

# SIEMENS

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

SAB 80C166W/  
83C166W/  
83C166W

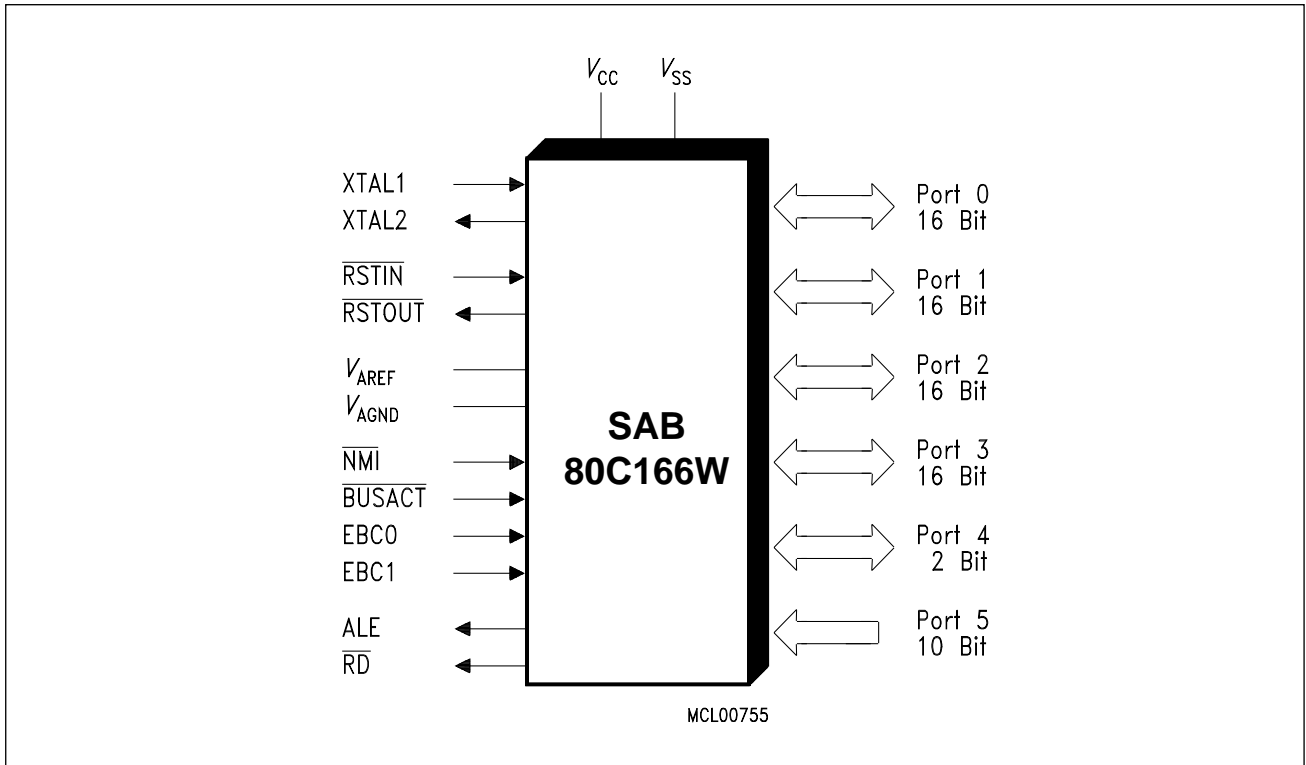
### Preliminary

## SAB 80C166W/83C166W / 83C166W 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
- 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 83C166W 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
- Direct clock input without 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 80C166W/83C166W is a 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. These devices derive the CPU clock signal (operating clock) directly from the on-chip oscillator without using a prescaler. This reduces the device's EME.



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

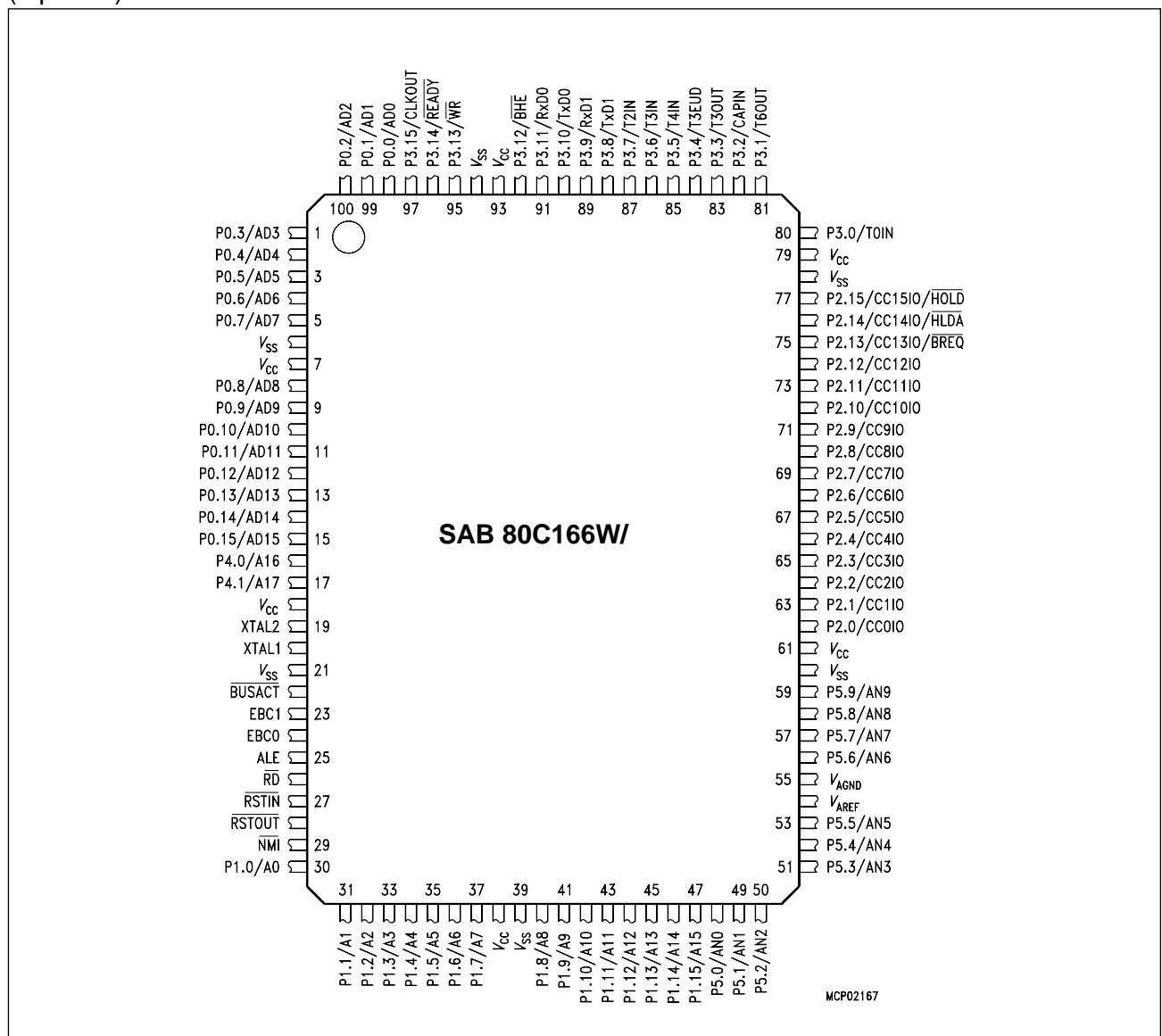
## Ordering Information

Type	Ordering Code	Package	Function
SAB 83C166W-5M	On Request	P-MQFP-100-2	16-bit microcontroller, 0 °C to +70 °C, 1 KByte RAM and 32 KByte ROM
SAB 83C166W-5M-T3	Q67120-D...	P-MQFP-100-2	16-bit microcontroller, -40 °C to +85 °C, 1 KByte RAM and 32 KByte ROM
SAB 83C166W-5M-T4	Q67120-D...	P-MQFP-100-2	16-bit microcontroller, -40 °C to +110 °C 1 KByte RAM and 32 KByte ROM
SAB 80C166W/83C166W-M	On Request	P-MQFP-100-2	16-bit microcontroller, 0 °C to +70 °C 1 KByte RAM

Type	Ordering Code	Package	Function
SAB 80C166W/ 83C166W-M-T3	Q67120-C864	P-MQFP-100-2	16-bit microcontroller, -40 °C to +85 °C 1 KByte RAM
SAB 80C166W/ 83C166W-M-T4	Q67120-C917	P-MQFP-100-2	16-bit microcontroller, -40 °C to +110 °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

Pin No.	Symbol	Input (I) Output (O)	Function																																				
16 - 17	P4.0 – P4.1	I/O	<p><b>Port 4</b> 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.</p> <p>In case of an external bus configuration, Port 4 can be used to output the segment address lines:</p>																																				
16		O	P4.0      A16      Least Significant Segment Addr. Line																																				
17		O	P4.1      A17      Most Significant Segment Addr. Line																																				
20	XTAL1	I	<p><b>XTAL1:</b> Input to the oscillator amplifier and input to the internal clock generator</p>																																				
19	XTAL2	O	<p><b>XTAL2:</b> Output of the oscillator amplifier circuit.</p> <p>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.</p>																																				
22 23 24	$\overline{\text{BUSACT}}$ , EBC1, EBC0	I I I	<p><b>External Bus Configuration</b> selection inputs. These pins are sampled during reset and select either the single chip mode or one of the four external bus configurations:</p> <table border="1"> <thead> <tr> <th><math>\overline{\text{BUSACT}}</math></th> <th>EBC1</th> <th>EBC0</th> <th>Mode/Bus Configuration</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>8-bit demultiplexed bus</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>8-bit multiplexed bus</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>16-bit multiplexed bus</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>16-bit demultiplexed bus</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>Single chip mode</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Reserved.</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Reserved.</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Reserved.</td> </tr> </tbody> </table> <p>ROMless versions must have pin <math>\overline{\text{BUSACT}}</math> tied to '0'.</p>	$\overline{\text{BUSACT}}$	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.
$\overline{\text{BUSACT}}$	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.																																				
27	$\overline{\text{RSTIN}}$	I	<p><b>Reset Input</b> with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the SAB 80C166W/83C166W. An internal pullup resistor permits power-on reset using only a capacitor connected to <math>V_{SS}</math>.</p>																																				
28	$\overline{\text{RSTOUT}}$	O	<p><b>Internal Reset Indication Output.</b> This pin is set to a low level when the part is executing either a hardware-, a software- or a watchdog timer reset. RSTOUT remains low until the EINIT (end of initialization) instruction is executed.</p>																																				

## Pin Definitions and Functions (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
29	$\overline{\text{NMI}}$	I	<b>Non-Maskable Interrupt</b> 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 80C166W/83C166W 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.
29	ALE	O	<b>Address Latch Enable</b> Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.
26	$\overline{\text{RD}}$	O	<b>External Memory Read Strobe.</b> $\overline{\text{RD}}$ is activated for every external instruction or data read access.
30 - 37 40 - 47	P1.0 – P1.15	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..
48 - 53 56 - 59	P5.0 – P5.9	I I	<b>Port 5</b> 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).
62 - 77	P2.0 – P2.15	I/O	<b>Port 2</b> 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:
62		I/O	P2.0 $\overline{\text{CC0IO}}$ CAPCOM: CC0 Cap.-In/Comp.Out
			...                      ...                      ...
75		I/O	P2.13 $\overline{\text{CC13IO}}$ CAPCOM: CC13 Cap.-In/Comp.Out,
		O	$\overline{\text{BREQ}}$ External Bus Request Output
76		I/O	P2.14 $\overline{\text{CC14IO}}$ CAPCOM: CC14 Cap.-In/Comp.Out,
		O	$\overline{\text{HLDA}}$ External Bus Hold Acknowl. Output
77		I/O	P2.15 $\overline{\text{CC15IO}}$ CAPCOM: CC15 Cap.-In/Comp.Out,
		I	$\overline{\text{HOLD}}$ External Bus Hold Request Input

## Pin Definitions and Functions (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function																		
80 - 92, 95 - 97	P3.0 – P3.15	I/O I/O	<p><b>Port 3</b> 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.</p> <p>The following Port 3 pins also serve for alternate functions:</p>																		
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 TxD0 ASC0 Clock/Data Output (Asyn./Syn.)																		
91		I/O	P3.11 RxD0 ASC0 Data Input (Asyn.) or I/O (Syn.)																		
92		O	P3.12 $\overline{\text{BHE}}$ Ext. Memory High Byte Enable Signal,																		
95		O	P3.13 $\overline{\text{WR}}$ External Memory Write Strobe																		
96		I	P3.14 $\overline{\text{READY}}$ Ready Signal Input																		
97		O	P3.15 CLKOUT System Clock Output (=CPU Clock)																		
98 – 5 8 – 15	P0.0 – P0.15	I/O	<p><b>Port 0</b> 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.</p> <p>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.</p> <p><b>Demultiplexed bus modes:</b></p> <table> <tr> <td>Data Path Width:</td> <td>8-bit</td> <td>16-bit</td> </tr> <tr> <td>P0.0 – P0.7:</td> <td>D0 – D7</td> <td>D0 - D7</td> </tr> <tr> <td>P0.8 – P0.15:</td> <td>output!</td> <td>D8 - D15</td> </tr> </table> <p><b>Multiplexed bus modes:</b></p> <table> <tr> <td>Data Path Width:</td> <td>8-bit</td> <td>16-bit</td> </tr> <tr> <td>P0.0 – P0.7:</td> <td>AD0 – AD7</td> <td>AD0 - AD7</td> </tr> <tr> <td>P0.8 – P0.15:</td> <td>A8 - A15</td> <td>AD8 - AD15</td> </tr> </table>	Data Path Width:	8-bit	16-bit	P0.0 – P0.7:	D0 – D7	D0 - D7	P0.8 – P0.15:	output!	D8 - D15	Data Path Width:	8-bit	16-bit	P0.0 – P0.7:	AD0 – AD7	AD0 - AD7	P0.8 – P0.15:	A8 - A15	AD8 - AD15
Data Path Width:	8-bit	16-bit																			
P0.0 – P0.7:	D0 – D7	D0 - D7																			
P0.8 – P0.15:	output!	D8 - D15																			
Data Path Width:	8-bit	16-bit																			
P0.0 – P0.7:	AD0 – AD7	AD0 - AD7																			
P0.8 – P0.15:	A8 - A15	AD8 - AD15																			
54	V <sub>AREF</sub>	-	<b>Reference voltage</b> for the A/D converter.																		
55	V <sub>AGND</sub>	-	<b>Reference ground</b> for the A/D converter.																		

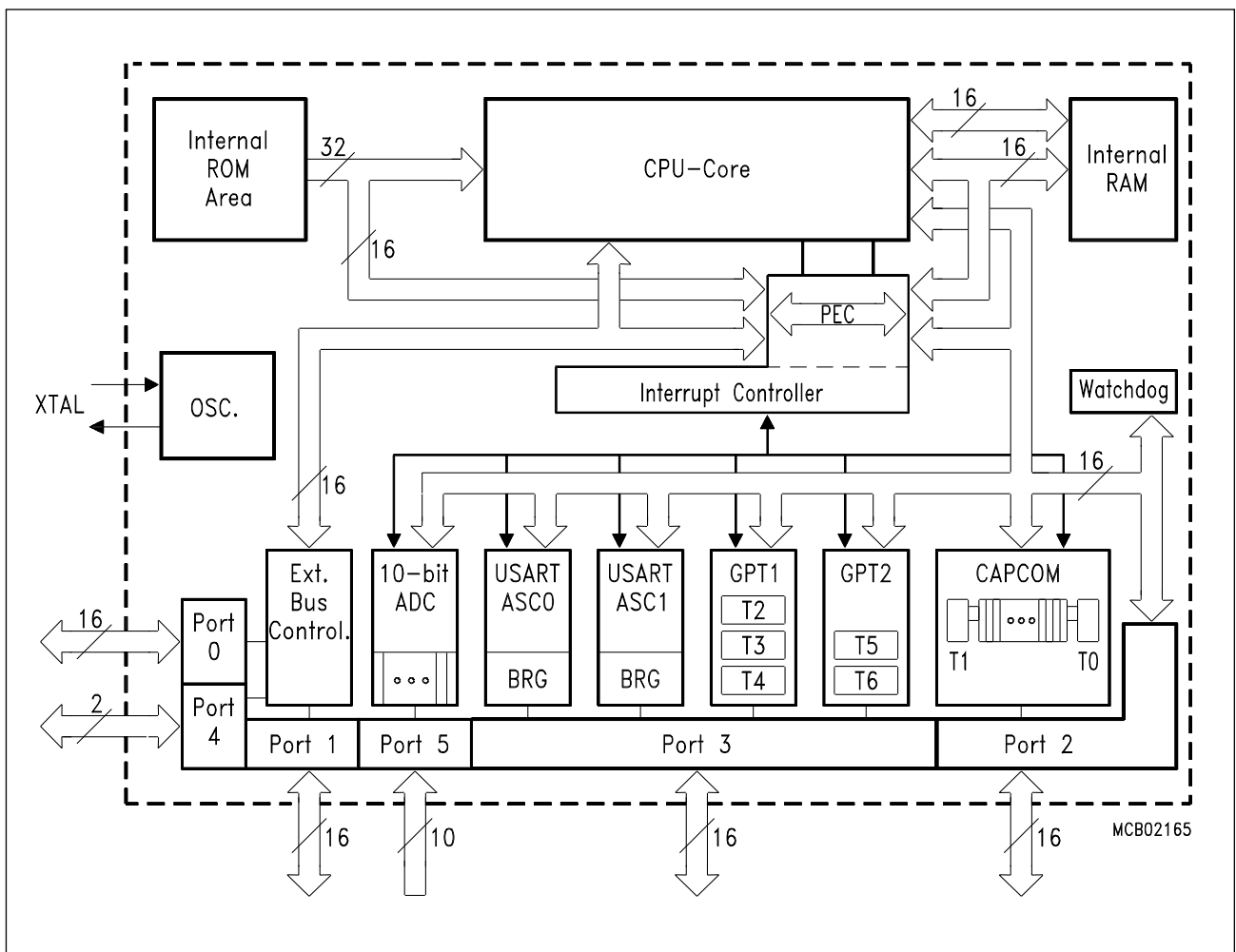
### Pin Definitions and Functions (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
7, 18, 38, 61, 79, 93	$V_{CC}$	-	<b>Digital Supply Voltage:</b> + 5 V during normal operation and idle mode. $\geq 2.5$ V during power down mode
6, 21, 39, 60, 78, 94	$V_{SS}$	-	<b>Digital Ground.</b>

## Functional Description

The architecture of the SAB 80C166W/83C166W 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 80C166W/83C166W.

**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 80C166W/83C166W 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 bitwise or wordwise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

The SAB 83C166W 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 bitwise (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, ie. 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.

