SIEMENS

C16x-Family of C165
High-Performance CMOS 16-Bit Microcontrollers

Preliminary
C165 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 bits), 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
- Up to 16 MBytes Linear Address Space for Code and Data
- 2 KBytes On-Chip RAM
- 4 KBytes On-Chip ROM (RM types only)
- Programmable External Bus Characteristics for Different Address Ranges
- 8-Bit or 16-Bit External Data Bus
- Multiplexed or Demultiplexed External Address/Data Buses
- Five Programmable Chip-Select Signals
- Hold- and Hold-Acknowledge Bus Arbitration Support
- 1024 Bytes On-Chip Special Function Register Area
- Idle and Power Down Modes
- 8-Channel Interrupt-Driven Single-Cycle Data Transfer Facilities via Peripheral Event Controller (PEC)
- 16-Priority-Level Interrupt System with 28 Sources, Sample-Rate down to 50 ns
- Two Multi-Functional General Purpose Timer Units with 5 Timers
- Two Serial Channels (Synchronous/Asynchronous and High-Speed-Synchronous)
- Programmable Watchdog Timer
- Up to 77 General Purpose I/O Lines
- 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
- 100-Pin MQFP Package (EIAJ)
- 100-Pin TQFP Package (Thin QFP)

Introduction

The C165 is a new derivative of the Siemens SAB 80C166 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.

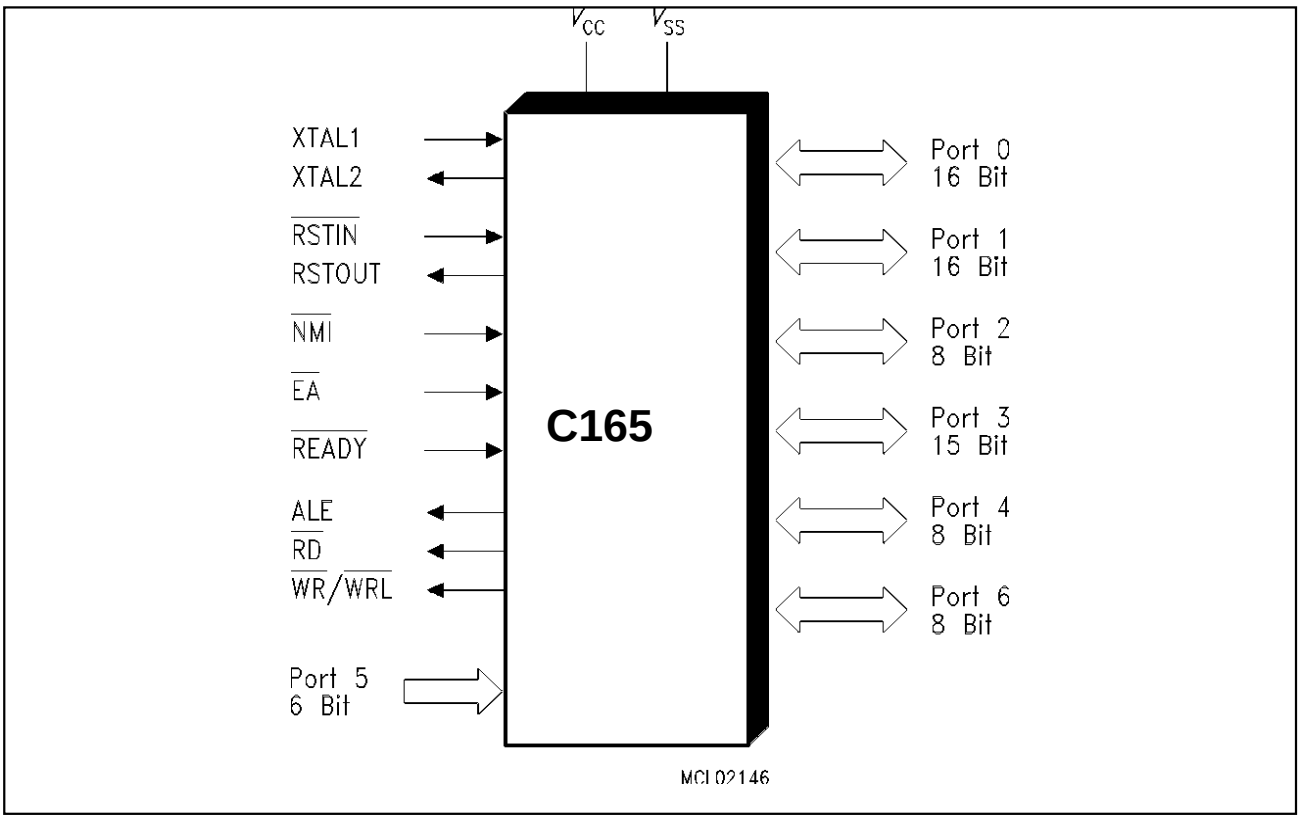


Figure 1
Logic Symbol

Ordering Information

Type	Ordering Code	Package	Function
SAB-C165-RM	Q67121-D...	P-MQFP-100-2	16-bit microcontroller with 2 KByte RAM and 4 KByte ROM Temperature range 0 to +70 °C
SAB-C165-LM	Q67121-C862	P-MQFP-100-2	16-bit microcontroller with 2 KByte RAM Temperature range 0 to +70 °C
SAF-C165-LM	Q67121-C923	P-MQFP-100-2	16-bit microcontroller with 2 KByte RAM Temperature range -40 to +85 °C

Note: The ordering codes (Q67121-D...) for the Mask-ROM versions are defined for each product after verification of the respective ROM code.

Ordering Information

Type	Ordering Code	Package	Function
SAB-C165-RF	Q67121-D...	P-TQFP-100-3	16-bit microcontroller with 2 KByte RAM and 4 KByte ROM Temperature range 0 to +70 ˚C
SAB-C165-LF	Q67121-C941	P-TQFP-100-3	16-bit microcontroller with 2 KByte RAM Temperature range 0 to +70 ˚C

Note: The ordering codes (Q67121-D...) for the Mask-ROM versions are defined for each product after verification of the respective ROM code.

Pin Configuration TQFP Package

(top view)

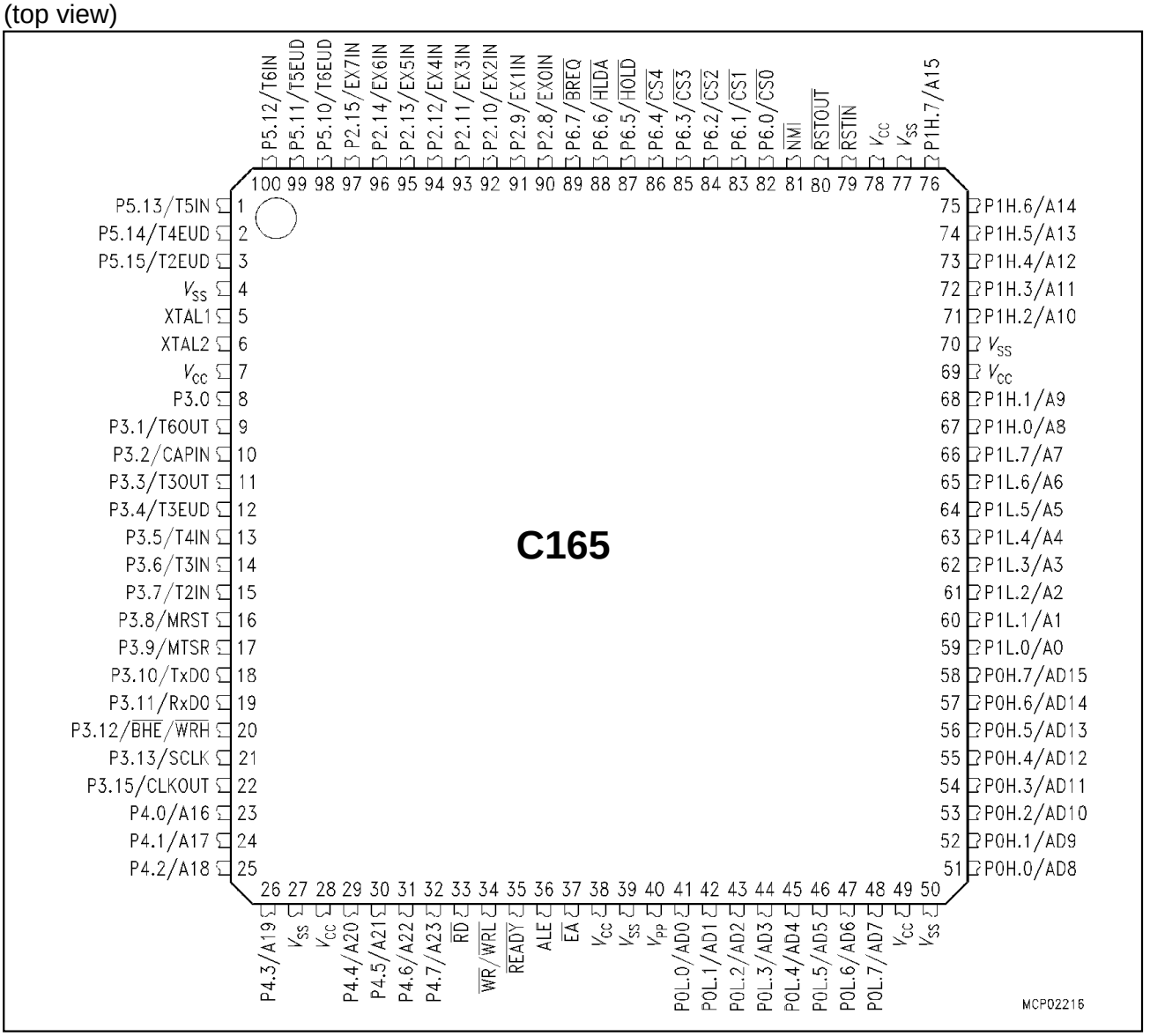


Figure 2

Pin Configuration MQFP Package
(top view)

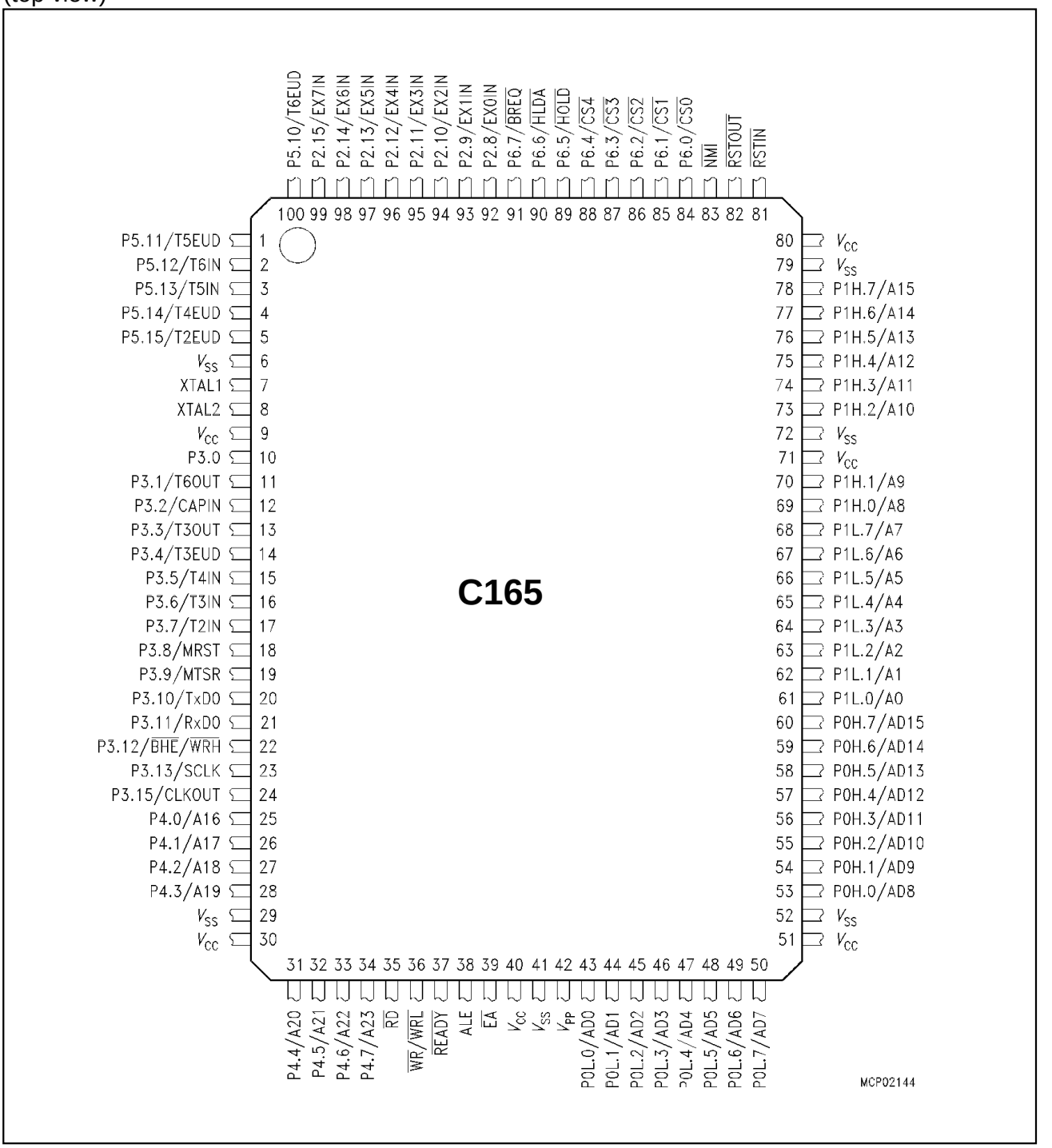


Figure 3

Pin Definitions and Functions

Symbol	Pin No.	Input (I) Output (O)	Function
P5.10 – P5.15	100 1 - 5 100 1 2 3 4 5	I I I I I I I I	Port 5 is a 6-bit input-only port with Schmitt-Trigger characteristics. The pins of Port 5 also serve as timer inputs: P5.10 T6EUD GPT2 Timer T6 Ext.Up/Down Ctrl.Input P5.11 T5EUD GPT2 Timer T5 Ext.Up/Down Ctrl.Input P5.12 T6IN GPT2 Timer T6 Count Input P5.13 T5IN GPT2 Timer T5 Count Input P5.14 T4EUD GPT1 Timer T4 Ext.Up/Down Ctrl.Input P5.15 T2EUD GPT1 Timer T2 Ext.Up/Down Ctrl.Input
XTAL1 XTAL2	7 8	I O	XTAL1: Input to the oscillator amplifier and input to the internal clock generator XTAL2: Output of the oscillator amplifier circuit. 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.
P3.0 – P3.13, P3.15	10 – 23, 24 11 12 13 14 15 16 17 18 19 20 21 22 23 24	I/O I/O I/O O I O I I I I I/O I/O O I/O O O I/O 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 following Port 3 pins also serve for alternate functions: P3.1 T6OUT GPT2 Timer T6 Toggle Latch Output P3.2 CAPIN GPT2 Register CAPREL Capture Input P3.3 T3OUT GPT1 Timer T3 Toggle Latch Output P3.4 T3EUD GPT1 Timer T3 Ext.Up/Down Ctrl.Input P3.5 T4IN GPT1 Timer T4 Input for Count/Gate/Reload/Capture P3.6 T3IN GPT1 Timer T3 Count/Gate Input P3.7 T2IN GPT1 Timer T2 Input for Count/Gate/Reload/Capture P3.8 MRST SSC Master-Rec./Slave-Transmit I/O P3.9 MTSR SSC Master-Transmit/Slave-Rec. O/I P3.10 T×D0 ASC0 Clock/Data Output (Asyn./Syn.) P3.11 R×D0 ASC0 Data Input (Asyn.) or I/O (Syn.) P3.12 $\overline{BHE}$ Ext. Memory High Byte Enable Signal, $\overline{WRH}$ Ext. Memory High Byte Write Strobe P3.13 SCLK SSC Master Clock Outp./Slave Cl. Inp. P3.15 CLKOUT System Clock Output (=CPU Clock)

Pin Definitions and Functions (cont'd)

Symbol	Pin No.	Input (I) Output (O)	Function
P4.0 – P4.7	25 - 28, 31 - 34	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: P4.0 A16 Least Significant Segment Addr. Line ... ... P4.7 A23 Most Significant Segment Addr. Line
	25 ... 34	O ... O	
$\overline{RD}$	35	O	External Memory Read Strobe. $\overline{RD}$ is activated for every external instruction or data read access.
$\overline{WR}$/ $\overline{WRL}$	36	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.
$\overline{READY}$	37	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	38	O	Address Latch Enable Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.
$\overline{EA}$	39	I	External Access Enable pin. A low level at this pin during and after Reset forces the C165 to begin instruction execution out of external memory. A high level forces execution out of the internal ROM. The C165 must have this pin tied to '0'.

Pin Definitions and Functions (cont'd)

Symbol	Pin No.	Input (I) Output (O)	Function
PORT0: P0L.0 – P0L.7, P0H.0 - P0H.7	43 – 50 53 – 60	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. 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. **Demultiplexed bus modes:** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: D0 – D7 D0 - D7 P0H.0 – P0H.7: I/O D8 - D15 **Multiplexed bus modes:** Data Path Width: 8-bit 16-bit P0L.0 – P0L.7: AD0 – AD7 AD0 - AD7 P0H.0 – P0H.7: A8 - A15 AD8 - AD15
PORT1: P1L.0 – P1L.7, P1H.0 - P1H.7	61 - 68 69 - 70, 73 - 78	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.
$\overline{\text{RSTIN}}$	81	I	Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the C165. An internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
$\overline{\text{RSTOUT}}$	82	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.
$\overline{\text{NMI}}$	83	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 C165 to go into power down mode. If $\overline{\text{NMI}}$ is high, when PWRDN is executed, the part will continue to run in normal mode. If not used, pin $\overline{\text{NMI}}$ should be pulled high externally.

Pin Definitions and Functions (cont'd)

Symbol	Pin No.	Input (I) Output (O)	Function
P6.0 – P6.7	84 - 91	I/O	Port 6 is an 8-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. Port 6 outputs can be configured as push/ pull or open drain drivers. The following Port 6 pins also serve for alternate functions: P6.0 $\overline{CS0}$ Chip Select 0 Output ... ... P6.4 $\overline{CS4}$ Chip Select 4 Output P6.5 $\overline{HOLD}$ External Master Hold Request Input P6.6 $\overline{HLDA}$ Hold Acknowledge Output P6.7 $\overline{BREQ}$ Bus Request Output
	84	O	
	...	...	
	88	O	
	89	I	
	90	O	
	91	O	
P2.8 – P2.15	92 - 99	I/O	Port 2 is an 8-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. Port 2 outputs can be configured as push/ pull or open drain drivers. The following Port 2 pins also serve for alternate functions: P2.8 EX0IN Fast External Interrupt 0 Input ... ... P2.15 EX7IN Fast External Interrupt 7 Input
	92	I	
	...	...	
	99	I	
V_{PP}	42	-	Flash programming voltage. This pin accepts the programming voltage for flash versions of the C165. **Note:** This pin is not connected (**NC**) on non-flash versions.
V_{CC}	9, 30, 40, 51, 71, 80	-	Digital Supply Voltage: + 5 V during normal operation and idle mode. ≥ 2.5 V during power down mode
V_{SS}	6, 29, 41, 52, 72, 79	-	Digital Ground.

Functional Description

The architecture of the C165 combines advantages of both RISC and CISC processors and of advanced peripheral subsystems in a very well-balanced way. The following block diagram gives an overview of the different on-chip components and of the advanced, high bandwidth internal bus structure of the C165.

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

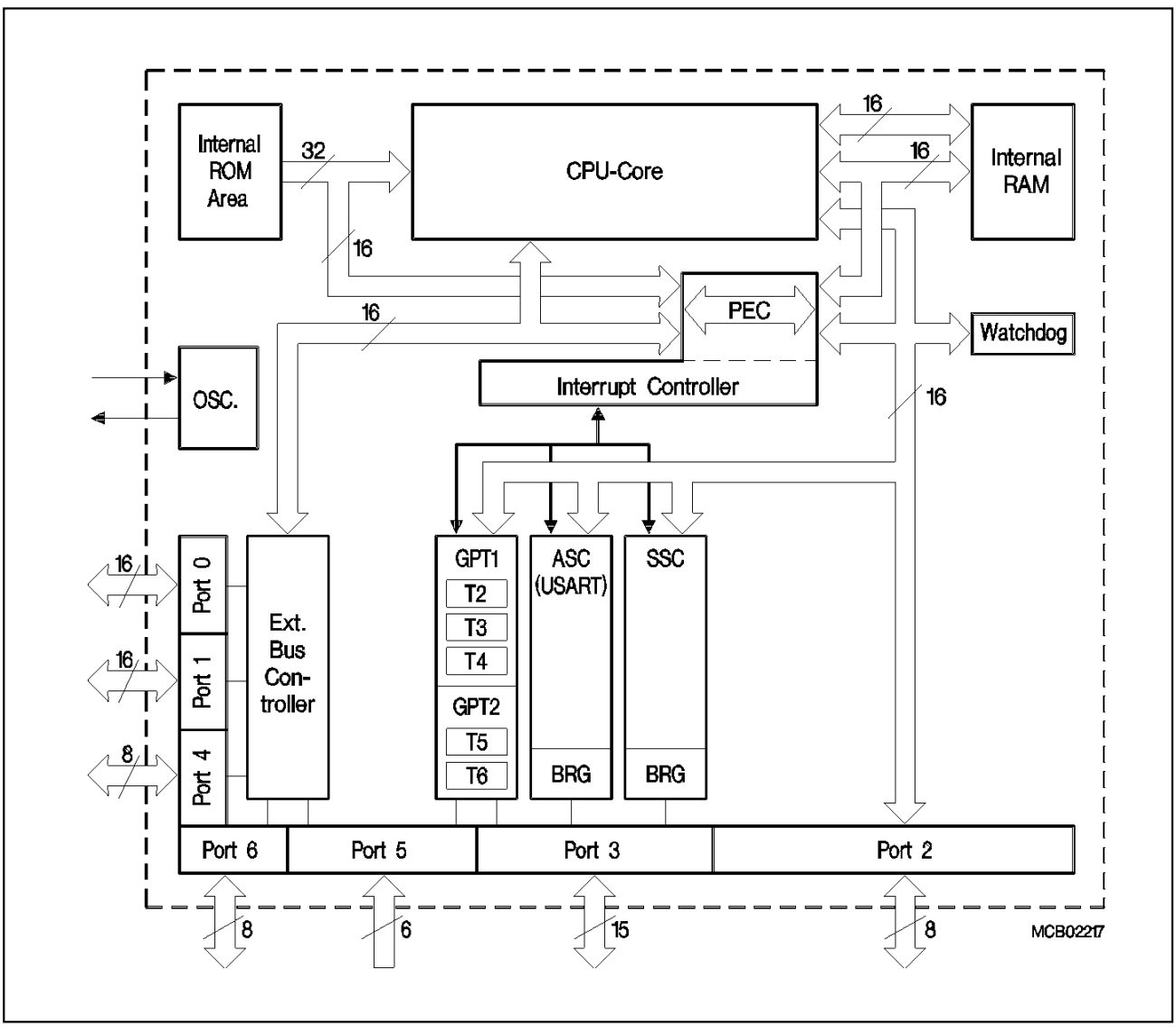


Figure 4
Block Diagram

Memory Organization

The memory space of the C165 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 bit addressable.

The C165 is prepared to incorporate on-chip mask-programmable ROM for code or constant data. Currently no ROM is integrated.

2 KBytes of on-chip 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 C165 family.

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

External Bus Controller

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

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

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

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

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

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

Important timing characteristics of the external bus interface (Memory Cycle Time, Memory Tri-State Time, Length of ALE and Read Write Delay) have been made programmable to allow the user the adaption of a wide range of different types of memories. In addition, different address ranges may be accessed with different bus characteristics. Up to 5 external $\overline{CS}$ signals can be generated in order to save external glue logic. Access to very slow memories is supported via a particular ‘Ready’ function. A $\overline{HOLD}$/$\overline{HLDA}$ protocol is available for bus arbitration.

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

Central Processing Unit (CPU)

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

Based on these hardware provisions, most of the C165's instructions can be executed in just one machine cycle which requires 100 ns at 20-MHz CPU clock. For example, shift and rotate instructions are always processed during one machine cycle independent of the number of bits to be shifted. All multiple-cycle instructions have been optimized so that they can be executed very fast as well: branches in 2 cycles, a 16 × 16 bit multiplication in 5 cycles and a 32-/16 bit division in 10 cycles. Another pipeline optimization, the so-called 'Jump Cache', allows reducing the execution time of repeatedly performed jumps in a loop from 2 cycles to 1 cycle.

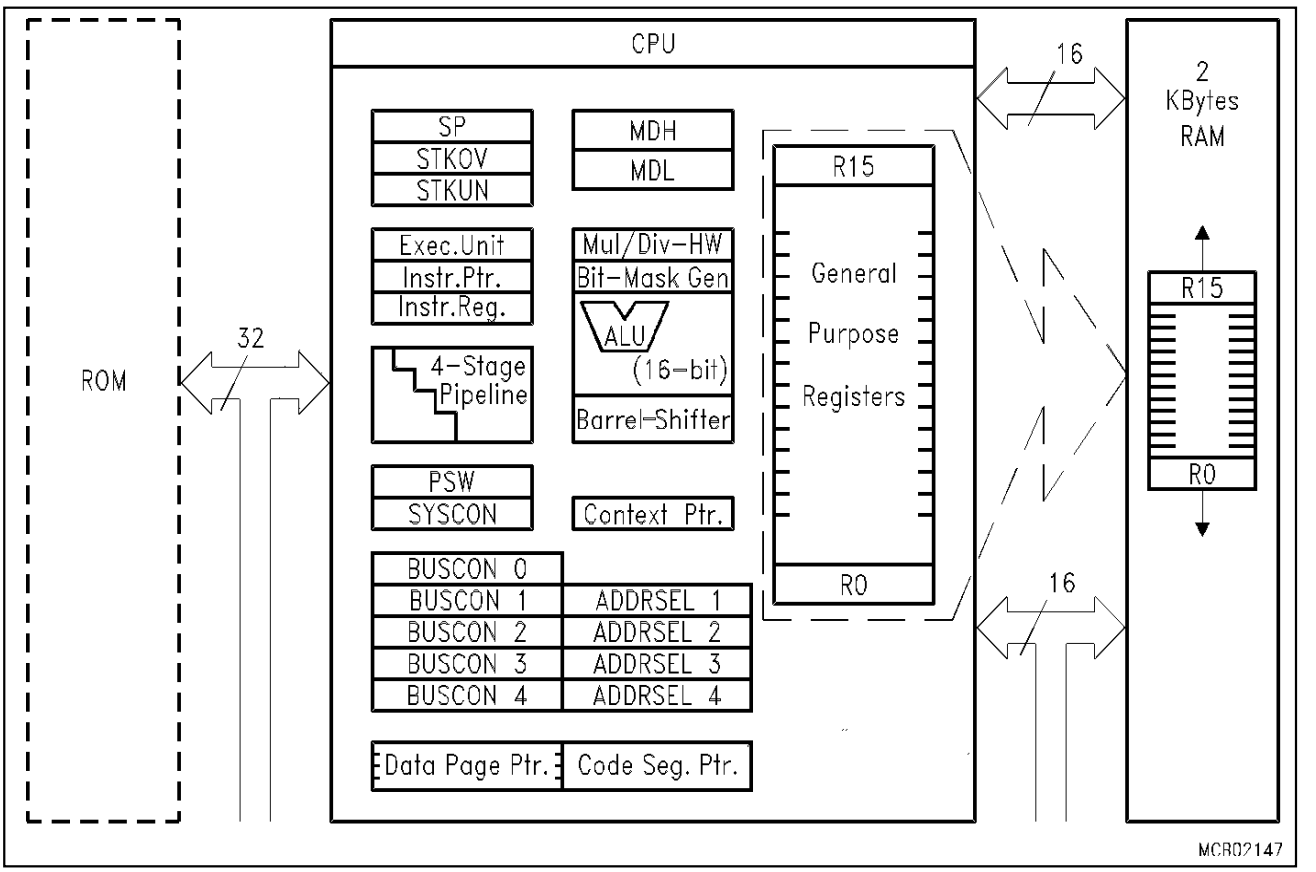


Figure 5
CPU Block Diagram

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

A system stack of up to 2048 bytes is provided as a storage for temporary data. The system stack is allocated in the on-chip RAM area, and it is accessed by the CPU via the stack pointer (SP) register. Two separate SFRs, STKOV and STKUN, are implicitly compared against the stack pointer value upon each stack access for the detection of a stack overflow or underflow.

The high performance offered by the hardware implementation of the CPU can efficiently be utilized by a programmer via the highly efficient C165 instruction set which includes the following instruction classes:

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

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

Interrupt System

With an interrupt response time within a range from just 250 ns to 600 ns (in case of internal program execution), the C165 is capable of reacting very fast to the occurrence of non-deterministic events.

The architecture of the C165 supports several mechanisms for fast and flexible response to service requests that can be generated from various sources internal or external to the microcontroller. Any of these interrupt requests can be programmed to being serviced by the Interrupt Controller or by the Peripheral Event Controller (PEC).

In contrast to a standard interrupt service where the current program execution is suspended and a branch to the interrupt vector table is performed, just one cycle is ‘stolen’ from the current CPU activity to perform a PEC service. A PEC service implies a single byte or word data transfer between any two memory locations with an additional increment of either the PEC source or the destination pointer. An individual PEC transfer counter is implicity decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited, for example, for supporting the transmission or reception of blocks of data. The C165 has 8 PEC channels each of which offers such fast interrupt-driven data transfer capabilities.

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

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

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

The following table shows all of the possible C165 interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers:

Note: Four nodes in the table (X-Peripheral nodes) are prepared to accept interrupt requests from integrated X-Bus peripherals. Nodes, where no X-Peripherals are connected, may be used to generate software controlled interrupt requests by setting the respective XPnIR bit. Also the three listed Software Nodes can be used for this purpose.

Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
External Interrupt 0	CC8IR	CC8IE	CC8INT	$00'0060_H$	18_H
External Interrupt 1	CC9IR	CC9IE	CC9INT	$00'0064_H$	19_H
External Interrupt 2	CC10IR	CC10IE	CC10INT	$00'0068_H$	$1A_H$
External Interrupt 3	CC11IR	CC11IE	CC11INT	$00'006C_H$	$1B_H$
External Interrupt 4	CC12IR	CC12IE	CC12INT	$00'0070_H$	$1C_H$
External Interrupt 5	CC13IR	CC13IE	CC13INT	$00'0074_H$	$1D_H$
External Interrupt 6	CC14IR	CC14IE	CC14INT	$00'0078_H$	$1E_H$
External Interrupt 7	CC15IR	CC15IE	CC15INT	$00'007C_H$	$1F_H$
GPT1 Timer 2	T2IR	T2IE	T2INT	$00'0088_H$	22_H
GPT1 Timer 3	T3IR	T3IE	T3INT	$00'008C_H$	23_H
GPT1 Timer 4	T4IR	T4IE	T4INT	$00'0090_H$	24_H
GPT2 Timer 5	T5IR	T5IE	T5INT	$00'0094_H$	25_H
GPT2 Timer 6	T6IR	T6IE	T6INT	$00'0098_H$	26_H
GPT2 CAPREL Register	CRIR	CRIE	CRINT	$00'009C_H$	27_H
ASC0 Transmit	S0TIR	S0TIE	S0TINT	$00'00A8_H$	$2A_H$
ASC0 Transmit Buffer	S0TBIR	S0TBIE	S0TBINT	$00'011C_H$	47_H
ASC0 Receive	S0RIR	S0RIE	S0RINT	$00'00AC_H$	$2B_H$
ASC0 Error	S0EIR	S0EIE	S0EINT	$00'00B0_H$	$2C_H$
SSC Transmit	SCTIR	SCTIE	SCTINT	$00'00B4_H$	$2D_H$
SSC Receive	SCRIR	SCRIE	SCRINT	$00'00B8_H$	$2E_H$
SSC Error	SCEIR	SCEIE	SCEINT	$00'00BC_H$	$2F_H$
X-Peripheral Node 0	XP0IR	XP0IE	XP0INT	$00'0100_H$	40_H
X-Peripheral Node 1	XP1IR	XP1IE	XP1INT	$00'0104_H$	41_H
X-Peripheral Node 2	XP2IR	XP2IE	XP2INT	$00'0108_H$	42_H
X-Peripheral Node 3	XP3IR	XP3IE	XP3INT	$00'010C_H$	43_H
Software Node	CC29IR	CC29IE	CC29INT	$00'0110_H$	44_H
Software Node	CC30IR	CC30IE	CC30INT	$00'0114_H$	45_H
Software Node	CC31IR	CC31IE	CC31INT	$00'0118_H$	46_H

The C165 also provides an excellent mechanism to identify and to process exceptions or error conditions that arise during run-time, so-called ‘Hardware Traps’. Hardware traps cause immediate non-maskable system reaction which is similar to a standard interrupt service (branching to a dedicated vector table location). The occurrence of a hardware trap is additionally signified by an individual bit in the trap flag register (TFR). Except when another higher prioritized trap service is in progress, a hardware trap will interrupt any actual program execution. In turn, hardware trap services can normally not be interrupted by standard or PEC interrupts.

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

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

General Purpose Timer (GPT) Unit

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

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

Each of the three timers T2, T3, T4 of module GPT1 can be configured individually for one of three basic modes of operation, which are Timer, Gated Timer, and Counter Mode. In Timer Mode, the input clock for a timer is derived from the CPU clock, divided by a programmable prescaler, while Counter Mode allows a timer to be clocked in reference to external events.
Pulse width or duty cycle measurement is supported in Gated Timer Mode, where the operation of a timer is controlled by the ‘gate’ level on an external input pin. For these purposes, each timer has one associated port pin (TxIN) which serves as gate or clock input. The maximum resolution of the timers in module GPT1 is 400 ns (@ 20-MHz CPU clock).

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

Timers T3 and T4 have output toggle latches (TxOTL) which change their state on each timer overflow/underflow. The state of these latches may be output on port pins (TxOUT) e.g. for time out monitoring of external hardware components, or may be used internally to clock timers T2 and T4 for measuring long time periods with high resolution.

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

With its maximum resolution of 200 ns (@ 20 MHz), the GPT2 module provides precise event control and time measurement. It includes two timers (T5, T6) and a capture/reload register (CAPREL). Both timers can be clocked with an input clock which is derived from the CPU clock via a programmable prescaler or with external signals. The count direction (up/down) for each timer is programmable by software or may additionally be altered dynamically by an external signal on a port pin (TxEUD). Concatenation of the timers is supported via the output toggle latch (T6OTL) of timer T6, which changes its state on each timer overflow/underflow.

The state of this latch may be used to clock timer T5, or it may be output on a port pin (T6OUT). The overflows/underflows of timer T6 can additionally be used to clock the CAPCOM timers T0 or T1, and to cause a reload from the CAPREL register. The CAPREL register may capture the contents of timer T5 based on an external signal transition on the corresponding port pin (CAPIN), and timer T5 may optionally be cleared after the capture procedure. This allows absolute time differences to be measured or pulse multiplication to be performed without software overhead.

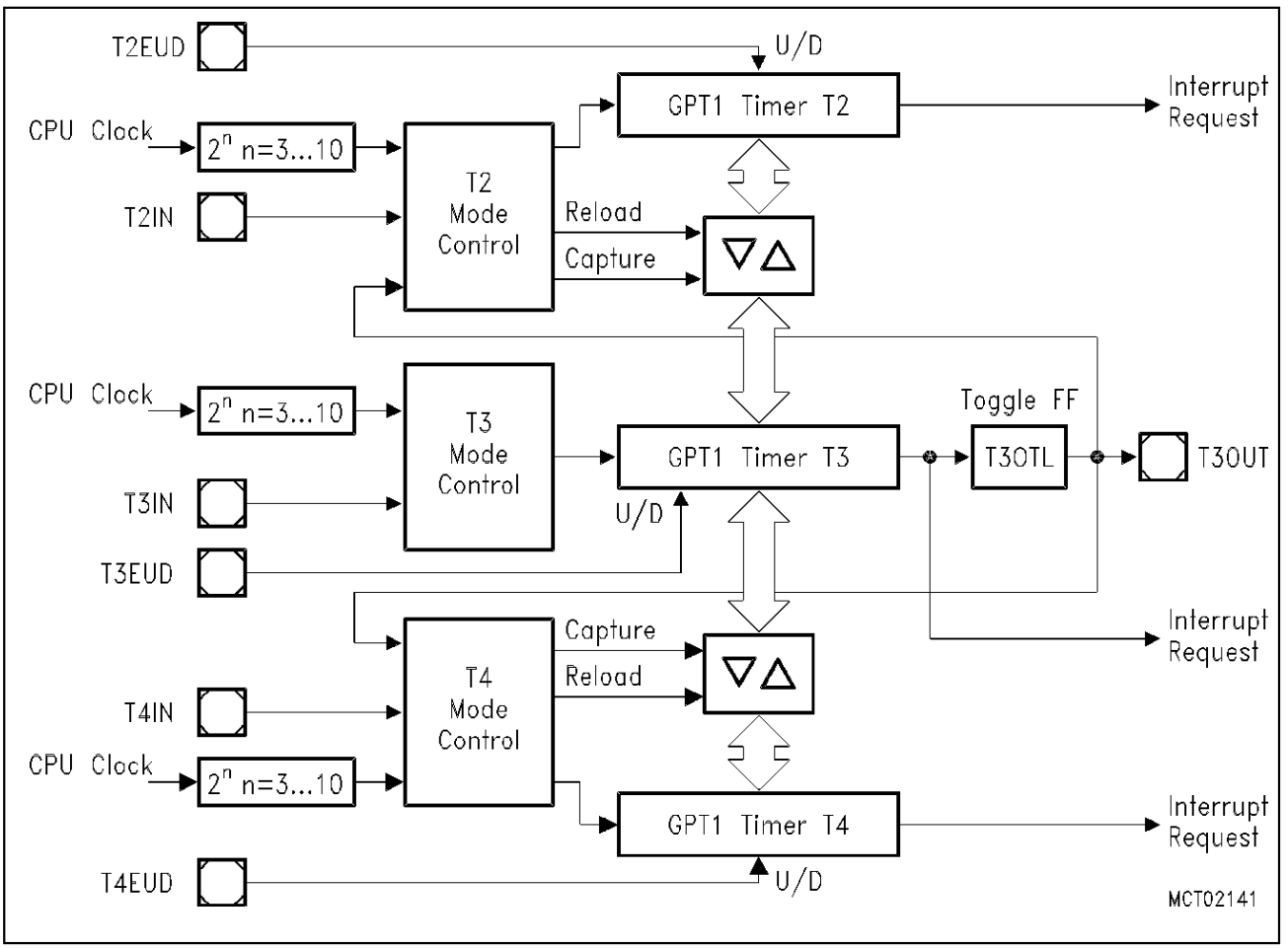


Figure 6
Block Diagram of GPT1

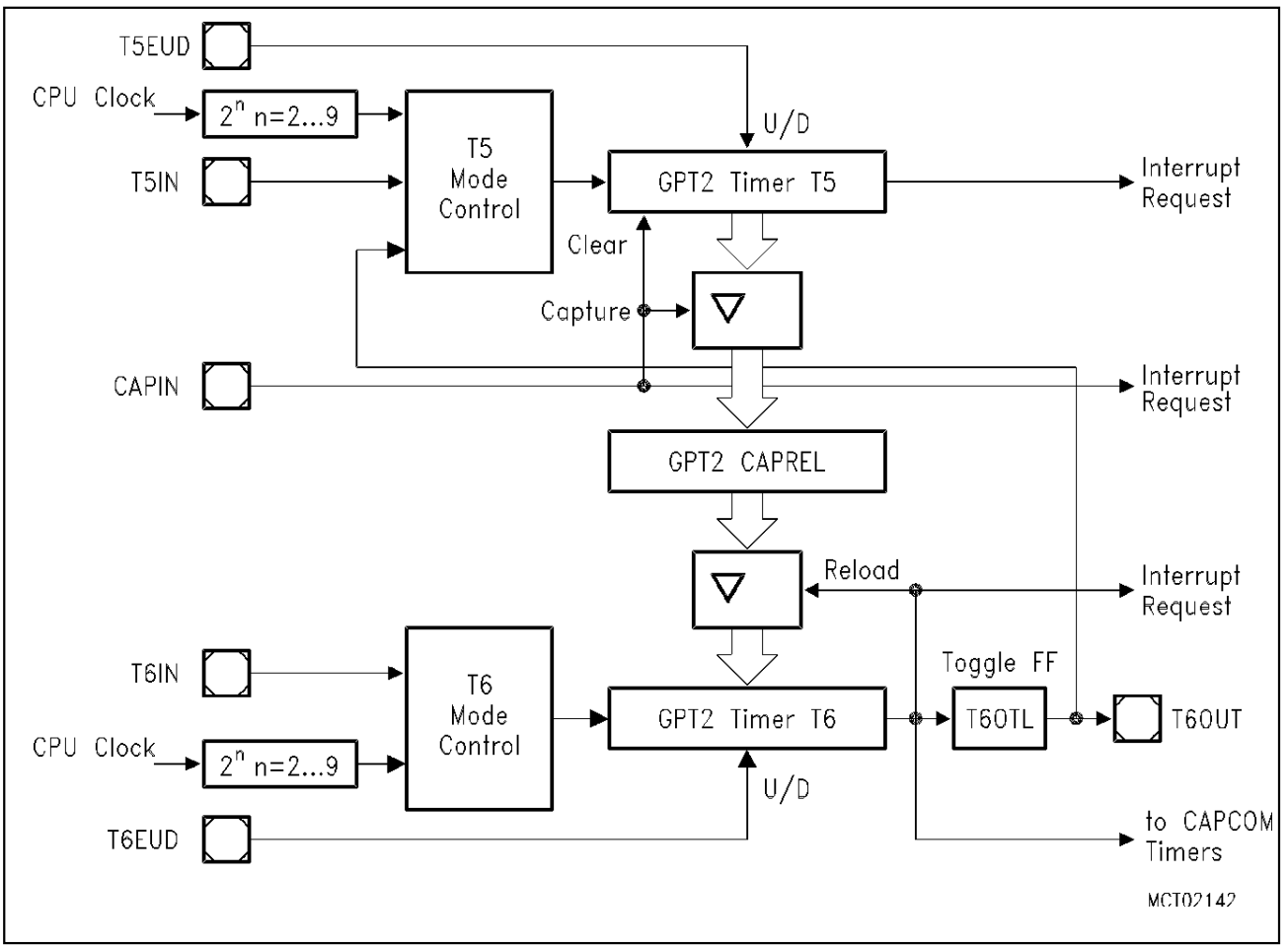


Figure 7
Block Diagram of GPT2

Parallel Ports

The C165 provides up to 77 I/O lines which are organized into six input/output ports and one input port. All port lines are bit-addressable, and all input/output lines are individually (bit-wise) programmable as inputs or outputs via direction registers. The I/O ports are true bidirectional ports which are switched to high impedance state when configured as inputs. The output drivers of three I/O ports can be configured (pin by pin) for push/pull operation or open-drain operation via control registers. During the internal reset, all port pins are configured as inputs.

All port lines have programmable alternate input or output functions associated with them. PORT0 and PORT1 may be used as address and data lines when accessing external memory, while Port 4 outputs the additional segment address bits A23/19/17...A16 in systems where segmentation is enabled to access more than 64 KBytes of memory. Port 6 provides optional bus arbitration signals ($\overline{\text{BREQ}}$, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$) and chip select signals. Port 3 includes alternate functions of timers, serial interfaces, the optional bus control signal $\overline{\text{BHE}}$ and the system clock output (CLKOUT). Port 5 is used for timer control signals. All port lines that are not used for these alternate functions may be used as general purpose I/O lines.

Serial Channels

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

They are upward compatible with the serial ports of the Siemens SAB 8051x microcontroller family and support full-duplex asynchronous communication up to 625 KBaud and half-duplex synchronous communication up to 5 Mbaud (2.5 Mbaud on the ASC0) @ 20-MHz system clock.

Two dedicated baud rate generators allow to set up all standard baud rates without oscillator tuning. For transmission, reception, and erroneous reception 3 separate interrupt vectors are provided for each serial channel.

In asynchronous mode, 8- or 9-bit data frames are transmitted or received, preceded by a start bit and terminated by one or two stop bits. For multiprocessor communication, a mechanism to distinguish address from data bytes has been included (8-bit data + wake up bit mode).
In synchronous mode, the ASC0 transmits or receives bytes (8 bits) synchronously to a shift clock which is generated by the ASC0. The SSC transmits or receives characters of 2...16 bits length synchronously to a shift clock which can be generated by the SSC (master mode) or by an external master (slave mode). The SSC can start shifting with the LSB or with the MSB, while the ASC0 always shifts the LSB first.
A loop back option is available for testing purposes.

