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.

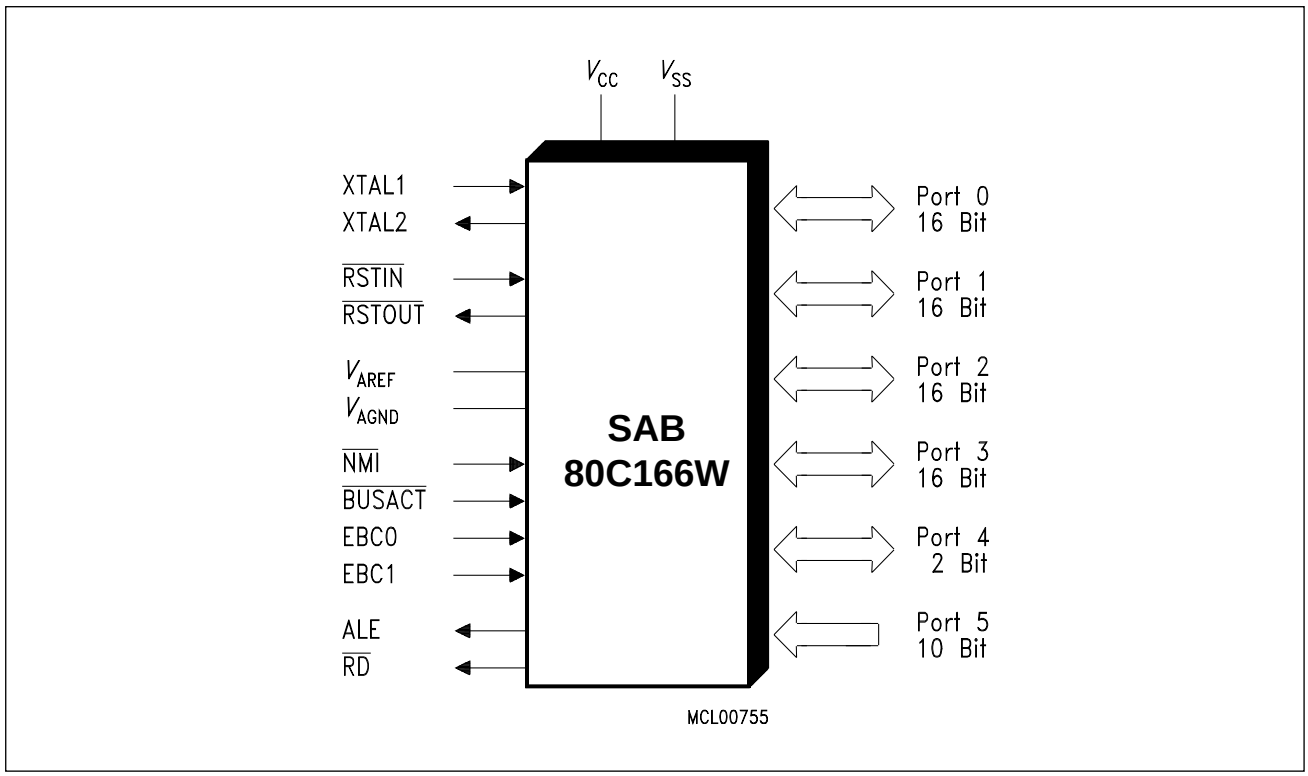


Figure 1
Logic Symbol

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)

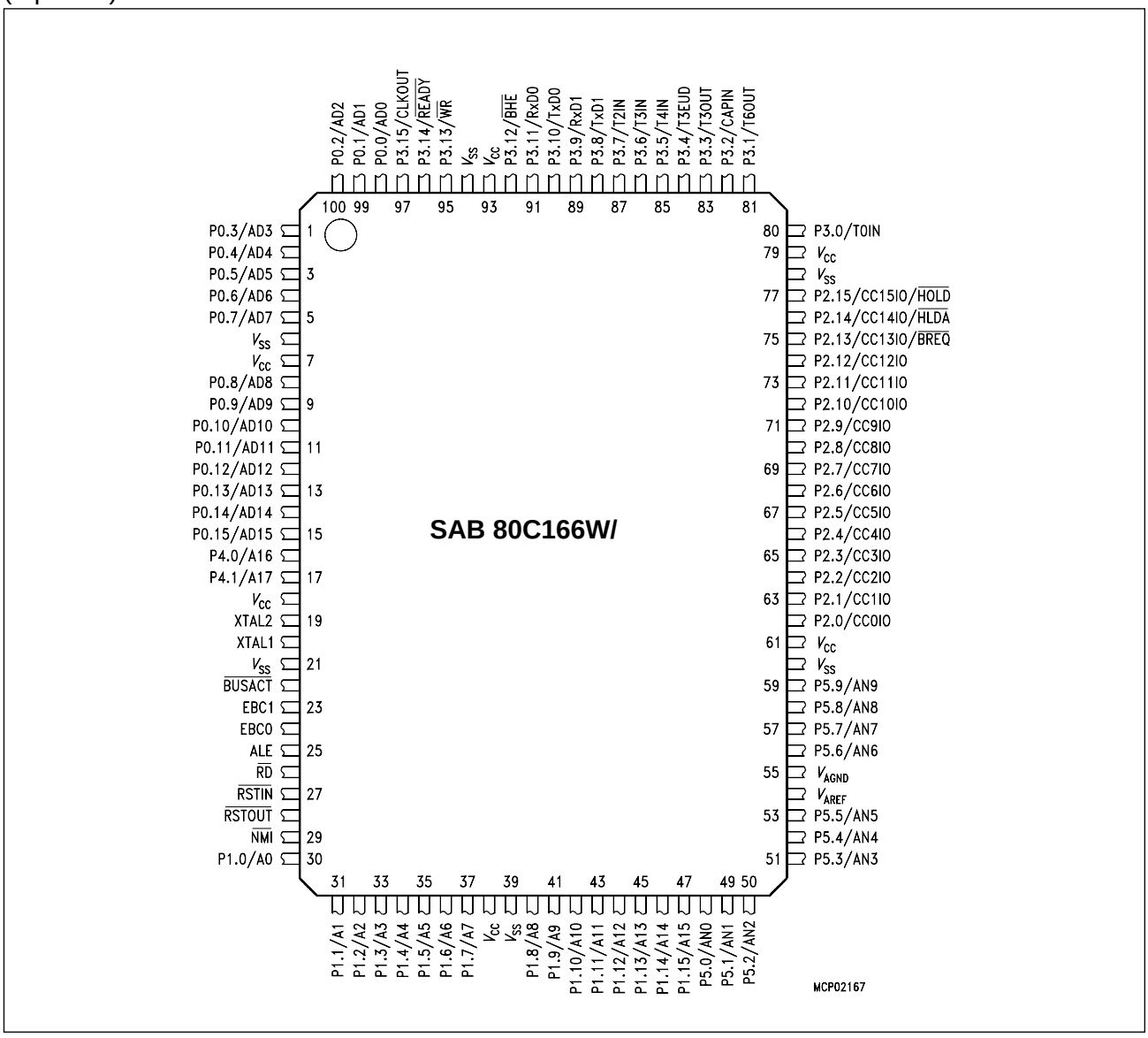


Figure 2

Pin Definitions and Functions

Pin No.	Symbol	Input (I) Output (O)	Function
16 - 17	P4.0 – P4.1	I/O	**Port 4** is a 2-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. In case of an external bus configuration, Port 4 can be used to output the segment address lines: 16 O P4.0 A16 Least Significant Segment Addr. Line 17 O P4.1 A17 Most Significant Segment Addr. Line
20	XTAL1	I	**XTAL1:** Input to the oscillator amplifier and input to the internal clock generator
19	XTAL2	O	**XTAL2:** Output of the oscillator amplifier circuit. To clock the device from an external source, drive XTAL1, while leaving XTAL2 unconnected. Minimum and maximum high/low and rise/fall times specified in the AC Characteristics must be observed.
22 23 24	$\overline{\text{BUSACT}}$, EBC1, EBC0	I I I	**External Bus Configuration** selection inputs. These pins are sampled during reset and select either the single chip mode or one of the four external bus configurations: $\overline{\text{BUSACT}}$ EBC1 EBC0 Mode/Bus Configuration 0 0 0 8-bit demultiplexed bus 0 0 1 8-bit multiplexed bus 0 1 0 16-bit muliplexed bus 0 1 1 16-bit demultiplexed bus 1 0 0 Single chip mode 1 0 1 Reserved. 1 1 0 Reserved. 1 1 1 Reserved. ROMless versions must have pin $\overline{\text{BUSACT}}$ tied to ‘0’.
27	$\overline{\text{RSTIN}}$	I	**Reset Input** with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the SAB 80C166W/83C166W. An internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
28	$\overline{\text{RSTOUT}}$	O	**Internal Reset Indication Output**. This pin is set to a low level when the part is executing either a hardware-, a software- or a watchdog timer reset. $\overline{\text{RSTOUT}}$ remains low until the EINIT (end of initialization) instruction is executed.

Pin Definitions and Functions (cont'd)

Pin No.	Symbol	Input (I) Output (O)	Function
29	$\overline{\text{NMI}}$	I	**Non-Maskable Interrupt** Input. A high to low transition at this pin causes the CPU to vector to the NMI trap routine. When the PWRDN (power down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the SAB 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	**Address Latch Enable** 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	**External Memory Read Strobe.** $\overline{\text{RD}}$ 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	**Port 5** is a 10-bit input-only port with Schmitt-Trigger characteristics. The pins of Port 5 also serve as the (up to 10) analog input channels for the A/D converter, where P5.x equals ANx (Analog input channel x).
62 - 77	P2.0 – P2.15	I/O	**Port 2** is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. The following Port 2 pins also serve for alternate functions:
62		I/O	P2.0 CC0IO CAPCOM: CC0 Cap.-In/Comp.Out
			... ...
75		I/O	P2.13 CC13IO CAPCOM: CC13 Cap.-In/Comp.Out,
		O	$\overline{\text{BREQ}}$ External Bus Request Output
76		I/O	P2.14 CC14IO CAPCOM: CC14 Cap.-In/Comp.Out,
		O	$\overline{\text{HLDA}}$ External Bus Hold Acknowl. Output
77		I/O	P2.15 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	**Port 3** is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. The following Port 3 pins also serve for alternate functions:
80		I	P3.0 T0IN CAPCOM Timer T0 Count Input
81		O	P3.1 T6OUT GPT2 Timer T6 Toggle Latch Output
82		I	P3.2 CAPIN GPT2 Register CAPREL Capture Input
83		O	P3.3 T3OUT GPT1 Timer T3 Toggle Latch Output
84		I	P3.4 T3EUD GPT1 Timer T3 Ext.Up/Down Ctrl.Input
85		I	P3.5 T4IN GPT1 Timer T4 Input for Count/Gate/Reload/Capture
86		I	P3.6 T3IN GPT1 Timer T3 Count/Gate Input
87		I	P3.7 T2IN GPT1 Timer T2 Input for Count/Gate/Reload/Capture
88		O	P3.8 TxD1 ASC1 Clock/Data Output (Asyn./Syn.)
89		I/O	P3.9 RxD1 ASC1 Data Input (Asyn.) or I/O (Syn.)
90		O	P3.10 T×D0 ASC0 Clock/Data Output (Asyn./Syn.)
91		I/O	P3.11 R×D0 ASC0 Data Input (Asyn.) or I/O (Syn.)
92		O	P3.12 $\overline{BHE}$ Ext. Memory High Byte Enable Signal,
95		O	P3.13 $\overline{WR}$ External Memory Write Strobe
96		I	P3.14 $\overline{READY}$ Ready Signal Input
97		O	P3.15 CLKOUT System Clock Output (=CPU Clock)
98 – 5 8 – 15	P0.0 – P0.15	I/O	**Port 0** is a 16-bit bidirectional IO port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. In case of an external bus configuration, Port 0 serves as the address (A) and address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes. **Demultiplexed bus modes:** Data Path Width: 8-bit 16-bit; P0.0 – P0.7: D0 – D7 D0 - D7; P0.8 – P0.15: output! D8 - D15. **Multiplexed bus modes:** Data Path Width: 8-bit 16-bit; P0.0 – P0.7: AD0 – AD7 AD0 - AD7; P0.8 – P0.15: A8 - A15 AD8 - AD15
54	V_{AREF}	-	**Reference voltage** for the A/D converter.
55	V_{AGND}	-	**Reference ground** 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}	-	**Digital Supply Voltage:** + 5 V during normal operation and idle mode. ≥ 2.5 V during power down mode
6, 21, 39, 60, 78, 94	V_{SS}	-	**Digital Ground.**

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).