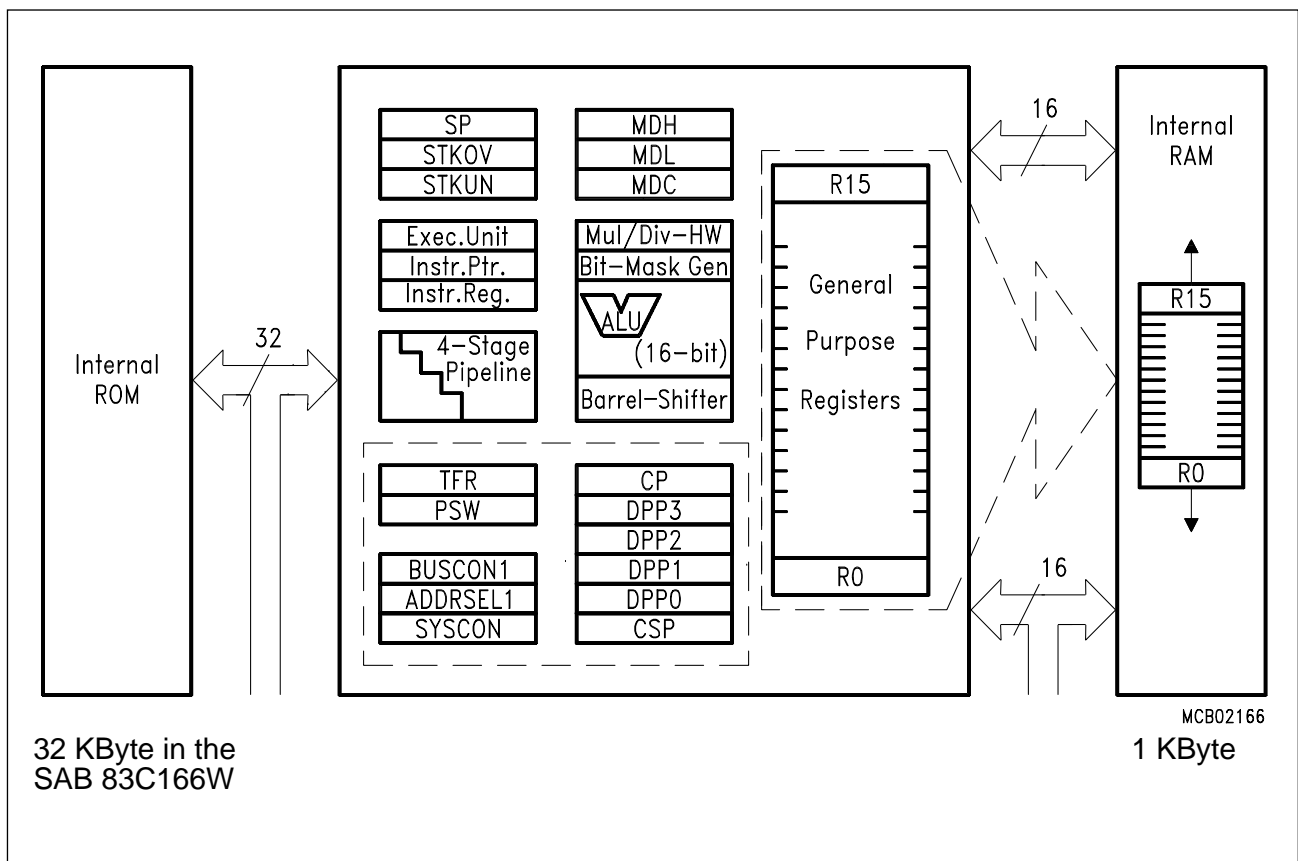
For applications which require less than 64 KBytes of external memory space, a non-segmented memory model can be selected. In this case all memory locations can be addressed by 16 bits and Port 4 is not required to output the additional segment address lines.

## 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 80C166W/83C166W'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  $16 \times 16$  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 banks 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 80C166W/83C166W 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 80C166W/83C166W is capable of reacting very fast to the occurrence of non-deterministic events.

The architecture of the SAB 80C166W/83C166W 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 implicitly 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 80C166W/83C166W 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 80C166W/83C166W 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 <sub>H</sub>	10 <sub>H</sub>
CAPCOM Register 1	CC1IR	CC1IE	CC1INT	44 <sub>H</sub>	11 <sub>H</sub>
CAPCOM Register 2	CC2IR	CC2IE	CC2INT	48 <sub>H</sub>	12 <sub>H</sub>
CAPCOM Register 3	CC3IR	CC3IE	CC3INT	4C <sub>H</sub>	13 <sub>H</sub>
CAPCOM Register 4	CC4IR	CC4IE	CC4INT	50 <sub>H</sub>	14 <sub>H</sub>
CAPCOM Register 5	CC5IR	CC5IE	CC5INT	54 <sub>H</sub>	15 <sub>H</sub>
CAPCOM Register 6	CC6IR	CC6IE	CC6INT	58 <sub>H</sub>	16 <sub>H</sub>
CAPCOM Register 7	CC7IR	CC7IE	CC7INT	5C <sub>H</sub>	17 <sub>H</sub>
CAPCOM Register 8	CC8IR	CC8IE	CC8INT	60 <sub>H</sub>	18 <sub>H</sub>
CAPCOM Register 9	CC9IR	CC9IE	CC9INT	64 <sub>H</sub>	19 <sub>H</sub>
CAPCOM Register 10	CC10IR	CC10IE	CC10INT	68 <sub>H</sub>	1A <sub>H</sub>
CAPCOM Register 11	CC11IR	CC11IE	CC11INT	6C <sub>H</sub>	1B <sub>H</sub>
CAPCOM Register 12	CC12IR	CC12IE	CC12INT	70 <sub>H</sub>	1C <sub>H</sub>
CAPCOM Register 13	CC13IR	CC13IE	CC13INT	74 <sub>H</sub>	1D <sub>H</sub>
CAPCOM Register 14	CC14IR	CC14IE	CC14INT	78 <sub>H</sub>	1E <sub>H</sub>
CAPCOM Register 15	CC15IR	CC15IE	CC15INT	7C <sub>H</sub>	1F <sub>H</sub>
CAPCOM Timer 0	T0IR	T0IE	T0INT	80 <sub>H</sub>	20 <sub>H</sub>
CAPCOM Timer 1	T1IR	T1IE	T1INT	84 <sub>H</sub>	21 <sub>H</sub>
GPT1 Timer 2	T2IR	T2IE	T2INT	88 <sub>H</sub>	22 <sub>H</sub>
GPT1 Timer 3	T3IR	T3IE	T3INT	8C <sub>H</sub>	23 <sub>H</sub>
GPT1 Timer 4	T4IR	T4IE	T4INT	90 <sub>H</sub>	24 <sub>H</sub>
GPT2 Timer 5	T5IR	T5IE	T5INT	94 <sub>H</sub>	25 <sub>H</sub>
GPT2 Timer 6	T6IR	T6IE	T6INT	98 <sub>H</sub>	26 <sub>H</sub>
GPT2 CAPREL Register	CRIR	CRIE	CRINT	9C <sub>H</sub>	27 <sub>H</sub>
A/D Conversion Complete	ADCIR	ADCIE	ADCINT	A0 <sub>H</sub>	28 <sub>H</sub>
A/D Overrun Error	ADEIR	ADEIE	ADEINT	A4 <sub>H</sub>	29 <sub>H</sub>
ASC0 Transmit	S0TIR	S0TIE	S0TINT	A8 <sub>H</sub>	2A <sub>H</sub>
ASC0 Receive	S0RIR	S0RIE	S0RINT	AC <sub>H</sub>	2B <sub>H</sub>
ASC0 Error	S0EIR	S0EIE	S0EINT	B0 <sub>H</sub>	2C <sub>H</sub>
ASC1 Transmit	S1TIR	S1TIE	S1TINT	B4 <sub>H</sub>	2D <sub>H</sub>
ASC1 Receive	S1RIR	S1RIE	S1RINT	B8 <sub>H</sub>	2E <sub>H</sub>
ASC1 Error	S1EIR	S1EIE	S1EINT	BC <sub>H</sub>	2F <sub>H</sub>

The SAB 80C166W/83C166W 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 Software Reset Watchdog Timer Overflow		RESET RESET RESET	0000 <sub>H</sub> 0000 <sub>H</sub> 0000 <sub>H</sub>	00 <sub>H</sub> 00 <sub>H</sub> 00 <sub>H</sub>	III III III
Class A Hardware Traps: Non-Maskable Interrupt Stack Overflow Stack Underflow	NMI STKOF STKUF	NMITRAP STOTRAP STUTRAP	0008 <sub>H</sub> 0010 <sub>H</sub> 0018 <sub>H</sub>	02 <sub>H</sub> 04 <sub>H</sub> 06 <sub>H</sub>	II II II
Class B Hardware Traps: Undefined Opcode Protected Instruction Fault Illegal Word Operand Access Illegal Instruction Access Illegal External Bus Access	UNDOPC PRTFLT ILLOPA ILLINA ILLBUS	BTRAP BTRAP BTRAP BTRAP BTRAP	0028 <sub>H</sub> 0028 <sub>H</sub> 0028 <sub>H</sub> 0028 <sub>H</sub> 0028 <sub>H</sub>	0A <sub>H</sub> 0A <sub>H</sub> 0A <sub>H</sub> 0A <sub>H</sub> 0A <sub>H</sub>	I I I I I
Reserved			[002C <sub>H</sub> – 003C <sub>H</sub> ]	[0B <sub>H</sub> – 0F <sub>H</sub> ]	
Software Traps TRAP Instruction			Any [0000 <sub>H</sub> – 01FC <sub>H</sub> ] in steps of 04 <sub>H</sub>	Any [00 <sub>H</sub> – 7F <sub>H</sub> ]	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.

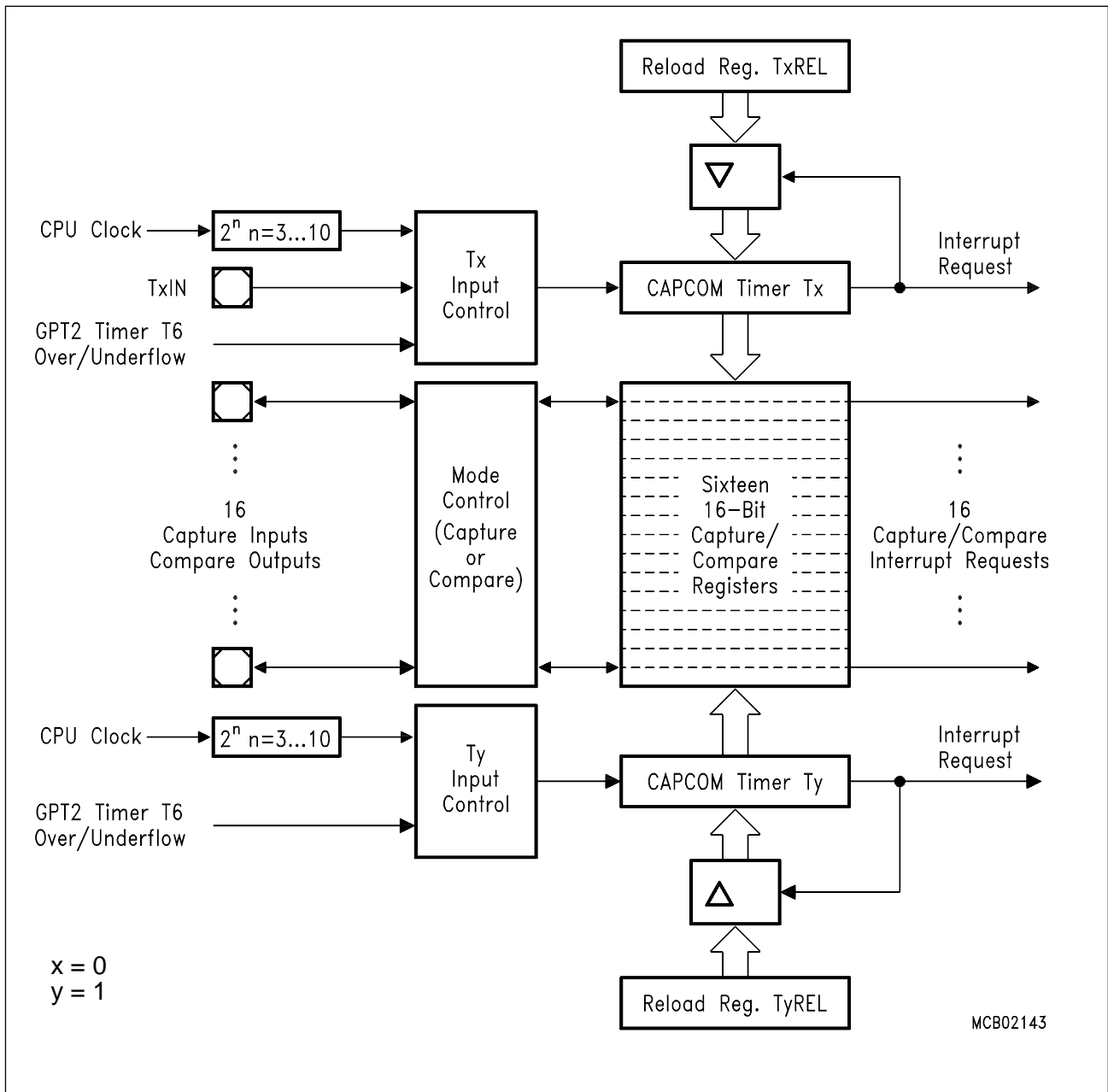


Figure 5  
CAPCOM-Unit Block Diagram



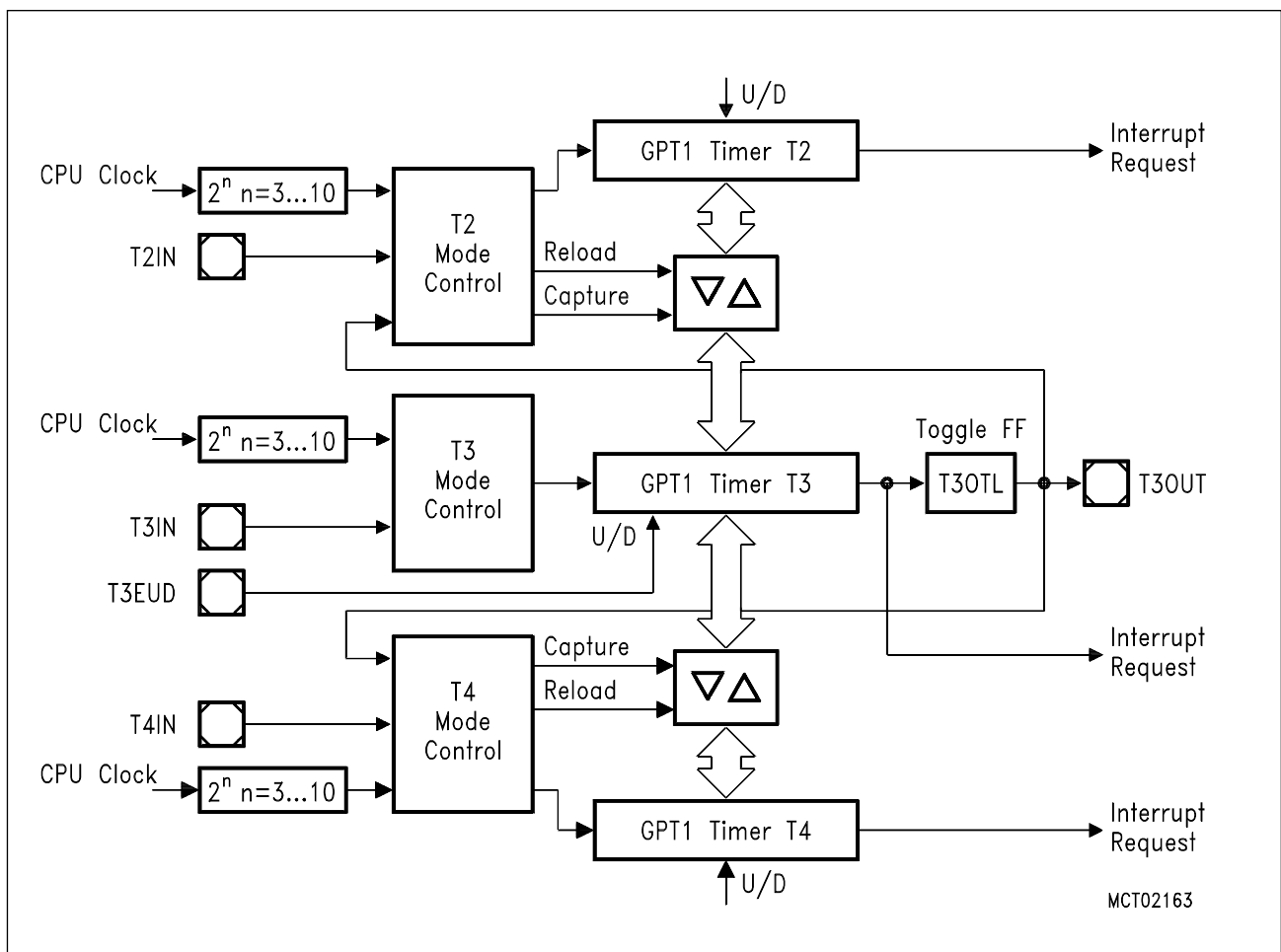
## General Purpose Timer (GPT) Unit

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).

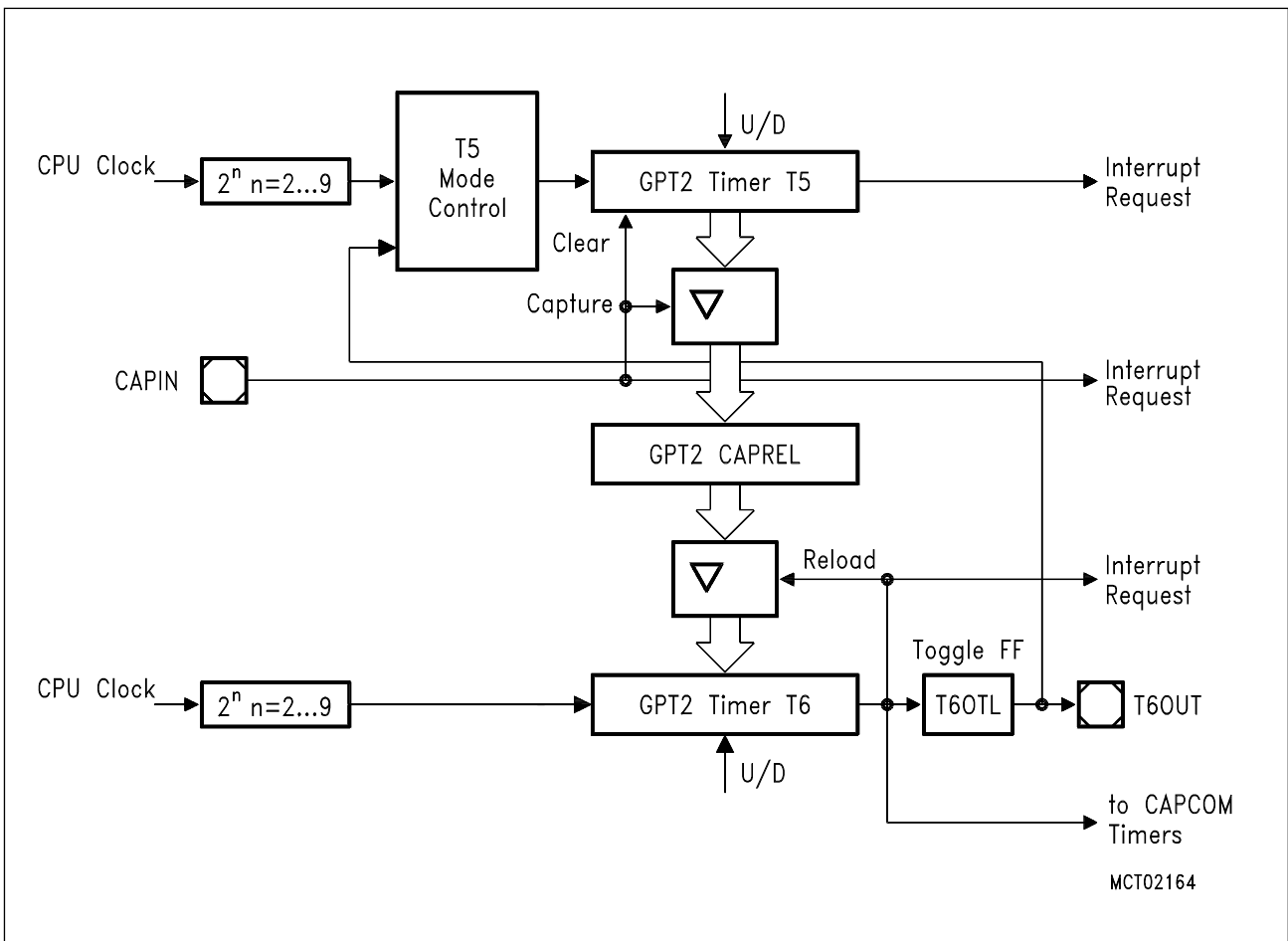


**Figure 6**  
**Block Diagram of GPT1**

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 overflow/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 80C166W/83C166W 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 80C166W/83C166W 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 80C166W/83C166W.

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 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  $\mu$ s 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 80C166W/83C166W provides a built-in bootstrap loader (BSL), which allows to start program execution out of the SAB 80C166W/83C166W's internal RAM. This start program is loaded via the serial interface ASC0 and does not require external memory or an internal ROM.

The SAB 80C166W/83C166W enters BSL mode, when ALE is sampled high at the end of a hardware reset and if 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 BUSACT.

After entering BSL mode the SAB 80C166W/83C166W scans the RXD0 line to receive a zero byte, ie. 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 80C166W/83C166W returns the value <55<sub>H</sub>>.

The next 32 bytes received via ASC0 are stored sequentially into locations 0FA40<sub>H</sub> through 0FA5F<sub>H</sub> of the internal RAM. To execute the loaded code the BSL then jumps to location 0FA40<sub>H</sub>. The loaded program may load additional code / data, change modes, etc.