A number of optional hardware error detection capabilities has been included to increase the reliability of data transfers. A parity bit can automatically be generated on transmission or be checked on reception. Framing error detection allows to recognize data frames with missing stop bits. An overrun error will be generated, if the last character received has not been read out of the receive buffer register at the time the reception of a new character is complete.

Watchdog Timer

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

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

The Watchdog Timer is a 16-bit timer, clocked with the system clock divided either by 2 or by 128. The high byte of the Watchdog Timer register can be set to a prespecified reload value (stored in WDTREL) in order to allow further variation of the monitored time interval. Each time it is serviced by the application software, the high byte of the Watchdog Timer is reloaded. Thus, time intervals between 25 µs and 420 ms can be monitored (@ 20 MHz). The default Watchdog Timer interval after reset is 6.55 ms (@ 20 MHz).

Instruction Set Summary

The table below lists the instructions of the C165 in a condensed way.
The various addressing modes that can be used with a specific instruction, the operation of the instructions, parameters for conditional execution of instructions, and the opcodes for each instruction can be found in the **“C16x Family Instruction Set Manual”**.

This document also provides a detailed description of each instruction.

Instruction Set Summary

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

Instruction Set Summary (cont'd)

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

Special Function Registers Overview

The following table lists all SFRs which are implemented in the C165 in alphabetical order. **Bit-addressable** SFRs are marked with the letter "**b**" in column "Name". SFRs within the **Extended SFR-Space** (ESFRs) are marked with the letter "**E**" in column "Physical Address".

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

Special Function Registers Overview

Name	Physical Address	8-Bit Address	Description	Reset Value
ADDRSEL1	$FE18_H$	$0C_H$	Address Select Register 1	0000_H
ADDRSEL2	$FE1A_H$	$0D_H$	Address Select Register 2	0000_H
ADDRSEL3	$FE1C_H$	$0E_H$	Address Select Register 3	0000_H
ADDRSEL4	$FE1E_H$	$0F_H$	Address Select Register 4	0000_H
BUSCON0 b	$FF0C_H$	86_H	Bus Configuration Register 0	$0XX0_H$
BUSCON1 b	$FF14_H$	$8A_H$	Bus Configuration Register 1	0000_H
BUSCON2 b	$FF16_H$	$8B_H$	Bus Configuration Register 2	0000_H
BUSCON3 b	$FF18_H$	$8C_H$	Bus Configuration Register 3	0000_H
BUSCON4 b	$FF1A_H$	$8D_H$	Bus Configuration Register 4	0000_H
CAPREL	$FE4A_H$	25_H	GPT2 Capture/Reload Register	0000_H
CC8IC b	$FF88_H$	$C4_H$	EX0IN Interrupt Control Register	0000_H
CC9IC b	$FF8A_H$	$C5_H$	EX1IN Interrupt Control Register	0000_H
CC10IC b	$FF8C_H$	$C6_H$	EX2IN Interrupt Control Register	0000_H
CC11IC b	$FF8E_H$	$C7_H$	EX3IN Interrupt Control Register	0000_H
CC12IC b	$FF90_H$	$C8_H$	EX4IN Interrupt Control Register	0000_H
CC13IC b	$FF92_H$	$C9_H$	EX5IN Interrupt Control Register	0000_H
CC14IC b	$FF94_H$	CA_H	EX6IN Interrupt Control Register	0000_H
CC15IC b	$FF96_H$	CB_H	EX7IN Interrupt Control Register	0000_H
CC29IC b	$F184_H$ **E**	$C2_H$	Software Node Interrupt Control Register	0000_H
CC30IC b	$F18C_H$ **E**	$C6_H$	Software Node Interrupt Control Register	0000_H
CC31IC b	$F194_H$ **E**	CA_H	Software Node Interrupt Control Register	0000_H
CP	$FE10_H$	08_H	CPU Context Pointer Register	$FC00_H$

Special Function Registers Overview (cont'd)

Name		Physical Address		8-Bit Address	Description	Reset Value
CRIC	**b**	$FF6A_H$		$B5_H$	GPT2 CAPREL Interrupt Control Register	0000_H
CSP		$FE08_H$		04_H	CPU Code Segment Pointer Register (read only)	0000_H
DP0L	**b**	$F100_H$	**E**	80_H	P0L Direction Control Register	00_H
DP0H	**b**	$F102_H$	**E**	81_H	P0H Direction Control Register	00_H
DP1L	**b**	$F104_H$	**E**	82_H	P1L Direction Control Register	00_H
DP1H	**b**	$F106_H$	**E**	83_H	P1H Direction Control Register	00_H
DP2	**b**	$FFC2_H$		$E1_H$	Port 2 Direction Control Register	0000_H
DP3	**b**	$FFC6_H$		$E3_H$	Port 3 Direction Control Register	0000_H
DP4	**b**	$FFCA_H$		$E5_H$	Port 4 Direction Control Register	00_H
DP6	**b**	$FFCE_H$		$E7_H$	Port 6 Direction Control Register	00_H
DPP0		$FE00_H$		00_H	CPU Data Page Pointer 0 Register (10 bits)	0000_H
DPP1		$FE02_H$		01_H	CPU Data Page Pointer 1 Register (10 bits)	0001_H
DPP2		$FE04_H$		02_H	CPU Data Page Pointer 2 Register (10 bits)	0002_H
DPP3		$FE06_H$		03_H	CPU Data Page Pointer 3 Register (10 bits)	0003_H
EXICON	**b**	$F1C0_H$	**E**	$E0_H$	External Interrupt Control Register	0000_H
MDC	**b**	$FF0E_H$		87_H	CPU Multiply Divide Control Register	0000_H
MDH		$FE0C_H$		06_H	CPU Multiply Divide Register – High Word	0000_H
MDL		$FE0E_H$		07_H	CPU Multiply Divide Register – Low Word	0000_H
ODP2	**b**	$F1C2_H$	**E**	$E1_H$	Port 2 Open Drain Control Register	0000_H
ODP3	**b**	$F1C6_H$	**E**	$E3_H$	Port 3 Open Drain Control Register	0000_H
ODP6	**b**	$F1CE_H$	**E**	$E7_H$	Port 6 Open Drain Control Register	00_H
ONES		$FF1E_H$		$8F_H$	Constant Value 1's Register (read only)	$FFFF_H$
P0L	**b**	$FF00_H$		80_H	Port 0 Low Register (Lower half of PORT0)	00_H
P0H	**b**	$FF02_H$		81_H	Port 0 High Register (Upper half of PORT0)	00_H
P1L	**b**	$FF04_H$		82_H	Port 1 Low Register (Lower half of PORT1)	00_H
P1H	**b**	$FF06_H$		83_H	Port 1 High Register (Upper half of PORT1)	00_H
P2	**b**	$FFC0_H$		$E0_H$	Port 2 Register	0000_H
P3	**b**	$FFC4_H$		$E2_H$	Port 3 Register	0000_H
P4	**b**	$FFC8_H$		$E4_H$	Port 4 Register (8 bits)	00_H

Special Function Registers Overview (cont'd)

Name		Physical Address	8-Bit Address	Description	Reset Value
P5	**b**	$FFA2_H$	$D1_H$	Port 5 Register (read only)	$XXXX_H$
P6	**b**	$FFCC_H$	$E6_H$	Port 6 Register (8 bits)	00_H
PECC0		$FEC0_H$	60_H	PEC Channel 0 Control Register	0000_H
PECC1		$FEC2_H$	61_H	PEC Channel 1 Control Register	0000_H
PECC2		$FEC4_H$	62_H	PEC Channel 2 Control Register	0000_H
PECC3		$FEC6_H$	63_H	PEC Channel 3 Control Register	0000_H
PECC4		$FEC8_H$	64_H	PEC Channel 4 Control Register	0000_H
PECC5		$FECA_H$	65_H	PEC Channel 5 Control Register	0000_H
PECC6		$FECC_H$	66_H	PEC Channel 6 Control Register	0000_H
PECC7		$FECE_H$	67_H	PEC Channel 7 Control Register	0000_H
PSW	**b**	$FF10_H$	88_H	CPU Program Status Word	0000_H
RP0H	**b**	$F108_H$ **E**	84_H	System Startup Configuration Register (Rd. only)	XX_H
S0BG		$FEB4_H$	$5A_H$	Serial Channel 0 Baud Rate Generator Reload Register	0000_H
S0CON	**b**	$FFB0_H$	$D8_H$	Serial Channel 0 Control Register	0000_H
S0EIC	**b**	$FF70_H$	$B8_H$	Serial Channel 0 Error Interrupt Control Register	0000_H
S0RBUF		$FEB2_H$	59_H	Serial Channel 0 Receive Buffer Register (read only)	XX_H
S0RIC	**b**	$FF6E_H$	$B7_H$	Serial Channel 0 Receive Interrupt Control Register	0000_H
S0TBIC	**b**	$F19C_H$ **E**	CE_H	Serial Channel 0 Transmit Buffer Interrupt Control Register	0000_H
S0TBUF		$FEB0_H$	58_H	Serial Channel 0 Transmit Buffer Register (write only)	00_H
S0TIC	**b**	$FF6C_H$	$B6_H$	Serial Channel 0 Transmit Interrupt Control Register	0000_H
SP		$FE12_H$	09_H	CPU System Stack Pointer Register	$FC00_H$
SSCBR		$F0B4_H$ **E**	$5A_H$	SSC Baudrate Register	0000_H
SSCCON	**b**	$FFB2_H$	$D9_H$	SSC Control Register	0000_H
SSCEIC	**b**	$FF76_H$	BB_H	SSC Error Interrupt Control Register	0000_H

Special Function Registers Overview (cont'd)

Name		Physical Address		8-Bit Address	Description	Reset Value
SSCRB		$F0B2_H$	**E**	59_H	SSC Receive Buffer (read only)	$XXXX_H$
SSCRIC	**b**	$FF74_H$		BA_H	SSC Receive Interrupt Control Register	0000_H
SSCTB		$F0B0_H$	**E**	58_H	SSC Transmit Buffer (write only)	0000_H
SSCTIC	**b**	$FF72_H$		$B9_H$	SSC Transmit Interrupt Control Register	0000_H
STKOV		$FE14_H$		$0A_H$	CPU Stack Overflow Pointer Register	$FA00_H$
STKUN		$FE16_H$		$0B_H$	CPU Stack Underflow Pointer Register	$FC00_H$
SYSCON	**b**	$FF12_H$		89_H	CPU System Configuration Register	$0xx0_H$*)
T2		$FE40_H$		20_H	GPT1 Timer 2 Register	0000_H
T2CON	**b**	$FF40_H$		$A0_H$	GPT1 Timer 2 Control Register	0000_H
T2IC	**b**	$FF60_H$		$B0_H$	GPT1 Timer 2 Interrupt Control Register	0000_H
T3		$FE42_H$		21_H	GPT1 Timer 3 Register	0000_H
T3CON	**b**	$FF42_H$		$A1_H$	GPT1 Timer 3 Control Register	0000_H
T3IC	**b**	$FF62_H$		$B1_H$	GPT1 Timer 3 Interrupt Control Register	0000_H
T4		$FE44_H$		22_H	GPT1 Timer 4 Register	0000_H
T4CON	**b**	$FF44_H$		$A2_H$	GPT1 Timer 4 Control Register	0000_H
T4IC	**b**	$FF64_H$		$B2_H$	GPT1 Timer 4 Interrupt Control Register	0000_H
T5		$FE46_H$		23_H	GPT2 Timer 5 Register	0000_H
T5CON	**b**	$FF46_H$		$A3_H$	GPT2 Timer 5 Control Register	0000_H
T5IC	**b**	$FF66_H$		$B3_H$	GPT2 Timer 5 Interrupt Control Register	0000_H
T6		$FE48_H$		24_H	GPT2 Timer 6 Register	0000_H
T6CON	**b**	$FF48_H$		$A4_H$	GPT2 Timer 6 Control Register	0000_H
T6IC	**b**	$FF68_H$		$B4_H$	GPT2 Timer 6 Interrupt Control Register	0000_H
TFR	**b**	$FFAC_H$		$D6_H$	Trap Flag Register	0000_H
WDT		$FEAE_H$		57_H	Watchdog Timer Register (read only)	0000_H
WDTCON		$FFAE_H$		$D7_H$	Watchdog Timer Control Register	0000_H
XP0IC	**b**	$F186_H$	**E**	$C3_H$	X-Peripheral 0 Interrupt Control Register	0000_H
XP1IC	**b**	$F18E_H$	**E**	$C7_H$	X-Peripheral 1 Interrupt Control Register	0000_H
XP2IC	**b**	$F196_H$	**E**	CB_H	X-Peripheral 2 Interrupt Control Register	0000_H

Special Function Registers Overview (cont'd)

Name		Physical Address	8-Bit Address	Description	Reset Value
XP3IC	**b**	$F19E_H$ **E**	CF_H	X-Peripheral 3 Interrupt Control Register	0000_H
ZEROS	**b**	$FF1C_H$	$8E_H$	Constant Value 0's Register (read only)	0000_H

*) The system configuration is selected during reset.

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

Absolute Maximum Ratings

Ambient temperature under bias (T_A):
SAB-C165-LM, SAB-C165-RM, SAB-C165-LF, SAB-C165-RF ..... 0 to + 70 °C
SAF-C165-LM ..... – 40 to + 85 °C
Storage temperature (T_{ST}) ..... – 65 to + 150 °C
Voltage on V_{CC} pins with respect to ground (V_{SS}) ..... –0.5 to + 6.5 V
Voltage on any pin with respect to ground (V_{SS}) ..... – 0.5 to V_{CC} + 0.5 V
Input current on any pin during overload condition ..... – 10 to + 10 mA
Absolute sum of all input currents during overload condition ..... |100 mA|
Power dissipation ..... 1.5 W

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

Parameter Interpretation

The parameters listed in the following partly represent the characteristics of the C165 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 C165 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 C165.

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-C165-LM, SAB-C165-RM, SAB-C165-LF, SAB-C165-RF
T_A = -40 to +85 °C for SAF-C165-LM

Parameter	Symbol		Limit Values		Unit	Test Condition
			min.	max.		
Input low voltage	V_{IL}	SR	– 0.5	0.2 V_{CC} – 0.1	V	–
Input high voltage (all except $\overline{\text{RSTIN}}$ and XTAL1)	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	–

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Output low voltage (PORT0, PORT1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT, $\overline{RSTOUT}$)	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, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT, $\overline{RSTOUT}$)	V_{OH} CC	0.9 V_{CC} 2.4	–	V	I_{OH} = – 500 µA I_{OH} = – 2.4 mA
Output high voltage [1] (all other outputs)	V_{OH1} CC	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 250 µA I_{OH} = – 1.6 mA
Input leakage current (Port 5)	I_{OZ1} CC	–	±200	nA	0 V < V_{IN} < V_{CC}
Input leakage current (all other)	I_{OZ2} CC	–	±500	nA	0 V < V_{IN} < V_{CC}
$\overline{RSTIN}$ pullup resistor	R_{RST} CC	50	150	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}
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 T_A = 25 ˚C
Power supply current	I_{CC}	–	10 + 4 * f_{CPU}	mA	$\overline{RSTIN}$ = V_{IL2} f_{CPU} in [MHz] [6]
Idle mode supply current	I_{ID}	–	2 + 1.2 * f_{CPU}	mA	$\overline{RSTIN}$ = V_{IH1} f_{CPU} in [MHz] [6]
Power-down mode supply current	I_{PD}	–	100	µA	V_{CC} = 5.5 V [7]

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 characterization.

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.

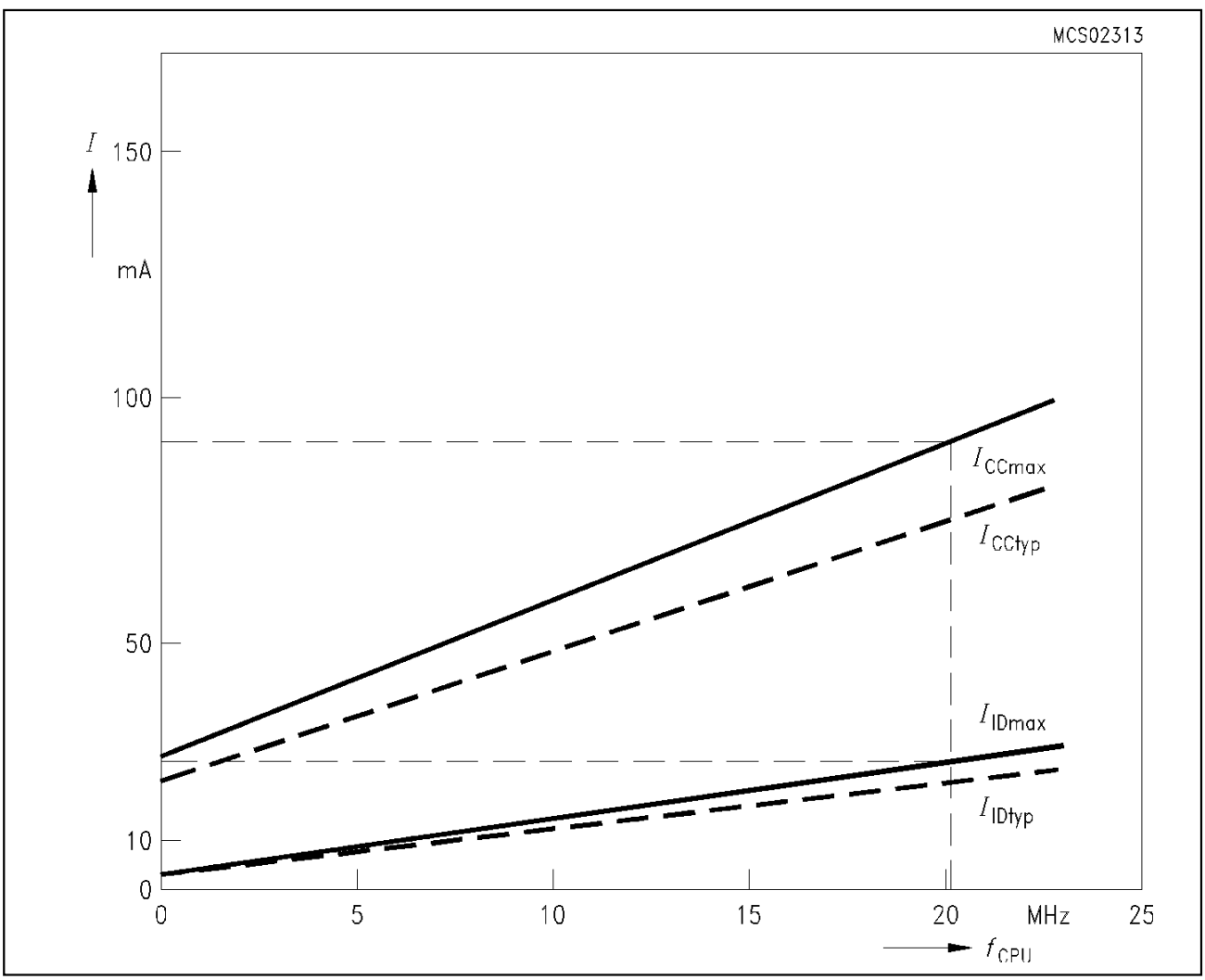
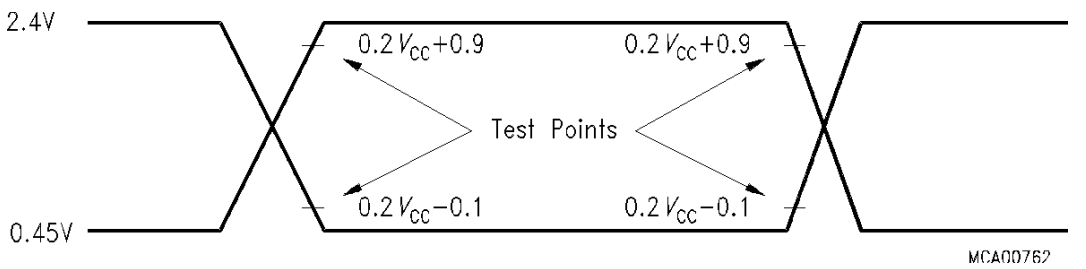


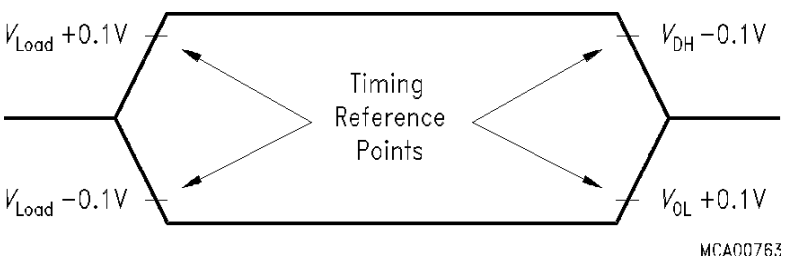
Figure 8
Supply/Idle Current as a Function of Operating Frequency

Testing Waveforms



AC inputs during testing are driven at 2.4 V for a logic ‘1’ and 0.4 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 9
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 10
Float Waveforms

AC Characteristics

External Clock Drive XTAL1

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 ˚C for SAB-C165-LM, SAB-C165-RM, SAB-C165-LF, SAB-C165-RF
T_A = -40 to +85 ˚C for SAF-C165-LM

Parameter	Symbol		Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
Oscillator period	TCL	SR	25	25	25	500	ns
High time	t_1	SR	6	–	6	–	ns
Low time	t_2	SR	6	–	6	–	ns
Rise time	t_3	SR	–	5	–	5	ns
Fall time	t_4	SR	–	5	–	5	ns

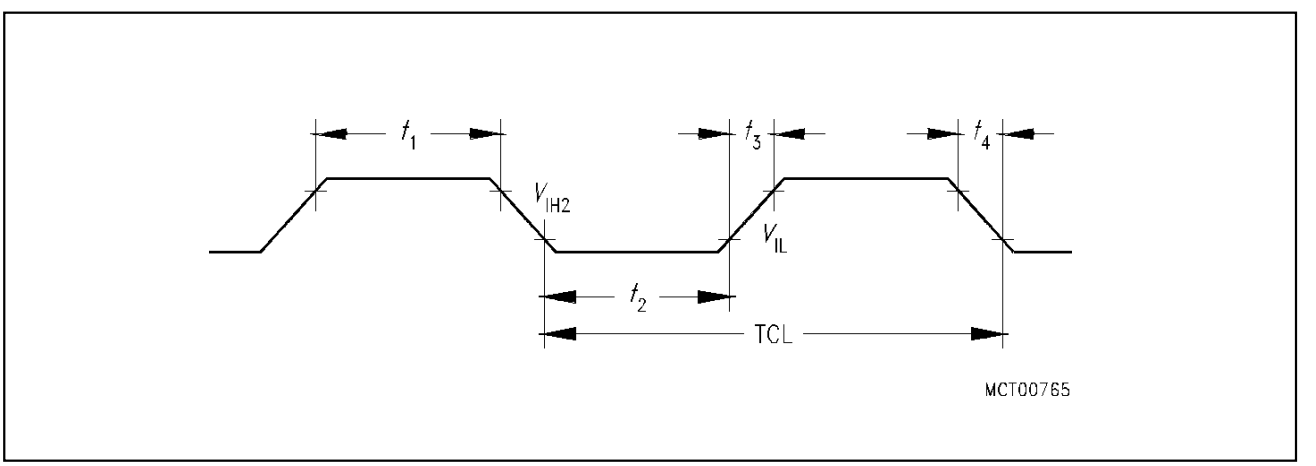


Figure 11
External Clock Drive XTAL1

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 (cont'd)

Multiplexed Bus

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB-C165-LM, SAB-C165-RM, SAB-C165-LF, SAB-C165-RF
T_A = -40 to +85 °C for SAF-C165-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 + 2t_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
$\overline{RD}$ 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 + 2t_A + t_C	–	4TCL - 30 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}	SR	0	–	0	–	ns
Data float after $\overline{RD}$	t_{19}	SR	–	35 + t_F	–	2TCL - 15 + t_F	ns
Data valid to $\overline{WR}$	t_{22}	SR	35 + t_C	–	2TCL - 15 + t_C	–	ns

Parameter	Symbol		Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
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 $\overline{RD}$, $\overline{WR}$	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 + 2t_A	–	3TCL - 20 + t_C + 2t_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
$\overline{RdCS}$ 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 $\overline{RdCS}$	t_{52}	SR	–	30 + t_F	–	2TCL - 20 + t_F	ns
Address hold after $\overline{RdCS}$, $\overline{WrCS}$	t_{54}	CC	30 + t_F	–	2TCL - 20 + t_F	–	ns
Data hold after $\overline{WrCS}$	t_{56}	CC	30 + t_F	–	2TCL - 20 + t_F	–	ns

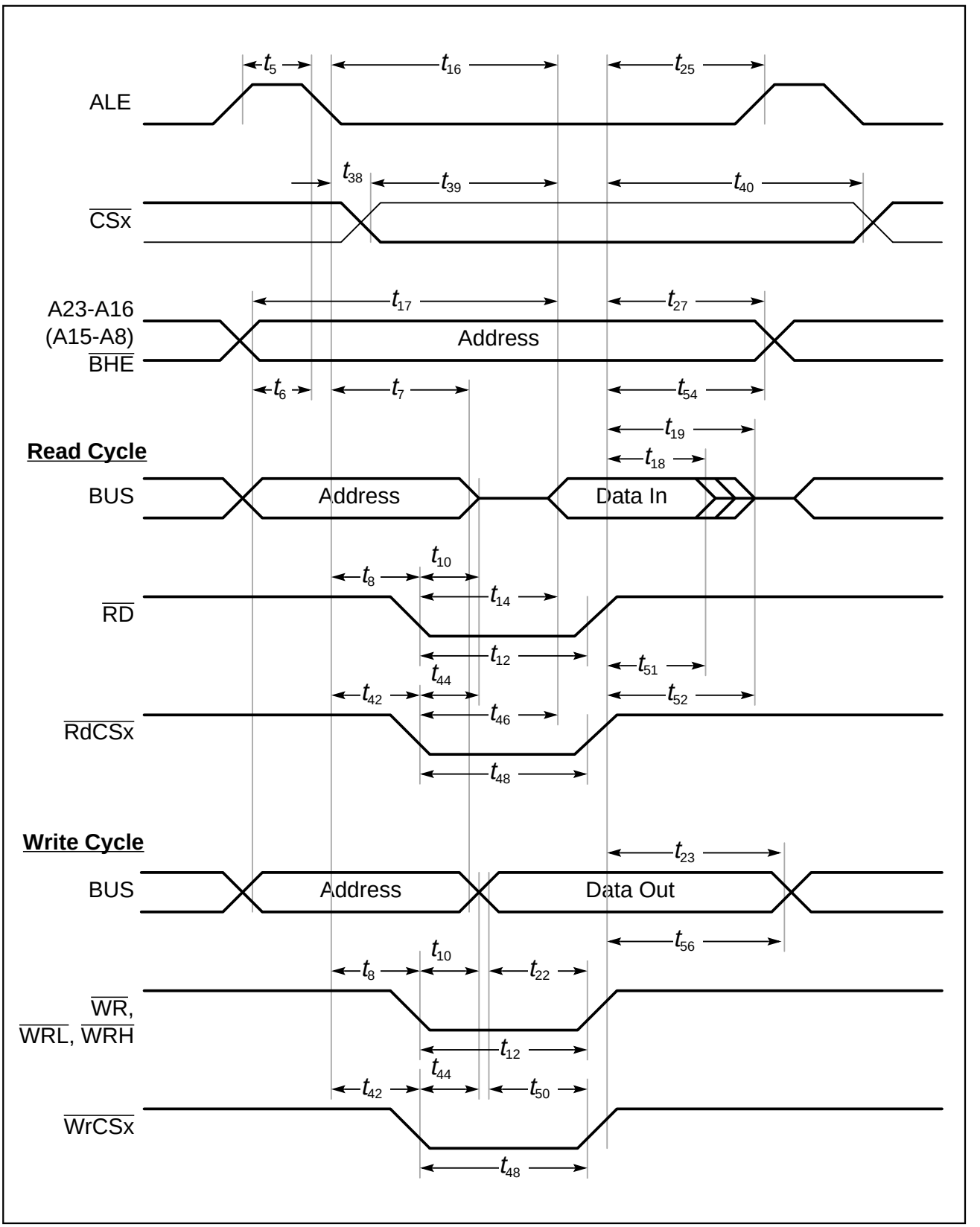


Figure 12-1
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE

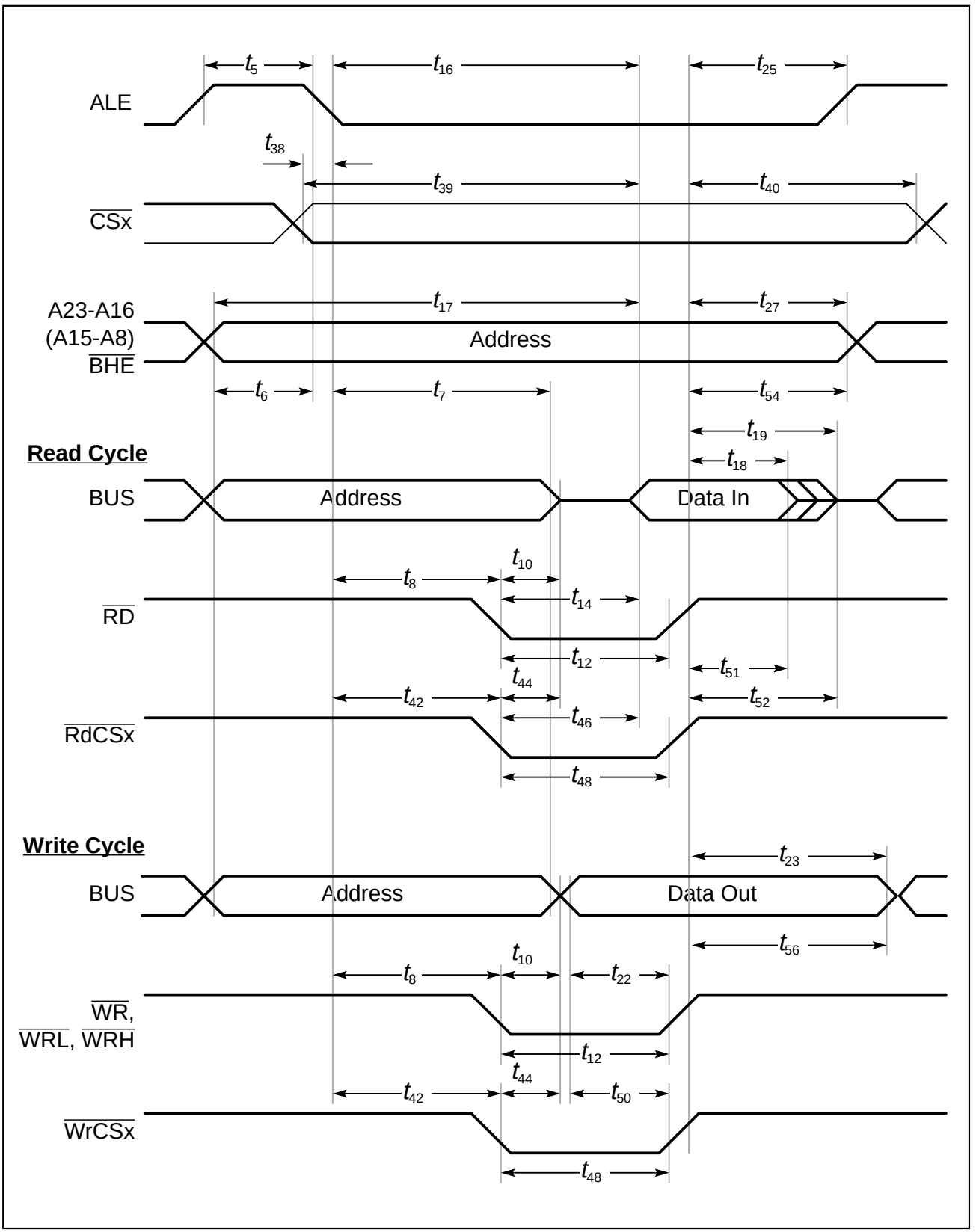


Figure 12-2
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Extended ALE

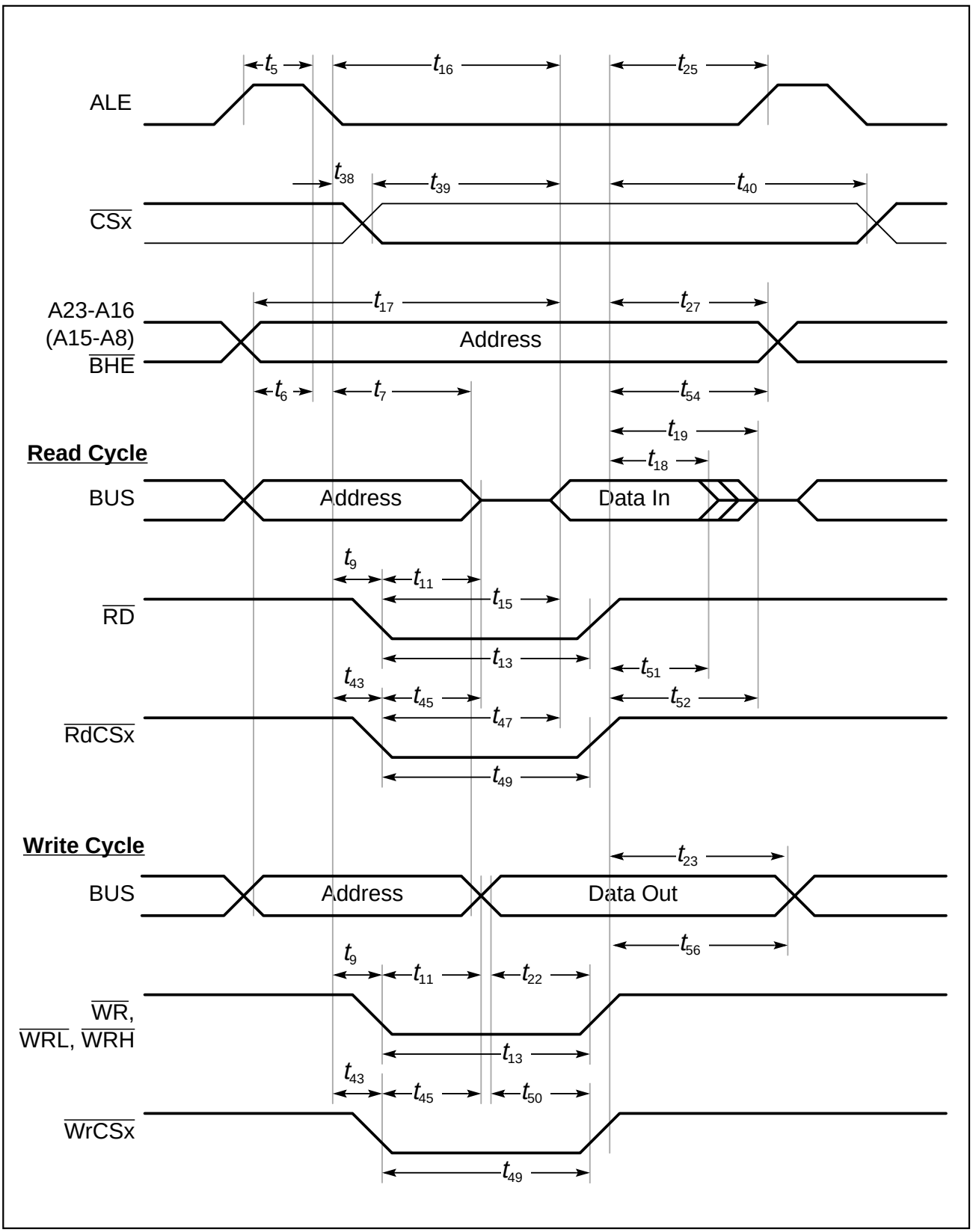


Figure 12-3
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE

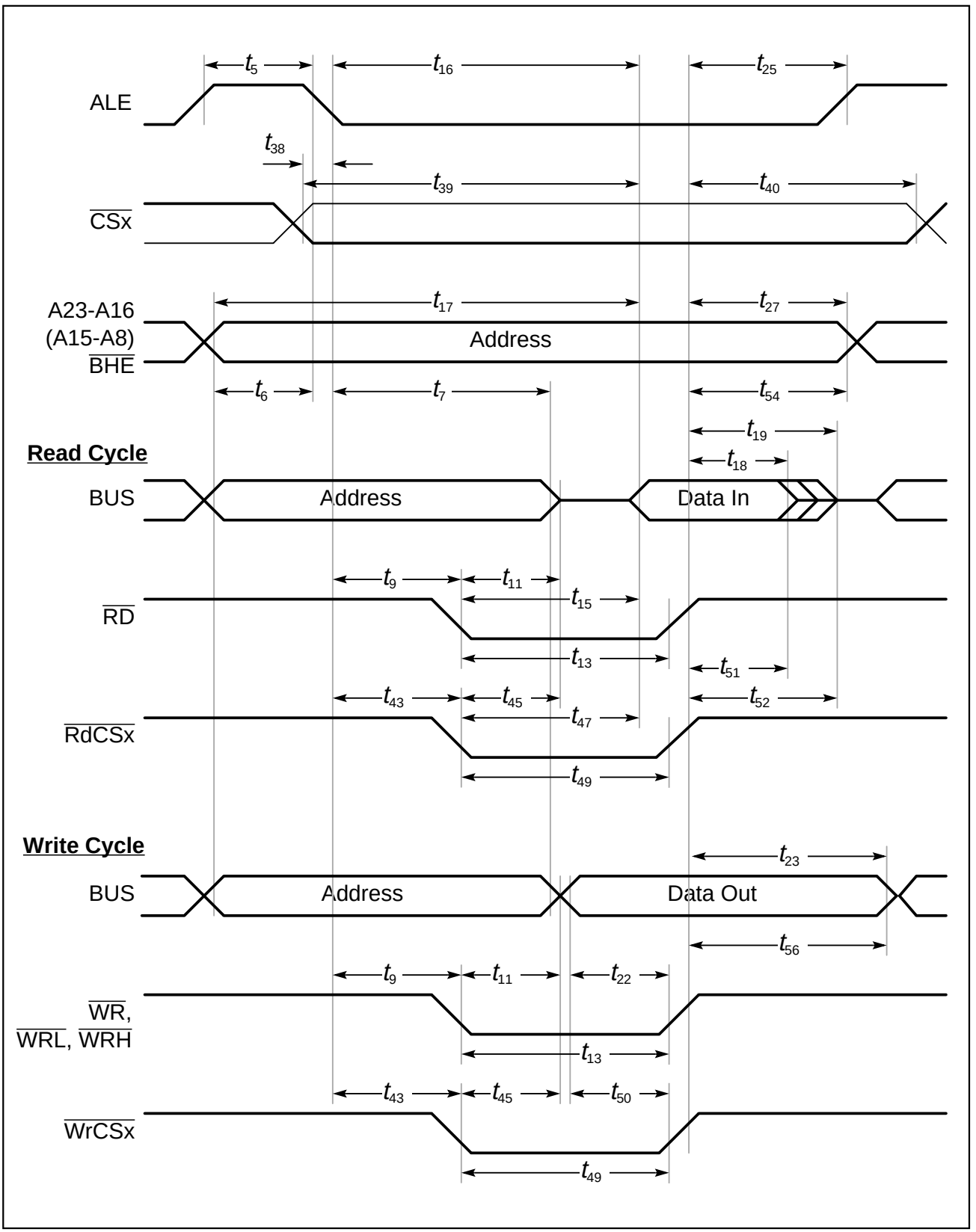


Figure 12-4
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

Demultiplexed Bus

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB-C165-LM, SAB-C165-RM, SAB-C165-LF, SAB-C165-RF
T_A = -40 to +85 °C for SAF-C165-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 + 2t_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 + 2t_A + t_C	–	4TCL - 30 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}	SR	0	–	0	–	ns
Data float after $\overline{RD}$ rising edge (with RW-delay)	t_{20}	SR	–	35 + t_F	–	2TCL - 15 + t_F	ns
Data float after $\overline{RD}$ rising edge (no RW-delay)	t_{21}	SR	–	15 + t_F	–	TCL - 10 + t_F	ns
Data valid to $\overline{WR}$	t_{22}	CC	35 + t_C	–	2TCL - 15 + 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 $\overline{RD}$, $\overline{WR}$	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 + 2t_A	–	3TCL - 20 + t_C + 2t_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	-5 + t_F	–	-5 + t_F	–	ns
Data hold after $\overline{WrCS}$	t_{57}	CC	10 + t_F	–	TCL - 15 + t_F	–	ns

