SIEMENS

# Microcomputer Components

## 16-Bit CMOS Single-Chip Microcontroller with 128 KByte Flash EPROM

# C167CR-16F



Data Sheet 03.97 Advance Information

SIEMENS

# Microcomputer Components

16-Bit CMOS Single-Chip Microcontroller
with 128 KByte Flash EPROM

# C167CR-16F



**Data Sheet** 03.97 Advance Information

| C167CR-16F<br>Revision History: | Original Version 03.97 (Advance Information) |
|---|---|
| Previous Releases: | - |
| **Page** | **Subjects** |
| | |

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

**Edition 03.97**

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

SIEMENS

## C166-Family of
## High-Performance CMOS 16-Bit Microcontrollers

# C167CR-16F

## Advance Information
## C167CR-16F 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 MByte Linear Address Space for Code and Data
- 2 KByte On-Chip Internal RAM (IRAM)
- 2 KByte On-Chip Extension RAM (XRAM)
- 128 KByte On-Chip Flash EPROM with Bank Erase Feature and Read Protection
- Dedicated Flash Control Register with Operation Lock Mechanism
- 12 V External Flash Programming Voltage
- Flash Program Verify and Erase Verify Modes with 1000 Program/Erase Cycles
- Programmable External Bus Characteristics for Different Address Ranges
- 8-Bit or 16-Bit External Data Bus
- Multiplexed or Demultiplexed Address/Data Buses with 5 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 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 and one 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-16FM** and the **SAF-C167CR-16FM** .
For simplicity all versions are referred to by the term **C167CR-16F** throughout this document.

## Introduction

The C167CR-16F 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 high-speed RAM, on-chip Flash memory and clock generation via PLL.

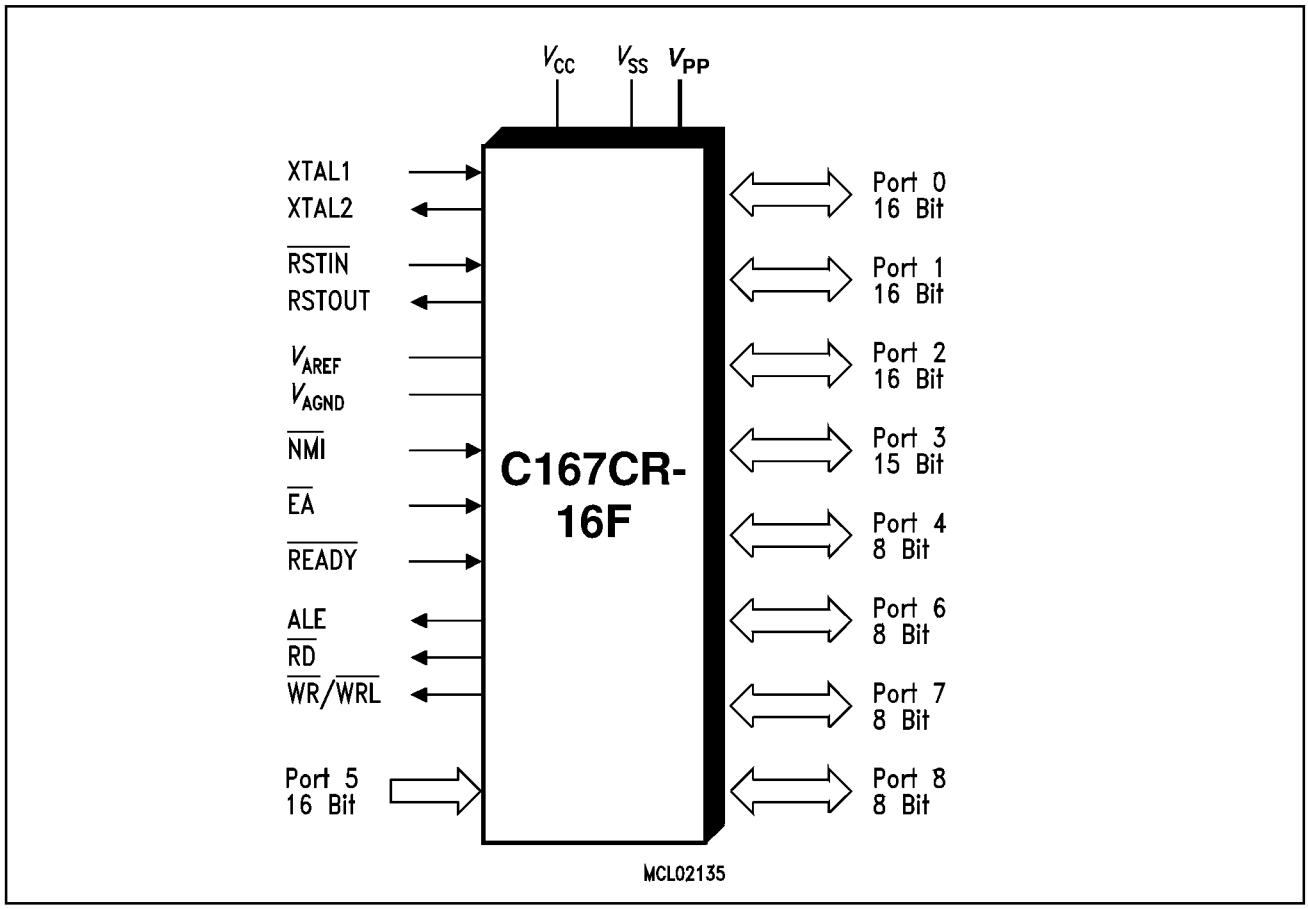



**Figure 1**
**Logic Symbol**

## 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-16F 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)

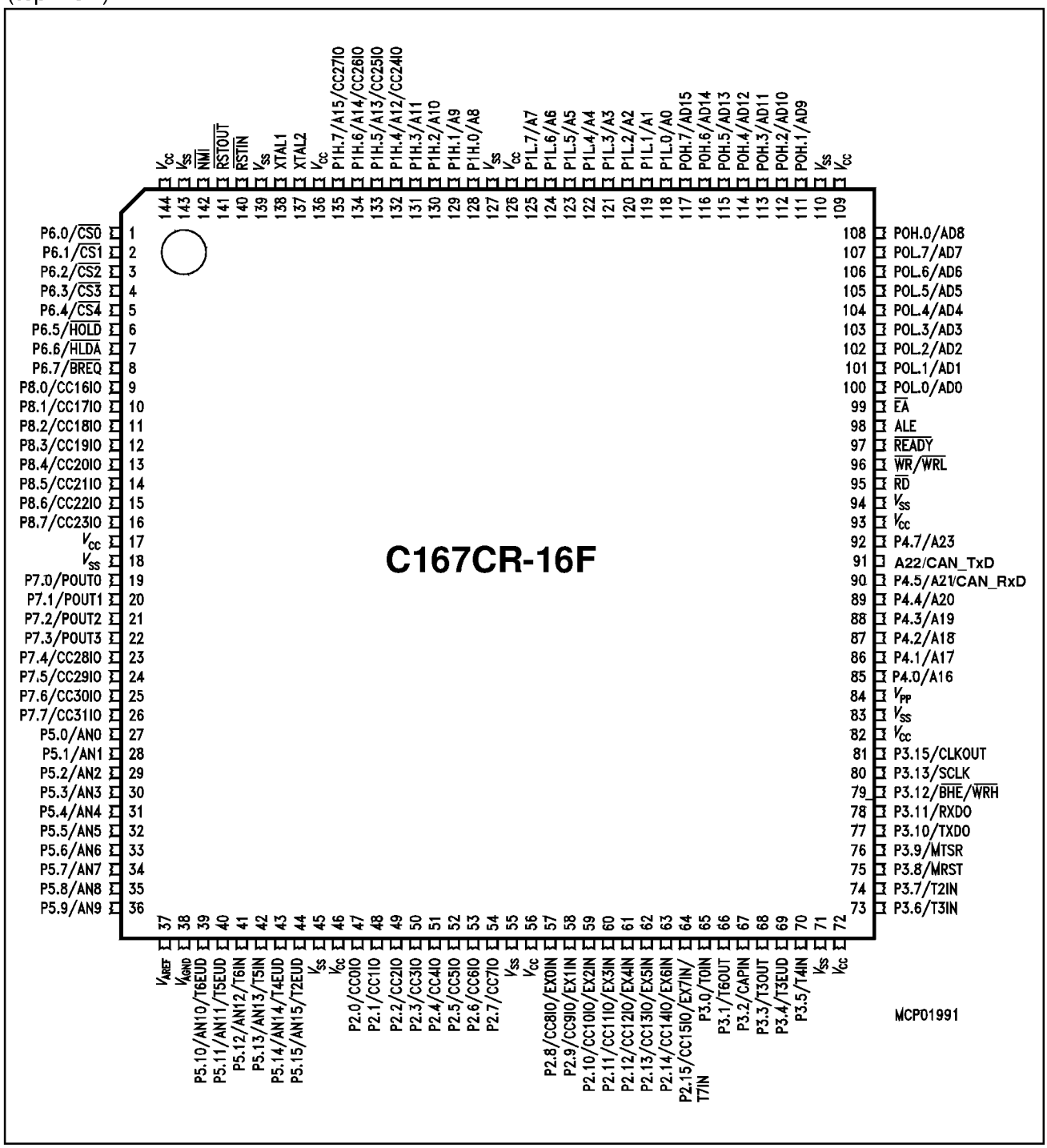



Figure 2

**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.<br>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).<br>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).<br>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<br>O | P3.12 | $\overline{BHE}$<br>$\overline{WRH}$ | Ext. Memory High Byte Enable Signal,<br>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<br>I | P4.5 | A21<br>CAN_RxD | Segment Address Line,<br>CAN Receive Data Input |
| | 91 | O<br>O | P4.6 | A22<br>CAN_TxD | Segment Address Line,<br>CAN Transmit Data Output |
| | 92 | O | P4.7 | A23 | Most Significant Segment Addr. Line |
| $\overline{RD}$ | 95 | O | External Memory Read Strobe. $\overline{RD}$ 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-16F to begin instruction execution out of external memory. A high level forces execution out of the internal Flash memory. |
| PORT0: P0L.0 – P0L.7, P0H.0 - P0H.7 | 100 – 107 108, 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:<br>P1L.0 –<br>P1L.7,<br>P1H.0 -<br>P1H.7 | <br>118 –<br>125<br>128 –<br>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: |
| | 132 | I | P1H.4 CC24IO CAPCOM2: CC24 Capture Input |
| | 133 | I | P1H.5 CC25IO CAPCOM2: CC25 Capture Input |
| | 134 | I | P1H.6 CC26IO CAPCOM2: CC26 Capture Input |
| | 135 | I | P1H.7 CC27IO CAPCOM2: CC27 Capture Input |
| XTAL1<br><br>XTAL2 | 138<br><br>137 | I<br><br>O | XTAL1: Input to the oscillator amplifier and input to the internal clock generator<br>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 | Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the C167CR-16F. An internal pullup resistor permits power-on reset using only a capacitor connected to $V_{SS}$. |
| $\overline{\text{RSTOUT}}$ | 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-16F 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 Analog/Digital converter. |
| $V_{AGND}$ | 38 | - | Reference ground for the Analog/Digital converter. |

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

| Symbol | Pin Number | Input (I) Output (O) | Function |
|---|---|---|---|
| $V_{PP}$ | 84 | - | Flash programming voltage.<br>This pin accepts the programming/erase voltage for the C167CR-16F ($V_{PP}$ = 12 V).<br>During normal operation (programming voltage $V_{PP}$ = 12 V not required) this pin must be connected to $V_{CC}$. |
| $V_{CC}$ | 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. |

**Note:** All $V_{SS}$ pins and all $V_{CC}$ pins must be connected to the system ground and the power supply, respectively.

## Functional Description

This document only describes specific properties of the C167CR-16F, eg. Flash memory functionality or specific DC and AC Characteristics, while for all other descriptions common for the C167CR-16F and the C167CR, eg. functional description, it refers to the respective Data Sheet for the Non-Flash device.

A detailled description of the C167CR-16F's instruction set can be found in the **"C16x Family Instruction Set Manual"**.

## Memory Organization

The memory space of the C167CR-16F 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.

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.

128 KBytes of on-chip Flash memory are provided to store user code or (read only) data. The Flash memory is divided into 4 blocks of different size which can be programmed/erased separately.

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.

### Flash Memory Overview

The C167CR-16F provides 128 KBytes of electrically erasable and reprogrammable non-volatile Flash EPROM on-chip for code or constant data.

A separate Flash Control Register (FCR) has been implemented to control Flash operations like programming or erasure. For programming or erasing an external 12 V programming voltage must be applied to the VPP pin. Flash programming and erasure is only possible during writing mode which is entered via a special key code sequence (Unlock sequence) to avoid unintended Flash operations.

The Flash memory is organized in blocks 32 bits wide which allows even double-word instructions to be fetched in just one machine cycle. The entire Flash memory is divided into four blocks with different sizes (48/48/24/8 KByte). This allows to erase each block separately, when only parts of the Flash memory need to be reprogrammed. Word or double word programming typically takes 100 µs, block erasing typically takes 1 s (@ 20 MHz CPU clock). The Flash memory features a typical endurance of 1000 erasing/programming cycles. Erased Flash memory cells contain all '1's, as known from standard EPROMs.

The Flash memory can be programmed both in an appropriate programming board and in the target system which provides a lot of flexibility. The C167CR-16F's on-chip bootstrap loader may be used to load and start the programming code. Any code that programs or erases Flash memory locations must be executed from memory outside the on-chip Flash memory itself (on-chip RAM or external memory).

To save the customer's know-how, a Flash memory protection option is provided in the C167CR-16F. If this was activated once, Flash memory contents cannot be read from any location outside the Flash memory itself (see section „Flash Protection").

The lower 32 KBytes of the on-chip Flash memory of the C167CR-16F can be mapped to either segment 0 ($00'0000_H$ to $00'7FFF_H$) or segment 1 ($01'0000_H$ to $01'7FFF_H$) during the initialization phase to allow external memory to be used for additional system flexibility. The upper 96 KBytes of the on-chip Flash memory are assigned to locations $01'8000_H$ to $02'FFFF_H$.

In standard mode (the normal operating mode) the Flash memory appears like the standard on-chip ROM of C167 devices with the same timing and functionality. Instruction fetches and data operand reads are performed with all addressing modes of the C16x instruction set.

In writing mode specific blocks of the on-chip Flash memory can be programmed (doublewords or words) or erased (banks). Writing mode is entered via a special key lock sequence and provides full access to the Flash Control Register (FCR). For programming as well as for erasing there are specific verify modes that allow to validate the previous operation in order to ensure secure Flash handling.

The following **terminology** is used in this document:

Flash **WRITING** means changing the state of the floating gate.
Flash **PROGRAMMING** means loading electrons onto the floating gate.
Flash **ERASING** means removing electrons from the floating gate.

Please refer to section „Fundamentals of Flash Technology".

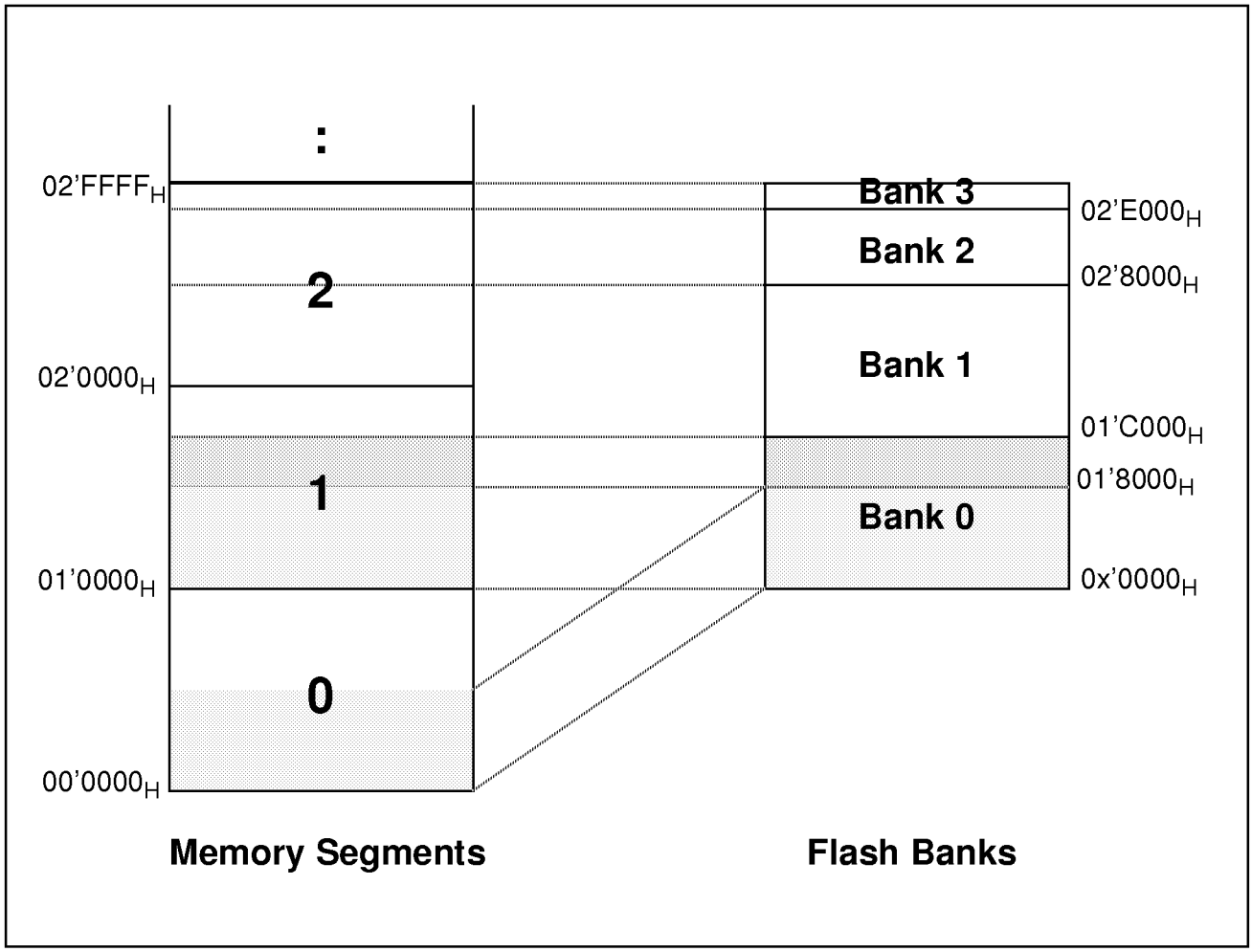



**Figure 3**
**Flash Memory Overview**

### Flash Memory Configuration

Upon reset the default memory configuration of the C167CR-16F is determined by the state of its $\overline{EA}$ pin. When $\overline{EA}$ is high the startup code is fetched from the on-chip Flash memory, when $\overline{EA}$ is low the internal ROM is disabled and the startup code is fetched from external memory.