The SAB 80C166W/83C166W 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 summary on the following pages lists the instructions of the SAB 80C166W/83C166W ordered into logical groups.

## Instruction Set Summary

Mnemonic		Description	Bytes
<b>Arithmetic Operations</b>			
ADD	Rw, Rw	Add direct word GPR to direct GPR	2
ADD	Rw, [Rw]	Add indirect word memory to direct GPR	2
ADD	Rw, [Rw +]	Add indirect word memory to direct GPR and post-increment source pointer by 2	2
ADD	Rw, #data3	Add immediate word data to direct GPR	2
ADD	reg, #data16	Add immediate word data to direct register	4
ADD	reg, mem	Add direct word memory to direct register	4
ADD	mem, reg	Add direct word register to direct memory	4
ADDB	Rb, Rb	Add direct byte GPR to direct GPR	2
ADDB	Rb, [Rw]	Add indirect byte memory to direct GPR	2
ADDB	Rb, [Rw +]	Add indirect byte memory to direct GPR and post-increment source pointer by 1	2
ADDB	Rb, #data3	Add immediate byte data to direct GPR	2
ADDB	reg, #data8	Add immediate byte data to direct register	4
ADDB	reg, mem	Add direct byte memory to direct register	4
ADDB	mem, reg	Add direct byte register to direct memory	4
ADDC	Rw, Rw	Add direct word GPR to direct GPR with Carry	2
ADDC	Rw, [Rw]	Add indirect word memory to direct GPR with Carry	2
ADDC	Rw, [Rw +]	Add indirect word memory to direct GPR with Carry and post-increment source pointer by 2	2
ADDC	Rw, #data3	Add immediate word data to direct GPR with Carry	2
ADDC	reg, #data16	Add immediate word data to direct register with Carry	4
ADDC	reg, mem	Add direct word memory to direct register with Carry	4
ADDC	mem, reg	Add direct word register to direct memory with Carry	4
ADDCB	Rb, Rb	Add direct byte GPR to direct GPR with Carry	2
ADDCB	Rb, [Rw]	Add indirect byte memory to direct GPR with Carry	2
ADDCB	Rb, [Rw +]	Add indirect byte memory to direct GPR with Carry and post-increment source pointer by 1	2
ADDCB	Rb, #data3	Add immediate byte data to direct GPR with Carry	2
ADDCB	reg, #data8	Add immediate byte data to direct register with Carry	4
ADDCB	reg, mem	Add direct byte memory to direct register with Carry	4

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Arithmetic Operations (cont'd)

ADDCB	mem, reg	Add direct byte register to direct memory with Carry	4
SUB	Rw, Rw	Subtract direct word GPR from direct GPR	2
SUB	Rw, [Rw]	Subtract indirect word memory from direct GPR	2
SUB	Rw, [Rw +]	Subtract indirect word memory from direct GPR and post-increment source pointer by 2	2
SUB	Rw, #data3	Subtract immediate word data from direct GPR	2
SUB	reg, #data16	Subtract immediate word data from direct register	4
SUB	reg, mem	Subtract direct word memory from direct register	4
SUB	mem, reg	Subtract direct word register from direct memory	4
SUBB	Rb, Rb	Subtract direct byte GPR from direct GPR	2
SUBB	Rb, [Rw]	Subtract indirect byte memory from direct GPR	2
SUBB	Rb, [Rw +]	Subtract indirect byte memory from direct GPR and post-increment source pointer by 1	2
SUBB	Rb, #data3	Subtract immediate byte data from direct GPR	2
SUBB	reg, #data8	Subtract immediate byte data from direct register	4
SUBB	reg, mem	Subtract direct byte memory from direct register	4
SUBB	mem, reg	Subtract direct byte register from direct memory	4
SUBC	Rw, Rw	Subtract direct word GPR from direct GPR with Carry	2
SUBC	Rw, [Rw]	Subtract indirect word memory from direct GPR with Carry	2
SUBC	Rw, [Rw +]	Subtract indirect word memory from direct GPR with Carry and post-increment source pointer by 2	2
SUBC	Rw, #data3	Subtract immediate word data from direct GPR with Carry	2
SUBC	reg, #data16	Subtract immediate word data from direct register with Carry	4
SUBC	reg, mem	Subtract direct word memory from direct register with Carry	4
SUBC	mem, reg	Subtract direct word register from direct memory with Carry	4
SUBCB	Rb, Rb	Subtract direct byte GPR from direct GPR with Carry	2
SUBCB	Rb, [Rw]	Subtract indirect byte memory from direct GPR with Carry	2
SUBCB	Rb, [Rw +]	Subtract indirect byte memory from direct GPR with Carry and post-increment source pointer by 1	2
SUBCB	Rb, #data3	Subtract immediate byte data from direct GPR with Carry	2
SUBCB	reg, #data8	Subtract immediate byte data from direct register with Carry	4

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Arithmetic Operations (cont'd)

SUBCB	reg, mem	Subtract direct byte memory from direct register with Carry	4
SUBCB	mem, reg	Subtract direct byte register from direct memory with Carry	4
MUL	Rw, Rw	Signed multiply direct GPR by direct GPR (16-16-bit)	2
MULU	Rw, Rw	Unsigned multiply direct GPR by direct GPR (16-16-bit)	2
DIV	Rw	Signed divide register MDL by direct GPR (16-/16-bit)	2
DIVL	Rw	Signed long divide register MD by direct GPR (32-/16-bit)	2
DIVLU	Rw	Unsigned long divide register MD by direct GPR (32-/16-bit)	2
DIVU	Rw	Unsigned divide register MDL by direct GPR (16-/16-bit)	2
CPL	Rw	Complement direct word GPR	2
CPLB	Rb	Complement direct byte GPR	2
NEG	Rw	Negate direct word GPR	2
NEGB	Rb	Negate direct byte GPR	2

### Logical Instructions

AND	Rw, Rw	Bitwise AND direct word GPR with direct GPR	2
AND	Rw, [Rw]	Bitwise AND indirect word memory with direct GPR	2
AND	Rw, [Rw +]	Bitwise AND indirect word memory with direct GPR and post-increment source pointer by 2	2
AND	Rw, #data3	Bitwise AND immediate word data with direct GPR	2
AND	reg, #data16	Bitwise AND immediate word data with direct register	4
AND	reg, mem	Bitwise AND direct word memory with direct register	4
AND	mem, reg	Bitwise AND direct word register with direct memory	4
ANDB	Rb, Rb	Bitwise AND direct byte GPR with direct GPR	2
ANDB	Rb, [Rw]	Bitwise AND indirect byte memory with direct GPR	2
ANDB	Rb, [Rw +]	Bitwise AND indirect byte memory with direct GPR and post-increment source pointer by 1	2
ANDB	Rb, #data3	Bitwise AND immediate byte data with direct GPR	2
ANDB	reg, #data8	Bitwise AND immediate byte data with direct register	4
ANDB	reg, mem	Bitwise AND direct byte memory with direct register	4
ANDB	mem, reg	Bitwise AND direct byte register with direct memory	4



## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Logical Instructions (cont'd)

OR	Rw, Rw	Bitwise OR direct word GPR with direct GPR	2
OR	Rw, [Rw]	Bitwise OR indirect word memory with direct GPR	2
OR	Rw, [Rw +]	Bitwise OR indirect word memory with direct GPR and post-increment source pointer by 2	2
OR	Rw, #data3	Bitwise OR immediate word data with direct GPR	2
OR	reg, #data16	Bitwise OR immediate word data with direct register	4
OR	reg, mem	Bitwise OR direct word memory with direct register	4
OR	mem, reg	Bitwise OR direct word register with direct memory	4
ORB	Rb, Rb	Bitwise OR direct byte GPR with direct GPR	2
ORB	Rb, [Rw]	Bitwise OR indirect byte memory with direct GPR	2
ORB	Rb, [Rw +]	Bitwise OR indirect byte memory with direct GPR and post-increment source pointer by 1	2
ORB	Rb, #data3	Bitwise OR immediate byte data with direct GPR	2
ORB	reg, #data8	Bitwise OR immediate byte data with direct register	4
ORB	reg, mem	Bitwise OR direct byte memory with direct register	4
ORB	mem, reg	Bitwise OR direct byte register with direct memory	4
XOR	Rw, Rw	Bitwise XOR direct word GPR with direct GPR	2
XOR	Rw, [Rw]	Bitwise XOR indirect word memory with direct GPR	2
XOR	Rw, [Rw +]	Bitwise XOR indirect word memory with direct GPR and post-increment source pointer by 2	2
XOR	Rw, #data3	Bitwise XOR immediate word data with direct GPR	2
XOR	reg, #data16	Bitwise XOR immediate word data with direct register	4
XOR	reg, mem	Bitwise XOR direct word memory with direct register	4
XOR	mem, reg	Bitwise XOR direct word register with direct memory	4
XORB	Rb, Rb	Bitwise XOR direct byte GPR with direct GPR	2
XORB	Rb, [Rw]	Bitwise XOR indirect byte memory with direct GPR	2
XORB	Rb, [Rw +]	Bitwise XOR indirect byte memory with direct GPR and post-increment source pointer by 1	2
XORB	Rb, #data3	Bitwise XOR immediate byte data with direct GPR	2
XORB	reg, #data8	Bitwise XOR immediate byte data with direct register	4
XORB	reg, mem	Bitwise XOR direct byte memory with direct register	4
XORB	mem, reg	Bitwise XOR direct byte register with direct memory	4

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Boolean Bit Manipulation Operations

BCLR	bitaddr	Clear direct bit	2
BSET	bitaddr	Set direct bit	2
BMOV	bitaddr, bitaddr	Move direct bit to direct bit	4
BMOVN	bitaddr, bitaddr	Move negated direct bit to direct bit	4
BAND	bitaddr, bitaddr	AND direct bit with direct bit	4
BOR	bitaddr, bitaddr	OR direct bit with direct bit	4
BXOR	bitaddr, bitaddr	XOR direct bit with direct bit	4
BCMP	bitaddr, bitaddr	Compare direct bit to direct bit	4
BFLDH	bitoff, #mask8, #data8	Bitwise modify masked high byte of bit-addressable direct word memory with immediate data	4
BFLDL	bitoff, #mask8, #data8	Bitwise modify masked low byte of bit-addressable direct word memory with immediate data	4
CMP	Rw, Rw	Compare direct word GPR to direct GPR	2
CMP	Rw, [Rw]	Compare indirect word memory to direct GPR	2
CMP	Rw, [Rw +]	Compare indirect word memory to direct GPR and post-increment source pointer by 2	2
CMP	Rw, #data3	Compare immediate word data to direct GPR	2
CMP	reg, #data16	Compare immediate word data to direct register	4
CMP	reg, mem	Compare direct word memory to direct register	4
CMPB	Rb, Rb	Compare direct byte GPR to direct GPR	2
CMPB	Rb, [Rw]	Compare indirect byte memory to direct GPR	2
CMPB	Rb, [Rw +]	Compare indirect byte memory to direct GPR and post-increment source pointer by 1	2
CMPB	Rb, #data3	Compare immediate byte data to direct GPR	2
CMPB	reg, #data8	Compare immediate byte data to direct register	4
CMPB	reg, mem	Compare direct byte memory to direct register	4

### Compare and Loop Control Instructions

CMPD1	Rw, #data4	Compare immediate word data to direct GPR and decrement GPR by 1	2
CMPD1	Rw, #data16	Compare immediate word data to direct GPR and decrement GPR by 1	4

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Compare and Loop Control Instructions (cont'd)

CMPD1	Rw, mem	Compare direct word memory to direct GPR and decrement GPR by 1	4
CMPD2	Rw, #data4	Compare immediate word data to direct GPR and decrement GPR by 2	2
CMPD2	Rw, #data16	Compare immediate word data to direct GPR and decrement GPR by 2	4
CMPD2	Rw, mem	Compare direct word memory to direct GPR and decrement GPR by 2	4
CMPI1	Rw, #data4	Compare immediate word data to direct GPR and increment GPR by 1	2
CMPI1	Rw, #data16	Compare immediate word data to direct GPR and increment GPR by 1	4
CMPI1	Rw, mem	Compare direct word memory to direct GPR and increment GPR by 1	4
CMPI2	Rw, #data4	Compare immediate word data to direct GPR and increment GPR by 2	2
CMPI2	Rw, #data16	Compare immediate word data to direct GPR and increment GPR by 2	4
CMPI2	Rw, mem	Compare direct word memory to direct GPR and increment GPR by 2	4

### Prioritize Instruction

PRIOR	Rw, Rw	Determine number of shift cycles to normalize direct word GPR and store result in direct word GPR	2
-------	--------	---	---

### Shift and Rotate Instructions

SHL	Rw, Rw	Shift left direct word GPR; number of shift cycles specified by direct GPR	2
SHL	Rw, #data4	Shift left direct word GPR; number of shift cycles specified by immediate data	2
SHR	Rw, Rw	Shift right direct word GPR; number of shift cycles specified by direct GPR	2

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Shift and Rotate Instructions (cont'd)

SHR	Rw, #data4	Shift right direct word GPR; number of shift cycles specified by immediate data	2
ROL	Rw, Rw	Rotate left direct word GPR; number of shift cycles specified by direct GPR	2
ROL	Rw, #data4	Rotate left direct word GPR; number of shift cycles specified by immediate data	2
ROR	Rw, Rw	Rotate right direct word GPR; number of shift cycles specified by direct GPR	2
ROR	Rw, #data4	Rotate right direct word GPR; number of shift cycles specified by immediate data	2
ASHR	Rw, Rw	Arithmetic (sign bit) shift right direct word GPR; number of shift cycles specified by direct GPR	2
ASHR	Rw, #data4	Arithmetic (sign bit) shift right direct word GPR; number of shift cycles specified by immediate data	2

### Data Movement

MOV	Rw, Rw	Move direct word GPR to direct GPR	2
MOV	Rw, #data4	Move immediate word data to direct GPR	2
MOV	reg, #data16	Move immediate word data to direct register	4
MOV	Rw, [Rw]	Move indirect word memory to direct GPR	2
MOV	Rw, [Rw+]	Move indirect word memory to direct GPR and post-increment source pointer by 2	2
MOV	[Rw], Rw	Move direct word GPR to indirect memory	2
MOV	[-Rw], Rw	Pre-decrement destination pointer by 2 and move direct word GPR to indirect memory	2
MOV	[Rw], [Rw]	Move indirect word memory to indirect memory	2
MOV	[Rw+], [Rw]	Move indirect word memory to indirect memory and post-increment destination pointer by 2	2
MOV	[Rw], [Rw+]	Move indirect word memory to indirect memory and post-increment source pointer by 2	2
MOV	Rw, [Rw + #data16]	Move indirect word memory by base plus constant to direct GPR	4
MOV	[Rw + #data16], Rw	Move direct word GPR to indirect memory by base plus constant	4

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
<b>Data Movement (cont'd)</b>		
MOV [Rw], mem	Move direct word memory to indirect memory	4
MOV mem, [Rw]	Move indirect word memory to direct memory	4
MOV reg, mem	Move direct word memory to direct register	4
MOV mem, reg	Move direct word register to direct memory	4
MOVB Rb, Rb	Move direct byte GPR to direct GPR	2
MOVB Rb, #data4	Move immediate byte data to direct GPR	2
MOVB reg, #data8	Move immediate byte data to direct register	4
MOVB Rb, [Rw]	Move indirect byte memory to direct GPR	2
MOVB Rb, [Rw +]	Move indirect byte memory to direct GPR and post-increment source pointer by 1	2
MOVB [Rw], Rb	Move direct byte GPR to indirect memory	2
MOVB [-Rw], Rb	Pre-decrement destination pointer by 1 and move direct byte GPR to indirect memory	2
MOVB [Rw], [Rw]	Move indirect byte memory to indirect memory	2
MOVB [Rw +], [Rw]	Move indirect byte memory to indirect memory and post-increment destination pointer by 1	2
MOVB [Rw], [Rw +]	Move indirect byte memory to indirect memory and post-increment source pointer by 1	2
MOVB Rb, [Rw + #data16]	Move indirect byte memory by base plus constant to direct GPR	4
MOVB [Rw + #data16], Rb	Move direct byte GPR to indirect memory by base plus constant	4
MOVB [Rw], mem	Move direct byte memory to indirect memory	4
MOVB mem, [Rw]	Move indirect byte memory to direct memory	4
MOVB reg, mem	Move direct byte memory to direct register	4
MOVB mem, reg	Move direct byte register to direct memory	4
MOVBS Rb, Rb	Move direct byte GPR with sign extension to direct word GPR	2
MOVBS reg, mem	Move direct byte memory with sign extension to direct word register	4
MOVBS mem, reg	Move direct byte register with sign extension to direct word memory	4

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Data Movement (cont'd)

MOVBZ	Rw, Rb	Move direct byte GPR with zero extension to direct word GPR	2
MOVBZ	reg, mem	Move direct byte memory with zero extension to direct word register	4
MOVBZ	mem, reg	Move direct byte register with zero extension to direct word memory	4

### Jump and Call Operations

JMPA	cc, caddr	Jump absolute if condition is met	4
JMPI	cc, [Rw]	Jump indirect if condition is met	2
JMPR	cc, rel	Jump relative if condition is met	2
JMPS	seg, caddr	Jump absolute to a code segment	4
JB	bitaddr, rel	Jump relative if direct bit is set	4
JBC	bitaddr, rel	Jump relative and clear bit if direct bit is set	4
JNB	bitaddr, rel	Jump relative if direct bit is not set	4
JNBS	bitaddr, rel	Jump relative and set bit if direct bit is not set	4
CALLA	cc, caddr	Call absolute subroutine if condition is met	4
CALLI	cc, [Rw]	Call indirect subroutine if condition is met	2
CALLR	rel	Call relative subroutine	2
CALLS	seg, caddr	Call absolute subroutine in any code segment	4
PCALL	reg, caddr	Push direct word register onto system stack and call absolute subroutine	4
TRAP	#trap7	Call interrupt service routine via immediate trap number	2

### System Stack Operations

POP	reg	Pop direct word register from system stack	2
PUSH	reg	Push direct word register onto system stack	2
SCXT	reg, #data16	Push direct word register onto system stack und update register with immediate data	4
SCXT	reg, mem	Push direct word register onto system stack und update register with direct memory	4

## Instruction Set Summary (cont'd)

Mnemonic	Description	Bytes
----------	-------------	-------

### Return Operations

RET	Return from intra-segment subroutine	2
RETS	Return from inter-segment subroutine	2
RETP     reg	Return from intra-segment subroutine and pop direct word register from system stack	2
RETI	Return from interrupt service subroutine	2

### System Control

SRST	Software Reset	4
IDLE	Enter Idle Mode	4
PWRDN	Enter Power Down Mode (supposes NMI-pin being low)	4
SRVWDT	Service Watchdog Timer	4
DISWDT	Disable Watchdog Timer	4
EINIT	Signify End-of-Initialization on RSTOUT-pin	4

### Miscellaneous

NOP	Null operation	2
-----	----------------	---

## Instruction Set Summary Notes

### Data Addressing Modes

- Rw: – Word GPR (R0, R1, ... , R15)
- Rb: – Byte GPR (RL0, RH0, ..., RL7, RH7)
- reg: – SFR or GPR  
(in case of a byte operation on an SFR, only the low byte can be accessed via 'reg')
- mem: – Direct word or byte memory location
- [...]: – Indirect word or byte memory location  
(Any word GPR can be used as indirect address pointer, except for the arithmetic, logical and compare instructions, where only R0 to R3 are allowed)
- bitaddr: – Direct bit in the bit-addressable memory area
- bitoff: – Direct word in the bit-addressable memory area
- #data: – Immediate constant  
(The number of significant bits which can be specified by the user is represented by the respective appendix 'x')
- #mask8: – Immediate 8-bit mask used for bit-field modifications

### Multiply and Divide Operations

The MDL and MDH registers are implicit source and/or destination operands of the multiply and divide instructions.

### Branch Target Addressing Modes

- caddr: – Direct 16-bit jump target address (Updates the Instruction Pointer)
- seg: – Direct 2-bit segment address  
(Updates the Code Segment Pointer)
- rel: – Signed 8-bit jump target word offset address relative to the Instruction Pointer of the following instruction
- #trap7: – Immediate 7-bit trap or interrupt number.