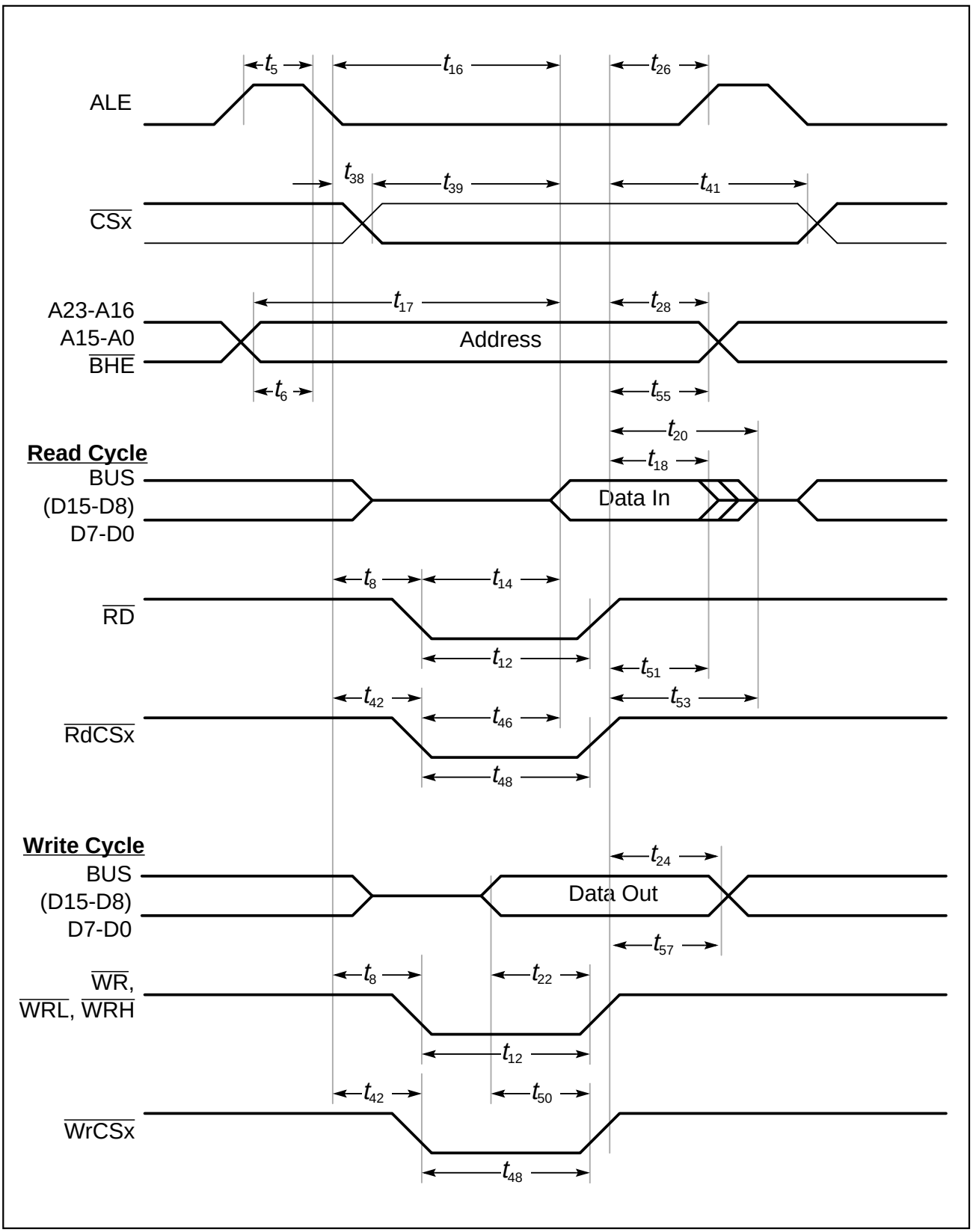


Figure 13-1
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Normal ALE

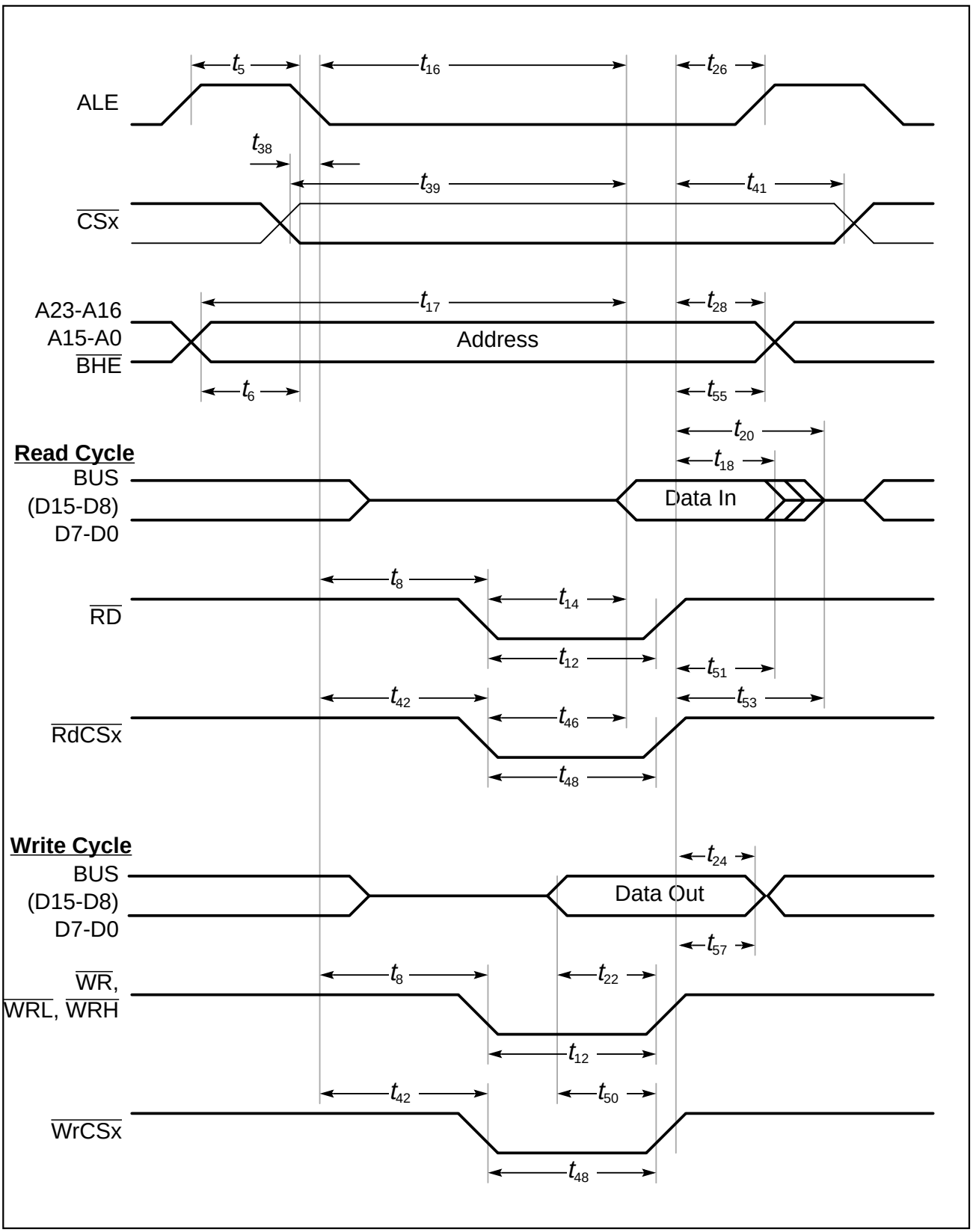


Figure 13-2
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE

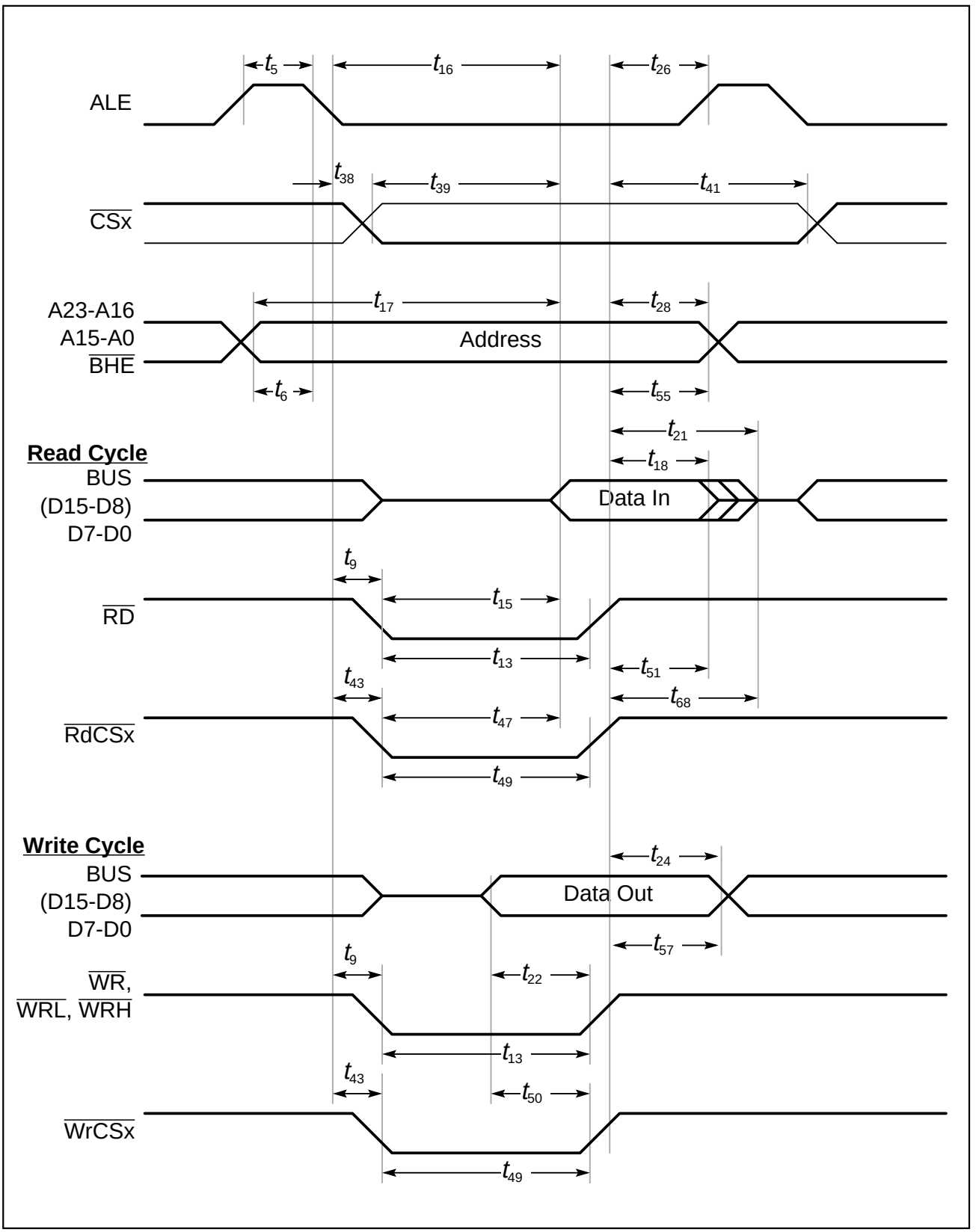


Figure 13-3
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE

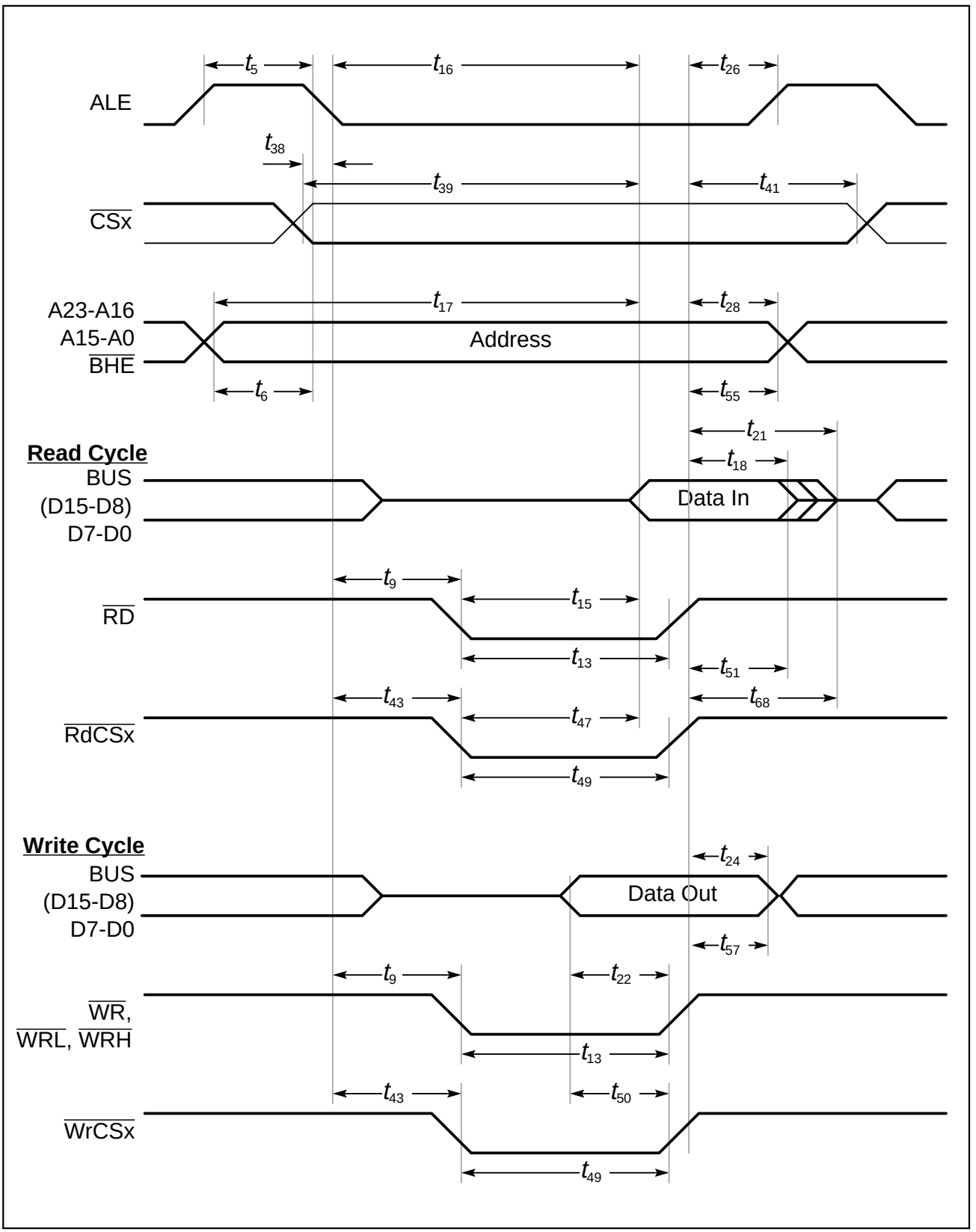


Figure 13-4
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

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

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB-C165-LM, SAB-C165-RM, SAB-C165-LF, SAB-C165-RF
T_A = -40 to +85 °C for SAF-C165-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	10	–	10	–	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 + 2t_A + t_F [2]	0	TCL - 25 + 2t_A + 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 refer to the next following bus cycle.

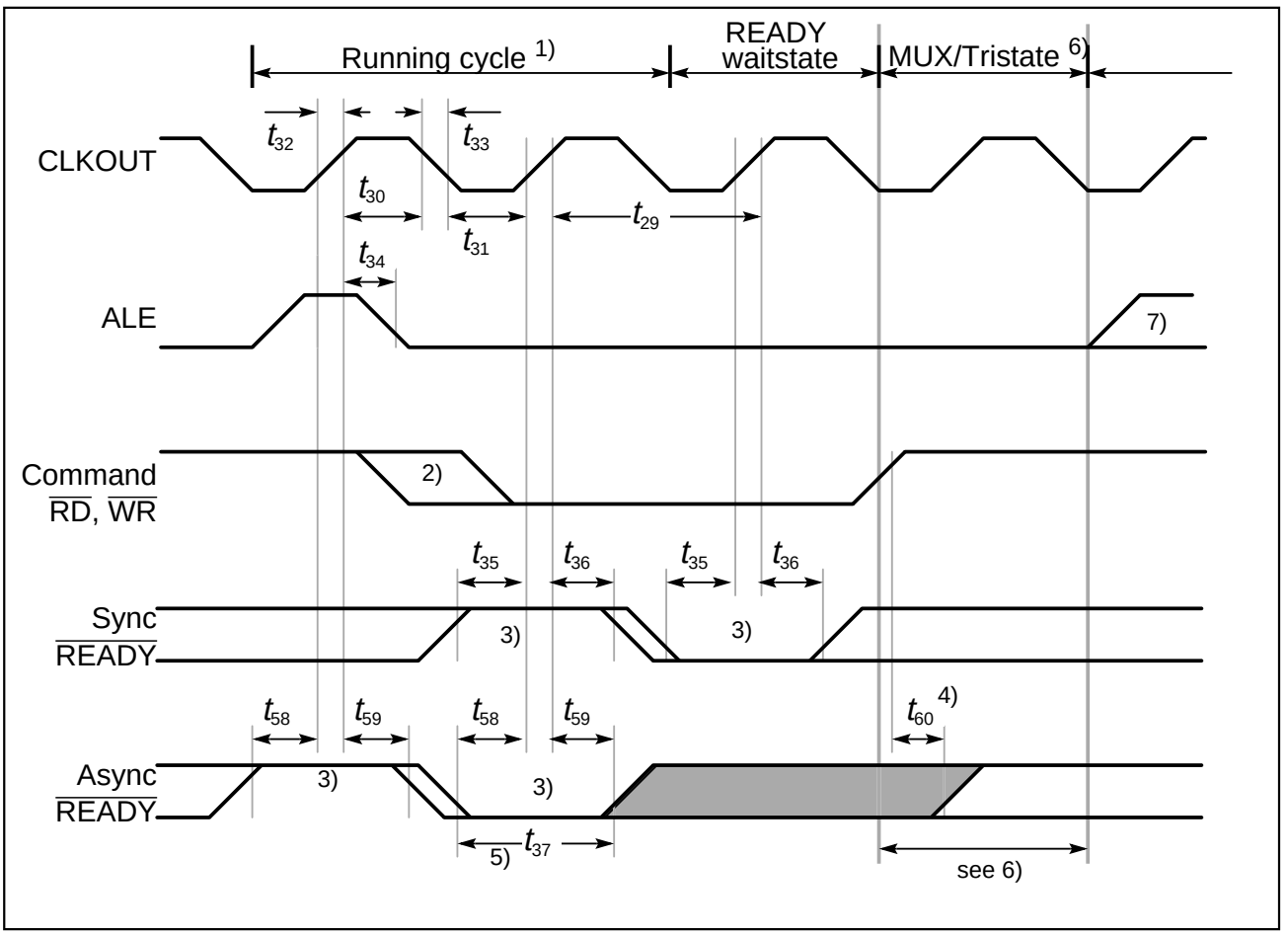


Figure 14
CLKOUT and $\overline{\text{READY}}$

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 (e.g. 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 response 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 (cont'd)

External Bus Arbitration

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB-C165-LM, SAB-C165-RM, SAB-C165-LF, SAB-C165-RF
T_A = -40 to +85 °C for SAF-C165-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 signals release	t_{66}	CC	–	20	–	20	ns
Other signals drive	t_{67}	CC	-5	25	-5	25	ns

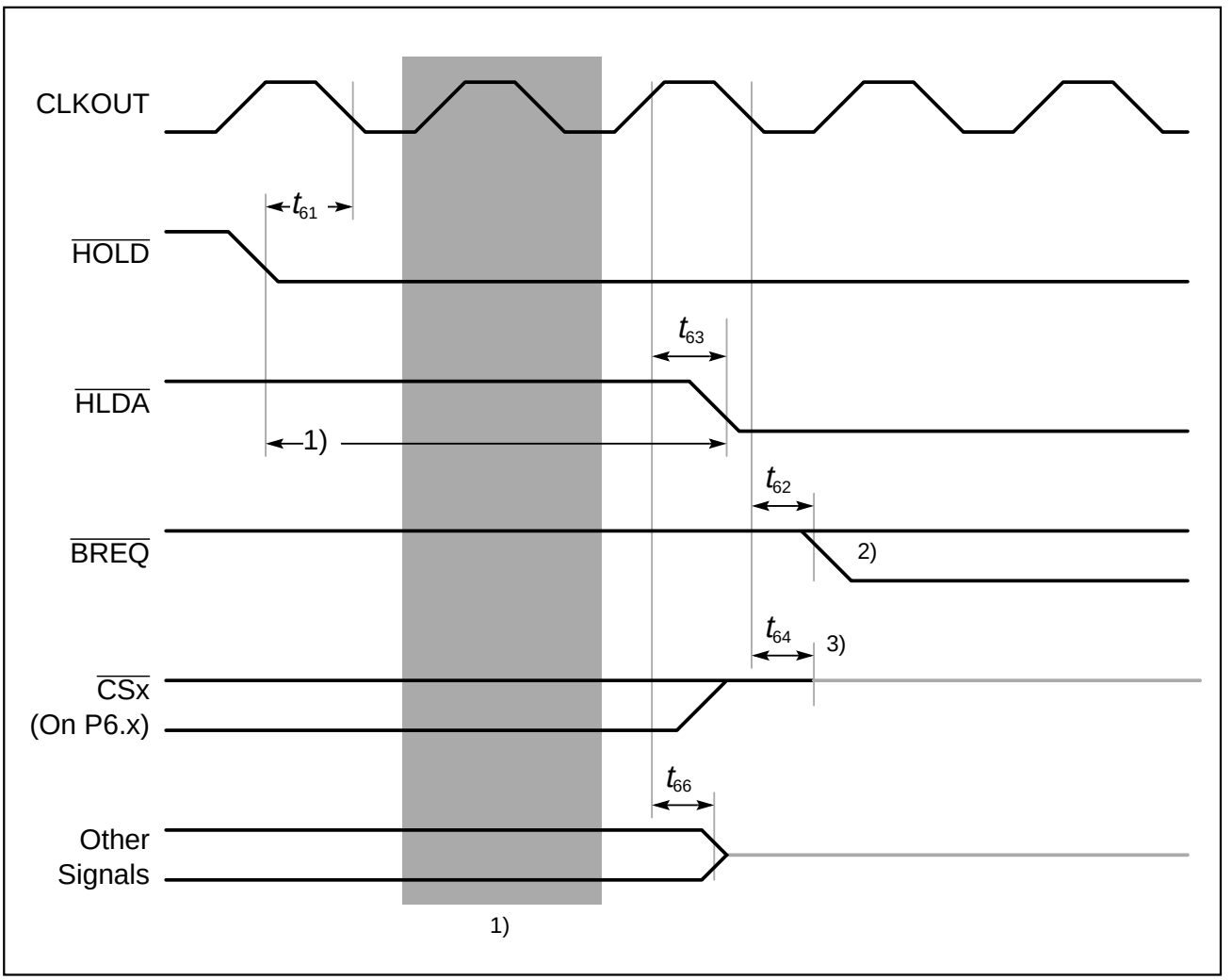


Figure 15
External Bus Arbitration, Releasing the Bus

Notes

1) The C165 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}.

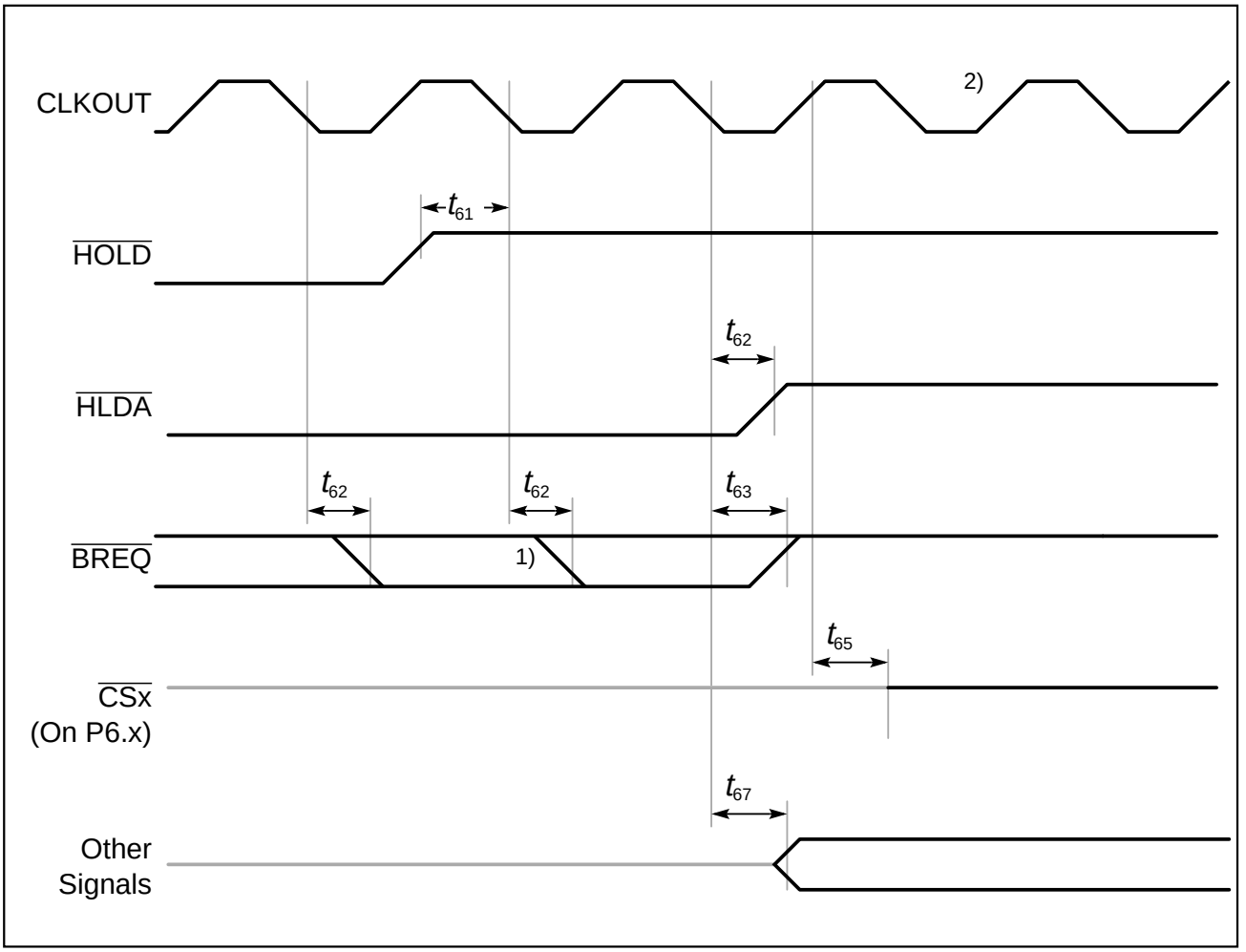


Figure 16
External Bus Arbitration, (Regaining the Bus)

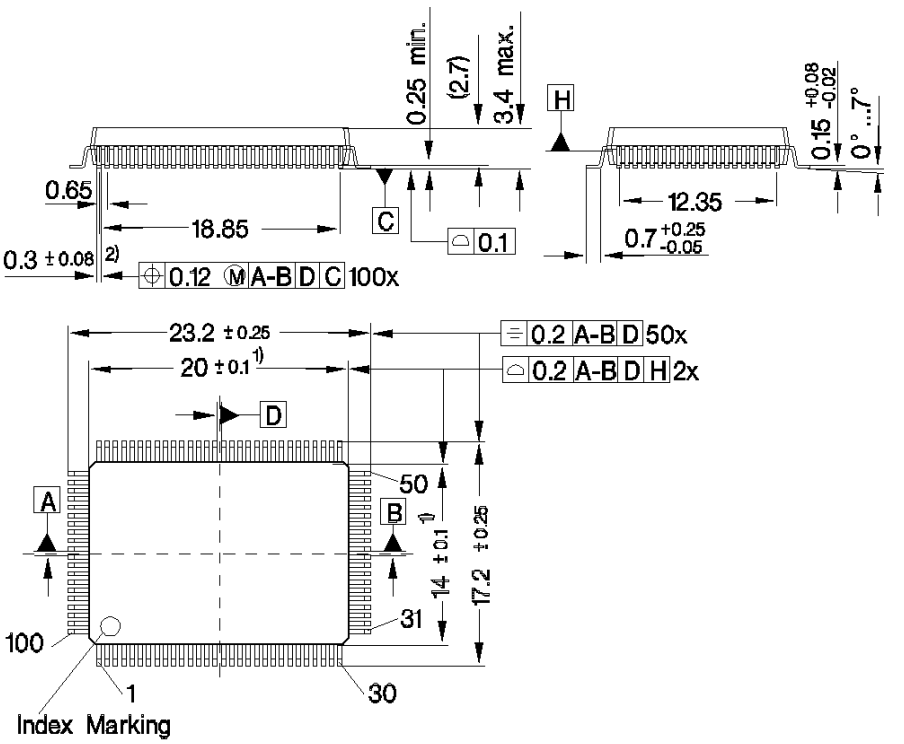
Notes

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 C165 requesting the bus.

2) The next C165 driven bus cycle may start here.

Package Outlines

Plastic Package, P-MQFP-100-2 (SMD)
(Plastic Metric Quad Flat Package)

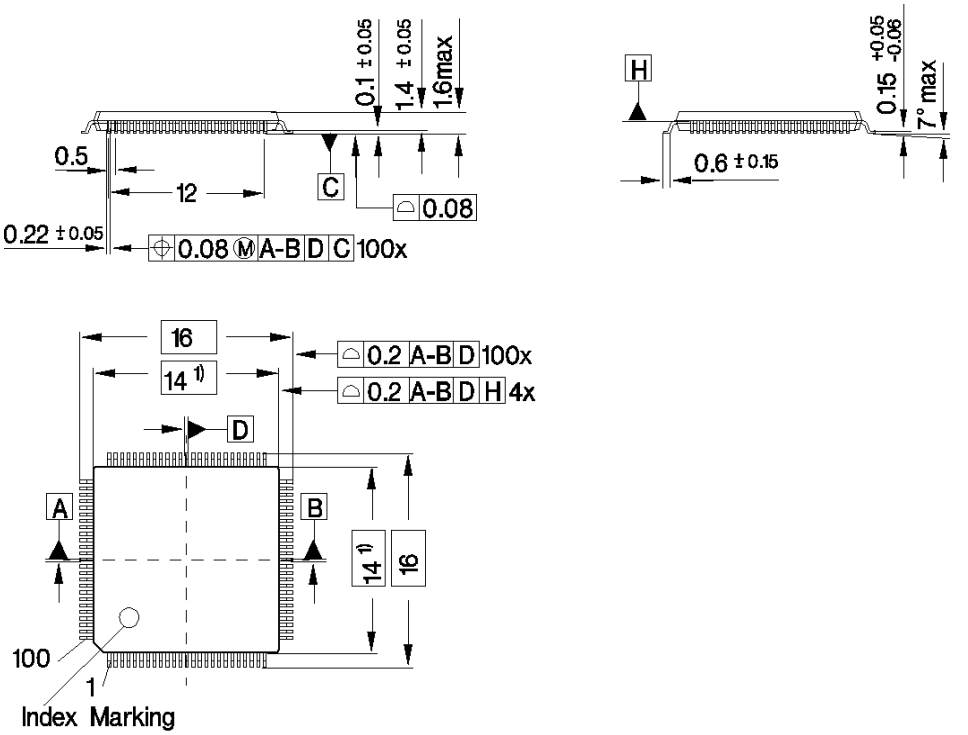


2) Does not include dambar protrusion of 0.08 max. per side
1) Does not include plastic or metal protrusion of 0.25 max. per side

Figure 17

Package Outline

Plastic Package, P-TQFP-100-3 (SMD)
(Plastic Thin Metric Quad Flat Package)



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

Figure 18

Sorts of Packing

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

SMD = Surface Mounted Device

SIEMENS

C16x-Family of SAB 80C166/83C166
High-Performance CMOS 16-Bit Microcontrollers

Preliminary
SAB 80C166/83C166 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
- Register-Based Design with Multiple Variable Register Banks
- Single-Cycle Context Switching Support
- Up to 256 KBytes Linear Address Space for Code and Data
- 1 KByte On-Chip RAM
- 32 KBytes On-Chip ROM (SAB 83C166 only)
- Programmable External Bus Characteristics for Different Address Ranges
- 8-Bit or 16-Bit External Data Bus
- Multiplexed or Demultiplexed External Address/Data Buses
- Hold and Hold-Acknowledge Bus Arbitration Support
- 512 Bytes On-Chip Special Function Register Area
- Idle and Power Down Modes
- 8-Channel Interrupt-Driven Single-Cycle Data Transfer Facilities via Peripheral Event Controller (PEC)
- 16-Priority-Level Interrupt System
- 10-Channel 10-bit A/D Converter with 9.7 µs Conversion Time
- 16-Channel Capture/Compare Unit
- Two Multi-Functional General Purpose Timer Units with 5 Timers
- Two Serial Channels (USARTs)
- Programmable Watchdog Timer
- Up to 76 General Purpose I/O Lines
- 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
- 100-Pin Plastic MQFP Package (EIAJ)

Introduction

The SAB 80C166 is the first representative of the Siemens SAB 80C166 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.

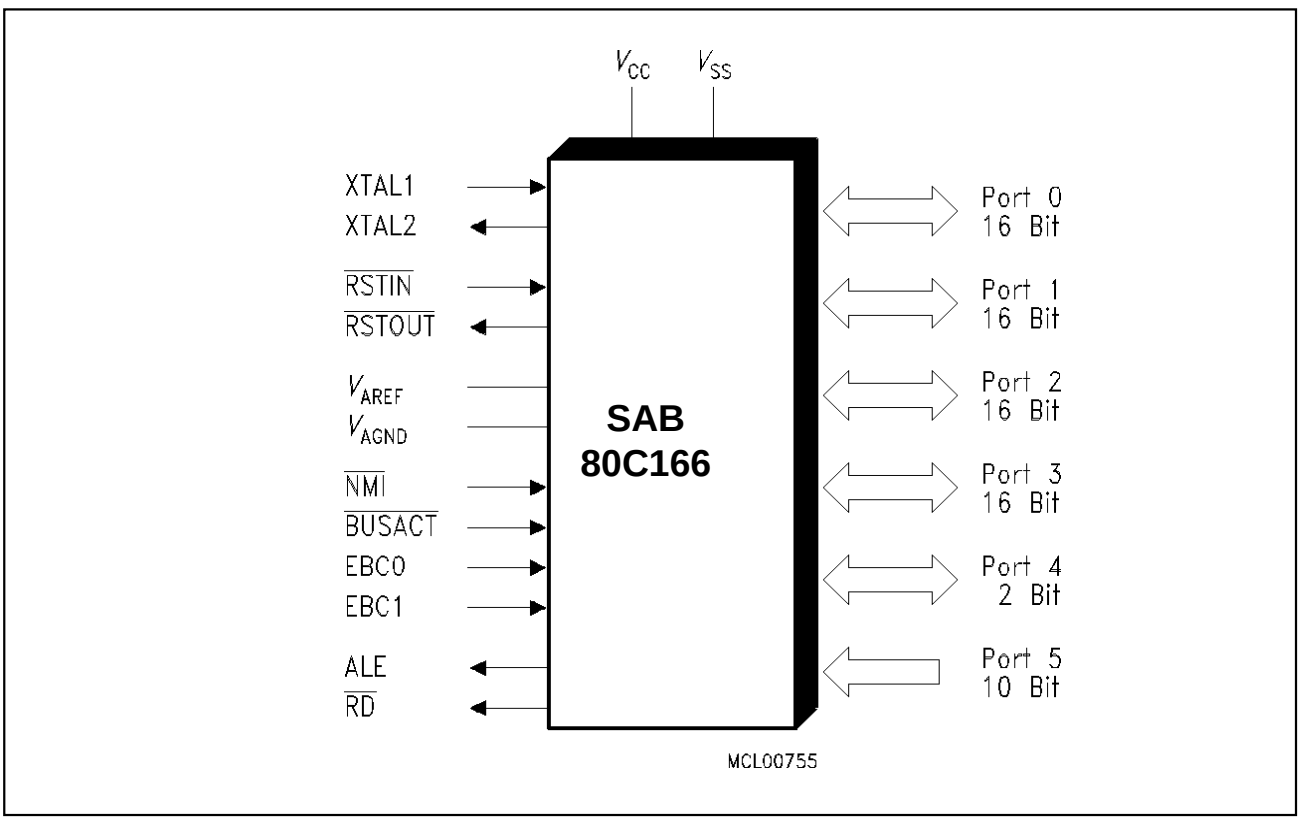


Figure 1
Logic Symbol

Ordering Information

Type	Ordering Code	Package	Function
SAB 83C166-5M	Q67121-D...	P-MQFP-100-2	16-bit microcontroller, 0 ˚C to +70 ˚C, 1 KByte RAM and 32 KByte ROM
SAB 83C166-5M-T3	Q67121-D...	P-MQFP-100-2	16-bit microcontroller, -40 ˚C to +85 ˚C, 1 KByte RAM and 32 KByte ROM
SAB 80C166-M	Q67121-C848	P-MQFP-100-2	16-bit microcontroller, 0 ˚C to +70 ˚C 1 KByte RAM
SAB 80C166-M-T3	Q67121-C900	P-MQFP-100-2	16-bit microcontroller, -40 ˚C to +85 ˚C 1 KByte RAM

Note: The ordering codes (Q67120-D...) for the Mask-ROM versions are defined for each product after verification of the respective ROM code.

Pin Configuration Rectangular P-MQFP-100-2
(top view)

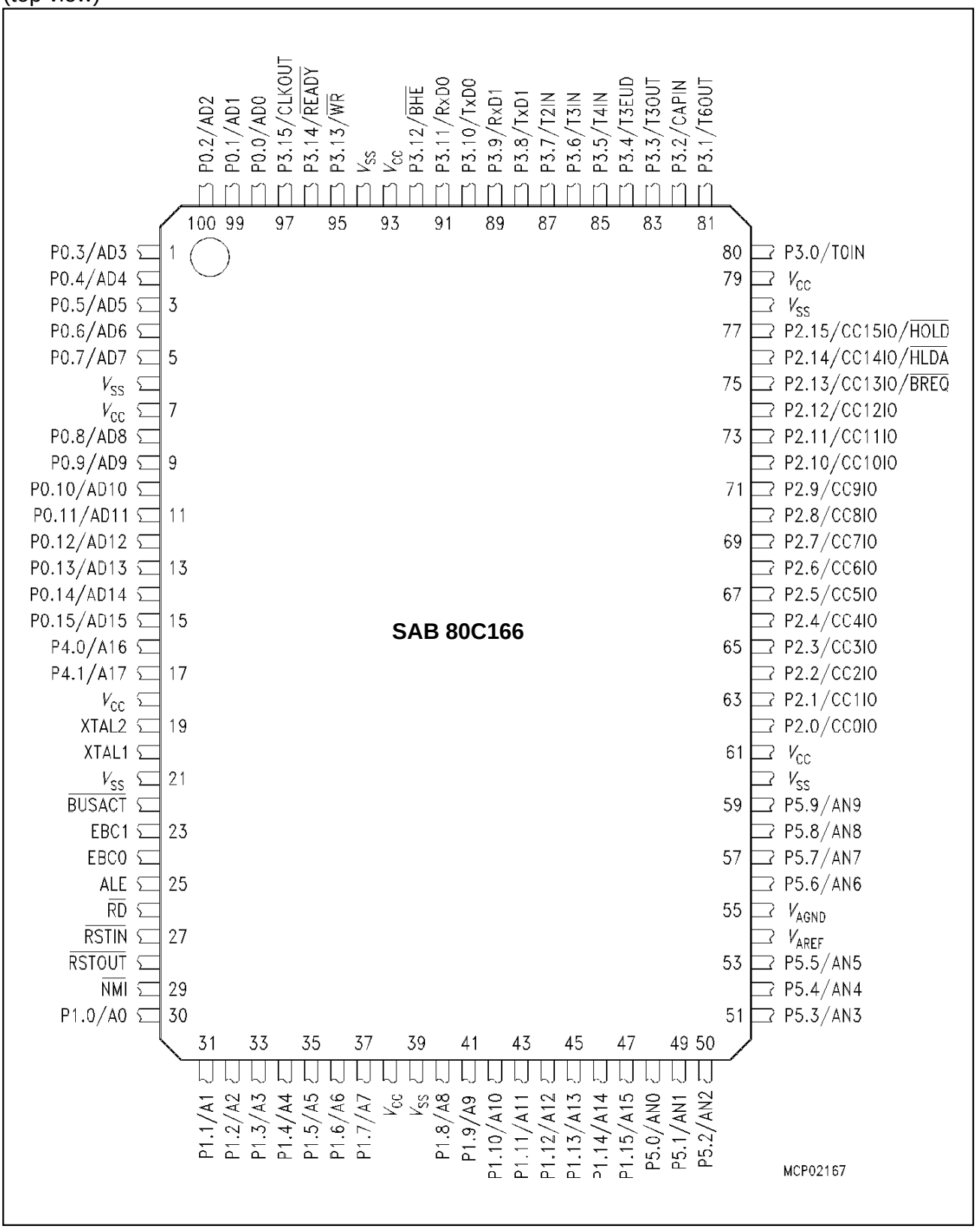


Figure 2

Pin Definitions and Functions

Symbol	Pin Number	Input Output	Function
P4.0 – P4.1	16-17	I/O	Port 4 is a 2-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: P4.0 A16 Least Significant Segment Addr. Line P4.1 A17 Most Significant Segment Addr. Line
	16 17	O O	
XTAL1 XTAL2	20 19	I O	XTAL1: Input to the oscillator amplifier and input to the internal clock generator XTAL2: Output of the oscillator amplifier circuit. 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.
$\overline{\text{BUSACT}}$, EBC1, EBC0	22 23 24	I I I	External Bus Configuration selection inputs. These pins are sampled during reset and select either the single chip mode or one of the four external bus configurations: $\overline{\text{BUSACT}}$ EBC1 EBC0 Mode/Bus Configuration 0 0 0 8-bit demultiplexed bus 0 0 1 8-bit multiplexed bus 0 1 0 16-bit multiplexed bus 0 1 1 16-bit demultiplexed bus 1 0 0 Single chip mode 1 0 1 Reserved. 1 1 0 Reserved. 1 1 1 Reserved. ROMless versions must have pin $\overline{\text{BUSACT}}$ tied to '0'.
$\overline{\text{RSTIN}}$	27	I	Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the SAB 80C166. An internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
$\overline{\text{RSTOUT}}$	28	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.

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input Output	Function
$\overline{\text{NMI}}$	29	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, pin $\overline{\text{NMI}}$ must be low in order to force the SAB 80C166 to go into power down mode. If $\overline{\text{NMI}}$ is high, when PWRDN is executed, the part will continue to run in normal mode. If not used, pull $\overline{\text{NMI}}$ high externally.
ALE	25	O	Address Latch Enable Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.
$\overline{\text{RD}}$	26	O	External Memory Read Strobe. $\overline{\text{RD}}$ is activated for every external instruction or data read access.
P1.0 – P1.15	30-37 40-47	I/O	Port 1 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 1 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.
P5.0 – P5.9	48-53 56-59	I I	Port 5 is a 10-bit input-only port with Schmitt-Trigger characteristics. The pins of Port 5 also serve as the (up to 10) analog input channels for the A/D converter, where P5.x equals ANx (Analog input channel x).
P2.0 – P2.15	62-77 62 75 76 77	I/O I/O I/O O I/O O I/O I	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. The following Port 2 pins also serve for alternate functions: P2.0 CC0IO CAPCOM: CC0 Cap.-In/Comp.Out ... ... P2.13 CC13IO CAPCOM: CC13 Cap.-In/Comp.Out, $\overline{\text{BREQ}}$ External Bus Request Output P2.14 CC14IO CAPCOM: CC14 Cap.-In/Comp.Out, $\overline{\text{HLDA}}$ External Bus Hold Acknowl. Output P2.15 CC15IO CAPCOM: CC15 Cap.-In/Comp.Out, $\overline{\text{HOLD}}$ External Bus Hold Request Input

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input Output	Function
P3.0 – P3.15	80-92, 95-97	I/O I/O	Port 3 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. The following Port 3 pins also serve for alternate functions: P3.0 T0IN CAPCOM Timer T0 Count Input 80 I P3.1 T6OUT GPT2 Timer T6 Toggle Latch Output 81 O P3.2 CAPIN GPT2 Register CAPREL Capture Input 82 I P3.3 T3OUT GPT1 Timer T3 Toggle Latch Output 83 O P3.4 T3EUD GPT1 Timer T3 Ext.Up/Down Ctrl.Input 84 I P3.5 T4IN GPT1 Timer T4 Input for 85 I Count/Gate/Reload/Capture P3.6 T3IN GPT1 Timer T3 Count/Gate Input 86 I P3.7 T2IN GPT1 Timer T2 Input for 87 I Count/Gate/Reload/Capture P3.8 TxD1 ASC1 Clock/Data Output (Asyn./Syn.) 88 O P3.9 RxD1 ASC1 Data Input (Asyn.) or I/O (Syn.) 89 I/O P3.10 T×D0 ASC0 Clock/Data Output (Asyn./Syn.) 90 O P3.11 R×D0 ASC0 Data Input (Asyn.) or I/O (Syn.) 91 I/O P3.12 $\overline{BHE}$ Ext. Memory High Byte Enable Signal 92 O P3.13 $\overline{WR}$ External Memory Write Strobe 95 O P3.14 $\overline{READY}$ Ready Signal Input 96 I P3.15 CLKOUT System Clock Output (=CPU Clock) 97 O
P0.0 – P0.15	98 – 5 8 – 15	I/O	Port 0 is a 16-bit bidirectional IO 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 0 serves as the address (A) and address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes. **Demultiplexed bus modes:** Data Path Width: 8-bit 16-bit P0.0 – P0.7: D0 – D7 D0 - D7 P0.8 – P0.15: output! D8 - D15 **Multiplexed bus modes:** Data Path Width: 8-bit 16-bit P0.0 – P0.7: AD0 – AD7 AD0 - AD7 P0.8 – P0.15: A8 - A15 AD8 - AD15
V_{AREF}	54	-	Reference voltage for the A/D converter.
V_{AGND}	55	-	Reference ground for the A/D converter.

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input Output	Function
V_{CC}	7, 18, 38, 61, 79, 93	-	Digital Supply Voltage: + 5 V during normal operation and idle mode. ≥ 2.5 V during power down mode
V_{SS}	6, 21, 39, 60, 78, 94	-	Digital Ground.

Functional Description

The architecture of the SAB 80C166 combines advantages of both RISC and CISC processors and of advanced peripheral subsystems in a very well-balanced way. The following block diagram gives an overview of the different on-chip components and of the advanced, high bandwidth internal bus structure of the SAB 80C166.