In order to access the on-chip Flash memory after booting from external memory the internal ROM must be enabled via software by setting bit ROMEN in register SYSCON. The lower 32 KBytes of the Flash memory can be mapped to segment 0 or to segment 1, controlled by bit ROMS1 in register SYSCON. Mapping to segment 1 preserves the external memory containing the startup code, while mapping to segment 0 replaces the lower 32 KBytes of the external memory with on-chip Flash memory. In this case a valid vector table must be provided. As the on-chip Flash memory covers more than segment 0 segmentation should be enabled (by clearing bit SGTDIS in register SYSCON) in order to ensure correct stack handling when branching to the upper segments.

Whenever the internal memory configuration of the C167CR-16F is changed (enable, disable, mapping) the following procedure must be used to ensure correct operation:

- Configure the internal ROM as required
- Execute an inter-segment branch (JMPS, CALLS, RETS)
- Reload all four DPP registers

**Note:** Instructions that configure the internal ROM may only be executed from internal RAM or from external memory, **not** from the ROM itself.
Register SYSCON can only be modified **before** the execution of the EINIT instruction.

The C167CR-16F's Bootstrap Loader provides a mechanism to load the startup code and/or the Flash progamming routines from a remote code source via the serial interface without requiring additional external memory. This allows for firmware updates of the Flash memory for program and/ or data values.

### The Flash Control Register (FCR)

In standard operation mode the Flash memory can be accessed like the normal mask-programmable on-chip ROM of the C167CR. So all appropriate direct and indirect addressing modes can be used for reading the Flash memory.

All programming or erase operations of the Flash memory are controlled via the 16-bit Flash Control Register FCR. To prevent unintentional writing to the Flash memory the FCR is locked and inactive during standard operation mode. Before a valid access to the FCR is enabled, the Flash memory writing mode must be entered. This is done via a special key code instruction sequence.

**Note:** The FCR is no real register (SFR or GPR) but is rather virtually mapped into the active address space of the Flash memory. All even direct (mem) word accesses refer to the FCR (no byte- or bit-addressing), while all indirect ([$Rw_n$]) accesses refer to the Flash memory array itself. ROM mapping and DPP referencing must be considered for FCR accesses.

**FCR (*Even Flash Address*)** **Reset Value: $00X0_H$ *)**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FWM SET | - | - | - | - | - | BE | | WDW W | CKCTL | | VPP REV | FC VPP | FBUSY RPROT | FEE | FWE |
| rw | rw | rw | rw | rw | rw | rw | | rw | rw | | r | rw | r/w | rw | rw |

| Bit | Function |
|---|---|
| **FWE** | **Flash Write Enable Bit** (see description below)<br>0 : Flash write operations (program / erase) disabled<br>1 : Flash write operations (program / erase) enabled |
| **FEE** | **Flash Erase Enable Bit** (Significant only, when FWE='1', see description below)<br>0 : Flash programming mode selected<br>1 : Flash erase mode selected |
| **FBUSY** | **Flash Busy Bit** (On read accesses)<br>0 : No Flash write operation in progress<br>1 : Flash write operation in progress |
| **RPROT** | **Flash Read Protection Activation Bit** (On write accesses)<br>0 : Deactivates Flash read protection<br>1 : Activates Flash read protection, if this is enabled |
| **FCVPP** | **Flash Control $V_{PP}$ Bit**<br>0 : No $V_{PP}$ failure occurred during a Flash write operation<br>1 : $V_{PP}$ failure occurred during a Flash write operation |
| **VPPREV** | **Flash $V_{PP}$ Revelation Bit**<br>0 : No valid $V_{PP}$ applied to pin $V_{PP}$<br>1 : $V_{PP}$ applied to pin $V_{PP}$ is valid |
| **CKCTL** | **Internal Flash Timer Clock Control**<br>Determines the width of an internal Flash write or erase pulse |
| **WDWW** | **Word / Double Word Writing Bit** (significant only in programming mode)<br>0 : 16-bit programming operation<br>1 : 32-bit programming operation |
| **BE** | **Bank Erase Select** (significant only in erasing mode)<br>Selects the Flash Bank to be erased |
| **FWMSET** | **Flash Writing Mode Set Bit** (see description below)<br>0 : Exit Flash writing mode, return to standard mode<br>1 : Stay in Flash writing mode |

*) The reset value of bit VPPREV depends on the voltage on pin $V_{PP}$.

**The selection of Flash Operation and Read Mode** is done via the three bits FWE, FEE and FWMSET. The table below shows the combinations for these bits to select a specific function:

| FWMSET | FEE | FWE | Flash Operation Mode | Flash Read Mode |
|---|---|---|---|---|
| 1 | 1 | 1 | Erasing mode | Erase-Verify-Read via [Rn] |
| 1 | 0 | 1 | Programming mode | Program-Verify-Read via [Rn] |
| 1 | X | 0 | Non-Verify mode | Normal Read via [Rn] |
| 0 | X | X | Standard mode | Normal Read via [Rn] or mem |

FWE enables/disables write operations, FEE selects erasing or programming, FWMSET indicates writing mode and is set automatically once the writing mode is entered. Bits FWE and FEE select an operation, but do not directly execute this operation.

**Note:** Watch the FWMSET bit when writing to register FCR (word access only) in order not to exit Flash writing mode unintentionally by clearing bit FWMSET.

**FBUSY:** This **read-only flag** is set to '1' while a Flash programming or erasing operation is in progress. FBUSY is set via hardware when the respective program/erase command is issued.

**RPROT:** This **write-only** Flash **Read Protection** bit determines whether Flash protection is active or inactive. RPROT is the only FCR bit which can be modified even in the Flash standard mode but only by an instruction executed from the on-chip Flash memory itself. Per reset, RPROT is set to '1'.

**Note:** RPROT is only significant if the general Flash memory protection is enabled.

**FCVPP and VPPREV:** These **read-only** bits allow to monitor the $V_{PP}$ voltage. The Flash **Vpp Revelation** bit VPPREV reflects the state of the $V_{PP}$ voltage in the Flash writing mode (VPPREV = '0' indicates that $V_{PP}$ is below the threshold value necessary for reliable programming or erasure, otherwise VPPREV = '1'). The **Flash Control $V_{PP}$** bit FCVPP indicates if $V_{PP}$ fell below the valid threshold value during a Flash programming or erase operation (FCVPP = '1') and the operation therefore might not have been executed properly. FCVPP = '0' after such an operation indicates that no critical discontinuity on $V_{PP}$ has occurred.

**CKCTL:** This **Flash Timer Clock Control** bitfield controls the width of the programming or erase pulses (TPRG) applied to Flash memory cells during the corresponding operation. The width of a single programming or erase pulse and the cumulated programming or erase time must not exceed certain values to avoid putting the Flash memory under critical stress (see table below). The pulse width and also the maximum number on programming or erase attempts allowed depends on the CPU clock frequency.

| Time Specification | Limit Value | |
|---|---|---|
| Maximum Programming Pulse Width | 200 | µs |
| Maximum Cumulated Programming Time | 2.5 | ms |
| Maximum Erase Pulse Width | 20 | ms |
| Maximum Cumulated Erase Time | 30 | s |

In order not to exceed the limit values listed above, a specific CKCTL setting requires a minimum CPU clock frequency, as listed below.

| Setting of CKCTL | Length of TPRG | TPRG @ $f_{CPU}$ = 20 MHz | $f_{CPUmin}$ for programming | $f_{CPUmin}$ for erasing |
|---|---|---|---|---|
| 0 0 | $2^8 * 1/f_{CPU}$ | 12.8 µs | 1.28 MHz | ( 12.8 KHz ) [1] |
| 0 1 | $2^{11} * 1/f_{CPU}$ | 102.4 µs | 10.24 MHz | ( 102.4 KHz ) [1] |
| 1 0 | $2^{15} * 1/f_{CPU}$ | 1.64 ms | --- | 1.64 MHz |
| 1 1 | $2^{18} * 1/f_{CPU}$ | 13.11 ms | --- | 13.11 MHz |

[1] Please note that these are computed values. Actual values must respect the operational range specified for the C167CR-16F.

The maximum number of allowed programming or erase attempts depends on the CPU clock frequency and on the CKCTL setting chosen in turn. This number results from the actual pulse width compared to the maximum pulse width (see above tables).
The table below lists some sample frequencies, the respective recommended CKCTL setting and the resulting maximum number of program / erase pulses:

| $f_{CPU}$ | Programming | | | Erasing | | |
|---|---|---|---|---|---|---|
| | CKCTL | TPROG | $N_{PROGmax}$ | CKCTL | TPROG | $N_{ERASEmax}$ |
| 1 MHz | 0 0 | 128 µs | 19 | 0 1 | 2.05 ms | 14648 |
| 10 MHz | 0 0 | 12.8 µs | 195 | 1 0 | 3.28 ms | 9155 |
| 16 MHz | 0 0 | 8 µs | 312 | 1 0 | 2.05 ms | 14648 |
| 20 MHz | 0 0 | 6.4 µs | 390 | 1 0 | 1.64 ms | 18310 |

**BE:** The Flash **Bank Erasing** bit field determines the Flash memory bank to be erased (see table below). The physical addresses of the lower 32 KBytes of bank 0 depend on the Flash memory mapping chosen.

| BE setting | Bank | Size | Addresses Selected for Erasure (x = 0 or 1) |
|---|---|---|---|
| 0 0 | 0 | 48 KB | 0x'$0000_H$ to 0x'$7FFF_H$ / 01'$8000_H$ to 01'$BFFF_H$ |
| 0 1 | 1 | 48 KB | 01'$C000_H$ to 01'$FFFF_H$ / 02'$0000_H$ to 02'$7FFF_H$ |
| 1 0 | 2 | 24 KB | 02'$8000_H$ to 02'$DFFF_H$ |
| 1 1 | 3 | 8 KB | 02'$E000_H$ to 02'$FFFF_H$ |

### Operation Modes of the Flash Memory

There are two basic operation modes for Flash accesses: The standard and the writing mode. Sub-modes of the writing mode are the programming, the erase and the non-verify mode.

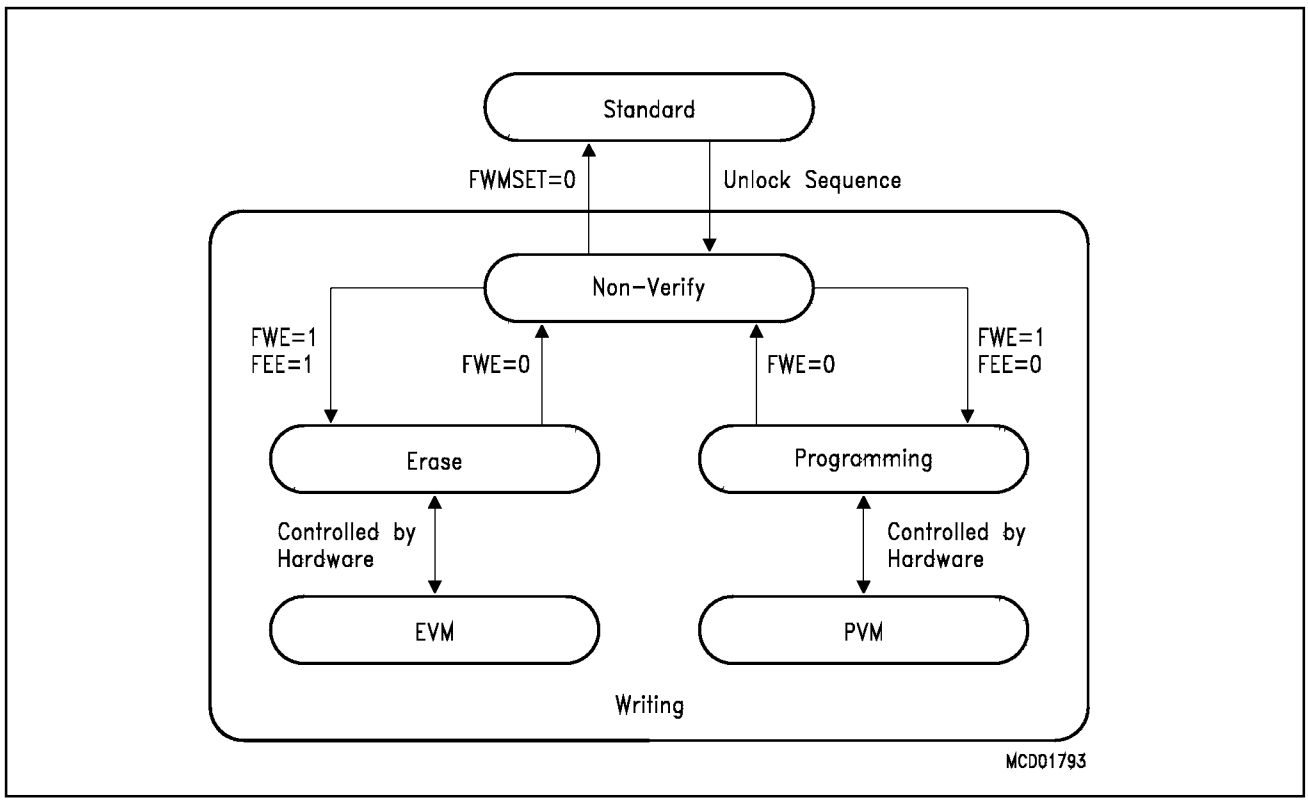



**Figure 4**
**Flash Operating Mode Transitions**

**In Standard Mode** the Flash memory can be accessed from any memory location (external memory, on-chip RAM or Flash memory) for instruction fetches and data operand reads. Data operand reads may use both direct 16-bit (mnemonic: mem) and indirect (mnemonic: [Rw]) addressing modes. Standard mode does not allow Flash write operations or accesses to the FCR except for the protection activation bit RPROT.

**Note:** When Flash protection is active, data operands can be accessed only by instructions that are executed out of the internal Flash memory and branches to the Flash memory from locations outside are inhibited.

**The Flash Writing Modes** must be entered for programming or erasing the Flash memory. The C167CR-16F enters these modes by a specific key code sequence, called UNLOCK sequence.
In writing mode the used addressing mode decides whether the FCR or a Flash memory location is accessed. The FCR can be accessed with any direct access to an even address in the active address space of the Flash memory. Only word operand instructions are allowed for FCR accesses. Accesses to Flash memory locations must use indirect addressing to even addresses.

| | | | |
|---|---|---|---|
| **direct 16-bit addressing mode:** | **mem** | **-->** | **Access to FCR** |
| **indirect addressing mode:** | **[$Rw_n$]** | **-->** | **Access to Flash location** |

After entering writing mode the first erase or programming operation must not be started for at least 10 µs. This absolute (!) delay time is required to set up the internal high voltage. In general, Flash write operations need a 12 V external $V_{PP}$ voltage to be applied to the $V_{PP}$ pin.

It is not possible to erase or to program the Flash memory via code executed from the Flash memory itself. The respective code must reside within the on-chip RAM or within external memory.

When programming or erasing 'on-line' in the target system, some considerations have to be taken: While these operations are in progress, the Flash memory cannot be accessed as usual. Therefore care must be taken that no branch is taken into the Flash memory and that no data reads are attempted from the Flash memory during programming or erasure. If the Flash memory is mapped to segment 0, it must especially be ensured that no interrupt or hardware trap can occur, because this would implicitly mean such a 'forbidden' branch to the Flash memory. Correct Flash operation is not guaranteed in this case.

**The UNLOCK sequence** is a specific key code sequence, which is required to enable the writing modes of the C167CR-16F. The UNLOCK sequence must use identical values (see example below) and the two accesses must not be interrupted:

```
MOV     FCR, Rwn                ;Dummy write to the FCR
MOV     [Rwn], Rwn              ;Both operands use the same GPR
CALL    cc_UC, WAIT_10          ;Delay for 10 µs (may be realized also by
                                ;instructions other than a delay loop
```

where $Rw_n$ can be any word GPR (R0…R15). [$Rw_n$] and FCR must point to even addresses within the active address space of the Flash memory.

**Note:** Data paging and Flash segment mapping, if active, must be considered in this context.

**In Flash Erase Mode** (FEE='1', FWE='1') the C167CR-16F is prepared to erase the bank selected by the Bank Erase (BE) bit field in the FCR. The width of the erase pulses generated internally is defined by the Internal Flash Timer Clock Control (CKCTL) bit field of the FCR. The maximum number of erase pulses ($EN_{max}$) applied to the Flash memory is determined by software in the Flash erase algorithm. The chosen values for CKCTL and $EN_{max}$ must guarantee a maximum cumulated erase time of 30 s per bank and a maximum erase pulse width of 10 ms.
The Flash bank erase operation will not start before the erase command is given. This provides additional security for the erase operation. The erase command can be any write operation to a Flash location, where the data and the even address written to must be identical:

```
MOV     [Rwn], Rwn              ; Both operands use the same GPR
```

Upon the execution of this instruction, the Flash Busy (FBUSY) flag is automatically set to '1' indicating the start of the operation. End of erasure can be detected by polling the FBUSY flag. $V_{PP}$ must stay within the valid margins during the entire erase process.

At the end of erasure the Erase-Verify-Mode **(EVM)** is entered automatically. This mode allows to check the effect of the erase operation (see description below).

**Note:** Before the erase algorithm can be properly executed, the respective bank of the Flash memory must be programmed to all zeros ('$0000_H$').

**In Flash Programming Mode** (FEE='0', FWE='1') the C167CR-16F is prepared to program Flash locations in the way specified by the Word or Double Word Write (WDWW) bit in the FCR. The width of the programming pulses generated internally is defined by the Internal Flash Timer Clock Control (CKCTL) bit field of the FCR. The maximum number of programming pulses ($PN_{max}$) applied to the Flash memory is determined by software in the Flash programming algorithm. The chosen values for CKCTL and $PN_{max}$ must guarantee a maximum cumulated programming time of 2.5 ms per cell and a maximum programming pulse width of 128 µs.

If 16-bit programming was selected, the operation will start automatically when a write instruction is executed, where the first operand specifies the address and the second operand the value to be programmed:

```
MOV     [Rwn], Rwm                ;Program one word
```

If 32-bit programming was selected, the operation will start automatically when the second of two subsequent write instructions is executed, which define the doubleword to be programmed.
Note that the destination pointers of both instructions refer to the same even double word address. The two instructions must be executed without any interruption.

```
MOV     [Rwn], Rwx                ;Prepare programming of first word
MOV     [Rwn], Rwy                ;Start programming of both words
```

Upon the execution of the second instruction (or the one and only in 16-bit programming mode), the Flash Busy (FBUSY) bit is automatically set to '1'. End of programming can be detected by polling the FBUSY bit. $V_{PP}$ must stay within the valid margins during the entire programming process.