## Branch Condition Codes

cc:	Symbolically specifiable condition codes
cc_UC	– Unconditional
cc_Z	– Zero
cc_NZ	– Not Zero
cc_V	– Overflow
cc_NV	– No Overflow
cc_N	– Negative
cc_NN	– Not Negative
cc_C	– Carry
cc_NC	– No Carry
cc_EQ	– Equal
cc_NE	– Not Equal
cc_ULT	– Unsigned Less Than
cc_ULE	– Unsigned Less Than or Equal
cc_UGE	– Unsigned Greater Than or Equal
cc_UGT	– Unsigned Greater Than
cc_SLE	– Signed Less Than or Equal
cc_SGE	– Signed Greater Than or Equal
cc_SGT	– Signed Greater Than
cc_NET	– Not Equal and Not End-of-Table

## Instruction Op Codes in Hexadecimal Order

The table on the following pages lists the SAB 80C166W/83C166W's instruction opcodes in a hexadecimal order. This table allows to find the instruction which is associated with a given opcode.

Hex-code	Number of Bytes	Mnemonic	Operands	Hex-code	Number of Bytes	Mnemonic	Operands
00	2	ADD	Rw, Rw	19	2	ADDCB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>
01	2	ADDB	Rb, Rb				
02	4	ADD	reg, mem	1A	4	BFLDH	bitoff, #mask8, #data8
03	4	ADDB	reg, mem				
04	4	ADD	mem, reg	1B	2	MULU	Rw, Rw
05	4	ADDB	mem, reg	1C	2	ROL	Rw, #data4
06	4	ADD	reg, #data16	1D	2	JMPR	cc_NET, rel
07	4	ADDB	reg, #data8	1E	2	BCLR	bitoff.1
08	2	ADD	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>	1F	2	BSET	bitoff.1
09	2	ADDB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>	20	2	SUB	Rw, Rw
0A	4	BFLDL	bitoff, #mask8, #data8	21	2	SUBB	Rb, Rb
0B	2	MUL	Rw, Rw	22	4	SUB	reg, mem
0C	2	ROL	Rw, Rw	23	4	SUBB	reg, mem
0D	2	JMPR	cc_UC, rel	24	4	SUB	mem, reg
0E	2	BCLR	bitoff.0	25	4	SUBB	mem, reg
0F	2	BSET	bitoff.0	26	4	SUB	reg, #data8
10	2	ADDC	Rw, Rw	27	4	SUBB	reg, #data8
11	2	ADDCB	Rb, Rb	28	2	SUB	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>
12	4	ADDC	reg, mem	29	2	SUBB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>
13	4	ADDCB	reg, mem	2A	4	BCMP	bitaddr, bitaddr
14	4	ADDC	mem, reg	2B	2	PRIOR	Rw, Rw
15	4	ADDCB	mem, reg	2C	2	ROR	Rw, Rw
16	4	ADDC	reg, #data16	2D	2	JMPR	cc_EQ, rel or cc_Z, rel
17	4	ADDCB	reg, #data8	2E	2	BCLR	bitoff.2
18	2	ADDC	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>	2F	2	BSET	bitoff.2
				30	2	SUBC	Rw, Rw
				31	2	SUBCB	Rb, Rb
				32	4	SUBC	reg, mem

Hex-code	Number of Bytes	Mnemonic	Operands	Hex-code	Number of Bytes	Mnemonic	Operands
33	4	SUBCB	reg, mem	4C	2	SHL	Rw, Rw
34	4	SUBC	mem, reg	4D	2	JMPR	cc_V, rel
35	4	SUBCB	mem, reg	4E	2	BCLR	bitoff.4
36	4	SUBC	reg, #data16	4F	2	BSET	bitoff.4
37	4	SUBCB	reg, #data8	50	2	XOR	Rw, Rw
38	2	SUBC	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>	51	2	XORB	Rb, Rb
39	2	SUBCB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>	52	4	XOR	reg, mem
3A	4	BMOVN	bitaddr, bitaddr	53	4	XORB	reg, mem
3B	-	-	-	54	4	XOR	mem, reg
3C	2	ROR	Rw, #data4	55	4	XORB	mem, reg
3D	2	JMPR	cc_NE, rel or cc_NZ, rel	56	4	XOR	reg, #data16
3E	2	BCLR	bitoff.3	57	4	XORB	reg, #data8
3F	2	BSET	bitoff.3	58	2	XOR	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>
40	2	CMP	Rw, Rw	59	2	XORB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>
41	2	CMPB	Rb, Rb	5A	4	BOR	bitaddr, bitaddr
42	4	CMP	reg, mem	5B	2	DIVU	Rw
43	4	CMPB	reg, mem	5C	2	SHL	Rw, #data4
44	-	-	-	5D	2	JMPR	cc_NV, rel
45	-	-	-	5E	2	BCLR	bitoff.5
46	4	CMP	reg, #data16	5F	2	BSET	bitoff.5
47	4	CMPB	reg, #data8	60	2	AND	Rw, Rw
48	2	CMP	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>	61	2	ANDB	Rb, Rb
49	2	CMPB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>	62	4	AND	reg, mem
4A	4	BMOV	bitaddr, bitaddr	63	4	ANDB	reg, mem
4B	2	DIV	Rw	64	4	AND	mem, reg
				65	4	ANDB	mem, reg
				66	4	AND	reg, #data16
				67	4	ANDB	reg, #data8

Hex-code	Number of Bytes	Mnemonic	Operands	Hex-code	Number of Bytes	Mnemonic	Operands
68	2	AND	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>	82	4	CMPI1	Rw, mem
				83	-	-	-
69	2	ANDB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>	84	4	MOV	[Rw], mem
				85	-	-	-
6A	4	BAND	bitaddr, bitaddr	86	4	CMPI1	Rw, #data16
6B	2	DIVL	Rw	87	4	IDLE	
6C	2	SHR	Rw, Rw	88	2	MOV	[-Rw], Rw
6D	2	JMPR	cc_N, rel	89	2	MOVB	[-Rw], Rb
6E	2	BCLR	bitoff.6	8A	4	JB	bitaddr, rel
6F	2	BSET	bitoff.6	8B	-	-	-
70	2	OR	Rw, Rw	8C	-	-	-
71	2	ORB	Rb, Rb	8D	2	JMPR	cc_C, rel or cc_ULT, rel
72	4	OR	reg, mem	8E	2	BCLR	bitoff.8
73	4	ORB	reg, mem	8F	2	BSET	bitoff.8
74	4	OR	mem, reg	90	2	CMPI2	Rw, #data4
75	4	ORB	mem, reg	91	2	CPL	Rw
76	4	OR	reg, #data16	92	4	CMPI2	Rw, mem
77	4	ORB	reg, #data8	93	-	-	-
78	2	OR	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 <sup>1)</sup>	94	4	MOV	mem, [Rw]
				95	-	-	-
79	2	ORB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 <sup>1)</sup>	96	4	CMPI2	Rw, #data16
				97	4	PWRDN	
7A	4	BXOR	bitaddr, bitaddr	98	2	MOV	Rw, [Rw+]
7B	2	DIVLU	Rw	99	2	MOVB	Rb, [Rw+]
7C	2	SHR	Rw, #data4	9A	4	JNB	bitaddr, rel
7D	2	JMPR	cc_NN, rel	9B	2	TRAP	#trap7
7E	2	BCLR	bitoff.7	9C	2	JMPI	cc, [Rw]
7F	2	BSET	bitoff.7	9D	2	JMPR	cc_NC, rel or cc_UGE, rel
80	2	CMPI1	Rw, #data4	9E	2	BCLR	bitoff.9
81	2	NEG	Rw	9F	2	BSET	bitoff.9
				A0	2	CPD1	Rw, #data4

Hex-code	Number of Bytes	Mnemonic	Operands	Hex-code	Number of Bytes	Mnemonic	Operands
A1	2	NEGB	Rb	C1	-	-	-
A2	4	CMPD1	Rw, mem	C2	4	MOVBZ	reg, mem
A3	-	-	-	C3	-	-	-
A4	4	MOVB	[Rw], mem	C4	4	MOV	[Rw + #data16], Rw
A5	4	DISWDT		C5	4	MOVBZ	mem, reg
A6	4	CMPD1	Rw, #data16	C6	4	SCXT	reg, #data16
A7	4	SRVWDT		C7	-	-	-
A8	2	MOV	Rw, [Rw]	C8	2	MOV	[Rw], [Rw]
A9	2	MOVB	Rb, [Rw]	C9	2	MOVB	[Rw], [Rw]
AA	4	JBC	bitaddr, rel	CA	4	CALLA	cc, addr
AB	2	CALLI	cc, [Rw]	CB	2	RET	
AC	2	ASHR	Rw, Rw	CC	2	NOP	
AD	2	JMPR	cc_SGT, rel	CD	2	JMPR	cc_SLT, rel
AE	2	BCLR	bitoff.10	CE	2	BCLR	bitoff.12
AF	2	BSET	bitoff.10	CF	2	BSET	bitoff.12
B0	2	CMPD2	Rw, #data4	D0	2	MOVBS	Rw, Rb
B1	2	CPLB	Rb	D1	-	-	-
B2	4	CMPD2	Rw, mem	D2	4	MOVBS	reg, mem
B3	-	-	-	D3	-	-	-
B4	4	MOVB	mem, [Rw]	D4	4	MOV	reg, mem
B5	4	EINIT		D5	4	MOVBS	mem, reg
B6	4	CMPD2	Rw, #data16	D6	4	SCXT	reg, mem
B7	4	SRST		D7	-	-	-
B8	2	MOV	[Rw], Rw	D8	2	MOV	[Rw+], [Rw]
B9	2	MOVB	[Rw], Rb	D9	2	MOVB	[Rw+], [Rw]
BA	4	JNBS	bitaddr, rel	DA	4	CALLS	seg, caddr
BB	2	CALLR	rel	DB	2	RETS	
BC	2	ASHR	Rw, #data4	DC	-	-	-
BD	2	JMPR	cc_SLE, rel	DD	2	JMPR	cc_SGE, rel
BE	2	BCLR	bitoff.11	DE	2	BCLR	bitoff.13
BF	2	BSET	bitoff.11	DF	2	BSET	bitoff.13
C0	2	MOVBZ	Rw, Rb				

Hex-code	Number of Bytes	Mnemonic	Operands	Hex-code	Number of Bytes	Mnemonic	Operands
E0	2	MOV	Rw, #data4	FE	2	BCLR	bitoff.15
E1	2	MOVB	Rb, #data4	FF	2	BSET	bitoff.15
E2	4	PCALL	reg, caddr				
E3	-	-	-				
E4	4	MOVB	[Rw + #data16], Rb				
E5	-	-	-				
E6	4	MOV	reg, #data16				
E7	4	MOVB	reg, #data8				
E8	2	MOV	[Rw], [Rw+]				
E9	2	MOVB	[Rw], [Rw+]				
EA	4	JMPA	cc, caddr				
EB	2	RETP	reg				
EC	2	PUSH	reg				
ED	2	JMPR	cc_UGT, rel				
EE	2	BCLR	bitoff.14				
EF	2	BSET	bitoff.14				
F0	2	MOV	Rw, Rw				
F1	2	MOVB	Rb, Rb				
F2	4	MOV	reg, mem				
F3	4	MOVB	reg, mem				
F4	4	MOVB	Rb, [Rw + #data16]				
F5	-	-	-				
F6	4	MOV	mem, reg				
F7	4	MOVB	mem, reg				
F8	-	-	-				
F9	-	-	-				
FA	4	JMPS	seg, caddr				
FB	2	RETI					
FC	2	POP	reg				
FD	2	JMPR	cc_ULE, rel				

## Notes

- 1) These instructions are encoded by means of additional bits in the operand field of the instruction

x0 <sub>H</sub> – x7 <sub>H</sub> :	Rw, #data3	or	Rb, #data3
x8 <sub>H</sub> – xB <sub>H</sub> :	Rw, [Rw]	or	Rb, [Rw]
xC <sub>H</sub> – xF <sub>H</sub> :	Rw, [Rw +]	or	Rb, [Rw +]

For these instructions only the lowest four GPRs, R0 to R3, can be used as indirect address pointers.

## Notes on the JMPR Instructions

The condition code to be tested for the JMPR instructions is specified by the opcode. Two mnemonic representation alternatives exist for some of the condition codes.

## Notes on the BCLR and BSET Instructions

The position of the bit to be set or to be cleared is specified by the opcode. The operand 'bitoff.n' (n = 0 to 15) refers to a particular bit within a bit-addressable word.

## Notes on the Undefined Opcodes

A hardware trap occurs when one of the undefined opcodes signified by '----' is decoded by the CPU.

## Special Function Registers Overview

The following table lists all SFRs which are implemented in the SAB 80C166W/83C166W 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
<b>ADCIC</b>	b FF98 <sub>H</sub>	CC <sub>H</sub>	A/D Converter End of Conversion Interrupt Control Register	0000 <sub>H</sub>
<b>ADCON</b>	b FFA0 <sub>H</sub>	D0 <sub>H</sub>	A/D Converter Control Register	0000 <sub>H</sub>
<b>ADDAT</b>	FEA0 <sub>H</sub>	50 <sub>H</sub>	A/D Converter Result Register	0000 <sub>H</sub>
<b>ADDRSEL1</b>	FE18 <sub>H</sub>	0C <sub>H</sub>	Address Select Register 1	0000 <sub>H</sub>
<b>ADEIC</b>	b FF9A <sub>H</sub>	CD <sub>H</sub>	A/D Converter Overrun Error Interrupt Control Register	0000 <sub>H</sub>
<b>BUSCON1</b>	b FF14 <sub>H</sub>	8A <sub>H</sub>	Bus Configuration Register 1	0000 <sub>H</sub>
<b>CAPREL</b>	FE4A <sub>H</sub>	25 <sub>H</sub>	GPT2 Capture/Reload Register	0000 <sub>H</sub>
<b>CC0</b>	FE80 <sub>H</sub>	40 <sub>H</sub>	CAPCOM Register 0	0000 <sub>H</sub>
<b>CC0IC</b>	b FF78 <sub>H</sub>	BC <sub>H</sub>	CAPCOM Register 0 Interrupt Control Register	0000 <sub>H</sub>
<b>CC1</b>	FE82 <sub>H</sub>	41 <sub>H</sub>	CAPCOM Register 1	0000 <sub>H</sub>
<b>CC1IC</b>	b FF7A <sub>H</sub>	BD <sub>H</sub>	CAPCOM Register 1 Interrupt Control Register	0000 <sub>H</sub>
<b>CC2</b>	FE84 <sub>H</sub>	42 <sub>H</sub>	CAPCOM Register 2	0000 <sub>H</sub>
<b>CC2IC</b>	b FF7C <sub>H</sub>	BE <sub>H</sub>	CAPCOM Register 2 Interrupt Control Register	0000 <sub>H</sub>
<b>CC3</b>	FE86 <sub>H</sub>	43 <sub>H</sub>	CAPCOM Register 3	0000 <sub>H</sub>
<b>CC3IC</b>	b FF7E <sub>H</sub>	BF <sub>H</sub>	CAPCOM Register 3 Interrupt Control Register	0000 <sub>H</sub>
<b>CC4</b>	FE88 <sub>H</sub>	44 <sub>H</sub>	CAPCOM Register 4	0000 <sub>H</sub>
<b>CC4IC</b>	b FF80 <sub>H</sub>	C0 <sub>H</sub>	CAPCOM Register 4 Interrupt Control Register	0000 <sub>H</sub>
<b>CC5</b>	FE8A <sub>H</sub>	45 <sub>H</sub>	CAPCOM Register 5	0000 <sub>H</sub>
<b>CC5IC</b>	b FF82 <sub>H</sub>	C1 <sub>H</sub>	CAPCOM Register 5 Interrupt Control Register	0000 <sub>H</sub>
<b>CC6</b>	FE8C <sub>H</sub>	46 <sub>H</sub>	CAPCOM Register 6	0000 <sub>H</sub>
<b>CC6IC</b>	b FF84 <sub>H</sub>	C2 <sub>H</sub>	CAPCOM Register 6 Interrupt Control Register	0000 <sub>H</sub>
<b>CC7</b>	FE8E <sub>H</sub>	47 <sub>H</sub>	CAPCOM Register 7	0000 <sub>H</sub>



## Special Function Registers Overview (cont'd)

Name	Physical Address	8-Bit Address	Description	Reset Value
<b>CC7IC</b>	<b>b</b> FF86 <sub>H</sub>	C3 <sub>H</sub>	CAPCOM Register 7 Interrupt Control Register	0000 <sub>H</sub>
<b>CC8</b>	FE90 <sub>H</sub>	48 <sub>H</sub>	CAPCOM Register 8	0000 <sub>H</sub>
<b>CC8IC</b>	<b>b</b> FF88 <sub>H</sub>	C4 <sub>H</sub>	CAPCOM Register 8 Interrupt Control Register	0000 <sub>H</sub>
<b>CC9</b>	FE92 <sub>H</sub>	49 <sub>H</sub>	CAPCOM Register 9	0000 <sub>H</sub>
<b>CC9IC</b>	<b>b</b> FF8A <sub>H</sub>	C5 <sub>H</sub>	CAPCOM Register 9 Interrupt Control Register	0000 <sub>H</sub>
<b>CC10</b>	FE94 <sub>H</sub>	4A <sub>H</sub>	CAPCOM Register 10	0000 <sub>H</sub>
<b>CC10IC</b>	<b>b</b> FF8C <sub>H</sub>	C6 <sub>H</sub>	CAPCOM Register 10 Interrupt Control Register	0000 <sub>H</sub>
<b>CC11</b>	FE96 <sub>H</sub>	4B <sub>H</sub>	CAPCOM Register 11	0000 <sub>H</sub>
<b>CC11IC</b>	<b>b</b> FF8E <sub>H</sub>	C7 <sub>H</sub>	CAPCOM Register 11 Interrupt Control Register	0000 <sub>H</sub>
<b>CC12</b>	FE98 <sub>H</sub>	4C <sub>H</sub>	CAPCOM Register 12	0000 <sub>H</sub>
<b>CC12IC</b>	<b>b</b> FF90 <sub>H</sub>	C8 <sub>H</sub>	CAPCOM Register 12 Interrupt Control Register	0000 <sub>H</sub>
<b>CC13</b>	FE9A <sub>H</sub>	4D <sub>H</sub>	CAPCOM Register 13	0000 <sub>H</sub>
<b>CC13IC</b>	<b>b</b> FF92 <sub>H</sub>	C9 <sub>H</sub>	CAPCOM Register 13 Interrupt Control Register	0000 <sub>H</sub>
<b>CC14</b>	FE9C <sub>H</sub>	4E <sub>H</sub>	CAPCOM Register 14	0000 <sub>H</sub>
<b>CC14IC</b>	<b>b</b> FF94 <sub>H</sub>	CA <sub>H</sub>	CAPCOM Register 14 Interrupt Control Register	0000 <sub>H</sub>
<b>CC15</b>	FE9E <sub>H</sub>	4F <sub>H</sub>	CAPCOM Register 15	0000 <sub>H</sub>
<b>CC15IC</b>	<b>b</b> FF96 <sub>H</sub>	CB <sub>H</sub>	CAPCOM Register 15 Interrupt Control Register	0000 <sub>H</sub>
<b>CCM0</b>	<b>b</b> FF52 <sub>H</sub>	A9 <sub>H</sub>	CAPCOM Mode Control Register 0	0000 <sub>H</sub>
<b>CCM1</b>	<b>b</b> FF54 <sub>H</sub>	AA <sub>H</sub>	CAPCOM Mode Control Register 1	0000 <sub>H</sub>
<b>CCM2</b>	<b>b</b> FF56 <sub>H</sub>	AB <sub>H</sub>	CAPCOM Mode Control Register 2	0000 <sub>H</sub>
<b>CCM3</b>	<b>b</b> FF58 <sub>H</sub>	AC <sub>H</sub>	CAPCOM Mode Control Register 3	0000 <sub>H</sub>
<b>CP</b>	FE10 <sub>H</sub>	08 <sub>H</sub>	CPU Context Pointer Register	FC00 <sub>H</sub>
<b>CRIC</b>	<b>b</b> FF6A <sub>H</sub>	B5 <sub>H</sub>	GPT2 CAPREL Interrupt Control Register	0000 <sub>H</sub>
<b>CSP</b>	FE08 <sub>H</sub>	04 <sub>H</sub>	CPU Code Segment Pointer Register (2 bits, read only)	0000 <sub>H</sub>
<b>DP0</b>	<b>b</b> FF02 <sub>H</sub>	81 <sub>H</sub>	Port 0 Direction Control Register	0000 <sub>H</sub>
<b>DP1</b>	<b>b</b> FF06 <sub>H</sub>	83 <sub>H</sub>	Port 1 Direction Control Register	0000 <sub>H</sub>
<b>DP2</b>	<b>b</b> FFC2 <sub>H</sub>	E1 <sub>H</sub>	Port 2 Direction Control Register	0000 <sub>H</sub>
<b>DP3</b>	<b>b</b> FFC6 <sub>H</sub>	E3 <sub>H</sub>	Port 3 Direction Control Register	0000 <sub>H</sub>
<b>DP4</b>	<b>b</b> FF0A <sub>H</sub>	85 <sub>H</sub>	Port 4 Direction Control Register (2 bits)	00 <sub>H</sub>