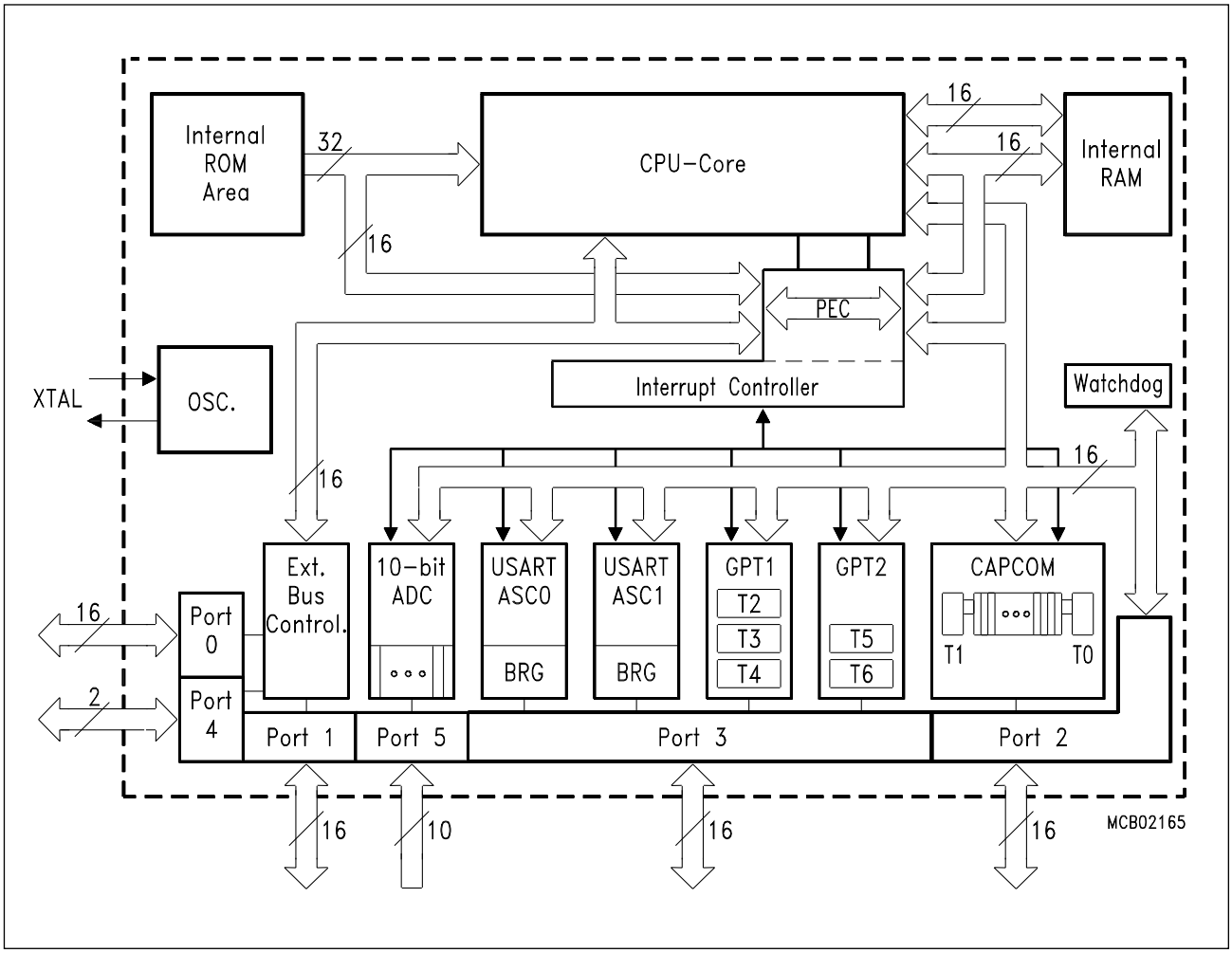


Figure 3
Block Diagram

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 bytewise 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 bytewide (RL0, RH0, …, RL7, RH7) so-called General Purpose Registers (GPRs).

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

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

External Bus Controller

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

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

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

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

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

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

Important timing characteristics of the external bus interface (Memory Cycle Time, Memory Tri-State Time, Read/Write Delay and Length of ALE, 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 × 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.

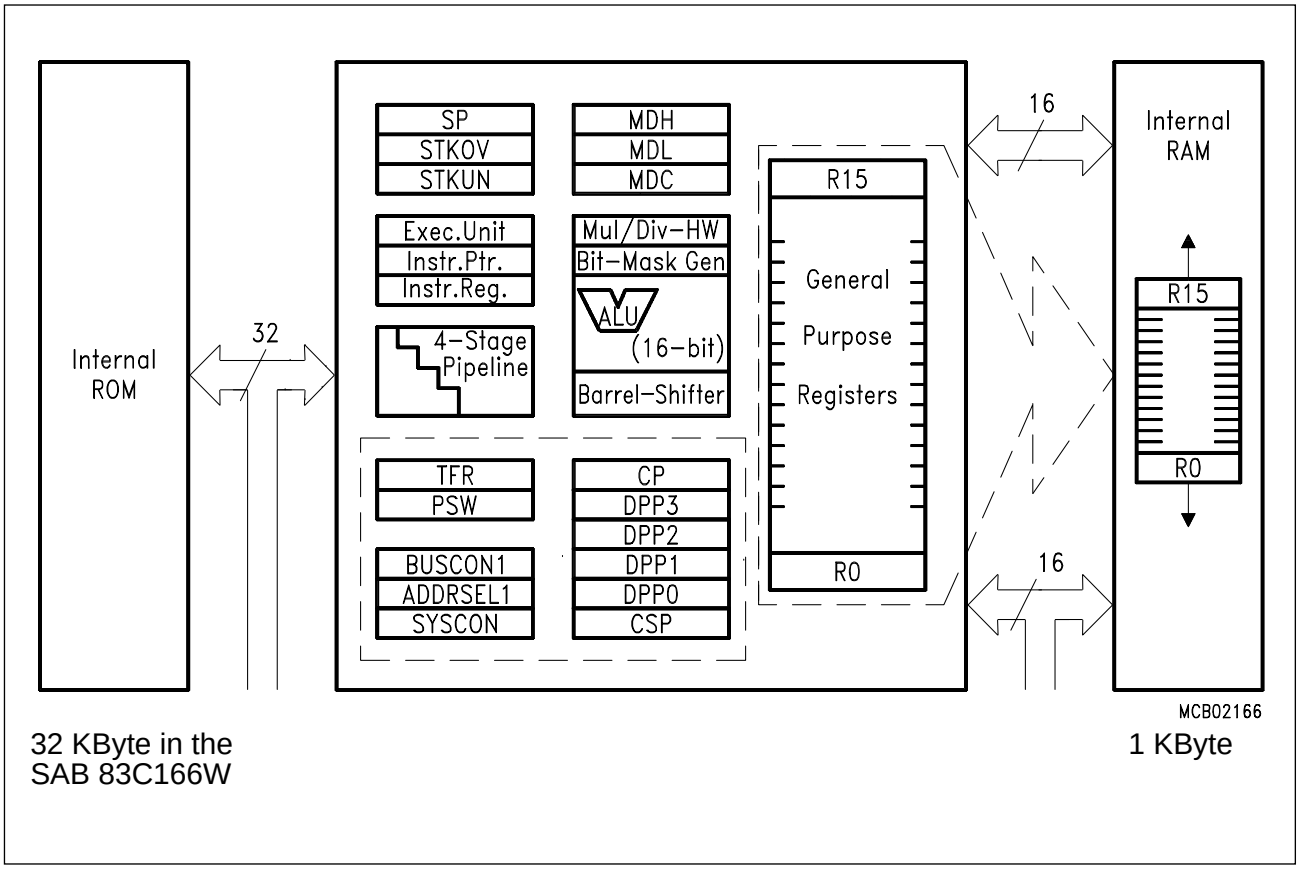


Figure 4
CPU Block Diagram

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 occurence 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 implicity decremented for each PEC service except when performing in the continuous transfer mode. When this counter reaches zero, a standard interrupt is performed to the corresponding source related vector location. PEC services are very well suited, for example, for supporting the transmission or reception of blocks of data, or for transferring A/D converted results to a memory table. The SAB 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_H	10_H
CAPCOM Register 1	CC1IR	CC1IE	CC1INT	44_H	11_H
CAPCOM Register 2	CC2IR	CC2IE	CC2INT	48_H	12_H
CAPCOM Register 3	CC3IR	CC3IE	CC3INT	$4C_H$	13_H
CAPCOM Register 4	CC4IR	CC4IE	CC4INT	50_H	14_H
CAPCOM Register 5	CC5IR	CC5IE	CC5INT	54_H	15_H
CAPCOM Register 6	CC6IR	CC6IE	CC6INT	58_H	16_H
CAPCOM Register 7	CC7IR	CC7IE	CC7INT	$5C_H$	17_H
CAPCOM Register 8	CC8IR	CC8IE	CC8INT	60_H	18_H
CAPCOM Register 9	CC9IR	CC9IE	CC9INT	64_H	19_H
CAPCOM Register 10	CC10IR	CC10IE	CC10INT	68_H	$1A_H$
CAPCOM Register 11	CC11IR	CC11IE	CC11INT	$6C_H$	$1B_H$
CAPCOM Register 12	CC12IR	CC12IE	CC12INT	70_H	$1C_H$
CAPCOM Register 13	CC13IR	CC13IE	CC13INT	74_H	$1D_H$
CAPCOM Register 14	CC14IR	CC14IE	CC14INT	78_H	$1E_H$
CAPCOM Register 15	CC15IR	CC15IE	CC15INT	$7C_H$	$1F_H$
CAPCOM Timer 0	T0IR	T0IE	T0INT	80_H	20_H
CAPCOM Timer 1	T1IR	T1IE	T1INT	84_H	21_H
GPT1 Timer 2	T2IR	T2IE	T2INT	88_H	22_H
GPT1 Timer 3	T3IR	T3IE	T3INT	$8C_H$	23_H
GPT1 Timer 4	T4IR	T4IE	T4INT	90_H	24_H
GPT2 Timer 5	T5IR	T5IE	T5INT	94_H	25_H
GPT2 Timer 6	T6IR	T6IE	T6INT	98_H	26_H
GPT2 CAPREL Register	CRIR	CRIE	CRINT	$9C_H$	27_H
A/D Conversion Complete	ADCIR	ADCIE	ADCINT	$A0_H$	28_H
A/D Overrun Error	ADEIR	ADEIE	ADEINT	$A4_H$	29_H
ASC0 Transmit	S0TIR	S0TIE	S0TINT	$A8_H$	$2A_H$
ASC0 Receive	S0RIR	S0RIE	S0RINT	AC_H	$2B_H$
ASC0 Error	S0EIR	S0EIE	S0EINT	$B0_H$	$2C_H$
ASC1 Transmit	S1TIR	S1TIE	S1TINT	$B4_H$	$2D_H$
ASC1 Receive	S1RIR	S1RIE	S1RINT	$B8_H$	$2E_H$
ASC1 Error	S1EIR	S1EIE	S1EINT	BC_H	$2F_H$

The SAB 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 occurence of a hardware trap is additionally signified by an individual bit in the trap flag register (TFR). Except when another higher prioritized trap service is in progress, a hardware trap will interrupt any actual program execution. In turn, hardware trap services can normally not be interrupted by standard or PEC interrupts.

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

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

Capture/Compare (CAPCOM) Unit

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

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

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

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

When a capture/compare register has been selected for capture mode, the current contents of the allocated timer will be latched (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.

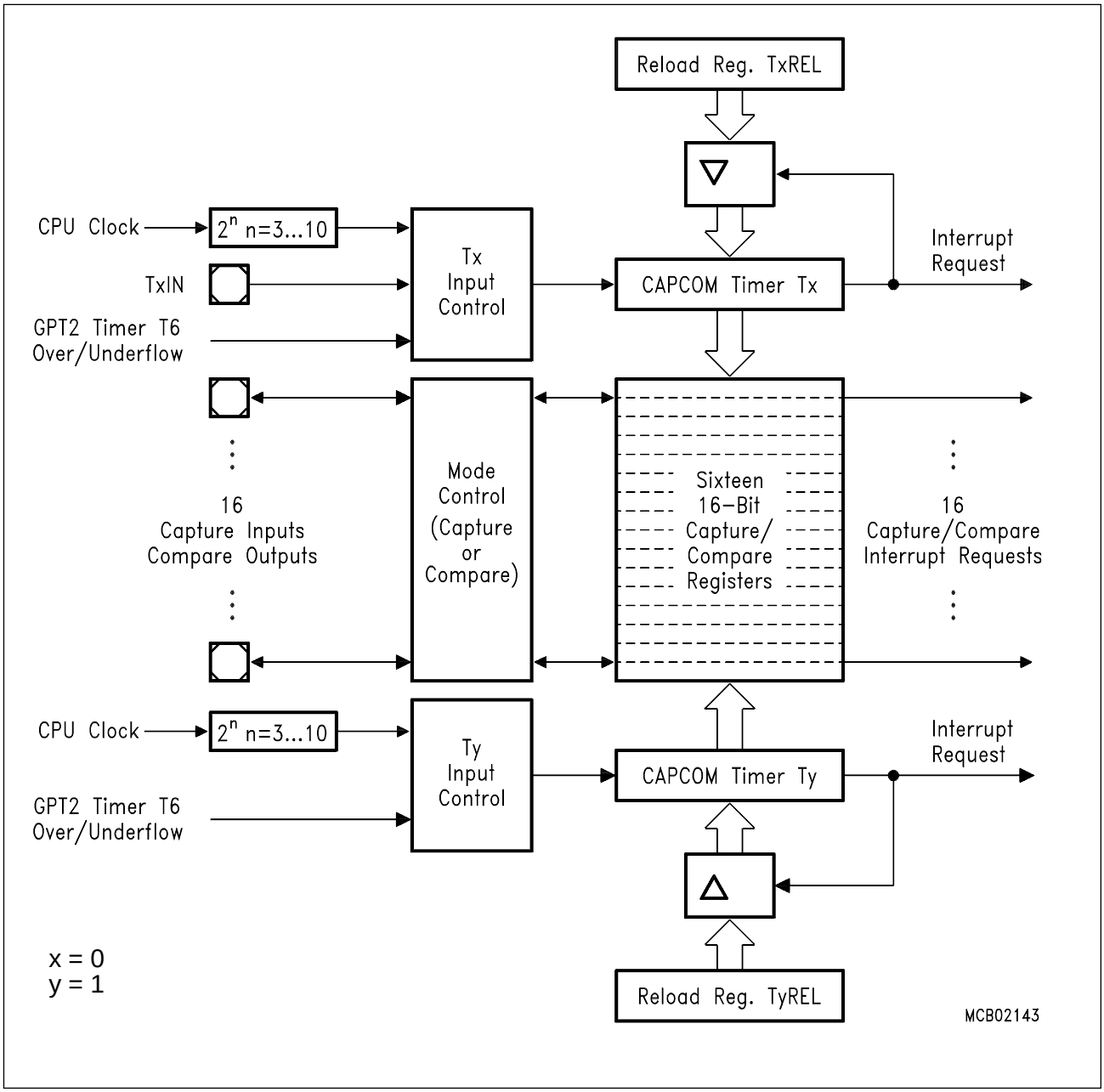


Figure 5
CAPCOM-Unit Block Diagram

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).