At the end of programming the Program-Verify-Mode **(PVM)** is entered automatically. This mode allows to check the effect of the erase operation (see description below).

**The Flash Verify-Modes** Erase-Verify-Mode (**EVM**) and Program-Verify-Mode (**PVM**) allow to verify the effect of an erase or programming operation. In these modes an internally generated margin voltage is applied to a Flash cell, which makes reading more critical than for standard read accesses. This ensures safe standard accesses after correct verification.
To get the contents of a Flash word in this mode, it has to be read in a particular way:

```
MOV     Rwm, [Rwn]                ;First (invalid) read of dedicated cell
...                               ;4 µs delay to stabilize...
                                  ;...the internal margin voltage
MOV     Rwm, [Rwn]                ;Second (valid) read of dedicated cell
```

Such a Flash verify read operation is different from the reading in the standard or in the non-verify mode. Correct verify reading needs a read operation performed twice on the same cell with an absolute time delay of 4 µs which is needed to stabilize the internal margin voltage applied to the cell. To verify that a Flash cell was erased or programmed properly, the value of the second verify read operation has to be compared against $FFFF_H$ or the target value, respectively.

Clearing bit FWE to '0' exits the Flash verify modes and returns to the Flash non-verify mode.

**In Flash non-verify mode** all Flash locations can be read as usual (via indirect addressing modes), which is not possible in Flash programming or Flash erase mode (see EVM and PVM).

### Flash Protection

If active, Flash protection prevents data operand accesses and program branches into the on-chip Flash area from any location outside the Flash memory itself. Data operand accesses and branches to Flash locations are exclusively allowed for instructions executed from the Flash memory itself. Erasing and programming of the Flash memory is not possible while Flash protection is active.

**Note:** A program running within the Flash memory may of course access any location outside the Flash memory and even branch to a location outside.
However, there is no way back, if Flash protection is active.

Flash protection is controlled by two different bits:
- The user-accessible write-only Protection Activation bit (RPROT) in register FCR and
- The one-time-programmable Protection Enable bit (UPROG).

Bit UPROG is a 'hidden' one-time-programmable bit only accessible in a special mode, which can be entered eg. via a Flash EPROM programming board. Once programmed to '1', this bit is unerasable, ie. it is not affected by the Flash Erase mechanism.

**To activate Flash Protection** bit UPROG must have been programmed to '1', and bit RPROT in register FCR must be set to '1'. Both bits must be '1' to activate Flash protection.

**To deactivate Flash Protection** bit RPROT in register FCR must be cleared to '0'. If any of the two bits (UPROG or RPROT) is '0', Flash protection is deactivated.
Generally Flash protection will remain active all the time. If it has to be deactivated intermittently, eg. to call an external routine or to reprogram the Flash memory, bit RPROT must be cleared to '0'.

**To access bit RPROT** in register FCR, an instruction with a 'mem, reg' addressing mode must be used, where the first operand has to represent the FCR address (any even address within the active address space of the Flash memory) and the second operand must refer to a value which sets the RPROT bit to '0', eg.:

```
MOV    FCR, ZEROS                ;Deactivate Flash Protection
```

RPROT is the only bit in the FCR which can be accessed in Flash standard mode without having to enter the Flash writing mode. Other bits in the FCR are not affected by such a write operation. However, this access requires an instruction executed out of the internal Flash memory itself.

**After reset** bit RPROT is set to '1'. For devices with protection disabled (UPROG='0') this has no effect. For devices with protection enabled this ensures that program execution starts with Flash protection active from the beginning.

**Note:** In order to maintain uninterrupted Flash protection, be sure not to clear bit RPROT unintentionally by FCR write operations. Otherwise the Flash protection is deactivated.

### Flash Programming Algorithm

The figure below shows the recommended Flash programming algorithm. The following example describes this algorithm in detail.

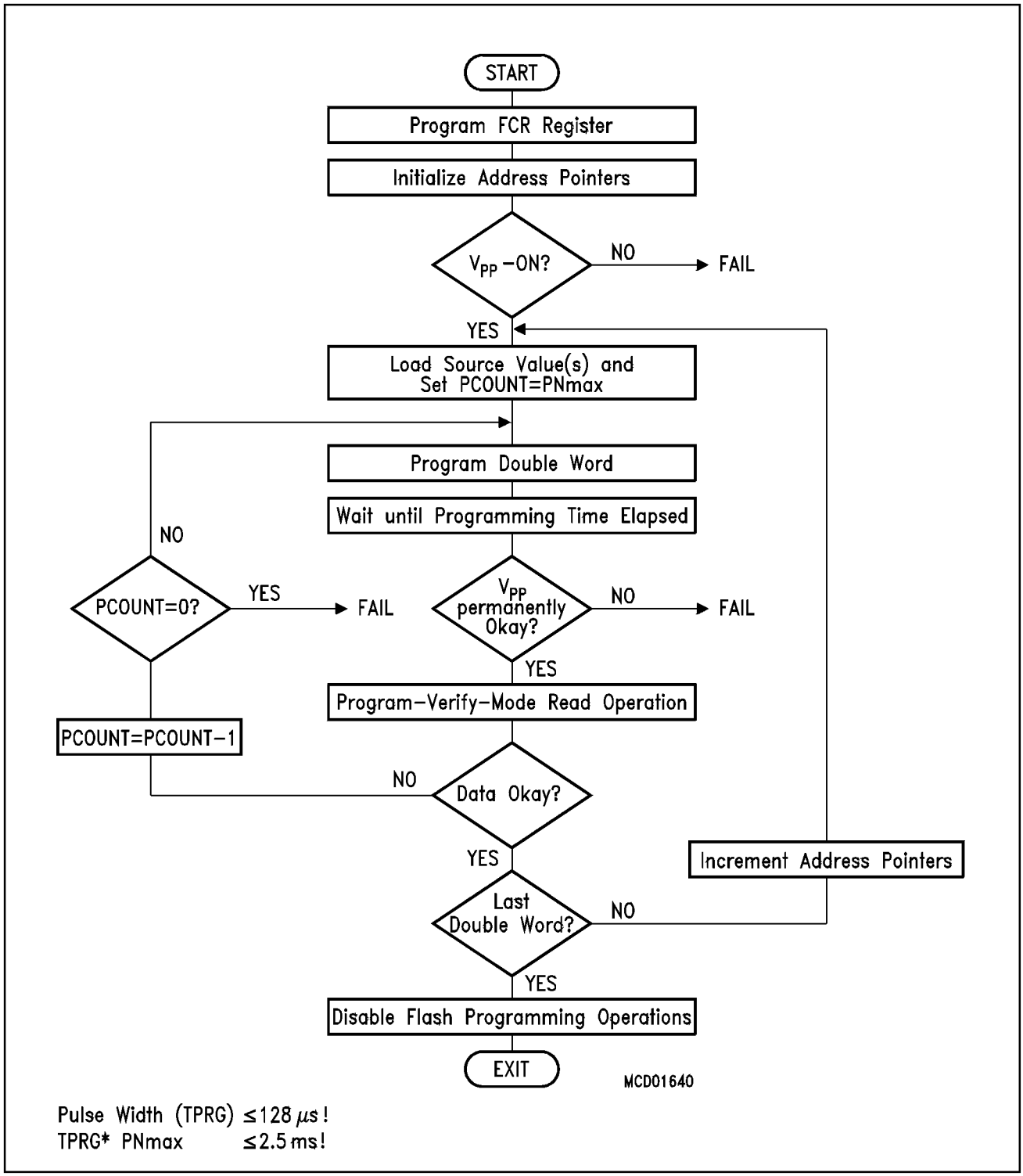



**Figure 5**
**Flash Programming Algorithm**

### Flash Programming Example

This example describes the Flash programming algorithm. A source block of code and/or data within the first 32 KBytes of segment 0 is copied (programmed) to a target block within the Flash memory, which is mapped to segment 1 in this case. The start and the end address of the source block to be copied are specified by the parameters SRC_START and SRC_END respectively. The target Flash memory block begins at location FLASH_START. This example uses 32-bit Flash programming.

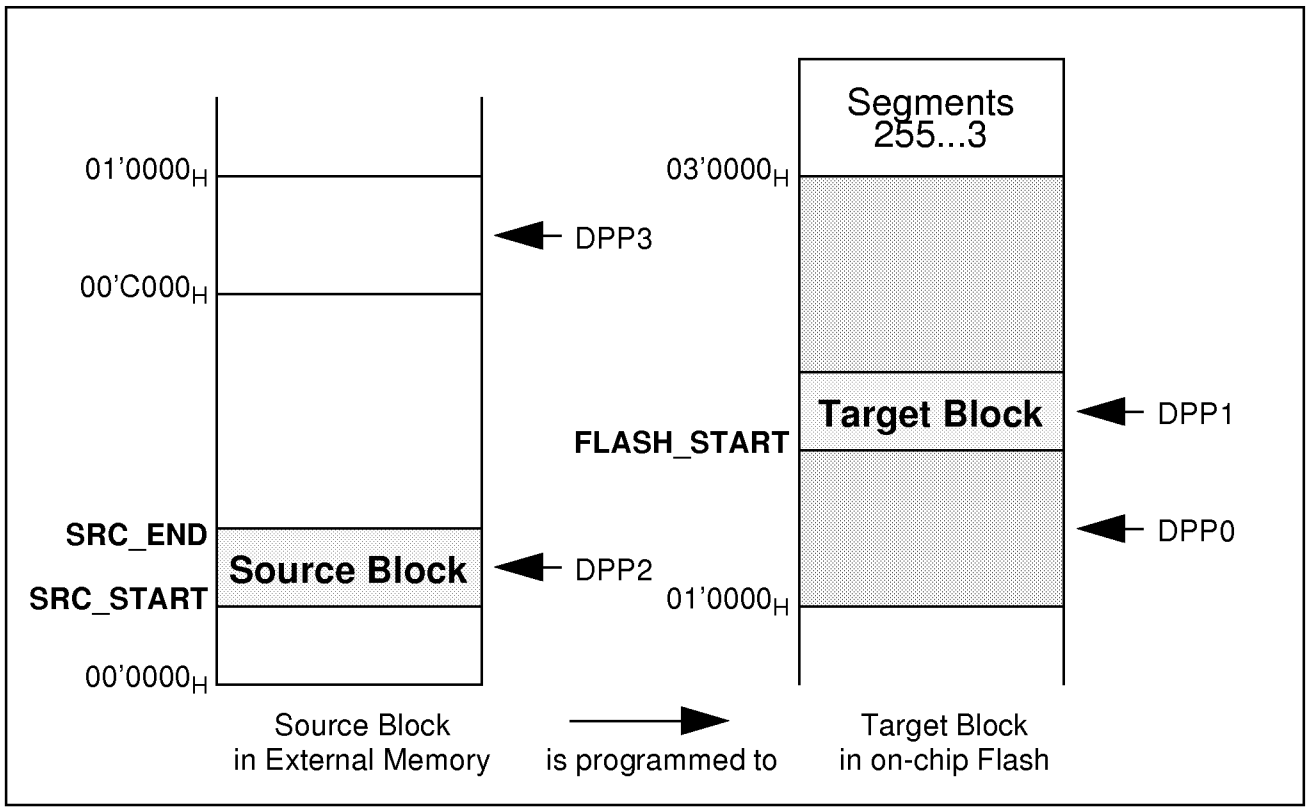



**Figure 6**
**Memory Allocation for Flash Programming Example**

**Note:** This example represents one possibility how to program the Flash memory. Other solutions may differ in the way they provide the source data (eg. without external memory), but use the same Flash programming algorithm.

The FCR has been defined with an EQU assembler directive. Accesses to bits of the FCR are made via an auxiliary GPR, as the FCR itself is not bit-addressable.

The shown example uses the following assumptions:

- Pin $V_{PP}$ receives a proper $V_{PP}$ supply voltage.
- The C167CR-16F runs at 20 MHz CPU clock (absolute time delays refer to this).
- The Flash memory is mapped to segment 1. All DPPs are set correctly.

• **Enter writing mode via unlock sequence** (prerequisite for any programming or erase operation).

```
MOV     FCR, Rwn                ;Dummy write to the FCR
MOV     [Rwn], Rwn              ;Both operands use the same GPR
CALL    cc_UC, WAIT_10          ;Delay for 10 µs
```

• **Program the FCR register** with a value that selects the desired operating mode. Note that this does not yet start the programming operation itself.

```
MOV     R15, #1000 0000 1010 0001B
        ;    #xxxx xxxx xxxx xxx1:FWE='1': Enable Flash write operations
        ;    #xxxx xxxx xxxx xx0x:FEE='0': Select programming mode
        ;    #xxxx xxxx x01x xxxx:CKCTL='01':102.4 µs prog.pulse(@20MHz)
        ;    #xxxx xxxx 1xxx xxxx:WDWW='1':Select 32-bit progr. mode
        ;    #1xxx xxxx xxxx xxxx:FWMSET='1':Stay in writing mode
MOV     DPP1:pof FCR, R15       ;Write Value to FCR using 16-bit access
```

• **Initialize pointers and counter** for the first transfer of the programming algorithm.
The source data block is accessed via the pointer SRC_PTR, initialized with SRC_START. All read operations via SRC_PTR use DPP2, which selects data page 1 in this example.
The Flash memory must be accessed indirectly and uses the pointer FLASH_PTR, initialized with FLASH_START.
The counter DWCOUNT defines the number of doublewords to be programmed.

• **Test for correct $V_{PP}$ margin at pin $V_{PP}$** before a programming operation is started. If bit VPPREV reads '1', the programming voltage is correct and the algorithm can be continued. Otherwise, the programming routine could wait in Flash writing mode until $V_{PP}$ reaches its correct value and resume programming then, or it could exit writing mode.

```
MOV     R15, DPP1:pof FCR       ;Read FCR contents using 16-bit access
JB      R15.4, Vpp_OK1          ;Test VPP via bit VPPREV (= FCR.4)
...                             ;VPPREV='0': Exit programming procedure
Vpp_OK1:                        ;VPPREV='1': Test Okay! Continue
```

• **Load source values and initialize loop counter** (PCOUNT) with the maximum number of programming trials (PNmax) to be performed before exiting the routine with a failure. Each trial means applying a pulse of 102.4 µs to the selected words in the Flash memory. According to the maximum cumulated programming time of 2.5 ms allowed per cell, PNmax must be '25' here.
The doubleword at memory location [SRC_PTR] is loaded into two auxiliary registers DATAWR1 and DATAWR2.

• **Program one doubleword** stored in the auxiliary data registers to the Flash memory location [FLASH_PTR]. FLASH_PTR is not incremented here, since in 32-bit programming mode the hardware automatically arranges the two data words correctly. The execution of the second write instruction automatically starts the programming of the entire double word.
This instruction sequence must not be interrupted.

```
MOV     [FLASH_PTR], DATAWR1      ;Write low word to Flash
MOV     [FLASH_PTR], DATAWR2      ;Write high word to Flash, starts prog.
```

• **Wait until programming time elapsed** (102.4 µs in this example), which depends on bit field CKCTL in the FCR register and on the CPU clock frequency. End of programming is detected by polling the FBUSY flag in the FCR register. The Flash memory switches to PVM mode automatically.

```
WAIT_PROG:                        ;Polling Loop to check bit FBUSY
MOV     R15, DPP1: pof FCR        ;Read FCR contents using 16-bit access
JB      R15.2, WAIT_PROG          ;Loop while bit FBUSY (FCR.2) is '1'
...                               ;Continue in PVM mode, when FBUSY is '0'
```

• **Verify $V_{PP}$ validity during programming** to make sure $V_{PP}$ did not exceed its valid margins during the programming operation. Otherwise programming may have not been performed properly. The FCVPP flag is set to '1' in case of this error condition. If FCVPP reads '1', the programming routine can abort, when $V_{PP}$ still fails, or repeat the programming operation, when $V_{PP}$ proves to be stable now.

• **Perform Program-Verify operation and compare with source data** in order to check whether a programming operation was performed correctly. PVM reading consists of two identical Flash read instructions with 4 µs delay in between. This example uses CMP instructions to access the Flash memory. In case of a mismatch the programming routine repeats the programming cycle provided that the maximum number of attempts was not yet reached. PVM reading and data comparison must be performed on both words of the double word to be tested.

```
CMP     DATAWR1, [FLASH_PTR]      ;1st step of PVM read (low word)
CALL    cc_UC, WAIT_4             ;Delay for 4 µs
CMP     DATAWR1, [FLASH_PTR]      ;2nd step of PVM read (low word)
JMP     cc_NZ, PROG_FAILED        ;Reprogram on mismatch if (PCOUNT)>0
MOV     R15, FLASH_PTR
ADD     R15, #0002H               ;Aux. pointer to upper word of doubleword
CMP     DATAWR2, [R15]            ;1st step of PVM read (high word)
CALL    cc_UC, WAIT_4             ;Delay for 4 µs
CMP     DATAWR2, [R15]            ;2nd step of PVM read (high word)
JMP     cc_NZ, PROG_FAILED        ;Reprogram on mismatch if (PCOUNT)>0
...                               ;Programming OK. Go on with next step.
```

• **Check number of programming attempts** to decide, if another programming attempt is allowed. PCOUNT is decremented by '1' upon each unsuccessful programming attempt. If it expires, the failing Flash cells are classified as unprogrammable and should be left out. This failure is very unlikely to occur. However, it should be checked for safe programming.

**Note:** This step is taken only in case of a program verify mismatch.

• **Check for last doubleword and increment pointers** to decide, if another programming cycle is required. The auxiliary counter DWCOUNT is decremented by '1' after each successful double word programming. If it expires, the complete data block is programmed and the programming routine is exited successfully. Otherwise source and target pointers (SRC_PTR and FLASH_PTR) are incremented to the next doubleword to be programmed.

• **Disable Flash programming operations and exit routine**, when the Flash memory block was programmed successfully or when a failure occurred. In either case bit FWE of the FCR is reset to '0' and the programming routine is exited. This means that the Flash non-verify mode is entered again, where the FCR stays accessible but Flash memory locations can be read normally again using indirect addressing. For returning to the Flash standard mode, bit FWMSET of the FCR must be reset to '0' by the calling routine. The programming routine may return an exit code that indicates correct programming or identifies the type of error.

### Flash Erase Algorithm

The figure below shows the recommended Flash erase algorithm. The following example describes this algorithm in detail.