## Special Function Registers Overview (cont'd)

Name	Physical Address	8-Bit Address	Description	Reset Value
<b>DPP0</b>	FE00 <sub>H</sub>	00 <sub>H</sub>	CPU Data Page Pointer 0 Register (4 bits)	0000 <sub>H</sub>
<b>DPP1</b>	FE02 <sub>H</sub>	01 <sub>H</sub>	CPU Data Page Pointer 1 Register (4 bits)	0001 <sub>H</sub>
<b>DPP2</b>	FE04 <sub>H</sub>	02 <sub>H</sub>	CPU Data Page Pointer 2 Register (4 bits)	0002 <sub>H</sub>
<b>DPP3</b>	FE06 <sub>H</sub>	03 <sub>H</sub>	CPU Data Page Pointer 3 Register (4 bits)	0003 <sub>H</sub>
<b>MDC</b> <b>b</b>	FF0E <sub>H</sub>	87 <sub>H</sub>	CPU Multiply / Divide Control Register	0000 <sub>H</sub>
<b>MDH</b>	FE0C <sub>H</sub>	06 <sub>H</sub>	CPU Multiply / Divide Register – High Word	0000 <sub>H</sub>
<b>MDL</b>	FE0E <sub>H</sub>	07 <sub>H</sub>	CPU Multiply / Divide Register – Low Word	0000 <sub>H</sub>
<b>ONES</b>	FF1E <sub>H</sub>	8F <sub>H</sub>	Constant Value 1's Register (read only)	FFFF <sub>H</sub>
<b>P0</b> <b>b</b>	FF00 <sub>H</sub>	80 <sub>H</sub>	Port 0 Register	0000 <sub>H</sub>
<b>P1</b> <b>b</b>	FF04 <sub>H</sub>	82 <sub>H</sub>	Port 1 Register	0000 <sub>H</sub>
<b>P2</b> <b>b</b>	FFC0 <sub>H</sub>	E0 <sub>H</sub>	Port 2 Register	0000 <sub>H</sub>
<b>P3</b> <b>b</b>	FFC4 <sub>H</sub>	E2 <sub>H</sub>	Port 3 Register	0000 <sub>H</sub>
<b>P4</b> <b>b</b>	FFC8 <sub>H</sub>	E4 <sub>H</sub>	Port 4 Register (2 bits)	00 <sub>H</sub>
<b>P5</b> <b>b</b>	FFA2 <sub>H</sub>	D1 <sub>H</sub>	Port 5 Register (read only)	XXXX <sub>H</sub>
<b>PECC0</b>	FEC0 <sub>H</sub>	60 <sub>H</sub>	PEC Channel 0 Control Register	0000 <sub>H</sub>
<b>PECC1</b>	FEC2 <sub>H</sub>	61 <sub>H</sub>	PEC Channel 1 Control Register	0000 <sub>H</sub>
<b>PECC2</b>	FEC4 <sub>H</sub>	62 <sub>H</sub>	PEC Channel 2 Control Register	0000 <sub>H</sub>
<b>PECC3</b>	FEC6 <sub>H</sub>	63 <sub>H</sub>	PEC Channel 3 Control Register	0000 <sub>H</sub>
<b>PECC4</b>	FEC8 <sub>H</sub>	64 <sub>H</sub>	PEC Channel 4 Control Register	0000 <sub>H</sub>
<b>PECC5</b>	FECA <sub>H</sub>	65 <sub>H</sub>	PEC Channel 5 Control Register	0000 <sub>H</sub>
<b>PECC6</b>	FEC <sub>H</sub>	66 <sub>H</sub>	PEC Channel 6 Control Register	0000 <sub>H</sub>
<b>PECC7</b>	FECE <sub>H</sub>	67 <sub>H</sub>	PEC Channel 7 Control Register	0000 <sub>H</sub>
<b>PSW</b> <b>b</b>	FF10 <sub>H</sub>	88 <sub>H</sub>	CPU Program Status Word	0000 <sub>H</sub>
<b>S0BG</b>	FEB4 <sub>H</sub>	5A <sub>H</sub>	Serial Channel 0 Baud Rate Generator Reload Register	0000 <sub>H</sub>
<b>S0CON</b> <b>b</b>	FFB0 <sub>H</sub>	D8 <sub>H</sub>	Serial Channel 0 Control Register	0000 <sub>H</sub>
<b>S0EIC</b> <b>b</b>	FF70 <sub>H</sub>	B8 <sub>H</sub>	Serial Channel 0 Error Interrupt Control Register	0000 <sub>H</sub>
<b>S0RBUF</b>	FEB2 <sub>H</sub>	59 <sub>H</sub>	Serial Channel 0 Receive Buffer Register (read only)	XX <sub>H</sub>

## Special Function Registers Overview (cont'd)

Name		Physical Address	8-Bit Address	Description	Reset Value
<b>S0RIC</b>	<b>b</b>	FF6E <sub>H</sub>	B7 <sub>H</sub>	Serial Channel 0 Receive Interrupt Control Register	0000 <sub>H</sub>
<b>S0TBUF</b>		FEB0 <sub>H</sub>	58 <sub>H</sub>	Serial Channel 0 Transmit Buffer Register (write only)	00 <sub>H</sub>
<b>S0TIC</b>	<b>b</b>	FF6C <sub>H</sub>	B6 <sub>H</sub>	Serial Channel 0 Transmit Interrupt Control Register	0000 <sub>H</sub>
<b>S1BG</b>		FEBC <sub>H</sub>	5E <sub>H</sub>	Serial Channel 1 Baud Rate Generator Reload Register	0000 <sub>H</sub>
<b>S1CON</b>	<b>b</b>	FFB8 <sub>H</sub>	DC <sub>H</sub>	Serial Channel 1 Control Register	0000 <sub>H</sub>
<b>S1EIC</b>	<b>b</b>	FF76 <sub>H</sub>	BB <sub>H</sub>	Serial Channel 1 Error Interrupt Control Register	0000 <sub>H</sub>
<b>S1RBUF</b>		FEBA <sub>H</sub>	5D <sub>H</sub>	Serial Channel 1 Receive Buffer Register (read only)	XX <sub>H</sub>
<b>S1RIC</b>	<b>b</b>	FF74 <sub>H</sub>	BA <sub>H</sub>	Serial Channel 1 Receive Interrupt Control Register	0000 <sub>H</sub>
<b>S1TBUF</b>		FEB8 <sub>H</sub>	5C <sub>H</sub>	Serial Channel 1 Transmit Buffer Register (write only)	00 <sub>H</sub>
<b>S1TIC</b>	<b>b</b>	FF72 <sub>H</sub>	B9 <sub>H</sub>	Serial Channel 1 Transmit Interrupt Control Register	0000 <sub>H</sub>
<b>SP</b>		FE12 <sub>H</sub>	09 <sub>H</sub>	CPU System Stack Pointer Register	FC00 <sub>H</sub>
<b>STKOV</b>		FE14 <sub>H</sub>	0A <sub>H</sub>	CPU Stack Overflow Pointer Register	FA00 <sub>H</sub>
<b>STKUN</b>		FE16 <sub>H</sub>	0B <sub>H</sub>	CPU Stack Underflow Pointer Register	FC00 <sub>H</sub>
<b>SYSCON</b>	<b>b</b>	FF0C <sub>H</sub>	86 <sub>H</sub>	CPU System Configuration Register	0xx0 <sub>H</sub> *)
<b>T0</b>		FE50 <sub>H</sub>	28 <sub>H</sub>	CAPCOM Timer 0 Register	0000 <sub>H</sub>
<b>T01CON</b>	<b>b</b>	FF50 <sub>H</sub>	A8 <sub>H</sub>	CAPCOM Timer 0 and Timer 1 Control Register	0000 <sub>H</sub>
<b>T0IC</b>	<b>b</b>	FF9C <sub>H</sub>	CE <sub>H</sub>	CAPCOM Timer 0 Interrupt Control Register	0000 <sub>H</sub>
<b>T0REL</b>		FE54 <sub>H</sub>	2A <sub>H</sub>	CAPCOM Timer 0 Reload Register	0000 <sub>H</sub>
<b>T1</b>		FE52 <sub>H</sub>	29 <sub>H</sub>	CAPCOM Timer 1 Register	0000 <sub>H</sub>
<b>T1IC</b>	<b>b</b>	FF9E <sub>H</sub>	CF <sub>H</sub>	CAPCOM Timer 1 Interrupt Control Register	0000 <sub>H</sub>
<b>T1REL</b>		FE56 <sub>H</sub>	2B <sub>H</sub>	CAPCOM Timer 1 Reload Register	0000 <sub>H</sub>
<b>T2</b>		FE40 <sub>H</sub>	20 <sub>H</sub>	GPT1 Timer 2 Register	0000 <sub>H</sub>
<b>T2CON</b>	<b>b</b>	FF40 <sub>H</sub>	A0 <sub>H</sub>	GPT1 Timer 2 Control Register	0000 <sub>H</sub>
<b>T2IC</b>	<b>b</b>	FF60 <sub>H</sub>	B0 <sub>H</sub>	GPT1 Timer 2 Interrupt Control Register	0000 <sub>H</sub>

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

Name	Physical Address	8-Bit Address	Description	Reset Value
<b>T3</b>	FE42 <sub>H</sub>	21 <sub>H</sub>	GPT1 Timer 3 Register	0000 <sub>H</sub>
<b>T3CON</b>	<b>b</b> FF42 <sub>H</sub>	A1 <sub>H</sub>	GPT1 Timer 3 Control Register	0000 <sub>H</sub>
<b>T3IC</b>	<b>b</b> FF62 <sub>H</sub>	B1 <sub>H</sub>	GPT1 Timer 3 Interrupt Control Register	0000 <sub>H</sub>
<b>T4</b>	FE44 <sub>H</sub>	22 <sub>H</sub>	GPT1 Timer 4 Register	0000 <sub>H</sub>
<b>T4CON</b>	<b>b</b> FF44 <sub>H</sub>	A2 <sub>H</sub>	GPT1 Timer 4 Control Register	0000 <sub>H</sub>
<b>T4IC</b>	<b>b</b> FF64 <sub>H</sub>	B2 <sub>H</sub>	GPT1 Timer 4 Interrupt Control Register	0000 <sub>H</sub>
<b>T5</b>	FE46 <sub>H</sub>	23 <sub>H</sub>	GPT2 Timer 5 Register	0000 <sub>H</sub>
<b>T5CON</b>	<b>b</b> FF46 <sub>H</sub>	A3 <sub>H</sub>	GPT2 Timer 5 Control Register	0000 <sub>H</sub>
<b>T5IC</b>	<b>b</b> FF66 <sub>H</sub>	B3 <sub>H</sub>	GPT2 Timer 5 Interrupt Control Register	0000 <sub>H</sub>
<b>T6</b>	FE48 <sub>H</sub>	24 <sub>H</sub>	GPT2 Timer 6 Register	0000 <sub>H</sub>
<b>T6CON</b>	<b>b</b> FF48 <sub>H</sub>	A4 <sub>H</sub>	GPT2 Timer 6 Control Register	0000 <sub>H</sub>
<b>T6IC</b>	<b>b</b> FF68 <sub>H</sub>	B4 <sub>H</sub>	GPT2 Timer 6 Interrupt Control Register	0000 <sub>H</sub>
<b>TFR</b>	<b>b</b> FFAC <sub>H</sub>	D6 <sub>H</sub>	Trap Flag Register	0000 <sub>H</sub>
<b>WDT</b>	FEAE <sub>H</sub>	57 <sub>H</sub>	Watchdog Timer Register (read only)	0000 <sub>H</sub>
<b>WDTCON</b>	FFAE <sub>H</sub>	D7 <sub>H</sub>	Watchdog Timer Control Register	0000 <sub>H</sub>
<b>ZEROS</b>	<b>b</b> FF1C <sub>H</sub>	8E <sub>H</sub>	Constant Value 0's Register (read only)	0000 <sub>H</sub>

\*) The system configuration is selected during reset.

## Absolute Maximum Ratings

Ambient temperature under bias ( $T_A$ ):

SAB 83C166W-5M, SAB 80C166W-M ..... 0 to + 70 °C

SAB 83C166W-5M-T3, SAB 80C166W-M-T3 ..... – 40 to + 85 °C

SAB 83C166W-5M-T4, SAB 80C166W-M-T4 ..... – 40 to + 110 °C

Storage temperature ( $T_{stg}$ ) ..... – 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 80C166W/83C166W 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 (Controller Characteristics):

The logic of the SAB 80C166W/83C166W will provide signals with the respective timing characteristics.

### SR (System Requirement):

The external system must provide signals with the respective timing characteristics to the SAB 80C166W/83C166W.

## DC Characteristics

$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$

$T_A = 0\text{ to } +70\text{ }^\circ\text{C}$  for SAB 83C166W-5M, SAB 80C166W/83C166W-M

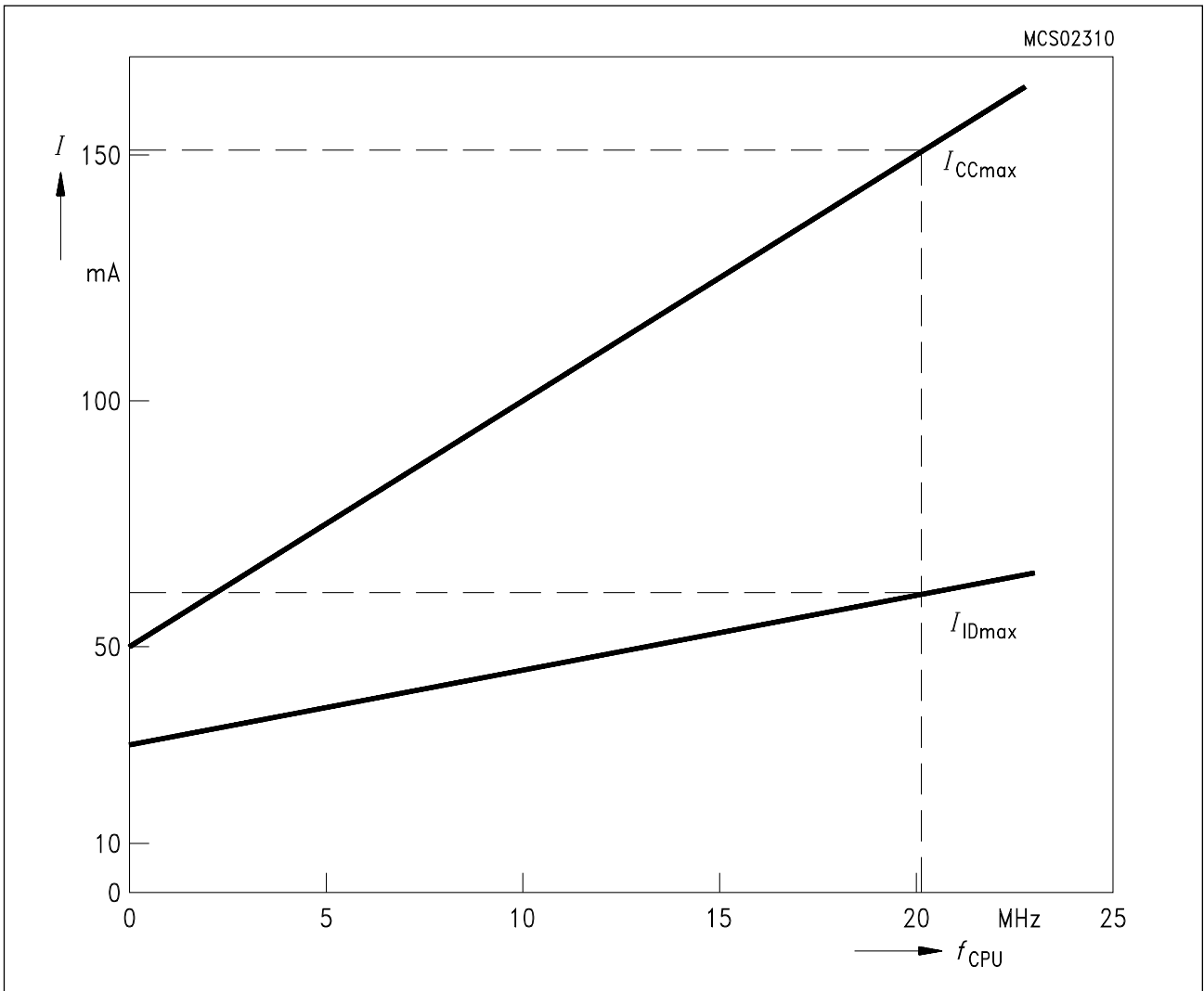
$T_A = -40\text{ to } +85\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3

$T_A = -40\text{ to } +110\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4

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, $\overline{\text{ALE}}$ , $\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{BHE}}$ , CLKOUT, $\overline{\text{RSTOUT}}$ )	$V_{OL}$ CC	-	0.45	V	$I_{OL} = 2.4\text{ mA}$
Output low voltage (all other outputs)	$V_{OL1}$ CC	-	0.45	V	$I_{OL1} = 1.6\text{ mA}$
Output high voltage (Port 0, Port 1, Port 4, $\overline{\text{ALE}}$ , $\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{BHE}}$ , CLKOUT, $\overline{\text{RSTOUT}}$ )	$V_{OH}$ CC	$0.9 V_{CC}$ 2.4	-	V	$I_{OH} = -500\text{ }\mu\text{A}$ $I_{OH} = -2.4\text{ mA}$
Output high voltage (all other outputs)	$V_{OH1}$ CC	$0.9 V_{CC}$ 2.4	-	V V	$I_{OH} = -250\text{ }\mu\text{A}$ $I_{OH} = -1.6\text{ mA}$
Input leakage current <sup>1)</sup>	$I_{OZ}$ CC	-	$\pm 1$	$\mu\text{A}$	$0\text{ V} < V_{IN} < V_{CC}$
$\overline{\text{RSTIN}}$ pullup resistor	$R_{RST}$ CC	50	150	k $\Omega$	-
Read inactive current <sup>5)</sup>	$I_{RH}$ <sup>3)</sup>	-	- 40	$\mu\text{A}$	$V_{OUT} = V_{OHmin}$
Read active current <sup>5)</sup>	$I_{RL}$ <sup>4)</sup>	- 500	-	$\mu\text{A}$	$V_{OUT} = V_{OLmax}$
ALE inactive current <sup>5)</sup>	$I_{ALEL}$ <sup>3)</sup>	-	150	$\mu\text{A}$	$V_{OUT} = V_{OLmax}$
ALE active current <sup>5)</sup>	$I_{ALEH}$ <sup>4)</sup>	2100	-	$\mu\text{A}$	$V_{OUT} = V_{OHmin}$
XTAL1 input current	$I_{IL}$ CC	-	$\pm 20$	$\mu\text{A}$	$0\text{ V} < V_{IN} < V_{CC}$
Pin capacitance <sup>6)</sup> (digital inputs/outputs)	$C_{IO}$ CC	-	10	pF	$f = 1\text{ MHz}$ $T_A = 25\text{ }^\circ\text{C}$
Power supply current	$I_{CC}$	-	50 + $5 \times f_{CPU}$	mA	Reset active $f_{CPU}$ in [MHz] <sup>7)</sup>
Idle mode supply current	$I_{ID}$	-	30 + $1.5 \times f_{CPU}$	mA	$f_{CPU}$ in [MHz] <sup>7)</sup>
Power-down mode supply current	$I_{PD}$	-	50	$\mu\text{A}$	$V_{CC} = 5.5\text{ V}$ <sup>8)</sup>

### Notes

- 1) This specification does not apply to the analog input (Port 5.x) which is currently converted.
- 3) The maximum current may be drawn while the respective signal line remains inactive.
- 4) The minimum current must be drawn in order to drive the respective signal line active.
- 5) This specification is only valid during Reset, or during Hold-mode.
- 6) Not 100% tested, guaranteed by design characterization.
- 7) The supply current is a function of the operating frequency. This dependency is illustrated in the figure below. These parameters are tested at 20 MHz CPU clock with all outputs open.
- 8) 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. 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

$$V_{CC} = 5 \text{ V} \pm 10 \% ; \quad V_{SS} = 0 \text{ V}$$

$$T_A = 0 \text{ to } +70 \text{ }^\circ\text{C} \quad \text{for SAB 83C166W-5M, SAB 80C166W/83C166W-M}$$

$$T_A = -40 \text{ to } +85 \text{ }^\circ\text{C} \quad \text{for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3}$$

$$T_A = -40 \text{ to } +110 \text{ }^\circ\text{C} \quad \text{for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4}$$

$$4.0 \text{ V} \leq V_{AREF} \leq V_{CC} + 0.1 \text{ V}; \quad V_{SS} - 0.1 \text{ V} \leq V_{AGND} \leq V_{SS} + 0.2 \text{ V}$$

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) 5)
Conversion time	$t_C$ CC	–	$10 t_{CC} + t_S + 4TCL$		3) 5)
Total unadjusted error	$TUE$ CC	–	$\pm 2$	LSB	$V_{AREF} = V_{CC}$ $V_{AGND} = V_{SS}$
Internal resistance of reference voltage source	$R_{AREF}$ CC	–	$t_{CC} / 250 - 0.25$	k $\Omega$	$t_{CC}$ in [ns] <sup>4)</sup>
Internal resistance of analog source	$R_{ASRC}$ CC	–	$t_S / 500 - 0.25$	k $\Omega$	$t_S$ in [ns] <sup>4)</sup>

## 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<sub>H</sub> or X3FF<sub>H</sub>, 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  $t_{SC} = T_{CL} \times 32$ .
- 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  $t_{CC} = T_{CL} \times 32$ .
- 4) During the conversion the ADC's capacitance must be repeatedly charged or discharged. The internal resistance of the respective current source must allow the capacitors to reach their final voltage level within the indicated time. The maximum internal resistance results from the CPU clock period.
- 5) This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.