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

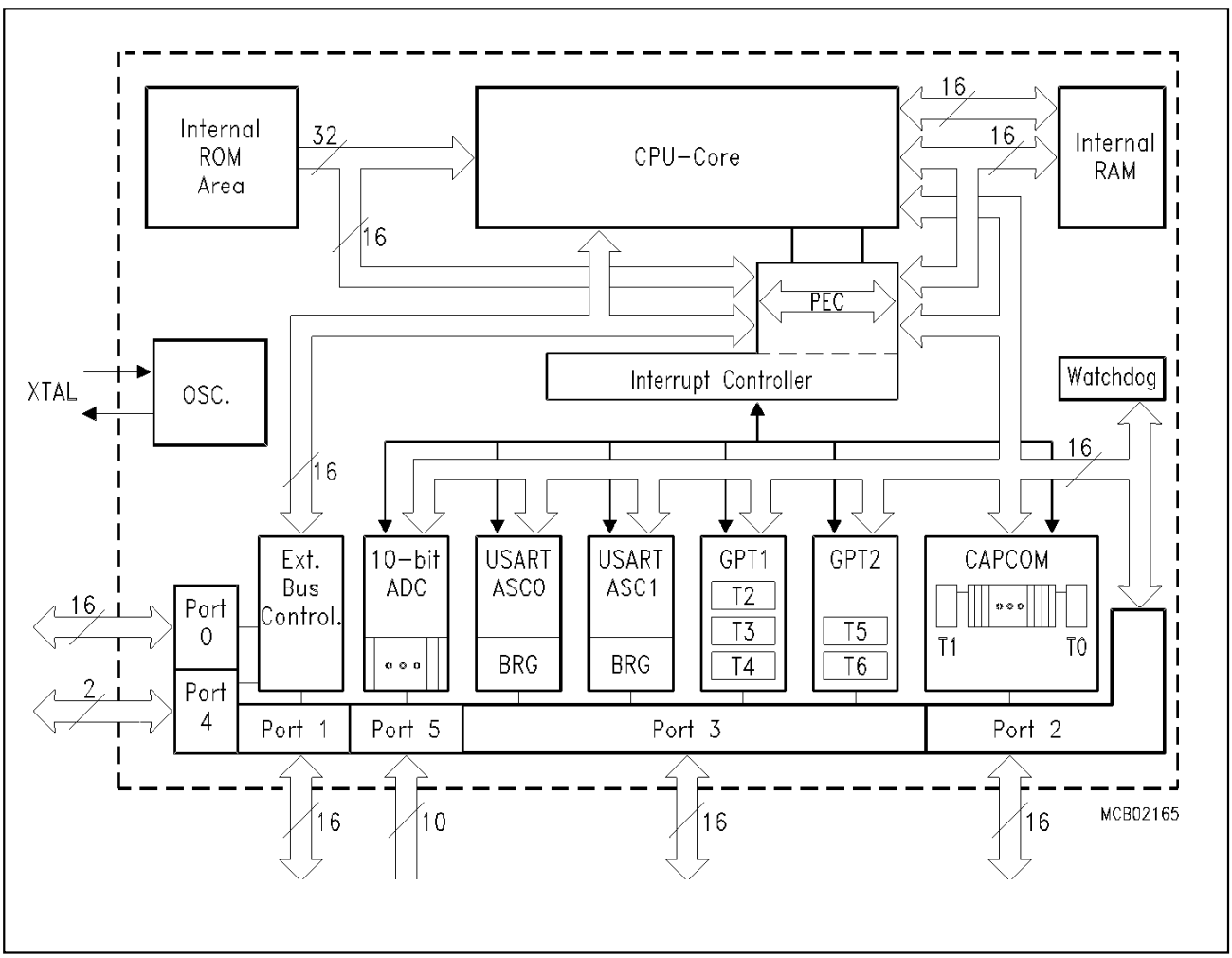


Figure 3
Block Diagram

Memory Organization

The memory space of the SAB 80C166 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 256 KBytes. Address space expansion to 16 MBytes is provided for future versions. The entire memory space can be accessed bytewise or wordwise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

The SAB 83C166 contains 32 KBytes of on-chip mask-programmable ROM for code or constant data. The ROM can be mapped to either segment 0 or segment 1.

1 KByte of on-chip RAM is 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).

512 bytes of the address space are reserved for the Special Function Register area. SFRs are wordwide registers which are used for controlling and monitoring functions of the different on-chip units. 98 SFRs are currently implemented. Unused SFR addresses are reserved for future members of the SAB 80C166 family.

In order to meet the needs of designs where more memory is required than is provided on chip, up to 256 KBytes of external RAM and/or ROM can be connected to the microcontroller.

External Bus Controller

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

– 16-/18-bit Addresses, 16-bit Data, Demultiplexed

– 16-/18-bit Addresses, 16-bit Data, Multiplexed

– 16-/18-bit Addresses, 8-bit Data, Multiplexed

– 16-/18-bit Addresses, 8-bit Data, Demultiplexed

In the demultiplexed bus modes, addresses are output on Port 1 and data is input/output on Port 0. In the multiplexed bus modes both addresses and data use Port 0 for input/output.

Important timing characteristics of the external bus interface (Memory Cycle Time, Memory Tri-State Time, Read/Write Delay and Length of ALE, i.e. address setup/hold time with respect to ALE) have been made programmable to allow the user the adaption of a wide range of different types of memories. In addition, different address ranges may be accessed with different bus characteristics. Access to very slow memories is supported via a particular ‘Ready’ function. A $\overline{\text{HOLD}}$/$\overline{\text{HLDA}}$ protocol is available for bus arbitration.

For applications which require less than 64 KBytes of external memory space, a non-segmented memory model can be selected. In this case all memory locations can be addressed by 16 bits and Port 4 is not required to output the additional segment address lines.

Central Processing Unit (CPU)

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

Based on these hardware provisions, most of the SAB 80C166's instructions can be executed in just one machine cycle which requires 100 ns at 20-MHz CPU clock. For example, shift and rotate instructions are always processed during one machine cycle independent of the number of bits to be shifted. All multiple-cycle instructions have been optimized so that they can be executed very fast as well: branches in 2 cycles, a 16 × 16 bit multiplication in 5 cycles and a 32-/16 bit division in 10 cycles. Another pipeline optimization, the so-called 'Jump Cache', allows reducing the execution time of repeatedly performed jumps in a loop from 2 cycles to 1 cycle.

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

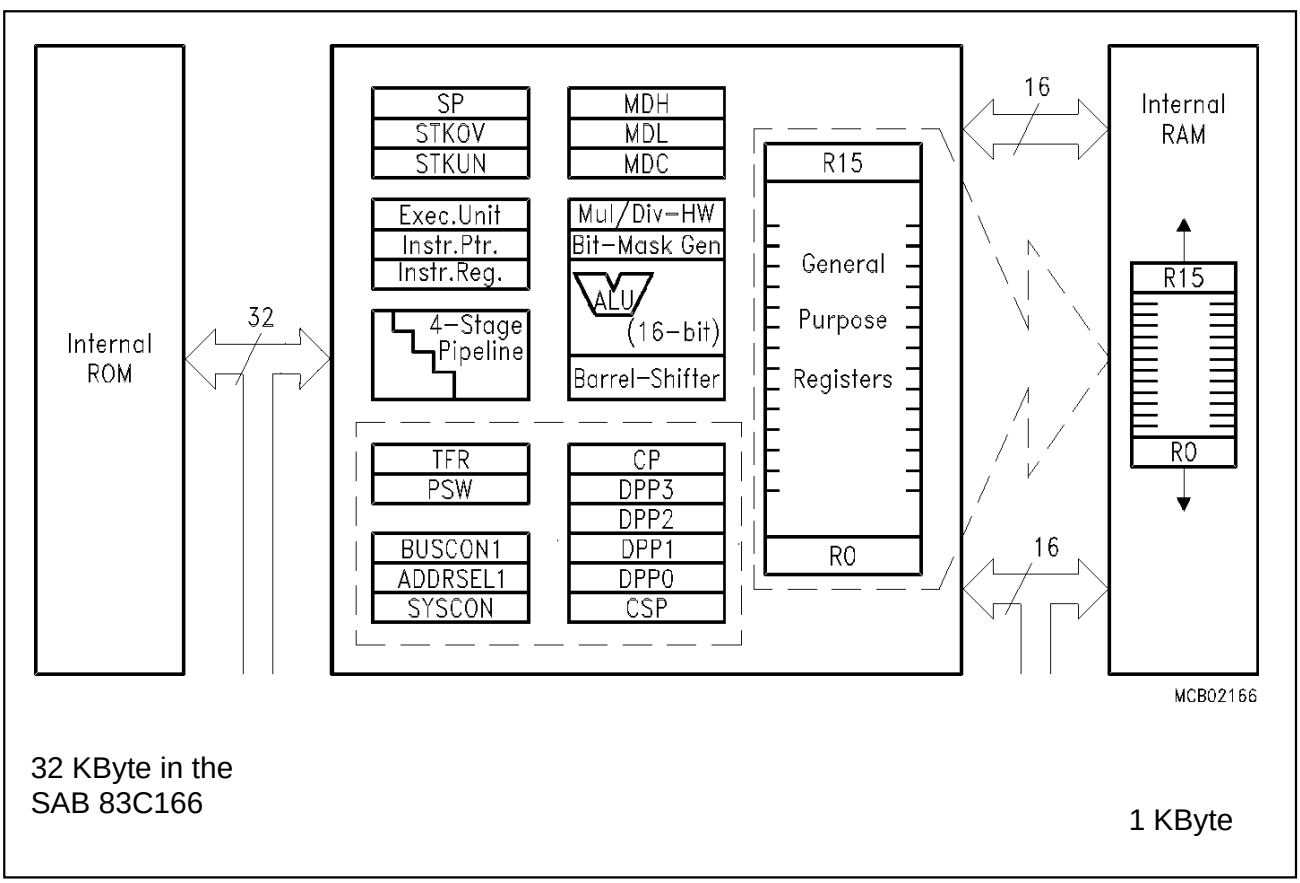


Figure 4
CPU Block Diagram

A system stack of up to 512 bytes is provided as a storage for temporary data. The system stack is allocated in the on-chip RAM area, and it is accessed by the CPU via the stack pointer (SP) register. Two separate SFRs, STKOV and STKUN, are implicitly compared against the stack pointer value upon each stack access for the detection of a stack overflow or underflow.

The high performance offered by the hardware implementation of the CPU can efficiently be utilized by a programmer via the highly efficient SAB 80C166 instruction set which includes the following instruction classes:

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

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

Interrupt System

With an interrupt response time within a range from just 250 ns to 600 ns (in case of internal program execution), the SAB 80C166 is capable of reacting very fast to the occurrence of non-deterministic events.

The architecture of the SAB 80C166 supports several mechanisms for fast and flexible response to service requests that can be generated from various sources internal or external to the microcontroller. Any of these interrupt requests can be programmed to being serviced by the Interrupt Controller or by the Peripheral Event Controller (PEC).

In contrast to a standard interrupt service where the current program execution is suspended and a branch to the interrupt vector table is performed, just one cycle is ‘stolen’ from the current CPU activity to perform a PEC service. A PEC service implies a single byte or word data transfer between any two memory locations with an additional increment of either the PEC source or the destination pointer. An individual PEC transfer counter is implicity decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited, for example, for supporting the transmission or reception of blocks of data, or for transferring A/D converted results to a memory table. The SAB 80C166 has 8 PEC channels each of which offers such fast interrupt-driven data transfer capabilities.

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

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

The following table shows all of the possible SAB 80C166 interrupt sources and the corresponding hardware-related interrupt flags, vectors, vector locations and trap (interrupt) numbers:

Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
CAPCOM Register 0	CC0IR	CC0IE	CC0INT	40_H	10_H
CAPCOM Register 1	CC1IR	CC1IE	CC1INT	44_H	11_H
CAPCOM Register 2	CC2IR	CC2IE	CC2INT	48_H	12_H
CAPCOM Register 3	CC3IR	CC3IE	CC3INT	$4C_H$	13_H
CAPCOM Register 4	CC4IR	CC4IE	CC4INT	50_H	14_H
CAPCOM Register 5	CC5IR	CC5IE	CC5INT	54_H	15_H
CAPCOM Register 6	CC6IR	CC6IE	CC6INT	58_H	16_H
CAPCOM Register 7	CC7IR	CC7IE	CC7INT	$5C_H$	17_H
CAPCOM Register 8	CC8IR	CC8IE	CC8INT	60_H	18_H
CAPCOM Register 9	CC9IR	CC9IE	CC9INT	64_H	19_H
CAPCOM Register 10	CC10IR	CC10IE	CC10INT	68_H	$1A_H$
CAPCOM Register 11	CC11IR	CC11IE	CC11INT	$6C_H$	$1B_H$
CAPCOM Register 12	CC12IR	CC12IE	CC12INT	70_H	$1C_H$
CAPCOM Register 13	CC13IR	CC13IE	CC13INT	74_H	$1D_H$
CAPCOM Register 14	CC14IR	CC14IE	CC14INT	78_H	$1E_H$
CAPCOM Register 15	CC15IR	CC15IE	CC15INT	$7C_H$	$1F_H$
CAPCOM Timer 0	T0IR	T0IE	T0INT	80_H	20_H
CAPCOM Timer 1	T1IR	T1IE	T1INT	84_H	21_H
GPT1 Timer 2	T2IR	T2IE	T2INT	88_H	22_H
GPT1 Timer 3	T3IR	T3IE	T3INT	$8C_H$	23_H
GPT1 Timer 4	T4IR	T4IE	T4INT	90_H	24_H
GPT2 Timer 5	T5IR	T5IE	T5INT	94_H	25_H
GPT2 Timer 6	T6IR	T6IE	T6INT	98_H	26_H
GPT2 CAPREL Register	CRIR	CRIE	CRINT	$9C_H$	27_H
A/D Conversion Complete	ADCIR	ADCIE	ADCINT	$A0_H$	28_H
A/D Overrun Error	ADEIR	ADEIE	ADEINT	$A4_H$	29_H
ASC0 Transmit	S0TIR	S0TIE	S0TINT	$A8_H$	$2A_H$
ASC0 Receive	S0RIR	S0RIE	S0RINT	AC_H	$2B_H$
ASC0 Error	S0EIR	S0EIE	S0EINT	$B0_H$	$2C_H$
ASC1 Transmit	S1TIR	S1TIE	S1TINT	$B4_H$	$2D_H$
ASC1 Receive	S1RIR	S1RIE	S1RINT	$B8_H$	$2E_H$
ASC1 Error	S1EIR	S1EIE	S1EINT	BC_H	$2F_H$

The SAB 80C166 also provides an excellent mechanism to identify and to process exceptions or error conditions that arise during run-time, so-called ‘Hardware Traps’. Hardware traps cause immediate non-maskable system reaction which is similar to a standard interrupt service (branching to a dedicated vector table location). The occurrence of a hardware trap is additionally signified by an individual bit in the trap flag register (TFR). Except when another higher prioritized trap service is in progress, a hardware trap will interrupt any actual program execution. In turn, hardware trap services can normally not be interrupted by standard or PEC interrupts.

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

Exception Condition	**Trap Flag**	**Trap Vector**	**Vector Location**	**Trap Number**	**Trap Priority**
Reset Functions:					
Hardware Reset		RESET	0000_H	00_H	III
Software Reset		RESET	0000_H	00_H	III
Watchdog Timer Overflow		RESET	0000_H	00_H	III
Class A Hardware Traps:					
Non-Maskable Interrupt	NMI	NMITRAP	0008_H	02_H	II
Stack Overflow	STKOF	STOTRAP	0010_H	04_H	II
Stack Underflow	STKUF	STUTRAP	0018_H	06_H	II
Class B Hardware Traps:					
Undefined Opcode	UNDOPC	BTRAP	0028_H	$0A_H$	I
Protected Instruction Fault	PRTFLT	BTRAP	0028_H	$0A_H$	I
Illegal Word Operand Access	ILLOPA	BTRAP	0028_H	$0A_H$	I
Illegal Instruction Access	ILLINA	BTRAP	0028_H	$0A_H$	I
Illegal External Bus Access	ILLBUS	BTRAP	0028_H	$0A_H$	I
Reserved			$[002C_H – 003C_H]$	$[0B_H – 0F_H]$	
Software Traps TRAP Instruction			Any $[0000_H – 01FC_H]$ in steps of 04_H	Any $[00_H – 7F_H]$	Current CPU Priority

Capture/Compare (CAPCOM) Unit

The CAPCOM unit supports generation and control of timing sequences on up to 16 channels with a maximum resolution of 400 ns (@ 20 MHz CPU clock). The CAPCOM unit is typically used to handle high speed I/O tasks such as pulse and waveform generation, pulse width modulation (PMW), Digital to Analog (D/A) conversion, software timing, or time recording relative to external events.

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

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

The capture/compare register array contains 16 dual purpose capture/compare registers, each of which may be individually allocated to either CAPCOM timer T0 or T1, and programmed for capture or compare function. Each register has one port pin associated with it which serves as an input pin for triggering the capture function, or as an output pin to indicate the occurrence of a compare event.

When a capture/compare register has been selected for capture mode, the current contents of the allocated timer will be latched (captured) into the capture/compare register in response to an external event at the port pin which is associated with this register. In addition, a specific interrupt request for this capture/compare register is generated. Either a positive, a negative, or both a positive and a negative external signal transition at the pin can be selected as the triggering event. The contents of all registers which have been selected for one of the five compare modes are continuously compared with the contents of the allocated timers. When a match occurs between the timer value and the value in a capture/compare register, specific actions will be taken based on the selected compare mode.

Compare Modes	Function
Mode 0	Interrupt-only compare mode; several compare interrupts per timer period are possible
Mode 1	Pin toggles on each compare match; several compare events per timer period are possible
Mode 2	Interrupt-only compare mode; only one compare interrupt per timer period is generated
Mode 3	Pin set ‘1’ on match; pin reset ‘0’ on compare time overflow; only one compare event per timer period is generated
Double Register Mode	Two registers operate on one pin; pin toggles on each compare match; several compare events per timer period are possible.

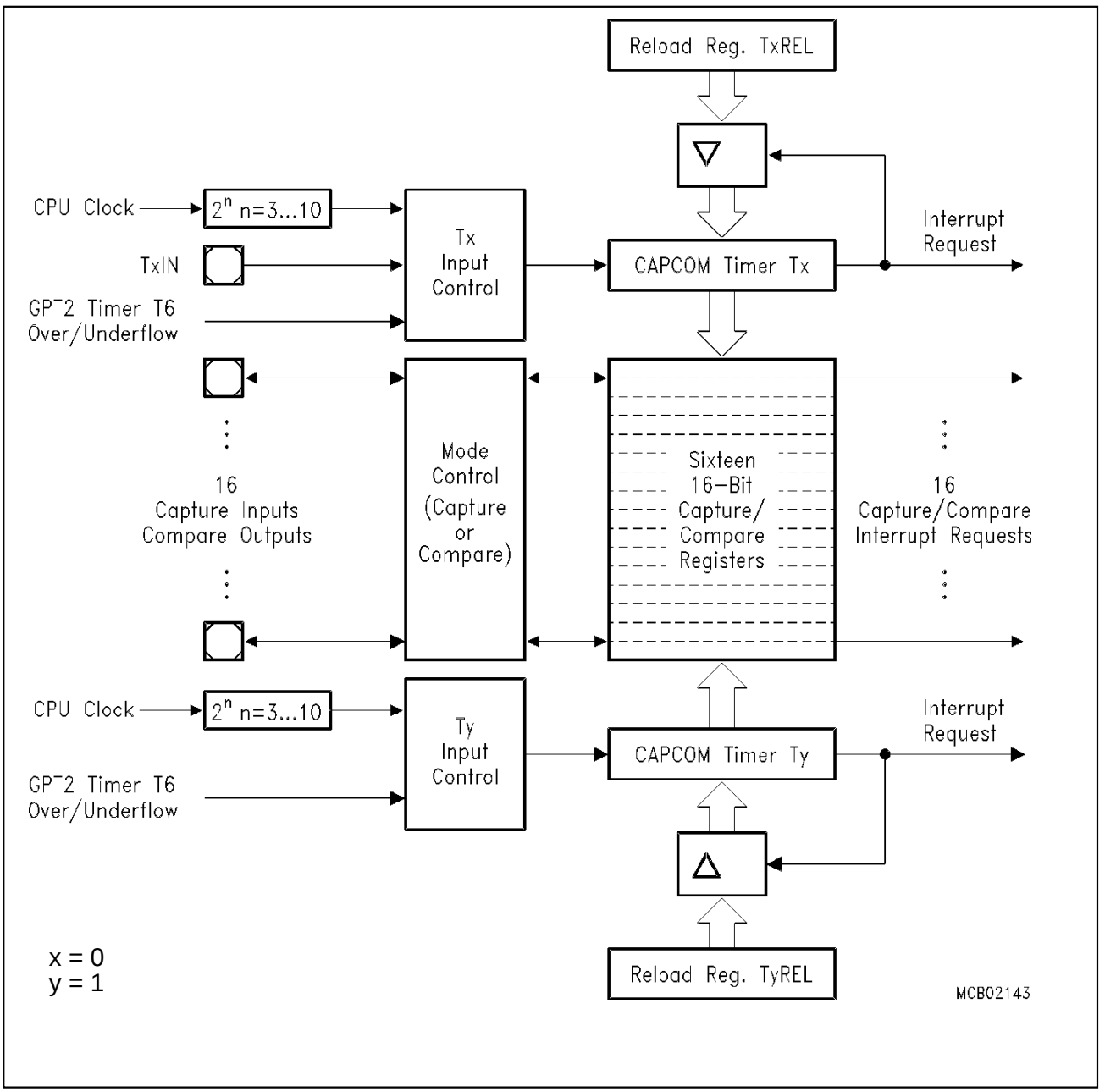


Figure 5
CAPCOM Unit Block Diagram

General Purpose Timer (GPT) Unit

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

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

Each of the three timers T2, T3, T4 of module GPT1 can be configured individually for one of three basic modes of operation, which are Timer, Gated Timer, and Counter Mode. In Timer Mode, the input clock for a timer is derived from the CPU clock, divided by a programmable prescaler, while Counter Mode allows a timer to be clocked in reference to external events.
Pulse width or duty cycle measurement is supported in Gated Timer Mode, where the operation of a timer is controlled by the ‘gate’ level on an external input pin. For these purposes, each timer has one associated port pin (TxIN) which serves as gate or clock input. The maximum resolution of the timers in module GPT1 is 400 ns (@ 20 MHz CPU clock).

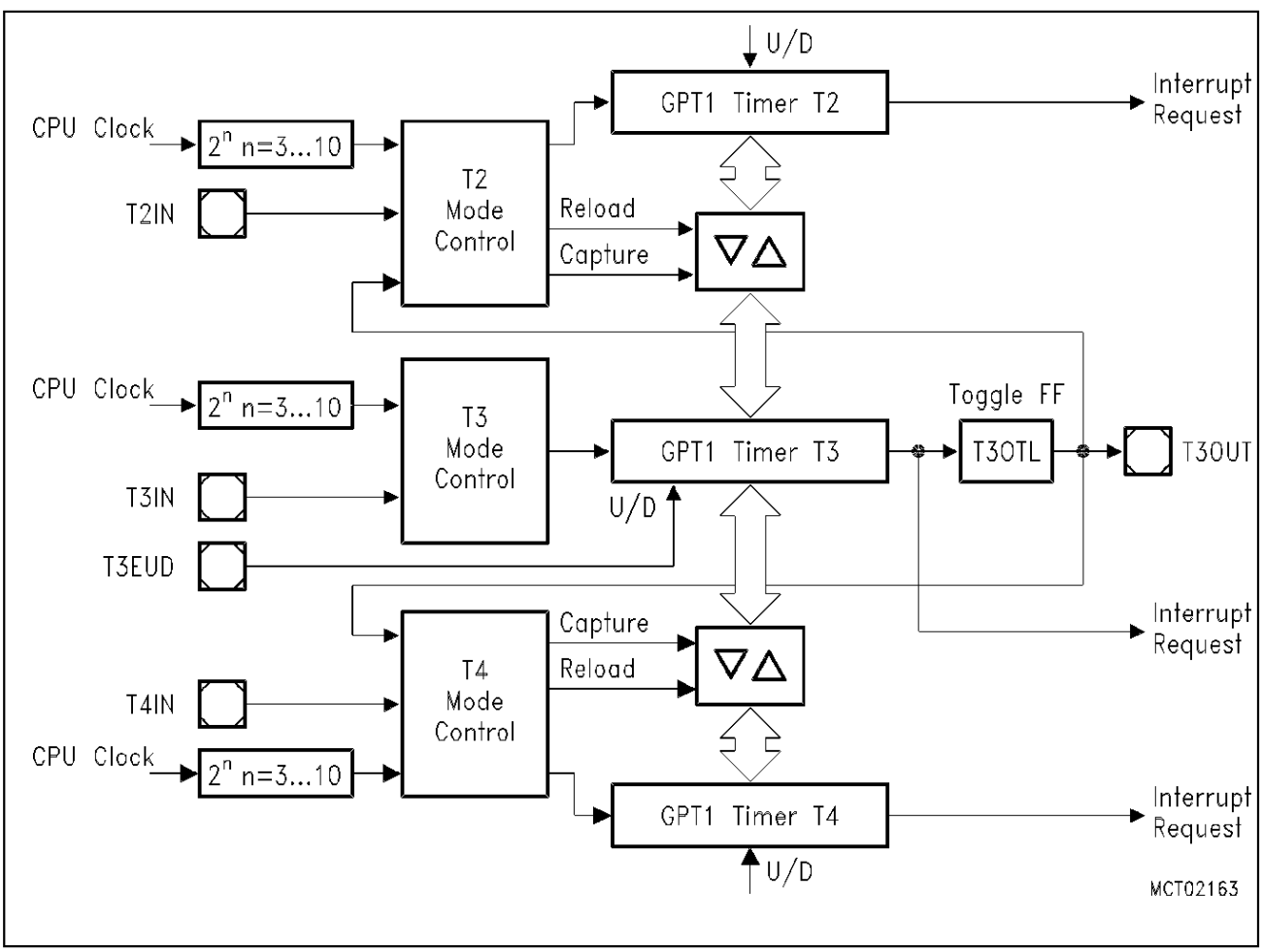


Figure 6
Block Diagram of GPT1

The count direction (up/down) for each timer is programmable by software. For timer T3 the count direction may additionally be altered dynamically by an external signal on a port pin (T3EUD) to facilitate e. g. position tracking.

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

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

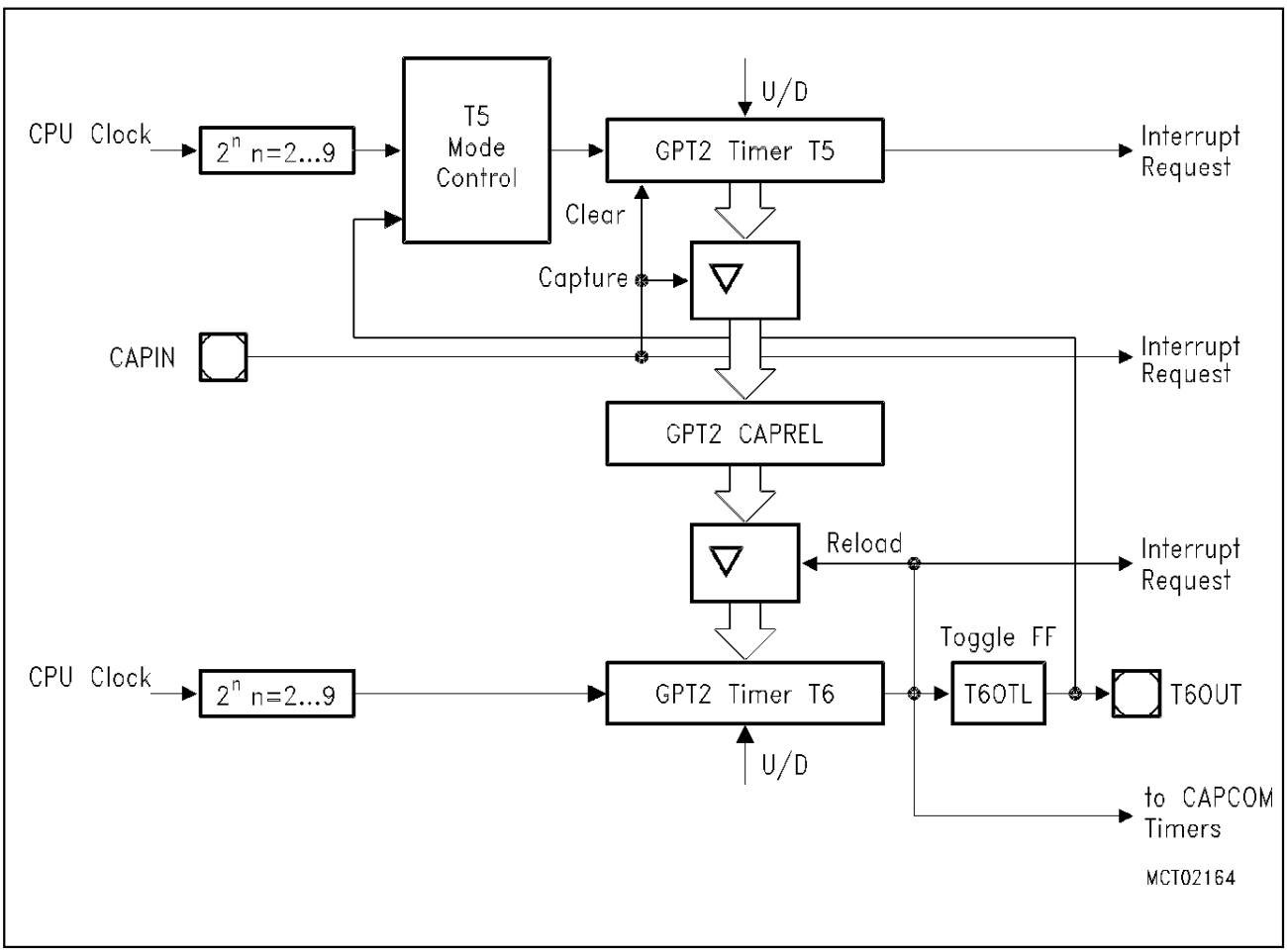


Figure 7
Block Diagram of GPT2

With its maximum resolution of 200 ns (@ 20 MHz), the GPT2 module provides precise event control and time measurement. It includes two timers (T5, T6) and a capture/reload register (CAPREL). Both timers can be clocked with an input clock which is derived from the CPU clock via a programmable prescaler. The count direction (up/down) for each timer is programmable by software. Concatenation of the timers is supported via the output toggle latch (T6OTL) of timer T6, which changes its state on each timer overflow/underflow.

The state of this latch may be used to clock timer T5, or it may be output on a port pin (T6OUT). The overflows/underflows of timer T6 can additionally be used to clock the CAPCOM timers T0 or T1, and to cause a reload from the CAPREL register. The CAPREL register may capture the contents of timer T5 based on an external signal transition on the corresponding port pin (CAPIN), and timer T5 may optionally be cleared after the capture procedure. This allows absolute time differences to be measured or pulse multiplication to be performed without software overhead.

A/D Converter

For analog signal measurement, a 10-bit A/D converter with 10 multiplexed input channels and a sample and hold circuit has been integrated on-chip. It uses the method of successive approximation. The sample time (for loading the capacitors) and the conversion time adds up to 9.7 us @ 20 MHz CPU clock.

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

For applications which require less than 10 analog input channels, the remaining channel inputs can be used as digital input port pins.

The A/D converter of the SAB 80C166 supports four different conversion modes. In the standard Single Channel conversion mode, the analog level on a specified channel is sampled once and converted to a digital result. In the Single Channel Continuous mode, the analog level on a specified channel is repeatedly sampled and converted without software intervention. In the Auto Scan mode, the analog levels on a prespecified number of channels are sequentially sampled and converted. In the Auto Scan Continuous mode, the number of prespecified channels is repeatedly sampled and converted.

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

Parallel Ports

The SAB 80C166 provides up to 76 I/O lines which are organized into five input/output ports and one input port. All port lines are bit-addressable, and all input/output lines are individually (bit-wise) programmable as inputs or outputs via direction registers. The I/O ports are true bidirectional ports which are switched to high impedance state when configured as inputs. During the internal reset, all port pins are configured as inputs.

All port lines have programmable alternate input or output functions associated with them. Port 0 and Port 1 may be used as address and data lines when accessing external memory, while Port 4 outputs the additional segment address bits A17/A16 in systems where segmentation is enabled to access more than 64 KBytes of memory. Port 2 is associated with the capture inputs or compare outputs of the CAPCOM unit and/or with optional bus arbitration signals ($\overline{\text{BREQ}}$, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$). Port 3 includes alternate functions of timers, serial interfaces, optional bus control signals ($\overline{\text{WR}}$, $\overline{\text{BHE}}$, $\overline{\text{READY}}$) and the system clock output (CLKOUT). Port 5 is used for the analog input channels to the A/D converter. All port lines that are not used for these alternate functions may be used as general purpose I/O lines.

Serial Channels

Serial communication with other microcontrollers, processors, terminals or external peripheral components is provided by two serial interfaces with identical functionality, Asynchronous/ Synchronous Serial Channels ASC0 and ASC1.

They are upward compatible with the serial ports of the Siemens SAB 8051x microcontroller family and support full-duplex asynchronous communication up to 625 Kbaud and half-duplex synchronous communication up to 2.5 Mbaud @ 20 MHz CPU clock.

Two dedicated baud rate generators allow to set up all standard baud rates without oscillator tuning. For transmission, reception, and erroneous reception 3 separate interrupt vectors are provided for each serial channel.

In asynchronous mode, 8- or 9-bit data frames are transmitted or received, preceded by a start bit and terminated by one or two stop bits. For multiprocessor communication, a mechanism to distinguish address from data bytes has been included (8-bit data + wake up bit mode).
In synchronous mode one data byte is transmitted or received synchronously to a shift clock which is generated by the SAB 80C166.

A loop back option is available for testing purposes.

A number of optional hardware error detection capabilities has been included to increase the reliability of data transfers. A parity bit can automatically be generated on transmission or be checked on reception. Framing error detection allows to recognize data frames with missing stop bits. An overrun error will be generated, if the last character received has not been read out of the receive buffer register at the time the reception of a new character is complete.

Watchdog Timer

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

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

The Watchdog Timer is a 16-bit timer, clocked with the CPU clock divided either by 2 or by 128. The high byte of the Watchdog Timer register can be set to a prespecified reload value (stored in WDTREL) in order to allow further variation of the monitored time interval. Each time it is serviced by the application software, the high byte of the Watchdog Timer is reloaded. Thus, time intervals between 25 µs and 420 ms can be monitored (@ 20 MHz CPU clock). The default Watchdog Timer interval after reset is 6.55 ms (@ 20 MHz CPU clock).

Bootstrap Loader

The SAB 80C166 provides a built-in bootstrap loader (BSL), which allows to start program execution out of the SAB 80C166's internal RAM. The program to be started is loaded via the serial interface ASC0 and does not require external memory or an internal ROM.

The SAB 80C166 enters BSL mode, when ALE is sampled high at the end of a hardware reset and if $\overline{\text{NMI}}$ becomes active directly after the end of the internal reset sequence. BSL mode is entered independent of the bus mode selected via EBC0, EBC1 and $\overline{\text{BUSACT}}$.

After entering BSL mode the SAB 80C166 scans the RXD0 line to receive a zero byte, i.e. one start bit, eight '0' data bits and one stop bit. From the duration of this zero byte it calculates the corresponding baudrate factor with respect to the current CPU clock and initializes ASC0 accordingly. Using this baudrate, an acknowledge byte is returned to the host that provides the loaded data. The SAB 80C166 returns the value $<55_H>$.

The next 32 bytes received via ASC0 are stored sequentially into locations $0FA40_H$ through $0FA5F_H$ of the internal RAM. To execute the loaded code the BSL then jumps to location $0FA40_H$. The loaded program may load additional code / data, change modes, etc.

The SAB 80C166 exits BSL mode upon a software reset (ignores the ALE level) or a hardware reset (remove conditions for entering BSL mode before).

Instruction Set Summary

The table below lists the instructions of the SAB 80C166 in a condensed way.
The various addressing modes that can be used with a specific instruction, the operation of the instructions, parameters for conditional execution of instructions, and the opcodes for each instruction can be found in the **“C16x Family Instruction Set Manual”**.

This document also provides a detailed description of each instruction.

Instruction Set Summary

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

Instruction Set Summary (cont'd)

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

Special Function Registers Overview

The following table lists all SFRs which are implemented in the SAB 80C166 in alphabetical order. Bit-addressable SFRs are marked with the letter “**b**” in column “Name”.

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

Special Function Registers Overview

Name	Physical Address	8-Bit Address	Description	Reset Value
ADCIC b	$FF98_H$	CC_H	A/D Converter End of Conversion Interrupt Control Register	0000_H
ADCON b	$FFA0_H$	$D0_H$	A/D Converter Control Register	0000_H
ADDAT	$FEA0_H$	50_H	A/D Converter Result Register	0000_H
ADDRSEL1	$FE18_H$	$0C_H$	Address Select Register 1	0000_H
ADEIC b	$FF9A_H$	CD_H	A/D Converter Overrun Error Interrupt Control Register	0000_H
BUSCON1 b	$FF14_H$	$8A_H$	Bus Configuration Register 1	0000_H
CAPREL	$FE4A_H$	25_H	GPT2 Capture/Reload Register	0000_H
CC0	$FE80_H$	40_H	CAPCOM Register 0	0000_H
CC0IC b	$FF78_H$	BC_H	CAPCOM Register 0 Interrupt Control Register	0000_H
CC1	$FE82_H$	41_H	CAPCOM Register 1	0000_H
CC1IC b	$FF7A_H$	BD_H	CAPCOM Register 1 Interrupt Control Register	0000_H
CC2	$FE84_H$	42_H	CAPCOM Register 2	0000_H
CC2IC b	$FF7C_H$	BE_H	CAPCOM Register 2 Interrupt Control Register	0000_H
CC3	$FE86_H$	43_H	CAPCOM Register 3	0000_H
CC3IC b	$FF7E_H$	BF_H	CAPCOM Register 3 Interrupt Control Register	0000_H
CC4	$FE88_H$	44_H	CAPCOM Register 4	0000_H
CC4IC b	$FF80_H$	$C0_H$	CAPCOM Register 4 Interrupt Control Register	0000_H
CC5	$FE8A_H$	45_H	CAPCOM Register 5	0000_H
CC5IC b	$FF82_H$	$C1_H$	CAPCOM Register 5 Interrupt Control Register	0000_H
CC6	$FE8C_H$	46_H	CAPCOM Register 6	0000_H
CC6IC b	$FF84_H$	$C2_H$	CAPCOM Register 6 Interrupt Control Register	0000_H
CC7	$FE8E_H$	47_H	CAPCOM Register 7	0000_H

Special Function Registers Overview (cont'd)

Name		Physical Address	8-Bit Address	Description	Reset Value
CC7IC	**b**	$FF86_H$	$C3_H$	CAPCOM Register 7 Interrupt Control Register	0000_H
CC8		$FE90_H$	48_H	CAPCOM Register 8	0000_H
CC8IC	**b**	$FF88_H$	$C4_H$	CAPCOM Register 8 Interrupt Control Register	0000_H
CC9		$FE92_H$	49_H	CAPCOM Register 9	0000_H
CC9IC	**b**	$FF8A_H$	$C5_H$	CAPCOM Register 9 Interrupt Control Register	0000_H
CC10		$FE94_H$	$4A_H$	CAPCOM Register 10	0000_H
CC10IC	**b**	$FF8C_H$	$C6_H$	CAPCOM Register 10 Interrupt Control Register	0000_H
CC11		$FE96_H$	$4B_H$	CAPCOM Register 11	0000_H
CC11IC	**b**	$FF8E_H$	$C7_H$	CAPCOM Register 11 Interrupt Control Register	0000_H
CC12		$FE98_H$	$4C_H$	CAPCOM Register 12	0000_H
CC12IC	**b**	$FF90_H$	$C8_H$	CAPCOM Register 12 Interrupt Control Register	0000_H
CC13		$FE9A_H$	$4D_H$	CAPCOM Register 13	0000_H
CC13IC	**b**	$FF92_H$	$C9_H$	CAPCOM Register 13 Interrupt Control Register	0000_H
CC14		$FE9C_H$	$4E_H$	CAPCOM Register 14	0000_H
CC14IC	**b**	$FF94_H$	CA_H	CAPCOM Register 14 Interrupt Control Register	0000_H
CC15		$FE9E_H$	$4F_H$	CAPCOM Register 15	0000_H
CC15IC	**b**	$FF96_H$	CB_H	CAPCOM Register 15 Interrupt Control Register	0000_H
CCM0	**b**	$FF52_H$	$A9_H$	CAPCOM Mode Control Register 0	0000_H
CCM1	**b**	$FF54_H$	AA_H	CAPCOM Mode Control Register 1	0000_H
CCM2	**b**	$FF56_H$	AB_H	CAPCOM Mode Control Register 2	0000_H
CCM3	**b**	$FF58_H$	AC_H	CAPCOM Mode Control Register 3	0000_H
CP		$FE10_H$	08_H	CPU Context Pointer Register	$FC00_H$
CRIC	**b**	$FF6A_H$	$B5_H$	GPT2 CAPREL Interrupt Control Register	0000_H
CSP		$FE08_H$	04_H	CPU Code Segment Pointer Register (2 bits, read only)	0000_H
DP0	**b**	$FF02_H$	81_H	Port 0 Direction Control Register	0000_H
DP1	**b**	$FF06_H$	83_H	Port 1 Direction Control Register	0000_H
DP2	**b**	$FFC2_H$	$E1_H$	Port 2 Direction Control Register	0000_H
DP3	**b**	$FFC6_H$	$E3_H$	Port 3 Direction Control Register	0000_H
DP4	**b**	$FF0A_H$	85_H	Port 4 Direction Control Register (2 bits)	00_H

Special Function Registers Overview (cont'd)

Name		Physical Address	8-Bit Address	Description	Reset Value
DPP0		$FE00_H$	00_H	CPU Data Page Pointer 0 Register (4 bits)	0000_H
DPP1		$FE02_H$	01_H	CPU Data Page Pointer 1 Register (4 bits)	0001_H
DPP2		$FE04_H$	02_H	CPU Data Page Pointer 2 Register (4 bits)	0002_H
DPP3		$FE06_H$	03_H	CPU Data Page Pointer 3 Register (4 bits)	0003_H
MDC	**b**	$FF0E_H$	87_H	CPU Multiply / Divide Control Register	0000_H
MDH		$FE0C_H$	06_H	CPU Multiply / Divide Register – High Word	0000_H
MDL		$FE0E_H$	07_H	CPU Multiply / Divide Register – Low Word	0000_H
ONES		$FF1E_H$	$8F_H$	Constant Value 1's Register (read only)	$FFFF_H$
P0	**b**	$FF00_H$	80_H	Port 0 Register	0000_H
P1	**b**	$FF04_H$	82_H	Port 1 Register	0000_H
P2	**b**	$FFC0_H$	$E0_H$	Port 2 Register	0000_H
P3	**b**	$FFC4_H$	$E2_H$	Port 3 Register	0000_H
P4	**b**	$FFC8_H$	$E4_H$	Port 4 Register (2 bits)	00_H
P5	**b**	$FFA2_H$	$D1_H$	Port 5 Register (read only)	$XXXX_H$
PECC0		$FEC0_H$	60_H	PEC Channel 0 Control Register	0000_H
PECC1		$FEC2_H$	61_H	PEC Channel 1 Control Register	0000_H
PECC2		$FEC4_H$	62_H	PEC Channel 2 Control Register	0000_H
PECC3		$FEC6_H$	63_H	PEC Channel 3 Control Register	0000_H
PECC4		$FEC8_H$	64_H	PEC Channel 4 Control Register	0000_H
PECC5		$FECA_H$	65_H	PEC Channel 5 Control Register	0000_H
PECC6		$FECC_H$	66_H	PEC Channel 6 Control Register	0000_H
PECC7		$FECE_H$	67_H	PEC Channel 7 Control Register	0000_H
PSW	**b**	$FF10_H$	88_H	CPU Program Status Word	0000_H
S0BG		$FEB4_H$	$5A_H$	Serial Channel 0 Baud Rate Generator Reload Register	0000_H
S0CON	**b**	$FFB0_H$	$D8_H$	Serial Channel 0 Control Register	0000_H
S0EIC	**b**	$FF70_H$	$B8_H$	Serial Channel 0 Error Interrupt Control Register	0000_H
S0RBUF		$FEB2_H$	59_H	Serial Channel 0 Receive Buffer Register (read only)	XX_H

Special Function Registers Overview (cont'd)

Name		Physical Address	8-Bit Address	Description	Reset Value
S0RIC	**b**	$FF6E_H$	$B7_H$	Serial Channel 0 Receive Interrupt Control Register	0000_H
S0TBUF		$FEB0_H$	58_H	Serial Channel 0 Transmit Buffer Register (write only)	00_H
S0TIC	**b**	$FF6C_H$	$B6_H$	Serial Channel 0 Transmit Interrupt Control Register	0000_H
S1BG		$FEBC_H$	$5E_H$	Serial Channel 1 Baud Rate Generator Reload Register	0000_H
S1CON	**b**	$FFB8_H$	DC_H	Serial Channel 1 Control Register	0000_H
S1EIC	**b**	$FF76_H$	BB_H	Serial Channel 1 Error Interrupt Control Register	0000_H
S1RBUF		$FEBA_H$	$5D_H$	Serial Channel 1 Receive Buffer Register (read only)	XX_H
S1RIC	**b**	$FF74_H$	BA_H	Serial Channel 1 Receive Interrupt Control Register	0000_H
S1TBUF		$FEB8_H$	$5C_H$	Serial Channel 1 Transmit Buffer Register (write only)	00_H
S1TIC	**b**	$FF72_H$	$B9_H$	Serial Channel 1 Transmit Interrupt Control Register	0000_H
SP		$FE12_H$	09_H	CPU System Stack Pointer Register	$FC00_H$
STKOV		$FE14_H$	$0A_H$	CPU Stack Overflow Pointer Register	$FA00_H$
STKUN		$FE16_H$	$0B_H$	CPU Stack Underflow Pointer Register	$FC00_H$
SYSCON	**b**	$FF0C_H$	86_H	CPU System Configuration Register	$0xx0_H$*)
T0		$FE50_H$	28_H	CAPCOM Timer 0 Register	0000_H
T01CON	**b**	$FF50_H$	$A8_H$	CAPCOM Timer 0 and Timer 1 Control Register	0000_H
T0IC	**b**	$FF9C_H$	CE_H	CAPCOM Timer 0 Interrupt Control Register	0000_H
T0REL		$FE54_H$	$2A_H$	CAPCOM Timer 0 Reload Register	0000_H
T1		$FE52_H$	29_H	CAPCOM Timer 1 Register	0000_H
T1IC	**b**	$FF9E_H$	CF_H	CAPCOM Timer 1 Interrupt Control Register	0000_H
T1REL		$FE56_H$	$2B_H$	CAPCOM Timer 1 Reload Register	0000_H
T2		$FE40_H$	20_H	GPT1 Timer 2 Register	0000_H
T2CON	**b**	$FF40_H$	$A0_H$	GPT1 Timer 2 Control Register	0000_H
T2IC	**b**	$FF60_H$	$B0_H$	GPT1 Timer 2 Interrupt Control Register	0000_H

Special Function Registers Overview (cont'd)

Name		Physical Address	8-Bit Address	Description	Reset Value
T3		$FE42_H$	21_H	GPT1 Timer 3 Register	0000_H
T3CON	**b**	$FF42_H$	$A1_H$	GPT1 Timer 3 Control Register	0000_H
T3IC	**b**	$FF62_H$	$B1_H$	GPT1 Timer 3 Interrupt Control Register	0000_H
T4		$FE44_H$	22_H	GPT1 Timer 4 Register	0000_H
T4CON	**b**	$FF44_H$	$A2_H$	GPT1 Timer 4 Control Register	0000_H
T4IC	**b**	$FF64_H$	$B2_H$	GPT1 Timer 4 Interrupt Control Register	0000_H
T5		$FE46_H$	23_H	GPT2 Timer 5 Register	0000_H
T5CON	**b**	$FF46_H$	$A3_H$	GPT2 Timer 5 Control Register	0000_H
T5IC	**b**	$FF66_H$	$B3_H$	GPT2 Timer 5 Interrupt Control Register	0000_H
T6		$FE48_H$	24_H	GPT2 Timer 6 Register	0000_H
T6CON	**b**	$FF48_H$	$A4_H$	GPT2 Timer 6 Control Register	0000_H
T6IC	**b**	$FF68_H$	$B4_H$	GPT2 Timer 6 Interrupt Control Register	0000_H
TFR	**b**	$FFAC_H$	$D6_H$	Trap Flag Register	0000_H
WDT		$FEAE_H$	57_H	Watchdog Timer Register (read only)	0000_H
WDTCON		$FFAE_H$	$D7_H$	Watchdog Timer Control Register	0000_H
ZEROS	**b**	$FF1C_H$	$8E_H$	Constant Value 0's Register (read only)	0000_H

*) The system configuration is selected during reset.