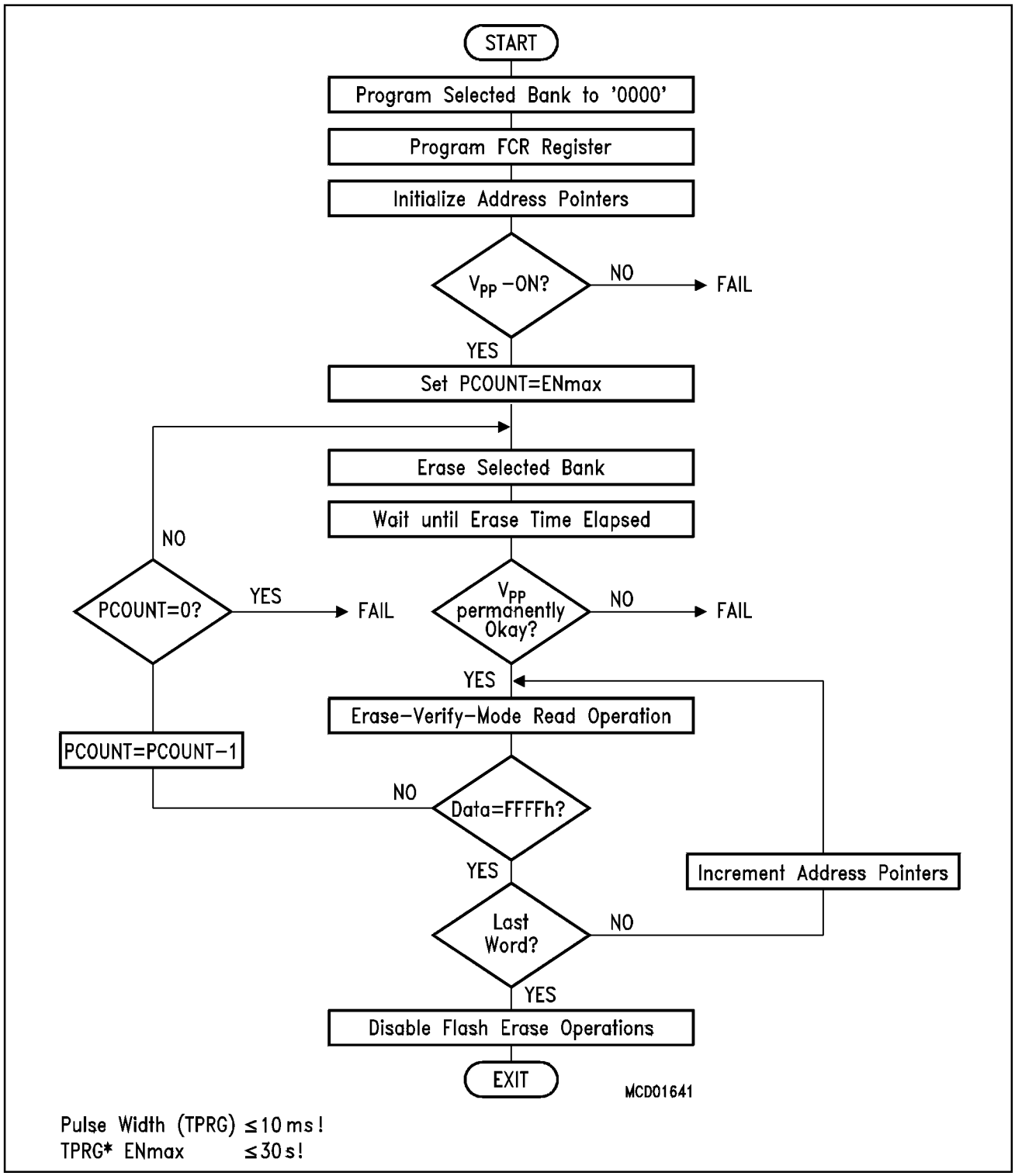



**Figure 7**
**Flash Erase Algorithm**

### Flash Erase Example

This example describes the Flash erase algorithm. The four banks of the Flash memory can be erased separately. The algorithm erases the Flash memory bank, which is selected by bitfield BE in the FCR. Start address and size of the selected Flash bank have to be considered.

**Note:** Before a bank can be erased, all its contents must be programmed to '$0000_H$'. This is required by the physics of the Flash memory cells and is done with the Flash programming algorithm already described.

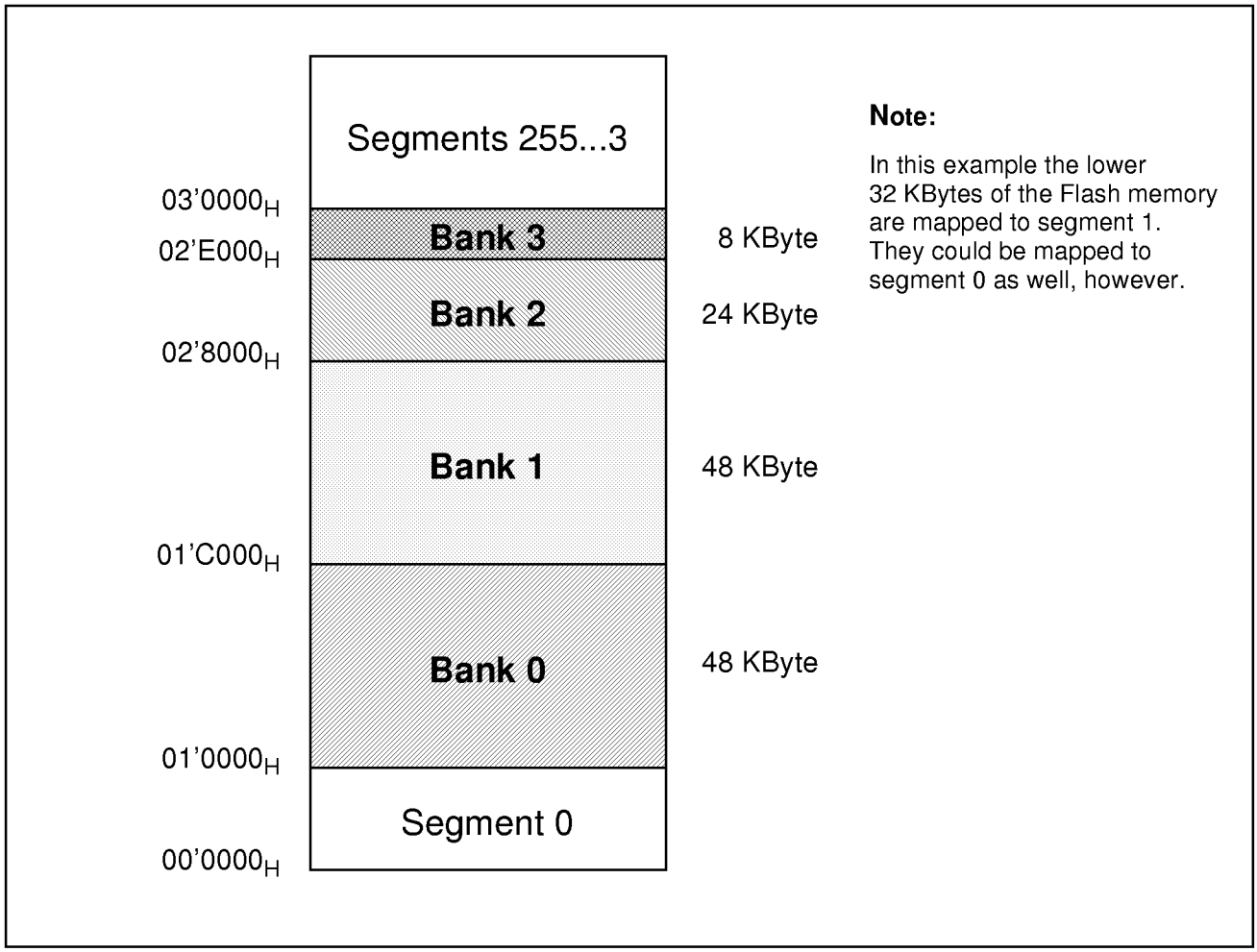



**Figure 8**
**Memory Banking for Flash Erasure**

The FCR has been defined with an EQU assembler directive. Accesses to bits of the FCR are made via an auxiliary GPR, as the FCR itself is not bit-addressable.

The shown example uses the following assumptions:

- Pin $V_{PP}$ receives a proper $V_{PP}$ supply voltage.
- The C167CR-16F runs at 20 MHz CPU clock (absolute time delays refer to this).
- The Flash memory is mapped to segment 1. All DPPs are set correctly.

• **Enter writing mode via unlock sequence** (prerequisite for any programming or erase operation).

```
MOV     FCR, Rwn                ;Dummy write to the FCR
MOV     [Rwn], Rwn              ;Both operands use the same GPR
CALL    cc_UC, WAIT_10          ;Delay for 10 µs
```

• **Program the FCR register** with a value that selects erase mode. Note that this does not yet start the erase operation itself.

```
MOV     R15, #1000 00XX 0100 0011B
        ;    #xxxx xxxx xxxx xxx1:FWE='1': Enable Flash write operations
        ;    #xxxx xxxx xxxx xx1x:FEE='1': Select erase mode
        ;    #xxxx xxxx x10x xxxx:CKCTL='10':1.64 ms erase pulse(@20MHz)
        ;    #xxxx xxXX xxxx xxxx:BE='xx': Select the desired bank(3..0)
        ;    #1xxx xxxx xxxx xxxx:FWMSET='1':Stay in writing mode
MOV     DPP1:pof FCR, R15       ;Write Value to FCR using 16-bit access
```

• **Initialize target pointer** with the start address of the selected Flash memory bank. The Flash memory must be accessed indirectly and uses the pointer FLASH_PTR. This pointer will apply to DPP0 or DPP1, which are expected to select data pages 4 or 5, respectively.

• **Test for correct $V_{PP}$ margin at pin $V_{PP}$** before an erase operation is started. If bit VPPREV reads '1', the erase voltage is correct and the algorithm can be continued. Otherwise, the erase routine could wait in Flash writing mode until $V_{PP}$ reaches its correct value and resume erasing then, or it could exit writing mode.

```
MOV     R15, DPP1:pof FCR       ;Read FCR contents using 16-bit access
JB      R15.4, Vpp_OK2          ;Test VPP via bit VPPREV (= FCR.4)
...                             ;VPPREV='0': Exit erase procedure
Vpp_OK2:                        ;VPPREV='1': Test Okay! Continue
```

• **Initialize loop counter** (PCOUNT) with the maximum number of erase trials (ENmax) to be performed before exiting the routine with a failure. Each trial means applying a pulse of 10 ms to the selected Flash memory bank. According to the maximum cumulated erase time of 30 s allowed per cell, ENmax must be '3000' here.

• **Erase selected Flash memory bank** by writing to a Flash memory location using the target address as write data.

```
MOV     [FLASH_PTR], FLASH_PTR  ;Write address to Flash, starts erasing
```

• **Wait until erase time elapsed**, which depends on bit field CKCTL in the FCR register and on the CPU clock frequency (10 ms in this example). End of erasing is detected by polling the FBUSY flag in the FCR register. The Flash memory switches to EVM mode automatically.

```
WAIT_ERASE:                     ;Polling Loop to check bit FBUSY
MOV     R15, DPP1: pof FCR      ;Read FCR contents using 16-bit access
JB      R15.2, WAIT_ERASE       ;Loop while bit FBUSY (FCR.2) is '1'
...                             ;Continue in EVM mode, when FBUSY is '0'
```

• **Verify $V_{PP}$ validity during erasing** to make sure $V_{PP}$ did not exceed its valid margins during the erase operation. Otherwise erasing may have not been performed properly. The FCVPP flag is set to '1' in case of this error condition. If FCVPP reads '1', the erase routine can abort, when $V_{PP}$ still fails, or repeat the erase operation, when $V_{PP}$ proves to be stable now.

• **Perform Erase-Verify operation and compare with 'FFFF$_H$'** in order to check whether an erase operation was performed correctly. EVM reading consists of two identical Flash read instructions with 4 µs delay in between. This example uses CMP instructions to access the Flash memory. In case of a mismatch the erase routine repeats the erase cycle provided that the maximum number of attempts was not yet reached.

```
MOV     R15, ONES                ;Load auxil. GPR with anticipated value
CMP     R15, [FLASH_PTR]         ;1st step of EVM read
CALL    cc_UC, WAIT_4            ;Delay for 4 µs
CMP     R15, [FLASH_PTR]         ;2nd step of EVM read
JMP     cc_NZ, ERASE_FAILED      ;Re-erase on mismatch if (PCOUNT)>0
...                              ;Erasing was OK. Go on with next step.
```

• **Check number of erase attempts** to decide, if another erase attempt is allowed. PCOUNT is decremented by '1' upon each unsuccessful erase attempt. If it expires, the failing Flash memory bank is classified as unerasable. This failure is very unlikely to occur. However, it should be checked for safe erasing.

**Note:** This step is taken only in case of a erase verify mismatch.

• **Check for last word and increment pointers** to decide, if another cell must be verified. The target pointer (FLASH_PTR) is incremented to the next word to be verified and checked against the upper limit of the respective bank. If the target pointer exceeds the bank limit, the erase routine is exited successfully.

• **Disable erase operations and exit routine**, when the Flash memory bank was erased successfully or when a failure occurred. In either case bit FWE of the FCR is reset to '0' and the erase routine is exited. This means that the Flash non-verify mode is entered again, where the FCR stays accessible but Flash memory locations can be read normally again using indirect addressing. For returning to the Flash standard mode, bit FWMSET of the FCR must be reset to '0' by the calling routine. The erase routine may return an exit code that indicates correct erasing or identifies the type of error.

### Fundamentals of Flash Technology

The Flash memory included in the C167CR-16F combines the EPROM programming mechanism with electrical erasability (like an EEPROM) to create a highly reliable and cost effective memory. A Flash memory cell consists of a single transistor with a floating gate for charge storage like an EPROM, uses a thinner gate oxide, however.

The programming mechanism of a Flash cell is based on 'hot' electron injection which works as follows: The cell control gate and drain are set to a high voltage and the cell source is grounded. This high voltage between drain and source forces 'hot' electrons supplied from the source to enter the channel. Attracted by the high voltage on the cell's control gate there, free electrons are trapped in the floating gate. The amount of negative charge on the floating gate is basically determined by the length and the number of programming pulses applied to the cell. A special read operation, Program-Verify, is provided for verifying that the charge put onto the floating gate represents a proper '0'.

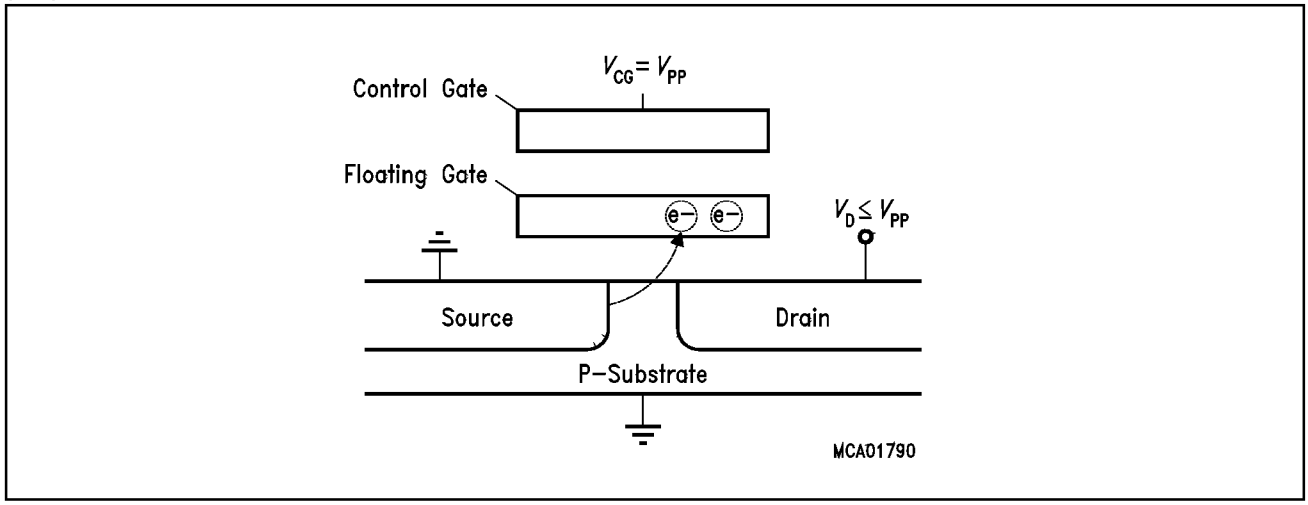



**Figure 9**
**Flash Memory Cell Programming Mechanism**

The cell erase mechanism is based on 'Fowler-Nordheim' tunnelling which works as follows:
A high voltage is applied to the cell's source whilst the control gate grounded. The cell's drain is disconnected in this case. Attracted by the high voltage on the cell's source, electrons migrate from the floating gate to the source. The amount of negative charge removed from the floating gate is basically determined by the length and the number of erasing pulse applied to the cell. A special read operation, Erase-Verify, is provided for verifying that the charge remaining on the floating gate represents a proper '1'.

Unlike a standard EEPROM, where individual bytes can be erased, the Flash memory of the C167CR-16F is erased block-wise which means that the high voltage is applied to all cells belonging to one block simultaneously.

One requirement for performing proper Flash programming and erase operations is to have all cells of a block set to a minimum threshold level before the operation is started. A cell erasing faster than others could have a threshold voltage too low or negative. In this case the corresponding transistor could become conductive and affect other cells placed in the same column of the transistor array. Thus, all cells of that column could erroneously be read as '1' instead of '0'.

To avoid this possible malfunction, the user must equalize the amount of charge on each cell by programming all cells of one block to '0' before performing a block erasure.