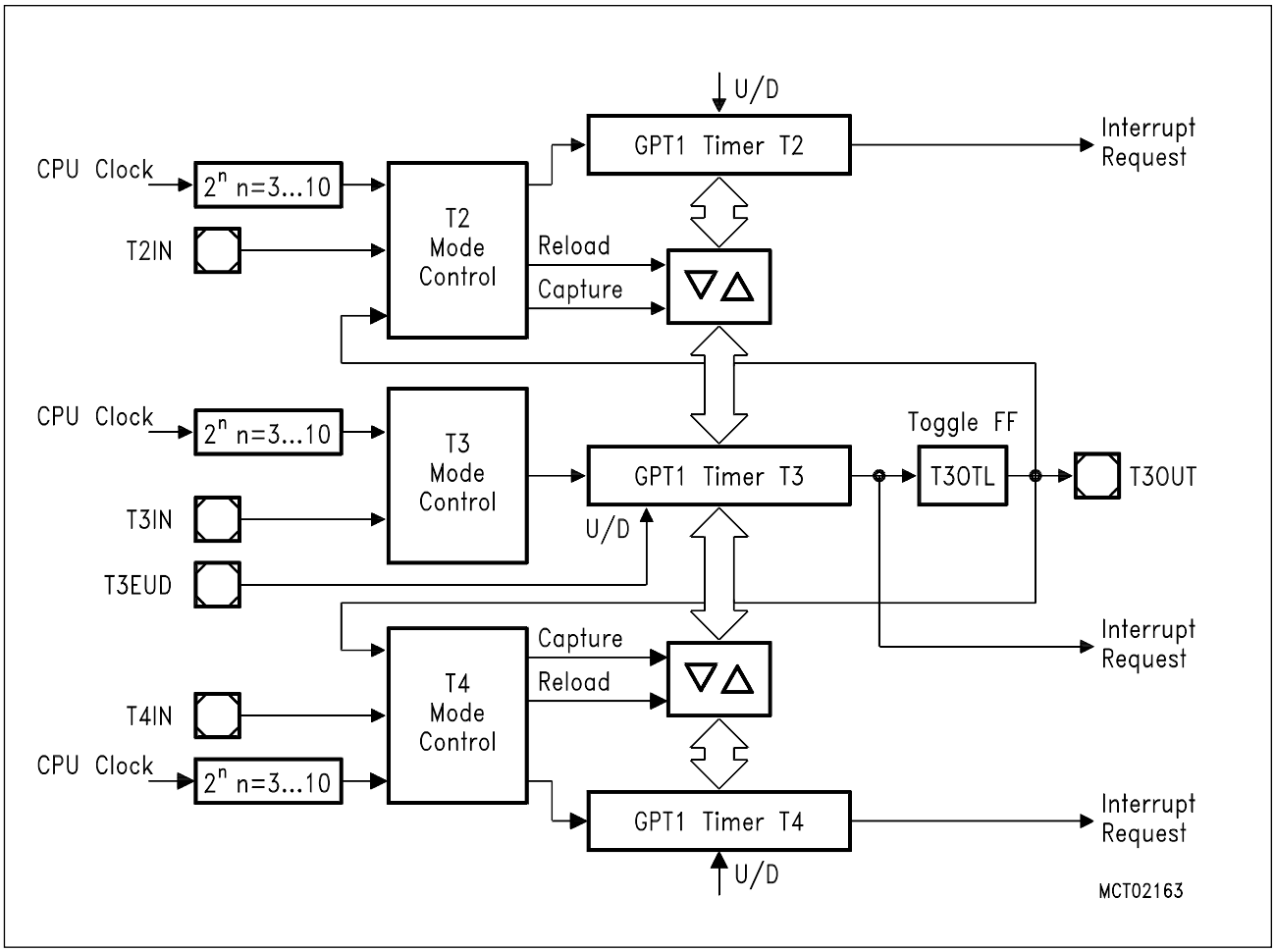


Figure 6
Block Diagram of GPT1

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.

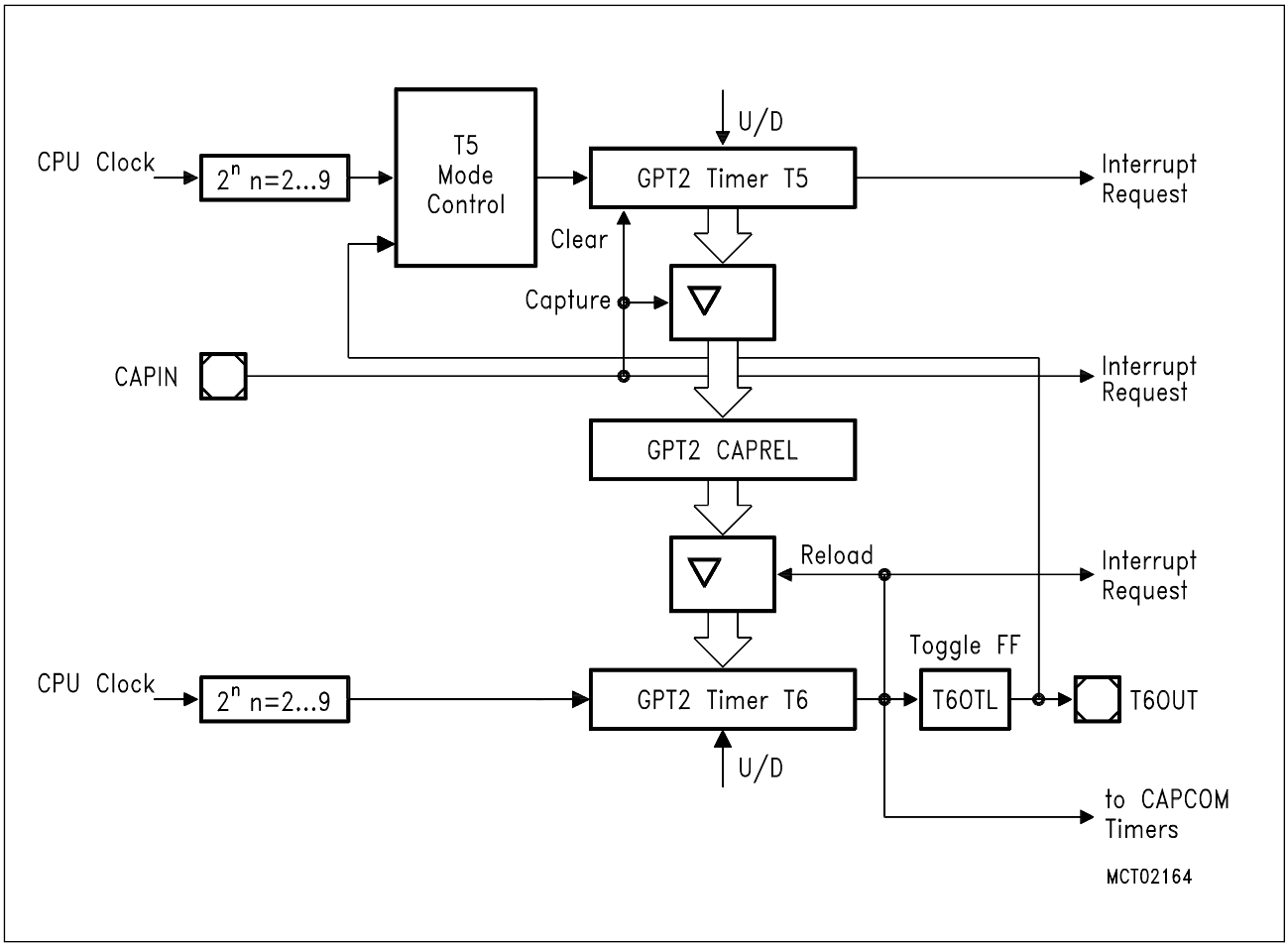


Figure 7
Block Diagram of GPT2

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 $\overline{\text{RSTOUT}}$ pin low in order to allow external hardware components to be reset.

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

Bootstrap Loader

The SAB 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 $\overline{\text{NMI}}$ becomes active directly after the end of the internal reset sequence. BSL mode is entered independent of the bus mode selected via EBC0, EBC1 and $\overline{\text{BUSACT}}$.

After entering BSL mode the SAB 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 $<55_H>$.

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

The SAB 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
Arithmetic Operations			
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
Data Movement (cont'd)			
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	Rw, 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 $\overline{\text{NMI}}$-pin being low)	4
SRVWDT	Service Watchdog Timer	4
DISWDT	Disable Watchdog Timer	4
EINIT	Signify End-of-Initialization on RSTOUT-pin	4
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
00	2	ADD	Rw, Rw
01	2	ADDB	Rb, Rb
02	4	ADD	reg, mem
03	4	ADDB	reg, mem
04	4	ADD	mem, reg
05	4	ADDB	mem, reg
06	4	ADD	reg, #data16
07	4	ADDB	reg, #data8
08	2	ADD	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
09	2	ADDB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
0A	4	BFLDL	bitoff, #mask8, #data8
0B	2	MUL	Rw, Rw
0C	2	ROL	Rw, Rw
0D	2	JMPR	cc_UC, rel
0E	2	BCLR	bitoff.0
0F	2	BSET	bitoff.0
10	2	ADDC	Rw, Rw
11	2	ADDCB	Rb, Rb
12	4	ADDC	reg, mem
13	4	ADDCB	reg, mem
14	4	ADDC	mem, reg
15	4	ADDCB	mem, reg
16	4	ADDC	reg, #data16
17	4	ADDCB	reg, #data8
18	2	ADDC	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
19	2	ADDCB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
1A	4	BFLDH	bitoff, #mask8, #data8
1B	2	MULU	Rw, Rw
1C	2	ROL	Rw, #data4
1D	2	JMPR	cc_NET, rel
1E	2	BCLR	bitoff.1
1F	2	BSET	bitoff.1
20	2	SUB	Rw, Rw
21	2	SUBB	Rb, Rb
22	4	SUB	reg, mem
23	4	SUBB	reg, mem
24	4	SUB	mem, reg
25	4	SUBB	mem, reg
26	4	SUB	reg, #data6
27	4	SUBB	reg, #data8
28	2	SUB	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
29	2	SUBB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
2A	4	BCMP	bitaddr, bitaddr
2B	2	PRIOR	Rw, Rw
2C	2	ROR	Rw, Rw
2D	2	JMPR	cc_EQ, rel or cc_Z, rel
2E	2	BCLR	bitoff.2
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
33	4	SUBCB	reg, mem
34	4	SUBC	mem, reg
35	4	SUBCB	mem, reg
36	4	SUBC	reg, #data16
37	4	SUBCB	reg, #data8
38	2	SUBC	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
39	2	SUBCB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
3A	4	BMOVN	bitaddr, bitaddr
3B	-	-	-
3C	2	ROR	Rw, #data4
3D	2	JMPR	cc_NE, rel or cc_NZ, rel
3E	2	BCLR	bitoff.3
3F	2	BSET	bitoff.3
40	2	CMP	Rw, Rw
41	2	CMPB	Rb, Rb
42	4	CMP	reg, mem
43	4	CMPB	reg, mem
44	-	-	-
45	-	-	-
46	4	CMP	reg, #data16
47	4	CMPB	reg, #data8
48	2	CMP	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
49	2	CMPB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
4A	4	BMOV	bitaddr, bitaddr
4B	2	DIV	Rw
4C	2	SHL	Rw, Rw
4D	2	JMPR	cc_V, rel
4E	2	BCLR	bitoff.4
4F	2	BSET	bitoff.4
50	2	XOR	Rw, Rw
51	2	XORB	Rb, Rb
52	4	XOR	reg, mem
53	4	XORB	reg, mem
54	4	XOR	mem, reg
55	4	XORB	mem, reg
56	4	XOR	reg, #data16
57	4	XORB	reg, #data8
58	2	XOR	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
59	2	XORB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
5A	4	BOR	bitaddr, bitaddr
5B	2	DIVU	Rw
5C	2	SHL	Rw, #data4
5D	2	JMPR	cc_NV, rel
5E	2	BCLR	bitoff.5
5F	2	BSET	bitoff.5
60	2	AND	Rw, Rw
61	2	ANDB	Rb, Rb
62	4	AND	reg, mem
63	4	ANDB	reg, mem
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
68	2	AND	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
69	2	ANDB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
6A	4	BAND	bitaddr, bitaddr
6B	2	DIVL	Rw
6C	2	SHR	Rw, Rw
6D	2	JMPR	cc_N, rel
6E	2	BCLR	bitoff.6
6F	2	BSET	bitoff.6
70	2	OR	Rw, Rw
71	2	ORB	Rb, Rb
72	4	OR	reg, mem
73	4	ORB	reg, mem
74	4	OR	mem, reg
75	4	ORB	mem, reg
76	4	OR	reg, #data16
77	4	ORB	reg, #data8
78	2	OR	Rw, [Rw +] or Rw, [Rw] or Rw, #data3 [1]
79	2	ORB	Rb, [Rw +] or Rb, [Rw] or Rb, #data3 [1]
7A	4	BXOR	bitaddr, bitaddr
7B	2	DIVLU	Rw
7C	2	SHR	Rw, #data4
7D	2	JMPR	cc_NN, rel
7E	2	BCLR	bitoff.7
7F	2	BSET	bitoff.7
80	2	CMPI1	Rw, #data4
81	2	NEG	Rw
82	4	CMPI1	Rw, mem
83	-	-	-
84	4	MOV	[Rw], mem
85	-	-	-
86	4	CMPI1	Rw, #data16
87	4	IDLE	
88	2	MOV	[-Rw], Rw
89	2	MOVB	[-Rw], Rb
8A	4	JB	bitaddr, rel
8B	-	-	-
8C	-	-	-
8D	2	JMPR	cc_C, rel or cc_ULT, rel
8E	2	BCLR	bitoff.8
8F	2	BSET	bitoff.8
90	2	CMPI2	Rw, #data4
91	2	CPL	Rw
92	4	CMPI2	Rw, mem
93	-	-	-
94	4	MOV	mem, [Rw]
95	-	-	-
96	4	CMPI2	Rw, #data16
97	4	PWRDN	
98	2	MOV	Rw, [Rw+]
99	2	MOVB	Rb, [Rw+]
9A	4	JNB	bitaddr, rel
9B	2	TRAP	#trap7
9C	2	JMPI	cc, [Rw]
9D	2	JMPR	cc_NC, rel or cc_UGE, rel
9E	2	BCLR	bitoff.9
9F	2	BSET	bitoff.9
A0	2	CMPD1	Rw, #data4