Absolute Maximum Ratings

Ambient temperature under bias (T_A):
SAB 83C166-5M, SAB 80C166-M ..... 0 to + 70 ˚C
SAB 83C166-5M-T3, SAB 80C166-M-T3 ..... – 40 to + 85 ˚C
Storage temperature (T_{ST}) ..... – 65 to + 150 ˚C
Voltage on V_{CC} pins with respect to ground (V_{SS}) ..... – 0.5 to + 6.5 V
Voltage on any pin with respect to ground (V_{SS}) ..... – 0.5 to V_{CC} + 0.5 V
Input current on any pin during overload condition ..... – 10 to + 10 mA
Absolute sum of all input currents during overload condition ..... |100 mA|
Power dissipation ..... 1 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 SAB 80C166 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 SAB 80C166 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 SAB 80C166.

DC Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 ˚C for SAB 83C166-5M, SAB 80C166-M
T_A = -40 to +85 ˚C for SAB 83C166-5M-T3, SAB 80C166-M-T3

Parameter	Symbol		Limit Values		Unit	Test Condition
			min.	max.		
Input low voltage	V_{IL}	SR	– 0.5	0.2 V_{CC} – 0.1	V	–
Input high voltage (all except $\overline{\text{RSTIN}}$ and XTAL1)	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	–
Output low voltage (Port 0, Port 1, 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 (Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$)	V_{OH}	CC	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 500 µA I_{OH} = – 2.4 mA
Output high voltage (all other outputs)	V_{OH1}	CC	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 250 µA I_{OH} = – 1.6 mA
Input leakage current (Port 5) [1]	I_{OZ1}	CC	–	±200	nA	0 V < V_{IN} < V_{CC}
Input leakage current (all other)	I_{OZ2}	CC	–	±500	nA	0 V < V_{IN} < V_{CC}
$\overline{\text{RSTIN}}$ pullup resistor	R_{RST}	CC	50	150	kΩ	–
Read inactive current [4]	I_{RH}	[2]	–	-40	µA	V_{OUT} = V_{OHmin}
Read active current [4]	I_{RL}	[3]	-500	–	µA	V_{OUT} = V_{OLmax}
ALE inactive current [4]	I_{ALEL}	[2]	–	150	µA	V_{OUT} = V_{OLmax}
ALE active current [4]	I_{ALEH}	[3]	2100	–	µA	V_{OUT} = V_{OHmin}
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 T_A = 25 ˚C
Power supply current	I_{CC}		–	50 + 5 * f_{CPU}	mA	Reset active f_{CPU} in [MHz] [6]
Idle mode supply current	I_{ID}		–	30 + 1.5 * f_{CPU}	mA	f_{CPU} in [MHz] [6]
Power-down mode supply current	I_{PD}		–	50	µA	V_{CC} = 5.5 V [7]

Notes

1) This specification does not apply to the analog input (Port 5.x) which is currently converted.

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-mode.

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

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 open.

7) 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.
A voltage of $V_{CC} \geq 2.5$ V is sufficient to retain the content of the internal RAM during power down mode.

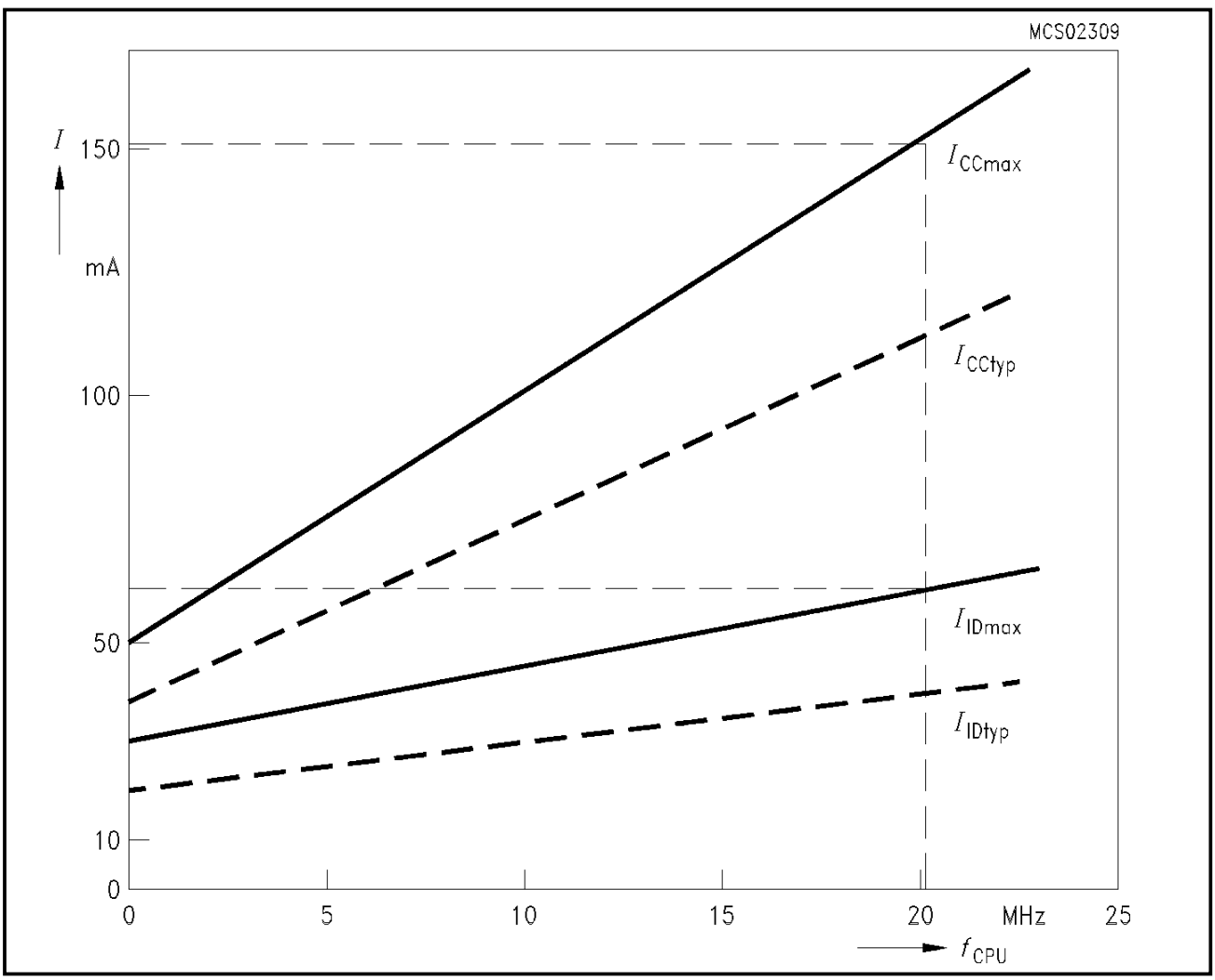


Figure 8
Supply/Idle Current as a Function of Operating Frequency

A/D Converter Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 ˚C for SAB 83C166-5M, SAB 80C166-M
T_A = -40 to +85 ˚C for SAB 83C166-5M-T3, SAB 80C166-M-T3
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	–	10 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} / 250 - 0.25	kΩ	t_{CC} in [ns] 6) 7)
Internal resistance of analog source	R_{ASRC}	SR	–	t_S / 500 - 0.25	kΩ	t_S in [ns] 2) 7)
ADC input capacitance	C_{AIN}	CC	–	50	pF	7)

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 capacitors to reach their 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. The value for the sample clock is t_{SC} = TCL * 32.

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.
The value for the conversion clock is t_{CC} = TCL * 32.

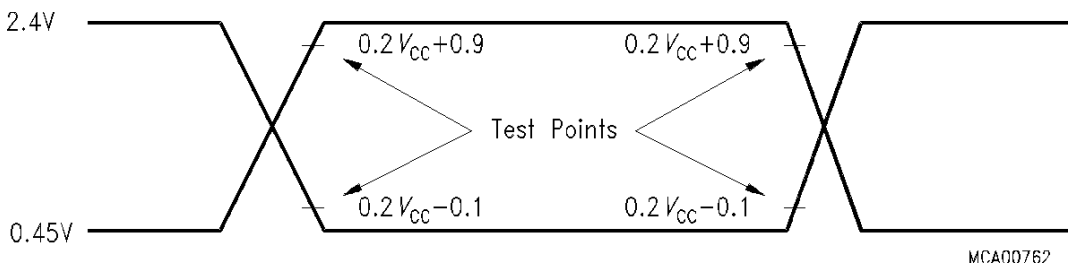
4) This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.

5) TUE is tested at V_{AREF} = 5.0V, V_{AGND} = 0 V, V_{CC} = 4.8 V. It is guaranteed by design characterization for all other voltages within the defined voltage range.

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 capacitors to reach their respective voltage level within t_{CC}. The maximum internal resistance results from the CPU clock period.

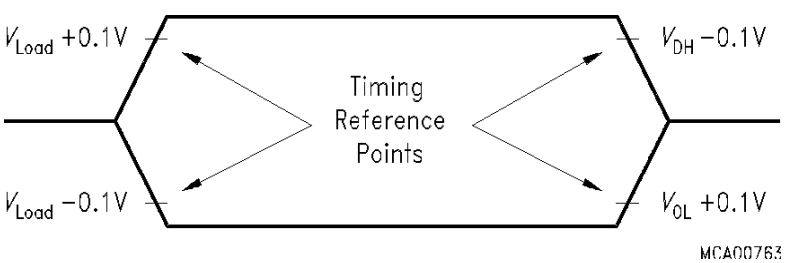
7) Not 100% tested, guaranteed by design characterization.

Testing Waveforms



AC inputs during testing are driven at 2.4 V for a logic ‘1’ and 0.4 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 9
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 10
Float Waveforms

AC Characteristics

External Clock Drive XTAL1

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 ˚C for SAB 83C166-5M, SAB 80C166-M
T_A = -40 to +85 ˚C for SAB 83C166-5M-T3, SAB 80C166-M-T3

Parameter	Symbol		Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
Oscillator period	TCL	SR	25	25	25	500	ns
High time	t_1	SR	6	–	6	–	ns
Low time	t_2	SR	6	–	6	–	ns
Rise time	t_3	SR	–	5	–	5	ns
Fall time	t_4	SR	–	5	–	5	ns

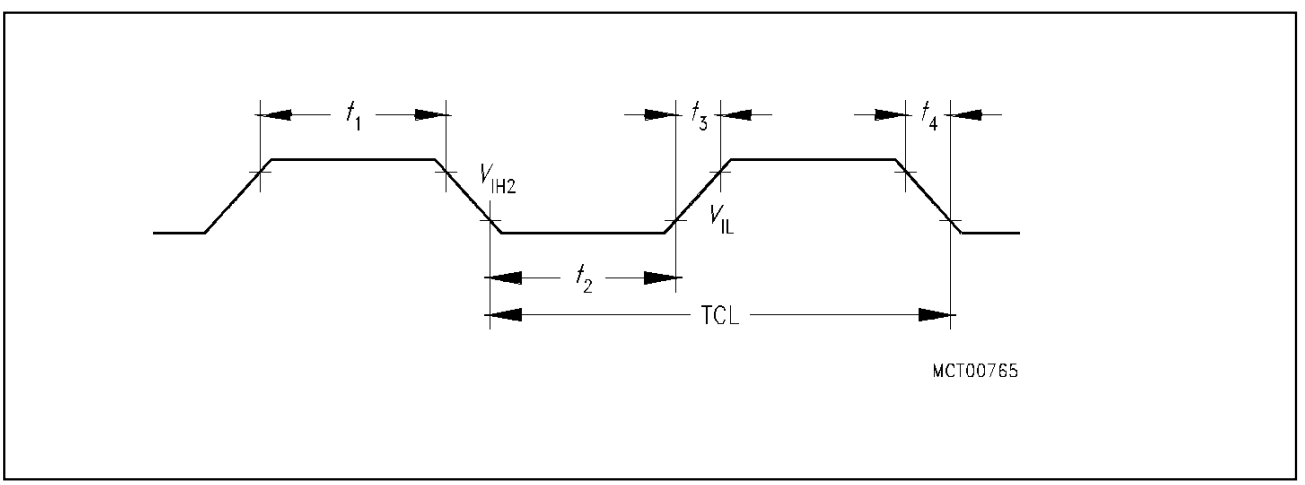


Figure 11
External Clock Drive XTAL1

Memory Cycle Variables

The timing tables below use three variables which are derived from registers SYSCON and BUSCON1 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 (cont'd)

Multiplexed Bus

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 ˚C for SAB 83C166-5M, SAB 80C166-M
T_A = -40 to +85 ˚C for SAB 83C166-5M-T3, SAB 80C166-M-T3
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 6 TCL + 2t_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
$\overline{RD}$ 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	–	75 + 2t_A + t_C	–	4TCL - 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}	SR	0	–	0	–	ns
Data float after $\overline{RD}$	t_{19}	SR	–	35 + t_F	–	2TCL - 15 + t_F	ns
Data valid to $\overline{WR}$	t_{22}	CC	35 + t_C	–	2TCL - 15 + t_C	–	ns

Parameter	Symbol		Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
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 $\overline{RD}$, $\overline{WR}$	t_{27}	CC	35 + t_F	–	2TCL - 15 + t_F	–	ns

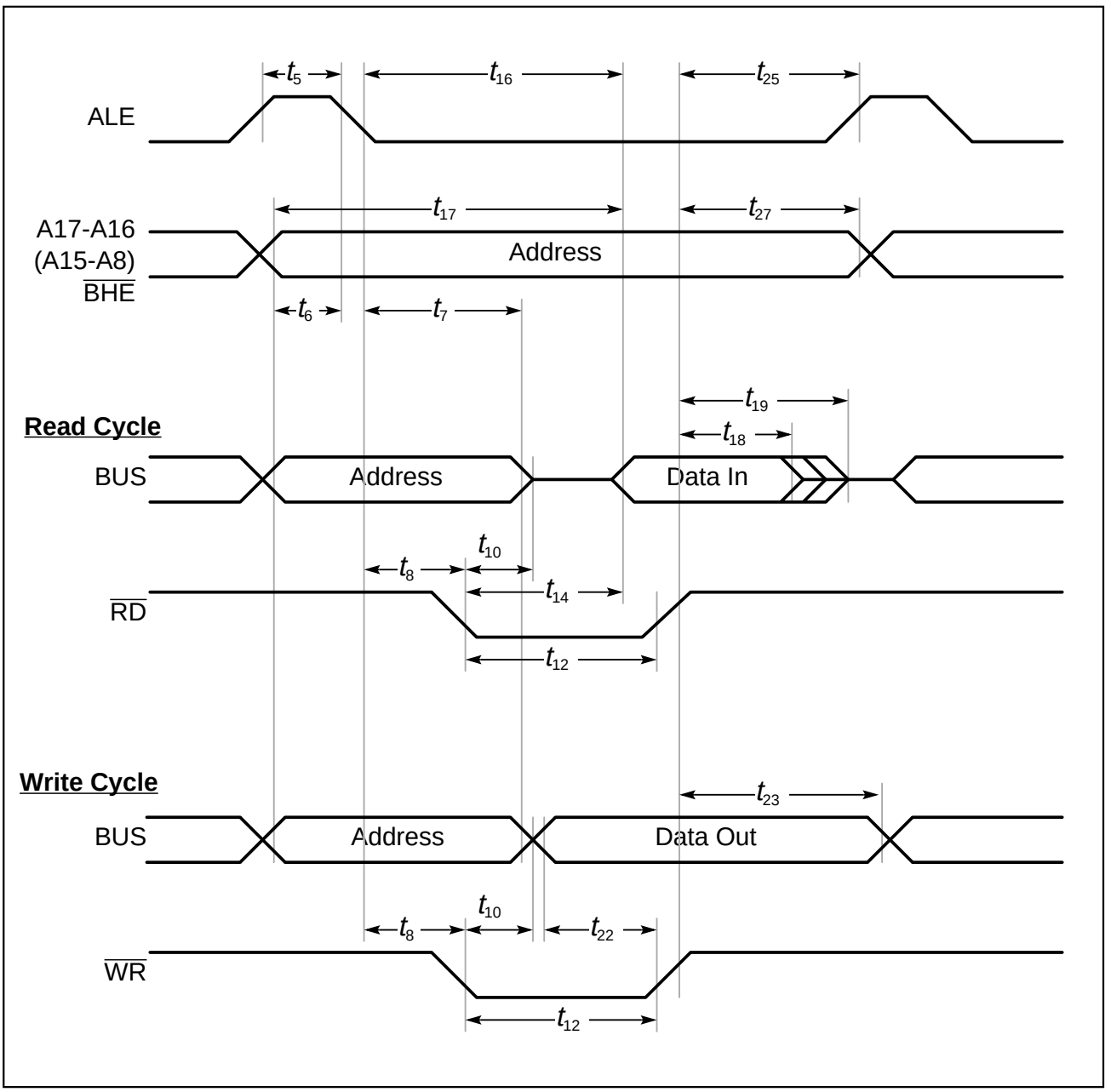


Figure 12-1
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE

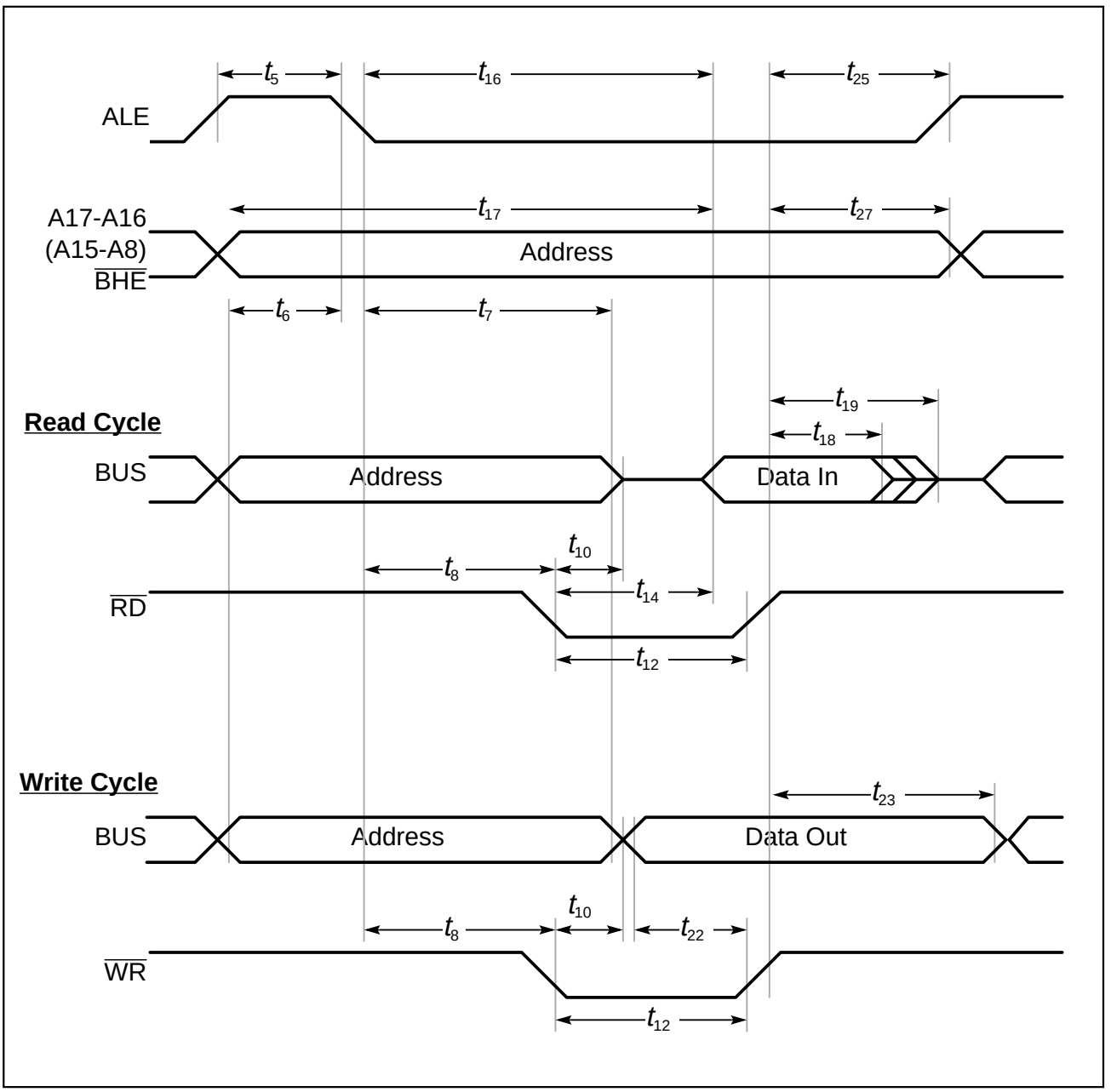


Figure 12-2
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Extended ALE

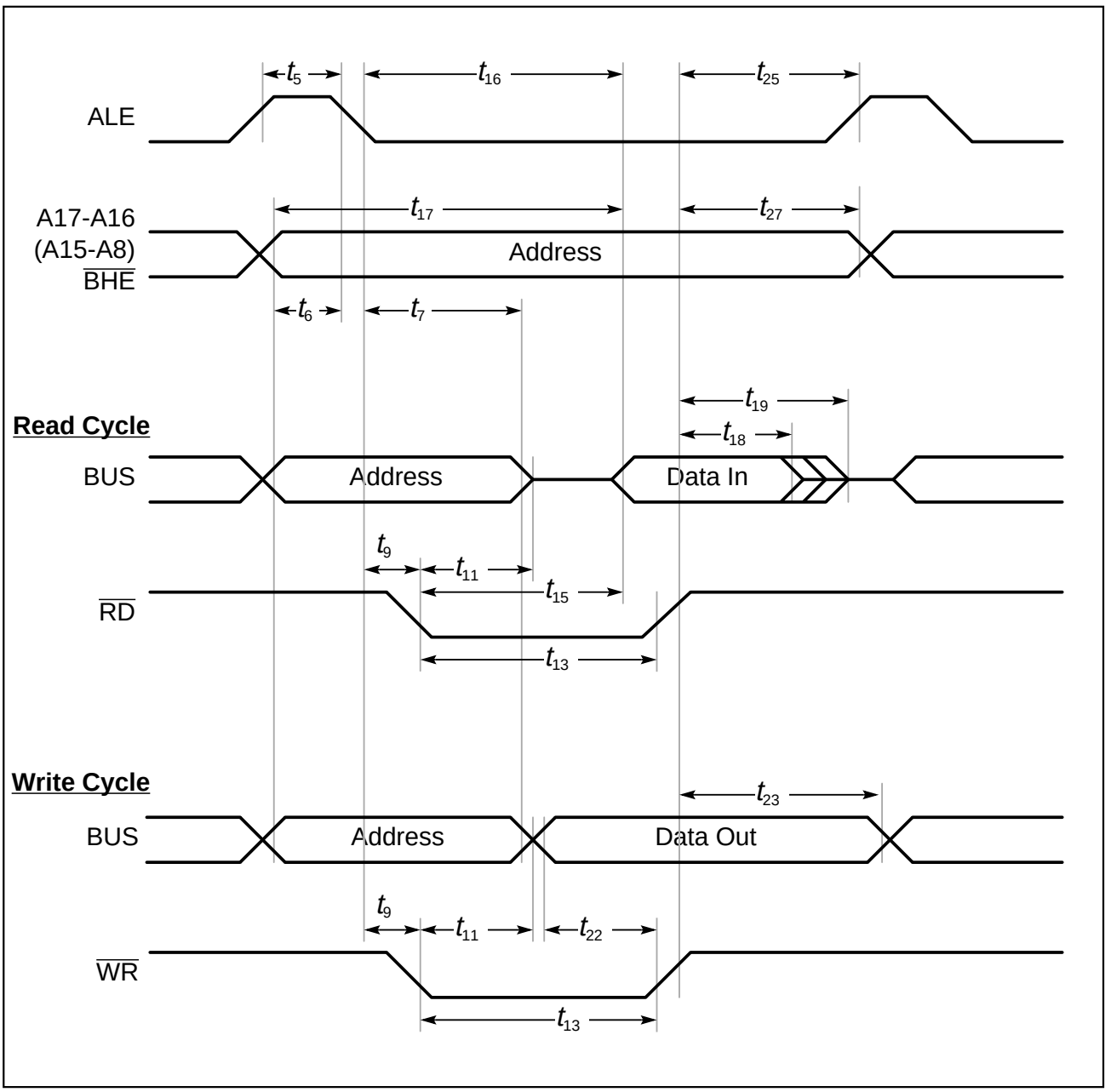


Figure 12-3
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE

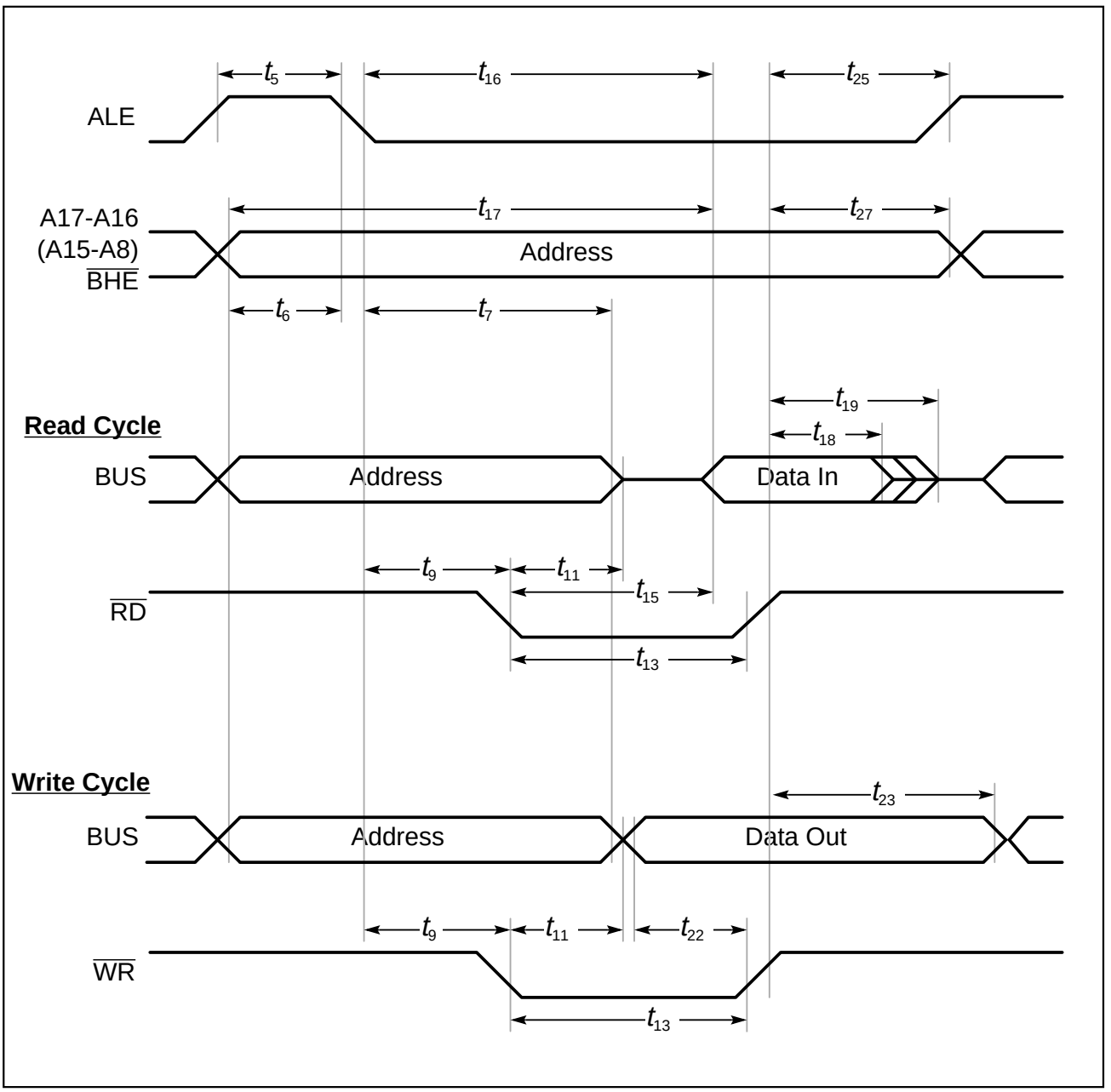


Figure 12-4
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

Demultiplexed Bus

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 ˚C for SAB 83C166-5M, SAB 80C166-M
T_A = -40 to +85 ˚C for SAB 83C166-5M-T3, SAB 80C166-M-T3
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 4 TCL + 2t_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	–	75 + 2t_A + t_C	–	4TCL - 25 + 2t_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)	t_{20}	SR	–	35 + t_F	–	2TCL - 15 + t_F	ns
Data float after $\overline{RD}$ rising edge (no RW-delay)	t_{21}	SR	–	15 + t_F	–	TCL - 10 + t_F	ns
Data valid to $\overline{WR}$	t_{22}	CC	35 + t_C	–	2TCL - 15 + t_C	–	ns
Data hold after $\overline{WR}$	t_{24}	CC	15 + t_F	–	TCL - 10 + t_F	–	ns

Parameter	Symbol		Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
ALE rising edge after $\overline{RD}$, $\overline{WR}$	t_{26}	CC	-10 + t_F	–	-10 + t_F	–	ns
Address hold after $\overline{RD}$, $\overline{WR}$	t_{28}	CC	0 + t_F	–	0 + t_F	–	ns

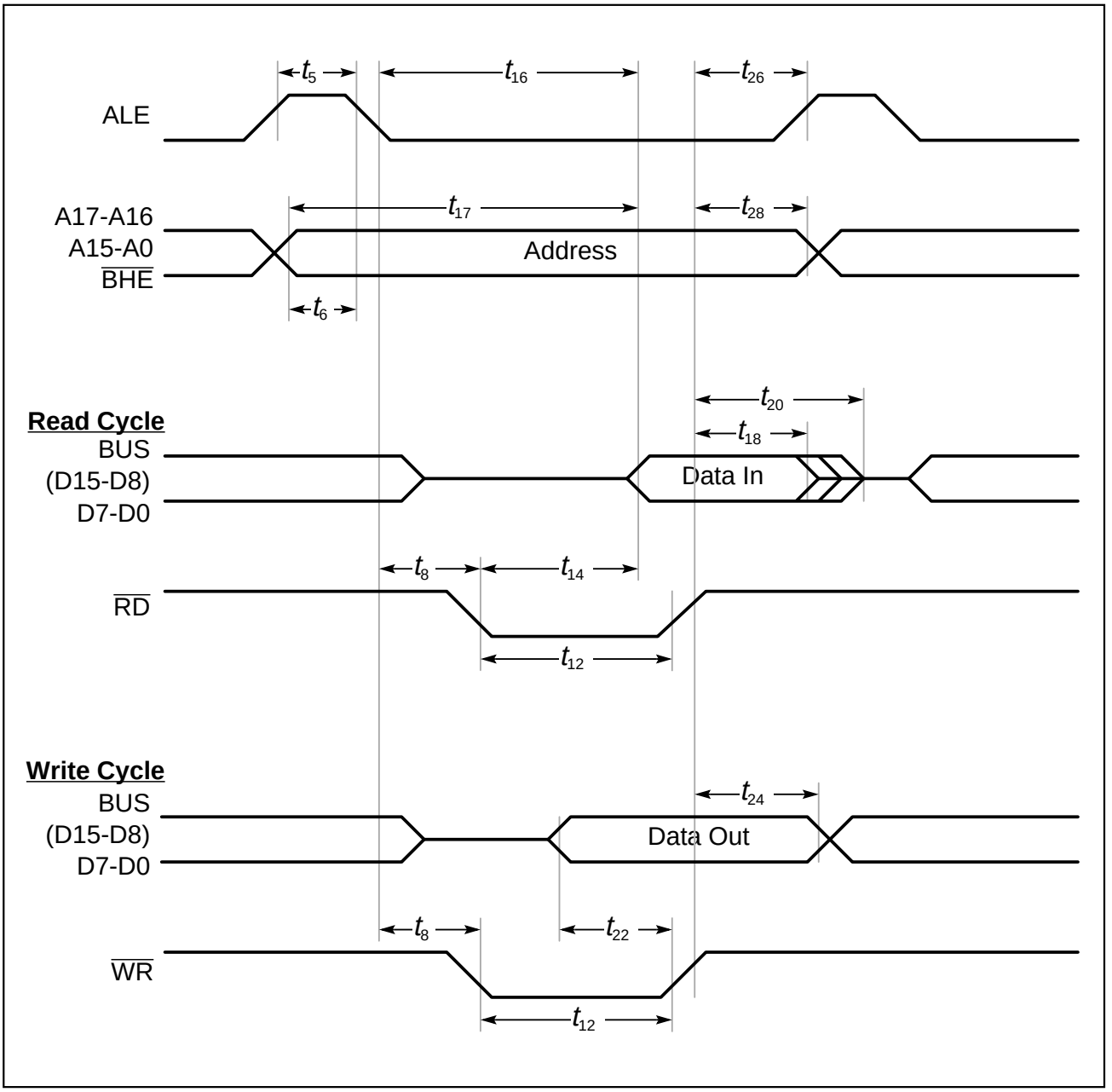


Figure 13-1
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Normal ALE

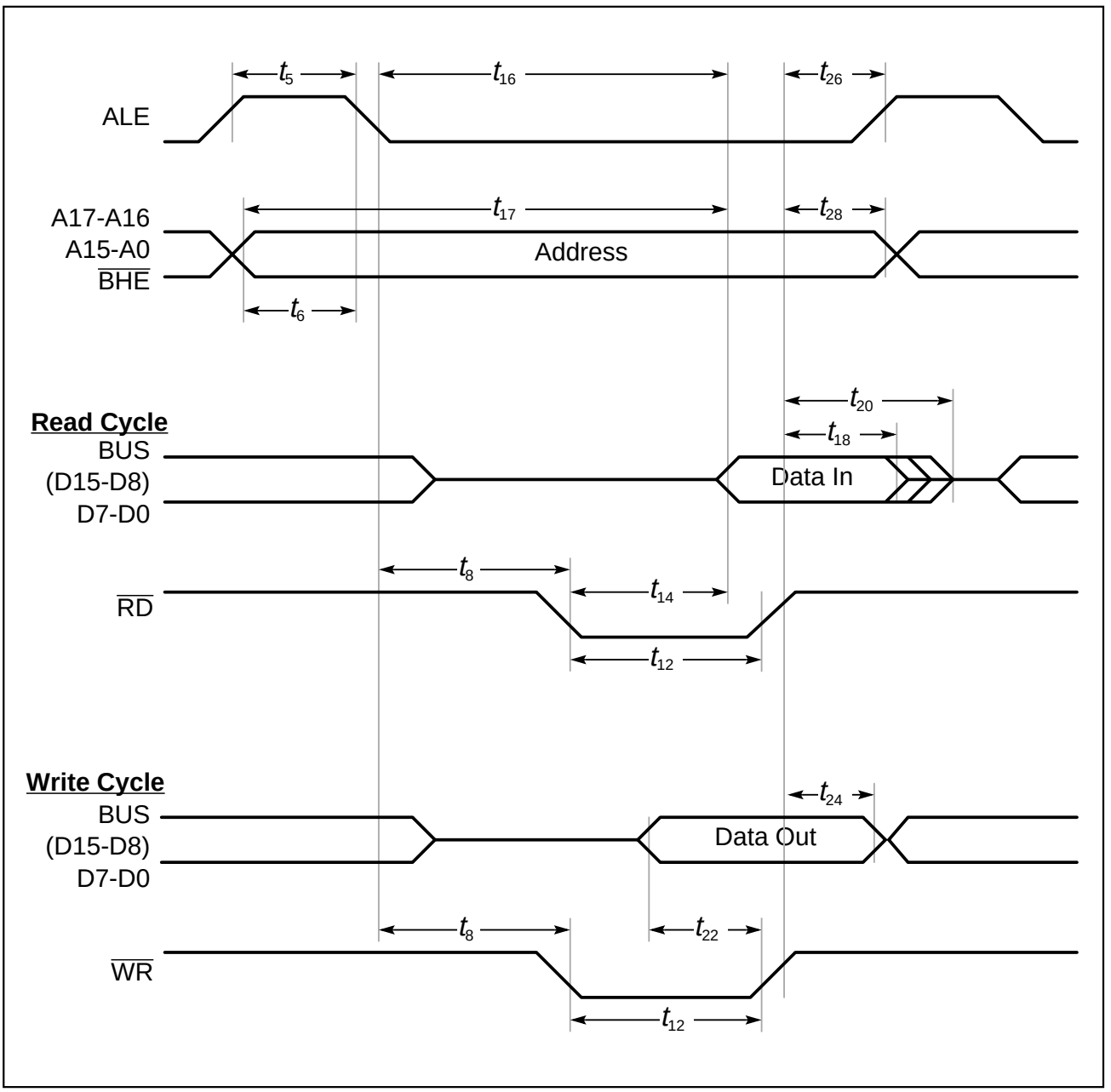


Figure 13-2
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE

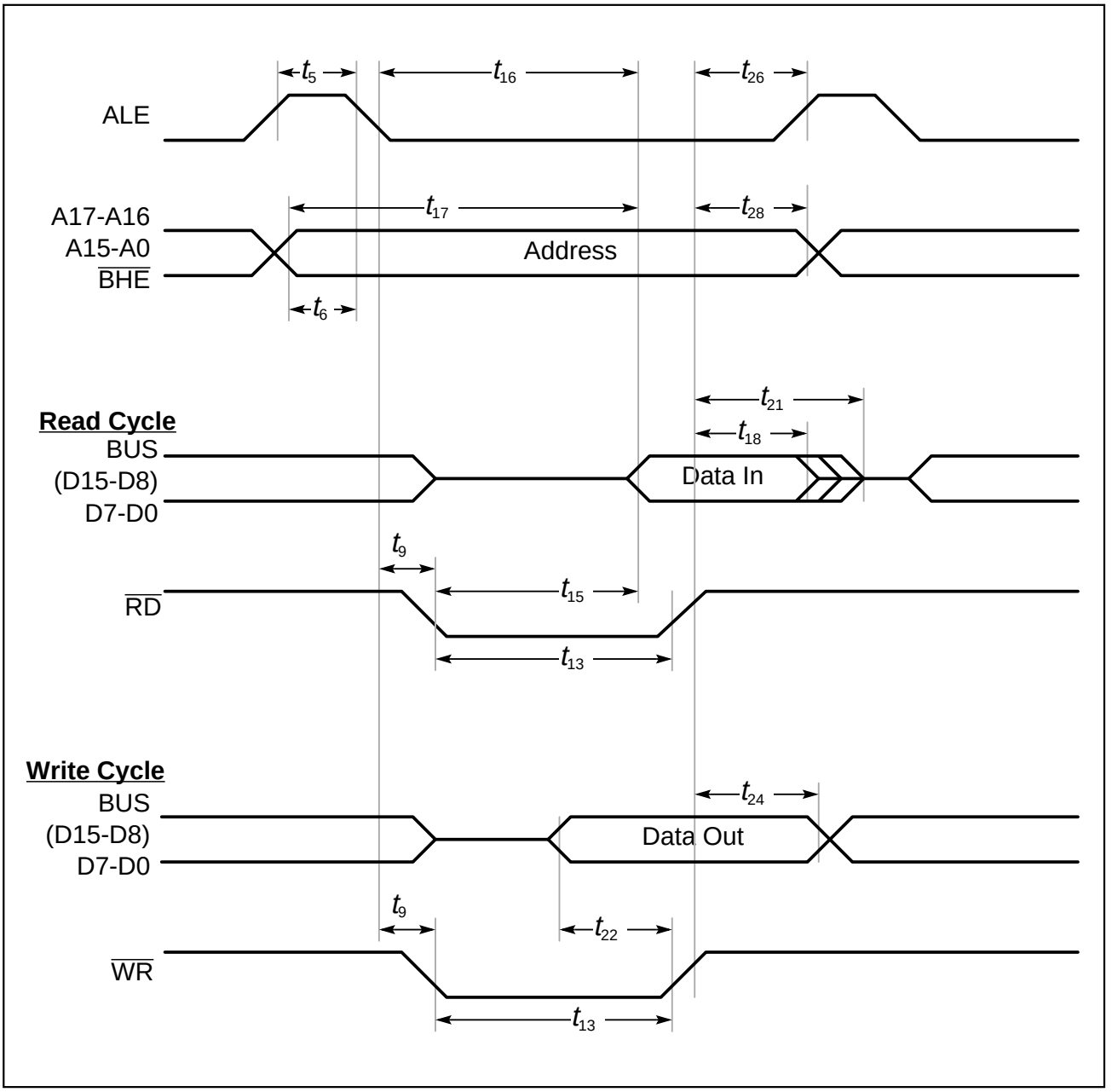


Figure 13-3
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE

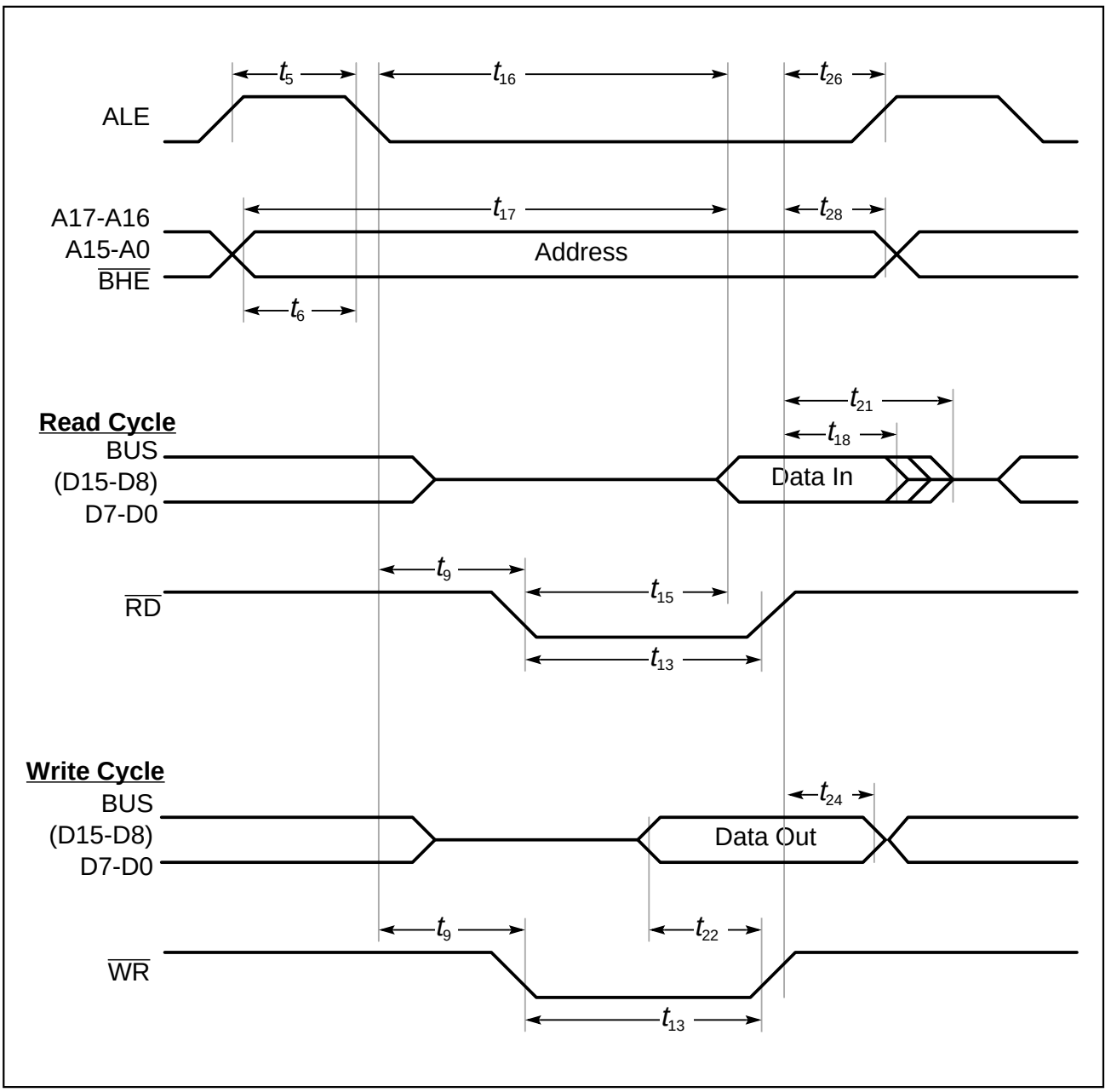


Figure 13-4
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

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

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB 83C166-5M, SAB 80C166-M
T_A = -40 to +85 °C for SAB 83C166-5M-T3, SAB 80C166-M-T3
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT) = 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	10	–	10	–	ns
Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	t_{36}	SR	10	–	10	–	ns
Asynchronous $\overline{\text{READY}}$ low time	t_{37}	SR	65	–	2TCL + 15	–	ns
Asynchronous $\overline{\text{READY}}$ setup time [1]	t_{58}	SR	20	–	20	–	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 + 2t_A + t_F [2]	0	TCL - 25 + 2t_A + 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 $\overline{\text{READY}}$.

