Watchdog Timer Register (read only) | $0000_H$ |
| **WDTCON** | | $FFAE_H$ | | $D7_H$ | Watchdog Timer Control Register | $000X_H$ [2] |
| **XP0IC** | **b** | $F186_H$ | **E** | $C3_H$ | CAN Module Interrupt Control Register | $0000_H$ |
| **XP1IC** | **b** | $F18E_H$ | **E** | $C7_H$ | X-Peripheral 1 Interrupt Control Register | $0000_H$ |
| **XP2IC** | **b** | $F196_H$ | **E** | $CB_H$ | X-Peripheral 2 Interrupt Control Register | $0000_H$ |
| **XP3IC** | **b** | $F19E_H$ | **E** | $CF_H$ | PLL 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] Bit WDTR indicates a watchdog timer triggered reset.

**Note:** The Interrupt Control Registers XPnIC are prepared to control interrupt requests from integrated X-Bus peripherals. Nodes, where no X-Peripherals are connected, may be used to generate software controlled interrupt requests by setting the respective XPnIR bit.

### Absolute Maximum Ratings

Ambient temperature under bias ($T_A$):
SAB-C167CR-4RM ..... 0 to +70 °C
SAF-C167CR-4RM ..... –40 to +85 °C
SAK-C167CR-4RM ..... –40 to +125 °C
Storage temperature ($T_{ST}$) ..... – 65 to +150 °C
Voltage on $V_{CC}$ pins with respect to ground ($V_{SS}$) ..... –0.5 to +6.5 V
Voltage on any pin with respect to ground ($V_{SS}$) ..... –0.5 to $V_{CC}$ +0.5 V
Input current on any pin during overload condition ..... –10 to +10 mA
Absolute sum of all input currents during overload condition ..... |100 mA|
Power dissipation ..... 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 overload conditions ($V_{IN}$>$V_{CC}$ or $V_{IN}$<$V_{SS}$) the voltage on pins with respect to ground ($V_{SS}$) must not exceed the values defined by the Absolute Maximum Ratings.

### Parameter Interpretation

The parameters listed in the following partly represent the characteristics of the C167CR-4RM 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 C167CR-4RM 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 C167CR-4RM.

## DC Characteristics

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V; $f_{CPU}$ = 20 MHz
$T_A$ = 0 to +70 °C for SAB-C167CR-4RM
$T_A$ = -40 to +85 °C for SAF-C167CR-4RM
$T_A$ = -40 to +125 °C for SAK-C167CR-4RM

| Parameter | Symbol | | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Input low voltage (TTL) | $V_{IL}$ | SR | – 0.5 | 0.2 $V_{CC}$ – 0.1 | V | – |
| Input low voltage (Special Threshold) | $V_{ILS}$ | SR | – 0.5 | 2.0 | V | – |
| Input high voltage, all except $\overline{\text{RSTIN}}$ and XTAL1 (TTL) | $V_{IH}$ | SR | 0.2 $V_{CC}$ + 0.9 | $V_{CC}$ + 0.5 | V | – |
| Input high voltage $\overline{\text{RSTIN}}$ | $V_{IH1}$ | SR | 0.6 $V_{CC}$ | $V_{CC}$ + 0.5 | V | – |
| Input high voltage XTAL1 | $V_{IH2}$ | SR | 0.7 $V_{CC}$ | $V_{CC}$ + 0.5 | V | – |
| Input high voltage (Special Threshold) | $V_{IHS}$ | SR | 0.8 $V_{CC}$ - 0.2 | $V_{CC}$ + 0.5 | V | – |
| Input Hysteresis (Special Threshold) | $HYS$ | | 400 | - | mV | – |
| Output low voltage [1] (PORT0, PORT1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$) | $V_{OL}$ | CC | – | 0.45 | V | $I_{OL}$ = 2.4 mA |
| Output low voltage [1] (all other outputs) | $V_{OL1}$ | CC | – | 0.45 | V | $I_{OL1}$ = 1.6 mA |
| Output high voltage [1] (PORT0, PORT1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$) | $V_{OH}$ | CC | 0.9 $V_{CC}$ 2.4 | – | V | $I_{OH}$ = – 500 µA $I_{OH}$ = – 2.4 mA |
| Output high voltage [1] [2] (all other outputs) | $V_{OH1}$ | CC | 0.9 $V_{CC}$ 2.4 | – | V V | $I_{OH}$ = – 250 µA $I_{OH}$ = – 1.6 mA |
| Input leakage current (Port 5) | $I_{OZ1}$ | CC | – | ±200 | nA | 0.45V < $V_{IN}$ < $V_{CC}$ |
| Input leakage current (all other) | $I_{OZ2}$ | CC | – | ±500 | nA | 0.45V < $V_{IN}$ < $V_{CC}$ |
| Overload current | $I_{OV}$ | SR | – | ±5 | mA | [3] [4] |
| $\overline{\text{RSTIN}}$ pullup resistor | $R_{RST}$ | CC | 50 | 250 | kΩ | – |
| Read/Write inactive current [5] | $I_{RWH}$ | [6] | – | -40 | µA | $V_{OUT}$ = 2.4 V |
| Read/Write active current [5] | $I_{RWL}$ | [7] | -500 | – | µA | $V_{OUT}$ = $V_{OLmax}$ |
| ALE inactive current [5] | $I_{ALEL}$ | [6] | – | 40 | µA | $V_{OUT}$ = $V_{OLmax}$ |
| ALE active current [5] | $I_{ALEH}$ | [7] | 500 | – | µA | $V_{OUT}$ = 2.4 V |
| Port 6 inactive current [5] | $I_{P6H}$ | [6] | – | -40 | µA | $V_{OUT}$ = 2.4 V |

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|
| | | min. | max. | | |
| Port 6 active current [5] | $I_{P6L}$ [7] | -500 | – | µA | $V_{OUT} = V_{OL1max}$ |
| PORT0 configuration current [5] | $I_{P0H}$ [6] | – | -10 | µA | $V_{IN} = V_{IHmin}$ |
| | $I_{P0L}$ [7] | -100 | – | µA | $V_{IN} = V_{ILmax}$ |
| XTAL1 input current | $I_{IL}$ CC | – | ±20 | µA | 0 V < $V_{IN}$ < $V_{CC}$ |
| Pin capacitance [3] (digital inputs/outputs) | $C_{IO}$ CC | – | 10 | pF | $f$ = 1 MHz<br>$T_A$ = 25 °C |
| Power supply current | $I_{CC}$ | – | 20 + 5 * $f_{CPU}$ | mA | $\overline{RSTIN} = V_{IL2}$<br>$f_{CPU}$ in [MHz] [8] |
| Idle mode supply current | $I_{ID}$ | – | 20 + 2 * $f_{CPU}$ | mA | $\overline{RSTIN} = V_{IH1}$<br>$f_{CPU}$ in [MHz] [8] |
| Power-down mode supply current | $I_{PD}$ | – | 100 | µA | $V_{CC}$ = 5.5 V [9] |

**Notes**

1) The pins of the C167CR-4RM are equipped with Low-Noise output drivers, which significantly improve the device's EMI performance. These Low-Noise drivers deliver their maximum current only until the respective target output level is reached. After that the output current is reduced. This results in an increased impedance of the driver, which attenuates electrical noise from the connected PCB tracks.
The current, which is specified in column „Test Conditions", is delivered in any case.

2) 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.

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

4) Overload conditions occur if the standard operatings conditions are exceeded, ie. the voltage on any pin exceeds the specified range (ie. $V_{OV} > V_{CC}$+0.5V or $V_{OV} < V_{SS}$-0.5V). The absolute sum of input overload currents on all port pins may not exceed **50 mA**. The supply voltage must remain within the specified limits.

5) This specification is only valid during Reset, or during Hold- or Adapt-mode. Port 6 pins are only affected, if they are used for $\overline{CS}$ output and the open drain function is not enabled.

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

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

8) The supply current is a function of the operating frequency. This dependency is illustrated in the figure below. These parameters are tested at $V_{CCmax}$ and 20 MHz CPU clock with all outputs disconnected and all inputs at $V_{IL}$ or $V_{IH}$.

9) This parameter is tested including leakage currents. All inputs (including pins configured as inputs) at 0 V to 0.1 V or at $V_{CC}$ – 0.1 V to $V_{CC}$, $V_{REF}$ = 0 V, all outputs (including pins configured as outputs) disconnected.

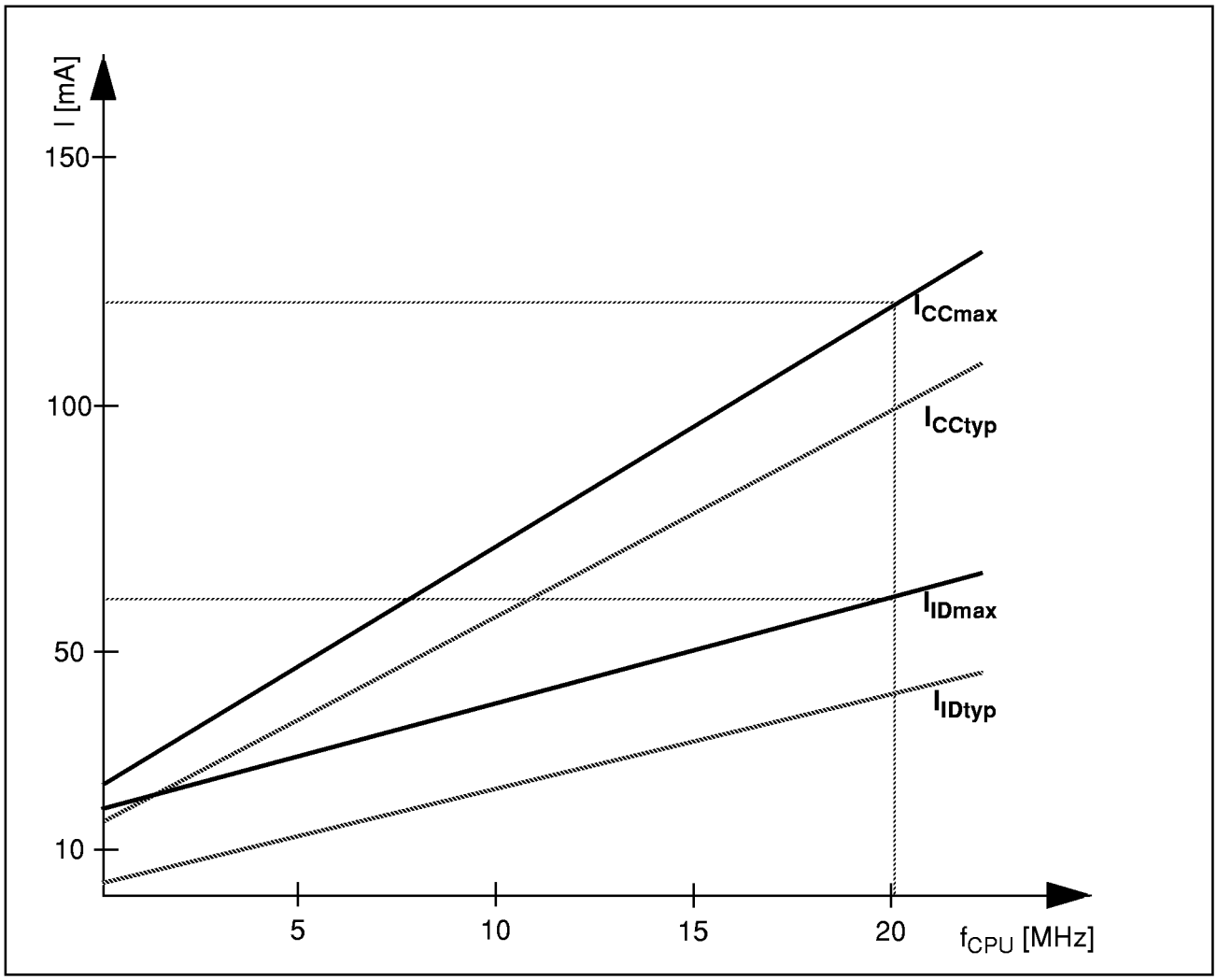



**Figure 9**
**Supply/Idle Current as a Function of Operating Frequency**

### A/D Converter Characteristics

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-4RM
$T_A$ = -40 to +85 °C for SAF-C167CR-4RM
$T_A$ = -40 to +125 °C for SAK-C167CR-4RM
4.0 V ≤ $V_{AREF}$ ≤ $V_{CC}$+0.1 V ; $V_{SS}$-0.1 V ≤ $V_{AGND}$ ≤ $V_{SS}$+0.2 V