Testing Waveforms

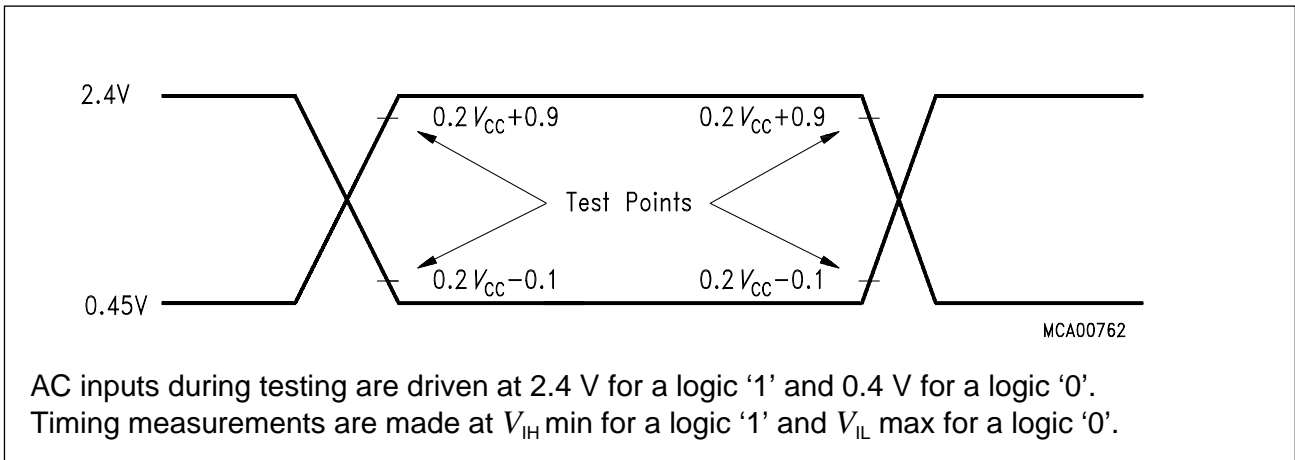


Figure 9  
Input Output Waveforms

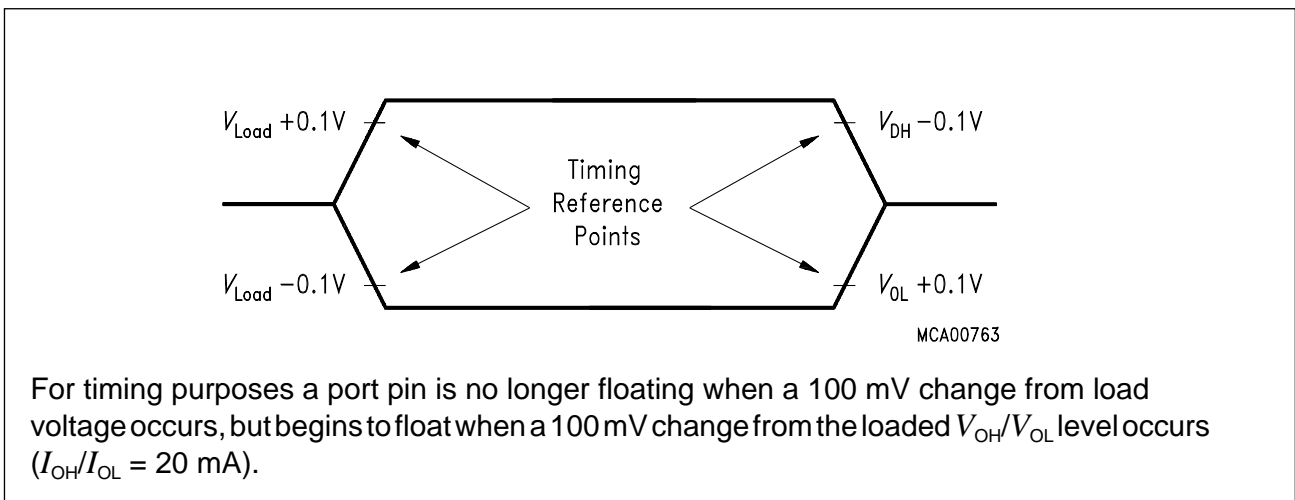


Figure 10  
Float Waveforms

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	$t_A$	$T_{CL} \times \langle ALECTL \rangle$
Memory Cycle Time Waitstates	$t_C$	$2T_{CL} \times (15 - \langle MCTC \rangle)$
Memory Tristate Time	$t_F$	$2T_{CL} \times (1 - \langle MTTC \rangle)$

## AC Characteristics

### External Clock Drive XTAL1

$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$

$T_A = 0\text{ to } +70\text{ }^\circ\text{C}$  for SAB 83C166W-5M, SAB 80C166W/83C166W-M

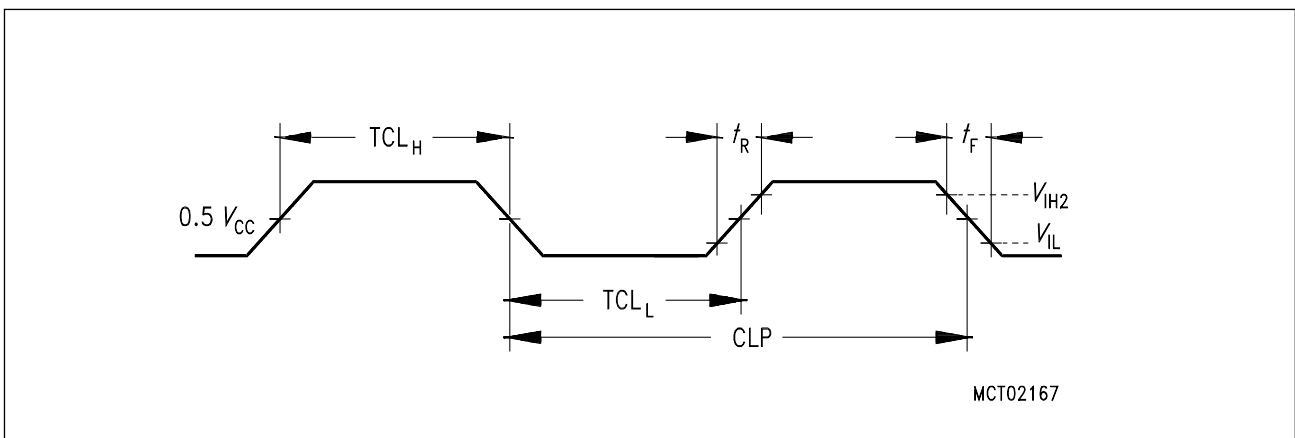
$T_A = -40\text{ to } +85\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3

$T_A = -40\text{ to } +110\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock $1/T_{CLP} = 1\text{ to }20\text{ MHz}$		Unit
		min.	max.	min.	max.	
Oscillator period	$T_{CLP}$ SR	62.5	62.5	50	1000	ns
High time	$t_{CLH}$ SR	25	–	25	$T_{CLP} - t_{CLL}$	ns
Low time	$t_{CLL}$ SR	25	–	25	$T_{CLP} - t_{CLH}$	ns
Rise time	$t_R$ SR	–	10	–	10	ns
Fall time	$t_F$ SR	–	10	–	10	ns
Oscillator duty cycle	DC SR	0.4	0.6	$25 / T_{CLP}$	$1 - 25 / T_{CLP}$	
Clock cycle	$T_{CL}$	25	37.5	$T_{CLP} \times DC_{min}$	$T_{CLP} \times DC_{max}$	ns

**Note:** In order to run the SAB 80C166W/83C166W 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 11**  
External Clock Drive XTAL1

## AC Characteristics (cont'd)

### Multiplexed Bus

$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$

$T_A = 0\text{ to }+70\text{ }^\circ\text{C}$  for SAB 83C166W-5M, SAB 80C166W/83C166W-M

$T_A = -40\text{ to }+85\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3

$T_A = -40\text{ to }+110\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4

$C_L$  (for Port 0, Port 1, Port 4, ALE,  $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ ,  $\overline{\text{BHE}}$ , CLKOUT) = 100 pF

ALE cycle time =  $6 T_{CL}$  (150 ns at 20-MHz CPU clock)

Parameter	Symbol		CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock $1/T_{CLP} = 1\text{ to }20\text{ MHz}$		Unit
			min.	max.	min.	max.	
ALE high time	$t_5$	CC	$15 + t_A$	–	$T_{CLmin} - 10 + t_A$	–	ns
Address setup to ALE	$t_6$	CC	$10 + t_A$	–	$T_{CLmin} - 15 + t_A$	–	ns
Address hold after ALE	$t_7$	CC	$15 + t_A$	–	$T_{CLmin} - 10 + t_A$	–	ns
ALE falling edge to $\overline{\text{RD}}$ , $\overline{\text{WR}}$ (with RW-delay)	$t_8$	CC	$15 + t_A$	–	$T_{CLmin} - 10 + t_A$	–	ns
ALE falling edge to $\overline{\text{RD}}$ , $\overline{\text{WR}}$ (no RW-delay)	$t_9$	CC	$-10 + t_A$	–	$-10 + t_A$	–	ns
Address float after $\overline{\text{RD}}$ , $\overline{\text{WR}}$ (with RW-delay)	$t_{10}$	CC	–	5	–	5	ns
Address float after $\overline{\text{RD}}$ , $\overline{\text{WR}}$ (no RW-delay)	$t_{11}$	CC	–	42.5	–	$T_{CLmax} + 5$	ns
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ low time (with RW-delay)	$t_{12}$	CC	$52.5 + t_C$	–	$T_{CLP} - 10 + t_C$	–	ns
$\overline{\text{RD}}$ $\overline{\text{WR}}$ low time (no RW-delay)	$t_{13}$	CC	$77.5 + t_C$	–	$T_{CLP} + T_{CLmin} - 10 + t_C$	–	ns
$\overline{\text{RD}}$ to valid data in (with RW-delay)	$t_{14}$	SR	–	$47.5 + t_C$	–	$T_{CLP} - 20 + t_C$	ns
$\overline{\text{RD}}$ to valid data in (no RW-delay)	$t_{15}$	SR	–	$72.5 + t_C$	–	$T_{CLP} + T_{CLmin} - 20 + t_C$	ns
ALE low to valid data in	$t_{16}$	SR	–	$72.5 + t_A + t_C$	–	$T_{CLP} + T_{CLmin} - 20 + t_C$	ns
Address to valid data in	$t_{17}$	SR	–	$100 + 2t_A + t_C$	–	$2T_{CLP} - 25 + 2t_A + t_C$	ns
Data hold after $\overline{\text{RD}}$ rising edge	$t_{18}$	SR	0	–	0	–	ns

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock $1/T_{CLP} = 1$ to 20 MHz		Unit
		min.	max.	min.	max.	
Data float after $\overline{RD}$	$t_{19}$ SR	–	$47.5 + t_F$	–	$T_{CLP} - 15 + t_F$	ns
Data valid to $\overline{WR}$	$t_{22}$ CC	$47.5 + t_C$	–	$T_{CLP} - 15 + t_C$	–	ns
Data hold after $\overline{WR}$	$t_{23}$ CC	$47.5 + t_F$	–	$T_{CLP} - 15 + t_F$	–	ns
ALE rising edge after $\overline{RD}$ , $\overline{WR}$	$t_{25}$ CC	$47.5 + t_F$	–	$T_{CLP} - 15 + t_F$	–	ns
Address hold after $\overline{RD}$ , $\overline{WR}$	$t_{27}$ CC	$47.5 + t_F$	–	$T_{CLP} - 15 + t_F$	–	ns

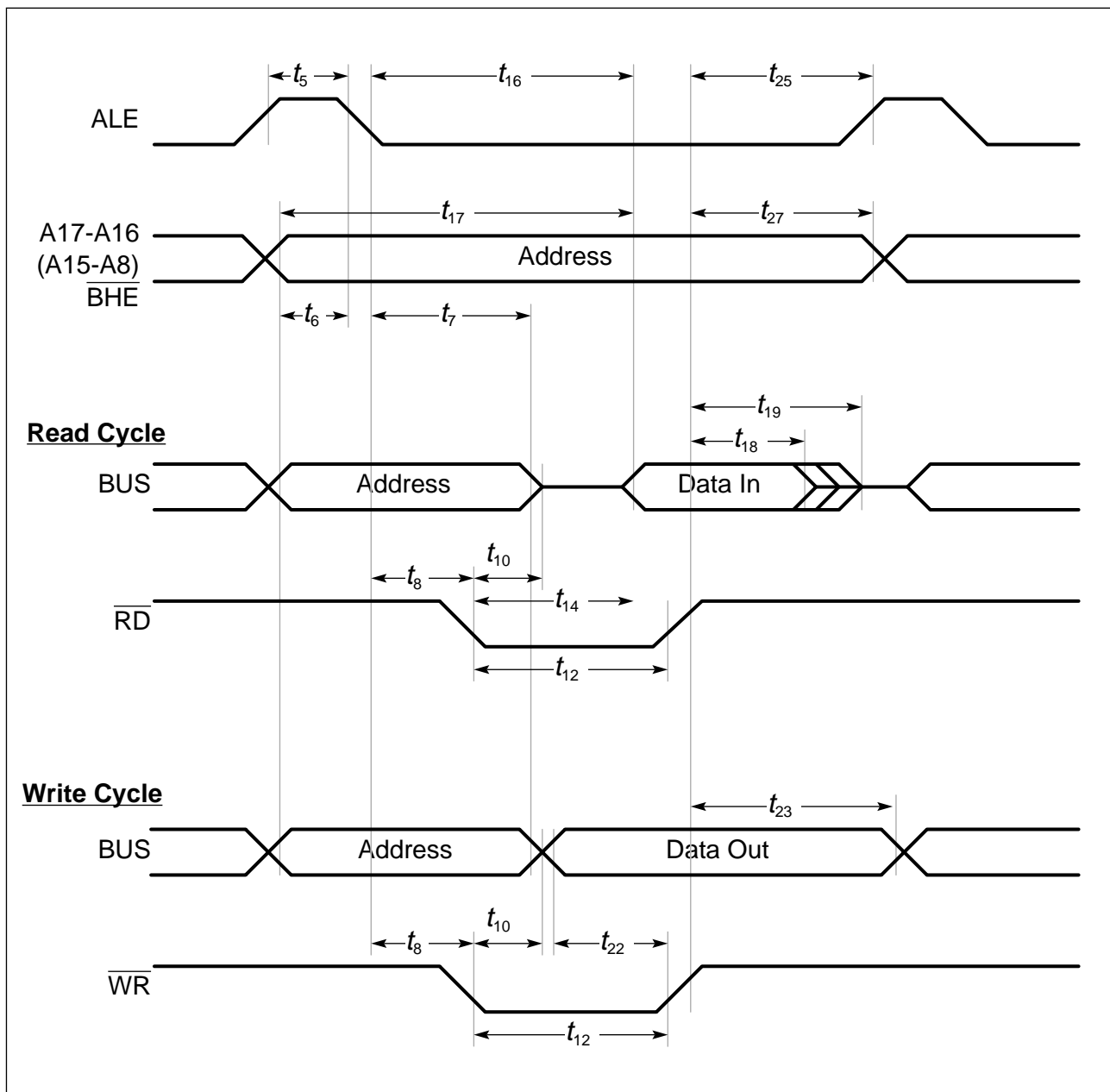
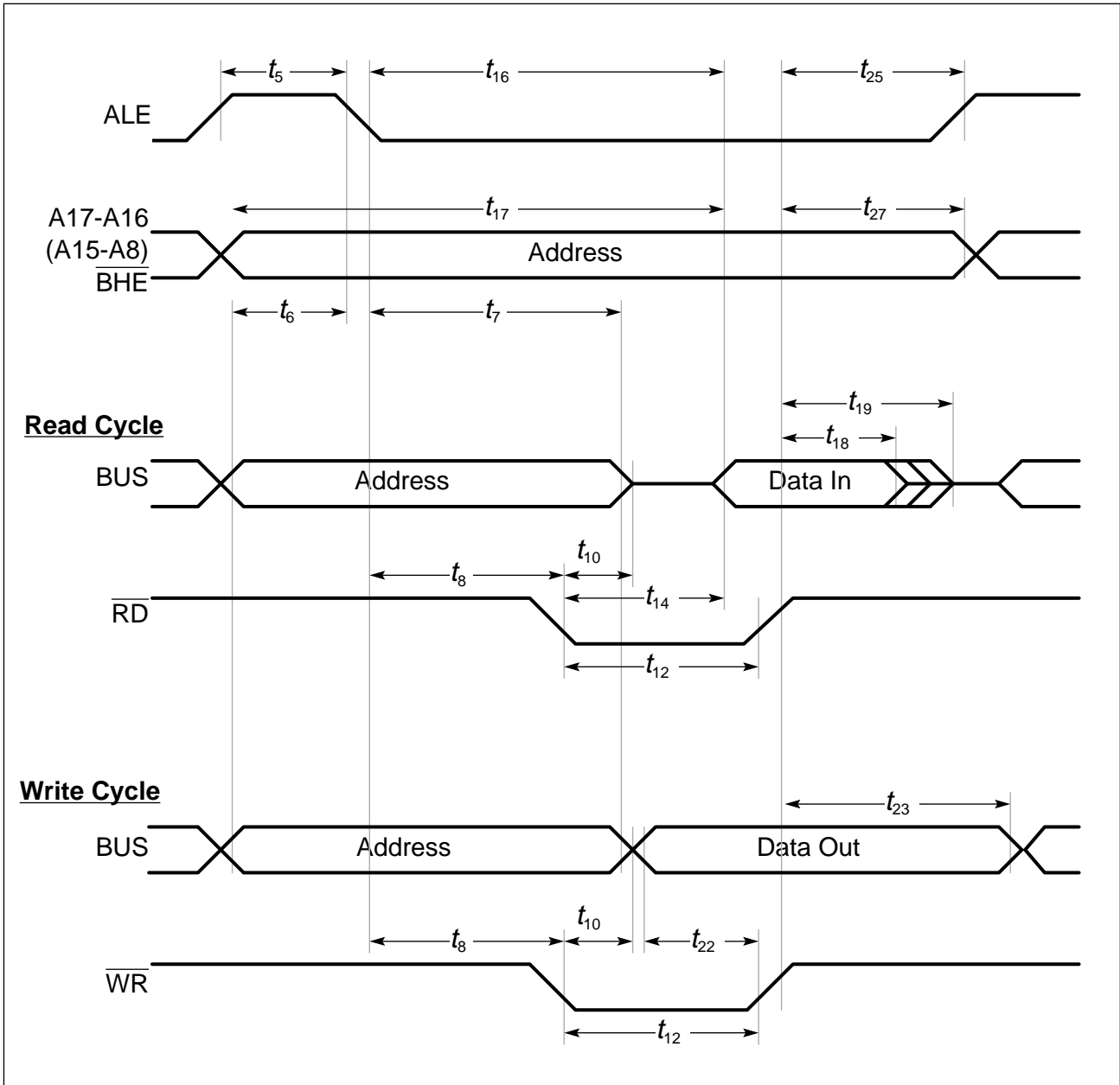
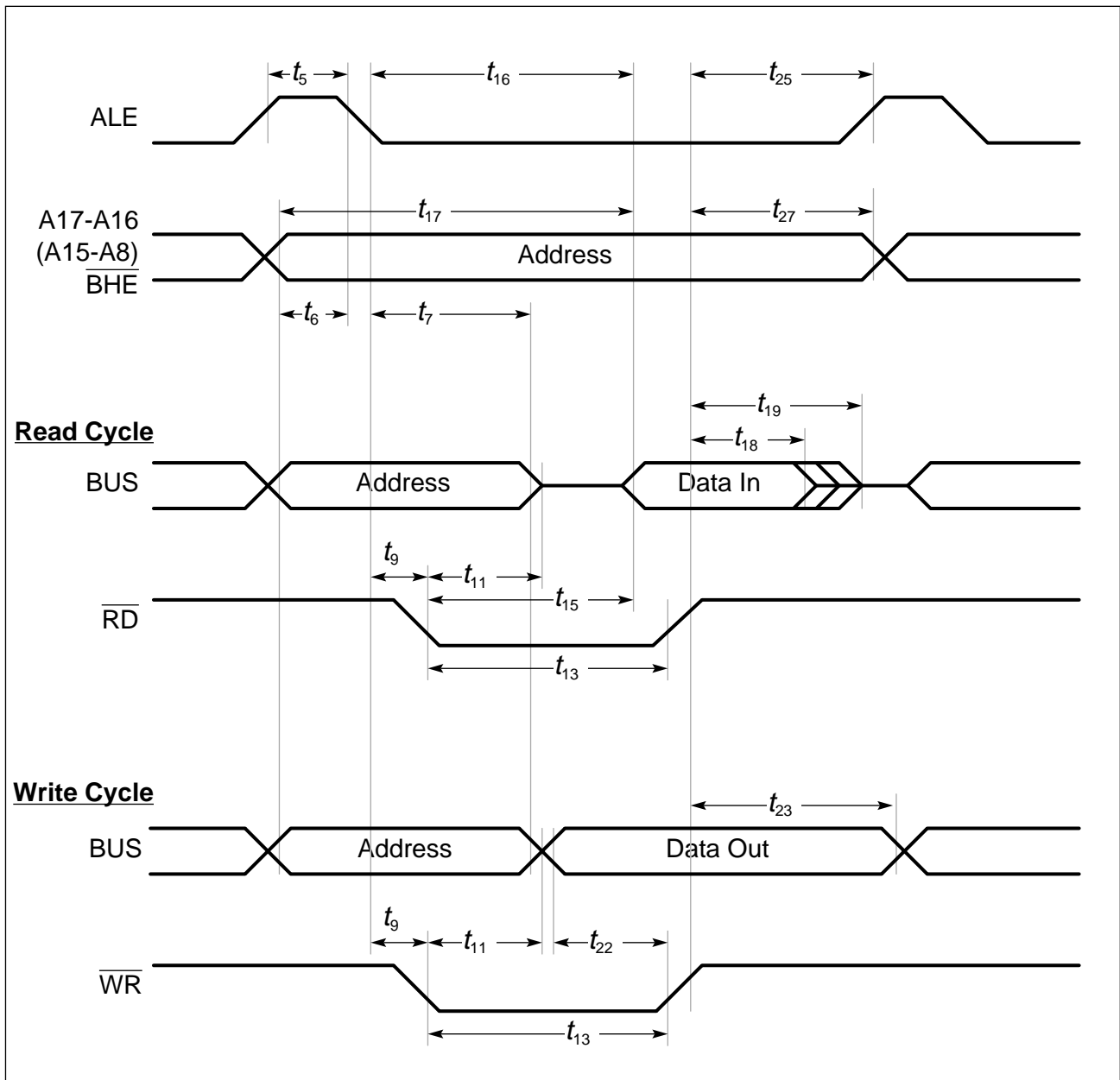


