SIEMENS

Microcomputer Components

16-Bit CMOS Single-Chip Microcontrollers
with/without oscillator prescaler

with 32 KByte Flash EPROM

SAB 88C166/88C166W

Data Sheet 05.94

SIEMENS

C16x-Family of High-Performance CMOS 16-Bit Microcontrollers **SAB 88C166(W)**

Preliminary

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

- High Performance 16-bit CPU with 4-Stage Pipeline
- 100 ns Instruction Cycle Time at 20 MHz CPU Clock
- 500 ns Multiplication (16 × 16 bit), 1 µs Division (32 / 16 bit)
- Enhanced Boolean Bit Manipulation Facilities
- Register-Based Design with Multiple Variable Register Banks
- Single-Cycle Context Switching Support
- Up to 256 KBytes Linear Address Space for Code and Data
- 1 KByte On-Chip RAM
- 32 KBytes On-Chip Flash EPROM with Bank Erase Feature
- Read-Protectable Flash Memory
- Dedicated Flash Control Register with Operation Lock Mechanism
- 12 V External Flash Programming Voltage
- Flash Program Verify and Erase Verify Modes
- 100 Flash Program/Erase Cycles guaranteed
- Programmable External Bus Characteristics for Different Address Ranges
- 8-Bit or 16-Bit External Data Bus
- Multiplexed or Demultiplexed External Address/Data Buses
- Hold and Hold-Acknowledge Bus Arbitration Support
- 512 Bytes On-Chip Special Function Register Area
- Idle and Power Down Modes
- 8-Channel Interrupt-Driven Single-Cycle Data Transfer Facilities via Peripheral Event Controller (PEC)
- 16-Priority-Level Interrupt System
- 10-Channel 10-bit A/D Converter with 9.7 µs Conversion Time
- 16-Channel Capture/Compare Unit
- Two Multi-Functional General Purpose Timer Units with 5 Timers
- Two Serial Channels (USARTs)
- Programmable Watchdog Timer
- Up to 76 General Purpose I/O Lines
- Direct clock input without prescaler in the SAB 88C166W (SAB 88C166 with prescaler)
- Supported by a Wealth of Development Tools like C-Compilers, Macro-Assembler Packages, Emulators, Evaluation Boards, HLL-Debuggers, Simulators, Logic Analyzer Disassemblers, Programming Boards
- On-Chip Bootstrap Loader
- 100-Pin Plastic MQFP Package (EIAJ)

Introduction

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

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

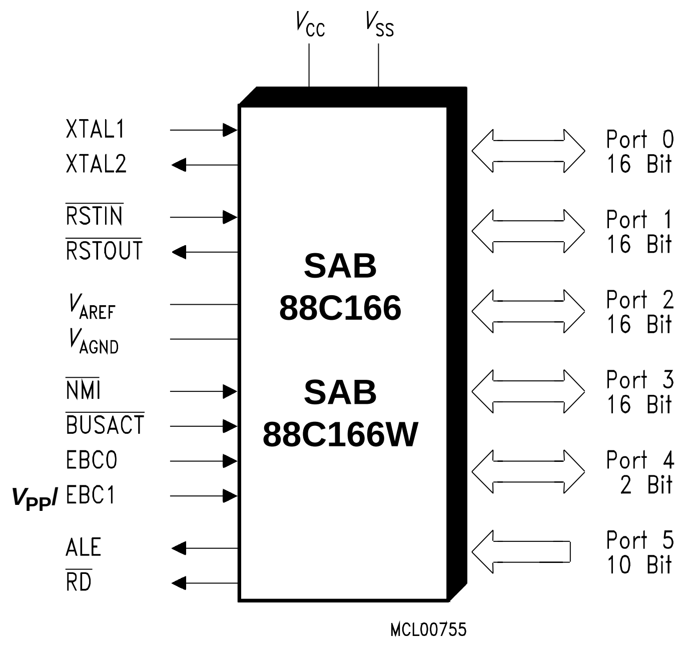


Figure 1
Logic Symbol

Ordering Information

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

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

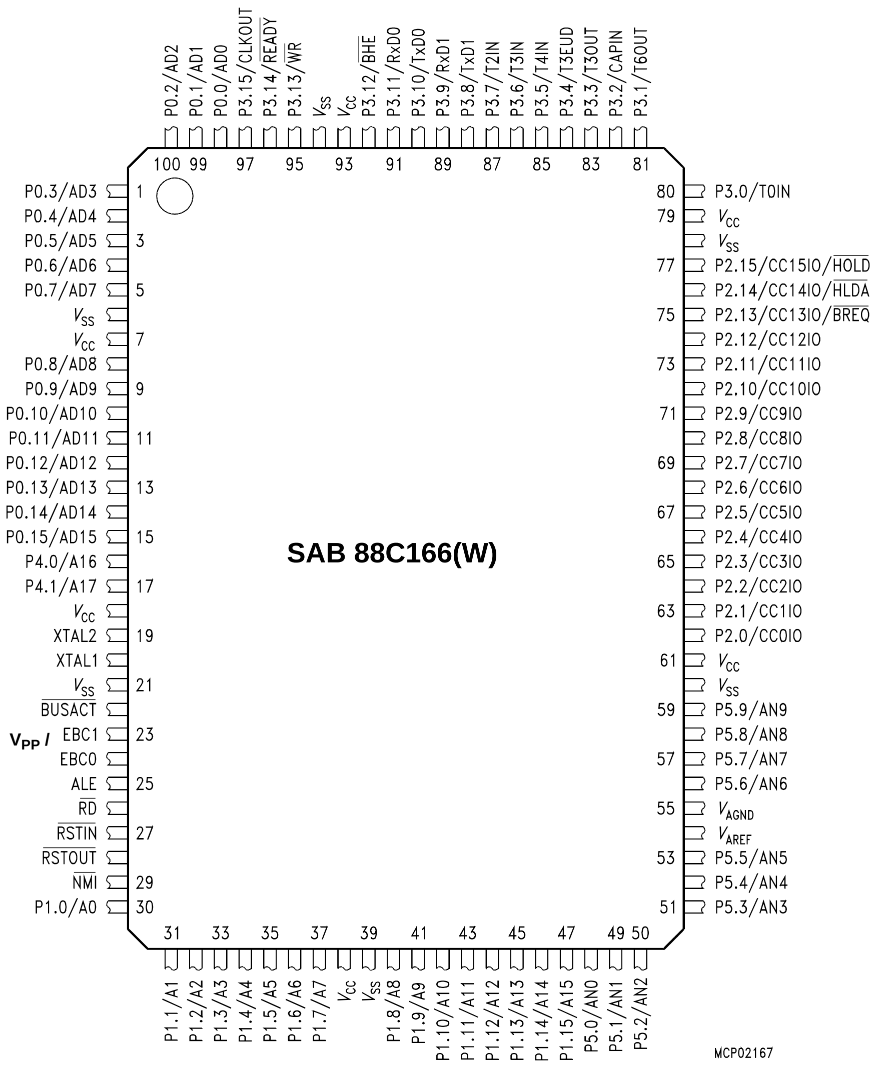


Figure 2

Pin Definitions and Functions

Symbol	Pin Number	Input (I) Output (O)	Function
P4.0 – P4.1	16 - 17	I/O	Port 4 is a 2-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. In case of an external bus configuration, Port 4 can be used to output the segment address lines:
	16	O	P4.0 A16 Least Significant Segment Addr. Line
	17	O	P4.1 A17 Most Significant Segment Addr. Line
XTAL1	20	I	XTAL1: Input to the oscillator amplifier and input to the internal clock generator
XTAL2	19	O	XTAL2: Output of the oscillator amplifier circuit. To clock the device from an external source, drive XTAL1, while leaving XTAL2 unconnected. Minimum and maximum high/low and rise/fall times specified in the AC Characteristics must be observed.
$\overline{\text{BUSACT}}$, EBC1, EBC0	22 23 24	I I I	External Bus Configuration selection inputs. These pins are sampled during reset and select either the single chip mode or one of the four external bus configurations: $\overline{\text{BUSACT}}$ EBC1 EBC0 Mode/Bus Configuration 0 0 0 8-bit demultiplexed bus 0 0 1 8-bit multiplexed bus 0 1 0 16-bit muliplexed bus 0 1 1 16-bit demultiplexed bus 1 0 0 Single chip mode 1 0 1 Reserved. 1 1 0 Reserved. 1 1 1 Reserved. After reset pin EBC1 accepts the programming voltage for the Flash EPROM as an “alternate function”:
V_{PP}	23		Flash EPROM Programming Voltage V_{PP} = 12 V.
$\overline{\text{RSTIN}}$	27	I	Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the SAB 88C166(W). An internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
$\overline{\text{RSTOUT}}$	28	O	Internal Reset Indication Output. This pin is set to a low level when the part is executing either a hardware-, a software- or a watchdog timer reset. $\overline{\text{RSTOUT}}$ remains low until the EINIT (end of initialization) instruction is executed.

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input (I) Output (O)	Function
$\overline{\text{NMI}}$	29	I	Non-Maskable Interrupt Input. A high to low transition at this pin causes the CPU to vector to the NMI trap routine. When the PWRDN (power down) instruction is executed, the $\overline{\text{NMI}}$ pin must be low in order to force the SAB 88C166(W) to go into power down mode. If $\overline{\text{NMI}}$ is high, when PWRDN is executed, the part will continue to run in normal mode. If not used, pull $\overline{\text{NMI}}$ high externally.
ALE	25	O	Address Latch Enable Output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.
$\overline{\text{RD}}$	26	O	External Memory Read Strobe. $\overline{\text{RD}}$ is activated for every external instruction or data read access.
P1.0 – P1.15	30 - 37 40 - 47	I/O	Port 1 is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. Port 1 is used as the 16-bit address bus (A) in demultiplexed bus modes and also after switching from a demultiplexed bus mode to a multiplexed bus mode..
P5.0 – P5.9	48 – 53 56 – 59	I I	Port 5 is a 10-bit input-only port with Schmitt-Trigger characteristics. The pins of Port 5 also serve as the (up to 10) analog input channels for the A/D converter, where P5.x equals ANx (Analog input channel x).
P2.0 – P2.15	62 – 77	I/O	Port 2 is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. The following Port 2 pins also serve for alternate functions:
	62	I/O	P2.0 CC0IO CAPCOM: CC0 Cap.-In/Comp.Out
	...		... ...
	75	I/O O	P2.13 CC13IO CAPCOM: CC13 Cap.-In/Comp.Out, $\overline{\text{BREQ}}$ External Bus Request Output
	76	I/O O	P2.14 CC14IO CAPCOM: CC14 Cap.-In/Comp.Out, $\overline{\text{HLDA}}$ External Bus Hold Acknowl. Output
	77	I/O I	P2.15 CC15IO CAPCOM: CC15 Cap.-In/Comp.Out, $\overline{\text{HOLD}}$ External Bus Hold Request Input

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input (I) Output (O)	Function		
P3.0 – P3.15	80 – 92, 95 – 97	I/O I/O	Port 3 is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. The following Port 3 pins also serve for alternate functions:		
	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	$\overline{\text{BHE}}$	Ext. Memory High Byte Enable Signal,
	95	O	P3.13	$\overline{\text{WR}}$	External Memory Write Strobe
	96	I	P3.14	$\overline{\text{READY}}$	Ready Signal Input
	97	O	P3.15	CLKOUT	System Clock Output (=CPU Clock)
P0.0 – P0.15	98 – 5 8 – 15	I/O	Port 0 is a 16-bit bidirectional IO port. It is bit-wise programmable for input or output via direction bits. For a pin configured as input, the output driver is put into high-impedance state. In case of an external bus configuration, Port 0 serves as the address (A) and address/data (AD) bus in multiplexed bus modes and as the data (D) bus in demultiplexed bus modes.		
			Demultiplexed bus modes:		
			Data Path Width:	8-bit	16-bit
			P0.0 – P0.7:	D0 – D7	D0 - D7
			P0.8 – P0.15:	output!	D8 - D15
			Multiplexed bus modes:		
			Data Path Width:	8-bit	16-bit
			P0.0 – P0.7:	AD0 – AD7	AD0 - AD7
			P0.8 – P0.15:	A8 - A15	AD8 - AD15
V_{AREF}	54	-	Reference voltage for the A/D converter.		
V_{AGND}	55	-	Reference ground for the A/D converter.		

Pin Definitions and Functions (cont'd)

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

Functional Description

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

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

Memory Organization

The memory space of the SAB 88C166(W) is configured in a Von Neumann architecture which means that code memory, data memory, registers and I/O ports are organized within the same linear address space which includes 256 KBytes. Address space expansion to 16 MBytes is provided for future versions. The entire memory space can be accessed bytewise or wordwise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

1 KByte of on-chip RAM is provided as a storage for user defined variables, for the system stack, general purpose register banks and even for code. A register bank can consist of up to 16 wordwide (R0 to R15) and/or bytewide (RL0, RH0, …, RL7, RH7) so-called General Purpose Registers (GPRs).

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

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

Flash Memory Overview

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

A separate Flash Control Register (FCR) has been implemented to control Flash operations like programming or erasure. For programming or erasing an external 12 V programming voltage must be applied to the VPP/EBC1 pin.

The Flash memory is organized in 8 K x 32 bits, which allows even double-word instructions to be fetched in just one machine cycle. The entire Flash memory is divided into four blocks with different sizes (12/12/6/2 KByte). This allows to erase each block separately, when only parts of the Flash memory need to be reprogrammed. Word or double word programming typically takes 100 µs, block erasing typically takes 1 s (@ 20 MHz CPU clock). The Flash memory features a typical endurance of 100 erasing/programming cycles. Erased Flash memory cells contain all ‘1’s, as known from standard EPROMs.

The Flash memory can be programmed both in an appropriate programming board and in the target system, which provides a lot of flexibility. The SAB 88C166(W)’s on-chip bootstrap loader may be used to load and start the programming code.