| Parameter | Symbol | | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|---|
| | | | min. | max. | | |
| Analog input voltage range | $V_{AIN}$ | SR | $V_{AGND}$ | $V_{AREF}$ | V | 1) |
| Sample time | $t_S$ | CC | – | 2 $t_{SC}$ | | 2) 4) |
| Conversion time | $t_C$ | CC | – | 14 $t_{CC}$ + $t_S$ + 4TCL | | 3) 4) |
| Total unadjusted error | TUE | CC | – | ± 2 | LSB | 5) |
| Internal resistance of reference voltage source | $R_{AREF}$ | SR | – | $t_{CC}$ / 165 - 0.25 | kΩ | $t_{CC}$ in [ns] 6) 7) |
| Internal resistance of analog source | $R_{ASRC}$ | SR | – | $t_S$ / 330 - 0.25 | kΩ | $t_S$ in [ns] 2) 7) |
| ADC input capacitance | $C_{AIN}$ | CC | – | 33 | pF | 7) |

Sample time and conversion time of the C167CR-4RM's ADC are programmable. The table below should be used to calculate the above timings.

| ADCON.15\|14 (ADCTC) | Conversion clock $t_{CC}$ | ADCON.13\|12 (ADSTC) | Sample clock $t_{SC}$ |
|---|---|---|---|
| 00 | TCL * 24 | 00 | $t_{CC}$ |
| 01 | Reserved, do not use | 01 | $t_{CC}$ * 2 |
| 10 | TCL * 96 | 10 | $t_{CC}$ * 4 |
| 11 | TCL * 48 | 11 | $t_{CC}$ * 8 |

**Notes**

1) $V_{AIN}$ 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.

2) During the sample time the input capacitance $C_I$ 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 clock $t_{SC}$ depend on programming and can be taken from the table above.

3) 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 conversion clock $t_{CC}$ depend on programming and can be taken from the table above.

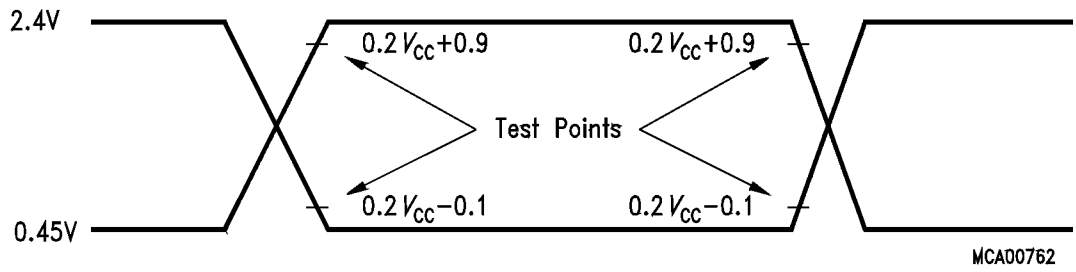
4) This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.

5) TUE is tested at $V_{AREF}$=5.0V, $V_{AGND}$=0V, $V_{CC}$=4.9V. It is guaranteed by design for all other voltages within the defined voltage range.
The specified TUE is guaranteed only if an overload condition (see $I_{OV}$ specification) occurs on maximum 2 not selected analog input pins and the absolute sum of input overload currents on all analog input pins does not exceed 10 mA.
During the reset calibration sequence the maximum TUE may be ±4 LSB.

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 $t_{CC}$. The maximum internal resistance results from the programmed conversion timing.

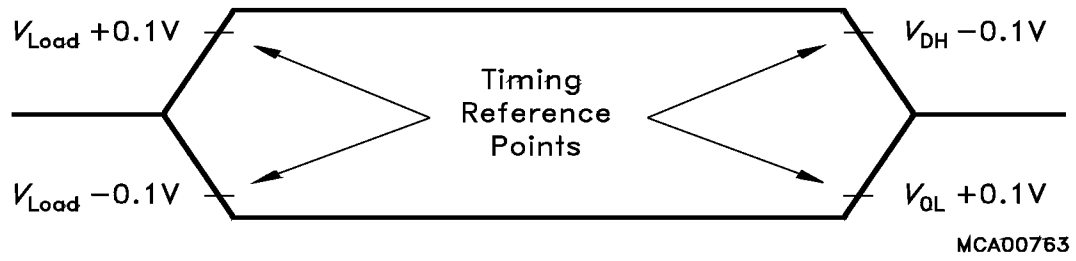
7) Not 100% tested, guaranteed by design characterization.

### 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'.

**Figure 10**
**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).

**Figure 11**
**Float Waveforms**

## AC Characteristics
### Definition of Internal Timing

The internal operation of the C167CR-4RM is controlled by the internal CPU clock $f_{CPU}$. Both edges of the CPU clock can trigger internal (eg. pipeline) or external (eg. 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 below).

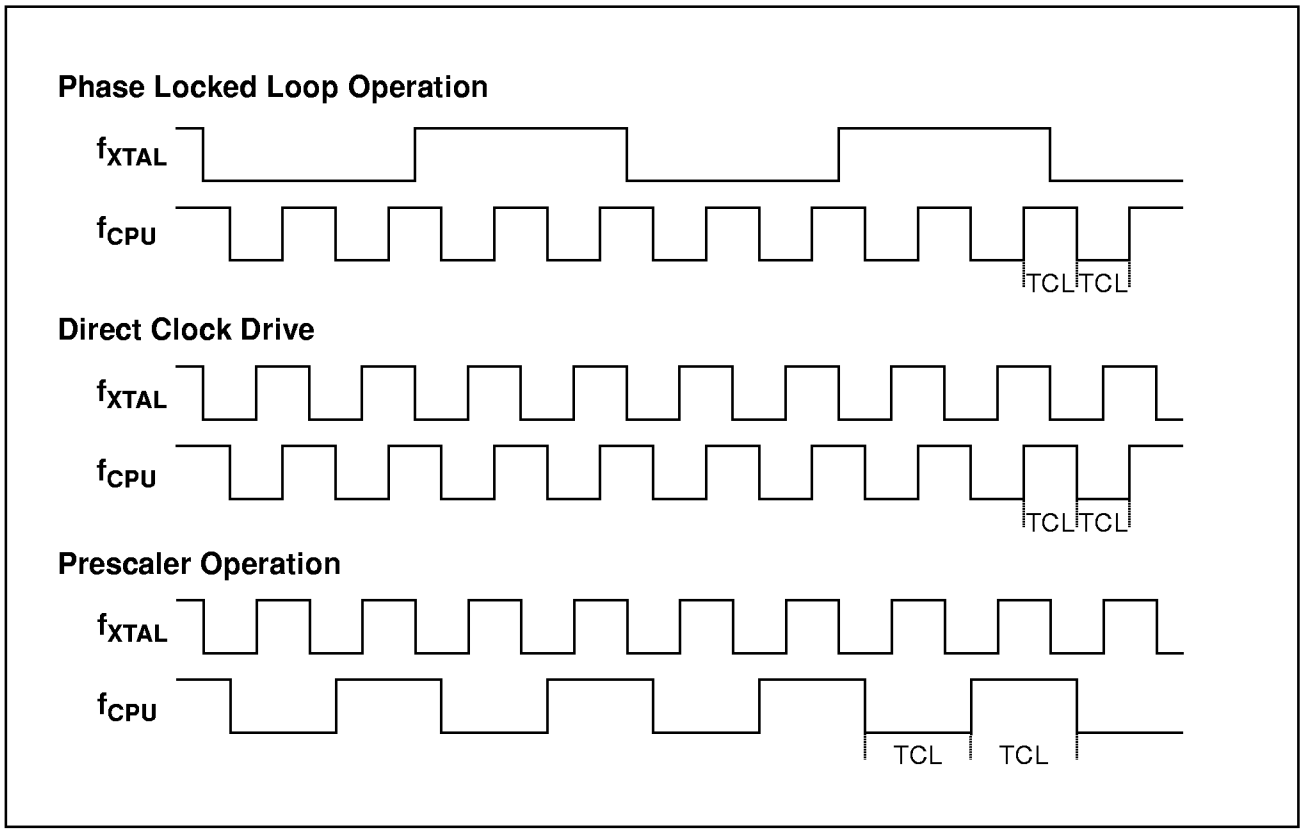



**Figure 12**
**Generation Mechanisms for the CPU Clock**

The CPU clock signal can be generated 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 C167CR-4RM.

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

The used mechanism to generate the CPU clock is selected during reset via the logic levels on pins P0.15-13 (P0H.7-5).

The table below associates the combinations of these three bits with the respective clock generation mode.

### C167CR-4RM Clock Generation Modes

| P0.15-13 (P0H.7-5) | | | CPU Frequency $f_{CPU} = f_{XTAL} * F$ | External Clock Input Range [1] | Notes |
|---|---|---|---|---|---|
| 1 | 1 | 1 | $f_{XTAL} * 4$ | 2.5 to 5 MHz | Default configuration |
| 1 | 1 | 0 | $f_{XTAL} * 3$ | 3.33 to 6.66 MHz | |
| 1 | 0 | 1 | $f_{XTAL} * 2$ | 5 to 10 MHz | |
| 1 | 0 | 0 | $f_{XTAL} * 5$ | 2 to 4 MHz | |
| 0 | 1 | 1 | $f_{XTAL} * 1$ | 1 to 20 MHz | Direct drive [2] |
| 0 | 1 | 0 | $f_{XTAL} * 1.5$ | 6.66 to 13.3 MHz | |
| 0 | 0 | 1 | $f_{XTAL} / 2$ | 2 to 40 MHz | CPU clock via prescaler |
| 0 | 0 | 0 | $f_{XTAL} * 2.5$ | 4 to 8 MHz | |

[1] The external clock input range refers to a CPU clock range of 10...20 MHz.
[2] The maximum frequency depends on the duty cycle of the external clock signal.