Figure 12-1  
 External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE



**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**

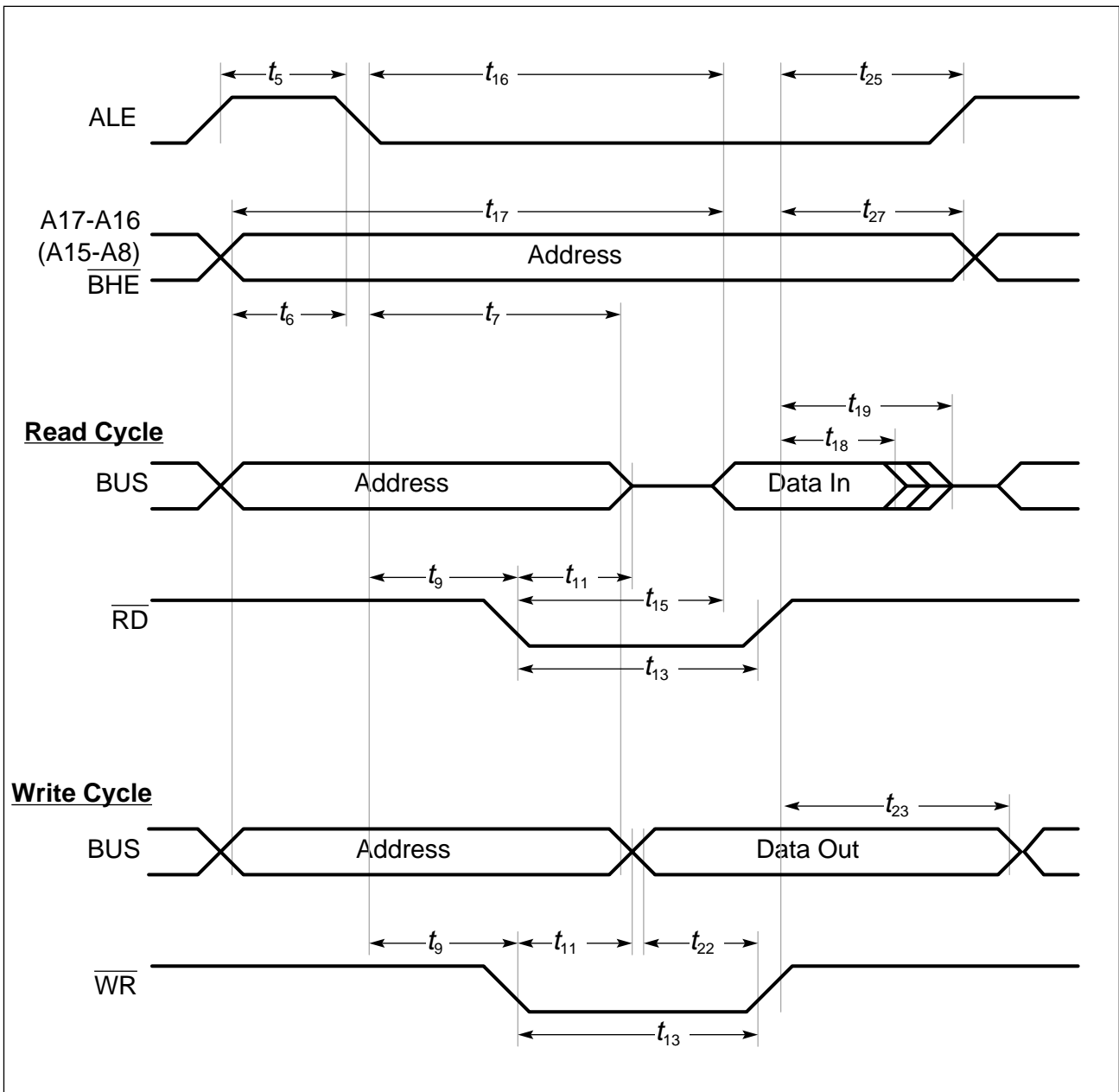


Figure 12-4  
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE



## AC Characteristics (cont'd)

### Demultiplexed Bus

$$V_{CC} = 5\text{ V} \pm 10\%; \quad V_{SS} = 0\text{ V}$$

$$T_A = 0\text{ to } +70\text{ }^\circ\text{C} \quad \text{for SAB 83C166W-5M, SAB 80C166W/83C166W-M}$$

$$T_A = -40\text{ to } +85\text{ }^\circ\text{C} \quad \text{for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3}$$

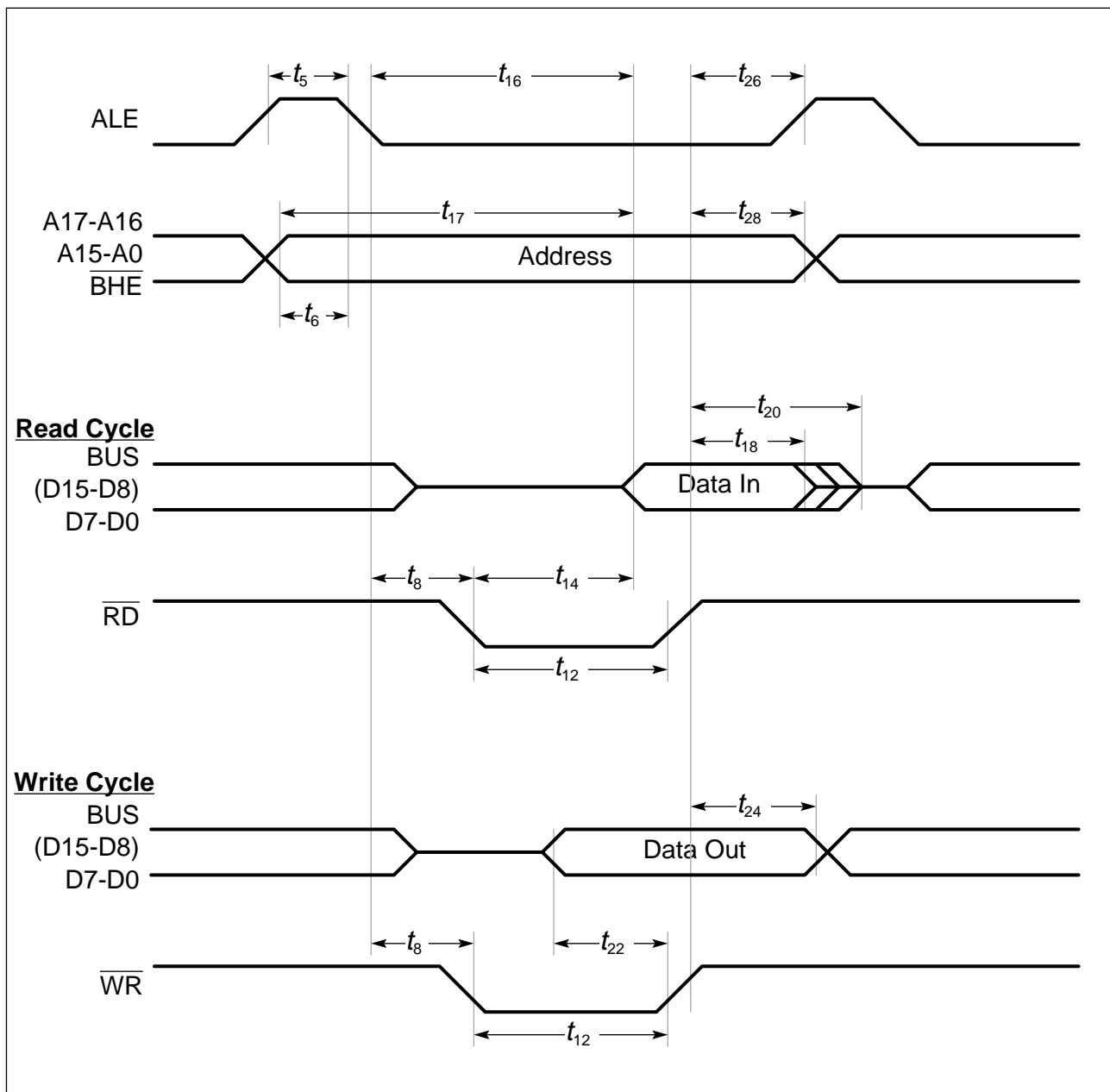
$$T_A = -40\text{ to } +110\text{ }^\circ\text{C} \quad \text{for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4}$$

$$C_L \text{ (for Port 0, Port 1, Port 4, ALE, } \overline{\text{RD}}, \overline{\text{WR}}, \overline{\text{BHE}}, \text{CLKOUT)} = 100\text{ pF}$$

$$\text{ALE cycle time} = 4\text{ TCL (100 ns at 20-MHz CPU clock)}$$

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6	Variable CPU Clock 1/T <sub>CLP</sub> = 1 to 20 MHz		Unit	
			min.	max.		
ALE high time	$t_5$ CC	$15 + t_A$	–	$T_{CLmin} - 10 + t_A$	–	ns
Address setup to ALE	$t_6$ CC	$10 + t_A$	–	$T_{CLmin} - 15 + t_A$	–	ns
ALE falling edge to $\overline{\text{RD}}$ , $\overline{\text{WR}}$ (with RW-delay)	$t_8$ CC	$15 + t_A$	–	$T_{CLmin} - 10 + t_A$	–	ns
ALE falling edge to $\overline{\text{RD}}$ , $\overline{\text{WR}}$ (no RW-delay)	$t_9$ CC	$-10 + t_A$	–	$-10 + t_A$	–	ns
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ low time (with RW-delay)	$t_{12}$ CC	$52.5 + t_C$	–	$T_{CLP} - 10 + t_C$	–	ns
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ low time (no RW-delay)	$t_{13}$ CC	$77.5 + t_C$	–	$T_{CLP} + T_{CLmin} - 10 + t_C$	–	ns
$\overline{\text{RD}}$ to valid data in (with RW-delay)	$t_{14}$ SR	–	$47.5 + t_C$	–	$T_{CLP} - 20 + t_C$	ns
$\overline{\text{RD}}$ to valid data in (no RW-delay)	$t_{15}$ SR	–	$72.5 + t_C$	–	$T_{CLP} + T_{CLmin} - 20 + t_C$	ns
ALE low to valid data in	$t_{16}$ SR	–	$72.5 + t_A + t_C$	–	$T_{CLP} + T_{CLmin} - 20 + t_A + t_C$	ns
Address to valid data in	$t_{17}$ SR	–	$100 + 2t_A + t_C$	–	$2T_{CLP} - 25 + 2t_A + t_C$	ns
Data hold after $\overline{\text{RD}}$ rising edge	$t_{18}$ SR	0	–	0	–	ns
Data float after $\overline{\text{RD}}$ rising edge (with RW-delay)	$t_{20}$ SR	–	$47.5 + t_F$	–	$T_{CLP} - 15 + t_F$	ns
Data float after $\overline{\text{RD}}$ rising edge (no RW-delay)	$t_{21}$ SR	–	$15 + t_F$	–	$T_{CLmin} - 10 + t_F$	ns
Data valid to $\overline{\text{WR}}$	$t_{22}$ CC	$47.5 + t_C$	–	$T_{CLP} - 15 + t_C$	–	ns

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock $1/T_{CLP} = 1$ to 20 MHz		Unit
		min.	max.	min.	max.	
Data hold after $\overline{WR}$	$t_{24}$ CC	$15 + t_F$	–	$T_{CLmin} - 10 + t_F$	–	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