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

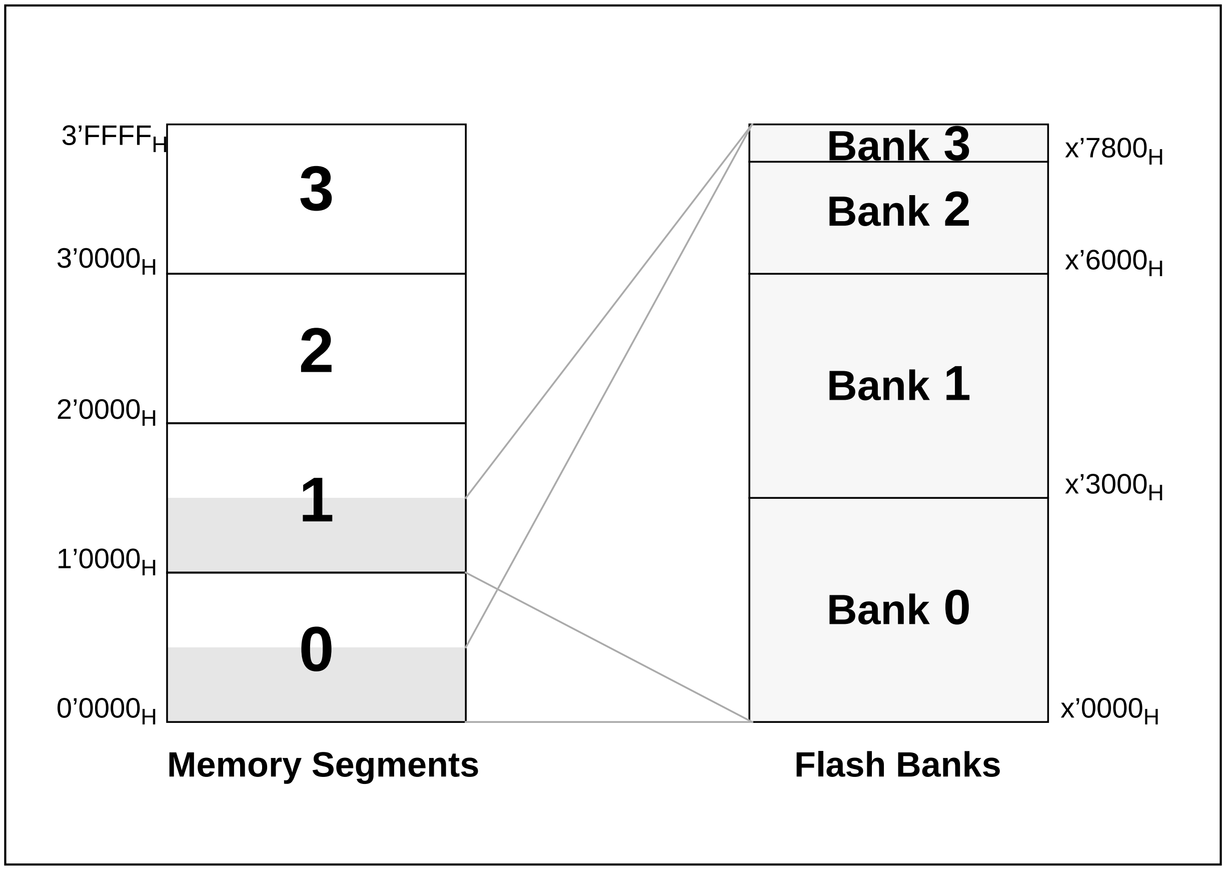


Figure 3
Flash Memory Overview

The Flash Control Register (FCR)

In standard operation mode the Flash memory can be accessed like the normal mask-programmable on-chip ROM of the SAB 83C166. So all appropriate direct and indirect addressing modes can be used for reading the Flash memory.

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

FCR (FFA0$_H$ / D0$_H$) SFR **Reset Value: 00X0$_H$*)**

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

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

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

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

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

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

FWE enables/disables write operations, FEE selects erasing or programming, FWMSET controls the writing mode. Bits FWE and FEE select an operation, but do not execute it directly.

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

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

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

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

FCVPP and VPPREV: These **read-only** bits allow to monitor the V_{PP} voltage. The Flash **Vpp Revelation** bit VPPREV reflects the state of the V_{PP} voltage in the Flash writing mode (VPPREV = ‘0’ indicates that V_{PP} is below the threshold value necessary for reliable programming or erasure, otherwise VPPREV = ‘1’). The Flash **Control** $\boldsymbol{V}_{PP}$ bit FCVPP indicates, if V_{PP} fell below the valid threshold value during a Flash programming or erase operation (FCVPP = ‘1’). FCVPP = ‘0’ after such an operation indicates that no critical discontinuity on $\boldsymbol{V}_{PP}$ has occurred.

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

Time Specification	Limit Value	
Maximum Programming Pulse Width	128	µs
Maximum Cumulated Programming Time	2.5	ms
Maximum Erase Pulse Width	10	ms
Maximum Cumulated Erase Time	30	s

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

Setting of CKCTL	Length of TPRG	TPRG @ f_{CPU} = 20 MHz	f_{CPUmin} for programming	f_{CPUmin} for erasing
0 0	2^7 * 1/f_{CPU}	6.4 µs	1 MHz	---
0 1	2^{11} * 1/f_{CPU}	102.4 µs	16 MHz	1 MHz
1 0	2^{15} * 1/f_{CPU}	1.64 ms	---	3.28 MHz
1 1	2^{18} * 1/f_{CPU}	13.11 ms	---	13.11 MHz

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

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

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

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

Operation Modes of the Flash Memory

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

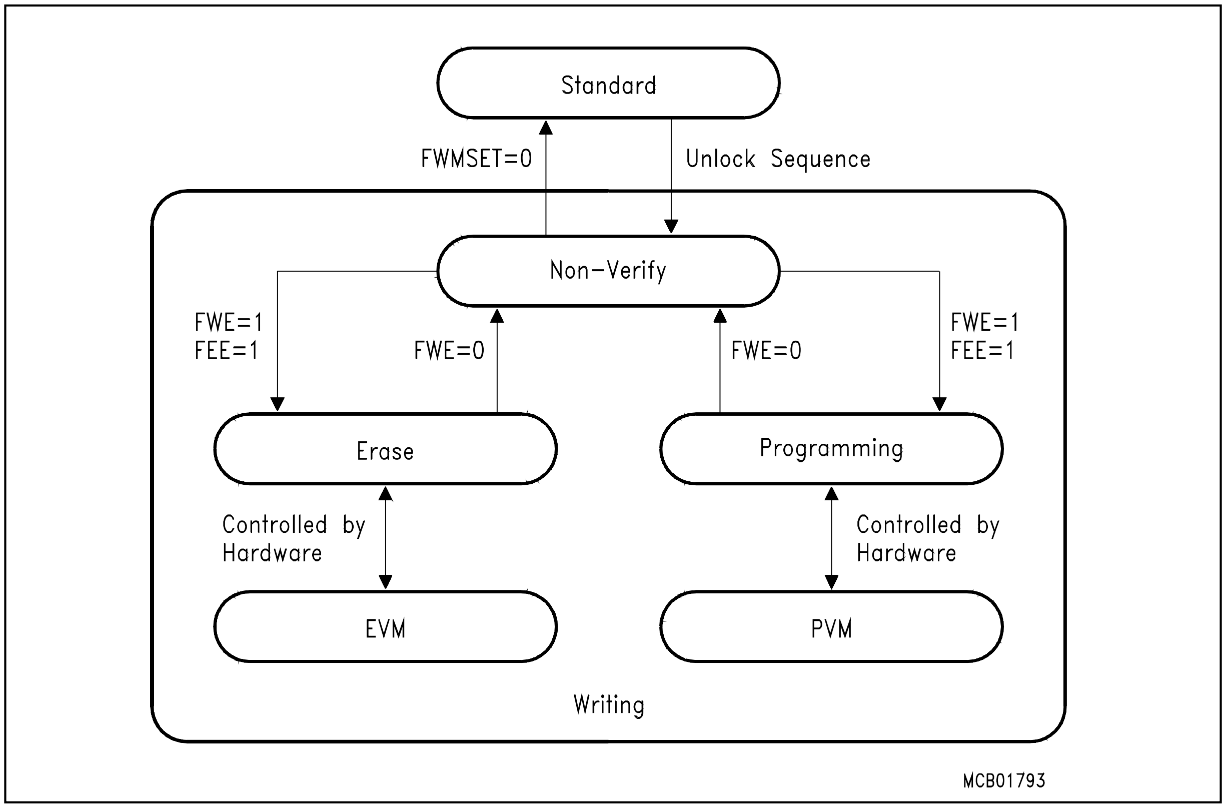


Figure 4
Flash Operating Mode Transitions

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

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

The Flash Writing Modes must be entered for programming or erasing the Flash memory. The SAB 88C166 enters these modes by a specific key code sequence, called UNLOCK sequence.

In writing mode the used addressing mode decides whether the FCR or a Flash memory location is accessed. The FCR can be accessed with any direct access to an even address in the active address space of the Flash memory. Only word operand instructions are allowed for FCR accesses. Accesses to Flash memory locations must use indirect addressing to even addresses.

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

After entering writing mode the first erase or programming operation must not be started for at least 10 µs. This absolute (!) delay time is required to set up the internal high voltage. In general, Flash write operations need a 12 V external V_{PP} voltage to be applied to the V_{PP}/EBC1 pin.

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

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

The UNLOCK sequence is a specific key code sequence, which is required to enable the writing modes of the SAB 88C166(W). The UNLOCK sequence must use identical values (see example below) and must not be interrupted:

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

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

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

In Flash Erase Mode (FEE='1', FWE='1') the SAB 88C166(W) is prepared to erase the bank selected by the Bank Erase (BE) bit field in the FCR. The width of the erase pulses generated internally is defined by the Internal Flash Timer Clock Control (CKCTL) bit field of the FCR. The maximum number of erase pulses (EN_{max}) applied to the Flash memory is determined by software in the Flash erase algorithm. The chosen values for CKCTL and EN_{max} must guarantee a maximum cumulated erase time of 30 s per bank and a maximum erase pulse width of 20 ms.
The Flash bank erase operation will not start before the erase command is given. This provides additional security for the erase operation. The erase command can be any write operation to a Flash location, where the data and the even address written to must be identical:

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

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

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

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

In Flash Programming Mode (FEE='0', FWE='1') the SAB 88C166(W) is prepared to program Flash locations in the way specified by the Word or Double Word Write (WDWW) bit in the FCR. The width of the programming pulses generated internally is defined by the Internal Flash Timer Clock Control (CKCTL) bit field of the FCR. The maximum number of programming pulses (PN_{max}) applied to the Flash memory is determined by software in the Flash programming algorithm. The chosen values for CKCTL and PN_{max} must guarantee a maximum cumulated programming time of 2.5 ms per cell and a maximum programming pulse width of 200 µs.

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

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

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

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

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

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

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

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

Such a Flash verify read operation is different from the reading in the standard or in the non-verify mode. Correct verify reading needs a read operation performed twice on the same cell with an absolute time delay of 4 µs which is needed to stabilize the internal margin voltage applied to the cell. To verify that a Flash cell was erased or programmed properly, the value of the second verify read operation has to be compared against $FFFF_H$ or the target value, respectively. Clearing bit FWE to '0' exits the Flash programming mode and returns to the Flash non-verify mode.

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

Flash Protection

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

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

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

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

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

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

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

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

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

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

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

Flash Programming Algorithm

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

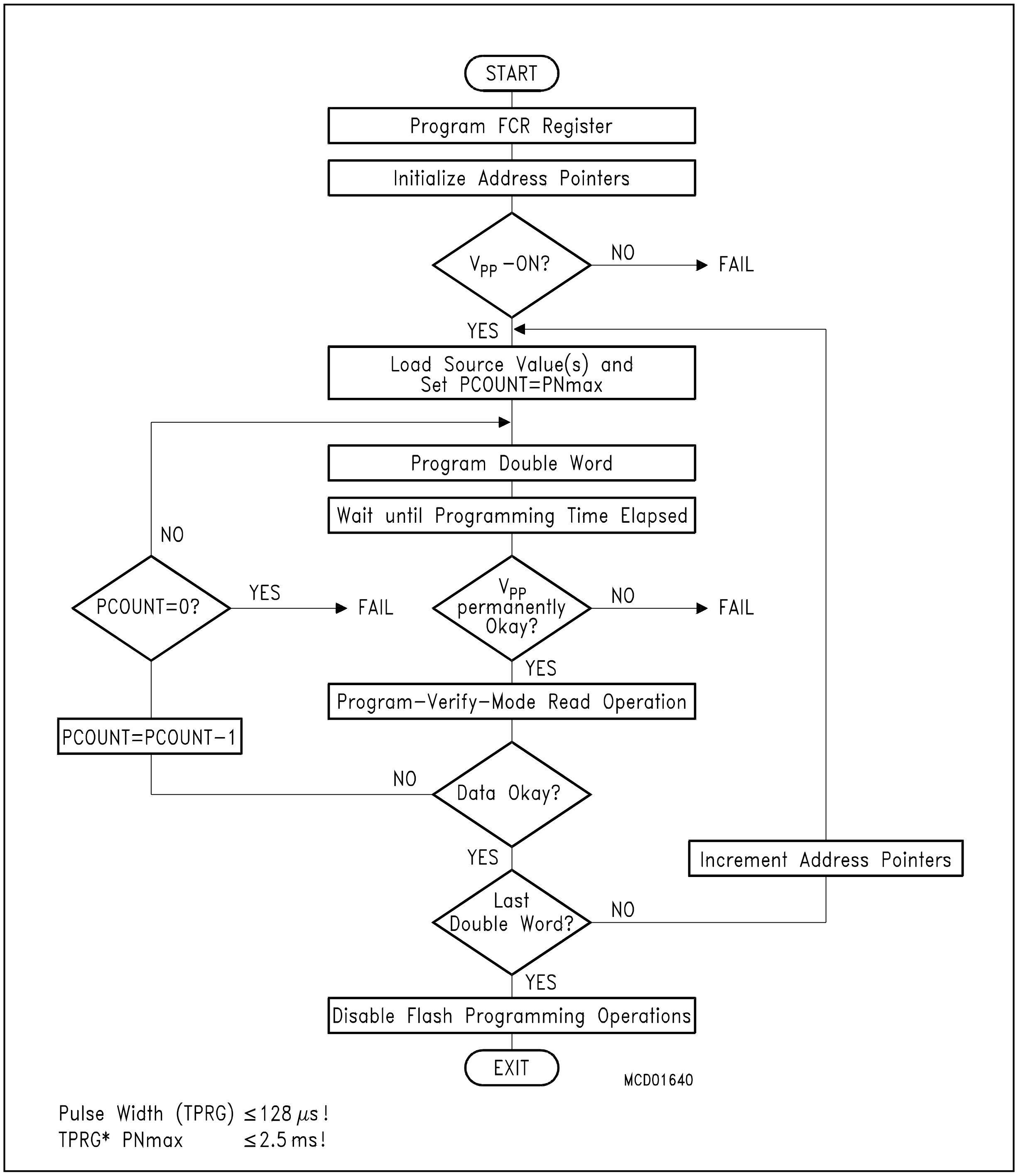

START
Program FCR Register
Initialize Address Pointers
V_{PP} –ON?
NO
FAIL
YES
Load Source Value(s) and Set PCOUNT=PNmax
Program Double Word
Wait until Programming Time Elapsed
NO
PCOUNT=0?
YES
FAIL
V_{PP} permanently Okay?
NO
FAIL
YES
Program–Verify–Mode Read Operation
PCOUNT=PCOUNT–1
NO
Data Okay?
YES
Increment Address Pointers
Last Double Word?
NO
YES
Disable Flash Programming Operations
EXIT
MCD01640
Pulse Width (TPRG) ≤ 128 µs !
TPRG* PNmax ≤ 2.5 ms!