#### Prescaler Operation

When pins P0.15-13 (P0H.7-5) equal '001' during reset 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 $f_{XTAL}$ and the high and low time of $f_{CPU}$ (ie. the duration of an individual TCL) is defined by the period of the input clock $f_{XTAL}$.

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

#### Direct Drive

When pins P0.15-13 (P0H.7-5) equal '011' during reset 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 $f_{XTAL}$ so the high and low time of $f_{CPU}$ (ie. the duration of an individual TCL) is defined by the duty cycle of the input clock $f_{XTAL}$.

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:

$$TCL_{min} = 1/f_{XTAL} * DC_{min} \qquad \text{(DC = duty cycle)}$$

For two consecutive TCLs the deviation caused by the duty cycle of $f_{XTAL}$ is compensated so the duration of 2TCL is always $1/f_{XTAL}$. 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 $2TCL = 1/f_{XTAL}$.

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

**Phase Locked Loop**

For all other combinations of pins P0.15-13 (P0H.7-5) during reset 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 (ie. $f_{CPU} = f_{XTAL} * \mathbf{F}$). With every **F**'th transition of $f_{XTAL}$ the PLL circuit synchronizes the CPU clock to the input clock. This synchronization is done smoothely, ie. 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 $f_{XTAL}$. 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 below).
For a period of ***N*** * TCL the minimum value is computed using the corresponding deviation $D_N$:

$$TCL_{min} = TCL_{NOM} * (1 - D_N / 100)$$

$D_N = \pm(4 - N/15)$ [%],
where ***N*** = number of consecutive TCLs
and $1 \le N \le 40$.

So for a period of 3 TCLs (ie. ***N*** = 3): $D_3 = 4 - 3/15 = 3.8\%$,
and $(3TCL)_{min} = 3TCL_{NOM} * (1 - 3.8 / 100) = 3TCL_{NOM} * 0.962$ (72.15 nsec @ $f_{CPU}$ = 20 MHz).

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

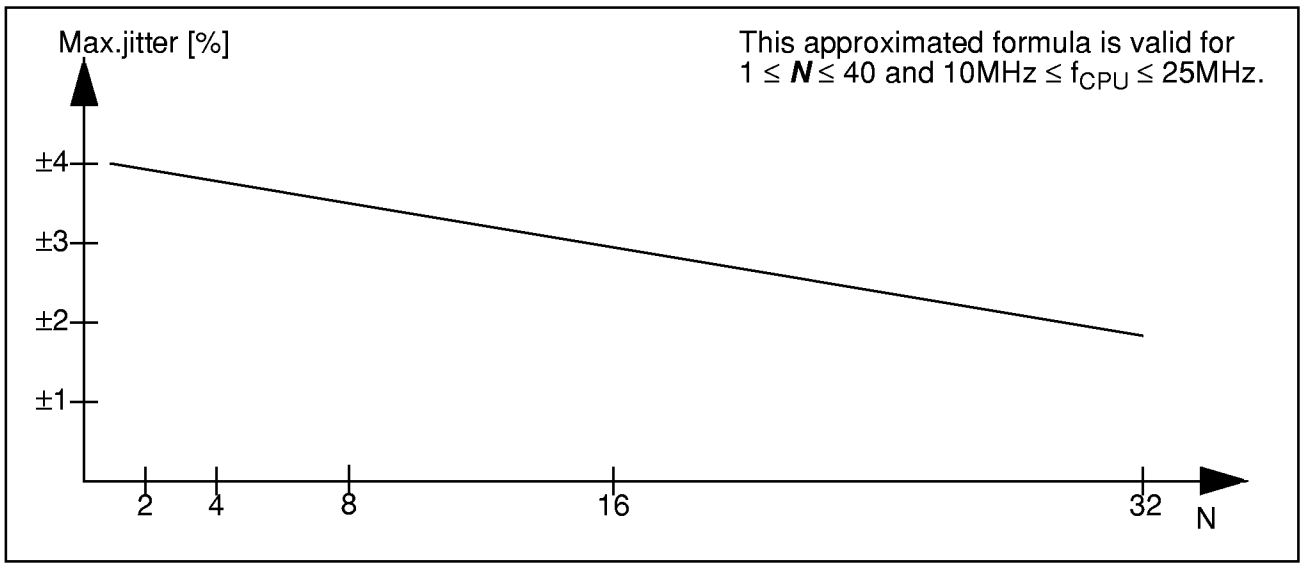



**Figure 13**
**Approximated Maximum PLL Jitter**

## AC Characteristics
## External Clock Drive XTAL1

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-4RM
$T_A$ = -40 to +85 °C for SAF-C167CR-4RM
$T_A$ = -40 to +125 °C for SAK-C167CR-4RM

| Parameter | Symbol | | Direct Drive 1:1 | | Prescaler 2:1 | | PLL 1:N | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | min. | max. | |
| Oscillator period | $t_{OSC}$ | SR | 50 | 1000 | 25 | 500 | 75 [1] | 500 [1] | ns |
| High time | $t_1$ | SR | 18 [2] | – | 6 | – | 10 | – | ns |
| Low time | $t_2$ | SR | 18 [2] | – | 6 | – | 10 | – | ns |
| Rise time | $t_3$ | SR | – | 10 [2] | – | 6 [2] | – | 10 [2] | ns |
| Fall time | $t_4$ | SR | – | 10 [2] | – | 6 [2] | – | 10 [2] | 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 $V_{IL}$ and $V_{IH2}$.

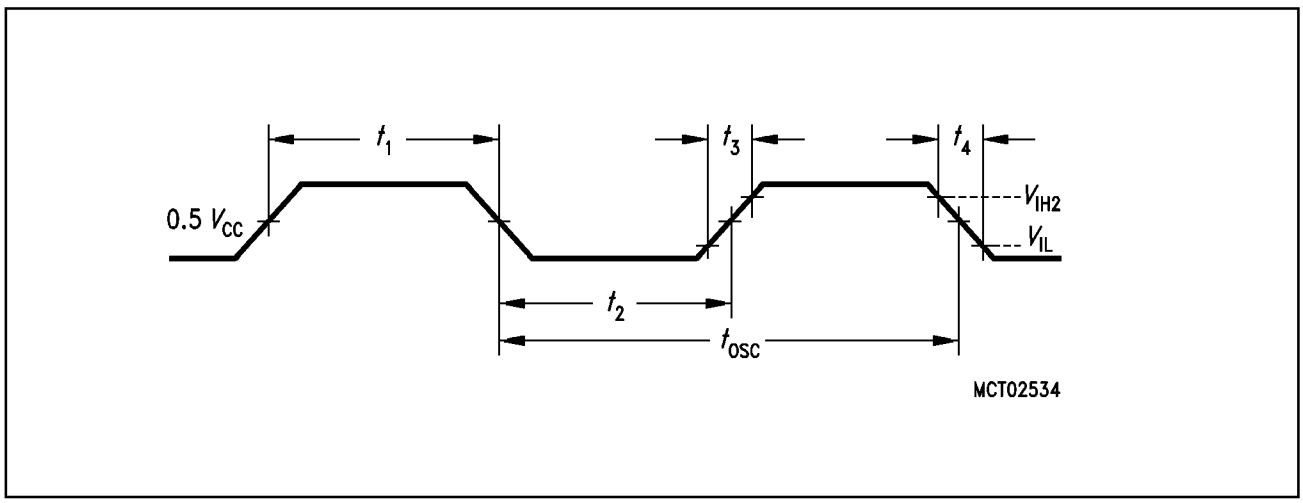



**Figure 14**
**External Clock Drive XTAL1**

### 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.

| 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>) |

### AC Characteristics
### Multiplexed Bus

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-4RM
$T_A$ = -40 to +85 °C for SAF-C167CR-4RM
$T_A$ = -40 to +125 °C for SAK-C167CR-4RM
$C_L$ (for PORT0, PORT1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
$C_L$ (for Port 6, $\overline{CS}$) = 100 pF
ALE cycle time = 6 TCL + 2$t_A$ + $t_C$ + $t_F$ (150 ns at 20-MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 20 MHz | | Variable CPU Clock 1/2TCL = 1 to 20 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| ALE high time | $t_5$ | CC | 15 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| Address setup to ALE | $t_6$ | CC | 9 + $t_A$ | – | TCL - 16 + $t_A$ | – | ns |
| Address hold after ALE | $t_7$ | CC | 15 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay) | $t_8$ | CC | 15 + $t_A$ | – | 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 | – | 31 | – | TCL + 6 | ns |
| $\overline{RD}$, $\overline{WR}$ low time (with RW-delay) | $t_{12}$ | CC | 40 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |
| $\overline{RD}$, $\overline{WR}$ low time (no RW-delay) | $t_{13}$ | CC | 65 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |

| Parameter | Symbol | | Max. CPU Clock = 20 MHz | | Variable CPU Clock 1/2TCL = 1 to 20 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| $\overline{RD}$ to valid data in (with RW-delay) | $t_{14}$ | SR | – | 30 + $t_C$ | – | 2TCL - 20 + $t_C$ | ns |
| $\overline{RD}$ to valid data in (no RW-delay) | $t_{15}$ | SR | – | 55 + $t_C$ | – | 3TCL - 20 + $t_C$ | ns |
| ALE low to valid data in | $t_{16}$ | SR | – | 55 + $t_A$ + $t_C$ | – | 3TCL - 20 + $t_A$ + $t_C$ | ns |
| Address to valid data in | $t_{17}$ | SR | – | 70 + 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 $\overline{RD}$ | $t_{19}$ | SR | – | 36 + $t_F$ | – | 2TCL - 14 + $t_F$ | ns |
| Data valid to $\overline{WR}$ | $t_{22}$ | CC | 30 + $t_C$ | – | 2TCL - 20 + $t_C$ | – | ns |
| Data hold after $\overline{WR}$ | $t_{23}$ | CC | 36 + $t_F$ | – | 2TCL - 14 + $t_F$ | – | ns |
| ALE rising edge after $\overline{RD}$, $\overline{WR}$ | $t_{25}$ | CC | 36 + $t_F$ | – | 2TCL - 14 + $t_F$ | – | ns |
| Address hold after $\overline{RD}$, $\overline{WR}$ | $t_{27}$ | CC | 36 + $t_F$ | – | 2TCL - 14 + $t_F$ | – | ns |
| ALE falling edge to $\overline{CS}$ | $t_{38}$ | CC | -4 - $t_A$ | 10 - $t_A$ | -4 - $t_A$ | 10 - $t_A$ | ns |
| $\overline{CS}$ low to Valid Data In | $t_{39}$ | SR | – | 55 + $t_C$ + 2$t_A$ | – | 3TCL - 20 + $t_C$ + 2$t_A$ | ns |
| $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$ | $t_{40}$ | CC | 61 + $t_F$ | – | 3TCL - 14 + $t_F$ | – | ns |
| ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay) | $t_{42}$ | CC | 21 + $t_A$ | – | TCL - 4 + $t_A$ | – | ns |
| ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay) | $t_{43}$ | CC | -4 + $t_A$ | – | -4 + $t_A$ | – | ns |
| Address float after $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay) | $t_{44}$ | CC | – | 0 | – | 0 | ns |
| Address float after $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay) | $t_{45}$ | CC | – | 25 | – | TCL | ns |
| $\overline{RdCS}$ to Valid Data In (with RW delay) | $t_{46}$ | SR | – | 26 + $t_C$ | – | 2TCL - 24 + $t_C$ | ns |