Hex-code	Num-ber of Bytes	Mnemonic	Operands
A1	2	NEGB	Rb
A2	4	CMPD1	Rw, mem
A3	-	-	-
A4	4	MOVB	[Rw], mem
A5	4	DISWDT	
A6	4	CMPD1	Rw, #data16
A7	4	SRVWDT	
A8	2	MOV	Rw, [Rw]
A9	2	MOVB	Rb, [Rw]
AA	4	JBC	bitaddr, rel
AB	2	CALLI	cc, [Rw]
AC	2	ASHR	Rw, Rw
AD	2	JMPR	cc_SGT, rel
AE	2	BCLR	bitoff.10
AF	2	BSET	bitoff.10
B0	2	CMPD2	Rw, #data4
B1	2	CPLB	Rb
B2	4	CMPD2	Rw, mem
B3	-	-	-
B4	4	MOVB	mem, [Rw]
B5	4	EINIT	
B6	4	CMPD2	Rw, #data16
B7	4	SRST	
B8	2	MOV	[Rw], Rw
B9	2	MOVB	[Rw], Rb
BA	4	JNBS	bitaddr, rel
BB	2	CALLR	rel
BC	2	ASHR	Rw, #data4
BD	2	JMPR	cc_SLE, rel
BE	2	BCLR	bitoff.11
BF	2	BSET	bitoff.11
C0	2	MOVBZ	Rw, Rb

Hex-code	Num-ber of Bytes	Mnemonic	Operands
C1	-	-	-
C2	4	MOVBZ	reg, mem
C3	-	-	-
C4	4	MOV	[Rw + #data16], Rw
C5	4	MOVBZ	mem, reg
C6	4	SCXT	reg, #data16
C7	-	-	-
C8	2	MOV	[Rw], [Rw]
C9	2	MOVB	[Rw], [Rw]
CA	4	CALLA	cc, addr
CB	2	RET	
CC	2	NOP	
CD	2	JMPR	cc_SLT, rel
CE	2	BCLR	bitoff.12
CF	2	BSET	bitoff.12
D0	2	MOVBS	Rw, Rb
D1	-	-	-
D2	4	MOVBS	reg, mem
D3	-	-	-
D4	4	MOV	reg, mem
D5	4	MOVBS	mem, reg
D6	4	SCXT	reg, mem
D7	-	-	-
D8	2	MOV	[Rw+], [Rw]
D9	2	MOVB	[Rw+], [Rw]
DA	4	CALLS	seg, caddr
DB	2	RETS	
DC	-	-	-
DD	2	JMPR	cc_SGE, rel
DE	2	BCLR	bitoff.13
DF	2	BSET	bitoff.13

Hex-code	Num-ber of Bytes	Mnemonic	Operands
E0	2	MOV	Rw, #data4
E1	2	MOVB	Rb, #data4
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
FE	2	BCLR	bitoff.15
FF	2	BSET	bitoff.15

Notes

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

$x0_H – x7_H$:	Rw, #data3	or	Rb, #data3
$x8_H – xB_H$:	Rw, [Rw]	or	Rb, [Rw]
$xC_H – xF_H$:	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
ADCIC b	$FF98_H$	CC_H	A/D Converter End of Conversion Interrupt Control Register	0000_H
ADCON b	$FFA0_H$	$D0_H$	A/D Converter Control Register	0000_H
ADDAT	$FEA0_H$	50_H	A/D Converter Result Register	0000_H
ADDRSEL1	$FE18_H$	$0C_H$	Address Select Register 1	0000_H
ADEIC b	$FF9A_H$	CD_H	A/D Converter Overrun Error Interrupt Control Register	0000_H
BUSCON1 b	$FF14_H$	$8A_H$	Bus Configuration Register 1	0000_H
CAPREL	$FE4A_H$	25_H	GPT2 Capture/Reload Register	0000_H
CC0	$FE80_H$	40_H	CAPCOM Register 0	0000_H
CC0IC b	$FF78_H$	BC_H	CAPCOM Register 0 Interrupt Control Register	0000_H
CC1	$FE82_H$	41_H	CAPCOM Register 1	0000_H
CC1IC b	$FF7A_H$	BD_H	CAPCOM Register 1 Interrupt Control Register	0000_H
CC2	$FE84_H$	42_H	CAPCOM Register 2	0000_H
CC2IC b	$FF7C_H$	BE_H	CAPCOM Register 2 Interrupt Control Register	0000_H
CC3	$FE86_H$	43_H	CAPCOM Register 3	0000_H
CC3IC b	$FF7E_H$	BF_H	CAPCOM Register 3 Interrupt Control Register	0000_H
CC4	$FE88_H$	44_H	CAPCOM Register 4	0000_H
CC4IC b	$FF80_H$	$C0_H$	CAPCOM Register 4 Interrupt Control Register	0000_H
CC5	$FE8A_H$	45_H	CAPCOM Register 5	0000_H
CC5IC b	$FF82_H$	$C1_H$	CAPCOM Register 5 Interrupt Control Register	0000_H
CC6	$FE8C_H$	46_H	CAPCOM Register 6	0000_H
CC6IC b	$FF84_H$	$C2_H$	CAPCOM Register 6 Interrupt Control Register	0000_H
CC7	$FE8E_H$	47_H	CAPCOM Register 7	0000_H

Special Function Registers Overview (cont'd)

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

Special Function Registers Overview (cont'd)

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

Special Function Registers Overview (cont'd)

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

Special Function Registers Overview (cont'd)

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

*) The system configuration is selected during reset.

Absolute Maximum Ratings

Ambient temperature under bias (T_A):
SAB 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 (**C**ontroller **C**haracteristics):
The logic of the SAB 80C166W/83C166W will provide signals with the respective timing characteristics.

SR (**S**ystem **R**equirement):
The external system must provide signals with the respective timing characteristics to the SAB 80C166W/83C166W.

DC Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ˚C for SAB 83C166W-5M, SAB 80C166W/83C166W-M
T_A = – 40 to + 85 ˚C for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3
T_A = – 40 to + 110 ˚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 RSTIN and XTAL1)	V_{IH}	SR	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage RSTIN	V_{IH1}	SR	0.6 V_{CC}	V_{CC} + 0.5	V	–
Input high voltage XTAL1	V_{IH2}	SR	0.7 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage (Port 0, Port 1, Port 4, ALE, RD, WR, BHE, CLKOUT, RSTOUT)	V_{OL}	CC	–	0.45	V	I_{OL} = 2.4 mA
Output low voltage (all other outputs)	V_{OL1}	CC	–	0.45	V	I_{OL1} = 1.6 mA
Output high voltage (Port 0, Port 1, Port 4, ALE, RD, WR, BHE, CLKOUT, RSTOUT)	V_{OH}	CC	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 500 µA I_{OH} = – 2.4 mA
Output high voltage (all other outputs)	V_{OH1}	CC	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 250 µA I_{OH} = – 1.6 mA
Input leakage current [1]	I_{OZ}	CC	–	± 1	µA	0 V < V_{IN} < V_{CC}
RSTIN pullup resistor	R_{RST}	CC	50	150	kΩ	–
Read inactive current [5]	I_{RH}	[3]	–	– 40	µA	V_{OUT} = V_{OHmin}
Read active current [5]	I_{RL}	[4]	– 500	–	µA	V_{OUT} = V_{OLmax}
ALE inactive current [5]	I_{ALEL}	[3]	–	150	µA	V_{OUT} = V_{OLmax}
ALE active current [5]	I_{ALEH}	[4]	2100	–	µA	V_{OUT} = V_{OHmin}
XTAL1 input current	I_{IL}	CC	–	± 20	µA	0 V < V_{IN} < V_{CC}
Pin capacitance [6] (digital inputs/outputs)	C_{IO}	CC	–	10	pF	f = 1 MHz T_A = 25 ˚C
Power supply current	I_{CC}		–	50 + 5 x f_{CPU}	mA	Reset active f_{CPU} in [MHz] [7]
Idle mode supply current	I_{ID}		–	30 + 1.5 x f_{CPU}	mA	f_{CPU} in [MHz] [7]
Power-down mode supply current	I_{PD}		–	50	µA	V_{CC} = 5.5 V [8]

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.

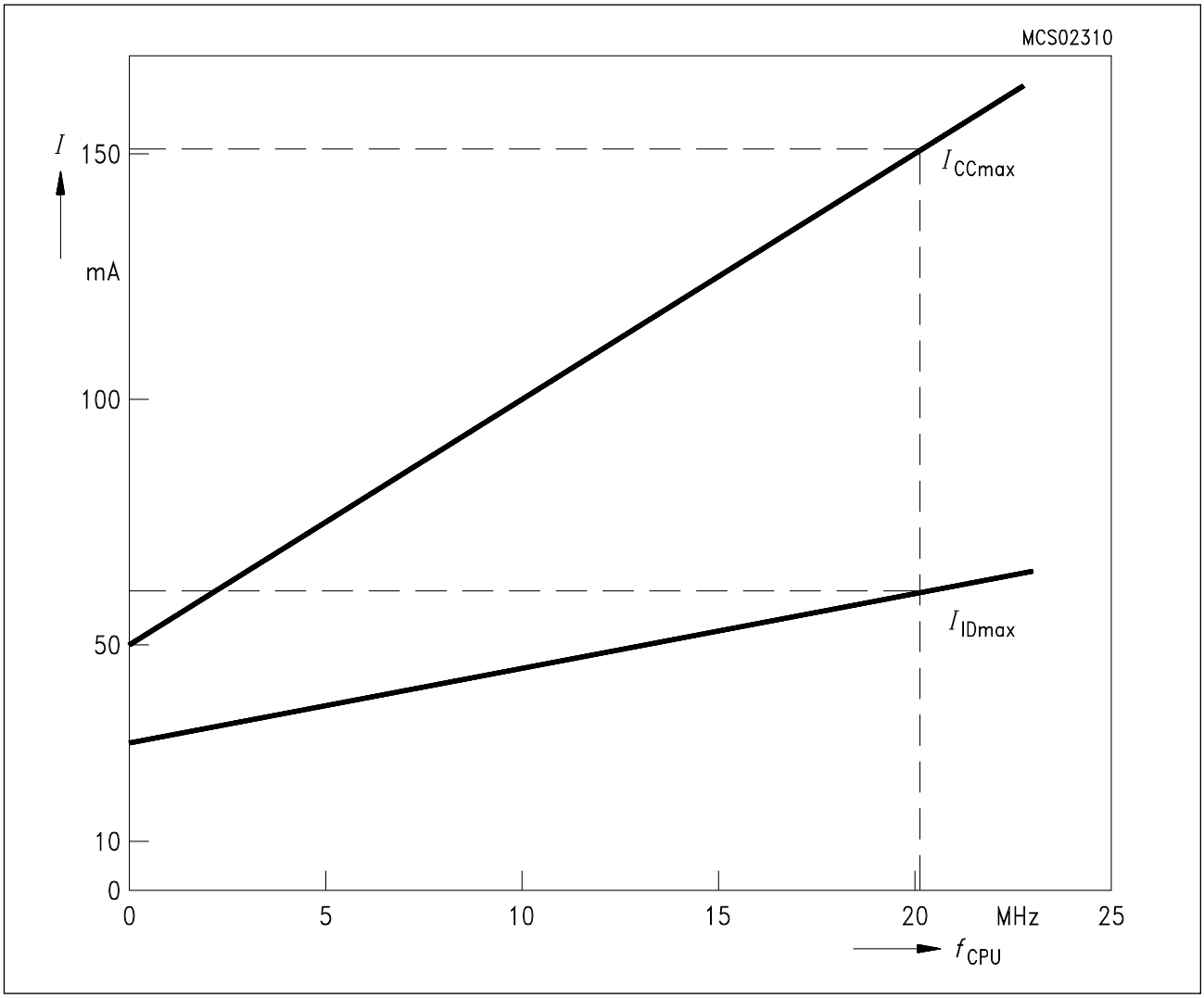


Figure 8
Supply/Idle Current as a Function of Operating Frequency

A/D Converter Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ˚C for SAB 83C166W-5M, SAB 80C166W/83C166W-M
T_A = – 40 to + 85 ˚C for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3
T_A = – 40 to + 110 ˚C for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4
4.0 V ≤ V_{AREF} ≤ V_{CC} + 0.1 V; V_{SS} – 0.1 V ≤ V_{AGND} ≤ V_{SS} + 0.2 V