Figure 5
Flash Programming Algorithm

Flash Programming Example

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

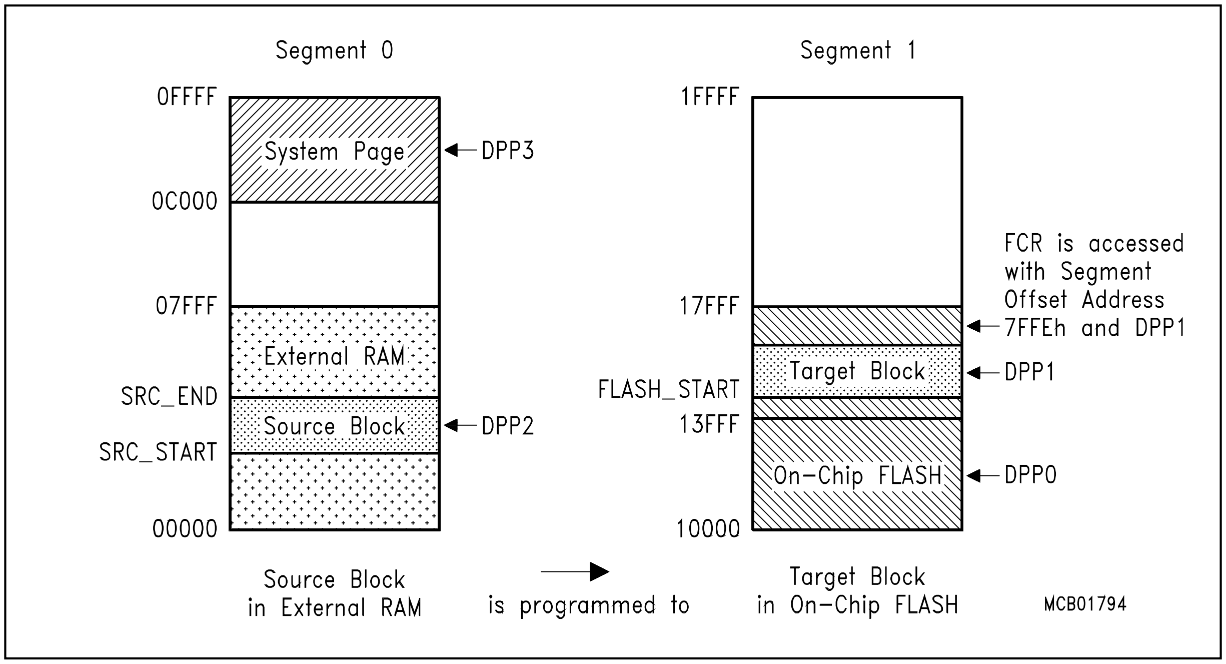


Figure 6
Memory Allocation for Flash Programming Example

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

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

The shown example uses the following assumptions:

- Pin V_{PP}/EBC1 receives a proper V_{PP} supply voltage.
- The SAB 88C166(W) runs at 20 MHz CPU clock (absolute time delays refer to it).
- The Flash memory is mapped to segment 1. All DPPs are set correctly.

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

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

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

```
MOV     R15, #1000 0000 1010 0001B
        ; #xxxx xxxx xxxx xxx1: FWE='1':      Enable Flash write operations
        ; #xxxx xxxx xxxx xx0x: FEE='0':      Select programming mode
        ; #xxxx xxxx x01x xxxx: CKCTL='01':   100 µs programming pulse (fCPU = 20 MHz)
        ; #xxxx xxxx 1xxx xxxx: WDWW='1':     Select 32-bit programming mode
        ; #1xxx xxxx xxxx xxxx: FWMSET='1':   Stay in writing mode
MOV     DPP1:pof FCR, R15                     ; Write Value to the FCR using 16-bit access
```

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

- **Test for correct V_{PP} margin at pin V_{PP}/EBC1** before a programming operation is started. If bit VPPREV reads '1', the programming voltage is correct and the algorithm can be continued. Otherwise, the programming routine could wait in Flash writing mode until V_{PP} reaches its correct value and resume programming then, or it could exit writing mode.

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

- **Load source values and initialize loop counter** (PCOUNT) with the maximum number of programming trials (PNmax) to be performed before exiting the routine with a failure. Each trial means applying a pulse of 100 µs to the selected words in the Flash memory. According to the maximum cumulated programming time of 2.5 ms allowed per cell, PNmax must be '25' here.
  The doubleword at memory location [SRC_PTR] is loaded into two auxiliary registers DATAWR1 and DATAWR2.

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

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

- **Wait until programming time elapsed** (100 µs in this example), which depends on bit field CKCTL in the FCR register and on the CPU clock frequency. End of programming is detected by polling the FBUSY flag in the FCR register. The Flash memory switches to PVM mode automatically.

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

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

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

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

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

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

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

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

Flash Erase Algorithm

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

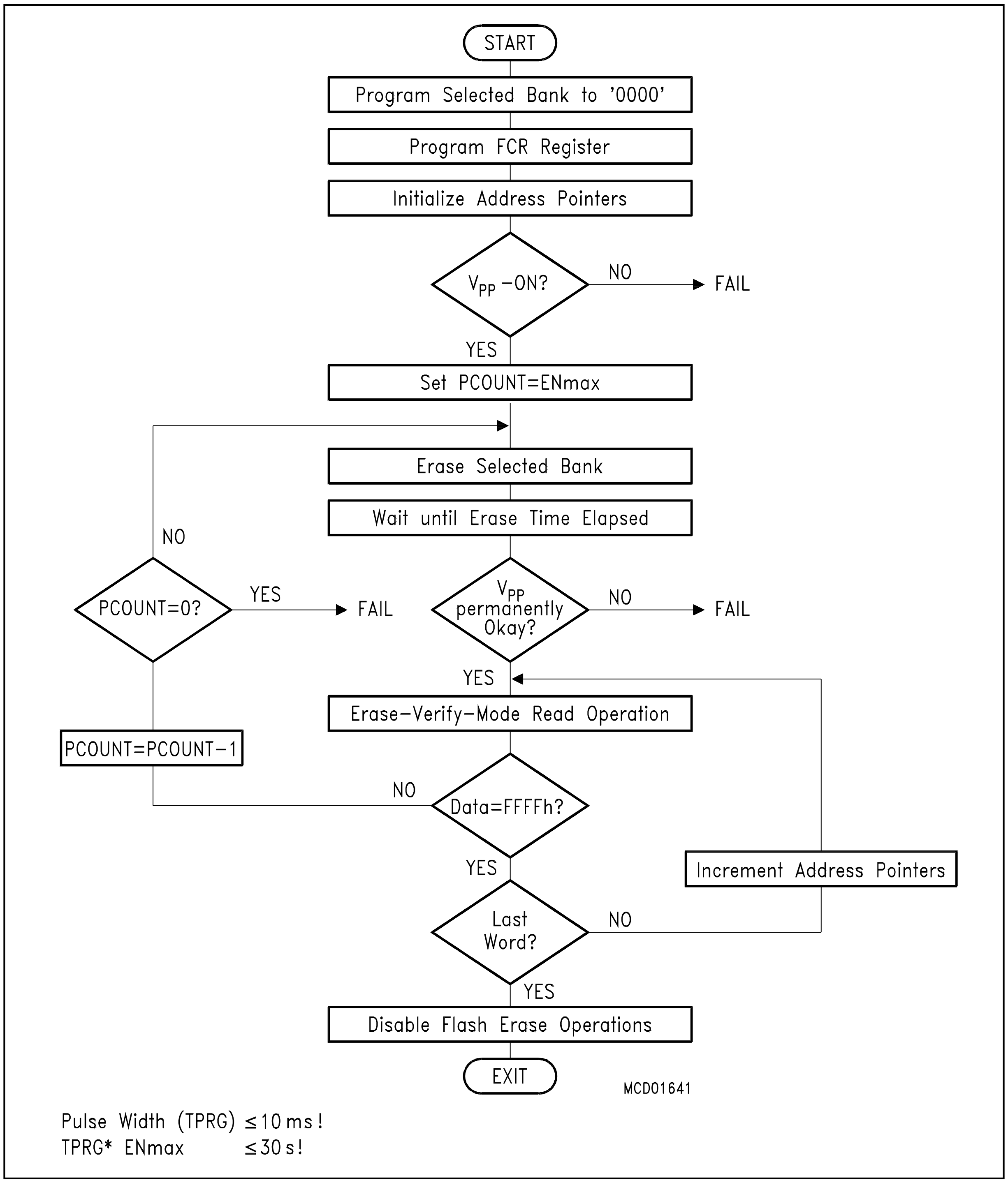


Figure 7
Flash Erase Algorithm

Flash Erase Example

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

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

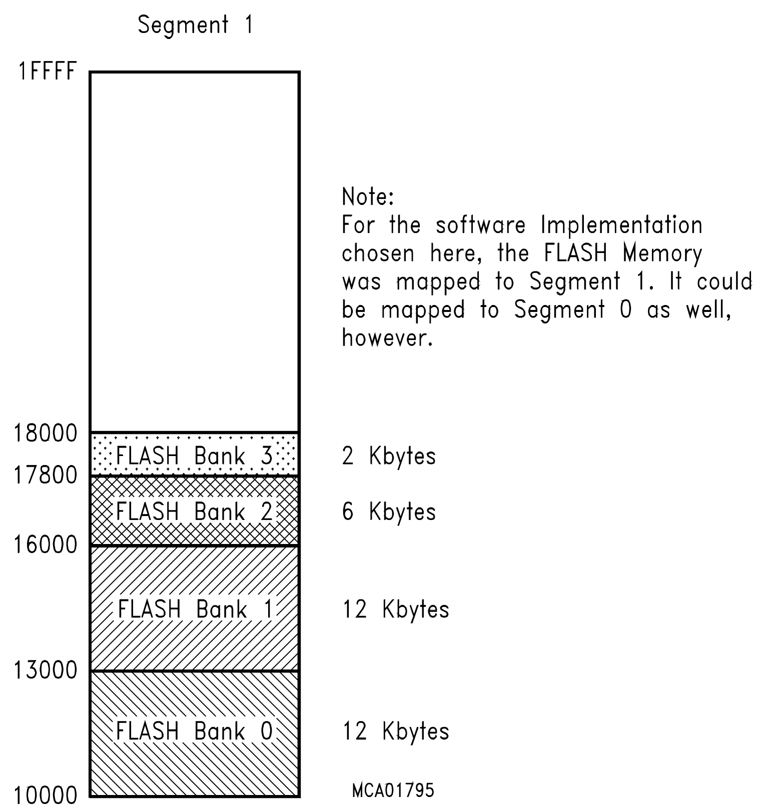

Segment 1
1FFFF
Note:
For the software Implementation chosen here, the FLASH Memory was mapped to Segment 1. It could be mapped to Segment 0 as well, however.
18000
17800
FLASH Bank 3
2 Kbytes
FLASH Bank 2
6 Kbytes
16000
FLASH Bank 1
12 Kbytes
13000
FLASH Bank 0
12 Kbytes
10000
MCA01795

Figure 8
Memory Banking for Flash Erasure

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

The shown example uses the following assumptions:

- Pin V_{PP}/EBC1 receives a proper V_{PP} supply voltage.
- The SAB 88C166(W) runs at 20 MHz CPU clock (absolute time delays refer to it).
- The Flash memory is mapped to segment 1. All DPPs are set correctly.

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

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

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

```
MOV     R15, #1000 00XX 0110 0011B
        ; #xxxx xxxx xxxx xxx1: FWE='1':      Enable Flash write operations
        ; #xxxx xxxx xxxx xx1x: FEE='1':      Select erase mode
        ; #xxxx xxxx x11x xxxx: CKCTL='11':   10 ms erase pulse (fCPU = 20 MHz)
        ; #xxxx xxXX xxxx xxxx: BE='xx':      Select the desired bank (3...0)
        ; #1xxx xxxx xxxx xxxx: FWMSET='1':   Stay in writing mode
MOV     DPP1:pof FCR, R15                     ; Write Value to the FCR using 16-bit access
```

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

- **Test for correct V_{PP} margin at pin V_{PP}/EBC1** before an erase operation is started. If bit VPPREV reads '1', the erase voltage is correct and the algorithm can be continued. Otherwise, the erase routine could wait in Flash writing mode until V_{PP} reaches its correct value and resume erasing then, or it could exit writing mode.

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

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

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

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

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

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

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

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

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

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

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

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

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

Fundamentals of Flash Technology

The Flash memory included in the SAB 88C166(W) combines the EPROM programming mechanism with electrical erasability (like an EEPROM) to create a highly reliable and cost effective memory. A Flash memory cell consists of a single transistor with a floating gate for charge storage like an EPROM, uses a thinner gate oxide, however.

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

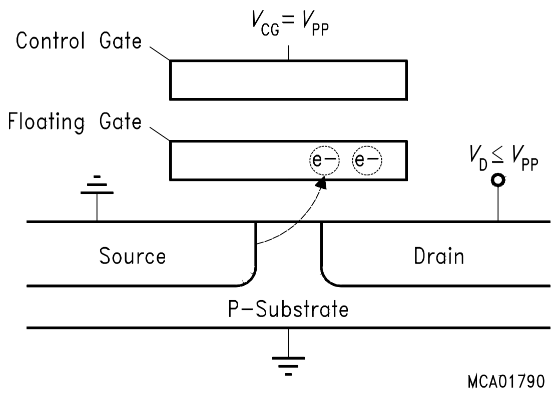


Figure 9
Flash Memory Cell Programming Mechanism

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

Unlike a standard EEPROM, where individual bytes can be erased, the Flash memory of the SAB 88C166(W) is erased block-wise which means that the high voltage is applied to all cells belonging to one block simultaneously.