| Parameter | Symbol | | Max. CPU Clock = 20 MHz | | Variable CPU Clock 1/2TCL = 1 to 20 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| $\overline{\text{RdCS}}$ to Valid Data In (no RW delay) | $t_{47}$ | SR | – | 51 + $t_C$ | – | 3TCL - 24 + $t_C$ | ns |
| $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ Low Time (with RW delay) | $t_{48}$ | CC | 40 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |
| $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ Low Time (no RW delay) | $t_{49}$ | CC | 65 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |
| Data valid to $\overline{\text{WrCS}}$ | $t_{50}$ | CC | 36 + $t_C$ | – | 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 | – | 30 + $t_F$ | – | 2TCL - 20 + $t_F$ | ns |
| Address hold after $\overline{\text{RdCS}}$, $\overline{\text{WrCS}}$ | $t_{54}$ | CC | 30 + $t_F$ | – | 2TCL - 20 + $t_F$ | – | ns |
| Data hold after $\overline{\text{WrCS}}$ | $t_{56}$ | CC | 30 + $t_F$ | – | 2TCL - 20 + $t_F$ | – | ns |

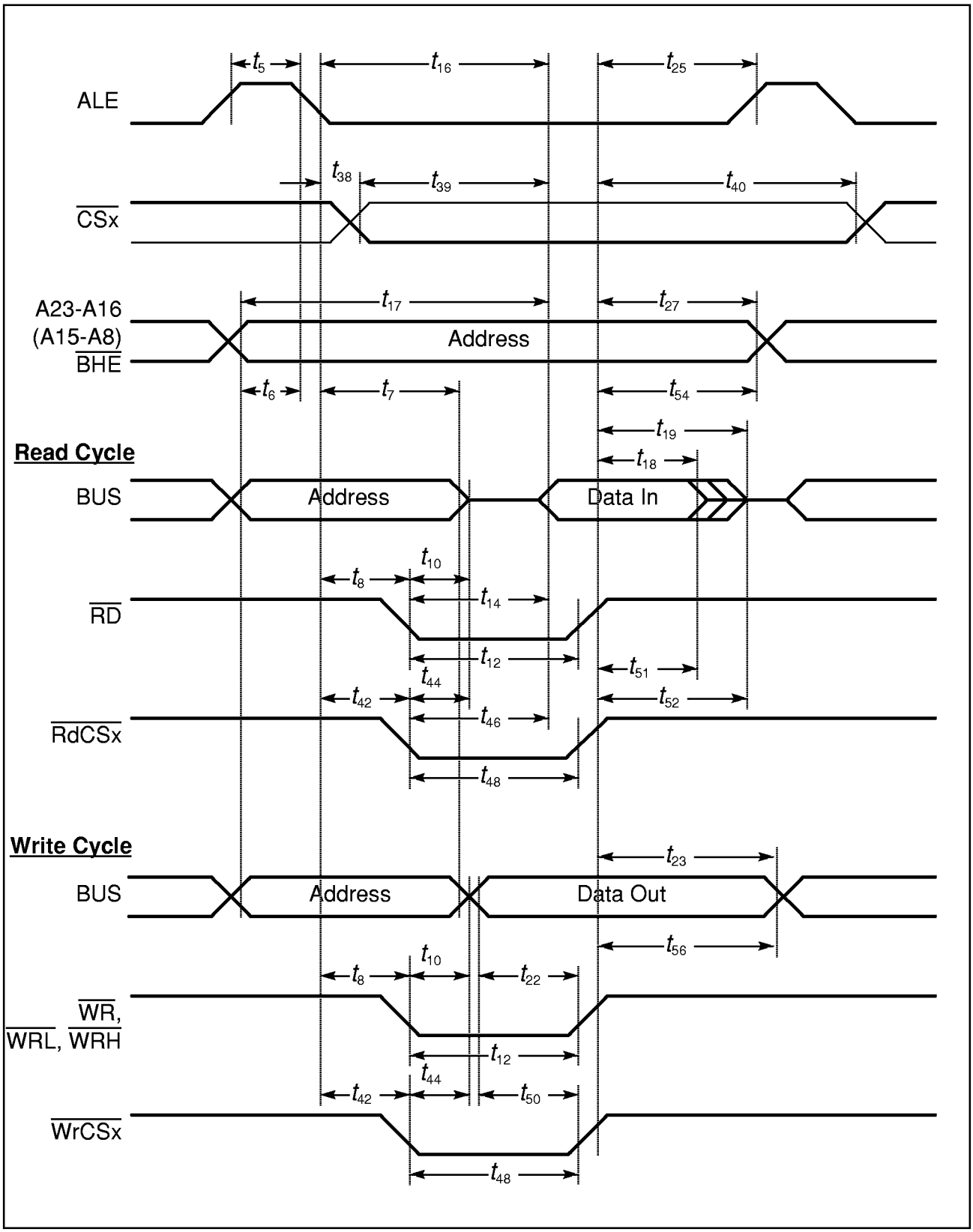



**Figure 15-1**
**External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE**

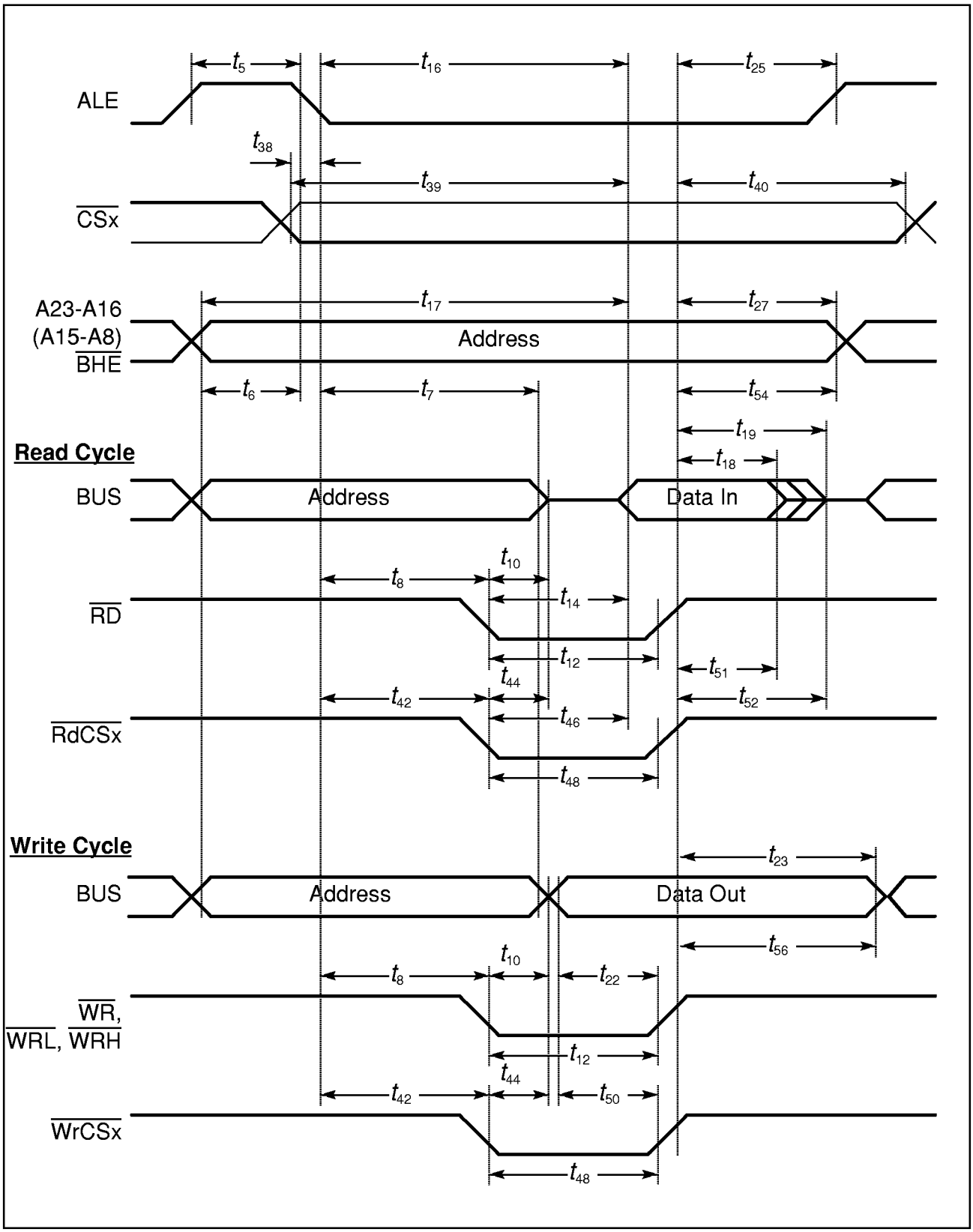



**Figure 15-2**
**External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Extended ALE**

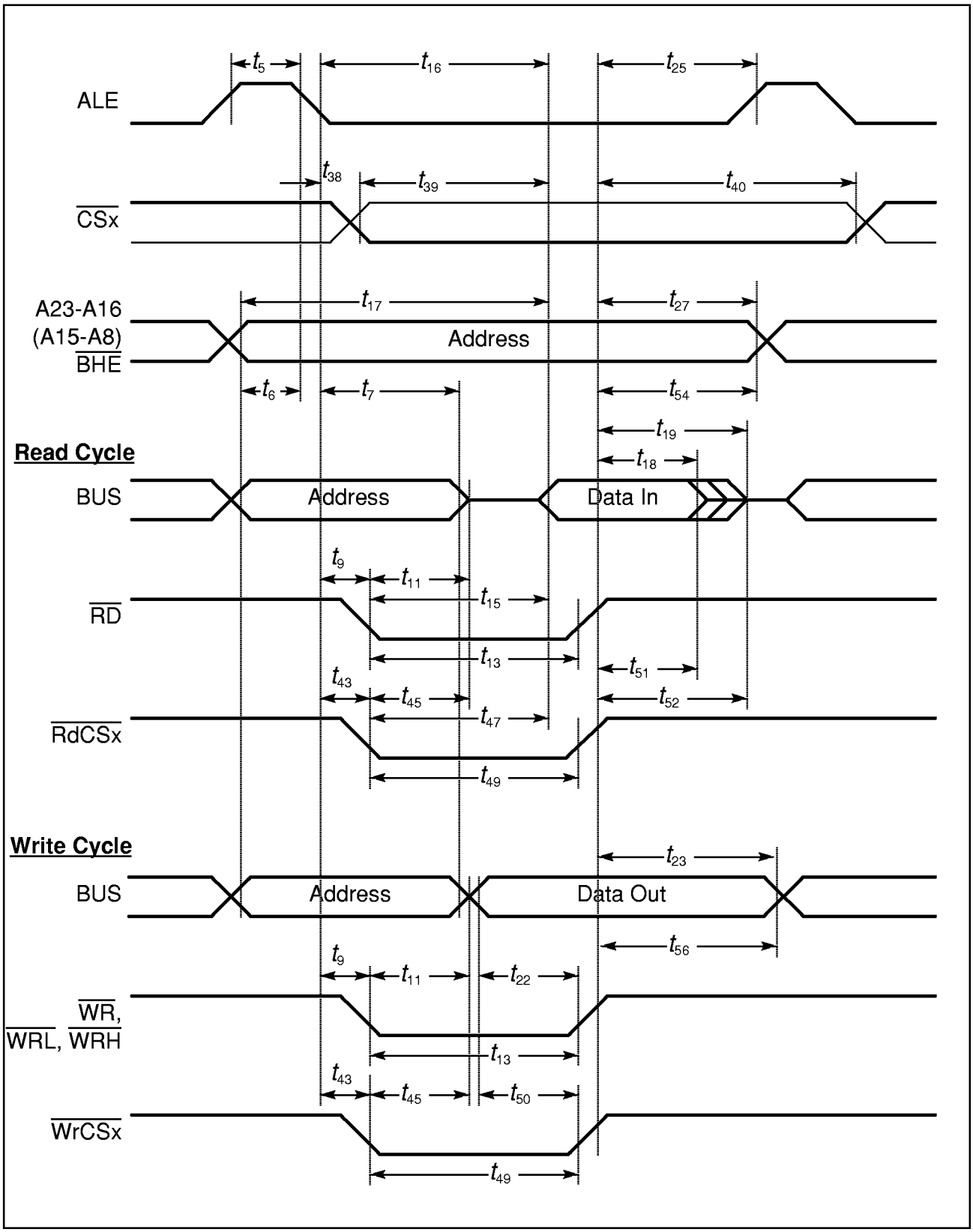



**Figure 15-3**
**External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE**

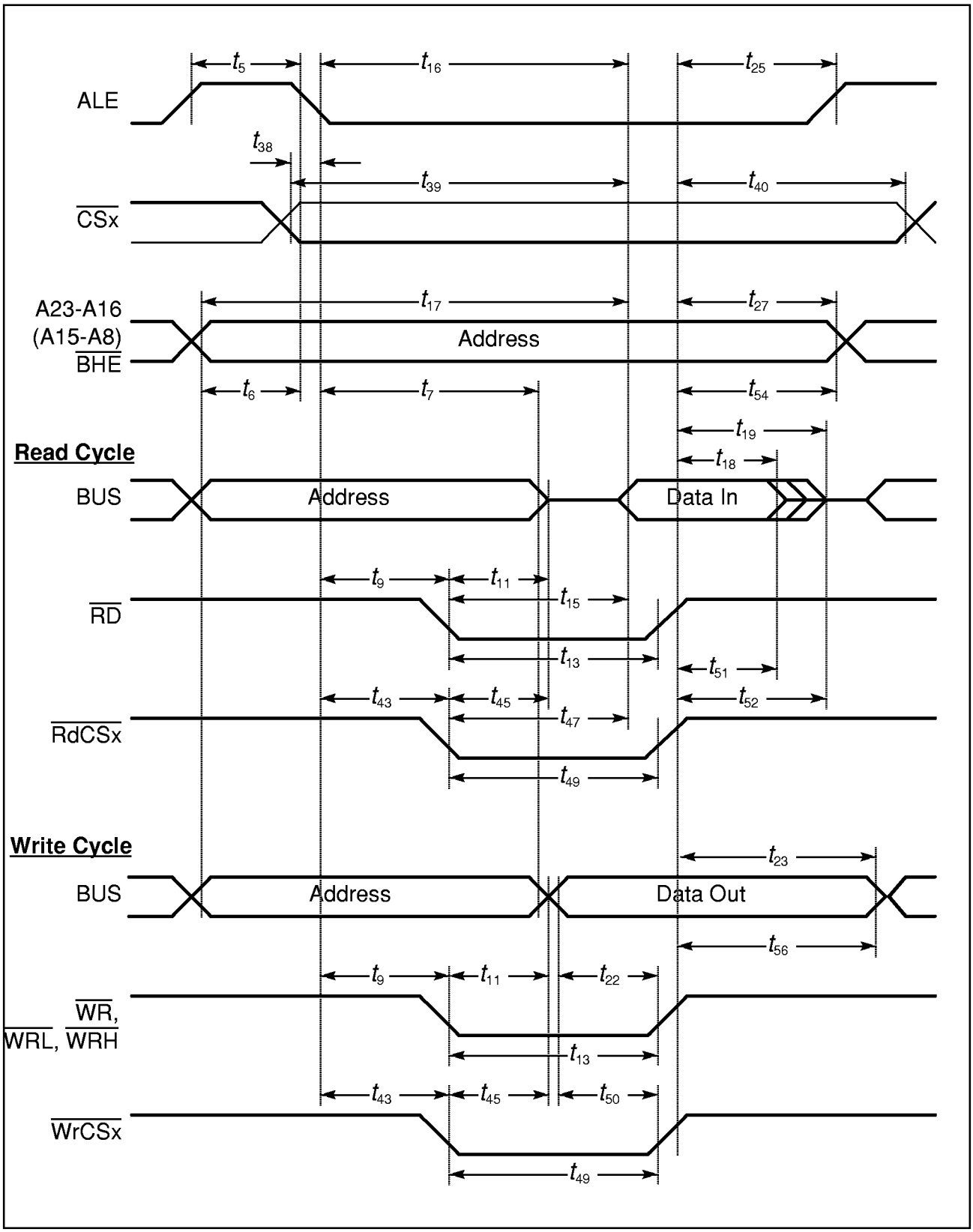



**Figure 15-4**
**External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE**

## AC Characteristics
## Demultiplexed Bus

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-4RM
$T_A$ = -40 to +85 °C for SAF-C167CR-4RM
$T_A$ = -40 to +125 °C for SAK-C167CR-4RM
$C_L$ (for PORT0, PORT1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
$C_L$ (for Port 6, $\overline{CS}$) = 100 pF
ALE cycle time = 4 TCL + 2$t_A$ + $t_C$ + $t_F$ (100 ns at 20-MHz CPU clock without waitstates)

| Parameter | Symbol | | Max. CPU Clock = 20 MHz | | Variable CPU Clock 1/2TCL = 1 to 20 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| ALE high time | $t_5$ | CC | 15 + $t_A$ | – | TCL - 10 + $t_A$ | – | ns |
| Address setup to ALE | $t_6$ | CC | 9 + $t_A$ | – | TCL - 16 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay) | $t_8$ | CC | 15 + $t_A$ | – | 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 | 40 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |
| $\overline{RD}$, $\overline{WR}$ low time (no RW-delay) | $t_{13}$ | CC | 65 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |
| $\overline{RD}$ to valid data in (with RW-delay) | $t_{14}$ | SR | – | 30 + $t_C$ | – | 2TCL - 20 + $t_C$ | ns |