Parameter	Symbol		Limit Values		Unit	Test Condition
			min.	max.		
Analog input voltage range	V_{AIN}	SR	V_{AGND}	V_{AREF}	V	1)
Sample time	t_S	CC	–	2 t_{SC}		2) 5)
Conversion time	t_C	CC	–	10 t_{CC} + t_S + 4TCL		3) 5)
Total unadjusted error	TUE	CC	–	± 2	LSB	V_{AREF} = V_{CC} V_{AGND} = V_{SS}
Internal resistance of reference voltage source	R_{AREF}	CC	–	t_{CC} / 250 - 0.25	kΩ	t_{CC} in [ns] 4)
Internal resistance of analog source	R_{ASRC}	CC	–	t_S / 500 - 0.25	kΩ	t_S in [ns] 4)

Notes

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

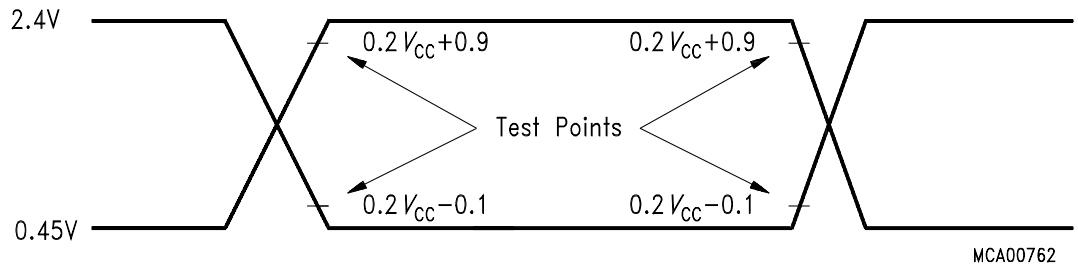
2) During the sample time the input capacitance C_I can be charged/discharged by the external source. The internal resistance of the analog source must allow the capacitors to reach their final voltage level within t_S. After the end of the sample time t_S, changes of the analog input voltage have no effect on the conversion result. The value for the sample clock is t_{SC} = T_{CL} x 32.

3) This parameter includes the sample time t_S, the time for determining the digital result and the time to load the result register with the conversion result.
The value for the conversion clock is t_{CC} = T_{CL} x 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.

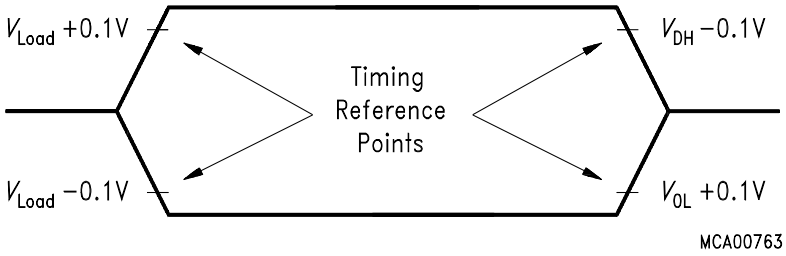
5) This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.

Testing Waveforms



AC inputs during testing are driven at 2.4 V for a logic ‘1’ and 0.4 V for a logic ‘0’. Timing measurements are made at V_{IH} min for a logic ‘1’ and V_{IL} max for a logic ‘0’.

Figure 9
Input Output Waveforms



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

Figure 10
Float Waveforms

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} x <ALECTL>
Memory Cycle Time Waitstates	t_C	$2T_{CL}$ x (15 - <MCTC>)
Memory Tristate Time	t_F	$2T_{CL}$ x (1 - <MTTC>)

AC Characteristics

External Clock Drive XTAL1

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ˚C for SAB 83C166W-5M, SAB 80C166W/83C166W-M
T_A = – 40 to + 85 ˚C for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3
T_A = – 40 to + 110 ˚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 to 20 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} x DC_{min}	T_{CLP} x 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.

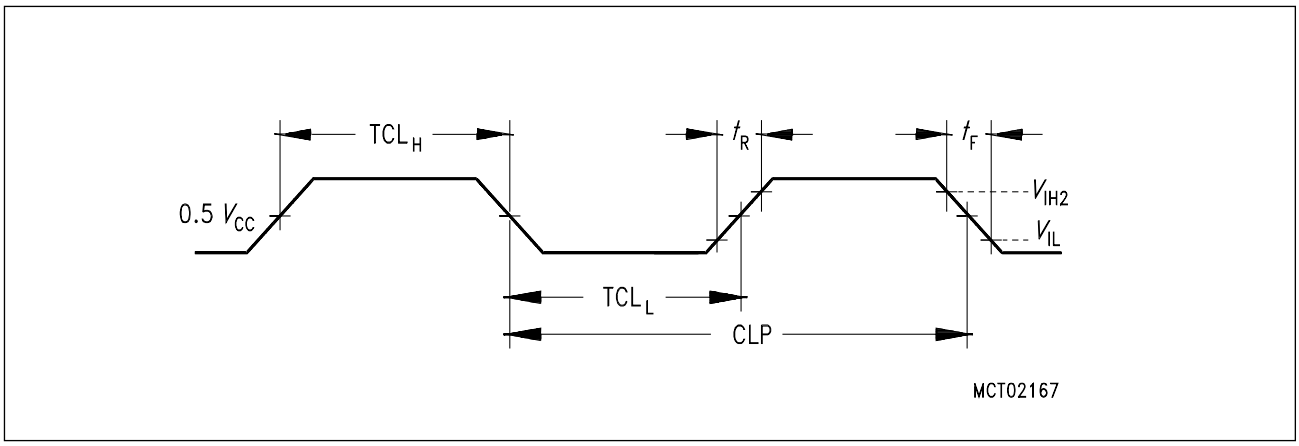


Figure 11
External Clock Drive XTAL1

AC Characteristics (cont'd)

Multiplexed Bus

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 83C166W-5M, SAB 80C166W/83C166W-M
T_A = – 40 to + 85 °C for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3
T_A = – 40 to +110 °C for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, 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 to 20 MHz		Unit
			min.	max.	min.	max.	
ALE high time	t_5	CC	15 + t_A	–	T_{CL}min - 10 + t_A	–	ns
Address setup to ALE	t_6	CC	10 + t_A	–	T_{CL}min - 15 + t_A	–	ns
Address hold after ALE	t_7	CC	15 + t_A	–	T_{CL}min - 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8	CC	15 + t_A	–	T_{CL}min - 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9	CC	-10 + t_A	–	-10 + t_A	–	ns
Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_{10}	CC	–	5	–	5	ns
Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_{11}	CC	–	42.5	–	T_{CL}max + 5	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}	CC	52.5 + t_C	–	T_{CLP} - 10 + t_C	–	ns
$\overline{RD}$ $\overline{WR}$ low time (no RW-delay)	t_{13}	CC	77.5 + t_C	–	T_{CLP} + T_{CL}min - 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}	SR	–	47.5 + t_C	–	T_{CLP} - 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}	SR	–	72.5 + t_C	–	T_{CLP} + T_{CL}min - 20 + t_C	ns
ALE low to valid data in	t_{16}	SR	–	72.5 + t_A + t_C	–	T_{CLP} + T_{CL}min - 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 $\overline{RD}$ 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