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

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

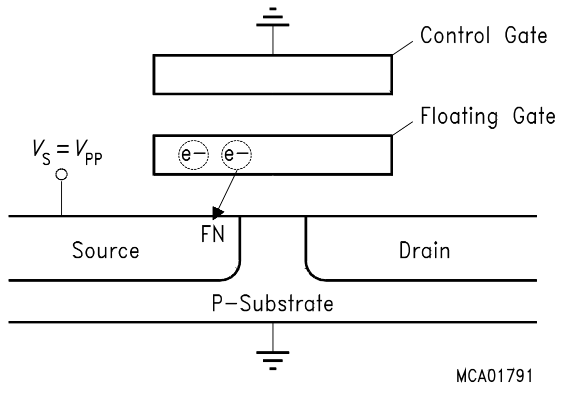
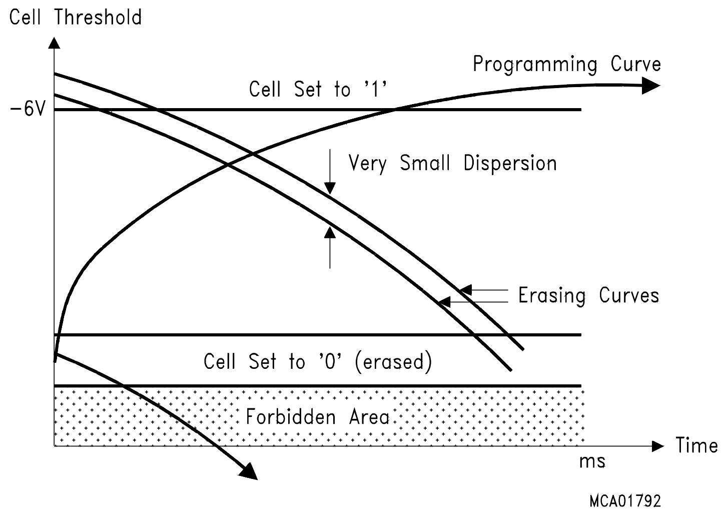


Figure 10
Flash Memory Cell Erase Mechanism

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



Erasing a block where just some cells are set to '0' could cause problems.
Thus, all cells must be set to '0' before erasure.

Figure 11
Flash Erasure

Note that the following terminology is used in this document: Flash WRITING means changing the state of the floating gate. Flash PROGRAMMING means loading electrons onto the floating gate. Flash ERASING means removing electrons from the floating gate.

Absolute Maximum Ratings

Ambient temperature under bias (T_A):
SAB 88C166(W)-5M .... 0 to + 70 ˚C
Storage temperature (T_{ST}) .... – 65 to + 125 ˚C
Voltage on V_{CC} pins with respect to ground (V_{SS}) .... – 0.5 to + 6.5 V
Voltage on any pin with respect to ground (V_{SS}) .... – 0.5 to V_{CC} + 0.5 V
Input current on any pin during overload condition .... – 10 to + 10 mA
Absolute sum of all input currents during overload condition .... |100 mA|
Power dissipation .... 1 W
Flash programming voltage (V_{PP}) .... – 0.3 to + 13.5 V

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

Parameter Interpretation

The parameters listed in the following partly represent the characteristics of the SAB 88C166(W) and partly its demands on the system. To aid in interpreting the parameters right, when evaluating them for a design, they are marked in column “Symbol”:

CC (**C**ontroller **C**haracteristics):
The logic of the SAB 88C166(W) will provide signals with the respective timing characteristics.

SR (**S**ystem **R**equirement):
The external system must provide signals with the respective timing characteristics to the SAB 88C166(W).

DC Characteristics

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

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Input low voltage EBC1/V_{PP}	V_{IL1}SR	– 0.3	0.2 V_{CC} – 0.1	V	–
Input low voltage (all except EBC1/V_{PP})	V_{IL2}SR	– 0.5	0.2 V_{CC} – 0.1	V	–
Input high voltage (all except $\overline{\text{RSTIN}}$ and XTAL1)	V_{IH}SR	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input high voltage $\overline{\text{RSTIN}}$	V_{IH1}SR	0.6 V_{CC}	V_{CC} + 0.5	V	–
Input high voltage XTAL1	V_{IH2}SR	0.7 V_{CC}	V_{CC} + 0.5	V	–
Output low voltage (Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$)	V_{OL}CC	–	0.45	V	I_{OL} = 2.4 mA
Output low voltage (all other outputs)	V_{OL1}CC	–	0.45	V	I_{OL1} = 1.6 mA
Output high voltage (Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT, $\overline{\text{RSTOUT}}$)	V_{OH}CC	0.9 V_{CC} 2.4	–	V	I_{OH} = – 500 µA I_{OH} = – 2.4 mA
Output high voltage (all other outputs)	V_{OH1}CC	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 250 µA I_{OH} = – 1.6 mA
Input leakage current (Port 5) [1]	I_{OZ1}CC	–	± 200	nA	0 V < V_{IN} < V_{CC}
Input leakage current (all other)	I_{OZ2}CC	–	± 500	nA	0 V < V_{IN} < V_{CC}
V_{PP} leakage current EBC1/V_{PP}	I_{PPS}CC	–	± 100	µA	$V_{PP} \leq V_{CC}$
$\overline{\text{RSTIN}}$ pullup resistor	R_{RST}CC	50	150	kΩ	–
Read inactive current [4]	I_{RH} [2]	–	– 40	µA	$V_{OUT} = V_{OHmin}$
Read active current [4]	I_{RL} [3]	-500	–	µA	$V_{OUT} = V_{OLmax}$
ALE inactive current [4]	I_{ALEL} [2]	–	150	µA	$V_{OUT} = V_{OLmax}$
ALE active current [4]	I_{ALEH} [3]	2100	–	µA	$V_{OUT} = V_{OHmin}$
XTAL1 input current	I_{IL}CC	–	± 20	µA	0 V < V_{IN} < V_{CC}
Pin capacitance [5] (digital inputs/outputs)	C_{IO}CC	–	10	pF	f = 1 MHz T_A = 25 ˚C
Power supply current	I_{CC}	–	50 + 5 x f_{CPU}	mA	$\overline{\text{RSTIN}}$ = V_{IL2} f_{CPU} in [MHz] [6]
Idle mode supply current	I_{ID}	–	30 + 1.5 x f_{CPU}	mA	$\overline{\text{RSTIN}}$ = V_{IH1} f_{CPU} in [MHz] [6]

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Power-down mode supply current	I_{PD}	–	50	µA	V_{CC} = 5.5 V [7)]
V_{PP} read current	I_{PPR}	–	200	µA	$V_{PP} > V_{CC}$
V_{PP} writing current	I_{PPW}	–	50	mA	1/TCL = 40 MHz 32-bit programming V_{PP} = 12 V
V_{PP} during write/read	V_{PP}	11.4	12.6	V	

Notes

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

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

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

4) This specification is only valid during Reset, or during Hold-mode.

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

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

7) All inputs (including pins configured as inputs) at 0 V to 0.1 V or at V_{CC} – 0.1 V to V_{CC}, V_{REF} = 0 V, all outputs (including pins configured as outputs) disconnected.
A voltage of $V_{CC} \geq 2.5$ V is sufficient to retain the content of the internal RAM during power down mode.

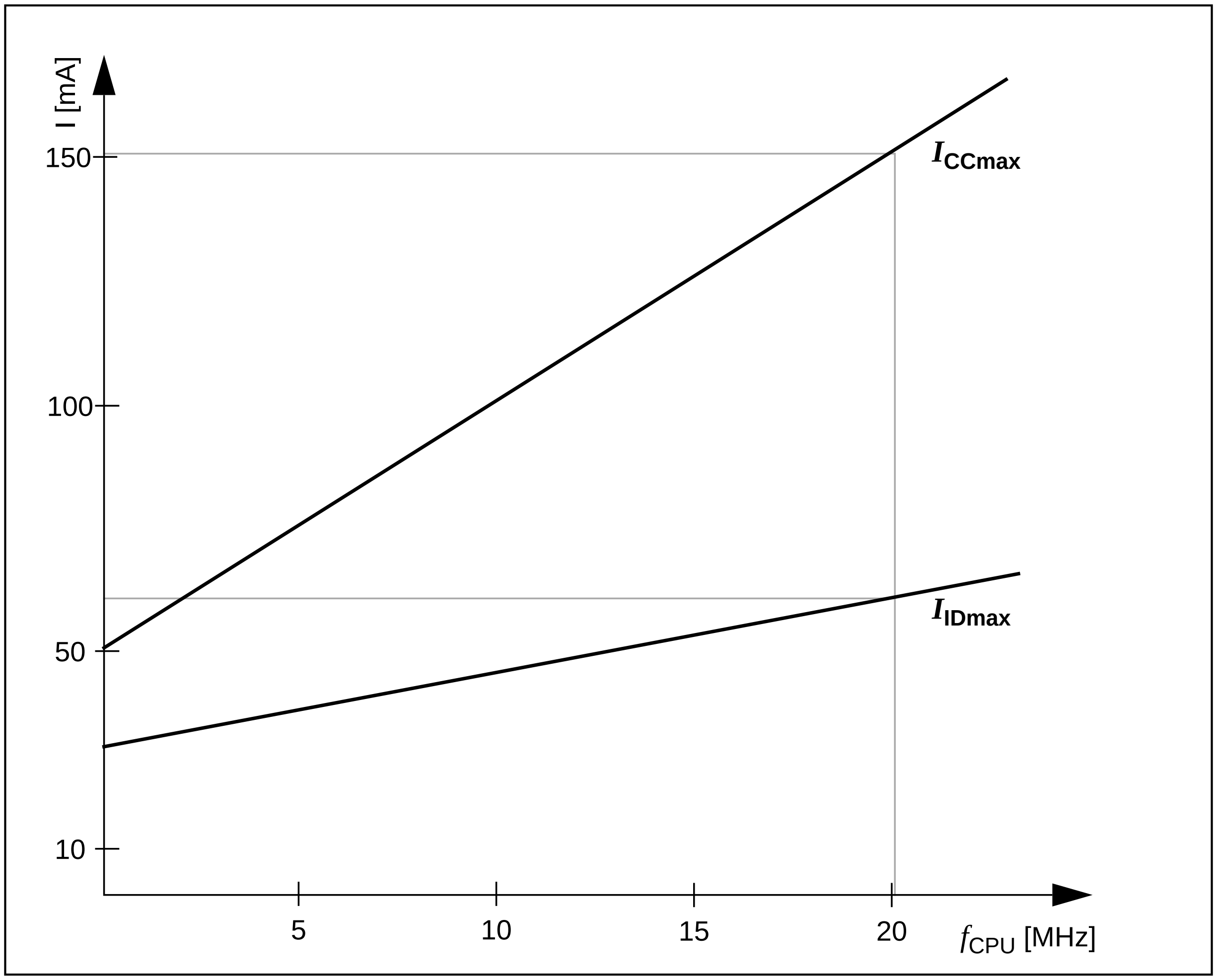


Figure 12
Supply/Idle Current as a Function of Operating Frequency

A/D Converter Characteristics

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ˚C for SAB 88C166(W)-5M
4.0 V ≤ V_{AREF} ≤ V_{CC} + 0.1 V; V_{SS} – 0.1 V ≤ V_{AGND} ≤ V_{SS} + 0.2 V

Parameter	Symbol	Limit Values		Unit	Test Condition
		min.	max.		
Analog input voltage range	V_{AIN}SR	V_{AGND}	V_{AREF}	V	1)
Sample time	t_SCC	–	2 t_{SC}		2) 4)
Conversion time	t_CCC	–	10 t_{CC} + t_S + 4TCL		3) 4)
Total unadjusted error	TUECC	–	± 2	LSB	5)
Internal resistance of reference voltage source	R_{AREF}CC	–	t_{CC} / 250 - 0.25	kΩ	t_{CC} in [ns] 6) 7)
Internal resistance of analog source	R_{ASRC}CC	–	t_S / 500 – 0.25	kΩ	t_S in [ns] 2) 7)
ADC input capacitance	C_{AIN}CC	–	50	pF	7)

Notes

1) V_{AIN} may exceed V_{AGND} or V_{AREF} up to the absolute maximum ratings. However, the conversion result in these cases will be X000$_H$ or X3FF$_H$, respectively.

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

3) This parameter includes the sample time t_S, the time for determining the digital result and the time to load the result register with the conversion result.
The value for the conversion clock is t_{CC} = TCL x 32.

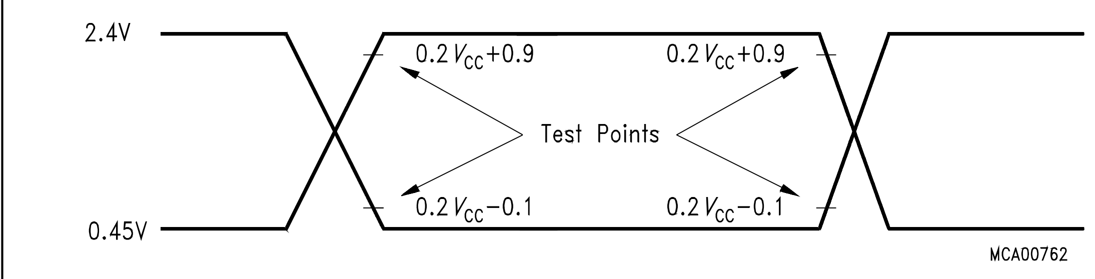
4) This parameter depends on the ADC control logic. It is not a real maximum value, but rather a fixum.

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

6) During the conversion the ADC's capacitance must be repeatedly charged or discharged. The internal resistance of the reference voltage source must allow the capacitors to reach their respective voltage level within t_{CC}. The maximum internal resistance results from the CPU clock period.

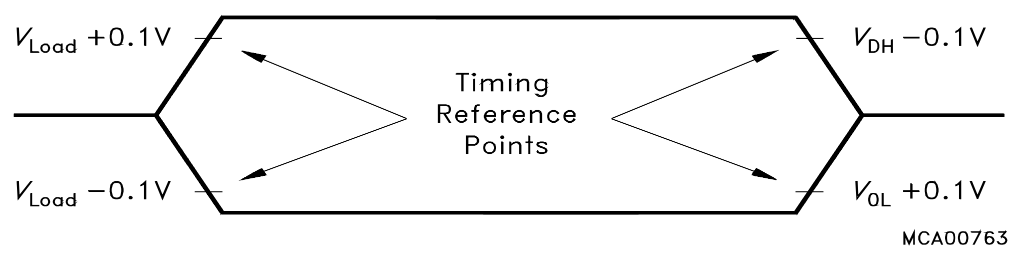
7) Not 100% tested, guaranteed by design characterization.