| $\overline{RD}$ to valid data in (no RW-delay) | $t_{15}$ | SR | – | 55 + $t_C$ | – | 3TCL - 20 + $t_C$ | ns |
| ALE low to valid data in | $t_{16}$ | SR | – | 55 + $t_A$ + $t_C$ | – | 3TCL - 20 + $t_A$ + $t_C$ | ns |
| Address to valid data in | $t_{17}$ | SR | – | 70 + 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 | – | 36 + $t_F$ | – | 2TCL - 14 + 2$t_A$ + $t_F$ [1] | ns |
| Data float after $\overline{RD}$ rising edge (no RW-delay [1]) | $t_{21}$ | SR | – | 15 + $t_F$ | – | TCL - 10 + 2$t_A$ + $t_F$ [1] | ns |
| Data valid to $\overline{WR}$ | $t_{22}$ | CC | 30 + $t_C$ | – | 2TCL - 20 + $t_C$ | – | ns |
| Data hold after $\overline{WR}$ | $t_{24}$ | CC | 15 + $t_F$ | – | TCL - 10 + $t_F$ | – | ns |

| Parameter | Symbol | | Max. CPU Clock = 20 MHz | | Variable CPU Clock 1/2TCL = 1 to 20 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| 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 $\overline{CS}$ | $t_{38}$ | CC | -5 - $t_A$ | 10 - $t_A$ | -5 - $t_A$ | 10 - $t_A$ | ns |
| $\overline{CS}$ low to Valid Data In | $t_{39}$ | SR | – | 55 + $t_C$ + 2$t_A$ | – | 3TCL - 20 + $t_C$ + 2$t_A$ | ns |
| $\overline{CS}$ hold after $\overline{RD}$, $\overline{WR}$ | $t_{41}$ | CC | 11 + $t_F$ | – | TCL - 14 + $t_F$ | – | ns |
| ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW-delay) | $t_{42}$ | CC | 21 + $t_A$ | – | TCL - 4 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW-delay) | $t_{43}$ | CC | -6 + $t_A$ | – | -4 + $t_A$ | – | ns |
| $\overline{RdCS}$ to Valid Data In (with RW-delay) | $t_{46}$ | SR | – | 26 + $t_C$ | – | 2TCL - 24 + $t_C$ | ns |
| $\overline{RdCS}$ to Valid Data In (no RW-delay) | $t_{47}$ | SR | – | 51 + $t_C$ | – | 3TCL - 24 + $t_C$ | ns |
| $\overline{RdCS}$, $\overline{WrCS}$ Low Time (with RW-delay) | $t_{48}$ | CC | 40 + $t_C$ | – | 2TCL - 10 + $t_C$ | – | ns |
| $\overline{RdCS}$, $\overline{WrCS}$ Low Time (no RW-delay) | $t_{49}$ | CC | 65 + $t_C$ | – | 3TCL - 10 + $t_C$ | – | ns |
| Data valid to $\overline{WrCS}$ | $t_{50}$ | CC | 36 + $t_C$ | – | 2TCL - 14 + $t_C$ | – | ns |
| Data hold after $\overline{RdCS}$ | $t_{51}$ | SR | 0 | – | 0 | – | ns |
| Data float after $\overline{RdCS}$ (with RW-delay) | $t_{53}$ | SR | – | 30 + $t_F$ | – | 2TCL - 20 + $t_F$ | ns |
| Data float after $\overline{RdCS}$ (no RW-delay) | $t_{68}$ | SR | – | 5 + $t_F$ | – | TCL - 20 + $t_F$ | ns |
| Address hold after $\overline{RdCS}$, $\overline{WrCS}$ | $t_{55}$ | CC | -10 + $t_F$ | – | -10 + $t_F$ | – | ns |
| Data hold after $\overline{WrCS}$ | $t_{57}$ | CC | 11 + $t_F$ | – | TCL - 14 + $t_F$ | – | ns |

[1] RW-delay and $t_A$ refer to the next following bus cycle.
[2] It is guaranteed by design that read data are latched before the address changes.