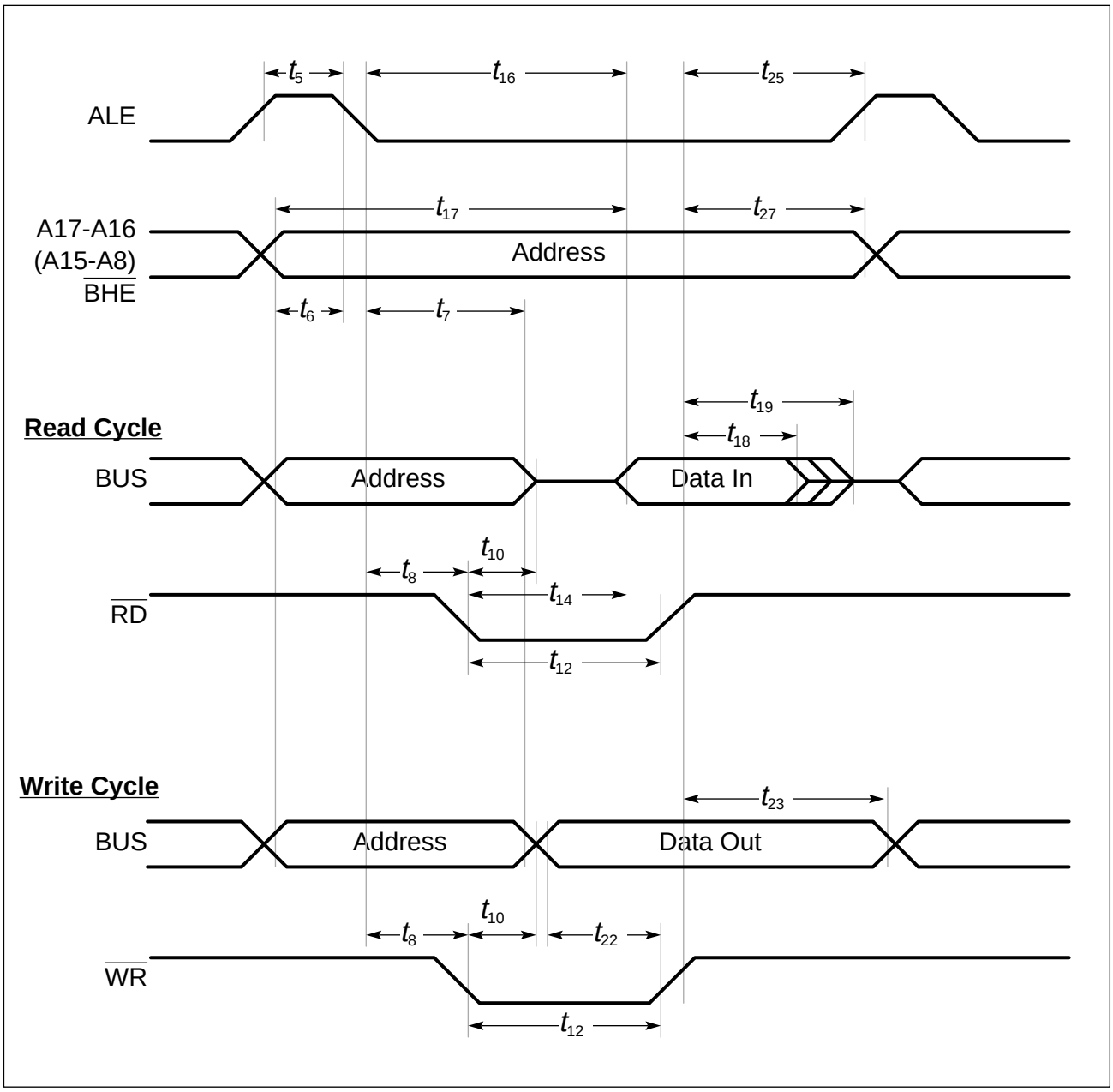


Figure 12-1
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE

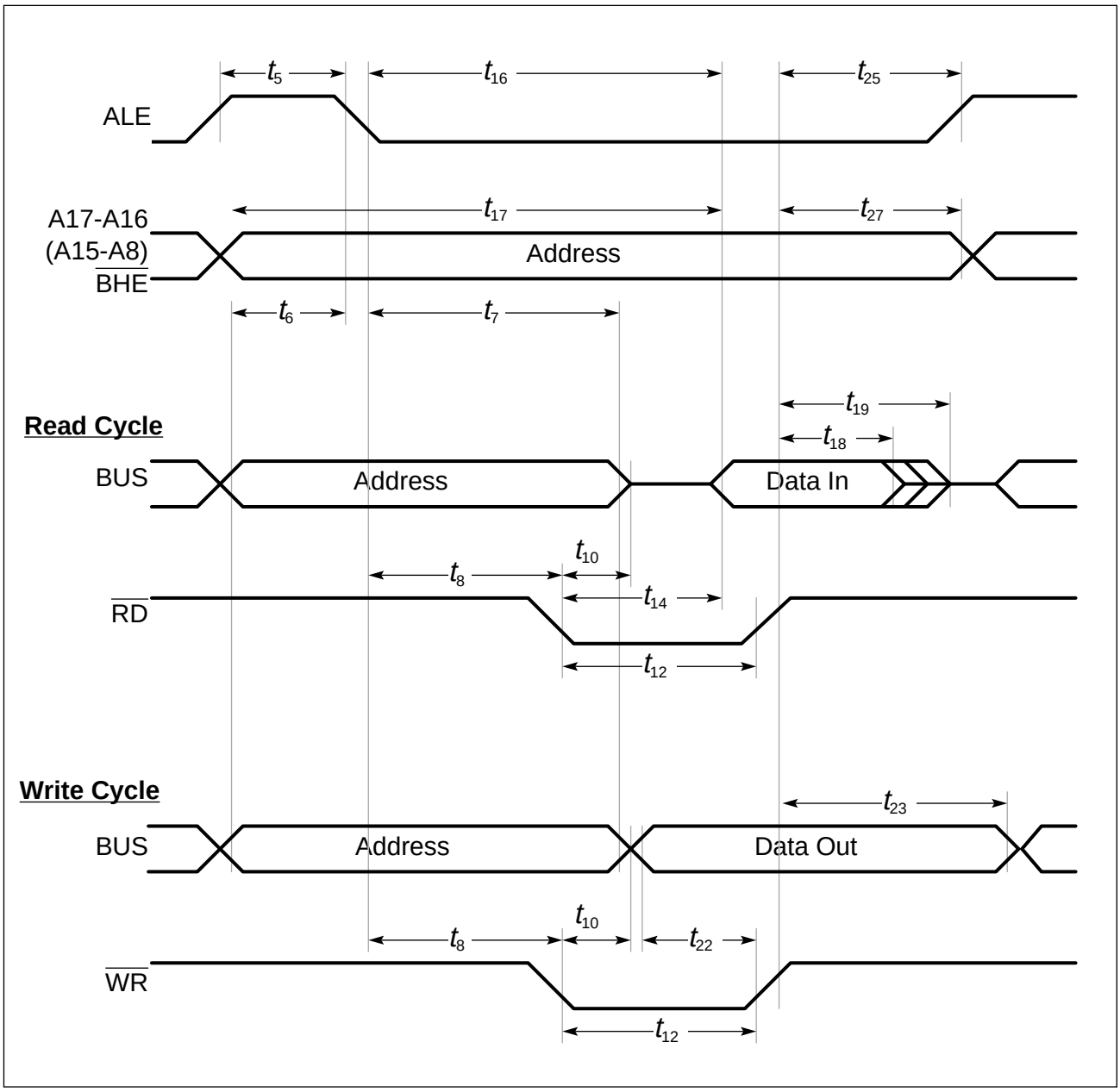


Figure 12-2
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Extended ALE

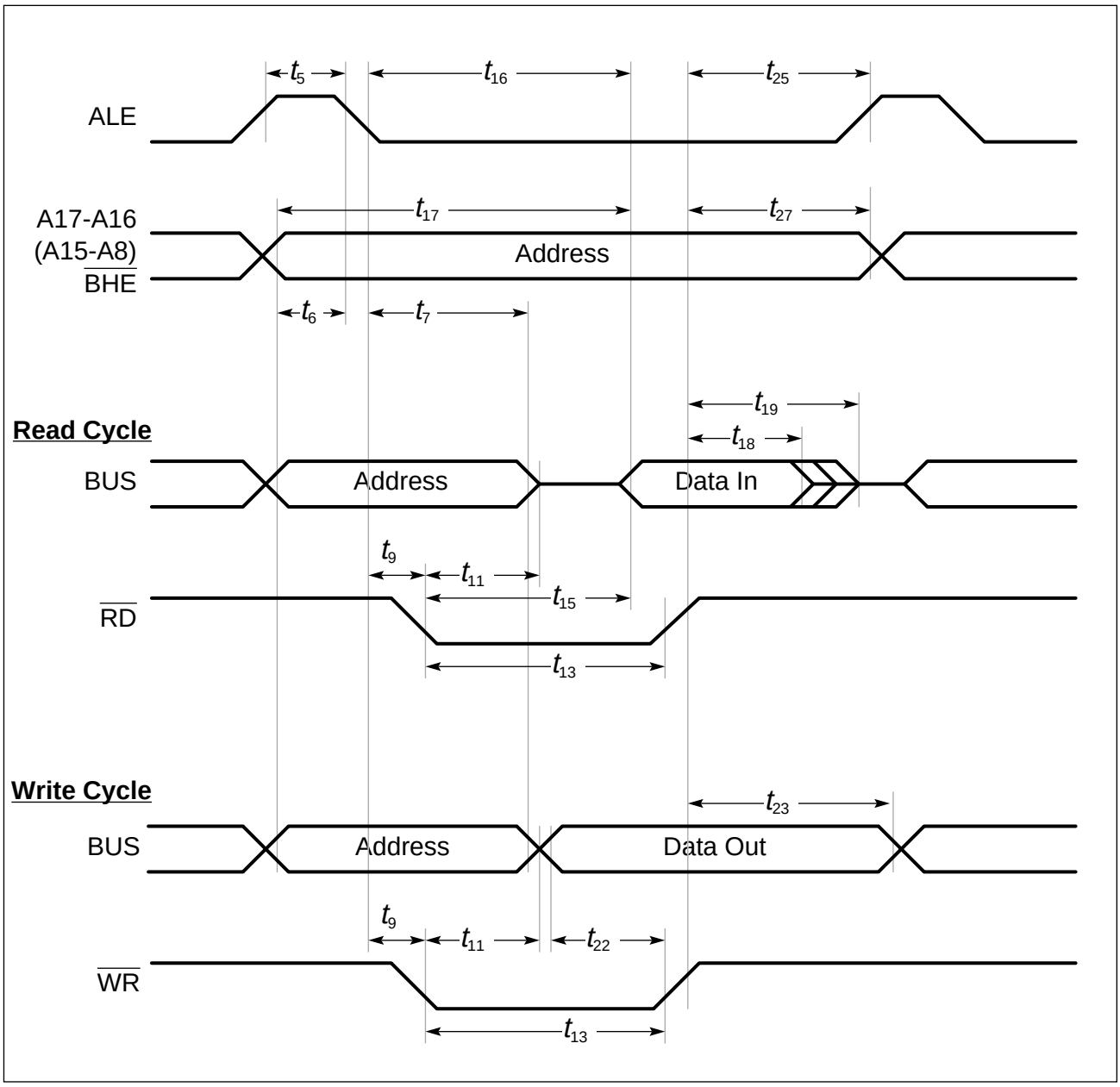


Figure 12-3
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE

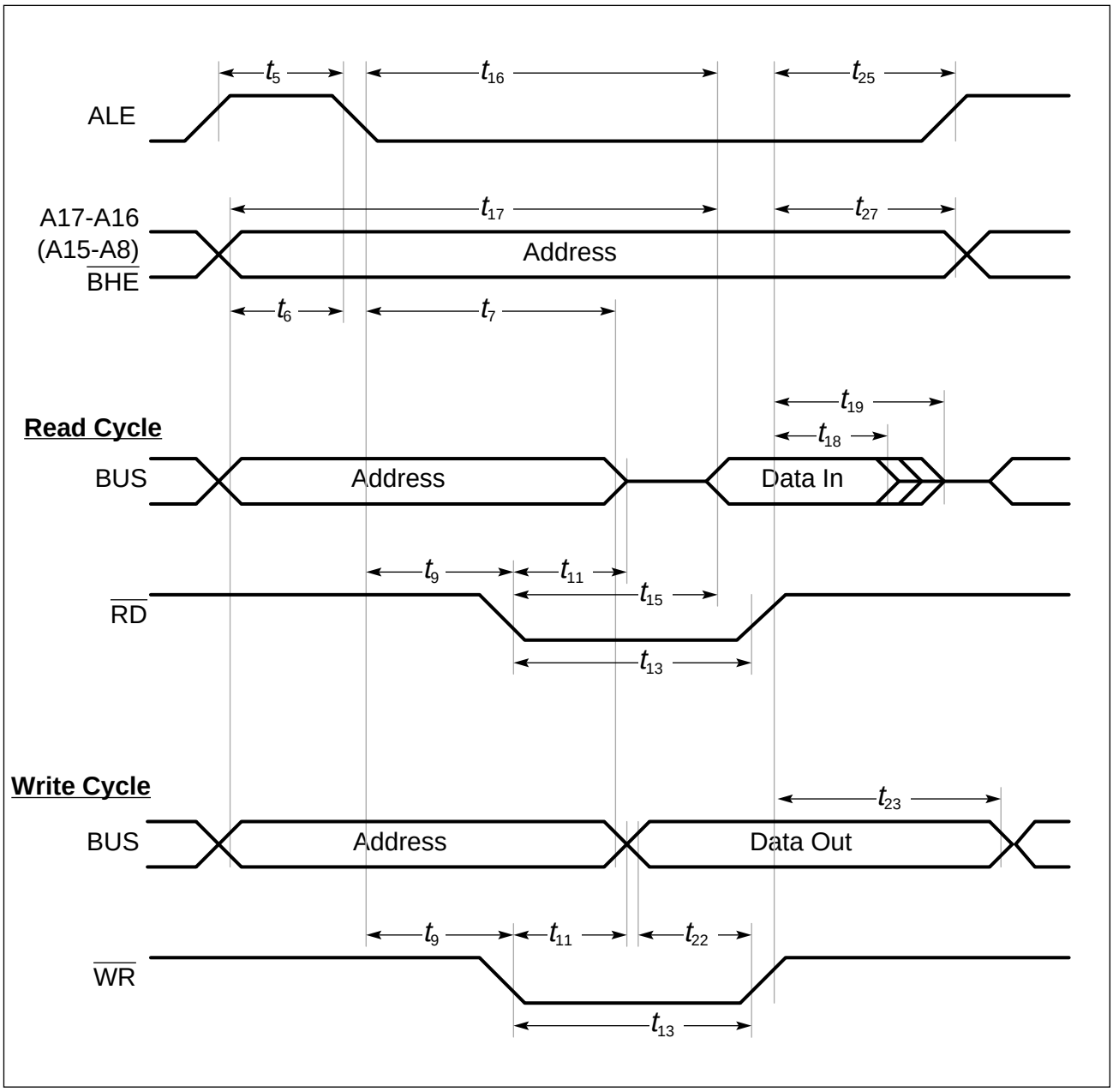


Figure 12-4
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

Demultiplexed Bus

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 83C166W-5M, SAB 80C166W/83C166W-M
T_A = – 40 to + 85 °C for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3
T_A = – 40 to + 110 °C for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 4 TCL (100 ns at 20-MHz CPU clock)

Parameter	Symbol		CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock $1/T_{CLP}$ = 1 to 20 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
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8	CC	15 + t_A	–	T_{CLmin} - 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9	CC	-10 + t_A	–	-10 + t_A	–	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}	CC	52.5 + t_C	–	T_{CLP} - 10 + t_C	–	ns
$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	t_{13}	CC	77.5 + t_C	–	T_{CLP}+T_{CLmin} - 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}	SR	–	47.5 + t_C	–	T_{CLP} - 20 + t_C	ns
$\overline{RD}$ 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 + 2 t_A + t_C	–	2T_{CLP} - 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}	SR	0	–	0	–	ns
Data float after $\overline{RD}$ rising edge (with RW-delay)	t_{20}	SR	–	47.5 + t_F	–	T_{CLP} - 15 + t_F	ns
Data float after $\overline{RD}$ rising edge (no RW-delay)	t_{21}	SR	–	15 + t_F	–	T_{CLmin} - 10 + t_F	ns
Data valid to $\overline{WR}$	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