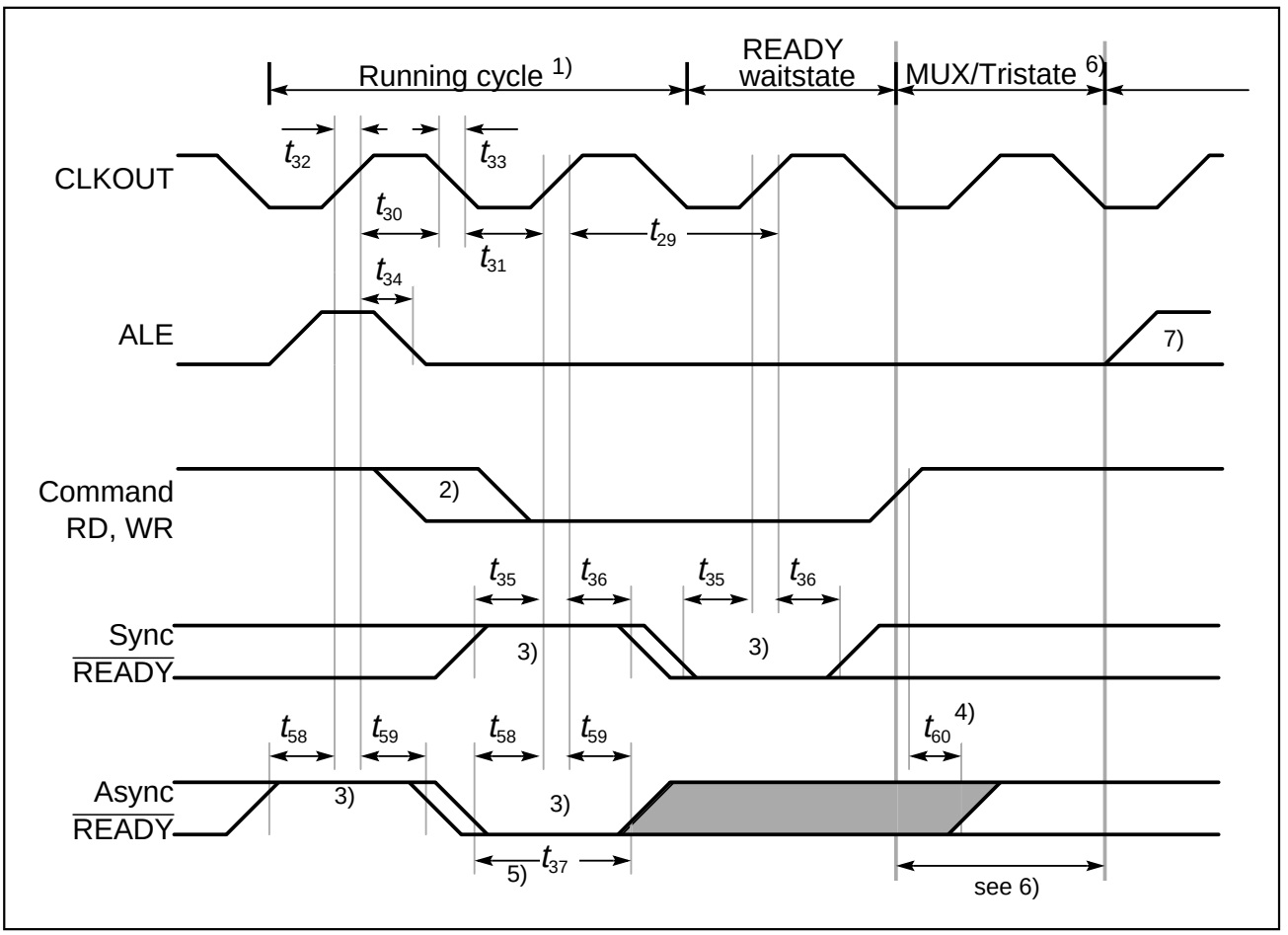


Figure 14
CLKOUT and $\overline{\text{READY}}$

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 (e.g. 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 response 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 (cont'd)
External Bus Arbitration

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB 83C166-5M, SAB 80C166-M
T_A = -40 to +85 °C for SAB 83C166-5M-T3, SAB 80C166-M-T3
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 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	–	50	–	50	ns
CLKOUT to $\overline{HLDA}$ low or $\overline{BREQ}$ high delay	t_{63}	CC	–	50	–	50	ns
Other signals release	t_{66}	CC	–	25	–	25	ns
Other signals drive	t_{67}	CC	-5	35	-5	35	ns

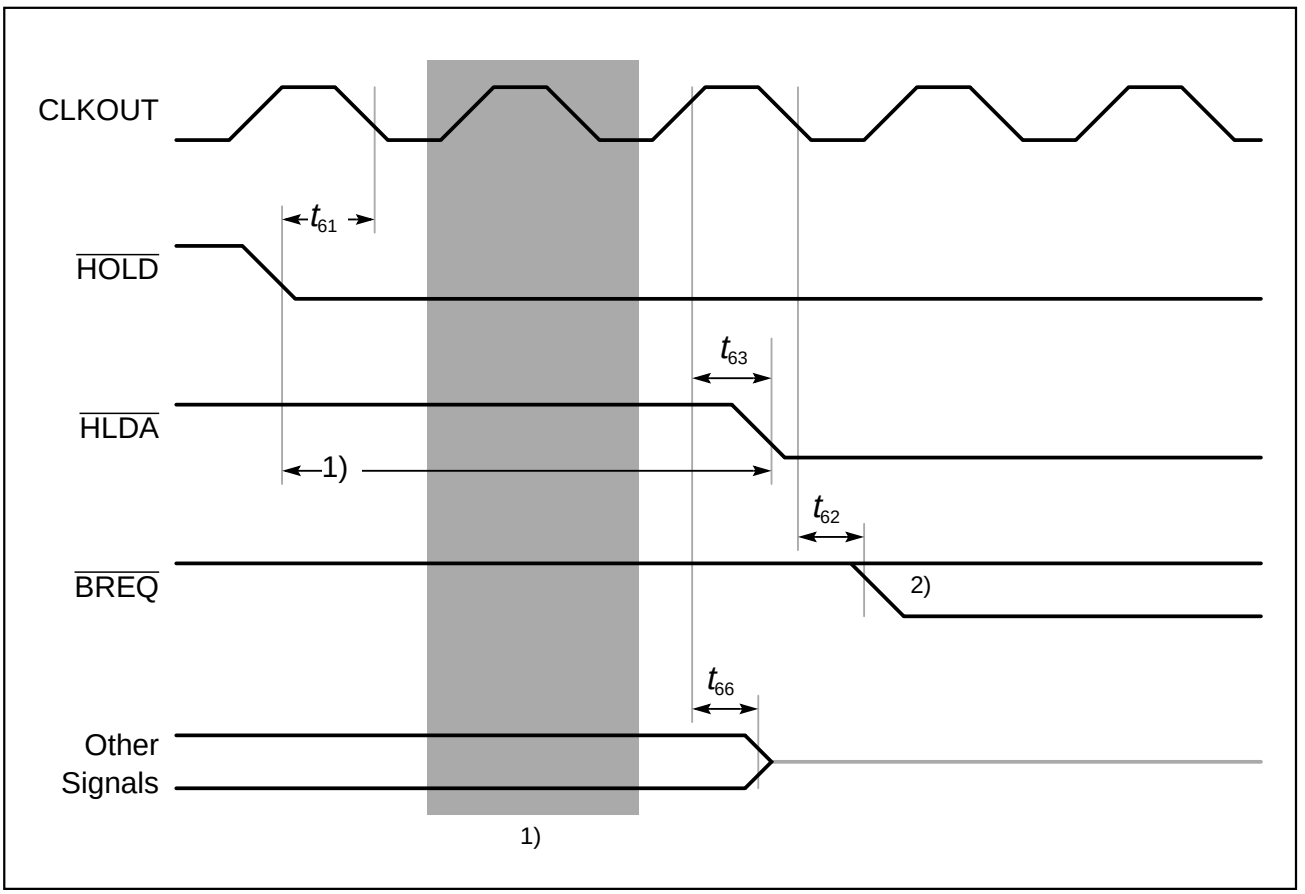


Figure 15
External Bus Arbitration, Releasing the Bus

Notes

1) The SAB 80C166 will complete the currently running bus cycle before granting bus access.

2) This is the first possibility for $\overline{\text{BREQ}}$ to get active.

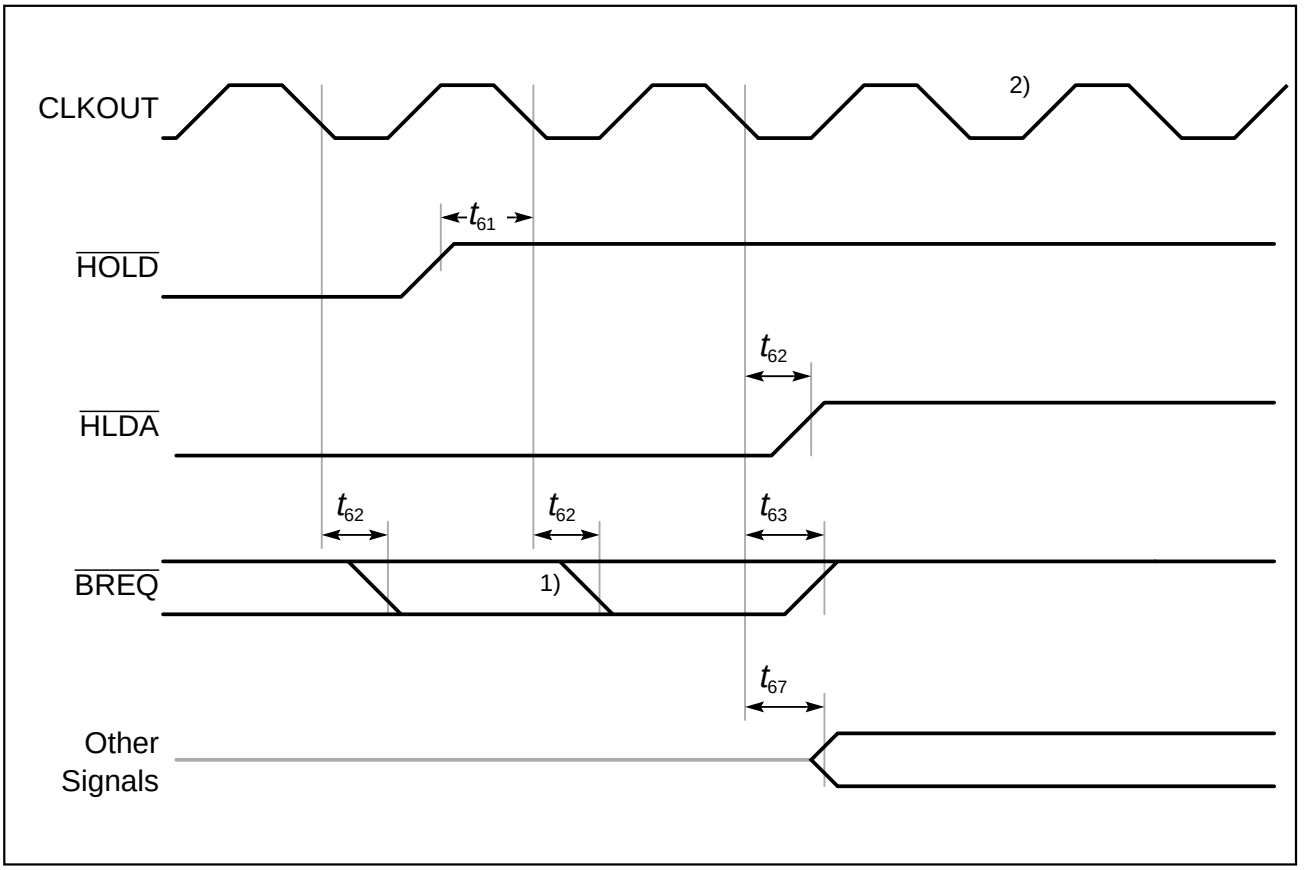


Figure 16
External Bus Arbitration, (Regaining the Bus)

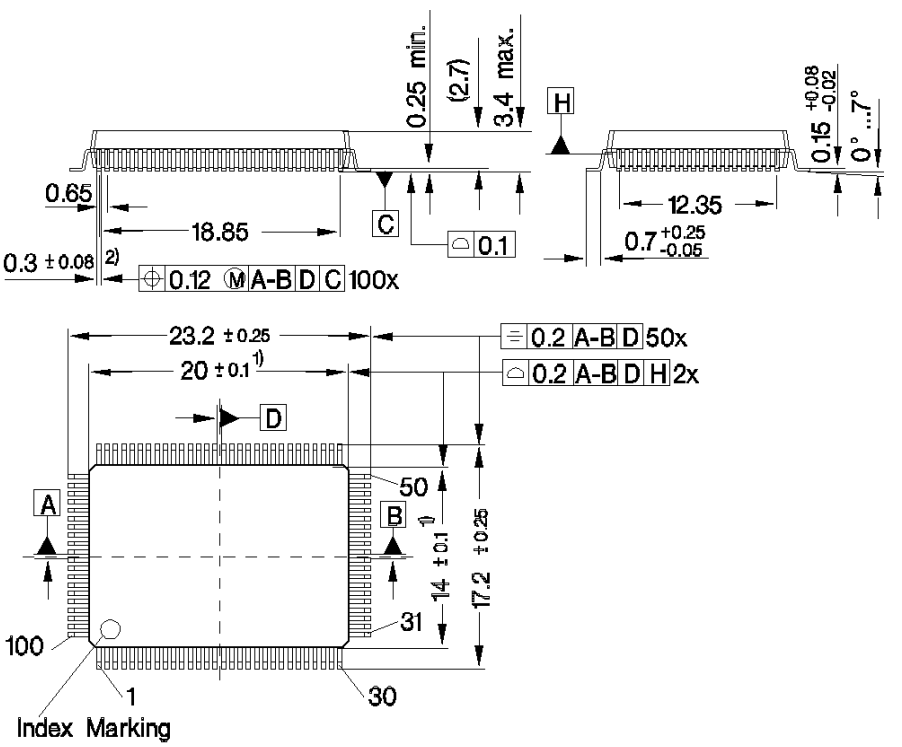
Notes

1) This is the last chance for $\overline{\text{BREQ}}$ to trigger the indicated regain-sequence.
Even if $\overline{\text{BREQ}}$ is activated earlier, the regain-sequence is initiated by $\overline{\text{HOLD}}$ going high.
Please note that $\overline{\text{HOLD}}$ may also be deactivated without the SAB 80C166 requesting the bus.

2) The next SAB 80C166 driven bus cycle may start here

Package Outline

Plastic Package, P-MQFP-100-2 (SMD)
(Plastic Metric Quad Flat Package)



2) Does not include dambar protrusion of 0.08 max. per side
1) Does not include plastic or metal protrusion of 0.25 max. per side

Figure 17

Sorts of Packing

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

SMD = Surface Mounted Device

SIEMENS

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

SAB 88C166(W)

Preliminary

SAB 88C166(W) 16-Bit Microcontrollers with 32 KByte Flash EPROM

- 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
- Register-Based Design with Multiple Variable Register Banks
- Single-Cycle Context Switching Support
- Up to 256 KBytes Linear Address Space for Code and Data
- 1 KByte On-Chip RAM
- 32 KBytes On-Chip Flash EPROM with Bank Erase Feature
- Read-Protectable Flash Memory
- Dedicated Flash Control Register with Operation Lock Mechanism
- 12 V External Flash Programming Voltage
- Flash Program Verify and Erase Verify Modes
- 100 Flash Program/Erase Cycles guaranteed
- Programmable External Bus Characteristics for Different Address Ranges
- 8-Bit or 16-Bit External Data Bus
- Multiplexed or Demultiplexed External Address/Data Buses
- Hold and Hold-Acknowledge Bus Arbitration Support
- 512 Bytes On-Chip Special Function Register Area
- Idle and Power Down Modes
- 8-Channel Interrupt-Driven Single-Cycle Data Transfer Facilities via Peripheral Event Controller (PEC)
- 16-Priority-Level Interrupt System
- 10-Channel 10-bit A/D Converter with 9.7 µs Conversion Time
- 16-Channel Capture/Compare Unit
- Two Multi-Functional General Purpose Timer Units with 5 Timers
- Two Serial Channels (USARTs)
- Programmable Watchdog Timer
- Up to 76 General Purpose I/O Lines
- Direct clock input without prescaler in the SAB 88C166W (SAB 88C166 with prescaler)
- 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
- 100-Pin Plastic MQFP Package (EIAJ)

Introduction

The SAB 88C166 and the SAB 88C166W are members of the Siemens SAB 80C166 family of full featured single-chip CMOS microcontrollers. They combine high CPU performance (up to 10 million instructions per second) with high peripheral functionality, enhanced IO-capabilities and an on-chip reprogrammable 32 KByte Flash EPROM.

The SAB 88C166W derives its CPU clock signal (operating clock) directly from the on-chip oscillator without using a prescaler, as known from the SAB 80C166W/83C166W. This reduces the device's EME.
The SAB 88C166 operates at half the oscillator clock frequency (using a 2:1 oscillator prescaler), as known from the SAB 80C166/83C166.

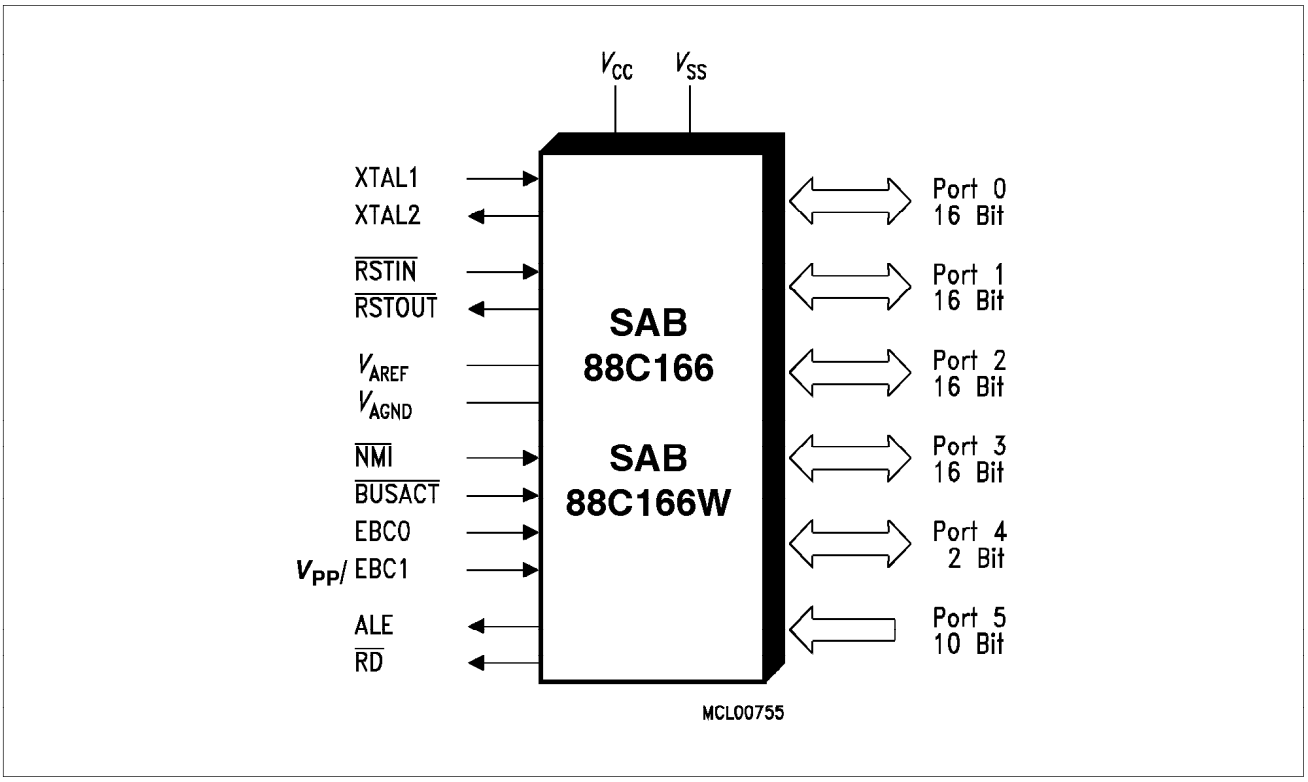


Figure 1
Logic Symbol

Ordering Information

Type	Ordering Code	Package	Function
SAB 88C166-5M	Q67120-C850	P-MQFP-100	16-bit microcontroller, 0 °C to + 70 °C, 1 KByte RAM, 32 KByte Flash EPROM
SAB 88C166W-5M	Q67120-C934	P-MQFP-100	16-bit microcontroller, 0 °C to + 70 °C, 1 KByte RAM, 32 KByte Flash EPROM

Pin Configuration Rectangular P-MQFP-100 (top view)

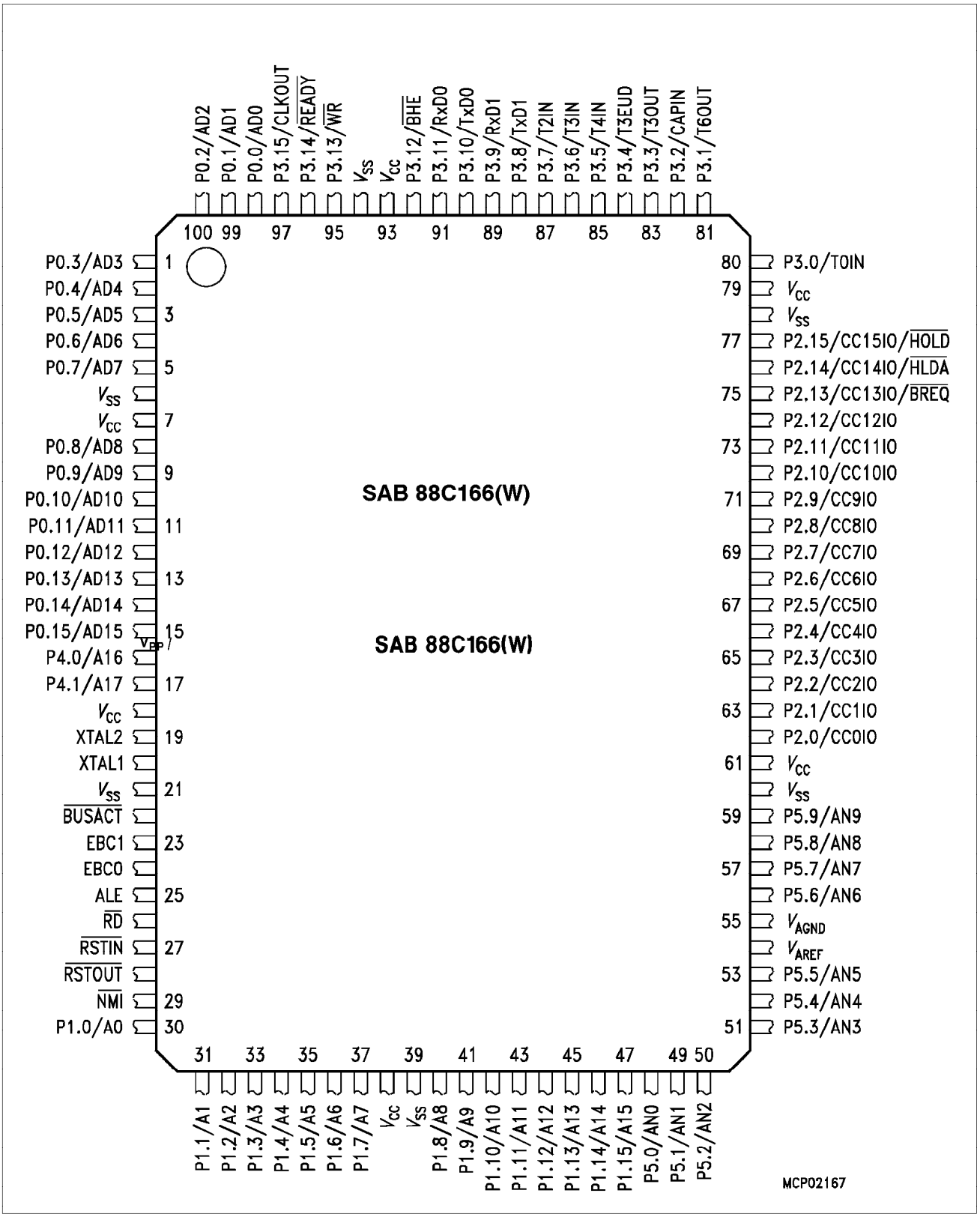


Figure 2

Pin Definitions and Functions

Symbol	Pin Number	Input (I) Output (O)	Function
P4.0 – P4.1	16 - 17	I/O	Port 4 is a 2-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: P4.0 A16 Least Significant Segment Addr. Line P4.1 A17 Most Significant Segment Addr. Line
	16	O	
	17	O	
XTAL1	20	I	XTAL1: Input to the oscillator amplifier and input to the internal clock generator
XTAL2	19	O	XTAL2: Output of the oscillator amplifier circuit. 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.
$\overline{\text{BUSACT}}$, EBC1, EBC0	22 23 24	I I I	External Bus Configuration selection inputs. These pins are sampled during reset and select either the single chip mode or one of the four external bus configurations: $\overline{\text{BUSACT}}$ EBC1 EBC0 Mode/Bus Configuration 0 0 0 8-bit demultiplexed bus 0 0 1 8-bit multiplexed bus 0 1 0 16-bit muliplexed bus 0 1 1 16-bit demultiplexed bus 1 0 0 Single chip mode 1 0 1 Reserved. 1 1 0 Reserved. 1 1 1 Reserved. After reset pin EBC1 accepts the programming voltage for the Flash EPROM as an “alternate function”:
V_{PP}	23		Flash EPROM Programming Voltage V_{PP} = 12 V.
$\overline{\text{RSTIN}}$	27	I	Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the SAB 88C166(W). An internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input (I) Output (O)	Function
$\overline{\text{RSTOUT}}$	28	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.
$\overline{\text{NMI}}$	29	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 SAB 88C166(W) to go into power down mode. If $\overline{\text{NMI}}$ is high, when PWRDN is executed, the part will continue to run in normal mode. If not used, pull $\overline{\text{NMI}}$ high externally.
ALE	25	O	Address Latch Enable Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.
$\overline{\text{RD}}$	26	O	External Memory Read Strobe. $\overline{\text{RD}}$ is activated for every external instruction or data read access.
P1.0 – P1.15	30 - 37 40 - 47	I/O	Port 1 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 1 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..
P5.0 – P5.9	48 – 53 56 – 59	I I	Port 5 is a 10-bit input-only port with Schmitt-Trigger characteristics. The pins of Port 5 also serve as the (up to 10) analog input channels for the A/D converter, where P5.x equals ANx (Analog input channel x).
P2.0 – P2.15	62 – 77	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. The following Port 2 pins also serve for alternate functions: P2.0 CC0IO CAPCOM: CC0 Cap.-In/Comp.Out ... ... P2.13 CC13IO CAPCOM: CC13 Cap.-In/Comp.Out, $\overline{\text{BREQ}}$ External Bus Request Output P2.14 CC14IO CAPCOM: CC14 Cap.-In/Comp.Out, $\overline{\text{HLDA}}$ External Bus Hold Acknowl. Output P2.15 CC15IO CAPCOM: CC15 Cap.-In/Comp.Out, $\overline{\text{HOLD}}$ External Bus Hold Request Input
	62 ... 75 76 77	I/O I/O O I/O O I/O I	

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input (I) Output (O)	Function
P3.0 – P3.15	80 – 92, 95 – 97	I/O	Port 3 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. The following Port 3 pins also serve for alternate functions:
	80	I	P3.0 T0IN CAPCOM Timer T0 Count Input
	81	O	P3.1 T6OUT GPT2 Timer T6 Toggle Latch Output
	82	I	P3.2 CAPIN GPT2 Register CAPREL Capture Input
	83	O	P3.3 T3OUT GPT1 Timer T3 Toggle Latch Output
	84	I	P3.4 T3EUD GPT1 Timer T3 Ext.Up/Down Ctrl.Input
	85	I	P3.5 T4IN GPT1 Timer T4 Input for Count/Gate/Reload/Capture
	86	I	P3.6 T3IN GPT1 Timer T3 Count/Gate Input
	87	I	P3.7 T2IN GPT1 Timer T2 Input for Count/Gate/Reload/Capture
	88	O	P3.8 TxD1 ASC1 Clock/Data Output (Asyn./Syn.)
	89	I/O	P3.9 RxD1 ASC1 Data Input (Asyn.) or I/O (Syn.)
	90	O	P3.10 T×D0 ASC0 Clock/Data Output (Asyn./Syn.)
	91	I/O	P3.11 R×D0 ASC0 Data Input (Asyn.) or I/O (Syn.)
	92	O	P3.12 $\overline{BHE}$ Ext. Memory High Byte Enable Signal,
	95	O	P3.13 $\overline{WR}$ External Memory Write Strobe
	96	I	P3.14 $\overline{READY}$ Ready Signal Input
	97	O	P3.15 CLKOUT System Clock Output (=CPU Clock)
P0.0 – P0.15	98 – 5 8 – 15	I/O	Port 0 is a 16-bit bidirectional IO 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 0 serves as the address (A) and address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes. **Demultiplexed bus modes:** Data Path Width: 8-bit 16-bit; P0.0 – P0.7: D0 – D7 D0 - D7; P0.8 – P0.15: output! D8 - D15 **Multiplexed bus modes:** Data Path Width: 8-bit 16-bit; P0.0 – P0.7: AD0 – AD7 AD0 - AD7; P0.8 – P0.15: A8 - A15 AD8 - AD15
V_{AREF}	54	-	Reference voltage for the A/D converter.
V_{AGND}	55	-	Reference ground for the A/D converter.

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input (I) Output (O)	Function
V_{CC}	7, 18, 38, 61, 79, 93	-	Digital Supply Voltage: + 5 V during normal operation and idle mode. ≥ 2.5 V during power down mode
V_{SS}	6, 21, 39, 60, 78, 94	-	Digital Ground.

Functional Description

This document only describes specific properties of the SAB 88C166(W), e.g. Flash memory functionality or specific DC and AC Characteristics, while for all other descriptions common for the SAB 88C166(W) and the SAB 80C166(W)/83C166(W), e.g. functional description, it refers to the respective Data Sheet for the Non-Flash device.

A detailled description of the SAB 88C166(W)'s instruction set can be found in the **"C16x Family Instruction Set Manual"**.

Memory Organization

The memory space of the SAB 88C166(W) 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 256 KBytes. Address space expansion to 16 MBytes is provided for future versions. The entire memory space can be accessed bytewise or wordwise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

1 KByte of on-chip RAM is 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).

512 bytes of the address space are reserved for the Special Function Register area. SFRs are wordwide registers which are used for controlling and monitoring functions of the different on-chip units. 98 SFRs are currently implemented. Unused SFR addresses are reserved for future members of the SAB 80C166 family.

In order to meet the needs of designs where more memory is required than is provided on chip, up to 256 KBytes of external RAM and/or ROM can be connected to the microcontroller.

Flash Memory Overview

The SAB 88C166(W) provides 32 KBytes of electrically erasable and reprogrammable non-volatile Flash EPROM on-chip for code or constant data, which can be mapped to either segment 0 ($0'0000_H$ to $0'7FFF_H$) or segment 1 ($1'0000_H$ to $1'7FFF_H$) during the initialization phase.

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/EBC1 pin.

The Flash memory is organized in 8 K x 32 bits, 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 (12/12/6/2 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 100 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 SAB 88C166(W)'s on-chip bootstrap loader may be used to load and start the programming code.

To save the customer's know-how, a Flash memory protection option is provided in the SAB 88C166(W). If this was activated once, Flash memory contents cannot be read from any location outside the Flash memory itself.

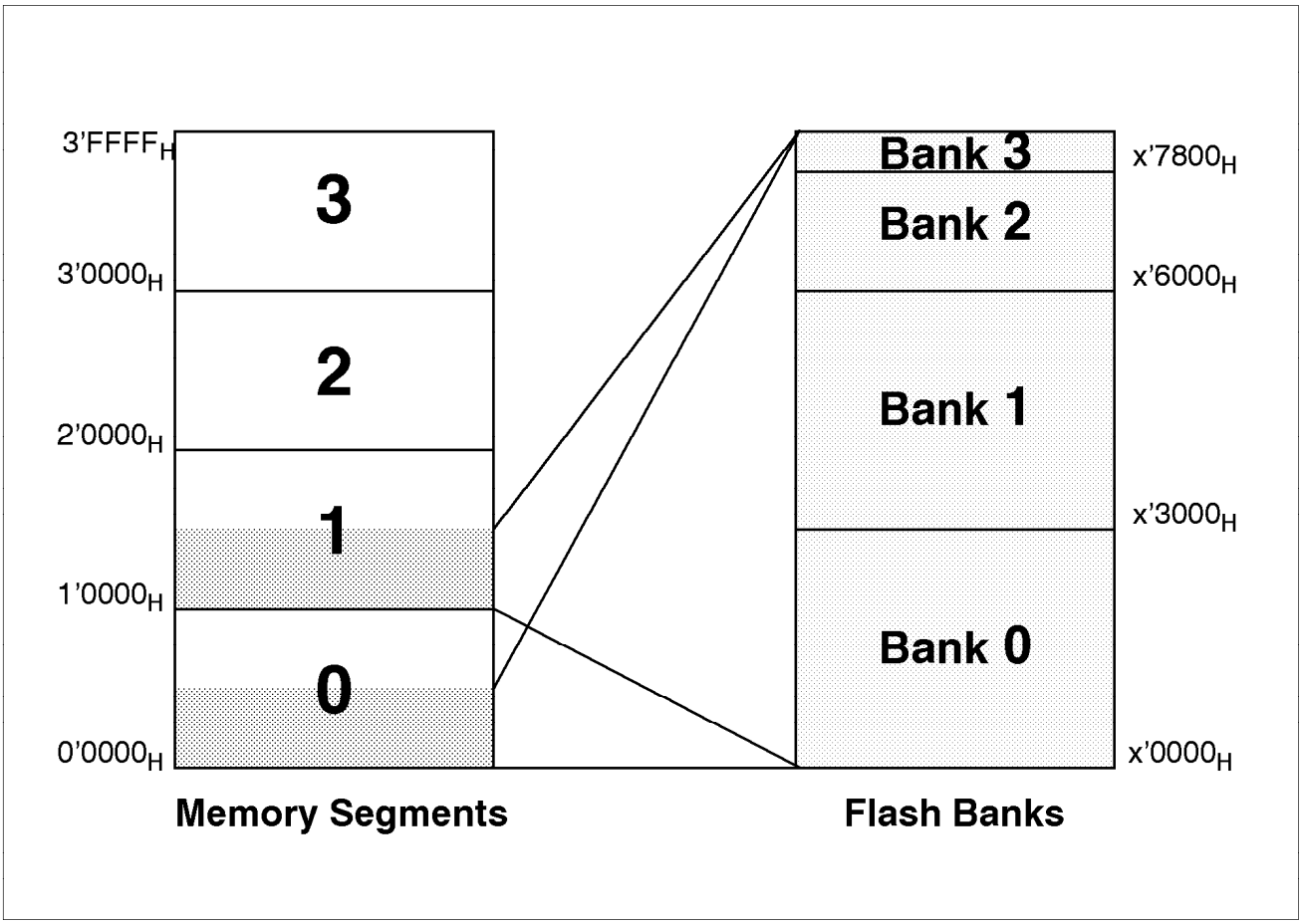


Figure 3
Flash Memory Overview

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 SAB 83C166. 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.

FCR ($FFA0_H$ / $D0_H$) SFR **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) 0 : Flash write operations (program / erase) disabled 1 : Flash write operations (program / erase) enabled
FEE	**Flash Erase Enable Bit** (Significant only, when FWE = '1', see description below) 0 : Flash programming mode selected 1 : Flash erase mode selected
FBUSY	**Flash Busy Bit** (On read accesses) 0 : No Flash write operation in progress 1 : Flash write operation in progress
RPROT	**Flash Read Protection Activation Bit** (On write accesses) 0 : Deactivates Flash read protection 1 : Activates Flash read protection, if this is enabled
FCVPP	**Flash Control V_{PP} Bit** 0 : No V_{PP} failure occurred during a Flash write operation 1 : V_{PP} failure occurred during a Flash write operation
VPPREV	**Flash V_{PP} Revelation Bit** 0 : No valid V_{PP} applied to pin V_{PP} 1 : V_{PP} applied to pin V_{PP} is valid
CKCTL	**Internal Flash Timer Clock Control** Determines the width of an internal Flash write or erase pulse
WDWW	**Word / Double Word Writing Bit** (significant only in programming mode) 0 : 16-bit programming operation 1 : 32-bit programming operation
BE	**Bank Erase Select** (significant only in erasing mode) Selects the Flash Bank to be erased
FWMSET	**Flash Writing Mode Set Bit** (see description below) 0 : Exit Flash writing mode, return to standard mode 1 : Stay in Flash writing mode

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

Note: The FCR is no real register but is rather virtually mapped into the active address space of the Flash memory while the Flash writing mode is active. In writing mode all direct (*mem*) accesses refer to the FCR, while all indirect ([Rw_n]) accesses refer to the Flash memory array itself.

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 controls the writing mode. Bits FWE and FEE select an operation, but do not execute it directly.

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 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'). 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).

Time Specification	Limit Value	
Maximum Programming Pulse Width	128	µs
Maximum Cumulated Programming Time	2.5	ms
Maximum Erase Pulse Width	10	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	$2^7 * 1/f_{CPU}$	6.4 µs	1 MHz	---
0 1	$2^{11} * 1/f_{CPU}$	102.4 µs	16 MHz	1 MHz
1 0	$2^{15} * 1/f_{CPU}$	1.64 ms	---	3.28 MHz
1 1	$2^{18} * 1/f_{CPU}$	13.11 ms	---	13.11 MHz

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 selected bank depend on the Flash memory mapping chosen.

BE setting	Bank	Addresses Selected for Erasure (x = 0 or 1)
0 0	0	$x'0000_H$ to $x'2FFF_H$
0 1	1	$x'3000_H$ to $x'5FFF_H$
1 0	2	$x'6000_H$ to $x'77FF_H$
1 1	3	$x'7800_H$ to $x'7FFF_H$

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.

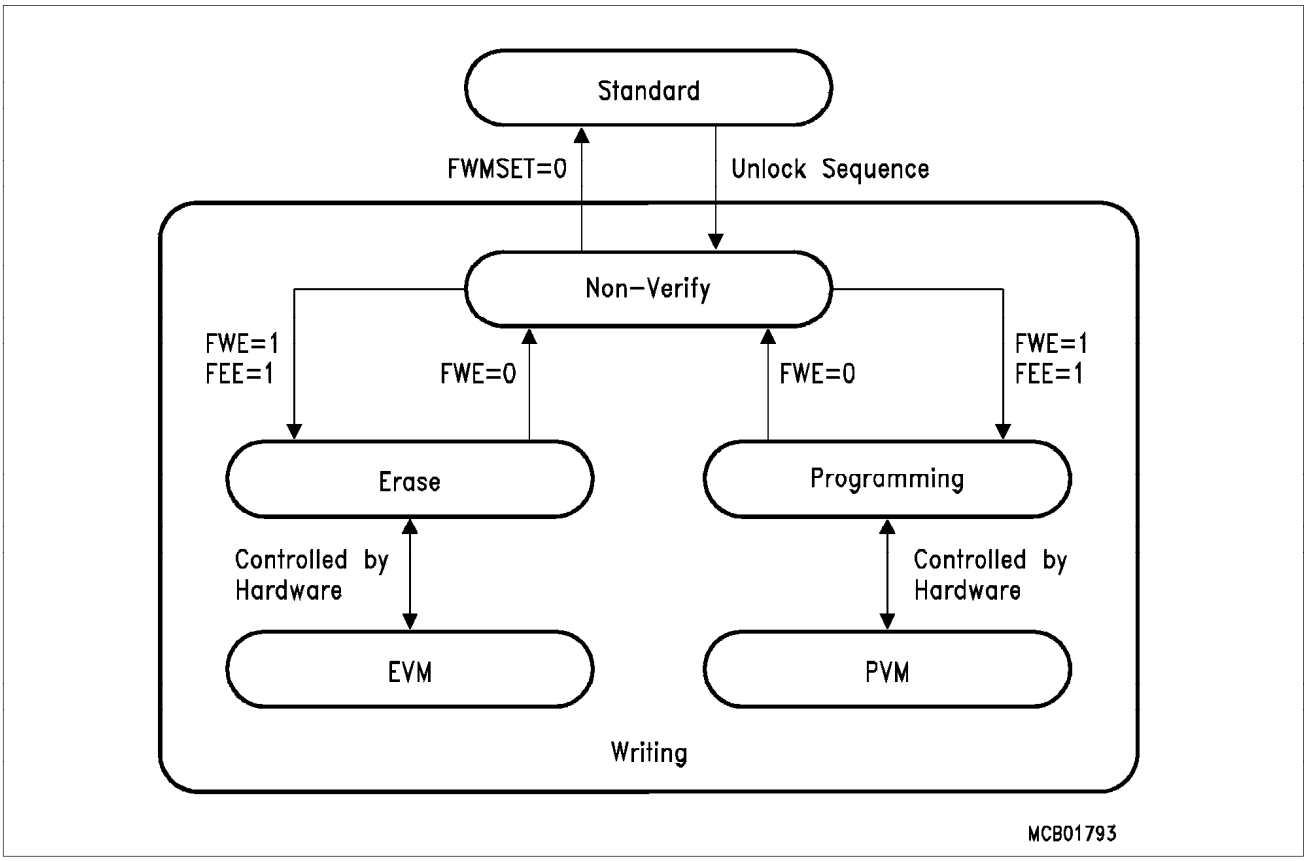


Figure 4
Flash Operating Mode Transitions

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 accesses to the FCR or Flash write operations.

Note: When Flash protection is active, data operands can be accessed only by instructions that are executed out of the internal Flash memory.