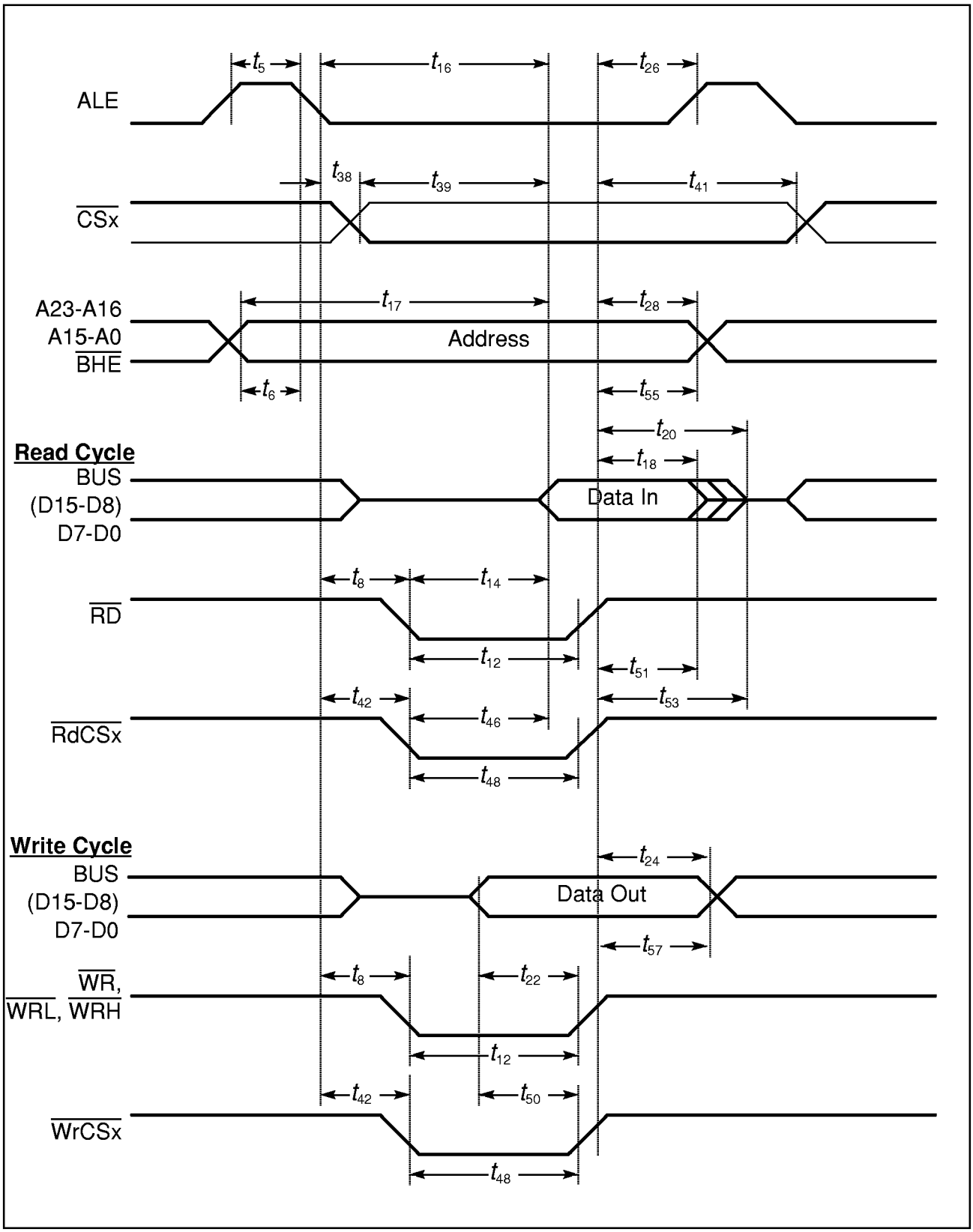



**Figure 16-1**
**External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Normal ALE**

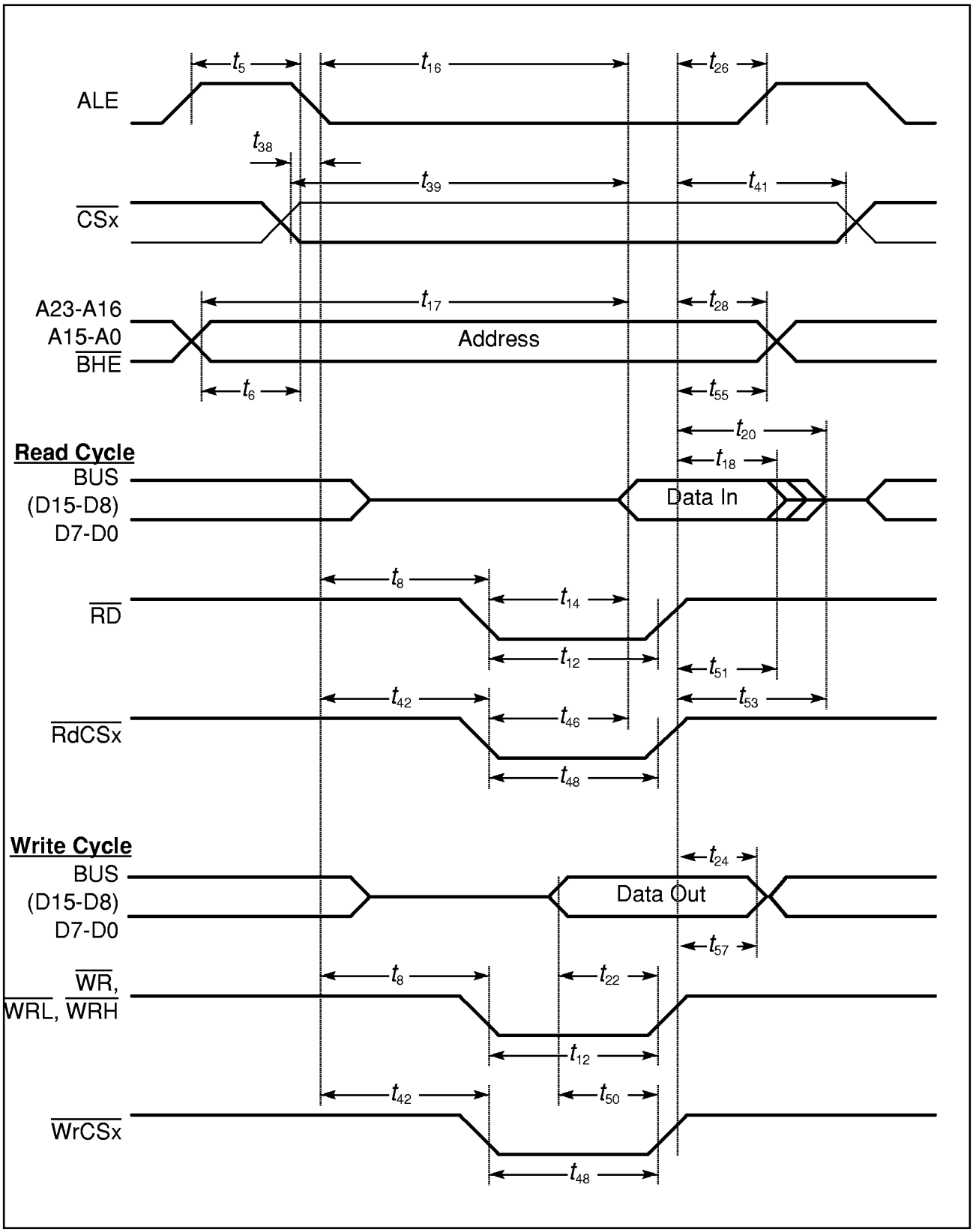



**Figure 16-2**
**External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE**

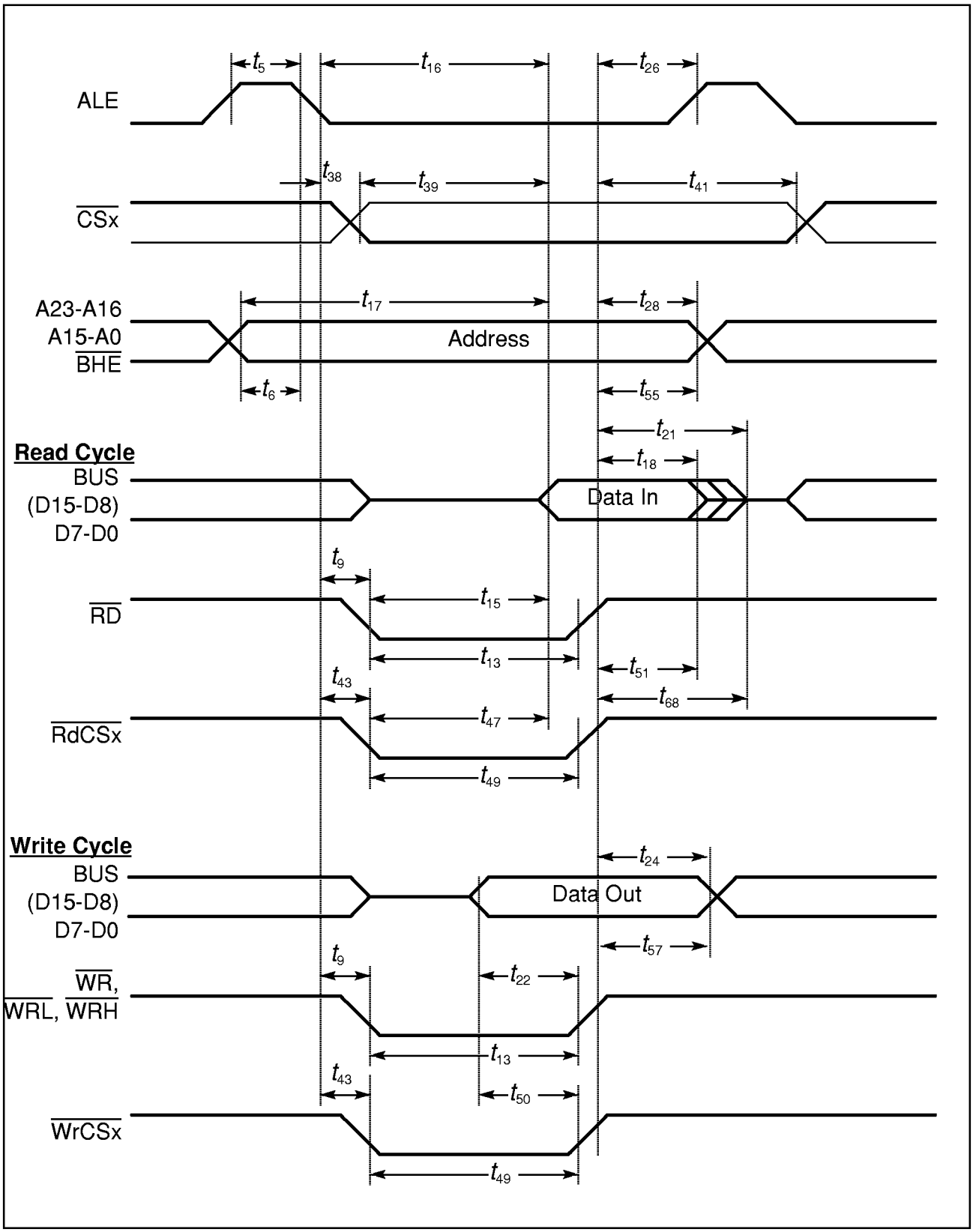



**Figure 16-3**
**External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE**

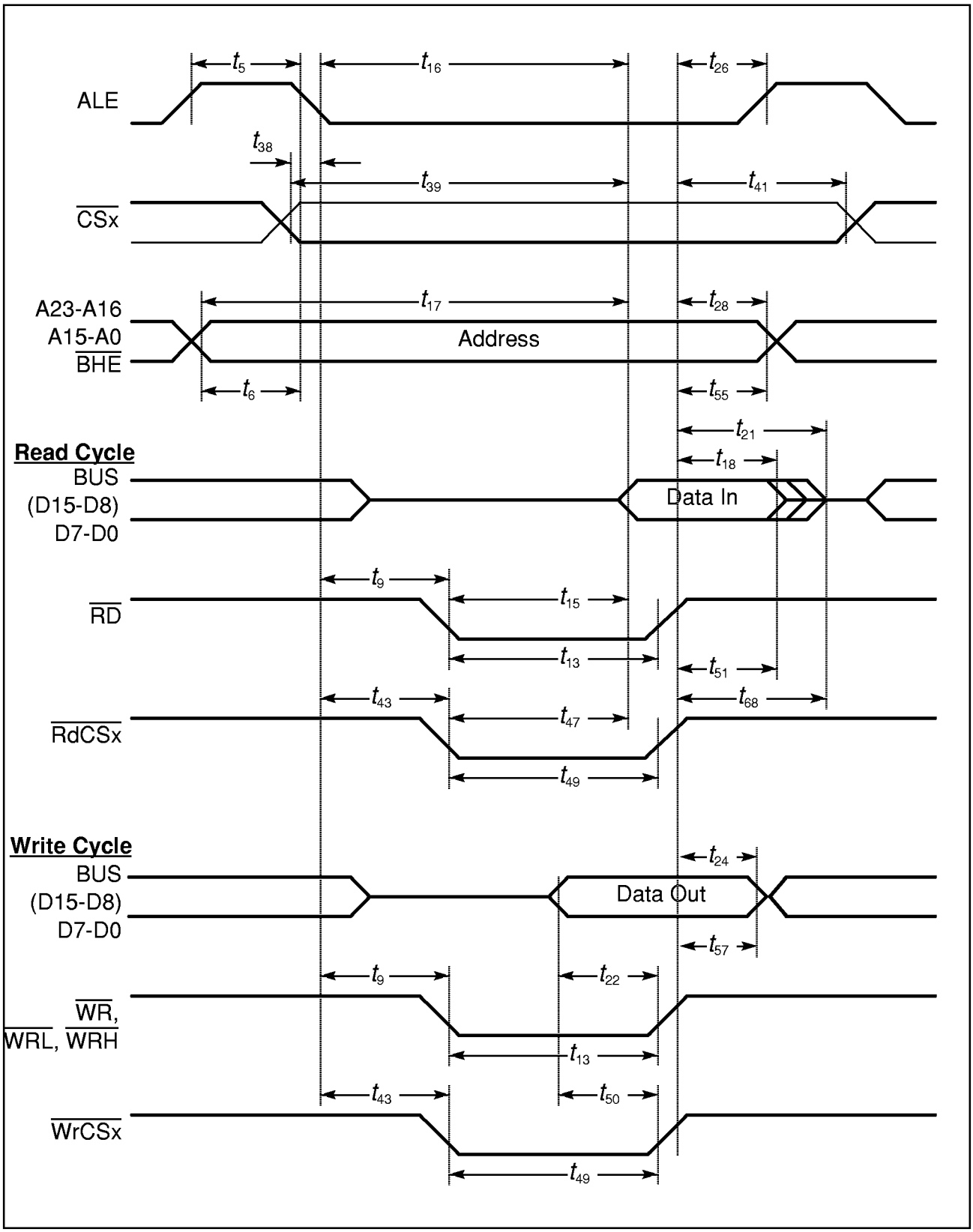



**Figure 16-4**
**External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE**

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

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-4RM
$T_A$ = -40 to +85 °C for SAF-C167CR-4RM
$T_A$ = -40 to +125 °C for SAK-C167CR-4RM
$C_L$ (for PORT0, PORT1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT) = 100 pF
$C_L$ (for Port 6, $\overline{\text{CS}}$) = 100 pF

| Parameter | Symbol | | Max. CPU Clock = 20 MHz | | Variable CPU Clock 1/2TCL = 1 to 20 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| CLKOUT cycle time | $t_{29}$ | CC | 50 | 50 | 2TCL | 2TCL | ns |