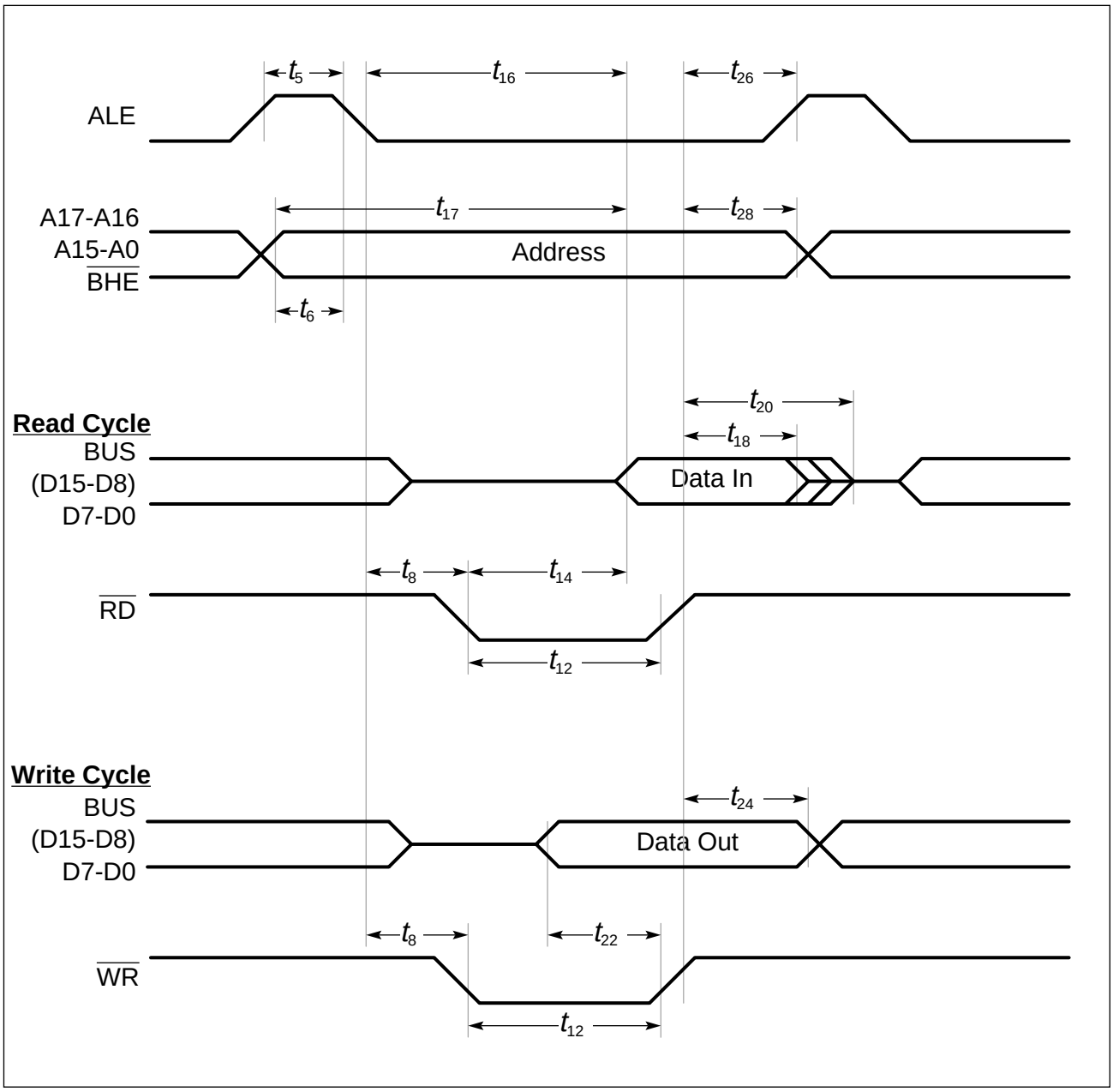


Figure 13-1
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Normal ALE

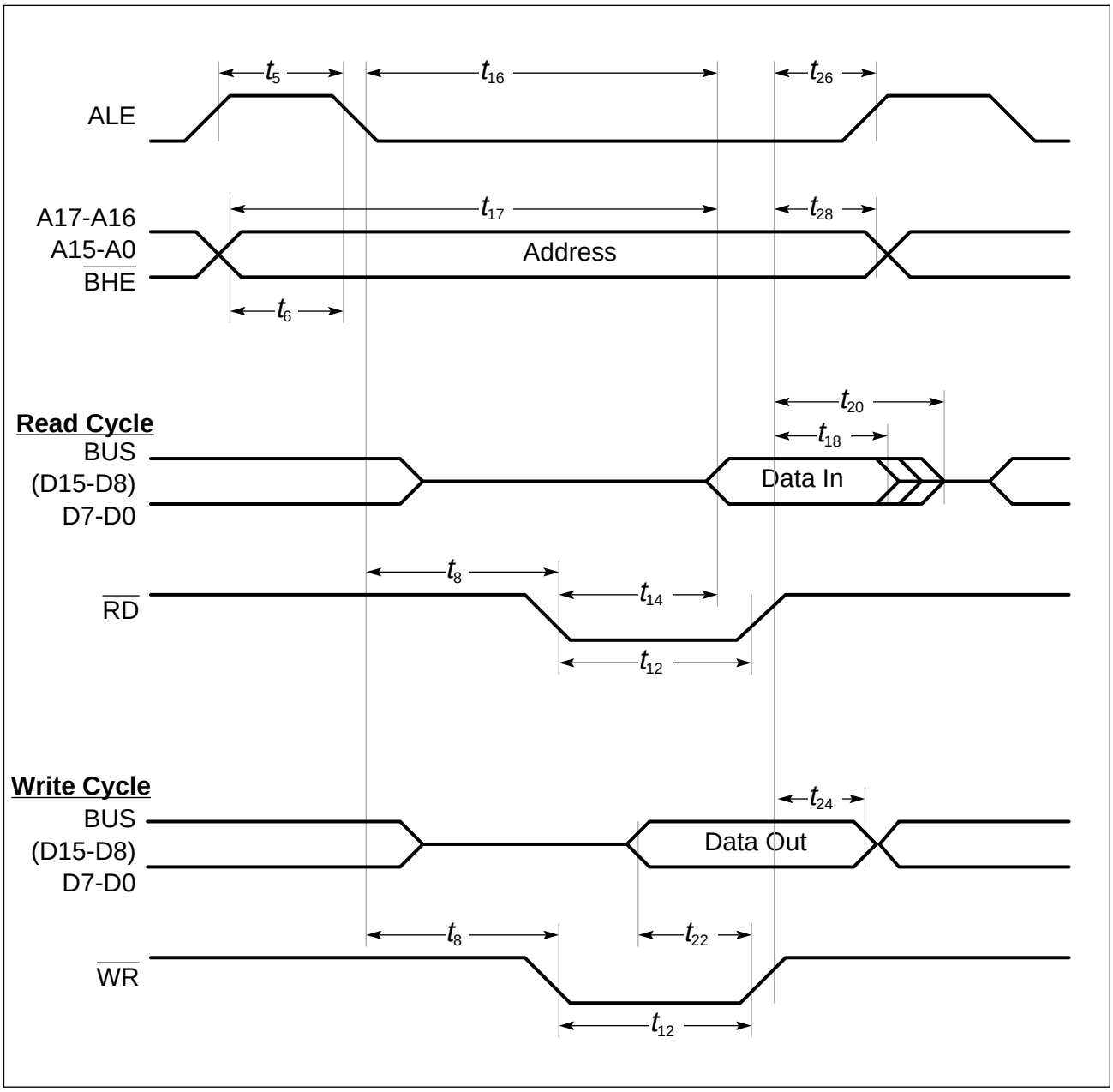


Figure 13-2
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE

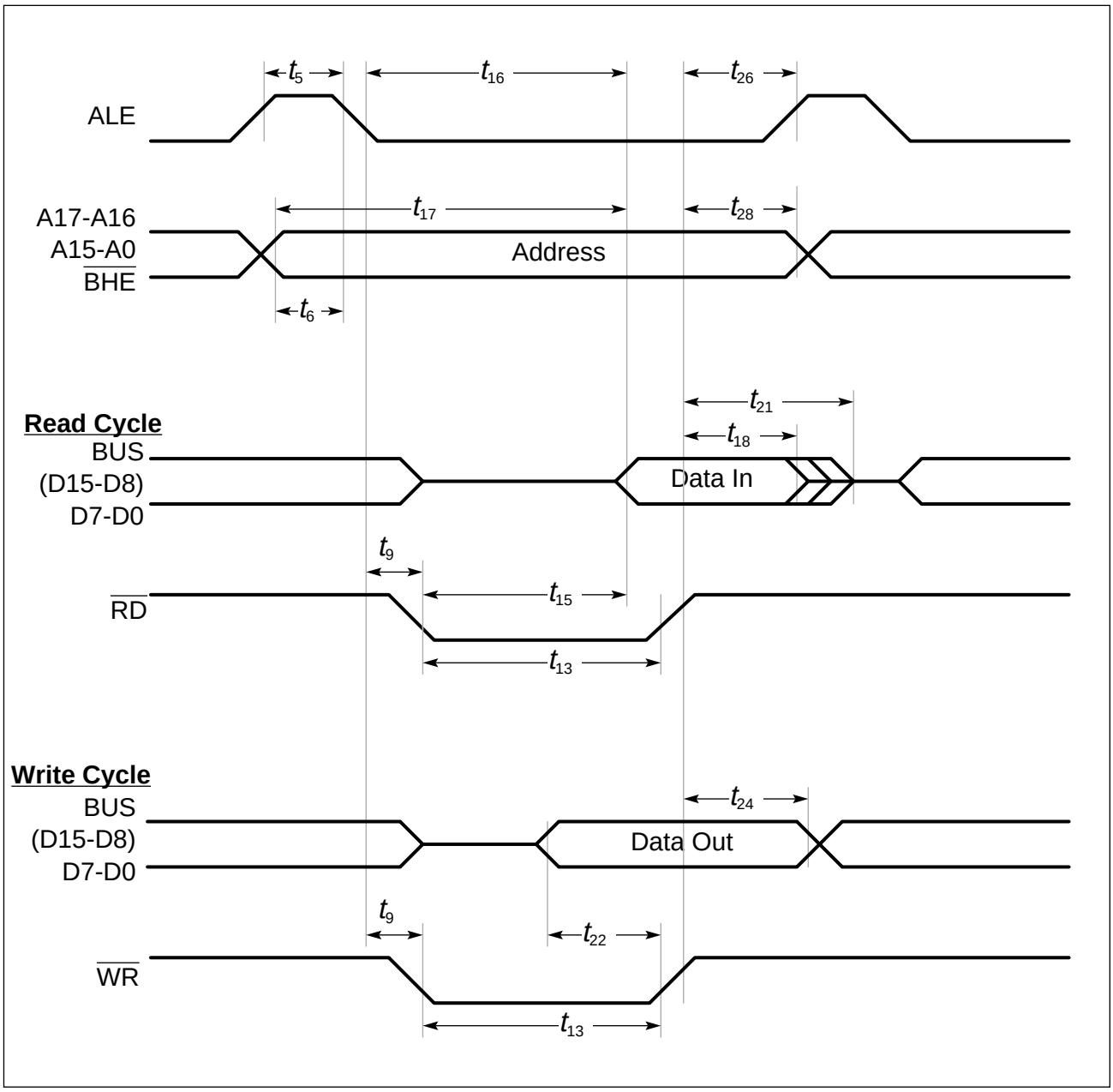


Figure 13-3
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE

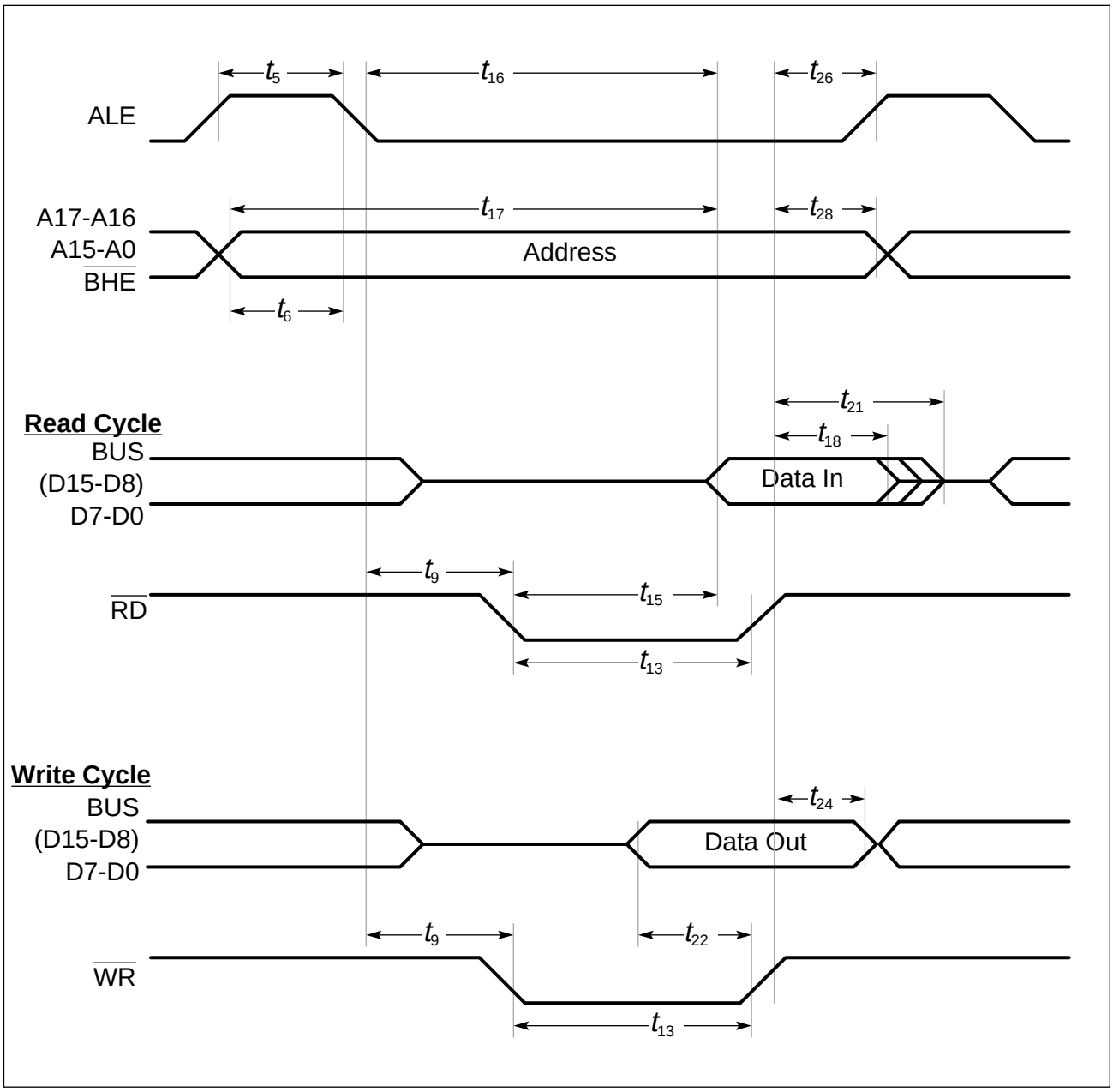


Figure 13-4
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

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

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 83C166W-5M, SAB 80C166W/83C166W-M
T_A = – 40 to + 85 °C for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3
T_A = – 40 to + 110 °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

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.	
CLKOUT cycle time	t_{29}	CC	62.5	62.5	T_{CLP}	T_{CLP}	ns
CLKOUT high time	t_{30}	CC	15	–	T_{CLmin} – 10	–	ns
CLKOUT low time	t_{31}	CC	15	–	T_{CLmin} – 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_{CLP} + 15	–	ns
Asynchronous $\overline{\text{READY}}$ setup time [1]	t_{58}	SR	20	–	20	–	ns
Asynchronous $\overline{\text{READY}}$ hold time [1]	t_{59}	SR	0	–	0	–	ns
Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus) [2]	t_{60}	SR	0	0 + 2t_A + t_F [2]	0	T_{CL} - 25 + 2t_A + t_F [2]	ns

Notes

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

2) Demultiplexed bus is the worst case. For multiplexed bus 2T_{CL} are to be added to the maximum values. This adds even more time for deactivating $\overline{\text{READY}}$.