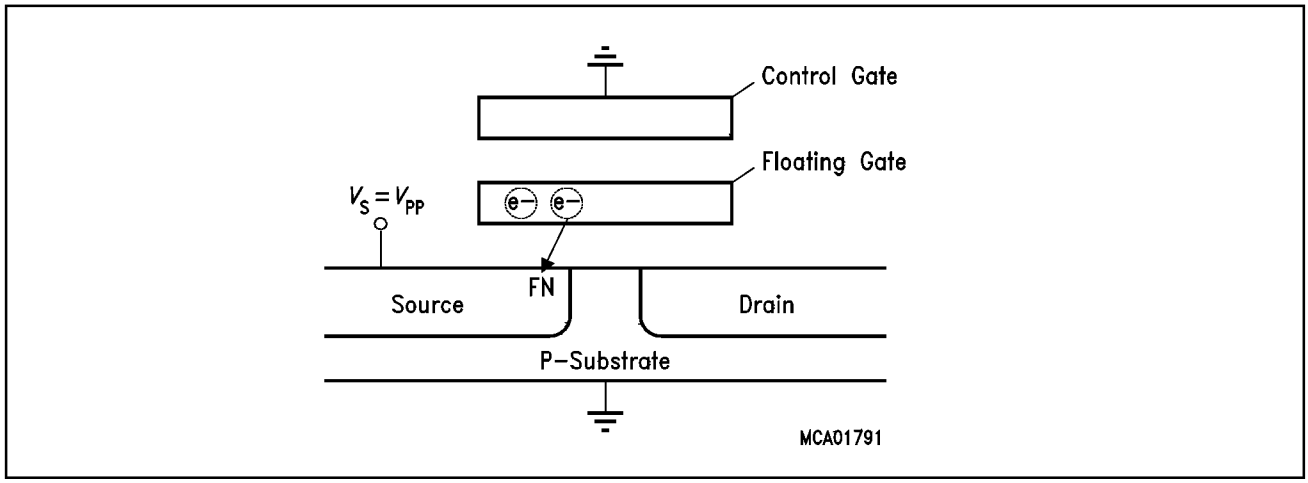



**Figure 10**
**Flash Memory Cell Erase Mechanism**

The introduced erase algorithm meets this requirement. In combination with the Flash technology used, it provides a tight erase threshold voltage distribution, generating a sufficient margin even to cells erasing faster than others.

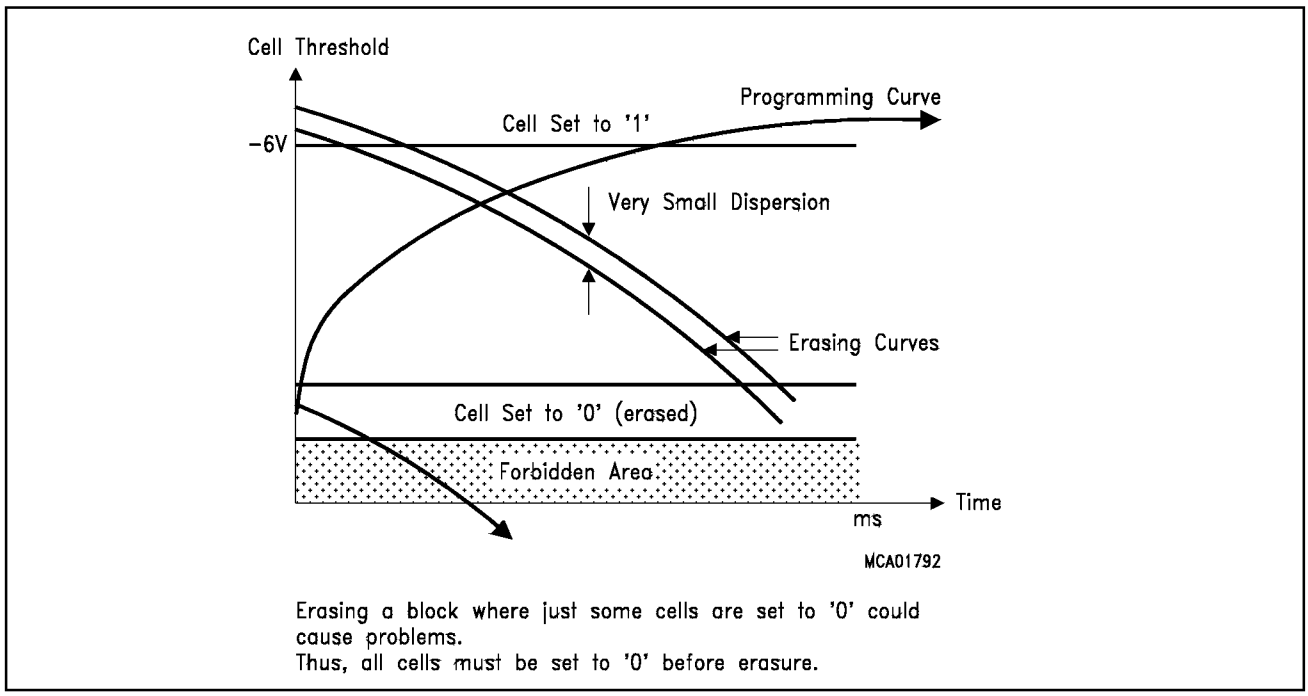



**Figure 11**
**Flash Erasure**

### Absolute Maximum Ratings

Ambient temperature under bias ($T_A$):
SAB-C167CR-16F-LM ..... 0 to +70 °C
SAF-C167CR-16F-LM ..... –40 to +85 °C
Storage temperature ($T_{ST}$) ..... – 65 to +125 °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-16F 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-16F 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-16F.

## DC Characteristics

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V; $f_{CPU}$ = 20 MHz; Reset active
$T_A$ = 0 to +70 °C for SAB-C167CR-16F-LM
$T_A$ = -40 to +85 °C for SAF-C167CR-16F-LM

| 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 (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 (all other outputs) | $V_{OL1}$ | CC | – | 0.45 | V | $I_{OL1}$ = 1.6 mA |
| Output high voltage (PORT0, PORT1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$) | $V_{OH}$ | CC | 0.9 $V_{CC}$<br>2.4 | – | V | $I_{OH}$ = – 500 µA<br>$I_{OH}$ = – 2.4 mA |
| Output high voltage [1] (all other outputs) | $V_{OH1}$ | CC | 0.9 $V_{CC}$<br>2.4 | – | V<br>V | $I_{OH}$ = – 250 µA<br>$I_{OH}$ = – 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 | [5] [8] |
| $\overline{\text{RSTIN}}$ pullup resistor | $R_{RST}$ | CC | 50 | 250 | kΩ | – |
| Read/Write inactive current [4] | $I_{RWH}$ | [2] | – | -40 | µA | $V_{OUT}$ = 2.4 V |
| Read/Write active current [4] | $I_{RWL}$ | [3] | -500 | – | µA | $V_{OUT}$ = $V_{OLmax}$ |
| ALE inactive current [4] | $I_{ALEL}$ | [2] | – | 40 | µA | $V_{OUT}$ = $V_{OLmax}$ |
| ALE active current [4] | $I_{ALEH}$ | [3] | 500 | – | µA | $V_{OUT}$ = 2.4 V |
| Port 6 inactive current [4] | $I_{P6H}$ | [2] | – | -40 | µA | $V_{OUT}$ = 2.4 V |
| Port 6 active current [4] | $I_{P6L}$ | [3] | -500 | – | µA | $V_{OUT}$ = $V_{OL1max}$ |

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---|---|---|---|---|---|
| | | min. | max. | | |
| PORT0 configuration current [4] | $I_{P0H}$ [2] | – | -10 | µA | $V_{IN} = V_{IHmin}$ |
| | $I_{P0L}$ [3] | -100 | – | µA | $V_{IN} = V_{ILmax}$ |
| XTAL1 input current | $I_{IL}$ CC | – | ±20 | µA | 0 V < $V_{IN}$ < $V_{CC}$ |
| Pin capacitance [5] (digital inputs/outputs) | $C_{IO}$ CC | – | 10 | pF | $f$ = 1 MHz<br>$T_A$ = 25 °C |
| Power supply current | $I_{CC}$ | – | 30 + 7 * $f_{CPU}$ | mA | $\overline{RSTIN}$ = $V_{IL2}$<br>$f_{CPU}$ in [MHz] [6] |
| Idle mode supply current | $I_{ID}$ | – | 30 + 2 * $f_{CPU}$ | mA | $\overline{RSTIN}$ = $V_{IH1}$<br>$f_{CPU}$ in [MHz] [6] |
| Power-down mode supply current | $I_{PD}$ | – | 100 | µA | $V_{CC}$ = 5.5 V [7] |
| $V_{PP}$ read current | $I_{PPR}$ | – | 200 | µA | $V_{PP}$ > $V_{CC}$ |
| $V_{PP}$ writing current | $I_{PPW}$ | – | 50 | mA | $f_{CPU}$ = 20 MHz;<br>$V_{PP}$ = 12 V; 32-bit programming |
| $V_{PP}$ during write/read | $V_{PP}$ | 11.4 | 12.6 | V | |

**Notes**

1) 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.

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

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

4) 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.

5) Not 100% tested, guaranteed by design.

6) 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 $V_{IH}$.

7) 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.

8) 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**.

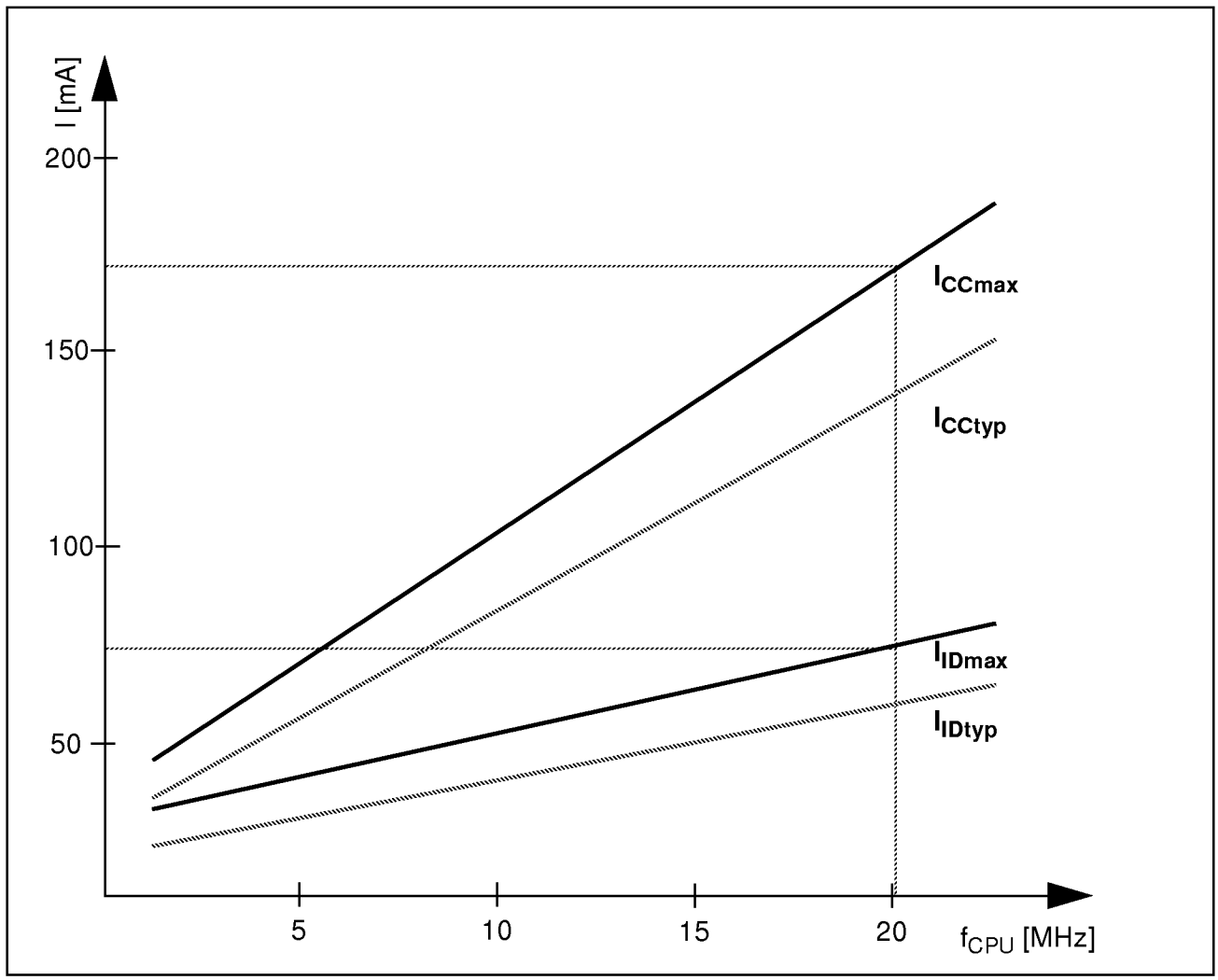



**Figure 12**
**Supply/Idle Current as a Function of Operating Frequency**

### A/D Converter Characteristics

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-16F-LM
$T_A$ = -40 to +85 °C for SAF-C167CR-16F-LM
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-16F's A/D Converter 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.

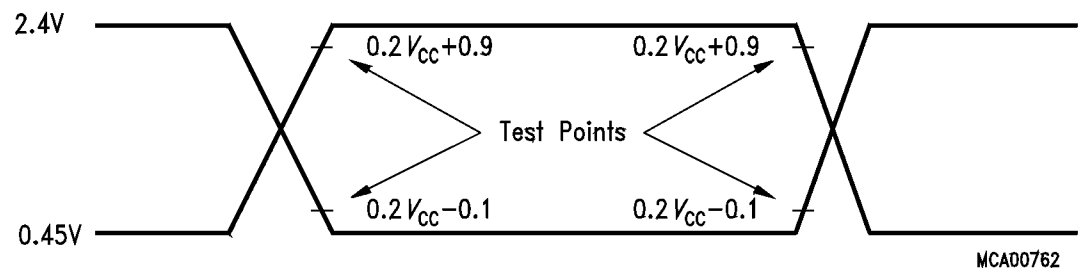
4) This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.

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. Please note that this test condition for $V_{AREF}$ represents a stress situation which should be avoided during normal operation.
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.
Please note that the given formula applies for direct supply of $V_{AREF}$. The internal resistance of the analog voltage source can be increased by providing a support capacitance close to the $V_{AREF}$ pin.

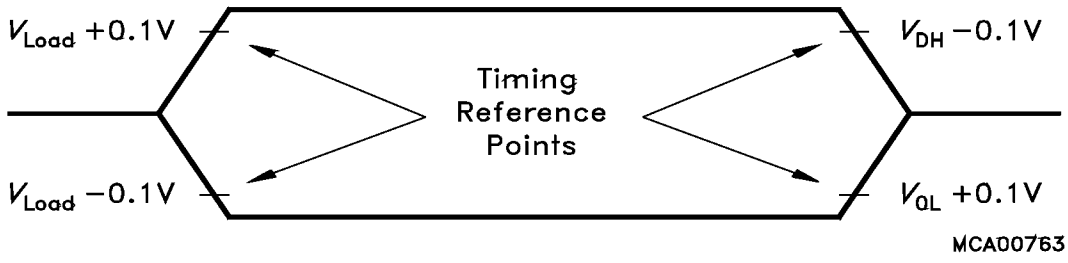
7) Not 100% tested, guaranteed by design.

### 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 13**
**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 14**
**Float Waveforms**

## AC Characteristics
## Definition of Internal Timing

The internal operation of the C167CR-16F 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).

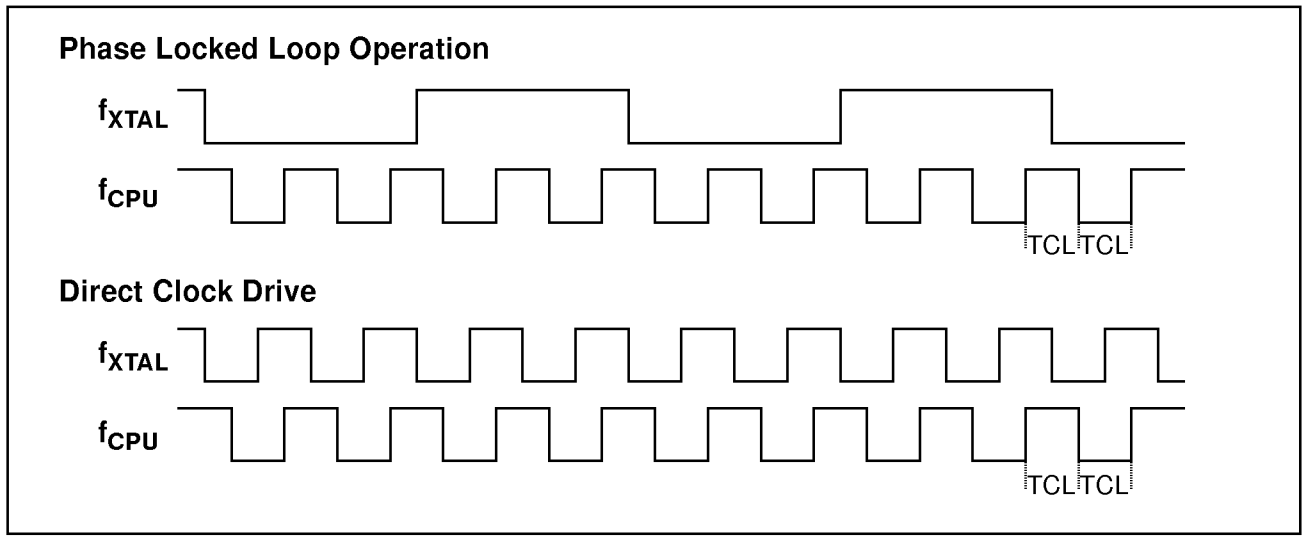



**Figure 15**
**Generation Mechanisms for the CPU Clock**

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-16F.

### Direct Drive

When pin P0.15 (P0H.7) is low (‘0’) 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

When pin P0.15 (P0H.7) is high ('1') during reset the on-chip phase locked loop is enabled and provides the CPU clock. The PLL multiplies the input frequency by 4 (ie. $f_{CPU} = f_{XTAL} * 4$). With every fourth transition of $f_{XTAL}$ the PLL circuit synchronizes the CPU clock to the input clock. This synchronization is done smoothly, 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 $D_N$:

$$(N\,TCL)_{min} = N\,TCL_{NOM} * (1 - D_N / 100)$$

$D_N = \pm(4 - N/15)$ [%],
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.