| CLKOUT high time | $t_{30}$ | CC | 19 | – | TCL – 6 | – | ns |
| CLKOUT low time | $t_{31}$ | CC | 15 | – | 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 | 64 | – | 2TCL + 14 | – | 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 | 1 + 2$t_A$ + $t_C$ + $t_F$ [2] | 0 | TCL - 24 + 2$t_A$ + $t_C$ + $t_F$ [2] | ns |

**Notes**

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 2$t_A$ and $t_C$ refer to the next following bus cycle, $t_F$ refers to the current bus cycle.

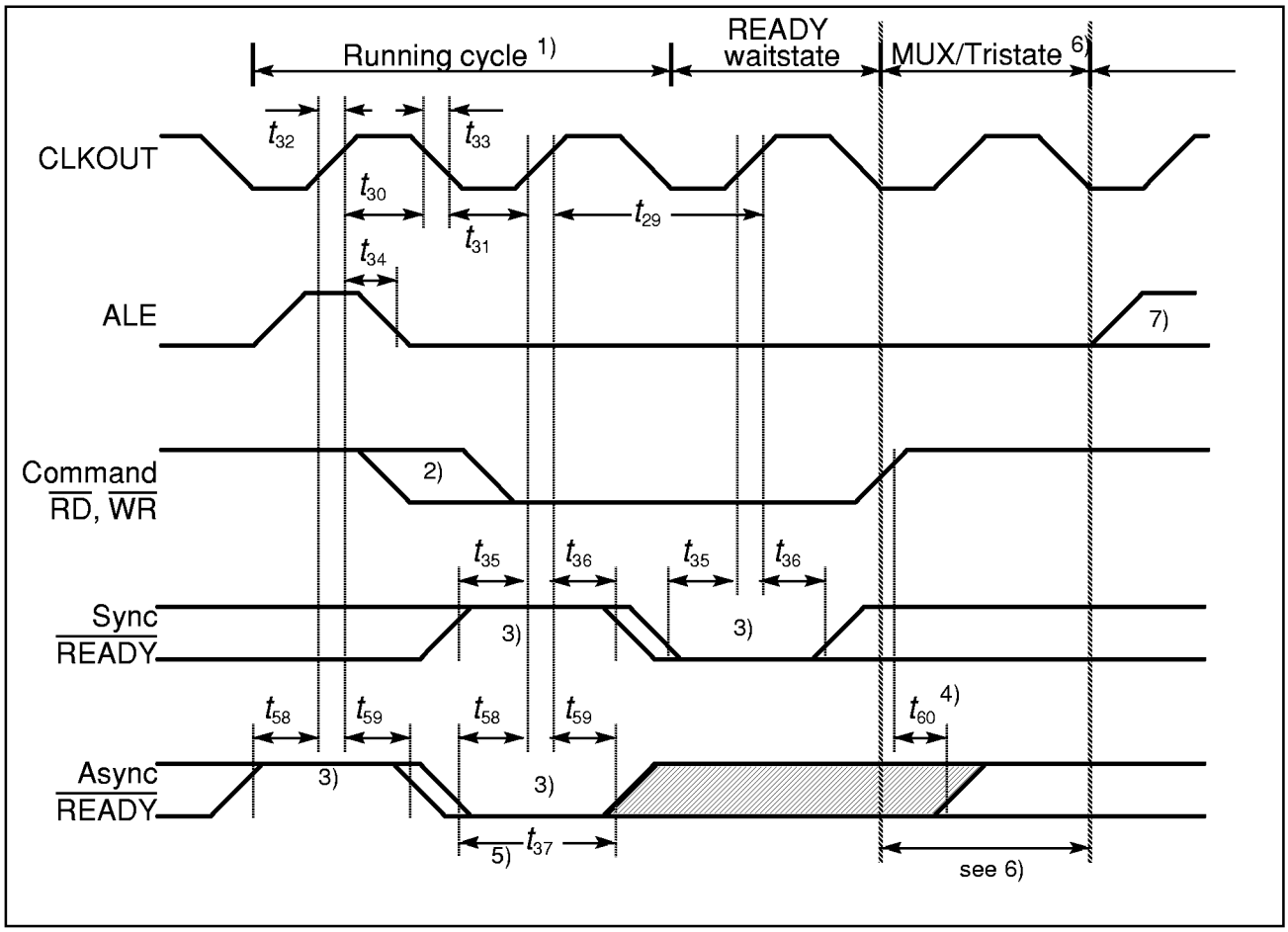



**Figure 17**
**CLKOUT and $\overline{\text{READY}}$**

**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) $\overline{\text{READY}}$ sampled HIGH at this sampling point generates a READY controlled waitstate,
$\overline{\text{READY}}$ sampled LOW at this sampling point terminates the currently running bus cycle.

4) $\overline{\text{READY}}$ may be deactivated in response to the trailing (rising) edge of the corresponding command ($\overline{\text{RD}}$ or $\overline{\text{WR}}$).