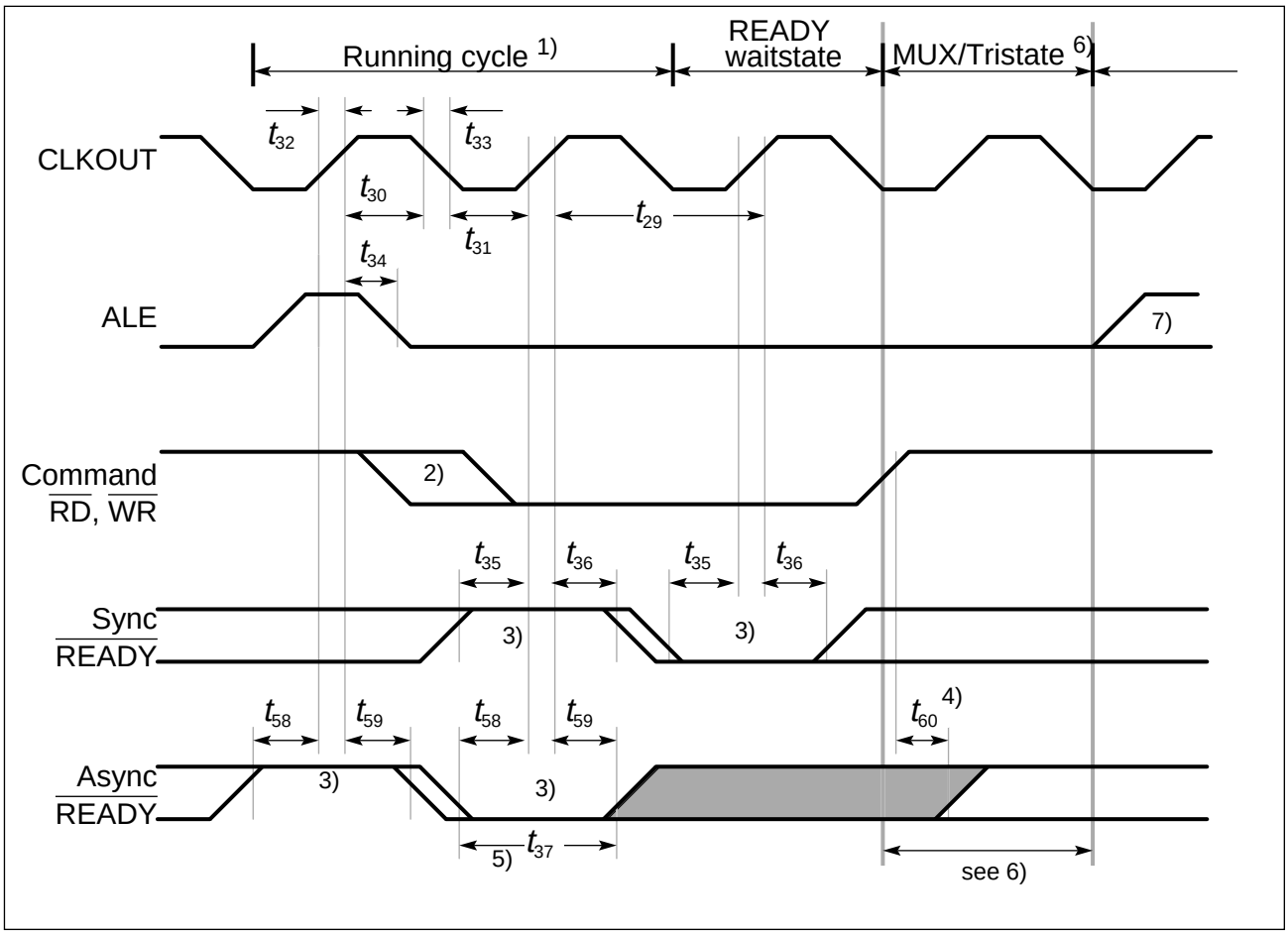


Figure 14
CLKOUT and READY

Notes

1) Cycle as programmed, including MCTC waitstates (Example shows 0 MCTC WS).

2) The leading edge of the respective command depends on RW-delay.

3) READY sampled HIGH at this sampling point generates a READY controlled waitstate,
READY sampled LOW at this sampling point terminates the currently running bus cycle.

4) READY may be deactivated in response to the trailing (rising) edge of the corresponding command (RD or WR).

5) If the Asynchronous READY signal does not fulfill the indicated setup and hold times with respect to CLKOUT (eg. because CLKOUT is not enabled), it must fulfill t_{37} in order to be safely synchronized. This is guaranteed, if READY is removed in reponse to the command (see Note 4)).

6) Multiplexed bus modes have a MUX waitstate added after a bus cycle, and an additional MTTC waitstate may be inserted here.
For a multiplexed bus with MTTC waitstate this delay is 2 CLKOUT cycles, for a demultiplexed bus without MTTC waitstate this delay is zero.

7) The next external bus cycle may start here.

AC Characteristics (cont'd)
External Bus Arbitration

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 83C166W-5M, SAB 80C166W/83C166W-M
T_A = – 40 to + 85 °C for SAB 83C166W-5M-T3, SAB 80C166W/83C166W-M-T3
T_A = – 40 to + 110 °C for SAB 83C166W-5M-T4, SAB 80C166W/83C166W-M-T4
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF

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.	
$\overline{HOLD}$ input setup time to CLKOUT	t_{61}	SR	20	–	20	–	ns
CLKOUT to $\overline{HLDA}$ high or $\overline{BREQ}$ low delay	t_{62}	CC	–	50	–	50	ns
CLKOUT to $\overline{HLDA}$ low or $\overline{BREQ}$ high delay	t_{63}	CC	–	60	–	60	ns
Other signals release	t_{66}	CC	–	25	–	25	ns
Other signals drive	t_{67}	CC	–	60	–	60	ns

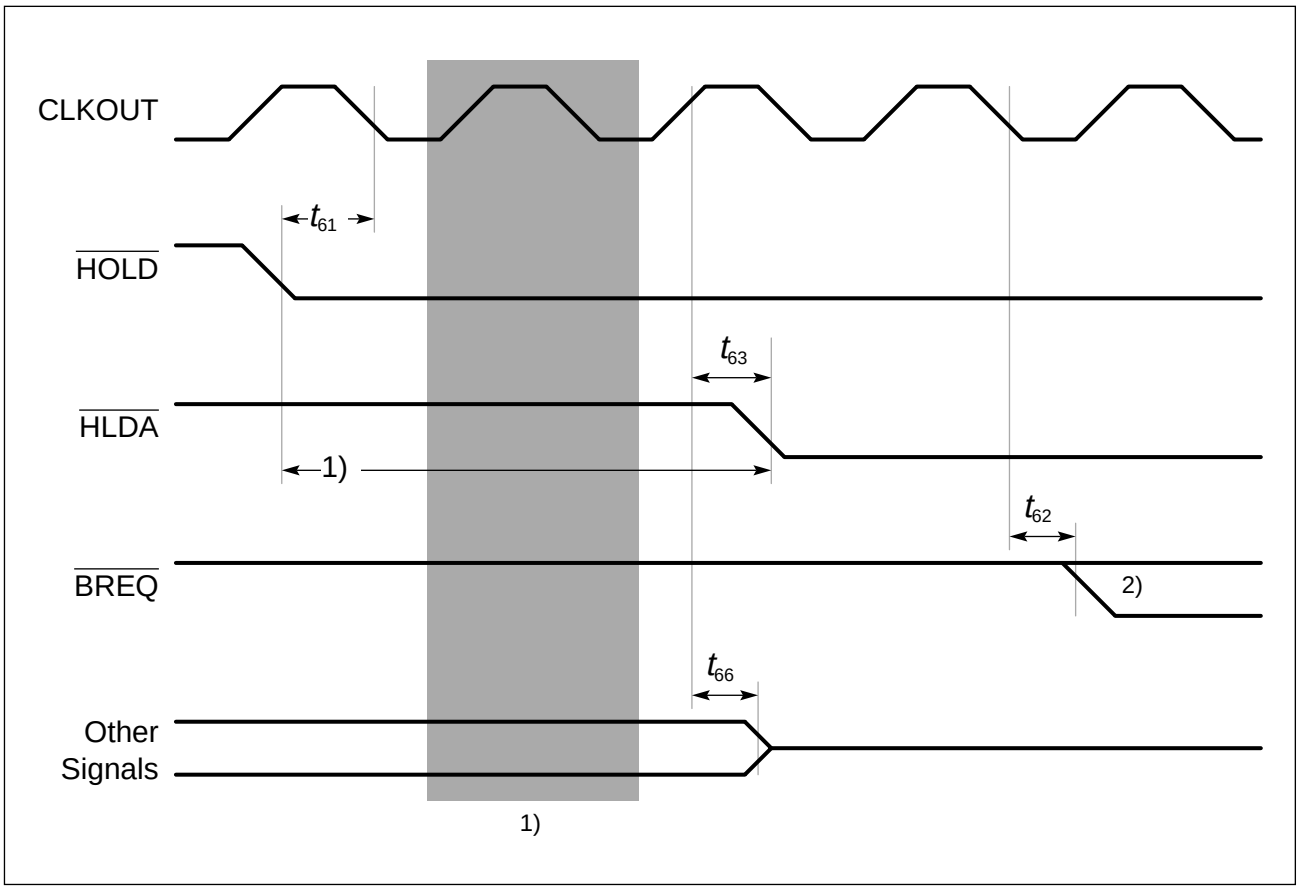


Figure 15
External Bus Arbitration, Releasing the Bus

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.

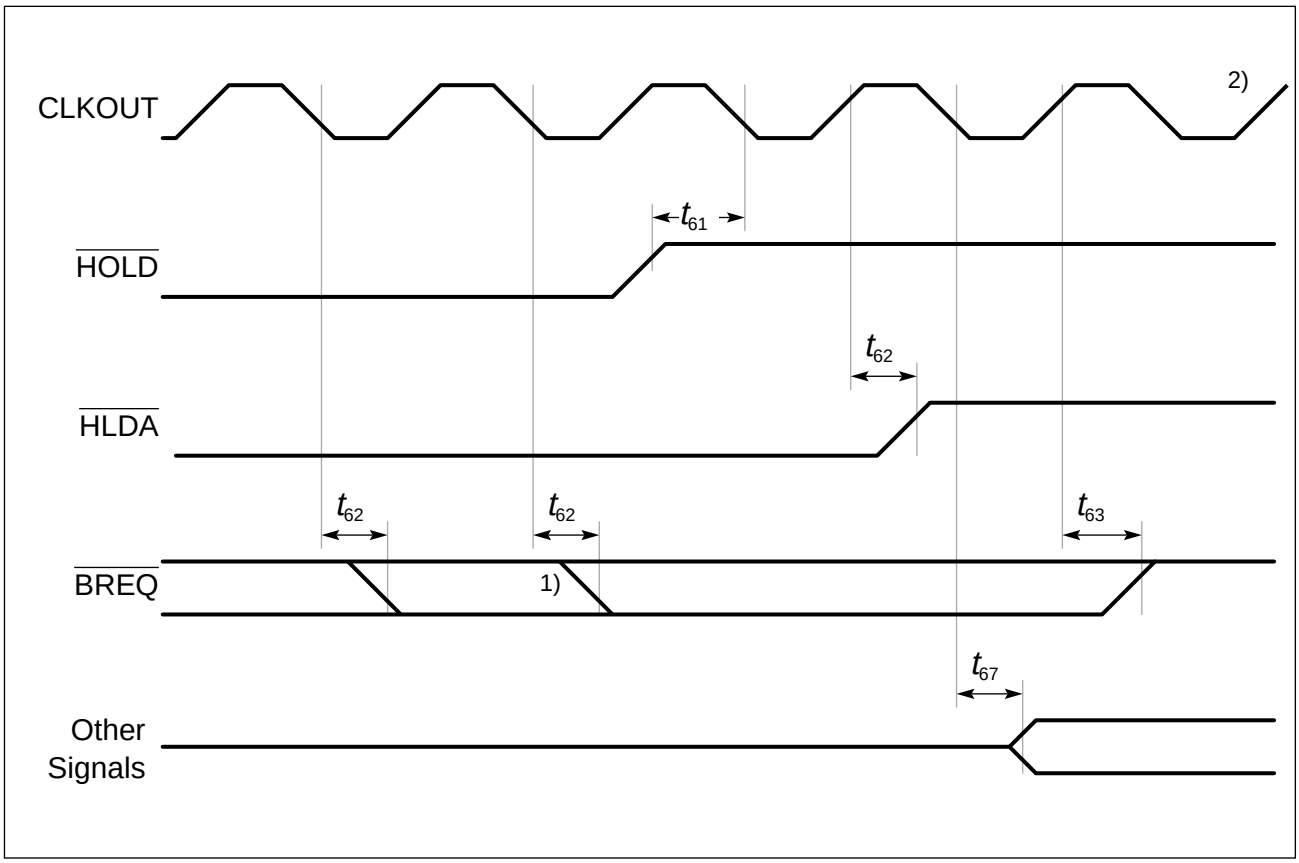


Figure 16
External Bus Arbitration, (Regaining the Bus)

Notes

1) This is the last chance for BREQ to trigger the indicated regain-sequence.
Even if BREQ is activated earlier, the regain-sequence is initiated by HOLD going high.
Please note that HOLD may also be deactivated without the SAB 80C166W/83C166W requesting the bus.

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