Testing Waveforms



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

Figure 13
Input Output Waveforms



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

Figure 14
Float Waveforms

Memory Cycle Variables

The timing tables below use three variables which are derived from registers SYSCON and BUSCON1 and represent the special characteristics of the programmed memory cycle. The following table describes, how these variables are to be computed.

Description	Symbol	Values
ALE Extension	t_A	TCL x <ALECTL>
Memory Cycle Time Waitstates	t_C	2TCL x (15 – <MCTC>)
Memory Tristate Time	t_F	2TCL x (1 – <MTTC>)

AC Characteristics

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

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

AC Characteristics

External Clock Drive XTAL1 for the SAB 88C166

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

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

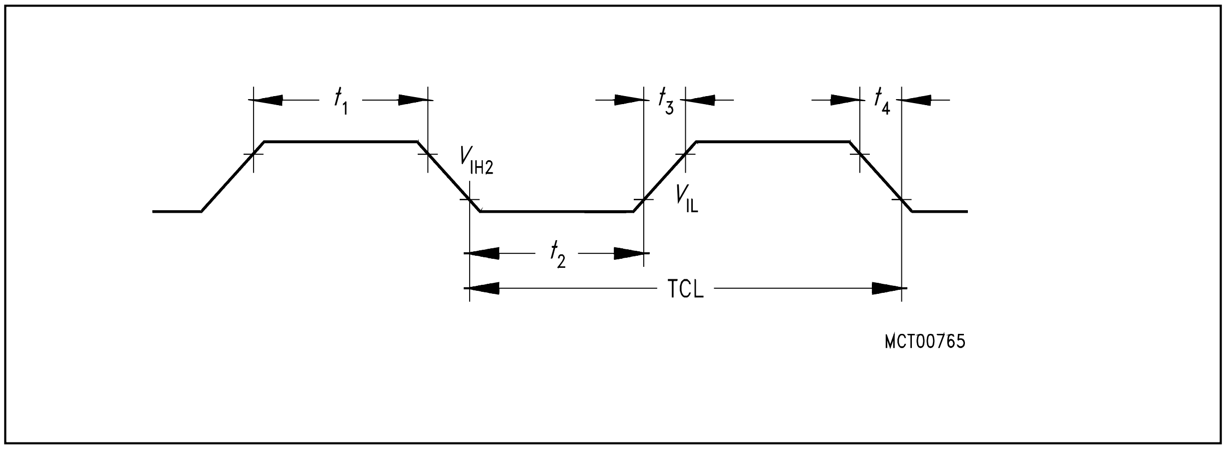


Figure 15
External Clock Drive XTAL1

AC Characteristics (cont'd)
External Clock Drive XTAL1 for the SAB 88C166W
V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166W-M

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

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

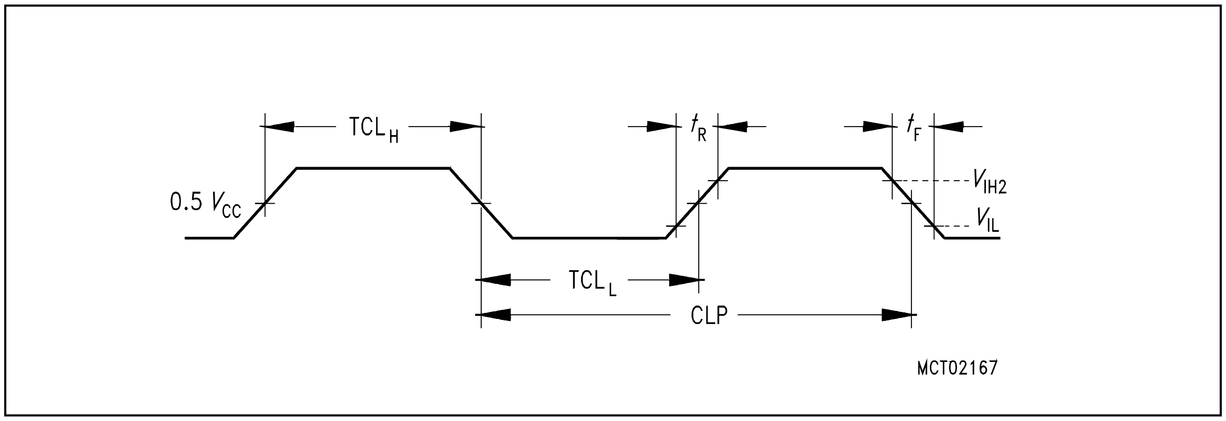


Figure 16
External Clock Drive XTAL1

AC Characteristics (cont'd)
Multiplexed Bus for the SAB 88C166
V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 ˚C for SAB 88C166-5M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{RD}$, $\overline{WR}$, $\overline{BHE}$, CLKOUT) = 100 pF
ALE cycle time = 6 TCL + 2t_A + t_C + t_F (150 ns at 20-MHz CPU clock without waitstates)

Parameter	Symbol	Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
ALE high time	t_5CC	15 + t_A	–	TCL – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL – 15 + t_A	–	ns
Address hold after ALE	t_7CC	15 + t_A	–	TCL – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 10 + t_A	–	– 10 + t_A	–	ns
Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_{10}CC	–	5	–	5	ns
Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_{11}CC	–	30	–	TCL + 5	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}CC	40 + t_C	–	2TCL – 10 + t_C	–	ns
$\overline{RD}$ $\overline{WR}$ low time (no RW-delay)	t_{13}CC	65 + t_C	–	3TCL – 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}SR	–	30 + t_C	–	2TCL – 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}SR	–	55 + t_C	–	3TCL – 20 + t_C	ns
ALE low to valid data in	t_{16}SR	–	55 + t_A + t_C	–	3TCL – 20 + t_A + t_C	ns
Address to valid data in	t_{17}SR	–	75 + 2t_A + t_C	–	4TCL – 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}SR	0	–	0	–	ns
Data float after $\overline{RD}$	t_{19}SR	–	35 + t_F	–	2TCL – 15 + t_F	ns
Data valid to $\overline{WR}$	t_{22}CC	35 + t_C	–	2TCL – 15 + t_C	–	ns
Data hold after $\overline{WR}$	t_{23}CC	35 + t_F	–	2TCL – 15 + t_F	–	ns

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

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

Parameter	**Symbol**	**CPU Clock = 16 MHz Duty cycle 0.4 to 0.6**		**Variable CPU Clock 1/CLP = 1 to 20 MHz**		**Unit**
		min.	**max.**	**min.**	**max.**	
ALE high time	t_5CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL_{min} – 15 + t_A	–	ns
Address hold after ALE	t_7CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 10 + t_A	–	– 10 + t_A	–	ns
Address float after $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_{10}CC	–	5	–	5	ns
Address float after $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_{11}CC	–	42.5	–	TCL_{max} + 5	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}CC	52.5 + t_C	–	CLP – 10 + t_C	–	ns
$\overline{RD}$ $\overline{WR}$ low time (no RW-delay)	t_{13}CC	77.5 + t_C	–	CLP+TCL_{min} – 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}SR	–	47.5 + t_C	–	CLP – 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}SR	–	72.5 + t_C	–	CLP+TCL_{min} – 20 + t_C	ns
ALE low to valid data in	t_{16}SR	–	72.5 + t_A + t_C	–	CLP+TCL_{min} – 20 + t_C	ns
Address to valid data in	t_{17}SR	–	100 + 2t_A + t_C	–	2CLP – 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}SR	0	–	0	–	ns
Data float after $\overline{RD}$	t_{19}SR	–	47.5 + t_F	–	CLP – 15 + t_F	ns

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

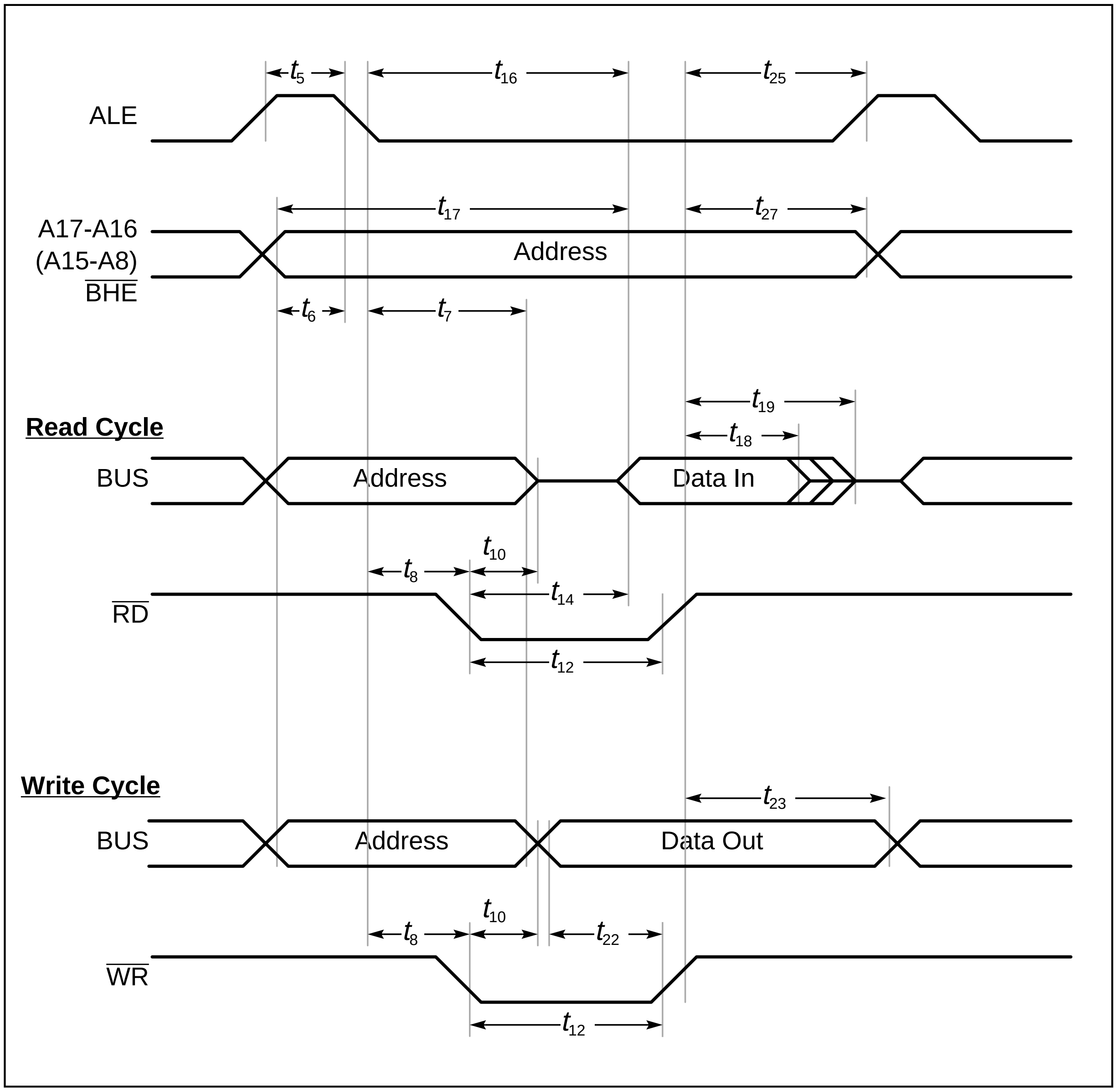


Figure 17
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Normal ALE

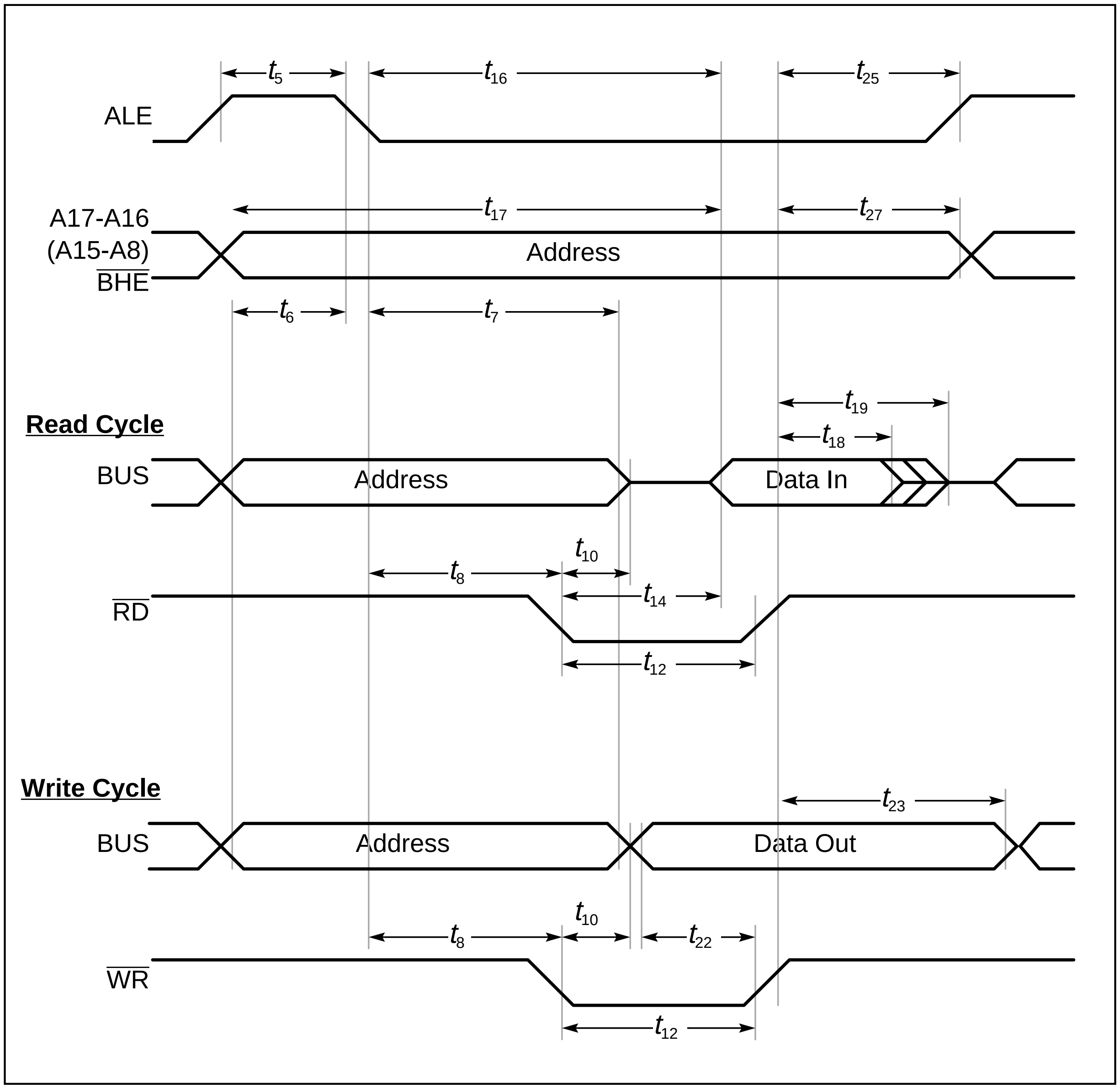


Figure 18
External Memory Cycle: Multiplexed Bus, With Read/Write Delay, Extended ALE