5) If the Asynchronous $\overline{\text{READY}}$ signal does not fulfill the indicated setup and hold times with respect to CLKOUT (eg. because CLKOUT is not enabled), it must fulfill $t_{37}$ in order to be safely synchronized. This is guaranteed, if $\overline{\text{READY}}$ 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

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-4RM
$T_A$ = -40 to +85 °C for SAF-C167CR-4RM
$T_A$ = -40 to +125 °C for SAK-C167CR-4RM
$C_L$ (for PORT0, PORT1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
$C_L$ (for Port 6, $\overline{CS}$) = 100 pF

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

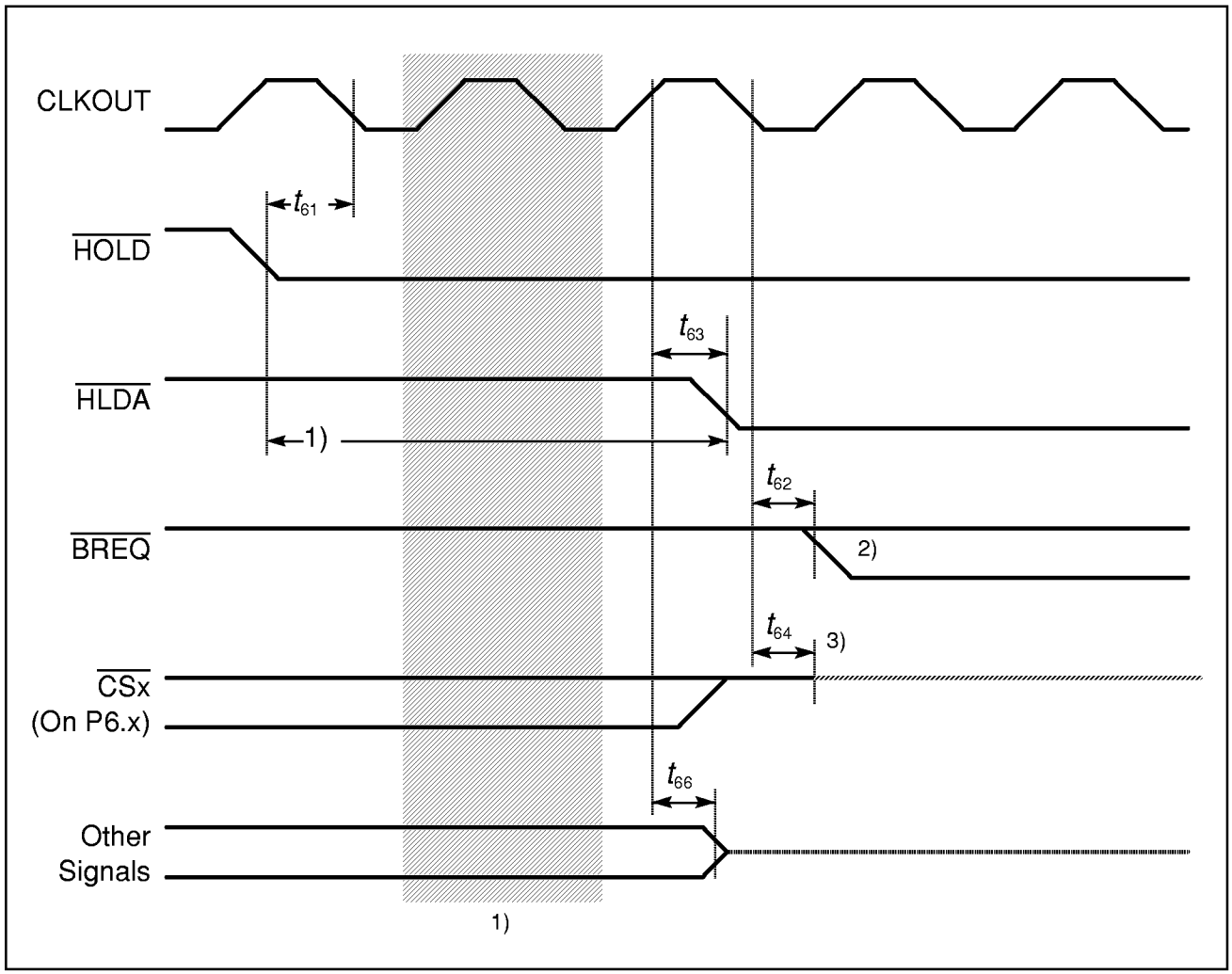



**Figure 18**
**External Bus Arbitration, Releasing the Bus**

**Notes**

1) The C167CR-4RM 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 $t_{64}$.

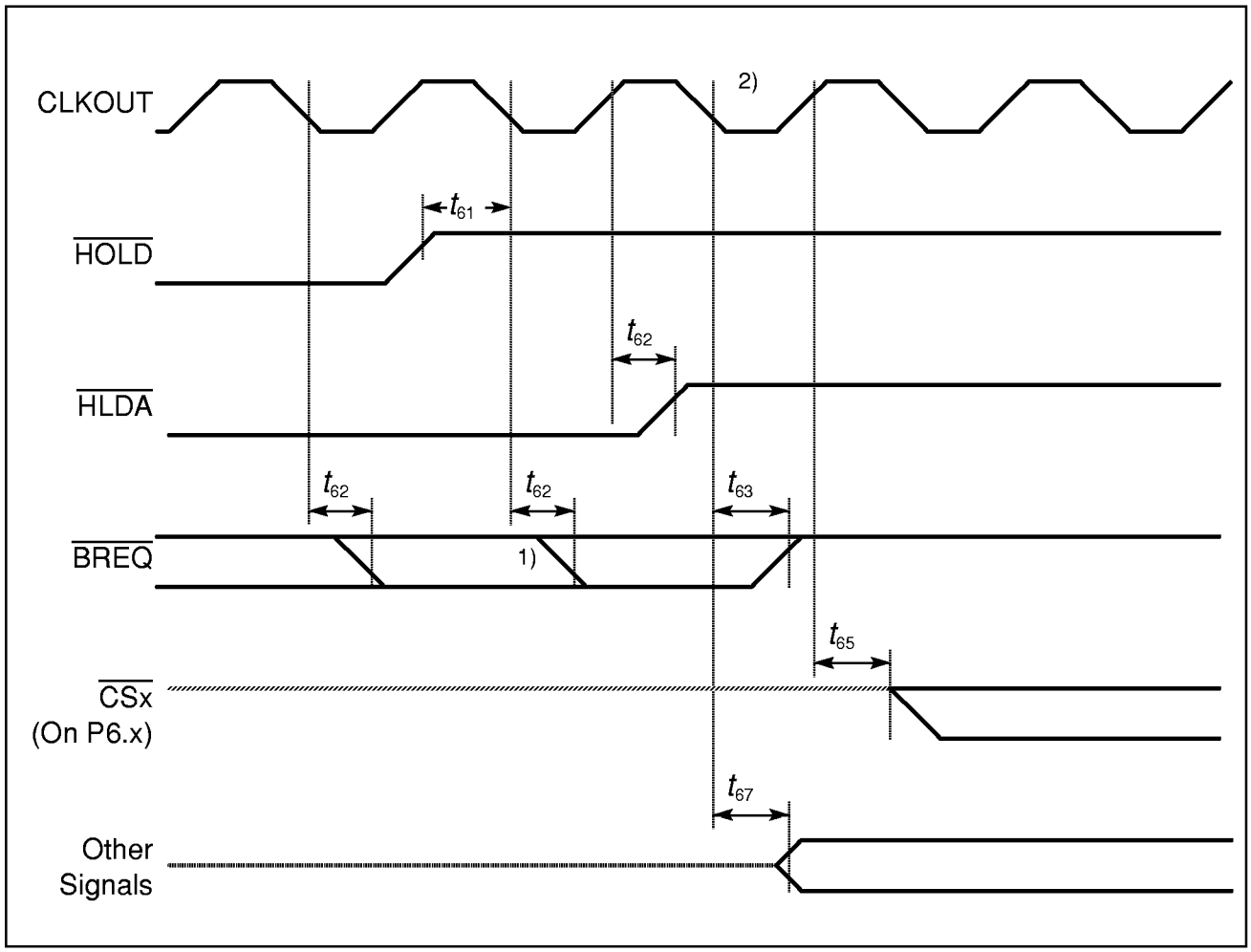



**Figure 19**
**External Bus Arbitration, (Regaining the Bus)**

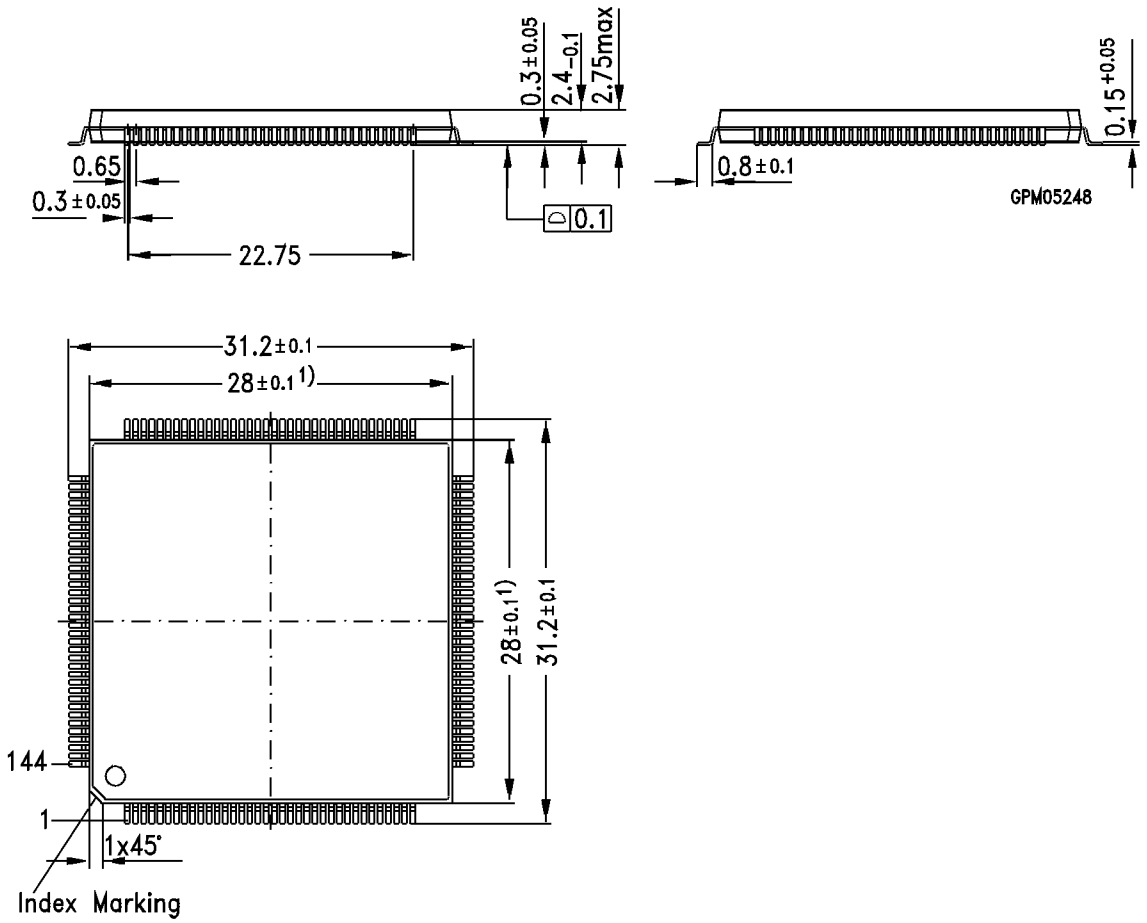
**Notes**

1) This is the last chance for $\overline{\text{BREQ}}$ to trigger the indicated regain-sequence.
Even if $\overline{\text{BREQ}}$ is activated earlier, the regain-sequence is initiated by $\overline{\text{HOLD}}$ going high.
Please note that $\overline{\text{HOLD}}$ may also be deactivated without the C167CR-4RM requesting the bus.

2) The next C167CR-4RM driven bus cycle may start here.

## Package Outline

**Plastic Package, P-MQFP-144-1 (SMD)**
(Plastic Metric Quad Flat Package)




1) Does not include plastic or metal protrusions of 0.25max per side

**Figure 20**

### Sorts of Packing

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

SMD = Surface Mounted Device

Dimensions in mm