The Flash Writing Modes must be entered for programming or erasing the Flash memory. The SAB 88C166 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}/EBC1 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 in this case.

The UNLOCK sequence is a specific key code sequence, which is required to enable the writing modes of the SAB 88C166(W). The UNLOCK sequence must use identical values (see example below) and 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 SAB 88C166(W) 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 20 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 SAB 88C166(W) 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 200 µs.

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

MOV [Rw_n], Rw_m ; Program one word

If 32-bit programming was selected, the operation will start automatically when the second of two subsequent 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 [Rw_n], Rw_x ; Prepare programming of first word
MOV [Rw_n], Rw_y ; Start programming of both words

Upon the execution of the second instruction (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 Rw_m, [Rw_n] ; First (invalid) read of dedicated cell
... ; 4 µs delay to stabilize internal margin voltage
MOV Rw_m, [Rw_n] ; 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 programming mode 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.

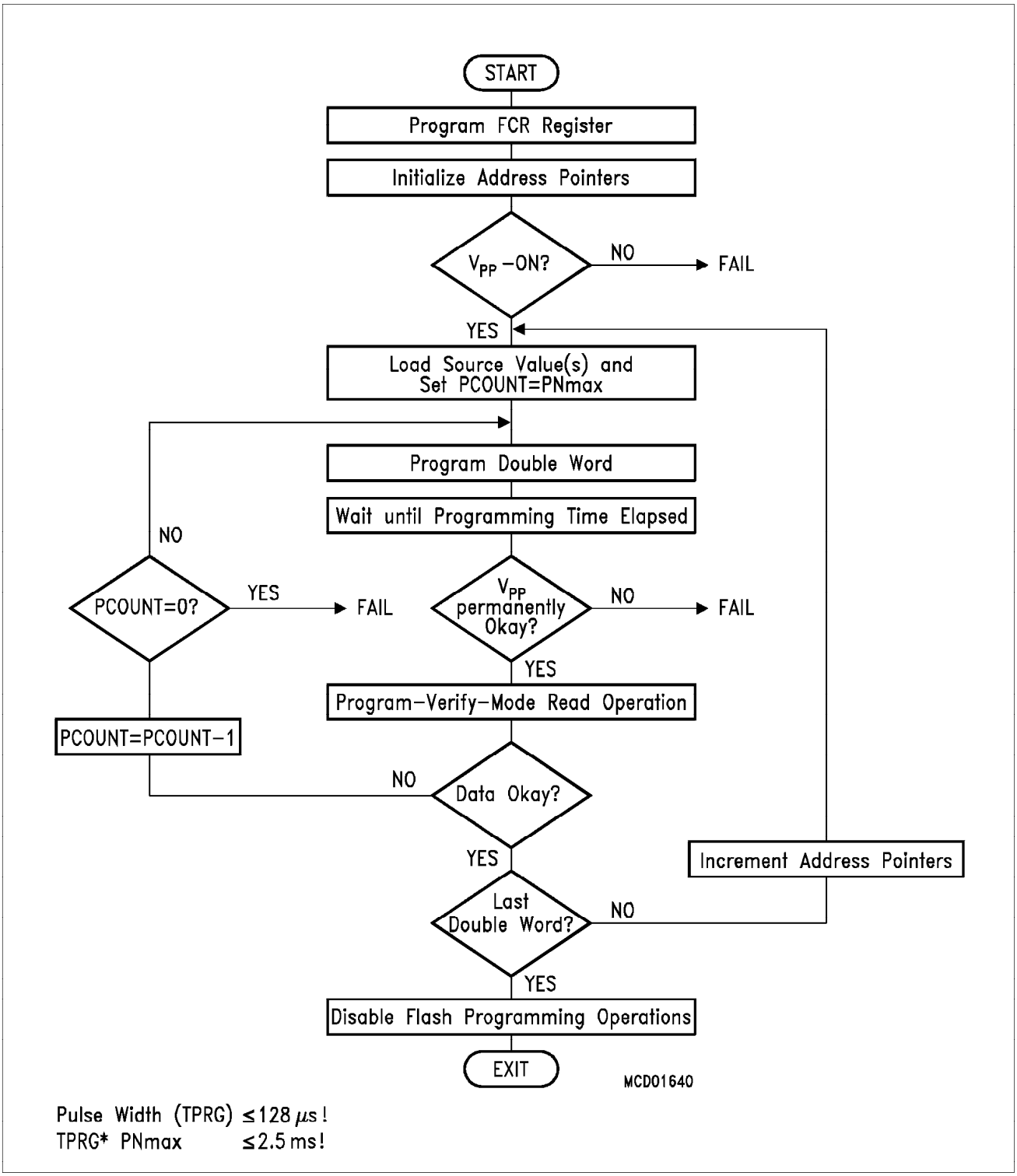


Figure 5
Flash Programming Algorithm

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 or SRC_END respectively. The target Flash memory block begins at location FLASH_START. This example uses 32-bit Flash programming.