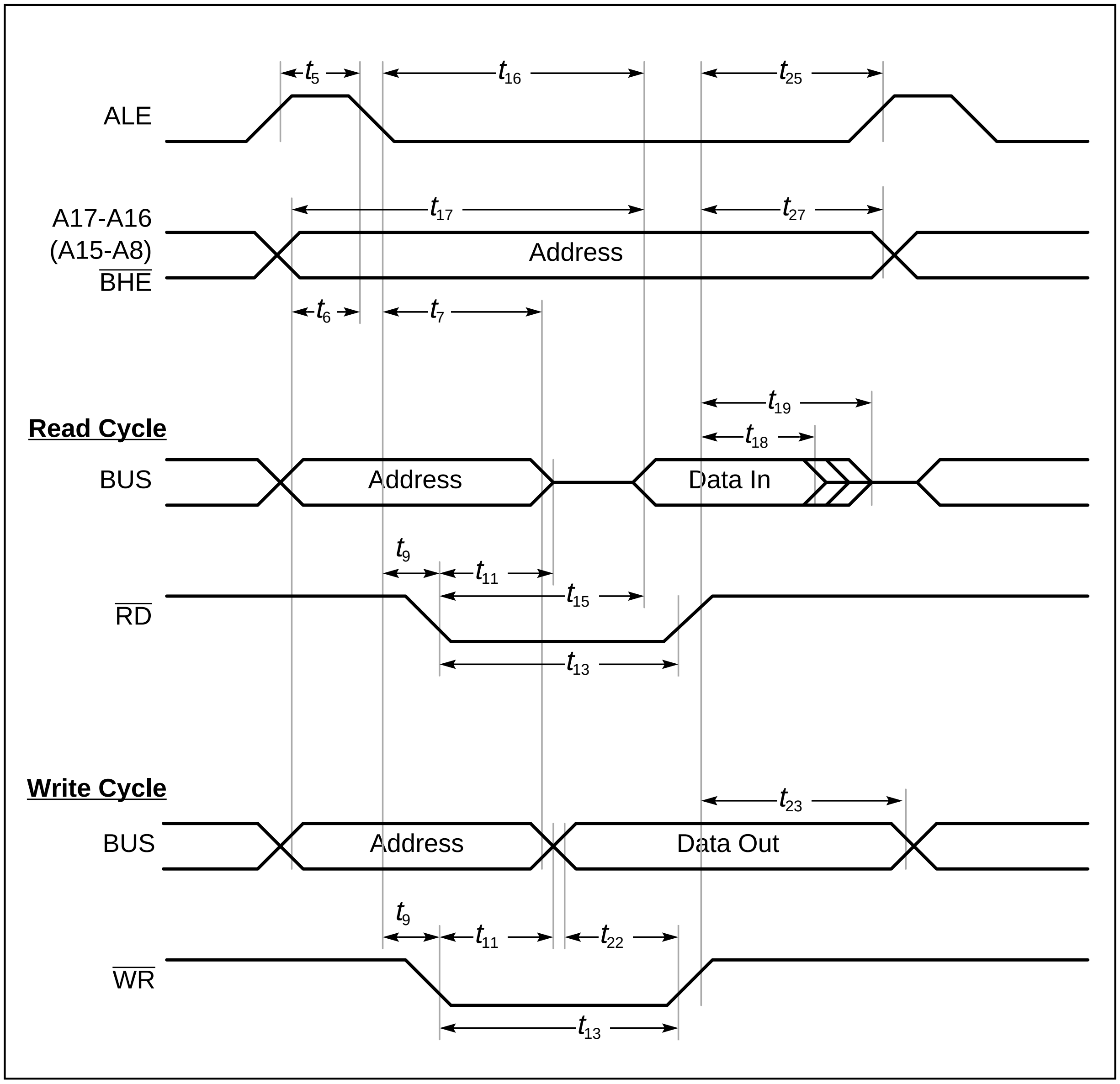


Figure 19
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Normal ALE

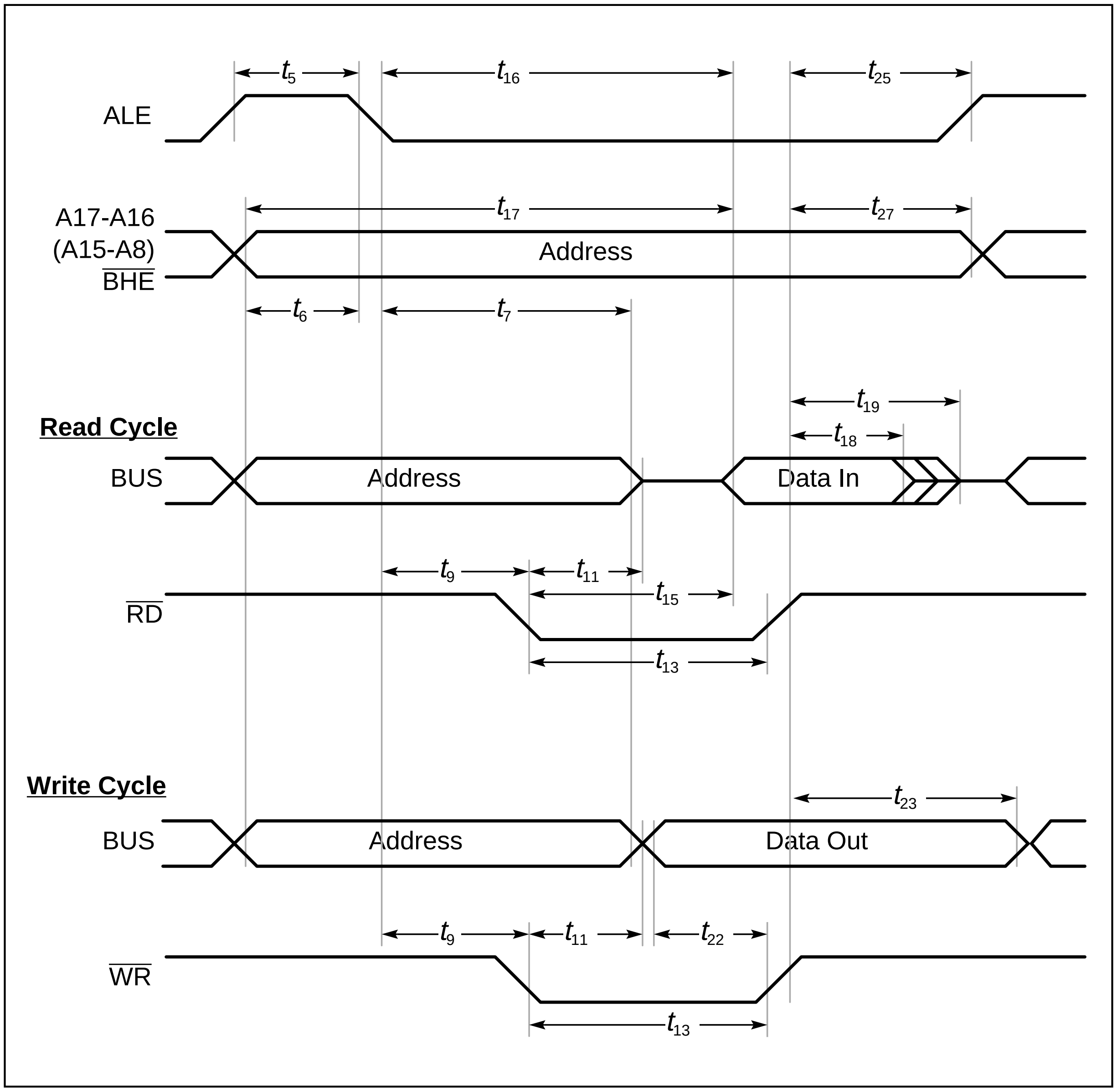


Figure 20
External Memory Cycle: Multiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)

Demultiplexed Bus for the SAB 88C166

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V

T_A = 0 to + 70 °C for SAB 88C166-5M

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

ALE cycle time = 4 TCL + 2t_A + t_C + t_F (100 ns at 20-MHz CPU clock without waitstates)

Parameter	Symbol	Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
		min.	max.	min.	max.	
ALE high time	t_5CC	15 + t_A	–	TCL – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL – 15 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 10 + t_A	–	– 10 + t_A	–	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}CC	40 + t_C	–	2TCL – 10 + t_C	–	ns
$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	t_{13}CC	65 + t_C	–	3TCL – 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}SR	–	30 + t_C	–	2TCL – 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}SR	–	55 + t_C	–	3TCL – 20 + t_C	ns
ALE low to valid data in	t_{16}SR	–	55 + t_A + t_C	–	3TCL – 20 + t_A + t_C	ns
Address to valid data in	t_{17}SR	–	75 + 2t_A + t_C	–	4TCL – 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}SR	0	–	0	–	ns
Data float after $\overline{RD}$ rising edge (with RW-delay)	t_{20}SR	–	35 + t_F	–	2TCL – 15 + t_F	ns
Data float after $\overline{RD}$ rising edge (no RW-delay)	t_{21}SR	–	15 + t_F	–	TCL – 10 + t_F	ns
Data valid to $\overline{WR}$	t_{22}CC	35 + t_C	–	2TCL – 15 + t_C	–	ns
Data hold after $\overline{WR}$	t_{24}CC	15 + t_F	–	TCL - 10 + t_F	–	ns
ALE rising edge after $\overline{RD}$, $\overline{WR}$	t_{26}CC	– 10 + t_F	–	– 10 + t_F	–	ns
Address hold after $\overline{RD}$, $\overline{WR}$	t_{28}CC	0 + t_F	–	0 + t_F	–	ns

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

Parameter	**Symbol**	**CPU Clock = 16 MHz Duty cycle 0.4 to 0.6**		**Variable CPU Clock 1/CLP = 1 to 20 MHz**		**Unit**
		min.	**max.**	**min.**	**max.**	
ALE high time	t_5CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
Address setup to ALE	t_6CC	10 + t_A	–	TCL_{min} – 15 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (with RW-delay)	t_8CC	15 + t_A	–	TCL_{min} – 10 + t_A	–	ns
ALE falling edge to $\overline{RD}$, $\overline{WR}$ (no RW-delay)	t_9CC	– 10 + t_A	–	– 10 + t_A	–	ns
$\overline{RD}$, $\overline{WR}$ low time (with RW-delay)	t_{12}CC	52.5 + t_C	–	CLP – 10 + t_C	–	ns
$\overline{RD}$, $\overline{WR}$ low time (no RW-delay)	t_{13}CC	77.5 + t_C	–	CLP+TCL_{min} – 10 + t_C	–	ns
$\overline{RD}$ to valid data in (with RW-delay)	t_{14}SR	–	47.5 + t_C	–	CLP – 20 + t_C	ns
$\overline{RD}$ to valid data in (no RW-delay)	t_{15}SR	–	72.5 + t_C	–	CLP+TCL_{min} – 20 + t_C	ns
ALE low to valid data in	t_{16}SR	–	72.5 + t_A + t_C	–	CLP+TCL_{min} – 20 + t_A + t_C	ns
Address to valid data in	t_{17}SR	–	100 + 2t_A + t_C	–	2CLP – 25 + 2t_A + t_C	ns
Data hold after $\overline{RD}$ rising edge	t_{18}SR	0	–	0	–	ns
Data float after $\overline{RD}$ rising edge (with RW-delay)	t_{20}SR	–	47.5 + t_F	–	CLP – 15 + t_F	ns
Data float after $\overline{RD}$ rising edge (no RW-delay)	t_{21}SR	–	15 + t_F	–	TCL_{min} – 10 + t_F	ns
Data valid to $\overline{WR}$	t_{22}CC	47.5 + t_C	–	CLP – 15 + t_C	–	ns
Data hold after $\overline{WR}$	t_{24}CC	15 + t_F	–	TCL_{min} – 10 + t_F	–	ns