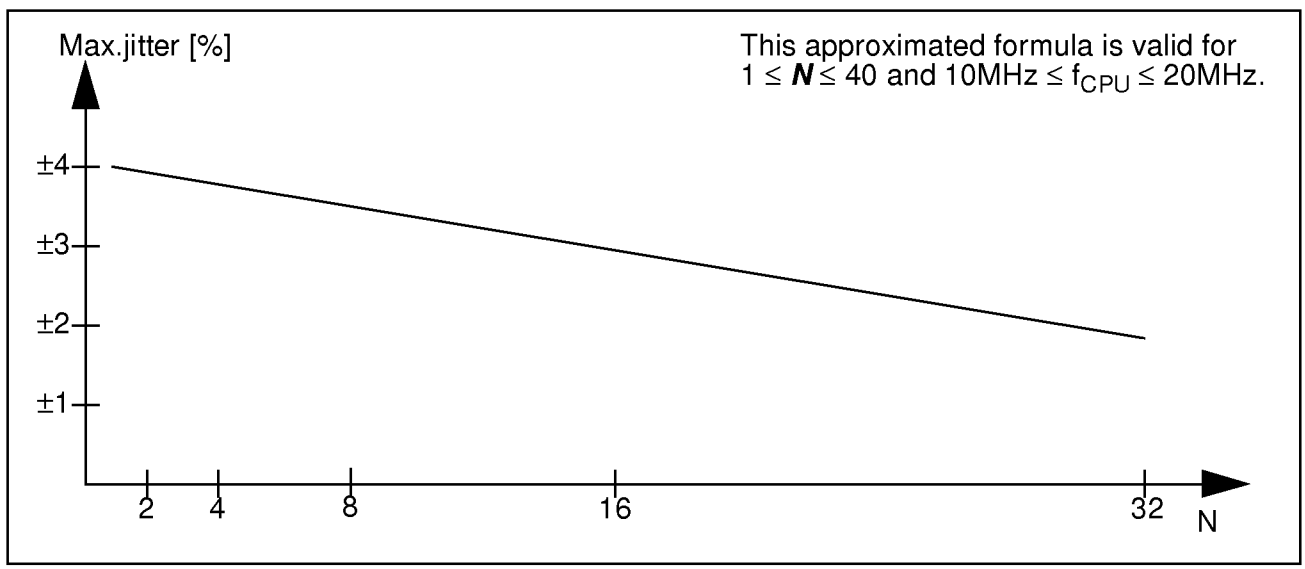



**Figure 16**
**Approximated Maximum PLL Jitter**

## AC Characteristics
## External Clock Drive XTAL1

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-16F-LM
$T_A$ = -40 to +85 °C for SAF-C167CR-16F-LM

| Parameter | Symbol | | Direct Drive 1:1 | | PLL 1:4 | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| Oscillator period | $t_{OSC}$ | SR | 50 | 1000 | 200 | 333 | ns |
| High time | $t_1$ | SR | 23 [1) 2)] | – | 10 | – | ns |
| Low time | $t_2$ | SR | 23 [1) 2)] | – | 10 | – | ns |
| Rise time [3)] | $t_3$ | SR | – | 10 [2)] | – | 10 [2)] | ns |
| Fall time [3)] | $t_4$ | SR | – | 10 [2)] | – | 10 [2)] | ns |

[1)] For temperatures above $T_A$ = +85 °C the minimum value for $t_1$ and $t_2$ is 25 ns.
[2)] The clock input signal must reach the defined levels $V_{IL}$ and $V_{IH2}$.
[3)] Not 100% tested, guaranteed by design.

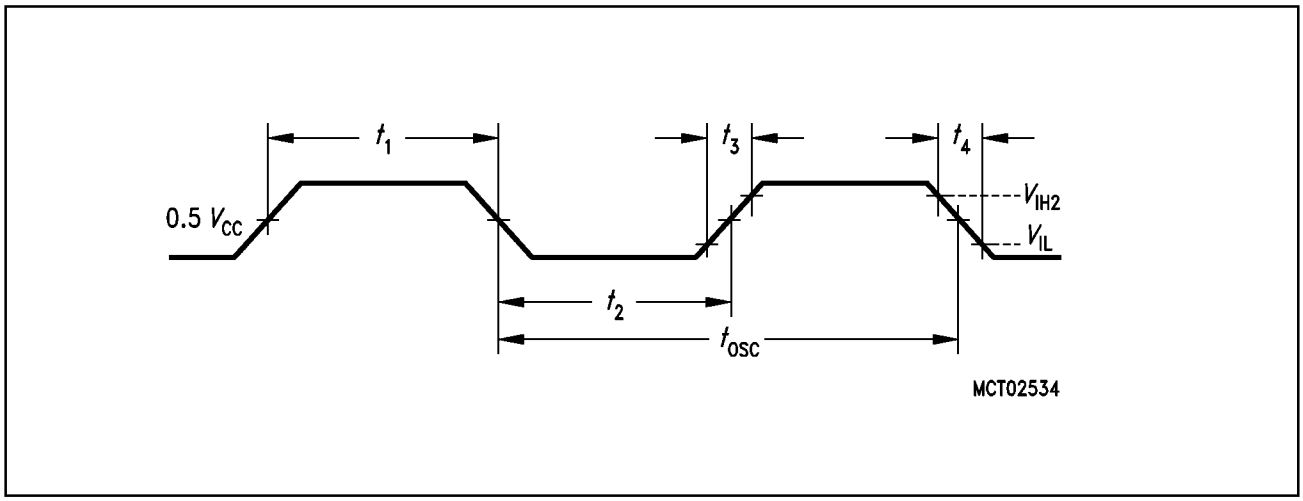



**Figure 17**
**External Clock Drive XTAL1**

### 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-16F-LM
$T_A$ = -40 to +85 °C for SAF-C167CR-16F-LM
$C_L$ (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 | 10 + $t_A$ | – | TCL - 15 + $t_A$ | – | 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 | – | 5 | – | 5 | ns |
| Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay) | $t_{11}$ | CC | – | 30 | – | TCL + 5 | 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 | – | 35 + $t_F$ | – | 2TCL - 15 + $t_F$ | ns |
| Data valid to $\overline{WR}$ | $t_{22}$ | SR | 25 + $t_C$ | – | 2TCL - 25 + $t_C$ | – | ns |
| Data hold after $\overline{WR}$ | $t_{23}$ | CC | 35 + $t_F$ | – | 2TCL - 15 + $t_F$ | – | ns |
| ALE rising edge after $\overline{RD}$, $\overline{WR}$ | $t_{25}$ | CC | 35 + $t_F$ | – | 2TCL - 15 + $t_F$ | – | ns |
| Address hold after $\overline{WR}$ [1] | $t_{27}$ | CC | 35 + $t_F$ | – | 2TCL - 15 + $t_F$ | – | 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 $\overline{RD}$, $\overline{WR}$ | $t_{40}$ | CC | 60 + $t_F$ | – | 3TCL - 15 + $t_F$ | – | ns |
| ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW delay) | $t_{42}$ | CC | 20 + $t_A$ | – | TCL - 5 + $t_A$ | – | ns |
| ALE fall. edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW delay) | $t_{43}$ | CC | -5 + $t_A$ | – | -5 + $t_A$ | – | 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 | – | 25 + $t_C$ | – | 2TCL - 25 + $t_C$ | 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 | – | 50 + $t_C$ | – | 3TCL - 25 + $t_C$ | 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 | 35 + $t_C$ | – | 2TCL - 15 + $t_C$ | – | 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 |

[1] It is guaranteed by design that read data are latched before the address changes.