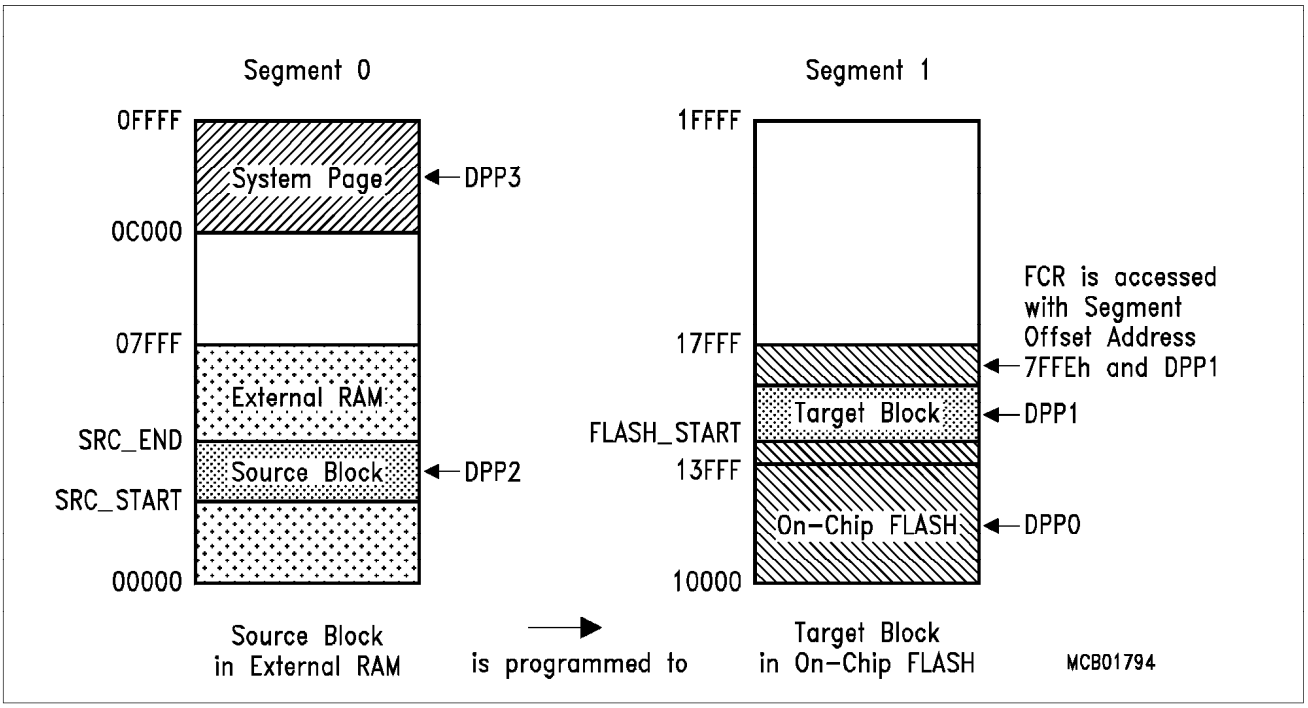


Figure 6
Memory Allocation for Flash Programming Example

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}/EBC1 receives a proper V_{PP} supply voltage.
- The SAB 88C166(W) runs at 20 MHz CPU clock (absolute time delays refer to it).
- 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':   100 µs programming pulse (fCPU = 20 MHz)
       ; #xxxx xxxx 1xxx xxxx: WDWW='1':     Select 32-bit programming mode
       ; #1xxx xxxx xxxx xxxx: FWMSET='1':   Stay in writing mode
MOV    DPP1:pof FCR, R15     ; Write Value to the 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}/EBC1** 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 100 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 programming
```

- **Wait until programming time elapsed** (100 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               ; Auxiliary 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 was 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.

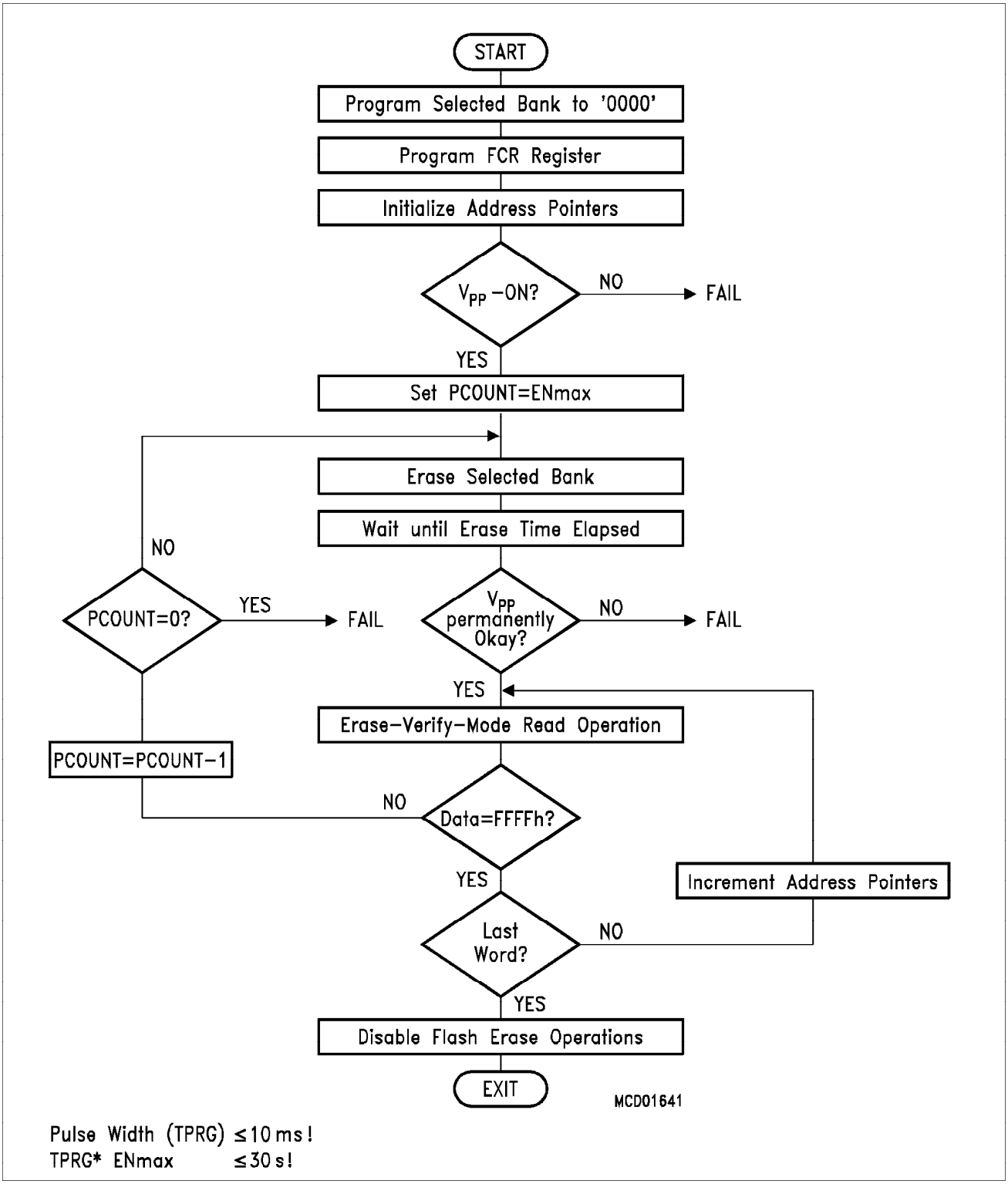


Figure 7
Flash Erase Algorithm

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.

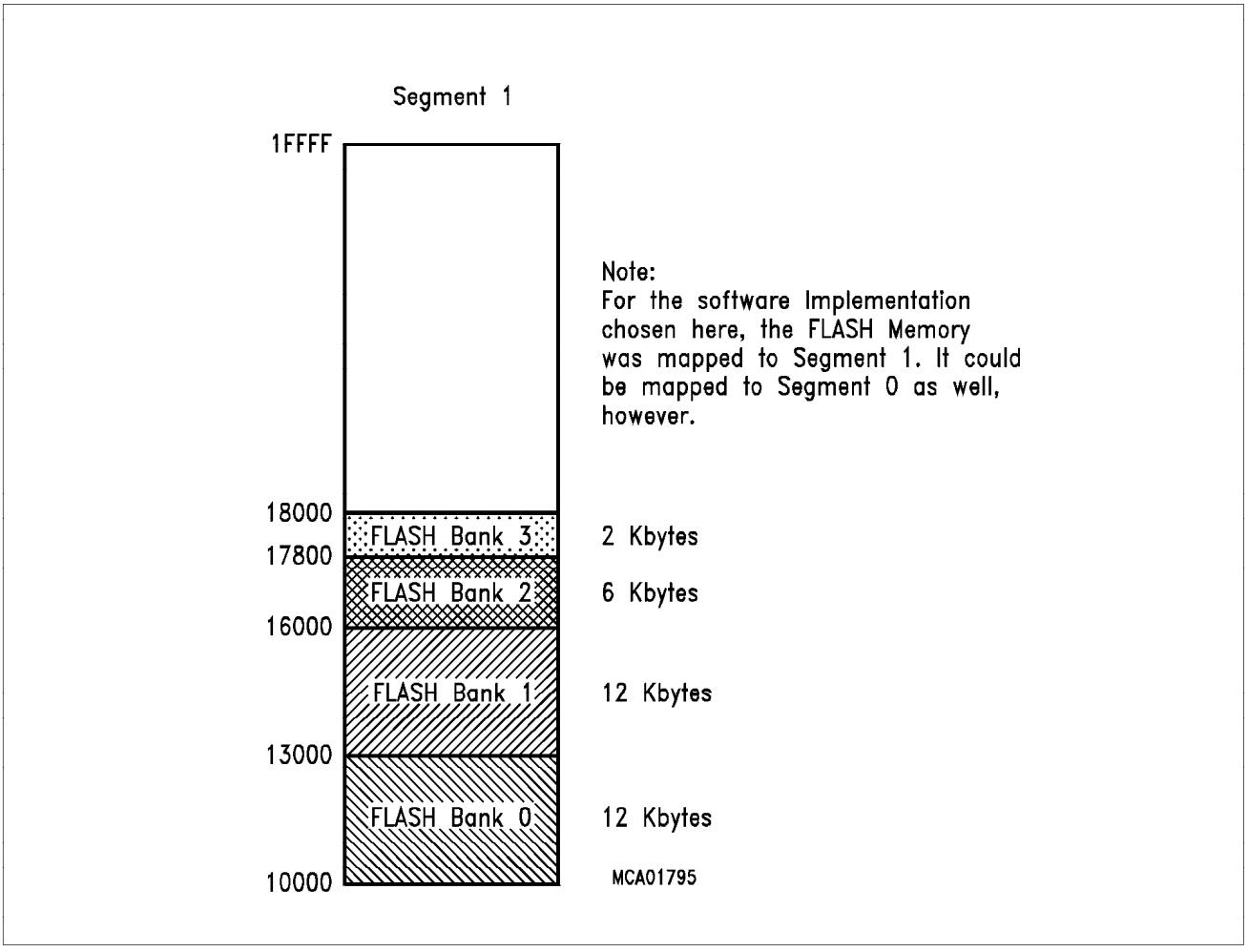


Figure 8
Memory Banking for Flash Erasure

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}/EBC1 receives a proper V_{PP} supply voltage.
- The SAB 88C166(W) runs at 20 MHz CPU clock (absolute time delays refer to it).
- 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 0110 0011B
       ; #xxxx xxxx xxxx xxx1: FWE='1':      Enable Flash write operations
       ; #xxxx xxxx xxxx xx1x: FEE='1':      Select erase mode
       ; #xxxx xxxx x11x xxxx: CKCTL='11':   10 ms erase pulse (fCPU = 20 MHz)
       ; #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 the 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}/EBC1** 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 auxiliary 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 SAB 88C166(W) 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 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 into 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'.

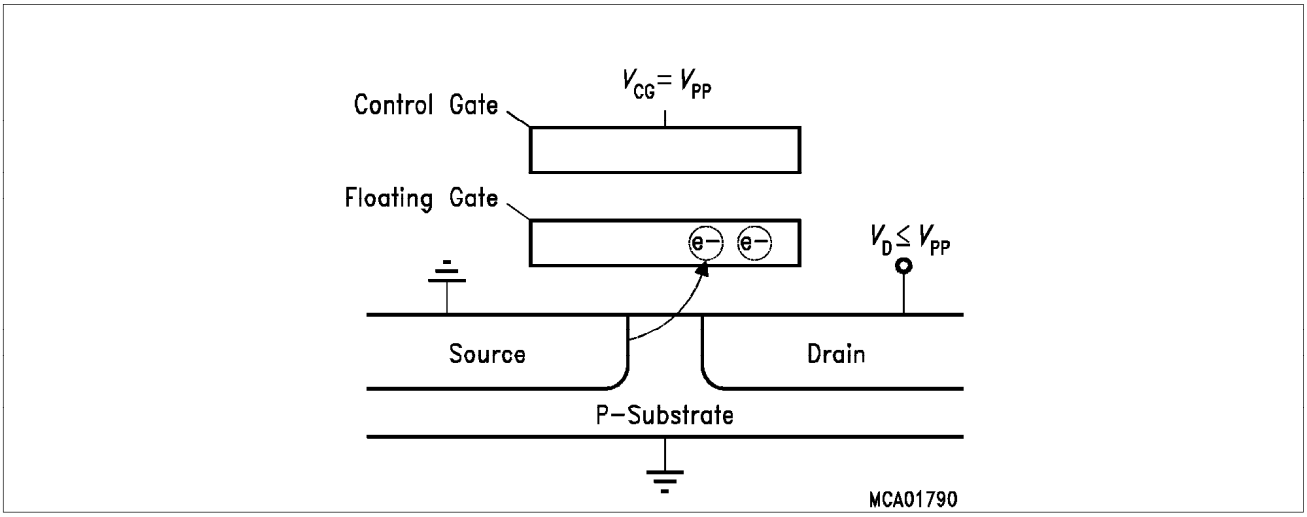


Figure 9
Flash Memory Cell Programming Mechanism

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 SAB 88C166(W) 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.

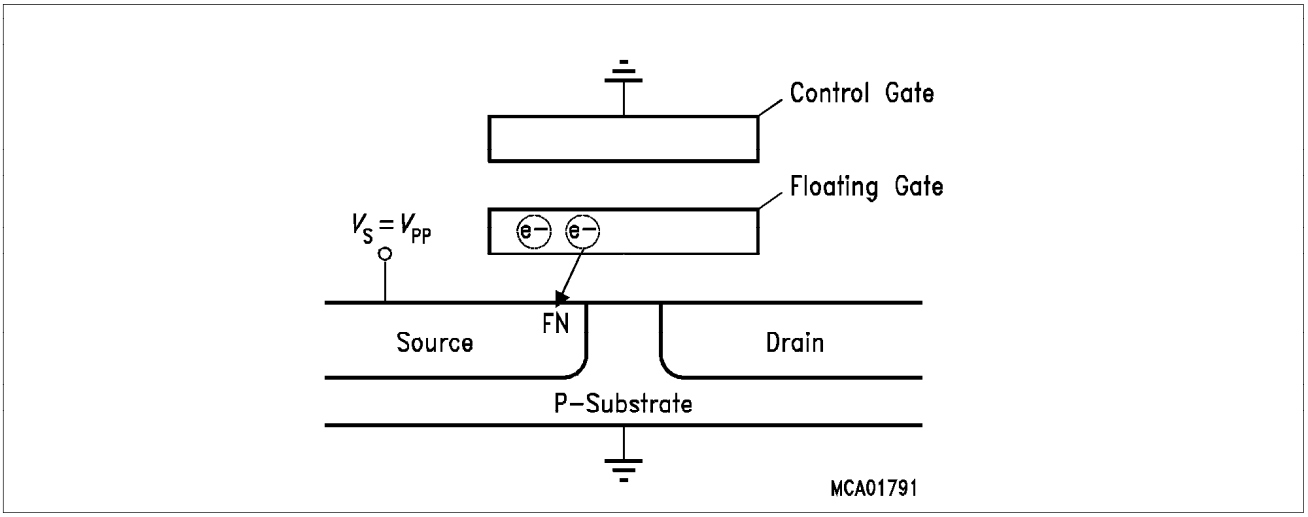


Figure 10
Flash Memory Cell Erase Mechanism

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

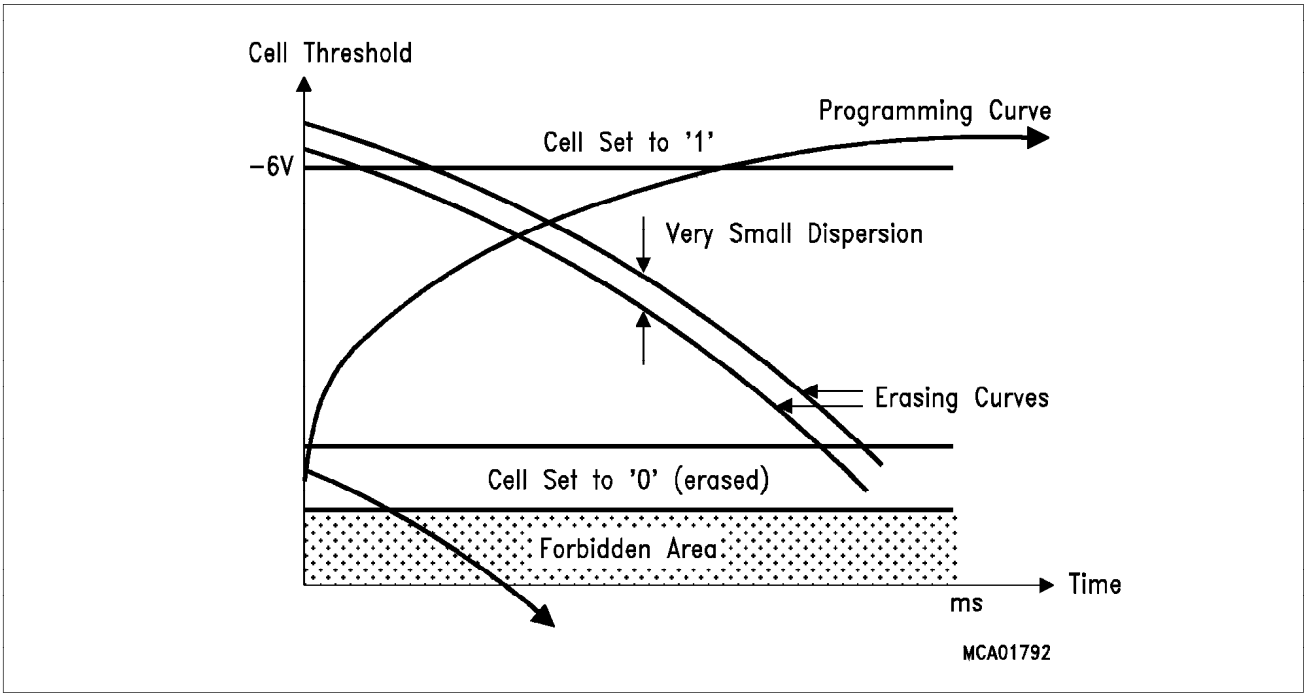


Figure 11
Flash Erasure

Note that 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.

Absolute Maximum Ratings

Ambient temperature under bias (T_A):
SAB 88C166(W)-5M ..... 0 to + 70 °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 W
Flash programming voltage (V_{PP})..... – 0.3 to + 13.5 V

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 SAB 88C166(W) 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 SAB 88C166(W) 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 SAB 88C166(W).

DC Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V; f_{CPU} = 20 MHz
T_A = 0 to + 70 °C for SAB 88C166(W)-5M

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage EBC1/V_{PP}	V_{IL1}SR	– 0.3	0.2 V_{CC} – 0.1	V	–
Input low voltage (all except EBC1/V_{PP})	V_{IL2}SR	– 0.5	0.2 V_{CC} – 0.1	V	–
Input high voltage (all except $\overline{\text{RSTIN}}$ and XTAL1)	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	–
Output low voltage (Port 0, Port 1, 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 (Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$)	V_{OH}CC	0.9 V_{CC} 2.4	–	V	I_{OH} = – 500 µA I_{OH} = – 2.4 mA
Output high voltage (all other outputs)	V_{OH1}CC	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 250 µA I_{OH} = – 1.6 mA
Input leakage current (Port 5) [1]	I_{OZ1}CC	–	± 200	nA	0 V < V_{IN} < V_{CC}
Input leakage current (all other)	I_{OZ2}CC	–	± 500	nA	0 V < V_{IN} < V_{CC}
V_{PP} leakage current EBC1/V_{PP}	I_{PPS}CC	–	± 100	µA	$V_{PP} \leq V_{CC}$
$\overline{\text{RSTIN}}$ pullup resistor	R_{RST}CC	50	150	kΩ	–
Read inactive current [4]	I_{RH} [2]	–	– 40	µA	V_{OUT} = V_{OHmin}
Read active current [4]	I_{RL} [3]	-500	–	µA	V_{OUT} = V_{OLmax}
ALE inactive current [4]	I_{ALEL} [2]	–	150	µA	V_{OUT} = V_{OLmax}
ALE active current [4]	I_{ALEH} [3]	2100	–	µA	V_{OUT} = V_{OHmin}
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 T_A = 25 °C
Power supply current	I_{CC}	–	50 + 5 x f_{CPU}	mA	$\overline{\text{RSTIN}}$ = V_{IL2} f_{CPU} in [MHz] [6]
Idle mode supply current	I_{ID}	–	30 + 1.5 x f_{CPU}	mA	$\overline{\text{RSTIN}}$ = V_{IH1} f_{CPU} in [MHz] [6]

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Power-down mode supply current	I_{PD}	–	50	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	1/TCL = 40 MHz 32-bit programming V_{PP} = 12 V
V_{PP} during write/read	V_{PP}	11.4	12.6	V	

Notes

1) This specification does not apply to the analog input (Port 5.x) which is currently converted.

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-mode.

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

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) 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.
A voltage of $V_{CC} \geq$ 2.5 V is sufficient to retain the content of the internal RAM during power down mode.

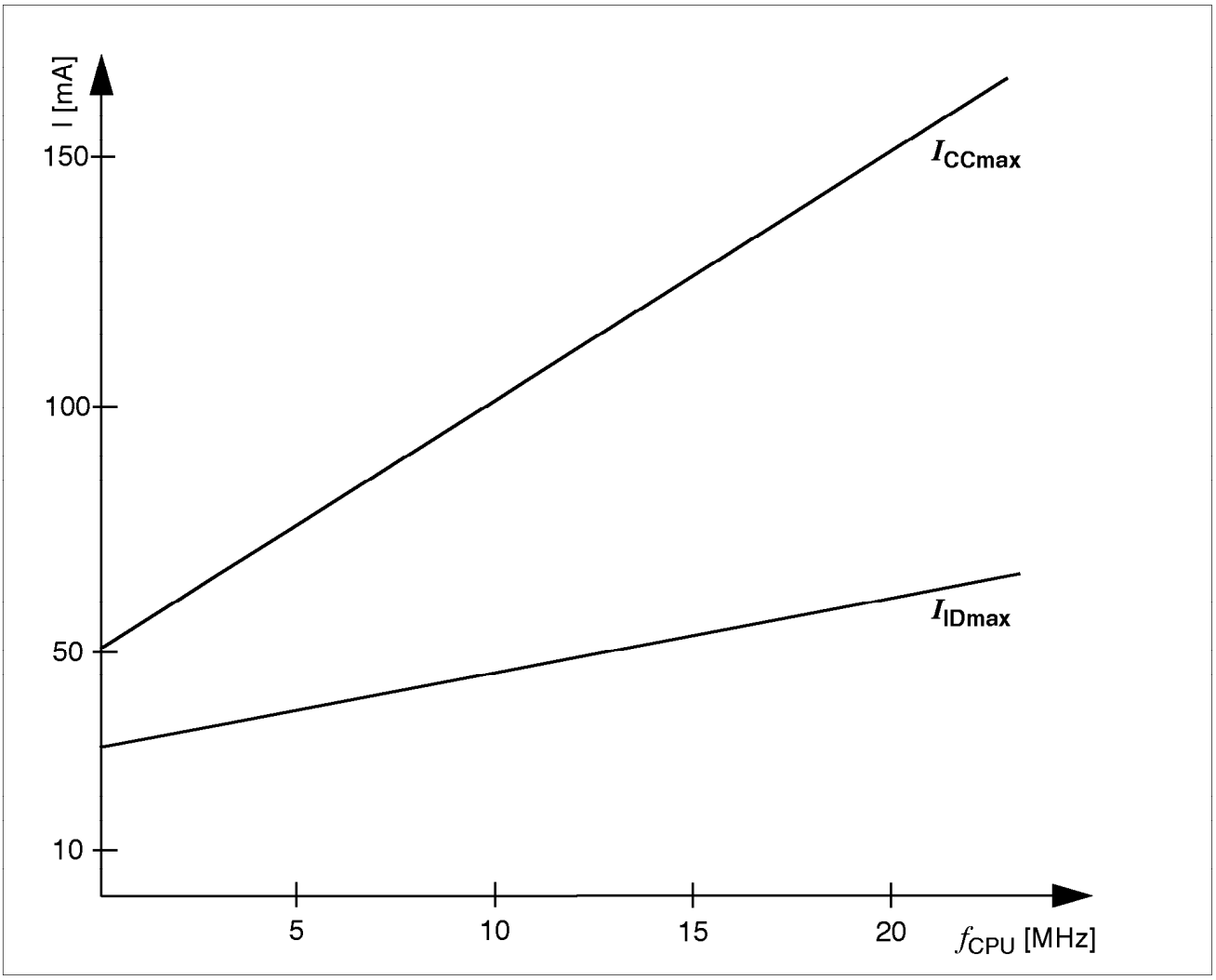


Figure 12
Supply/Idle Current as a Function of Operating Frequency

A/D Converter Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166(W)-5M
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_SCC	–	2 t_{SC}		2) 4)
Conversion time	t_CCC	–	10 t_{CC} + t_S + 4TCL		3) 4)
Total unadjusted error	TUECC	–	± 2	LSB	5)
Internal resistance of reference voltage source	R_{AREF}CC	–	t_{CC} / 250 - 0.25	kΩ	t_{CC} in [ns] 6) 7)
Internal resistance of analog source	R_{ASRC}CC	–	t_S / 500 – 0.25	kΩ	t_S in [ns] 2) 7)
ADC input capacitance	C_{AIN}CC	–	50	pF	7)

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 capacitors to reach their 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. The value for the sample clock is t_{SC} = TCL x 32.

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.
The value for the conversion clock is t_{CC} = TCL x 32.

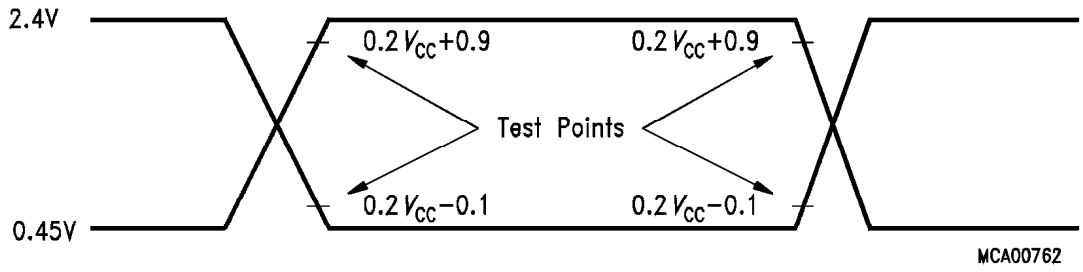
4) This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.

5) TUE is tested at V_{AREF} = 5.0 V, V_{AGND} = 0 V, V_{CC} = 4.8 V. It is guaranteed by design characterization for all other voltages within the defined voltage range.

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 capacitors to reach their respective voltage level within t_{CC}. The maximum internal resistance results from the CPU clock period.

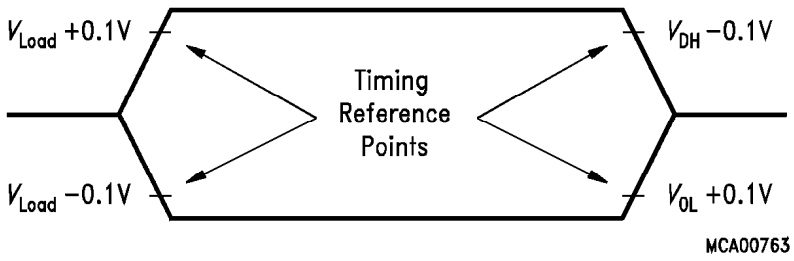
7) Not 100% tested, guaranteed by design characterization.

Testing Waveforms



AC inputs during testing are driven at 2.4 V for a logic '1' and 0.4 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

Memory Cycle Variables

The timing tables below use three variables which are derived from registers SYSCON and BUSCON1 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 x <ALECTL>
Memory Cycle Time Waitstates	t_C	2TCL x (15 – <MCTC>)
Memory Tristate Time	t_F	2TCL x (1 – <MTTC>)

AC Characteristics

The specification of the timings depends on the CPU clock signal that is used in the respective device. In this regard the specification for the SAB 88C166 and the SAB 88C166W are different. While the SAB 88C166W directly uses the clock signal fed to XTAL1 and therefore has to take into account the duty cycle variation of this signal, the SAB 88C166 derives its CPU clock from the XTAL1 signal via a 2:1 prescaler and therefore is independant from these variations.

For these reasons the following pages provide the timing specifications for SAB 88C166 and for SAB 88C166W separately (where applicable).

AC Characteristics

External Clock Drive XTAL1 for the SAB 88C166

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to +70 °C for SAB 88C166-5M

Parameter	Symbol	Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
Oscillator period	TCLSR	25	25	25	500	ns
High time	t_1SR	6	–	6	–	ns
Low time	t_2SR	6	–	6	–	ns
Rise time	t_3SR	–	5	–	5	ns
Fall time	t_4SR	–	5	–	5	ns

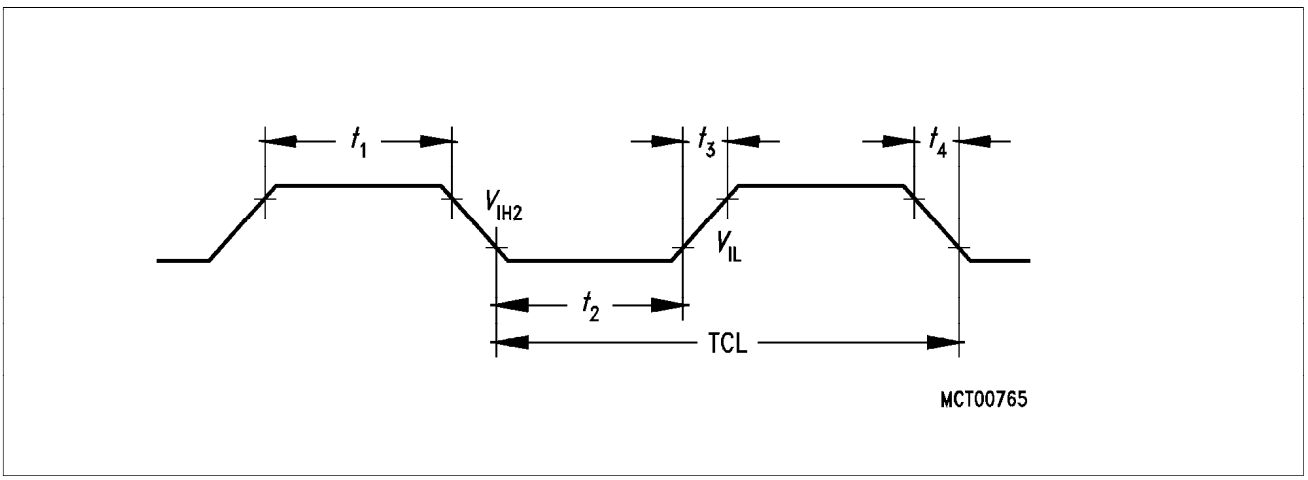


Figure 15
External Clock Drive XTAL1

AC Characteristics (cont'd)

External Clock Drive XTAL1 for the SAB 88C166W

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V

T_A = 0 to + 70 °C for SAB 88C166W-M

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock 1/CLP = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
Oscillator period	CLPSR	62.5	62.5	50	1000	ns
High time	TCL_HSR	25	–	25	$CLP-TCL_L$	ns
Low time	TCL_LSR	25	–	25	$CLP-TCL_H$	ns
Rise time	t_RSR	–	10	–	10	ns
Fall time	t_FSR	–	10	–	10	ns
Oscillator duty cycle	DCSR	0.4	0.6	25 / CLP	1 – 25 / CLP	
Clock cycle	TCLSR	25	37.5	CLP x DC_{min}	CLP x DC_{max}	ns

Note: In order to run the SAB 88C166W at a CPU clock of 20 MHz the duty cycle of the oscillator clock must be 0.5, ie. the relation between the oscillator high and low phases must be 1:1. So the variation of the duty cycle of the oscillator clock limits the maximum operating speed of the device.
The 16 MHz values in the tables are given as an example for a typical duty cycle variation of the oscillator clock from 0.4 to 0.6.

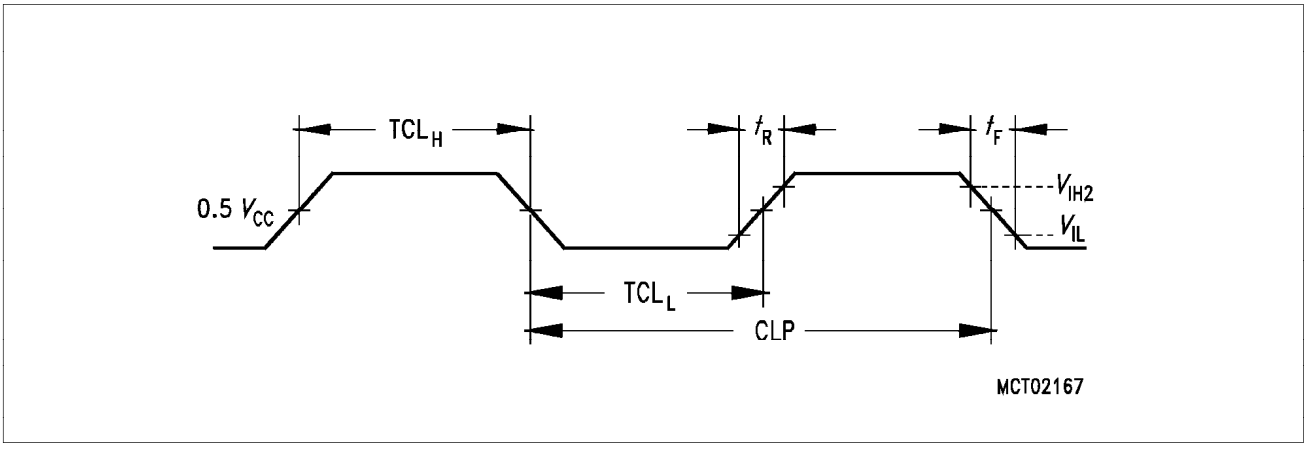


Figure 16
External Clock Drive XTAL1

AC Characteristics (cont'd)

Multiplexed Bus for the SAB 88C166

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ℃ for SAB 88C166-5M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 6 TCL + 2t_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_5CC	15 + t_A	–	TCL – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL – 15 + t_A	–	ns
Address hold after ALE	t_7CC	15 + t_A	–	TCL – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 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
$\overline{RD}$ 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	–	75 + 2t_A + t_C	–	4TCL – 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}SR	0	–	0	–	ns
Data float after $\overline{RD}$	t_{19}SR	–	35 + t_F	–	2TCL – 15 + t_F	ns
Data valid to $\overline{WR}$	t_{22}CC	35 + t_C	–	2TCL – 15 + t_C	–	ns
Data hold after $\overline{WR}$	t_{23}CC	35 + t_F	–	2TCL – 15 + t_F	–	ns

Parameter	Symbol	Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
ALE rising edge after $\overline{RD}$, $\overline{WR}$	t_{25}CC	35 + t_F	–	2TCL – 15 + t_F	–	ns
Address hold after $\overline{RD}$, $\overline{WR}$	t_{27}CC	35 + t_F	–	2TCL – 15 + t_F	–	ns

AC Characteristics (cont'd)

Multiplexed Bus for the SAB 88C166W

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ℃ for SAB 88C166W-M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 6 TCL + 2t_A + t_C + t_F (150 ns at 20-MHz CPU clock without waitstates)

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock 1/CLP = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
ALE high time	t_5CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL_{min} – 15 + t_A	–	ns
Address hold after ALE	t_7CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 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	–	42.5	–	TCL_{max} + 5	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}CC	52.5 + t_C	–	CLP – 10 + t_C	–	ns
$\overline{RD}$ $\overline{WR}$ low time (no RW-delay)	t_{13}CC	77.5 + t_C	–	CLP+TCL_{min} – 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}SR	–	47.5 + t_C	–	CLP – 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}SR	–	72.5 + t_C	–	CLP+TCL_{min} – 20 + t_C	ns
ALE low to valid data in	t_{16}SR	–	72.5 + t_A + t_C	–	CLP+TCL_{min} – 20 + t_C	ns
Address to valid data in	t_{17}SR	–	100 + 2t_A + t_C	–	2CLP – 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}SR	0	–	0	–	ns
Data float after $\overline{RD}$	t_{19}SR	–	47.5 + t_F	–	CLP – 15 + t_F	ns

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock 1/CLP = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
Data valid to $\overline{WR}$	t_{22}CC	47.5 + t_C	–	CLP – 15 + t_C	–	ns
Data hold after $\overline{WR}$	t_{23}CC	47.5 + t_F	–	CLP – 15 + t_F	–	ns
ALE rising edge after $\overline{RD}$, $\overline{WR}$	t_{25}CC	47.5 + t_F	–	CLP – 15 + t_F	–	ns
Address hold after $\overline{RD}$, $\overline{WR}$	t_{27}CC	47.5 + t_F	–	CLP – 15 + t_F	–	ns

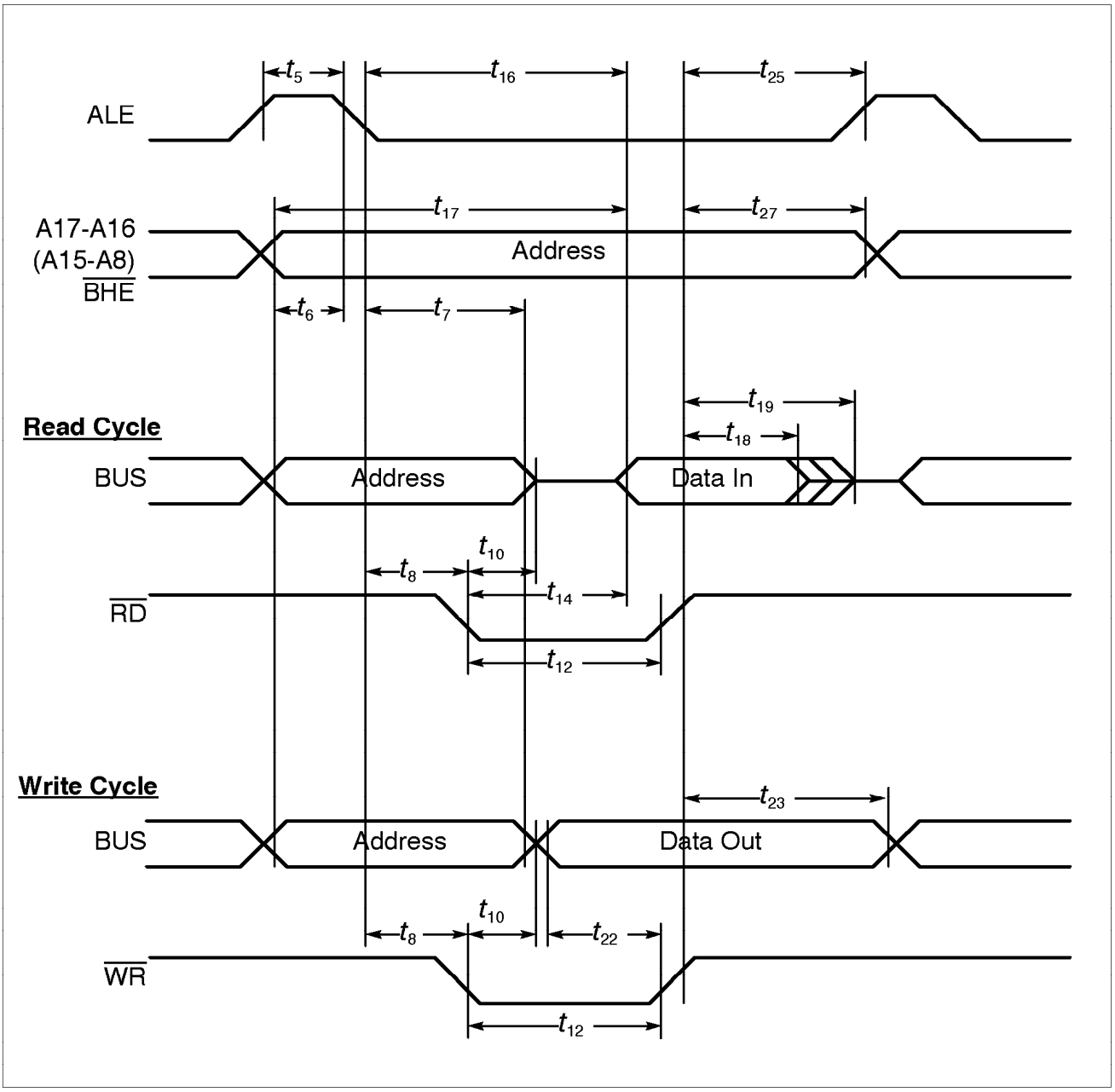


Figure 17
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE

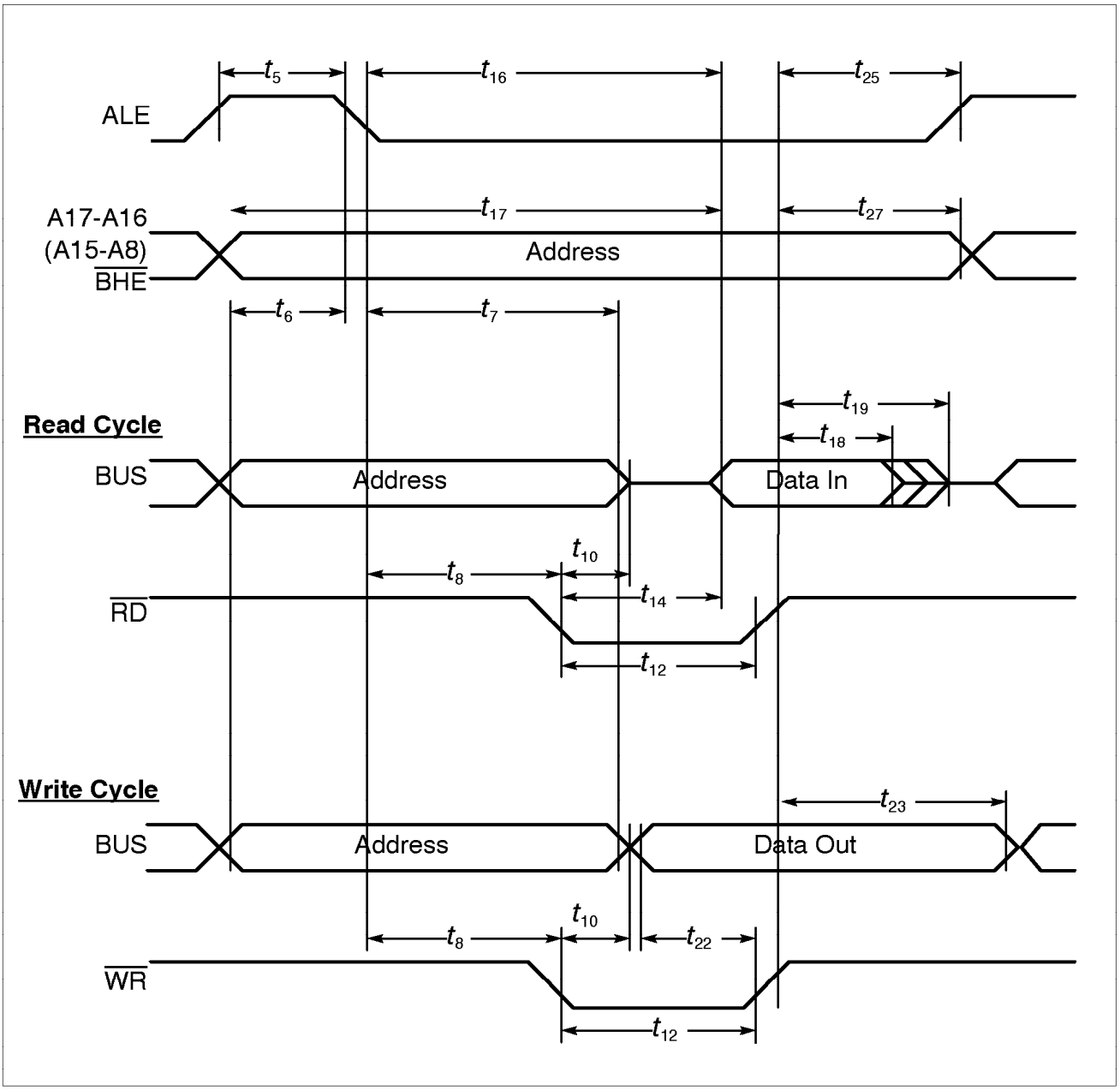


Figure 18
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Extended ALE

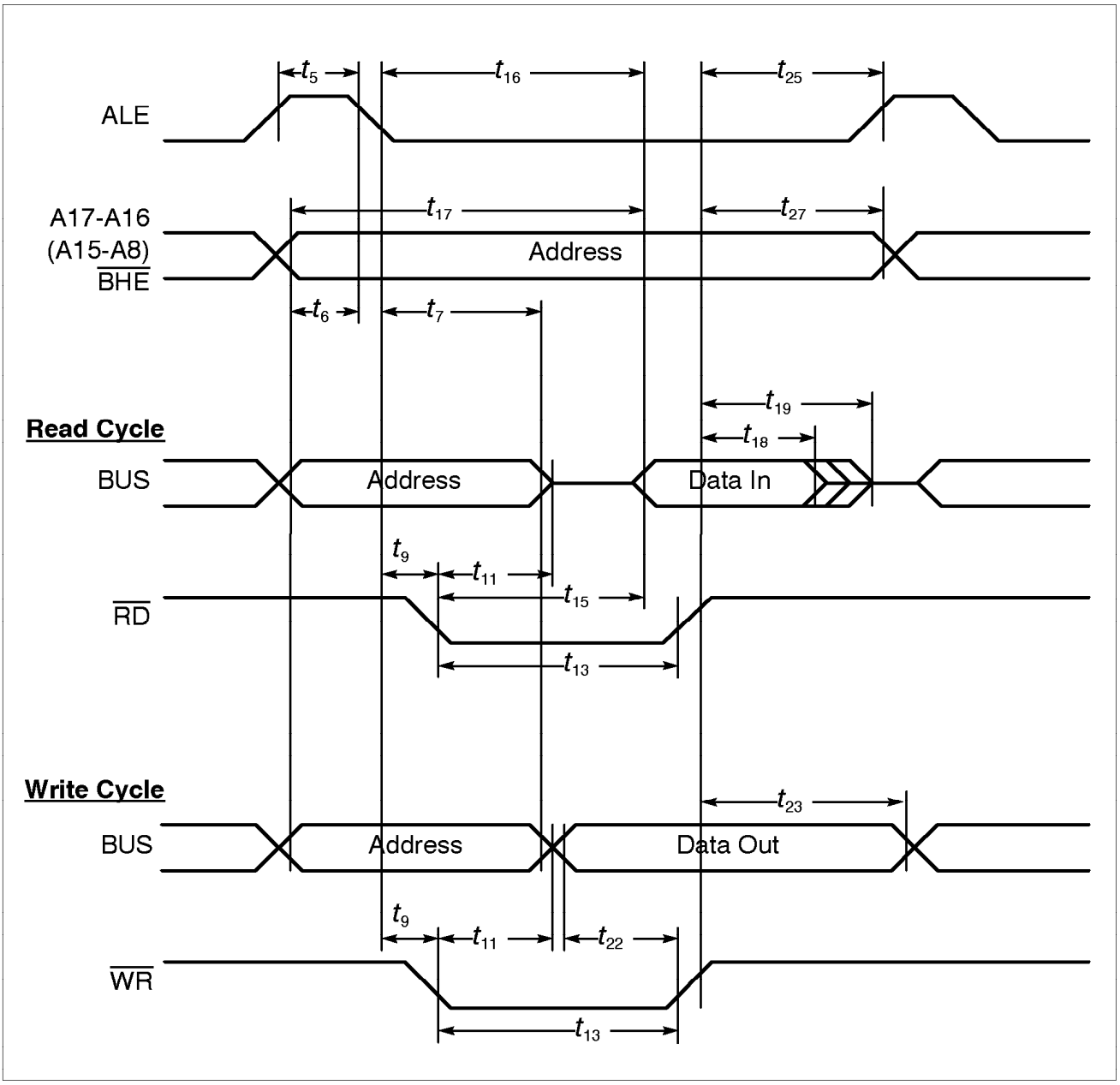


Figure 19
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE

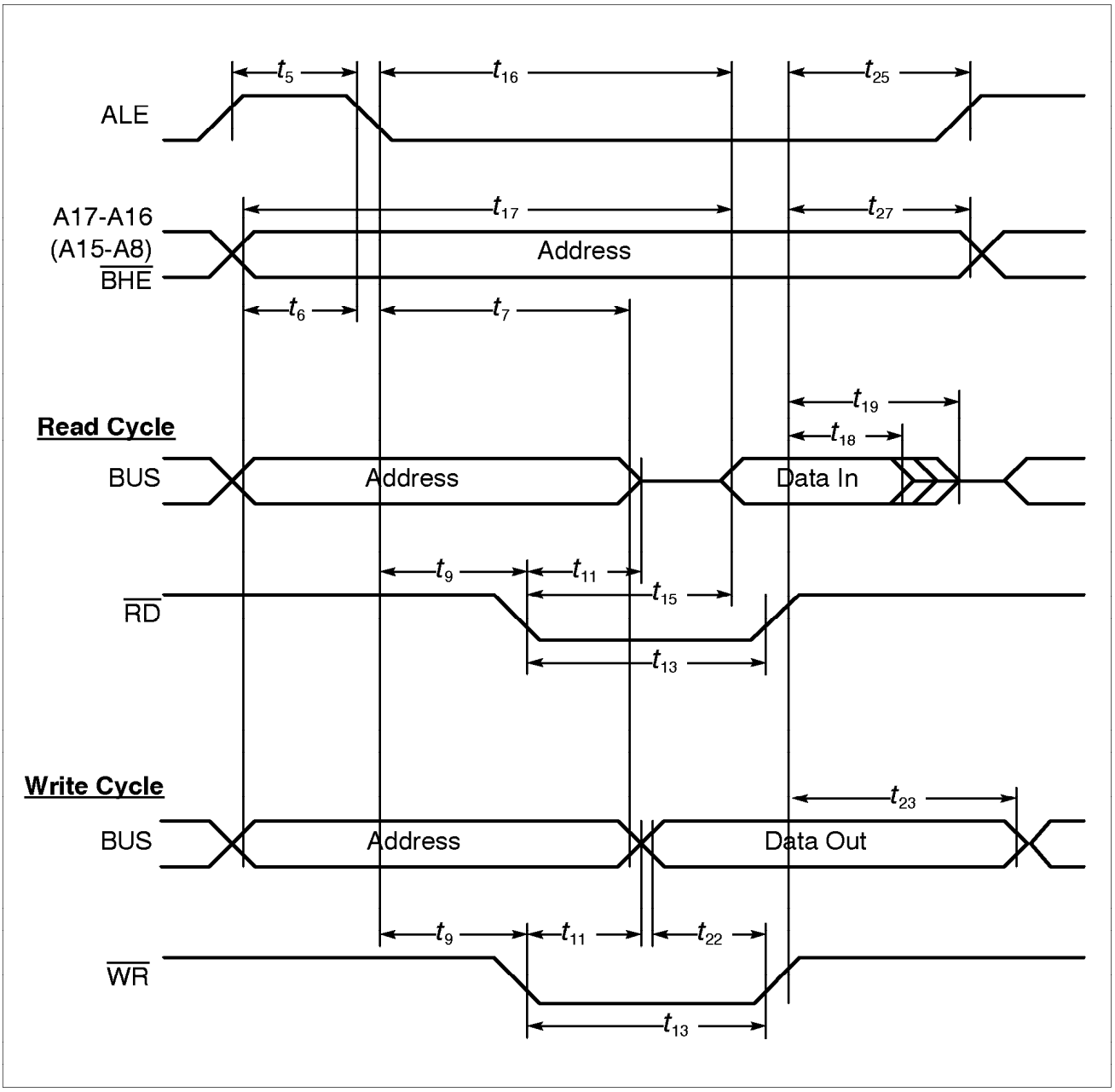


Figure 20
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

Demultiplexed Bus for the SAB 88C166

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166-5M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 4 TCL + 2t_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_5CC	15 + t_A	–	TCL – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL – 15 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 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	–	75 + 2t_A + t_C	–	4TCL – 25 + 2t_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)	t_{20}SR	–	35 + t_F	–	2TCL – 15 + t_F	ns
Data float after $\overline{RD}$ rising edge (no RW-delay)	t_{21}SR	–	15 + t_F	–	TCL – 10 + t_F	ns
Data valid to $\overline{WR}$	t_{22}CC	35 + t_C	–	2TCL – 15 + 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
Address hold after $\overline{RD}$, $\overline{WR}$	t_{28}CC	0 + t_F	–	0 + t_F	–	ns

AC Characteristics (cont'd)

Demultiplexed Bus for the SAB 88C166W

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ℃ for SAB 88C166W-M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 4 TCL + 2t_A + t_C + t_F (100 ns at 20-MHz CPU clock without waitstates)

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock 1/CLP = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
ALE high time	t_5CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL_{min} – 15 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 10 + t_A	–	– 10 + t_A	–	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}CC	52.5 + t_C	–	CLP – 10 + t_C	–	ns
$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	t_{13}CC	77.5 + t_C	–	CLP+TCL_{min} – 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}SR	–	47.5 + t_C	–	CLP – 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}SR	–	72.5 + t_C	–	CLP+TCL_{min} – 20 + t_C	ns
ALE low to valid data in	t_{16}SR	–	72.5 + t_A + t_C	–	CLP+TCL_{min} – 20 + t_A + t_C	ns
Address to valid data in	t_{17}SR	–	100 + 2t_A + t_C	–	2CLP – 25 + 2t_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)	t_{20}SR	–	47.5 + t_F	–	CLP – 15 + t_F	ns
Data float after $\overline{RD}$ rising edge (no RW-delay)	t_{21}SR	–	15 + t_F	–	TCL_{min} – 10 + t_F	ns
Data valid to $\overline{WR}$	t_{22}CC	47.5 + t_C	–	CLP – 15 + t_C	–	ns
Data hold after $\overline{WR}$	t_{24}CC	15 + t_F	–	TCL_{min} – 10 + t_F	–	ns

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock 1/CLP = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
ALE rising edge after $\overline{RD}$, $\overline{WR}$	t_{26}CC	– 10 + t_F	–	– 10 + t_F	–	ns
Address hold after $\overline{RD}$, $\overline{WR}$	t_{28}CC	0 + t_F	–	0 + t_F	–	ns

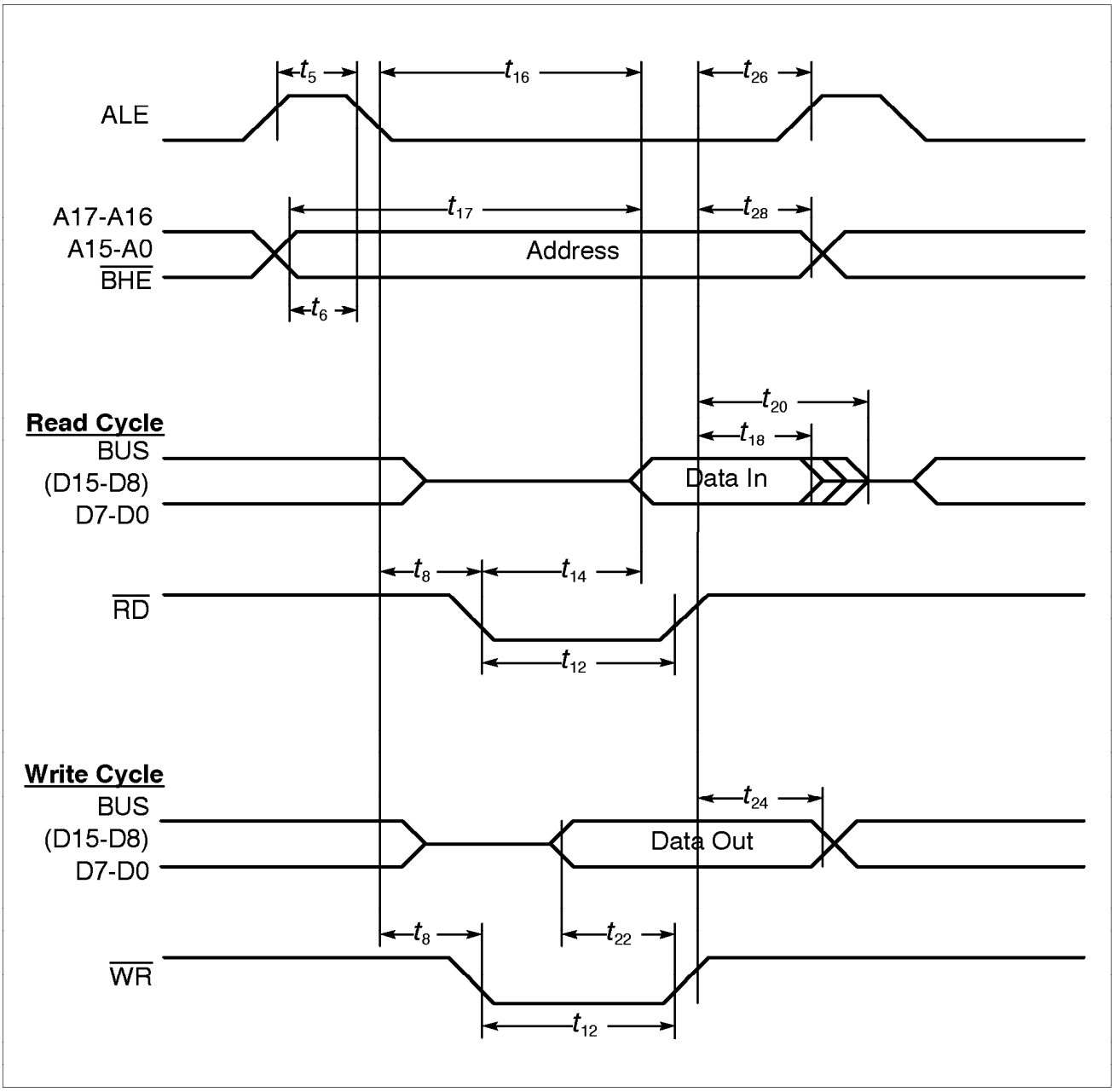


Figure 21
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Normal ALE

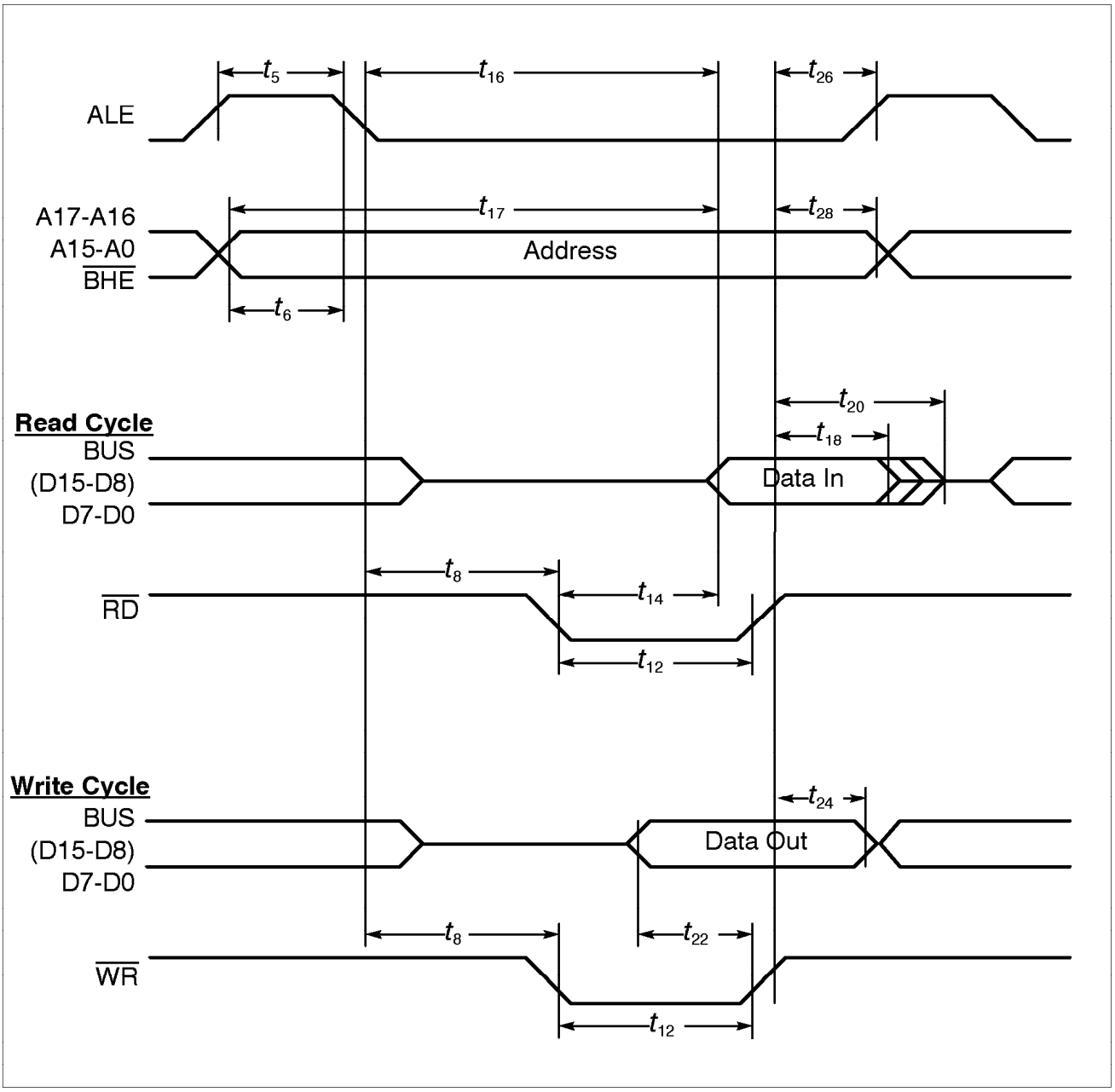


Figure 22
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE

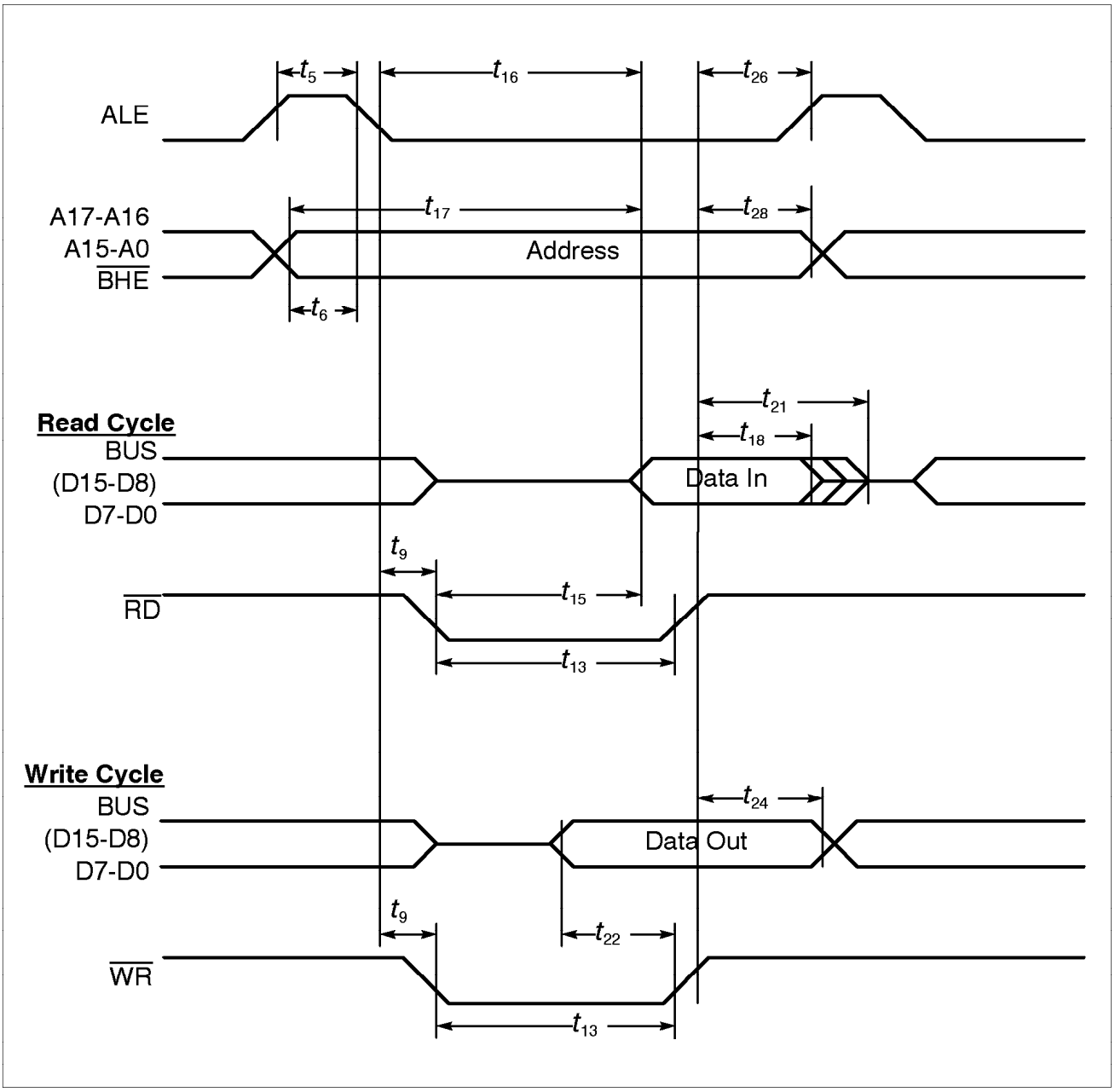


Figure 23
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE

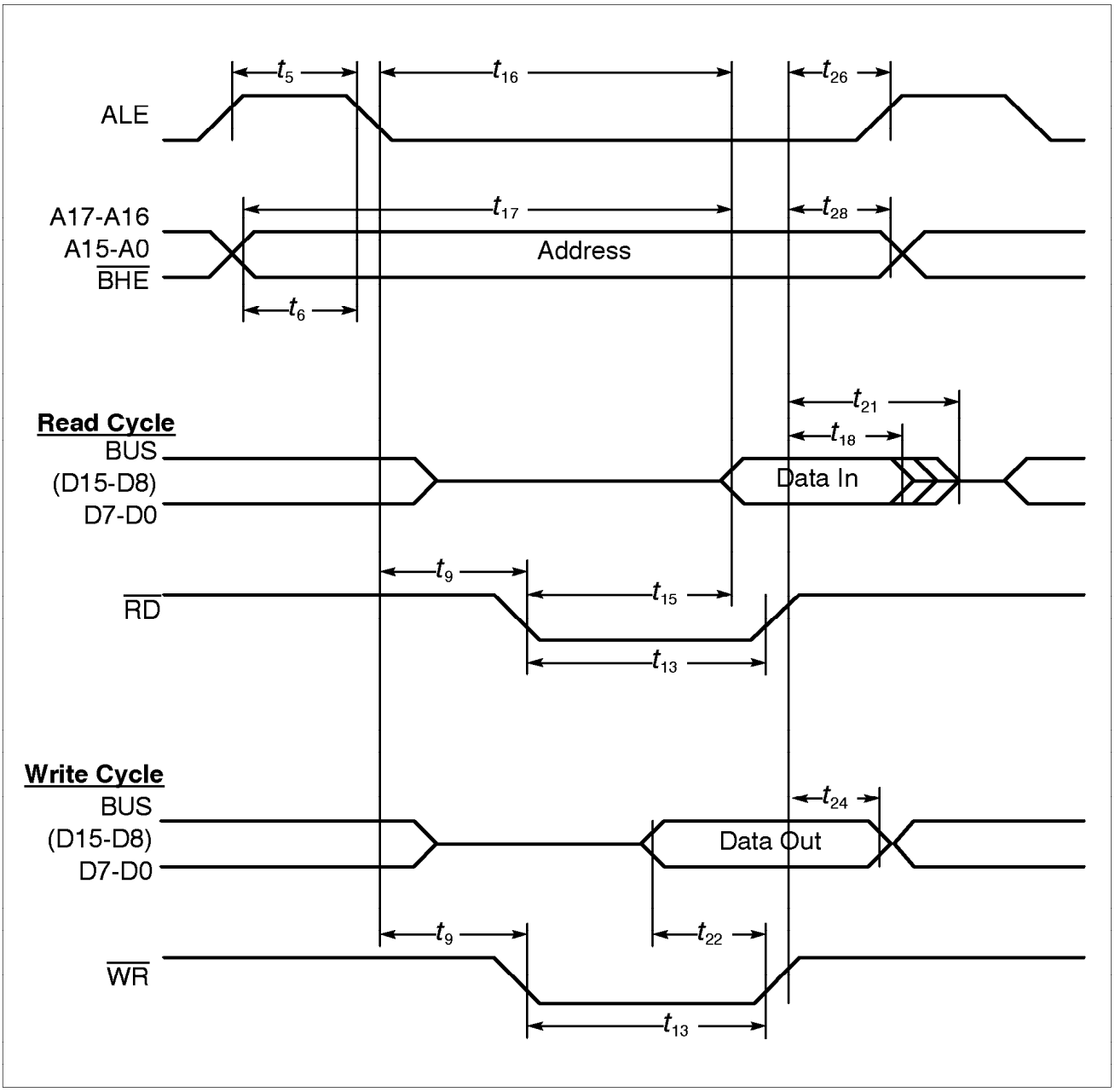


Figure 24
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

CLKOUT and $\overline{\text{READY}}$ for SAB 88C166

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166-5M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT) = 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	10	–	10	–	ns
Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	t_{36}	SR	10	–	10	–	ns
Asynchronous $\overline{\text{READY}}$ low time	t_{37}	SR	65	–	2TCL + 15	–	ns
Asynchronous $\overline{\text{READY}}$ setup time [1]	t_{58}	SR	20	–	20	–	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 + $2t_A$ + t_F [2]	0	TCL – 25 + $2t_A$ + 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 $\overline{\text{READY}}$.

AC Characteristics (cont'd)

CLKOUT and $\overline{\text{READY}}$ for SAB 88C166W

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166W-M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT) = 100 pF

Parameter	Symbol		CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock 1/CLP = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
CLKOUT cycle time	t_{29}	CC	62.5	62.5	CLP	CLP	ns
CLKOUT high time	t_{30}	CC	20	–	TCL_{min} – 5	–	ns
CLKOUT low time	t_{31}	CC	15	–	TCL_{min} – 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	10	–	10	–	ns
Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	t_{36}	SR	10	–	10	–	ns
Asynchronous $\overline{\text{READY}}$ low time	t_{37}	SR	77.5	–	CLP + 15	–	ns
Asynchronous $\overline{\text{READY}}$ setup time [1]	t_{58}	SR	20	–	20	–	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 + $2t_A$ + t_F [2]	0	TCL – 25 + $2t_A$ + 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 $2t_A$ refer to the next following bus cycle.

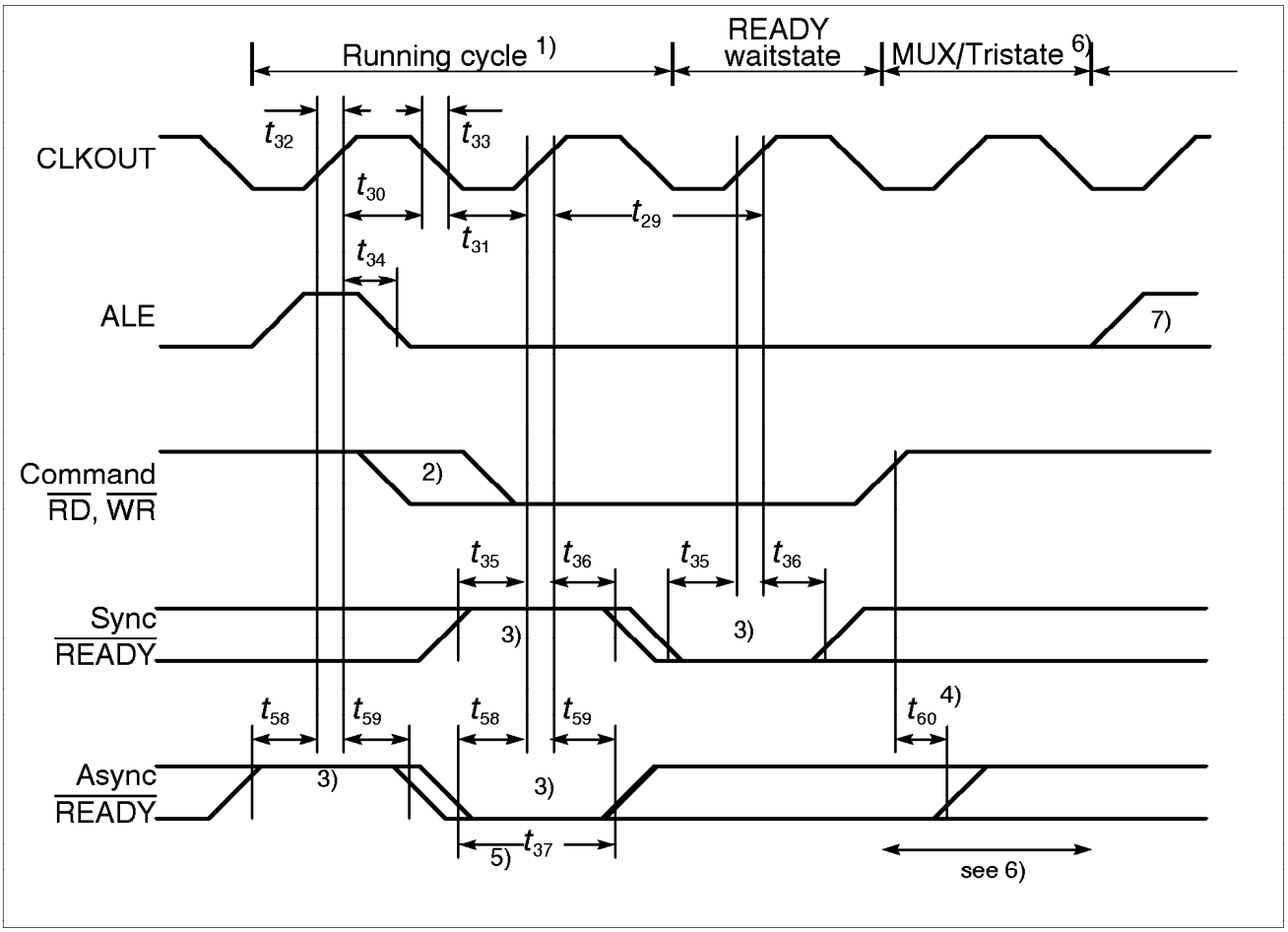


Figure 25
CLKOUT and $\overline{\text{READY}}$

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 (cont'd)
External Bus Arbitration

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166(W)-5M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 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	–	50	–	50	ns
CLKOUT to $\overline{HLDA}$ low or $\overline{BREQ}$ high delay	t_{63}	CC	–	50	–	50	ns
Other signals release	t_{66}	CC	–	25	–	25	ns
Other signals drive	t_{67}	CC	– 5	35	– 5	35	ns

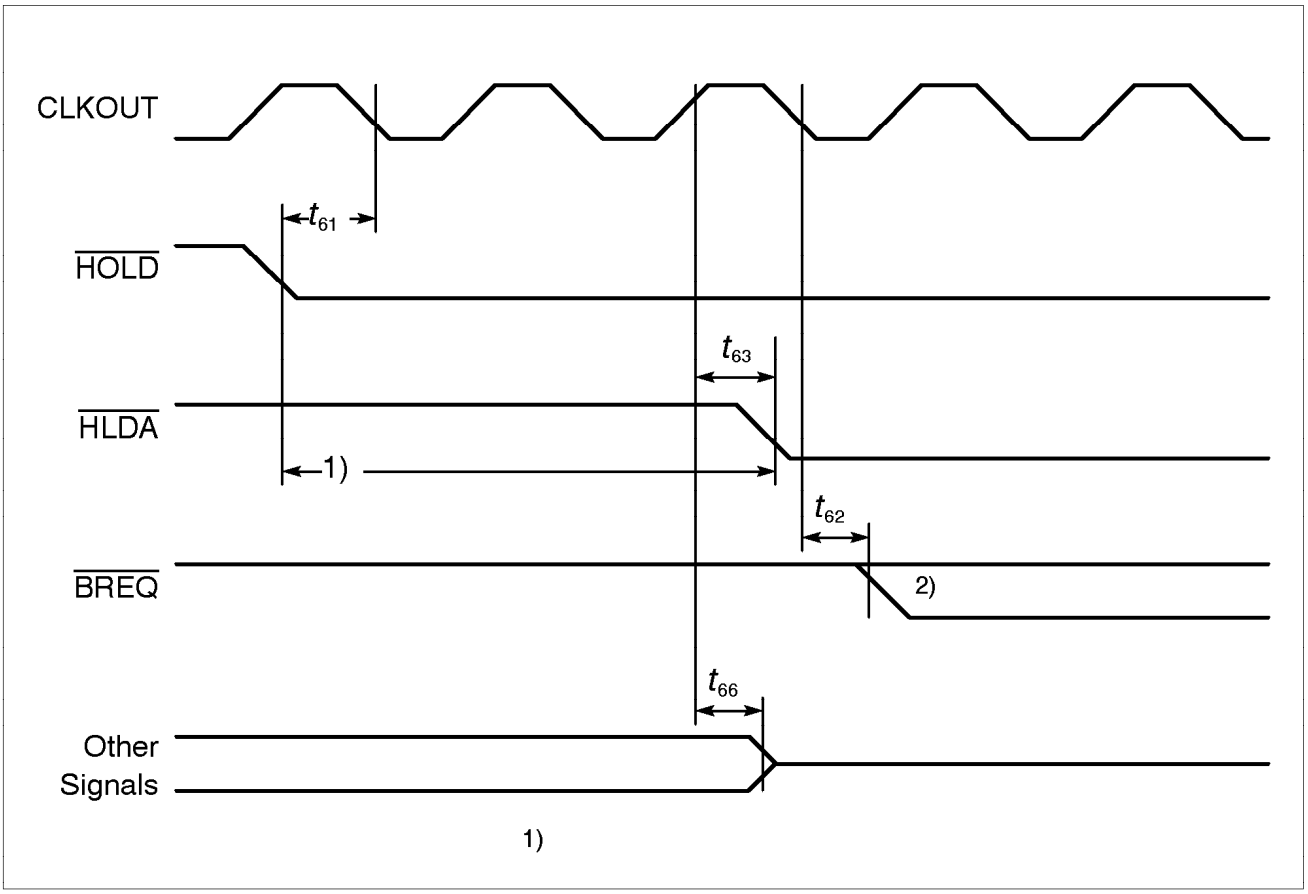


Figure 26
External Bus Arbitration, Releasing the Bus

Notes

1) The SAB 88C166(W) will complete the currently running bus cycle before granting bus access.

2) This is the first possibility for $\overline{\text{BREQ}}$ to get active.

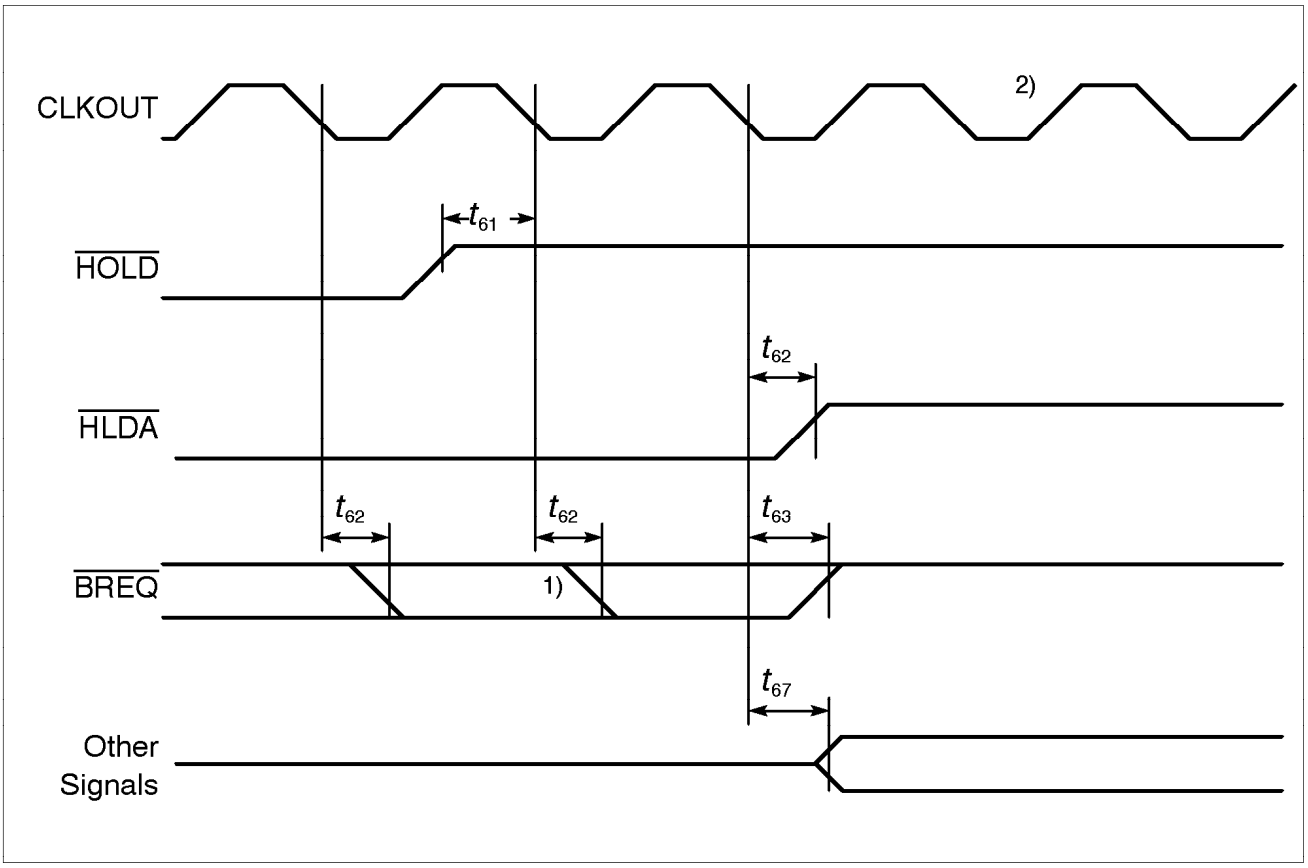


Figure 27
External Bus Arbitration, (Regaining the Bus)

Notes

1) This is the last chance for $\overline{\text{BREQ}}$ to trigger the indicated regain-sequence.
Even if $\overline{\text{BREQ}}$ is activated earlier, the regain-sequence is initiated by $\overline{\text{HOLD}}$ going high.
Please note that $\overline{\text{HOLD}}$ may also be deactivated without the SAB 88C166(W) requesting the bus.

2) The next SAB 88C166(W) driven bus cycle may start here.

Dim	mm			inches		
	Min	Typ	Max	Min	Typ	Max
A			3.30			0.130
A2	2.55	2.80	3.05	0.100	0.110	0.120
D	23.65	23.90	24.15	0.931	0.941	0.951
D1	19.90	20.00	20.10	0.783	0.787	0.791
D3		18.85			0.742	
E	17.65	17.90	18.15	0.695	0.705	0.715
E1	13.90	14.00	14.10	0.547	0.551	0.555
E3		12.35			0.486	
e		0.65			0.026	
Number of Pins						
ND	30					
NE	20					
N	100					

Figure 28
Package Outline Rectangular P-MQFP100