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

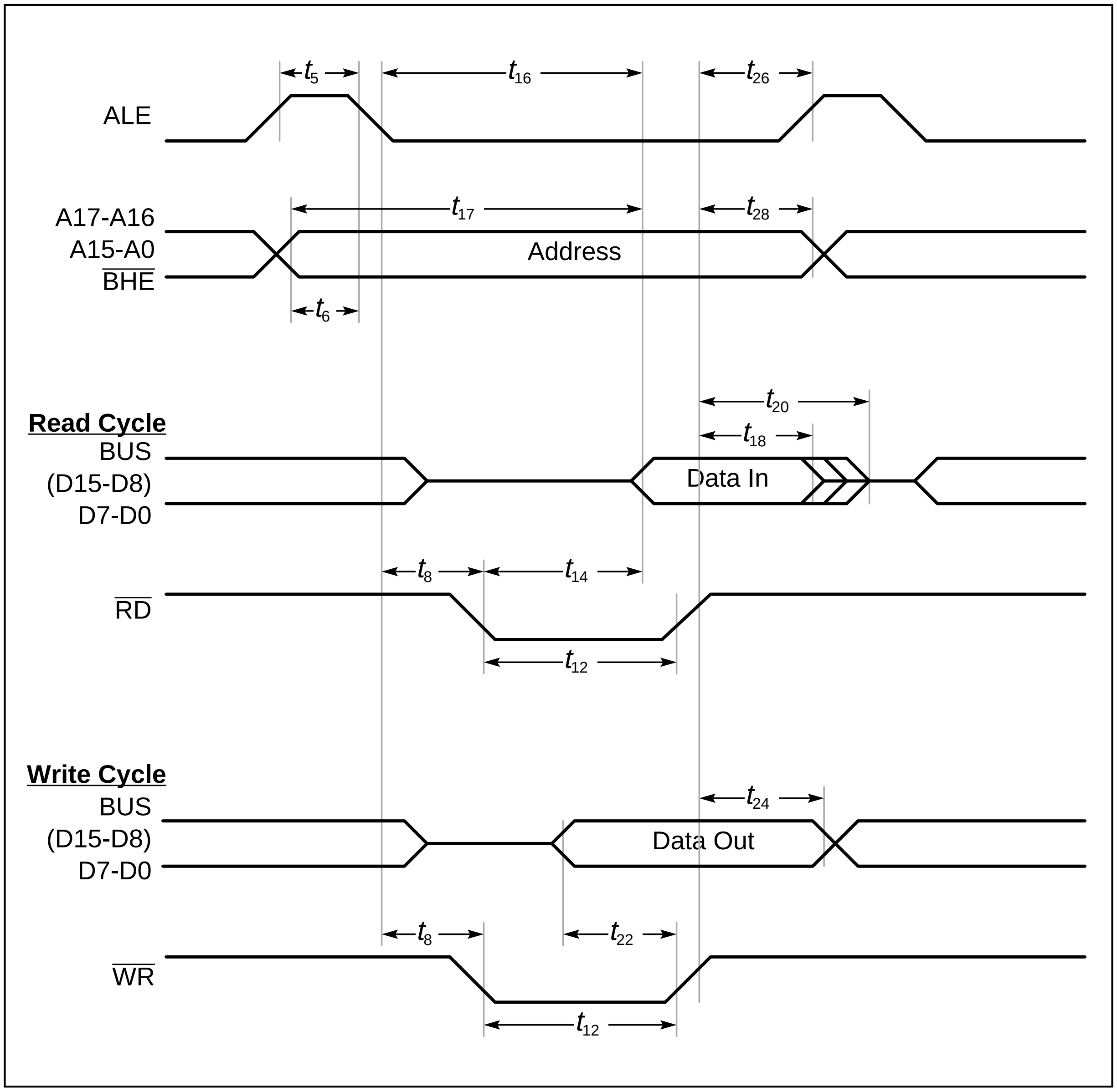


Figure 21
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Normal ALE

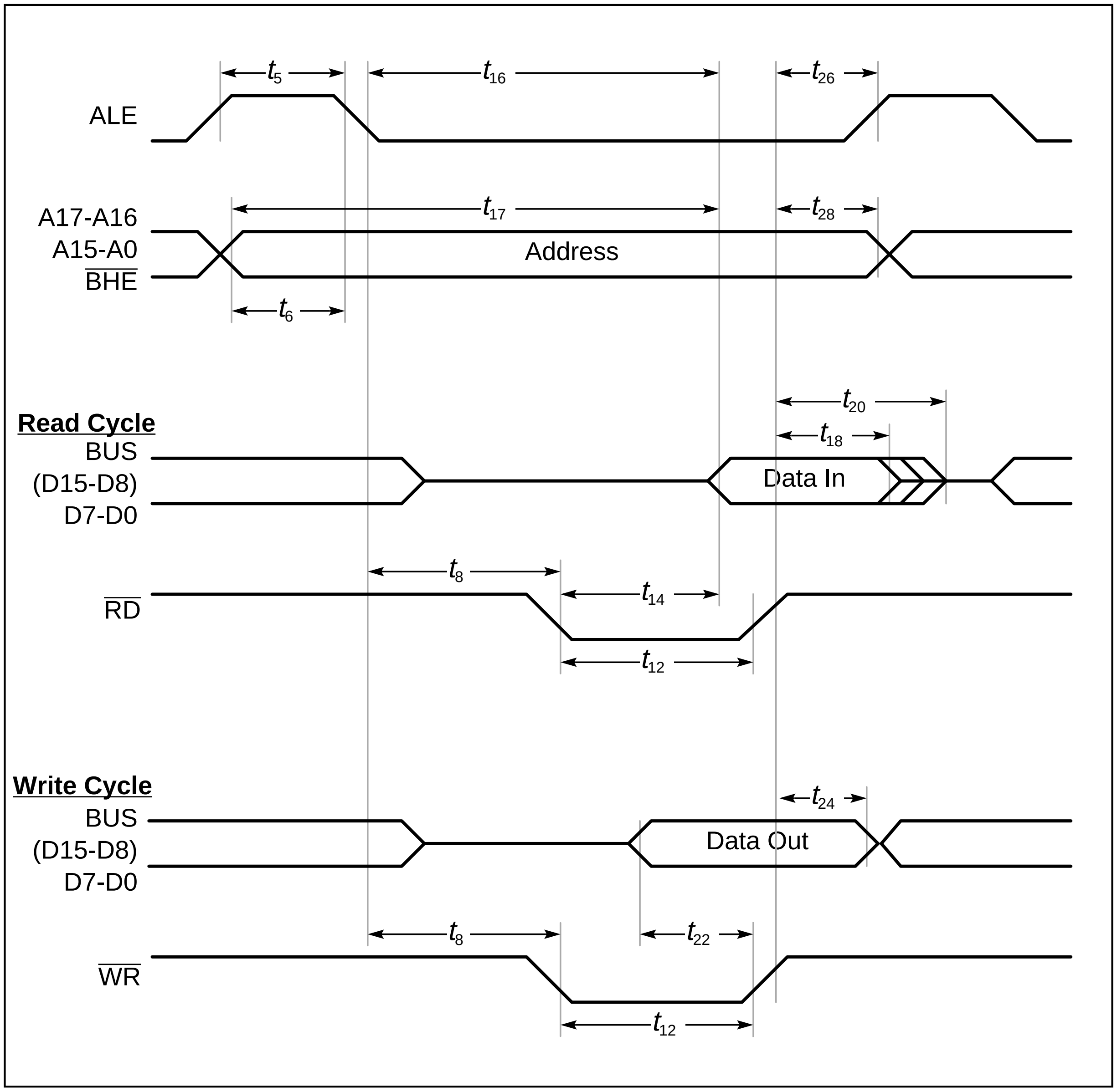


Figure 22
External Memory Cycle: Demultiplexed Bus, With Read/Write Delay, Extended ALE

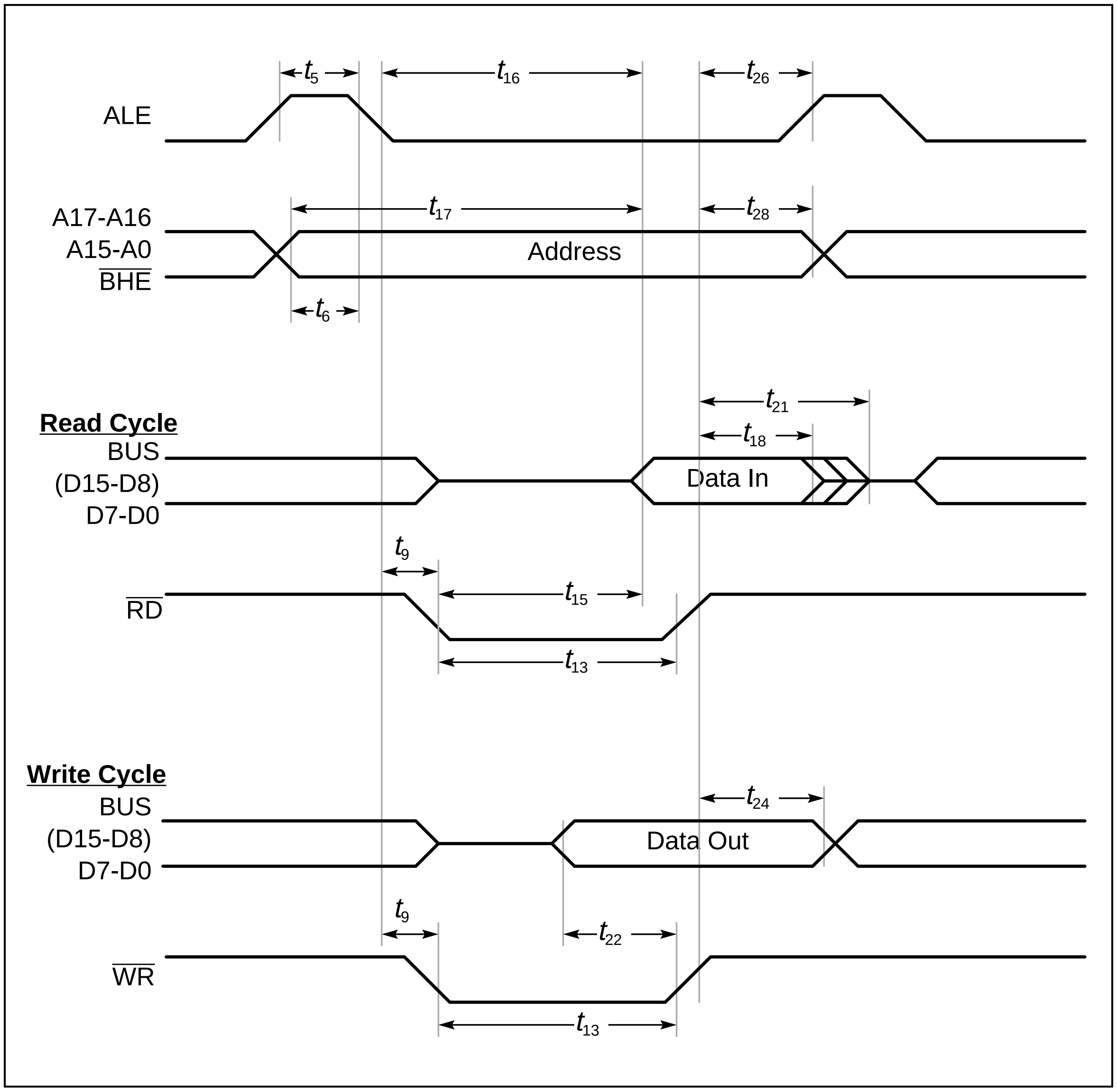


Figure 23
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Normal ALE

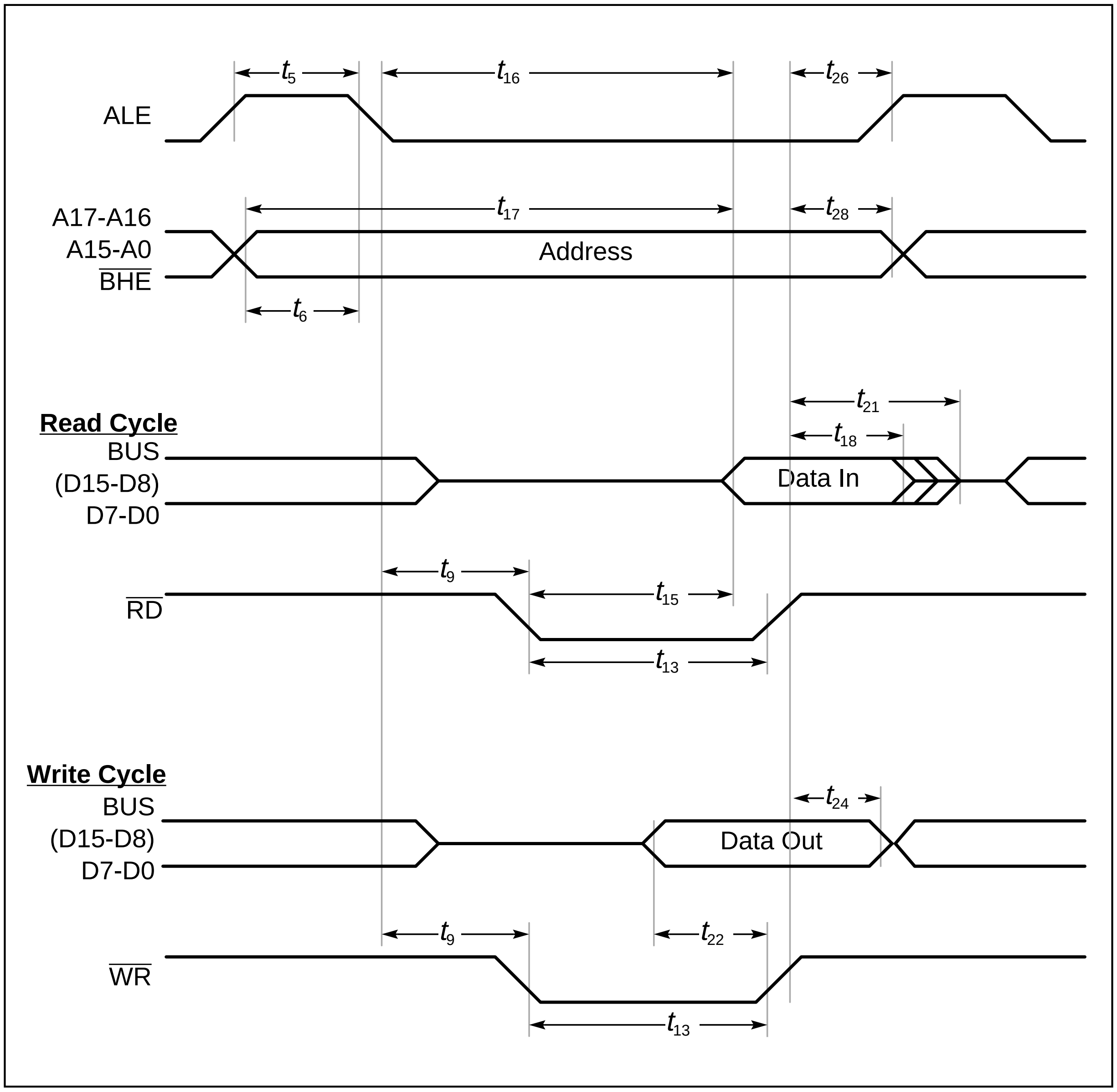


Figure 24
External Memory Cycle: Demultiplexed Bus, No Read/Write Delay, Extended ALE

AC Characteristics (cont'd)
CLKOUT and $\overline{\text{READY}}$ for SAB 88C166
V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166-5M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT) = 100 pF