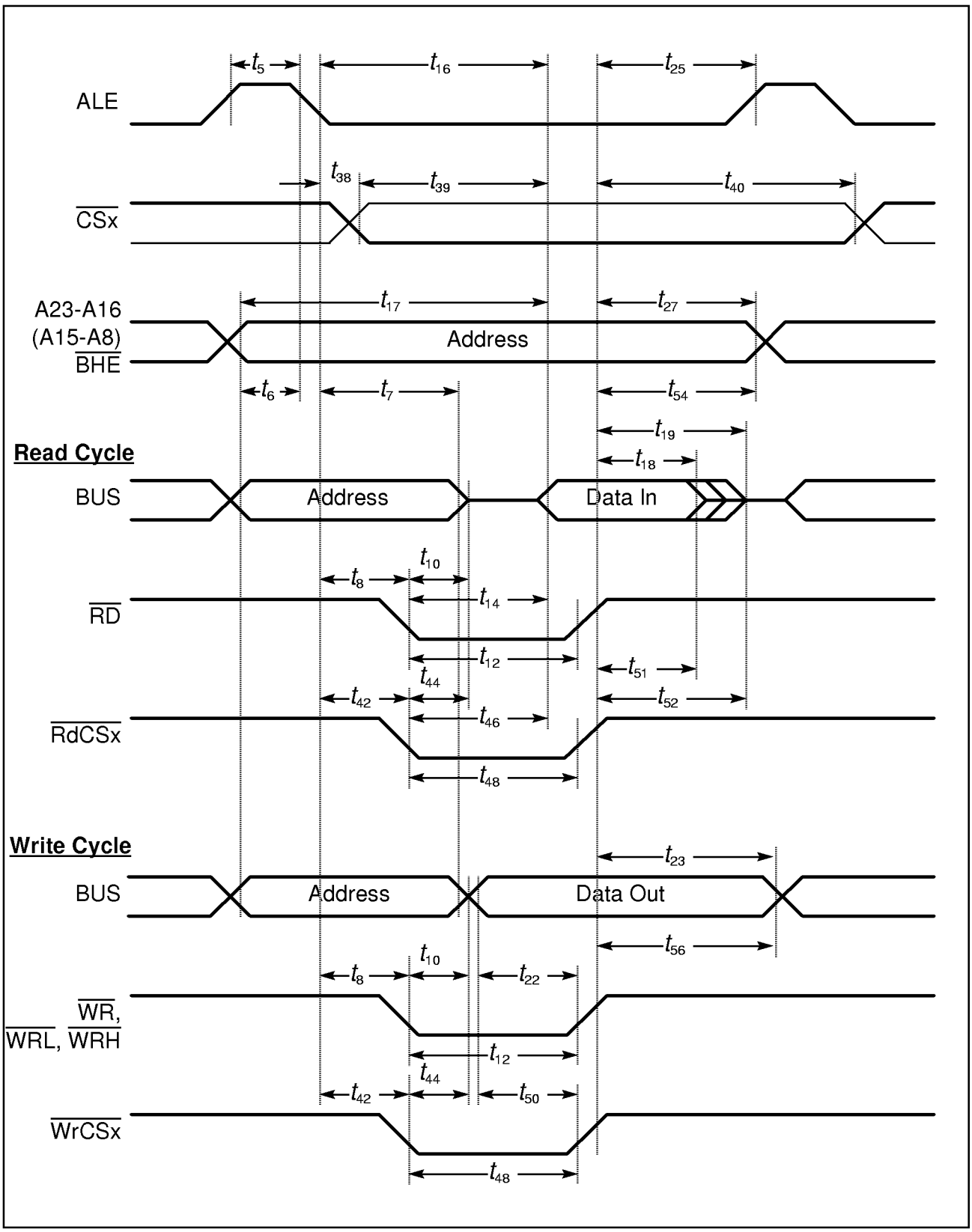



**Figure 18-1**
**External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE**

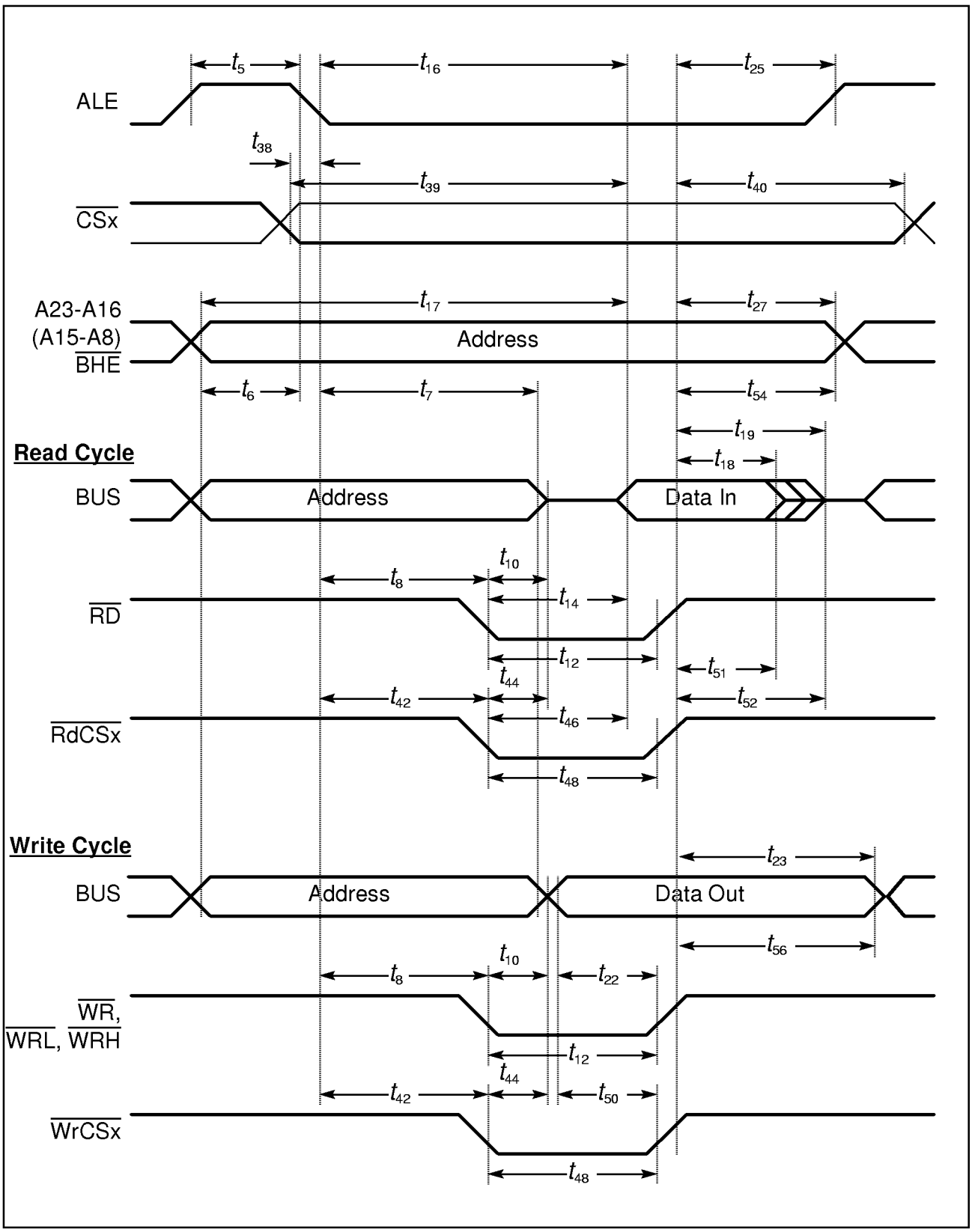



**Figure 18-2**
**External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Extended ALE**

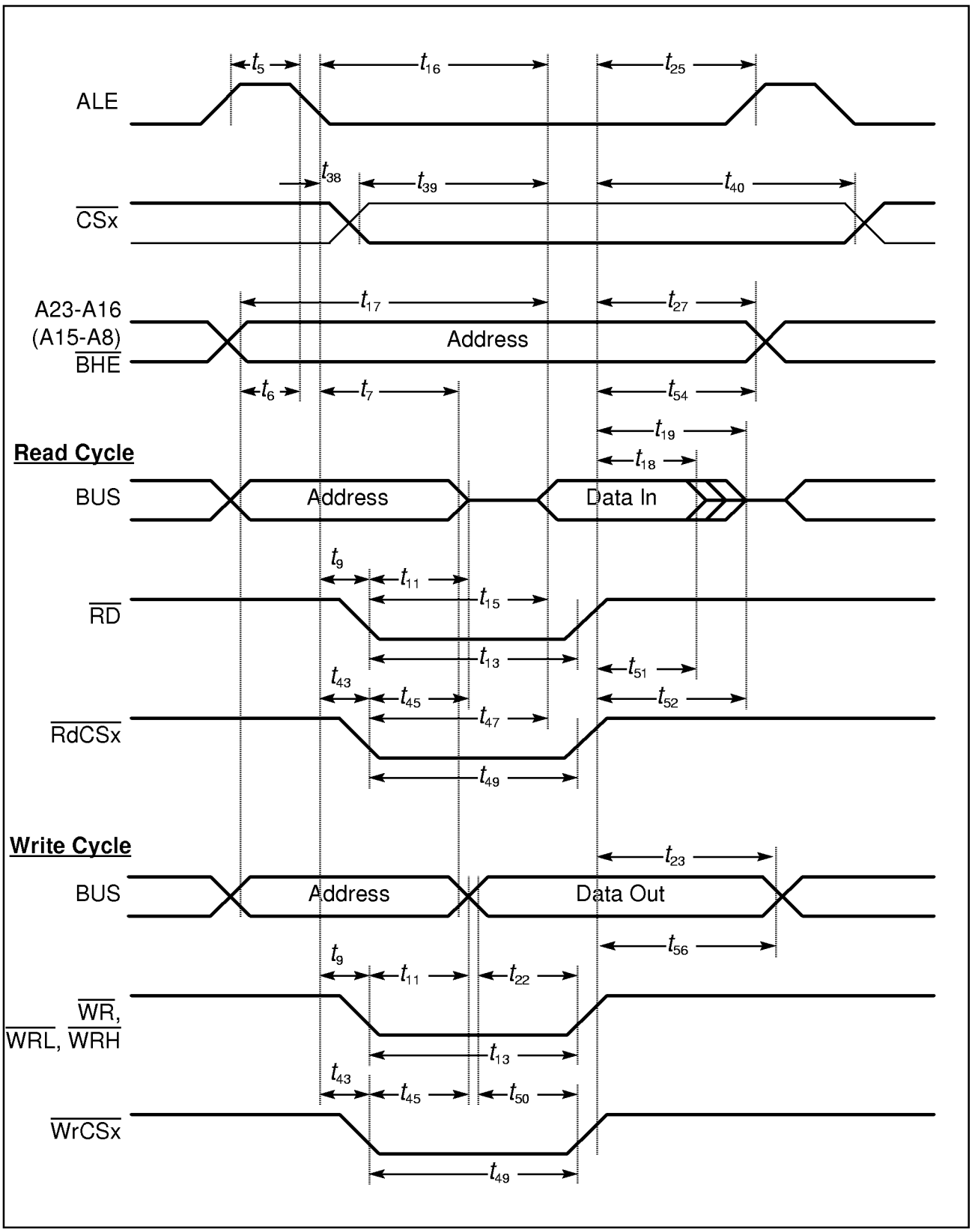



**Figure 18-3**
**External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE**

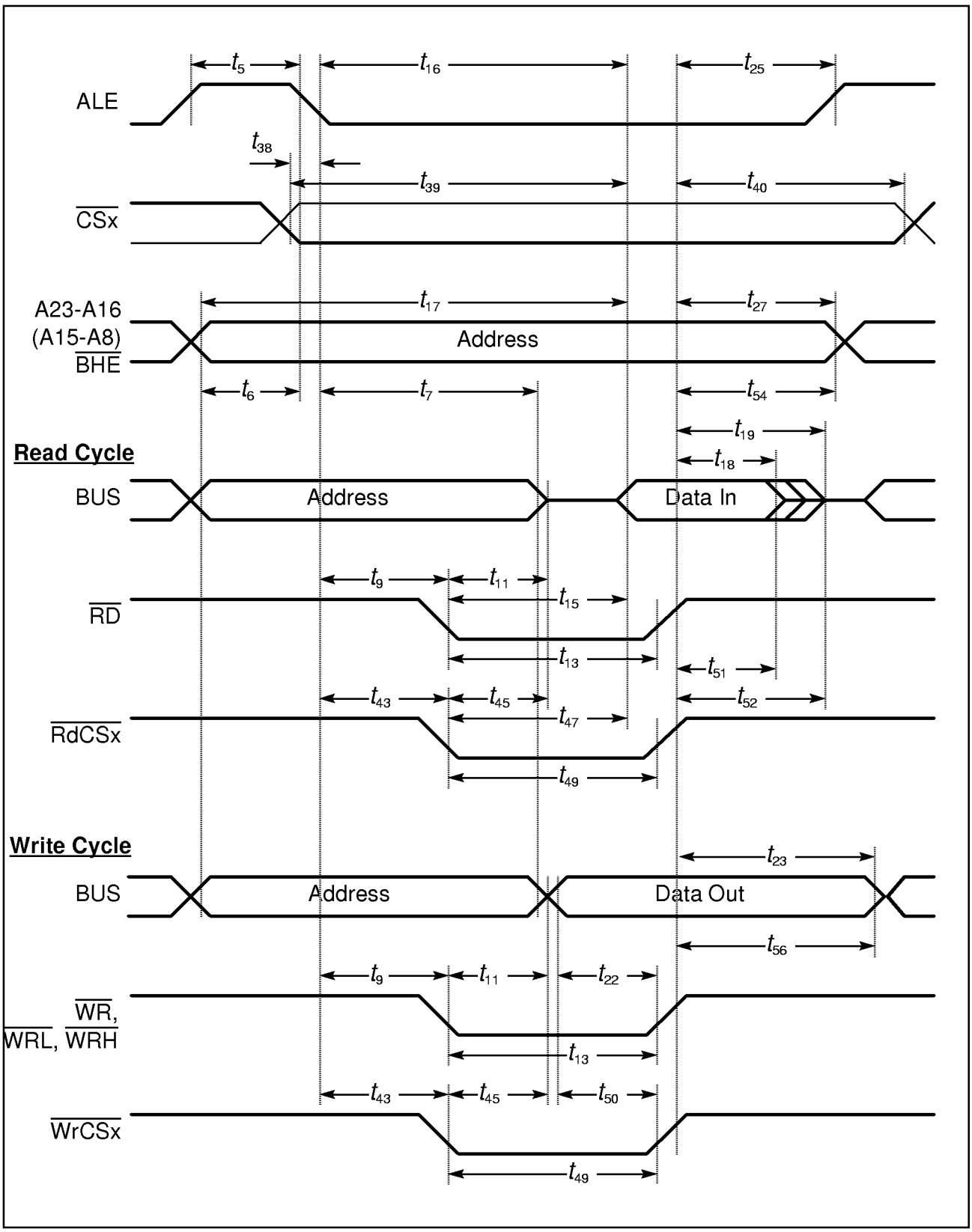



**Figure 18-4**
**External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE**

## AC Characteristics
## Demultiplexed Bus

$V_{CC}$ = 5 V ± 10 %; $V_{SS}$ = 0 V
$T_A$ = 0 to +70 °C for SAB-C167CR-16F-LM
$T_A$ = -40 to +85 °C for SAF-C167CR-16F-LM
$C_L$ (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 | 10 + $t_A$ | – | TCL - 15 + $t_A$ | – | 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 | – | 35 + $t_F$ | – | 2TCL - 15 + 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 | 25 + $t_C$ | – | 2TCL - 25 + $t_C$ | – | ns |
| Data hold after $\overline{WR}$ | $t_{24}$ | CC | 15 + $t_F$ | – | TCL - 10 + $t_F$ | – | ns |
| ALE rising edge after $\overline{RD}$, $\overline{WR}$ | $t_{26}$ | CC | -10 + $t_F$ | – | -10 + $t_F$ | – | ns |

| Parameter | Symbol | | Max. CPU Clock = 20 MHz | | Variable CPU Clock 1/2TCL = 1 to 20 MHz | | Unit |
|---|---|---|---|---|---|---|---|
| | | | min. | max. | min. | max. | |
| 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 | 10 + $t_F$ | – | TCL - 15 + $t_F$ | – | ns |
| ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (with RW-delay) | $t_{42}$ | CC | 20 + $t_A$ | – | TCL - 5 + $t_A$ | – | ns |
| ALE falling edge to $\overline{RdCS}$, $\overline{WrCS}$ (no RW-delay) | $t_{43}$ | CC | -5 + $t_A$ | – | -5 + $t_A$ | – | ns |
| $\overline{RdCS}$ to Valid Data In (with RW-delay) | $t_{46}$ | SR | – | 25 + $t_C$ | – | 2TCL - 25 + $t_C$ | ns |
| $\overline{RdCS}$ to Valid Data In (no RW-delay) | $t_{47}$ | SR | – | 50 + $t_C$ | – | 3TCL - 25 + $t_C$ | 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 | 35 + $t_C$ | – | 2TCL - 15 + $t_C$ | – | 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 | 10 + $t_F$ | – | TCL - 15 + $t_F$ | – | 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.

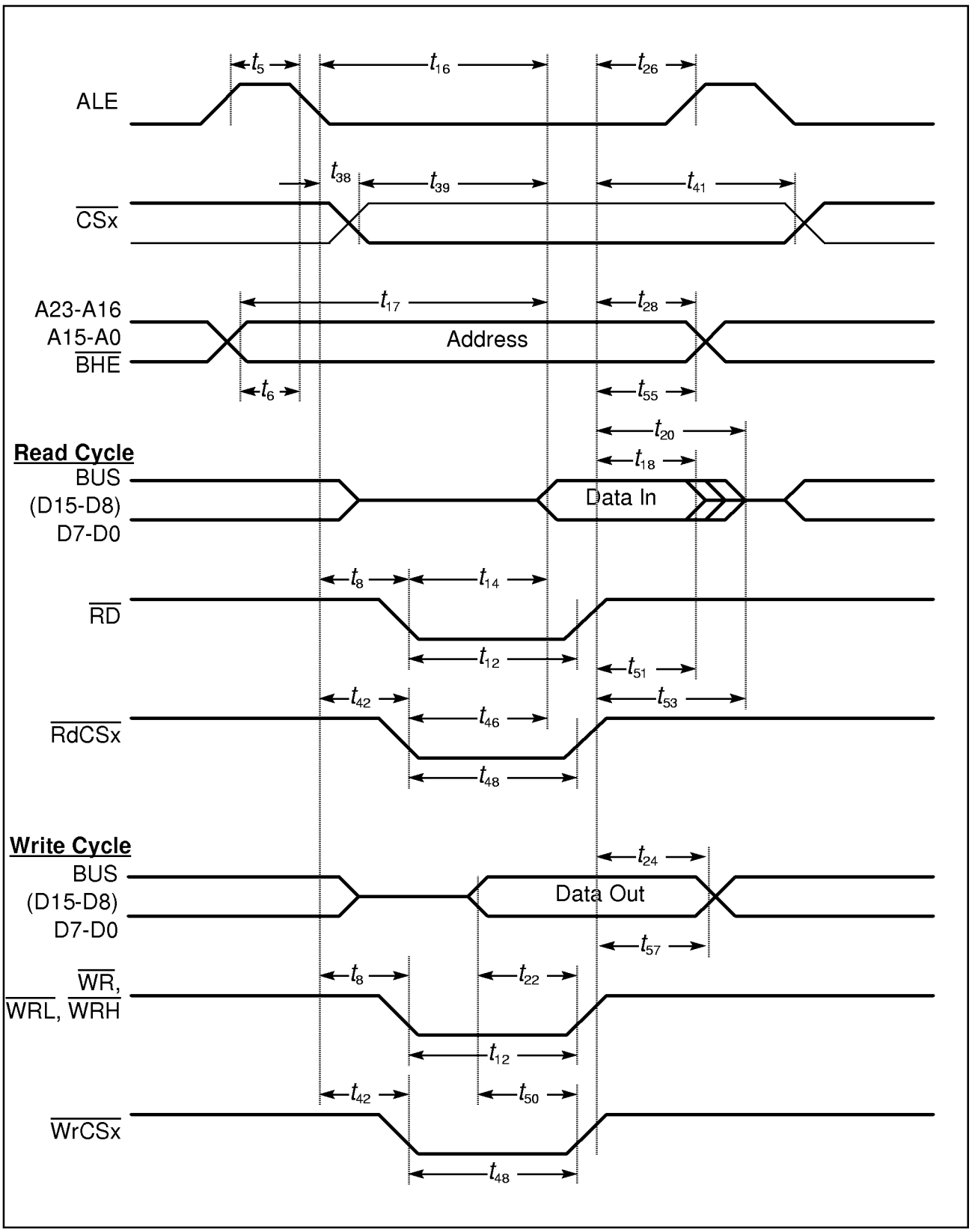



**Figure 19-1**
**External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Normal ALE**

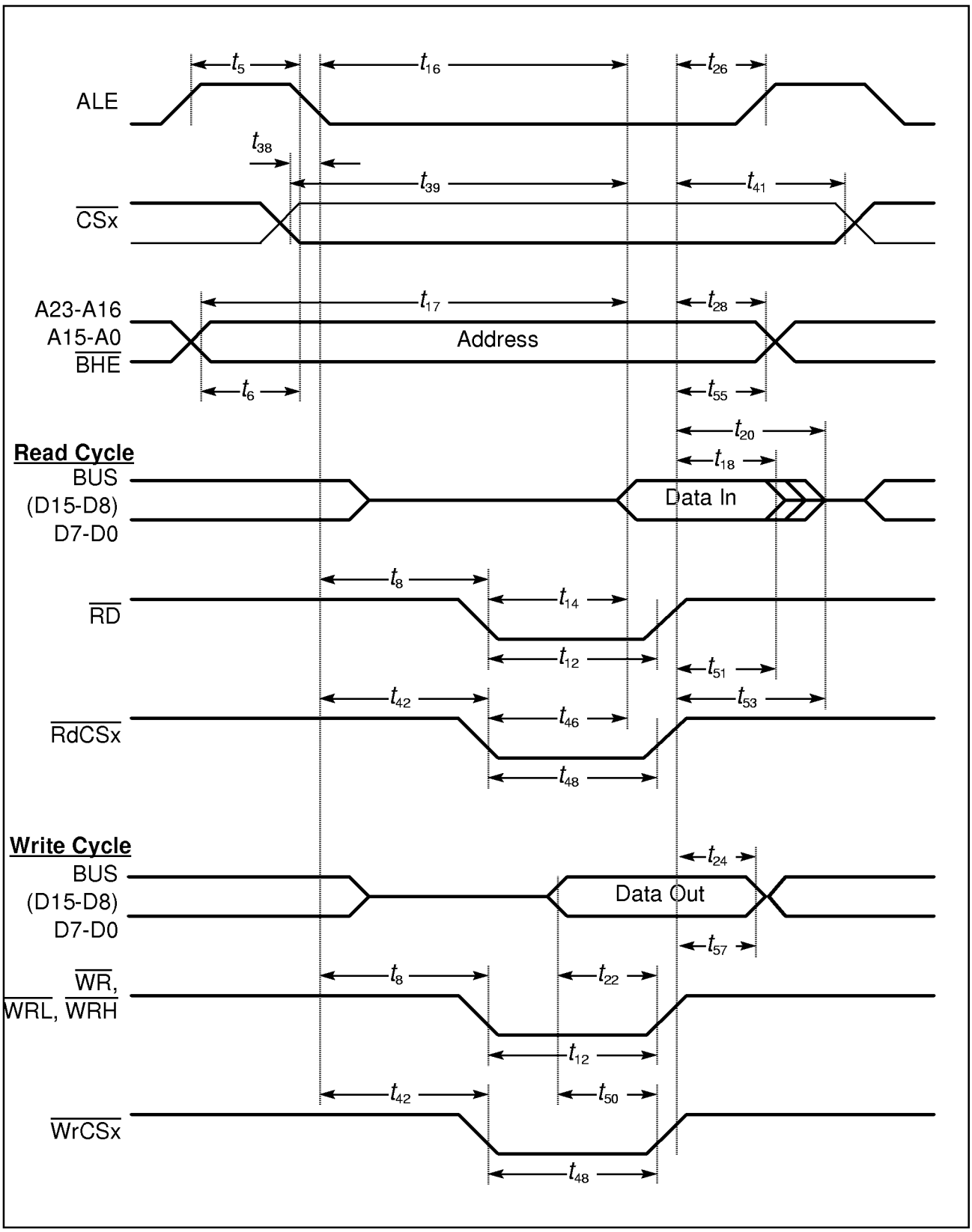



**Figure 19-2**
**External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE**

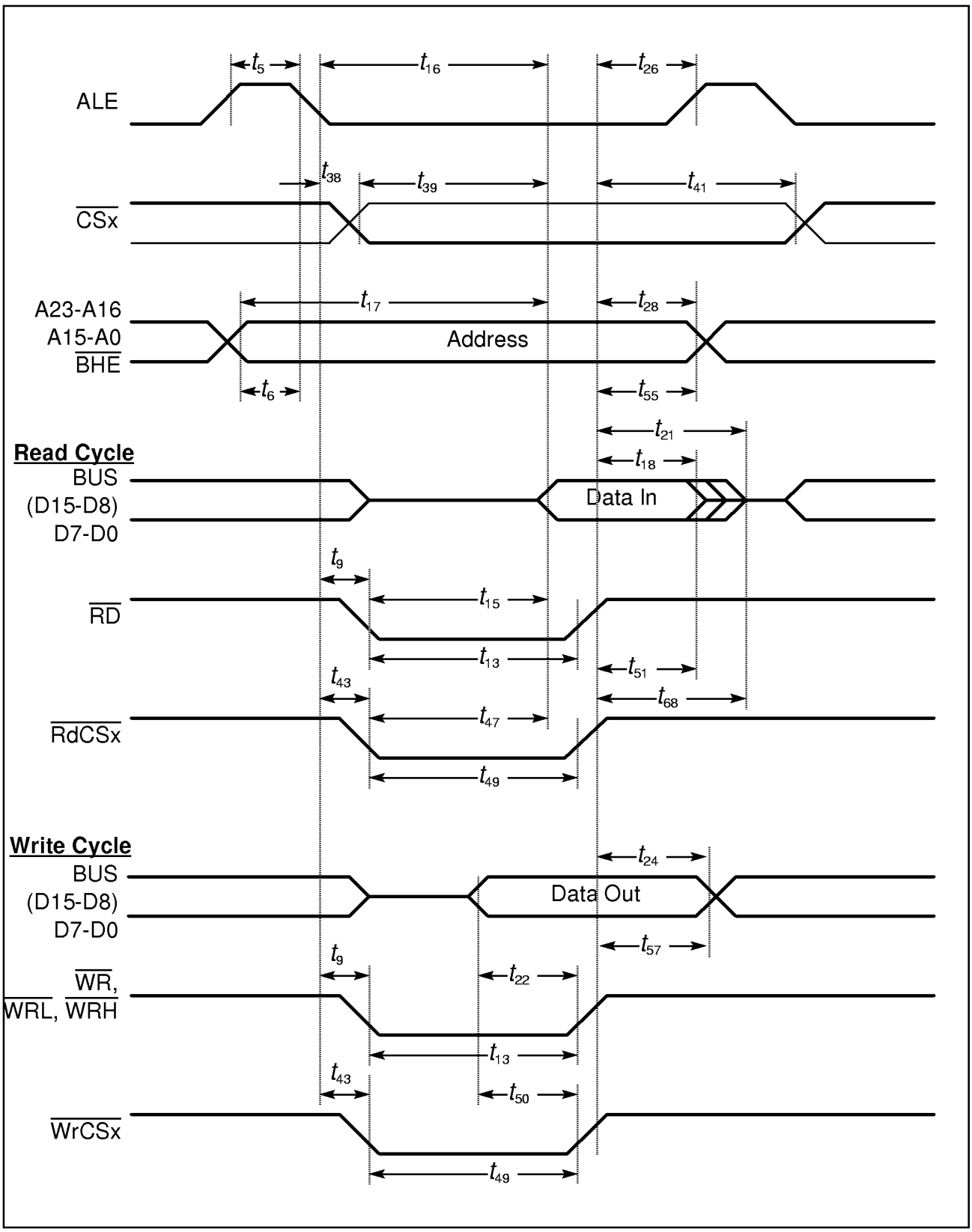



**Figure 19-3**
**External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE**

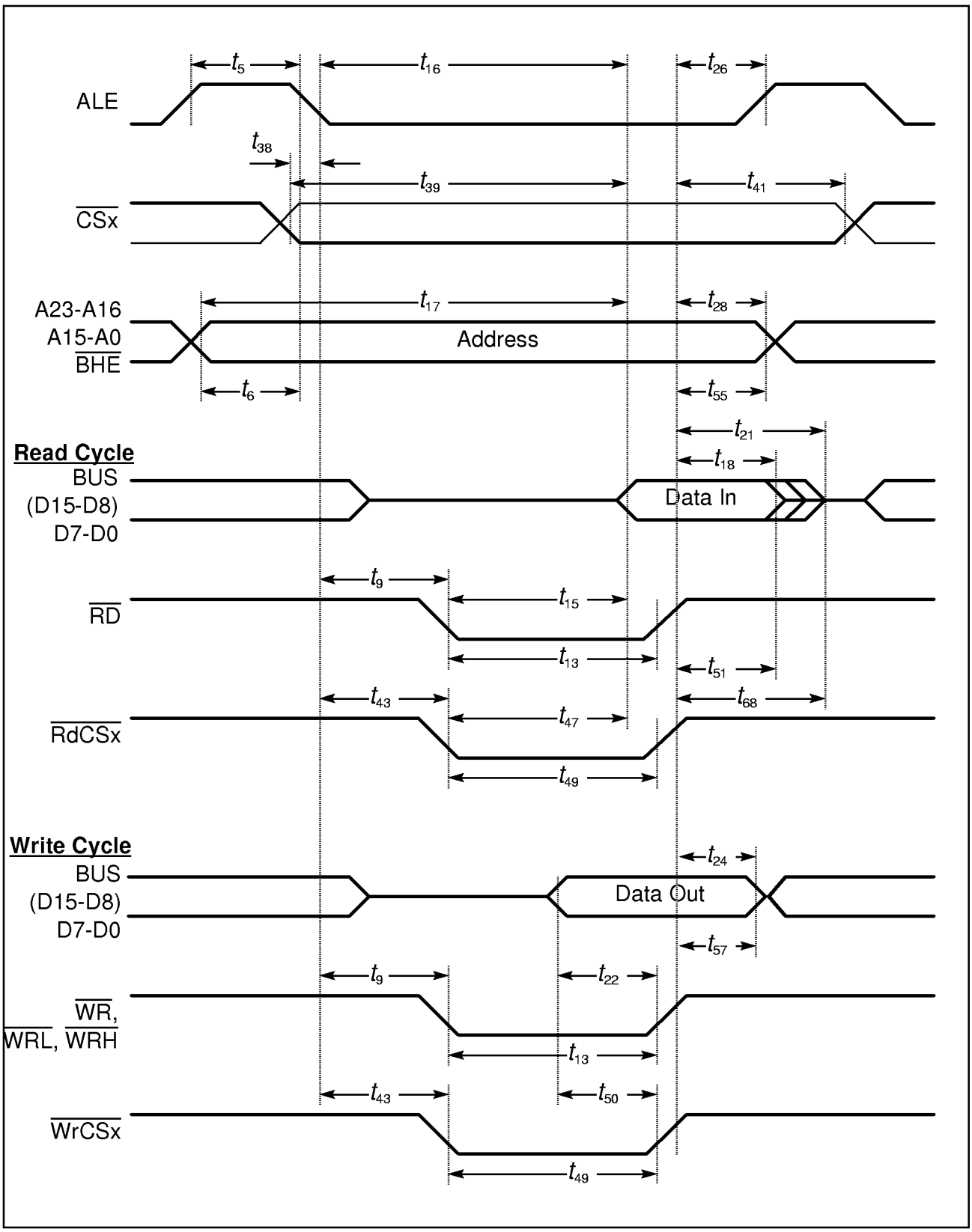



**Figure 19-4**
**External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE**

## 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-16F-LM
$T_A$ = -40 to +85 °C for SAF-C167CR-16F-LM
$C_L$ (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 | 20 | – | TCL – 5 | – | ns |
| CLKOUT low time | $t_{31}$ | CC | 15 | – | TCL – 10 | – | ns |
| CLKOUT rise time | $t_{32}$ | CC | – | 5 | – | 5 | ns |
| CLKOUT fall time | $t_{33}$ | CC | – | 5 | – | 5 | 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 | 15 | – | 15 | – | ns |
| Synchronous $\overline{\text{READY}}$ hold time after CLKOUT | $t_{36}$ | SR | 0 | – | 0 | – | ns |
| Asynchronous $\overline{\text{READY}}$ low time | $t_{37}$ | SR | 65 | – | 2TCL + 15 | – | ns |
| Asynchronous $\overline{\text{READY}}$ setup time [1] | $t_{58}$ | SR | 15 | – | 15 | – | ns |
| Asynchronous $\overline{\text{READY}}$ hold time [1] | $t_{59}$ | SR | 0 | – | 0 | – | ns |
| Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus) [2] | $t_{60}$ | SR | 0 | 0 + 2$t_A$ + $t_C$ + $t_F$ [2] | 0 | TCL - 25 + 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 $2t_A$ and $t_C$ refer to the next following bus cycle, $t_F$ refers to the current bus cycle.