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

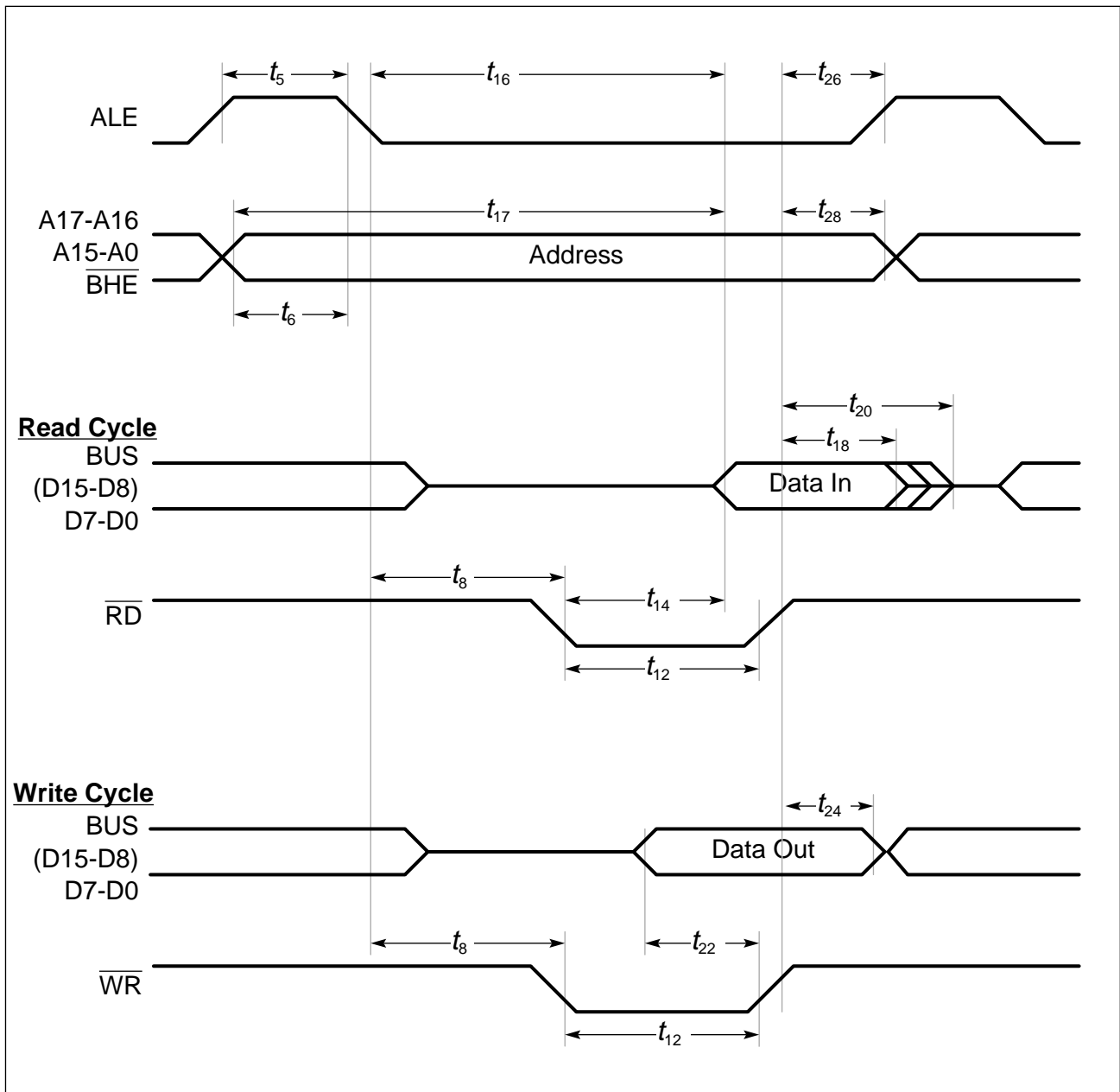


Figure 13-2  
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE

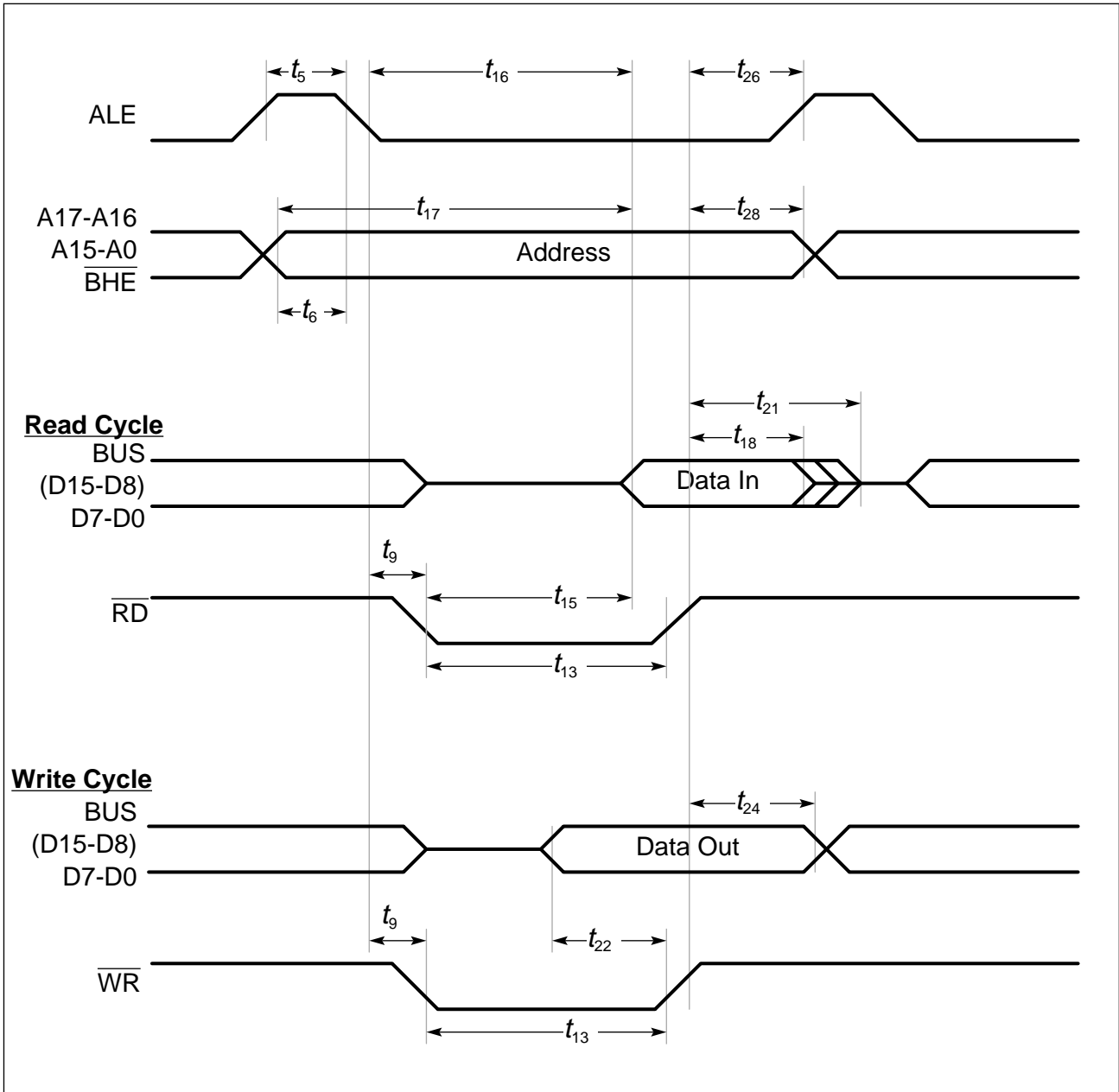


Figure 13-3  
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE

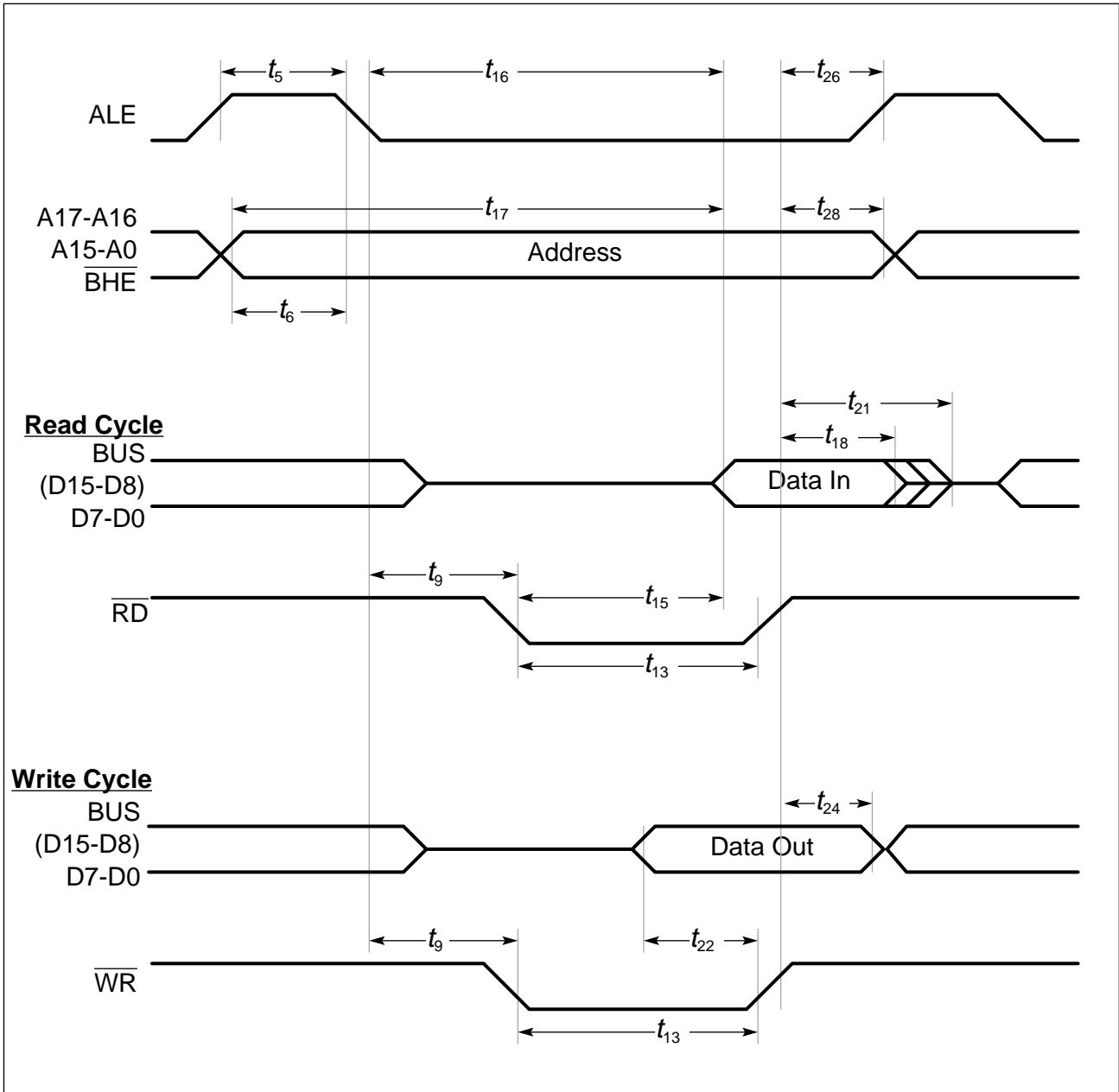


Figure 13-4  
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE

## AC Characteristics (cont'd)

### CLKOUT and READY

$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$

$T_A = 0\text{ to }+70\text{ }^\circ\text{C}$  for SAB 83C166W-5M, SAB 80C166W/83C166W-M

$T_A = -40\text{ to }+85\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3

$T_A = -40\text{ to }+110\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4

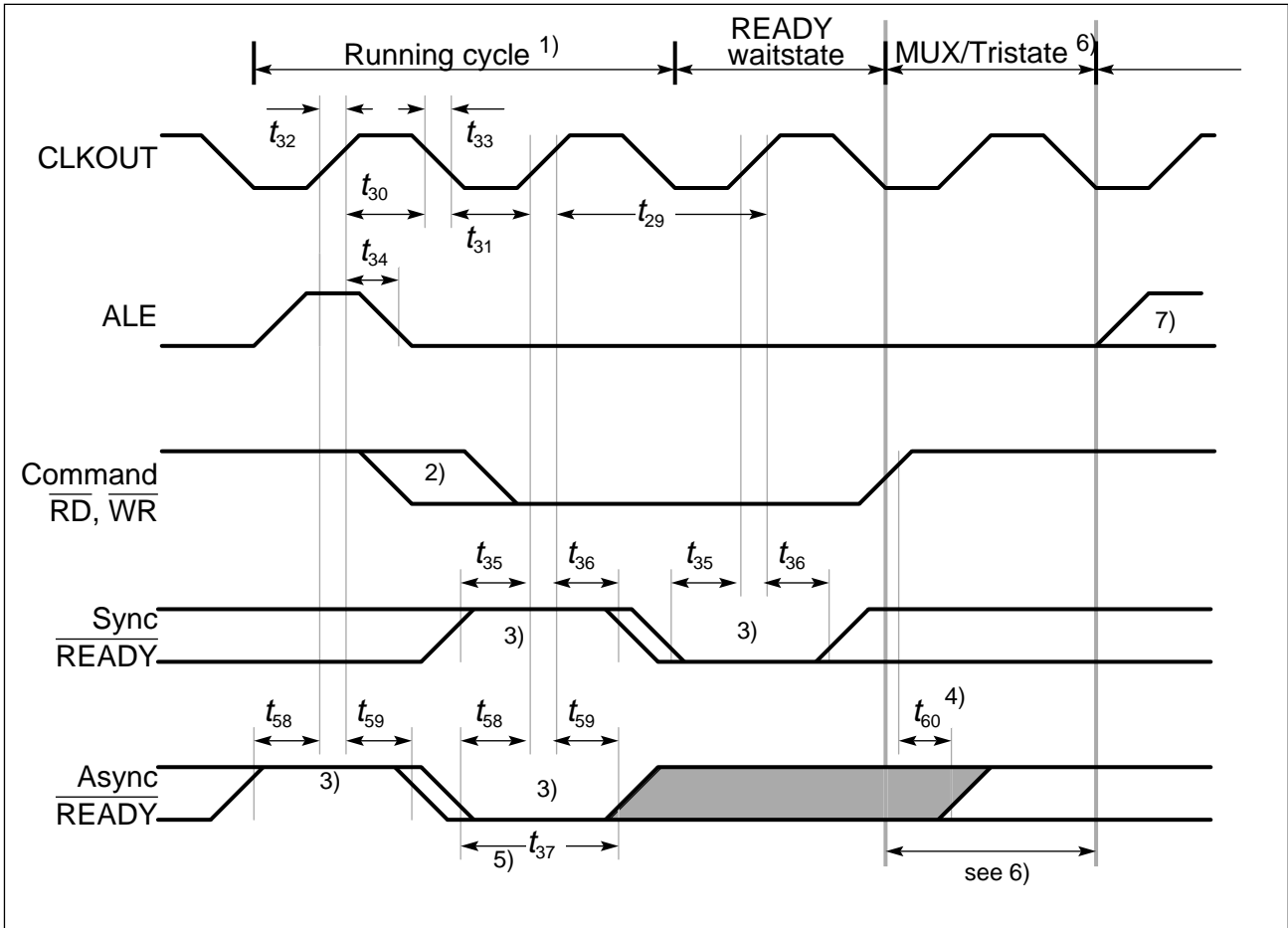
$C_L$  (for Port 0, Port 1, Port 4, ALE, RD, WR, BHE, CLKOUT) = 100 pF

Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6	Variable CPU Clock 1/T <sub>CLP</sub> = 1 to 20 MHz		Unit	
			min.	max.		min.
CLKOUT cycle time	$t_{29}$ CC	62.5	62.5	T <sub>CLP</sub>	T <sub>CLP</sub>	ns
CLKOUT high time	$t_{30}$ CC	15	–	T <sub>CLmin</sub> – 10	–	ns
CLKOUT low time	$t_{31}$ CC	15	–	T <sub>CLmin</sub> – 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	–	T <sub>CLP</sub> + 15	–	ns
Asynchronous $\overline{\text{READY}}$ setup time <sup>1)</sup>	$t_{58}$ SR	20	–	20	–	ns
Asynchronous $\overline{\text{READY}}$ hold time <sup>1)</sup>	$t_{59}$ SR	0	–	0	–	ns
Async. $\overline{\text{READY}}$ hold time after RD, WR high (Demultiplexed Bus) <sup>2)</sup>	$t_{60}$ SR	0	0 + 2 $t_A$ + $t_F$ <sup>2)</sup>	0	T <sub>CL</sub> - 25 + 2 $t_A$ + $t_F$ <sup>2)</sup>	ns

### Notes

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

<sup>2)</sup> Demultiplexed bus is the worst case. For multiplexed bus 2T<sub>CL</sub> 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 (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

$V_{CC} = 5\text{ V} \pm 10\%$ ;  $V_{SS} = 0\text{ V}$

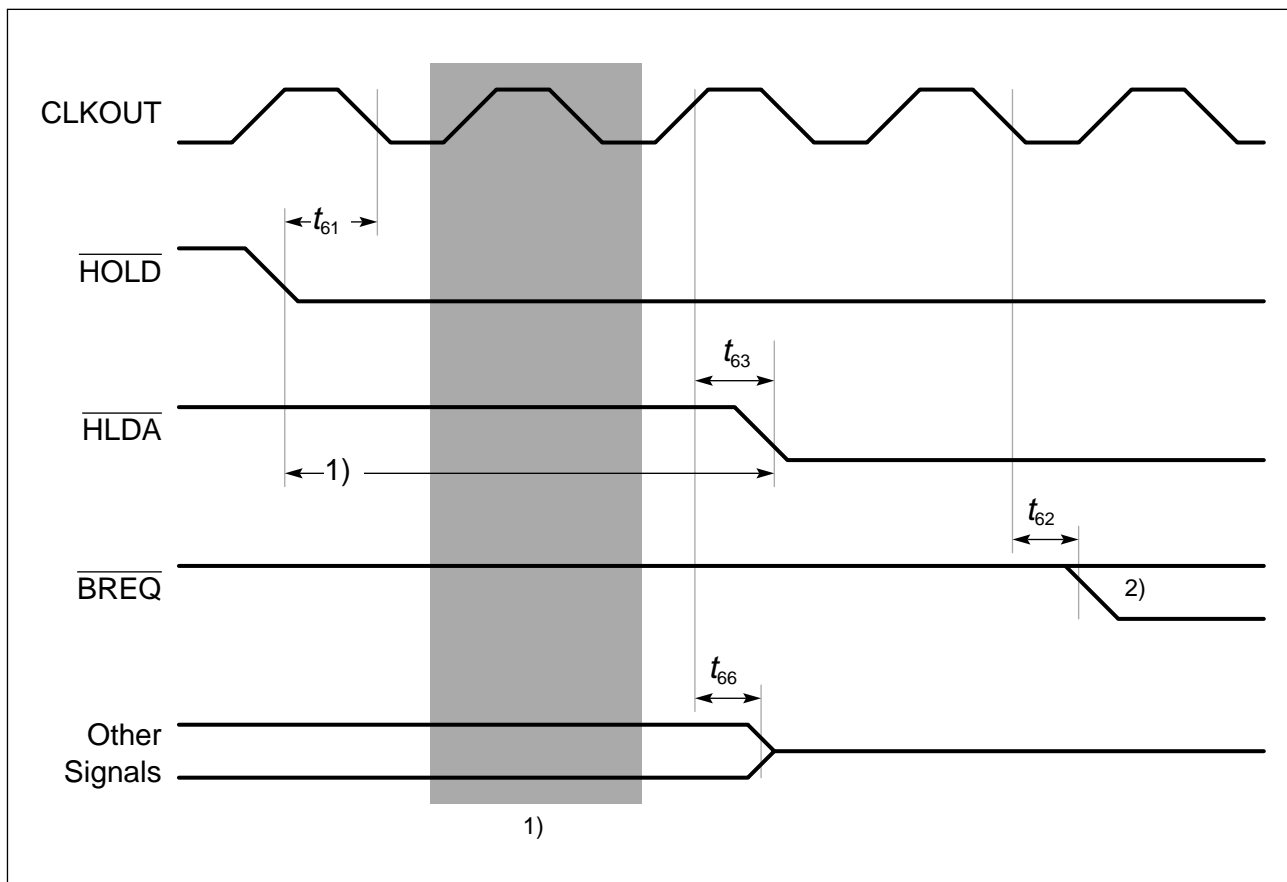
$T_A = 0\text{ to }+70\text{ }^\circ\text{C}$  for SAB 83C166W-5M, SAB 80C166W/83C166W-M

$T_A = -40\text{ to }+85\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3

$T_A = -40\text{ to }+110\text{ }^\circ\text{C}$  for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4

$C_L$  (for Port 0, Port 1, Port 4,  $\overline{\text{ALE}}$ ,  $\overline{\text{RD}}$ ,  $\overline{\text{WR}}$ ,  $\overline{\text{BHE}}$ , CLKOUT) = 100 pF

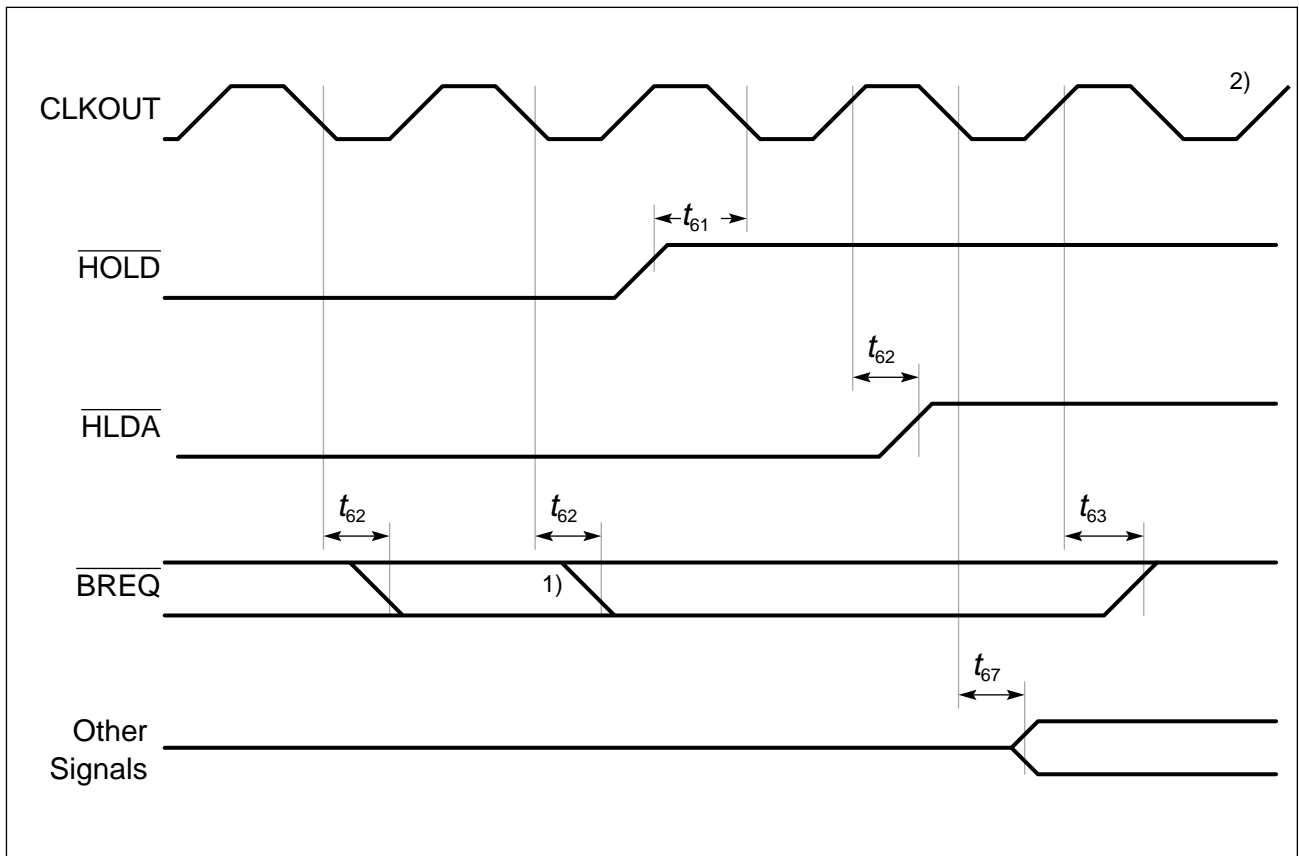
Parameter	Symbol	CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock $1/T_{CLP} = 1\text{ to }20\text{ MHz}$		Unit
		min.	max.	min.	max.	
HOLD input setup time to CLKOUT	$t_{61}$ SR	20	–	20	–	ns
CLKOUT to $\overline{\text{HLDA}}$ high or $\overline{\text{BREQ}}$ low delay	$t_{62}$ CC	–	50	–	50	ns
CLKOUT to $\overline{\text{HLDA}}$ low or $\overline{\text{BREQ}}$ high delay	$t_{63}$ CC	–	60	–	60	ns
Other signals release	$t_{66}$ CC	–	25	–	25	ns
Other signals drive	$t_{67}$ CC	–	60	–	60	ns



**Figure 15**  
**External Bus Arbitration, Releasing the Bus**

**Notes**

- 1) The SAB 80C166W/83C166W 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 80C166W/83C166W requesting the bus.
- 2) The next SAB 80C166W/83C166W driven bus cycle may start here.