Parameter	Symbol		Max. CPU Clock = 20 MHz		Variable CPU Clock 1/2TCL = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
CLKOUT cycle time	t_{29}	CC	50	50	2TCL	2TCL	ns
CLKOUT high time	t_{30}	CC	20	–	TCL – 5	–	ns
CLKOUT low time	t_{31}	CC	15	–	TCL – 10	–	ns
CLKOUT rise time	t_{32}	CC	–	5	–	5	ns
CLKOUT fall time	t_{33}	CC	–	5	–	5	ns
CLKOUT rising edge to ALE falling edge	t_{34}	CC	$0 + t_A$	$10 + t_A$	$0 + t_A$	$10 + t_A$	ns
Synchronous $\overline{\text{READY}}$ setup time to CLKOUT	t_{35}	SR	10	–	10	–	ns
Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	t_{36}	SR	10	–	10	–	ns
Asynchronous $\overline{\text{READY}}$ low time	t_{37}	SR	65	–	2TCL + 15	–	ns
Asynchronous $\overline{\text{READY}}$ setup time [1]	t_{58}	SR	20	–	20	–	ns
Asynchronous $\overline{\text{READY}}$ hold time [1]	t_{59}	SR	0	–	0	–	ns
Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus) [2]	t_{60}	SR	0	$0 + 2t_A + t_F$ [2]	0	$\text{TCL} - 25 + 2t_A + t_F$ [2]	ns

Notes

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

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

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

Parameter	Symbol		CPU Clock = 16 MHz Duty cycle 0.4 to 0.6		Variable CPU Clock 1/CLP = 1 to 20 MHz		Unit
			min.	max.	min.	max.	
CLKOUT cycle time	t_{29}	CC	62.5	62.5	CLP	CLP	ns
CLKOUT high time	t_{30}	CC	20	–	$TCL_{min} – 5$	–	ns
CLKOUT low time	t_{31}	CC	15	–	$TCL_{min} – 10$	–	ns
CLKOUT rise time	t_{32}	CC	–	5	–	5	ns
CLKOUT fall time	t_{33}	CC	–	5	–	5	ns
CLKOUT rising edge to ALE falling edge	t_{34}	CC	$0 + t_A$	$10 + t_A$	$0 + t_A$	$10 + t_A$	ns
Synchronous $\overline{\text{READY}}$ setup time to CLKOUT	t_{35}	SR	10	–	10	–	ns
Synchronous $\overline{\text{READY}}$ hold time after CLKOUT	t_{36}	SR	10	–	10	–	ns
Asynchronous $\overline{\text{READY}}$ low time	t_{37}	SR	77.5	–	CLP + 15	–	ns
Asynchronous $\overline{\text{READY}}$ setup time [1)]	t_{58}	SR	20	–	20	–	ns
Asynchronous $\overline{\text{READY}}$ hold time [1)]	t_{59}	SR	0	–	0	–	ns
Async. $\overline{\text{READY}}$ hold time after $\overline{\text{RD}}$, $\overline{\text{WR}}$ high (Demultiplexed Bus) [2)]	t_{60}	SR	0	0 $+ 2t_A + t_F$ [2)]	0	TCL – 25 $+ 2t_A + t_F$ [2)]	ns

Notes

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

2) Demultiplexed bus is the worst case. For multiplexed bus 2TCL are to be added to the maximum values. This adds even more time for deactivating $\overline{\text{READY}}$.
The $2t_A$ refer to the next following bus cycle.

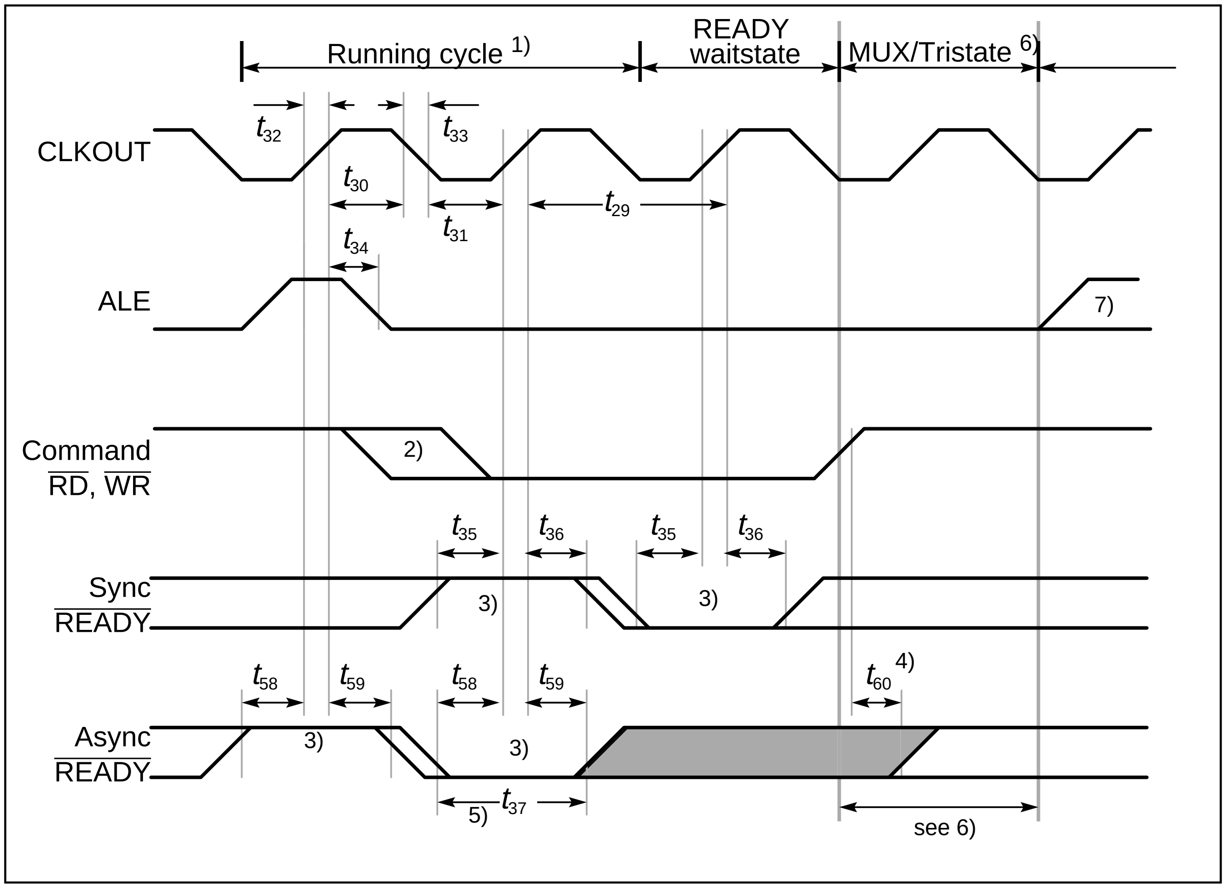


Figure 25
CLKOUT and $\overline{\text{READY}}$

Notes

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

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

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

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

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

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

7) The next external bus cycle may start here.

AC Characteristics (cont'd)
External Bus Arbitration

V_{CC} = 5 V ± 10 %; V_{SS} = 0 V
T_A = 0 to + 70 °C for SAB 88C166(W)-5M
C_L (for Port 0, Port 1, Port 4, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{BHE}}$, CLKOUT) = 100 pF

Parameter	**Symbol**		**Max. CPU Clock = 20 MHz**		**Variable CPU Clock 1/2TCL = 1 to 20 MHz**		**Unit**
			min.	**max.**	**min.**	**max.**	
$\overline{\text{HOLD}}$ input setup time to CLKOUT	t_{61}	SR	20	–	20	–	ns
CLKOUT to $\overline{\text{HLDA}}$ high or $\overline{\text{BREQ}}$ low delay	t_{62}	CC	–	50	–	50	ns
CLKOUT to $\overline{\text{HLDA}}$ low or $\overline{\text{BREQ}}$ high delay	t_{63}	CC	–	50	–	50	ns
Other signals release	t_{66}	CC	–	25	–	25	ns
Other signals drive	t_{67}	CC	– 5	35	– 5	35	ns

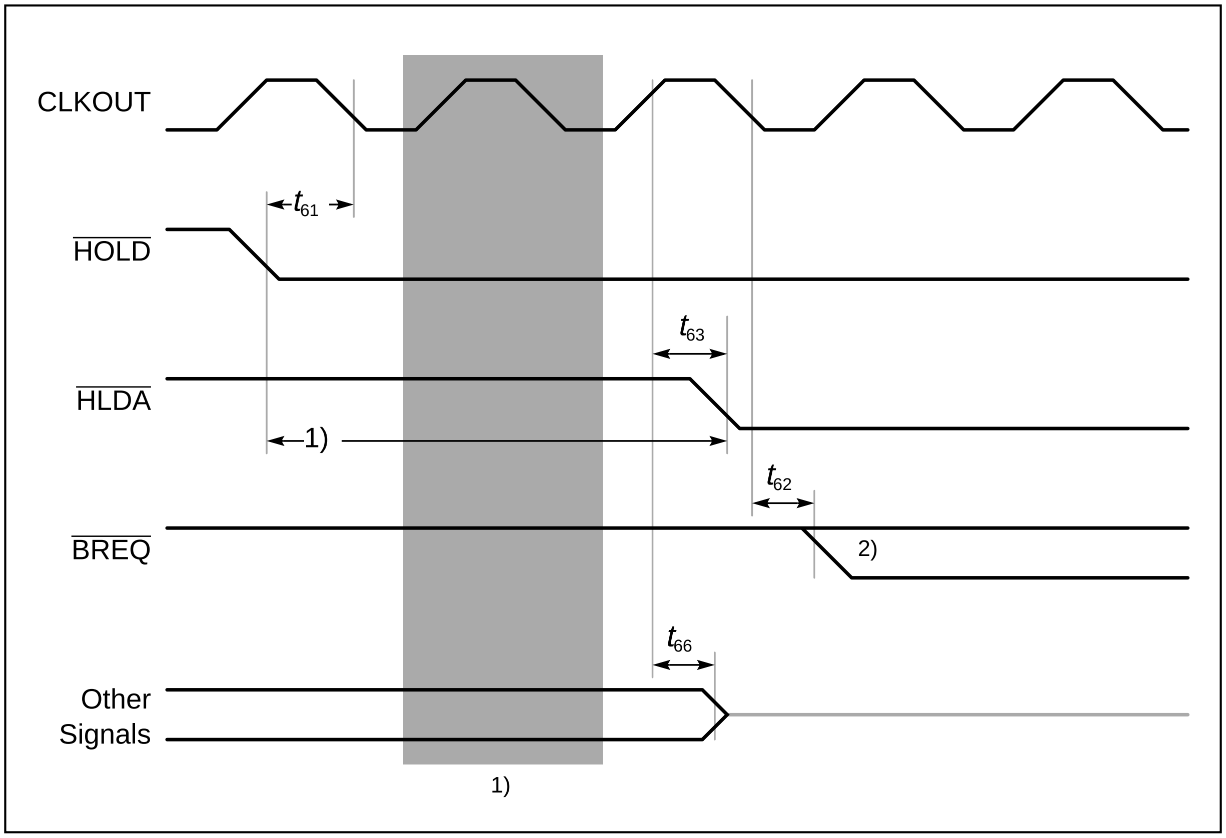


Figure 26
External Bus Arbitration, Releasing the Bus

Notes

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

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

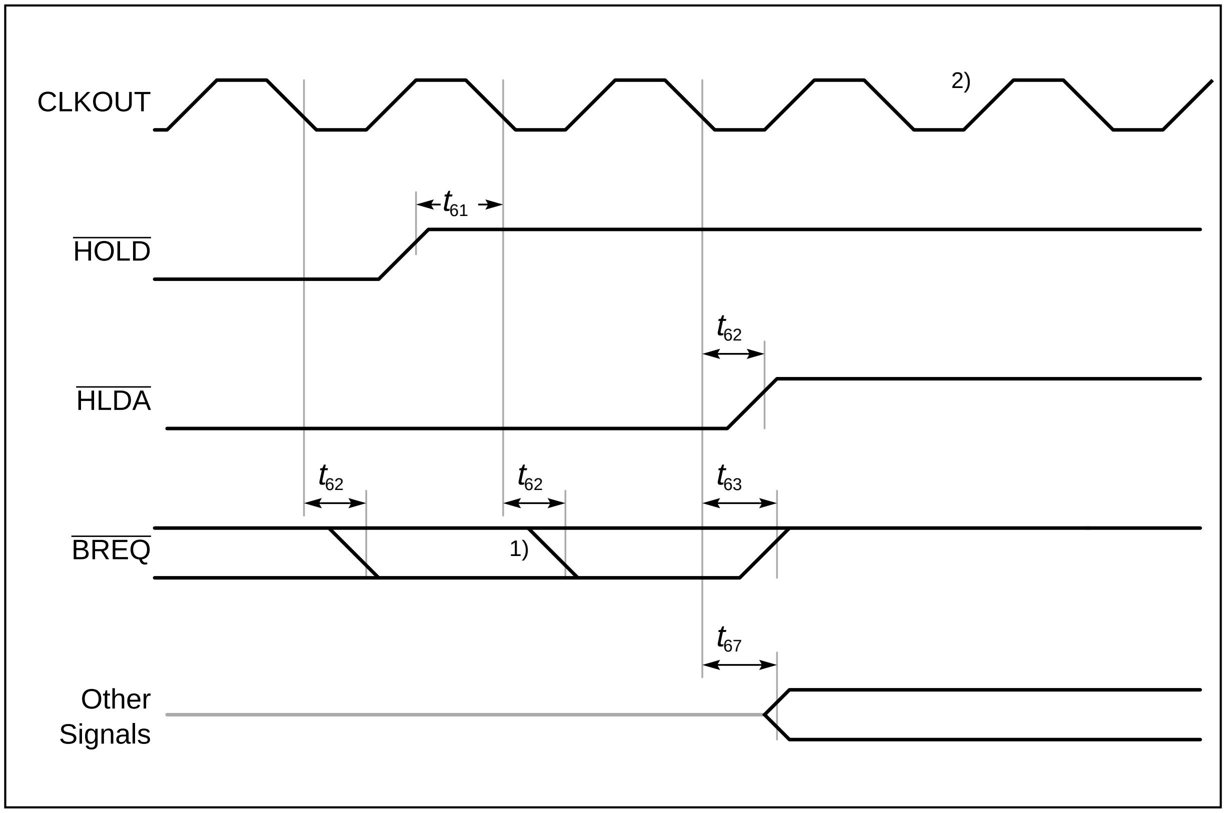


Figure 27
External Bus Arbitration, (Regaining the Bus)

Notes

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

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

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

Figure 28
Package Outline Rectangular P-MQFP100