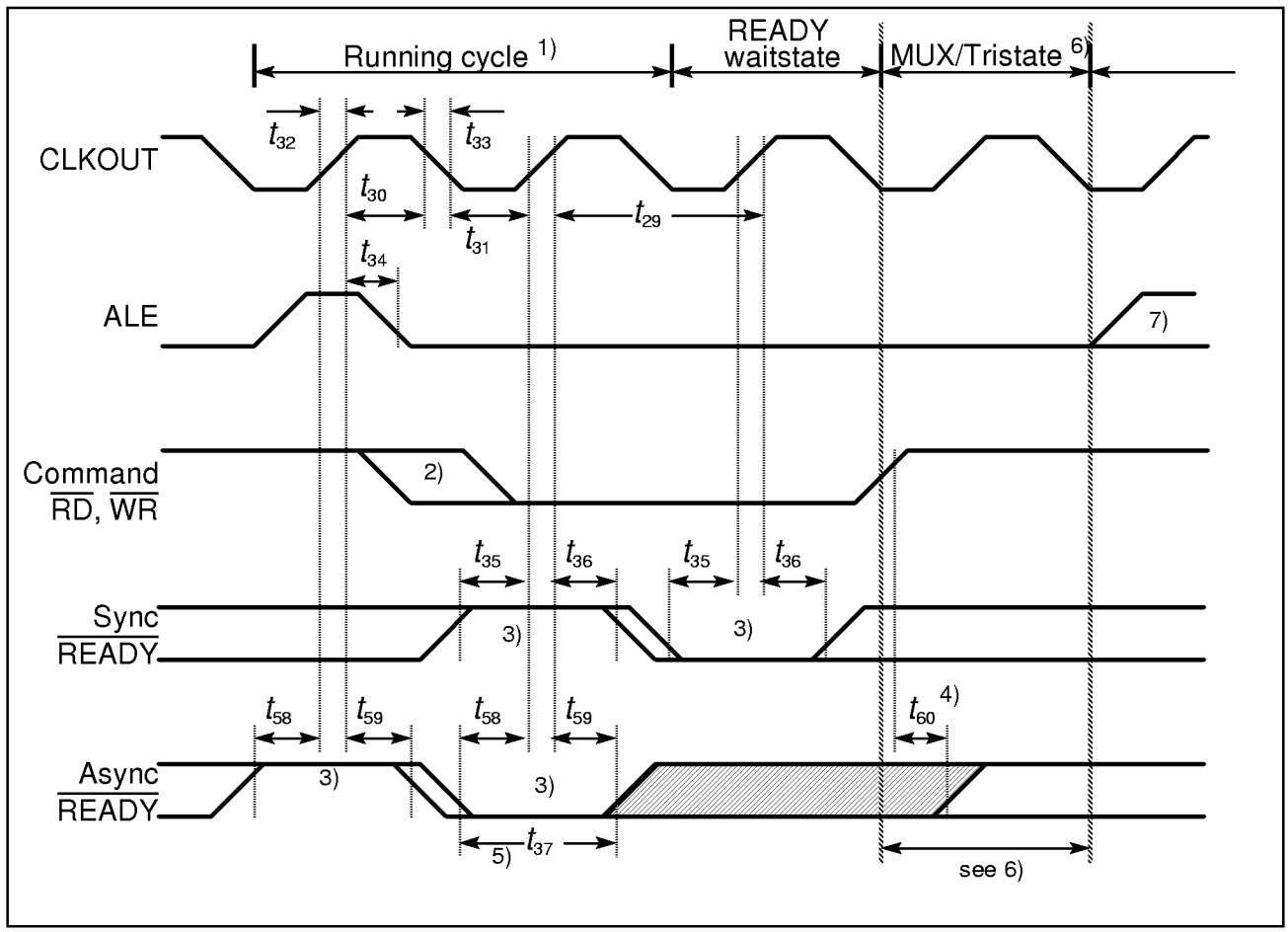



**Figure 20**
**CLKOUT and $\overline{\text{READY}}$**

**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-16F-LM
$T_A$ = -40 to +85 °C for SAF-C167CR-16F-LM
$C_L$ (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 | -5 | 25 | -5 | 25 | ns |
| Other bus signals release | $t_{66}$ | CC | – | 20 | – | 20 | ns |
| Other bus signals drive | $t_{67}$ | CC | -5 | 25 | -5 | 25 | ns |

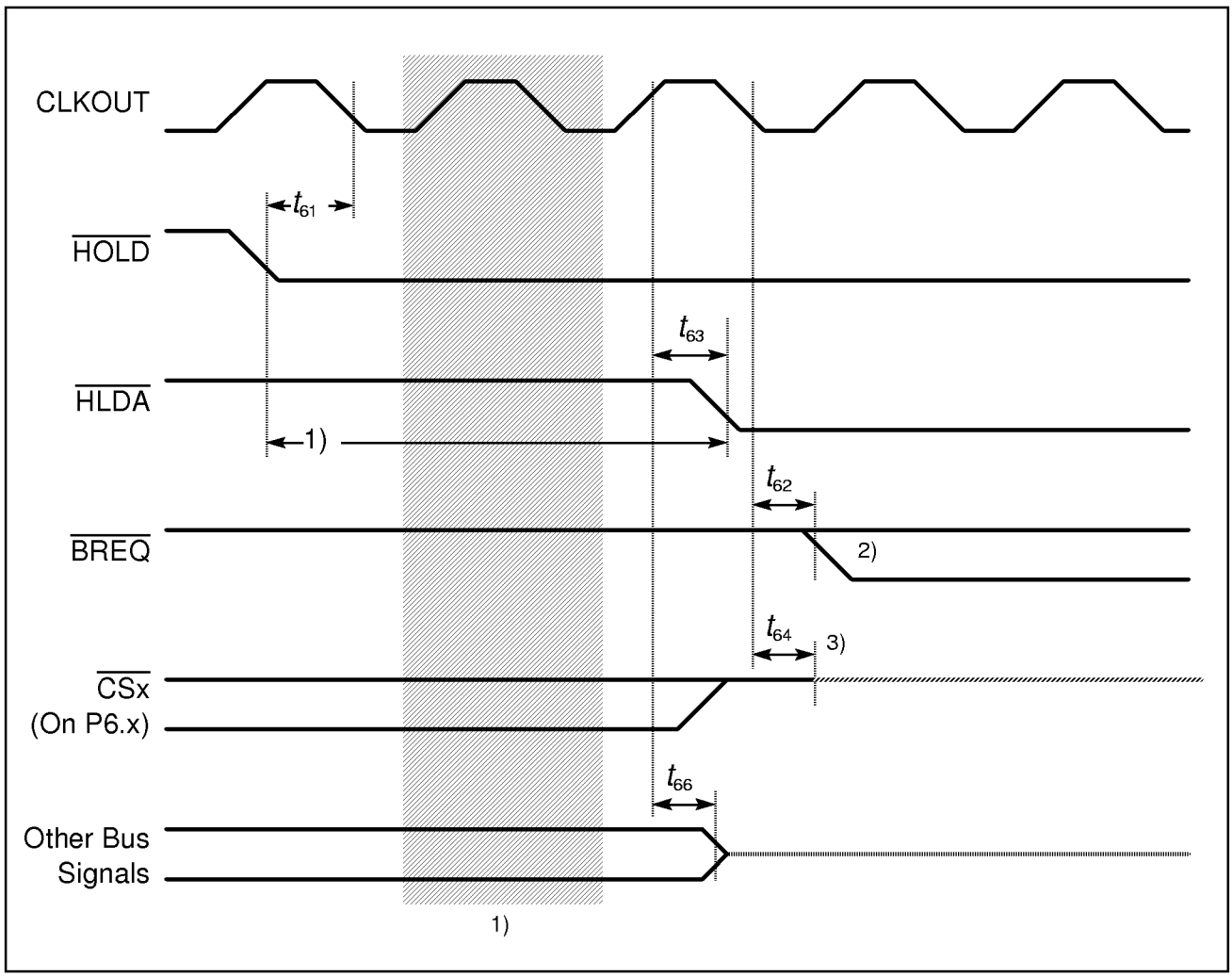



**Figure 21**
**External Bus Arbitration, Releasing the Bus**

**Notes**

1) The C167CR-16F 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 $t_{64}$.

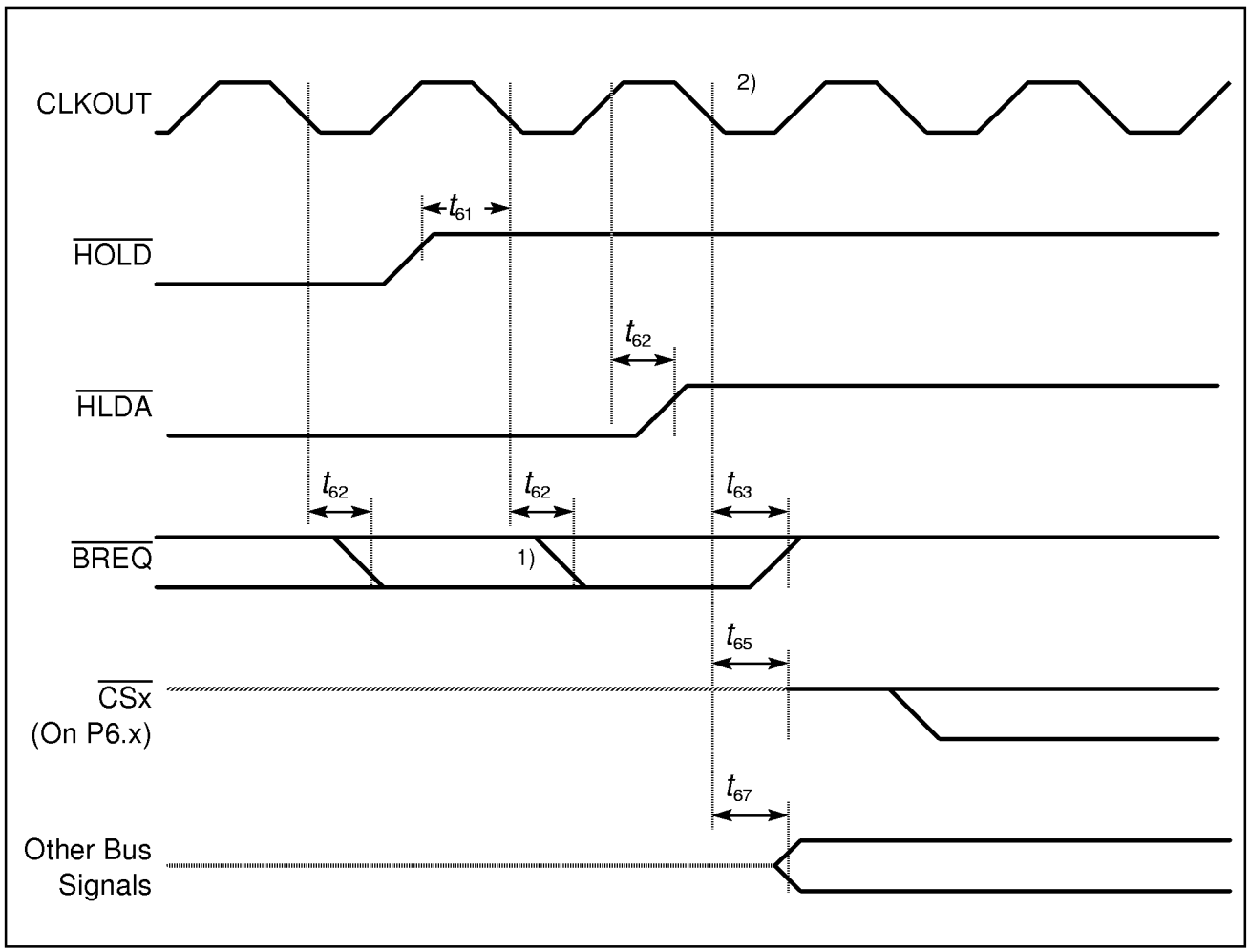



**Figure 22**
**External Bus Arbitration, (Regaining the Bus)**

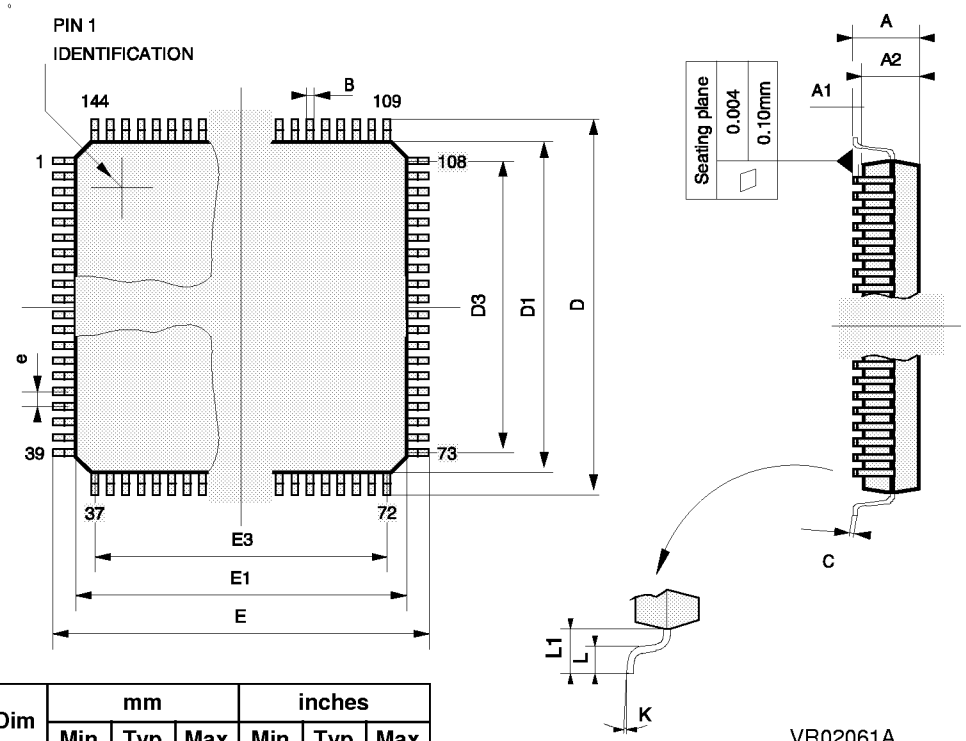
**Notes**

1) This is the last chance for BREQ to trigger the indicated regain-sequence.
Even if BREQ is activated earlier, the regain-sequence is initiated by HOLD going high.
Please note that HOLD may also be deactivated without the C167CR-16F requesting the bus.

2) The next C167CR-16F driven bus cycle may start here.

## Package Outline

**Plastic Package, P-MQFP-144-4 (SMD)**
(Plastic Metric Quad Flat Package)




| Dim | mm | | | inches | | |
|---|---|---|---|---|---|---|
| | Min | Typ | Max | Min | Typ | Max |
| A | | | 4.07 | | | 0.160 |
| A1 | 0.25 | | | 0.010 | | |
| A2 | 3.17 | 3.42 | 3.67 | 0.125 | 0.135 | 0.144 |
| B | 0.22 | | 0.38 | 0.009 | | 0.015 |
| C | 0.13 | | 0.23 | 0.005 | | 0.009 |
| D | 30.95 | 31.20 | 31.45 | 1.219 | 1.228 | 1.238 |
| D1 | 27.90 | 28.00 | 28.10 | 1.098 | 1.102 | 1.106 |
| D3 | | 22.75 | | | 0.896 | |
| e | | 0.65 | | | 0.026 | |
| E | 30.95 | 31.20 | 31.45 | 1.219 | 1.228 | 1.238 |
| E1 | 27.90 | 28.00 | 28.10 | 1.098 | 1.102 | 1.106 |
| E3 | | 22.75 | | | 0.896 | |
| L | 0.65 | 0.80 | 0.95 | 0.026 | 0.031 | 0.037 |
| L1 | | 1.60 | | | 0.063 | |
| K | 0°(min.), 7°(max.) | | | | | |

**Figure 23**

### Sorts of Packing

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

SMD = Surface Mounted Device

Dimensions in mm