SAB 80C166 Revision History:	Original Version 3.90
Previous Releases:	-
Page	Subjects (changes since last revision)

Note: Signals signified by a "#" are negated signals ($\overline{WR}$ = WR#)

Published by Siemens AG, Bereich Halbleiter, Marketing-Kommunikation
Balanstraße 73, D-8000 München 80.

SIEMENS

SAB 80C166/83C166

High-Performance 16-Bit CMOS Single-Chip Microcontrollers for Embedded Control Applications

Advance Information

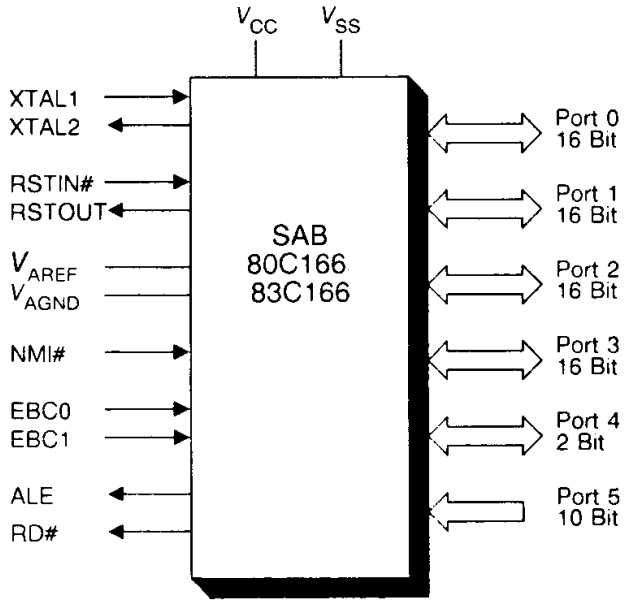
SAB 83C166 -3S 16-bit Microcontroller with 8K Mask-Programmable ROM
SAB 80C166 -S 16-bit Microcontroller for External Program Memory

- 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), 1µs Division (32-/16 bit)
- Enhanced Boolean Bit Manipulation Facilities
- Register-Based Design with Multiple Variable Register Banks
- Single-Cycle Context Switching Support
- 256 Kbytes Linear Address Space for Code and Data
- 1 Kbyte On-Chip RAM
- 8 Kbytes On-Chip ROM (for the SAB 83C166, only)
- 512 bytes On-Chip Special Function Register Area
- 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 9.75 µs Conversion Time
- 16-Channel Capture/Compare Unit
- 2 Multi-Functional General Purpose Timer Units
- 2 Serial Channels (USARTs)
- Programmable Watchdog Timer
- 76 General Purpose I/O Lines
- Temperature Range: 0 to 70 °C (in preparation: - 40 to 85 °C, - 40 to 110 °C)
- 1.2 micron Siemens Low-Power CMOS Process
- Complete Set of Development Tools 'C'-Compiler, Macro Assembler, Linker, Locater, Librarian, Simulator, Emulator, Evaluation Board
- 100 Pin Plastic Quad Flat Pack (PQFP) Package

Introduction

The SAB 80C166 and the SAB 83C166 are the first representatives of the new Siemens SAB 80C166 family of full featured single-chip CMOS micrcocontrollers. They combine high CPU performance (up to 10 million instructions per second) with high peripheral functionality.

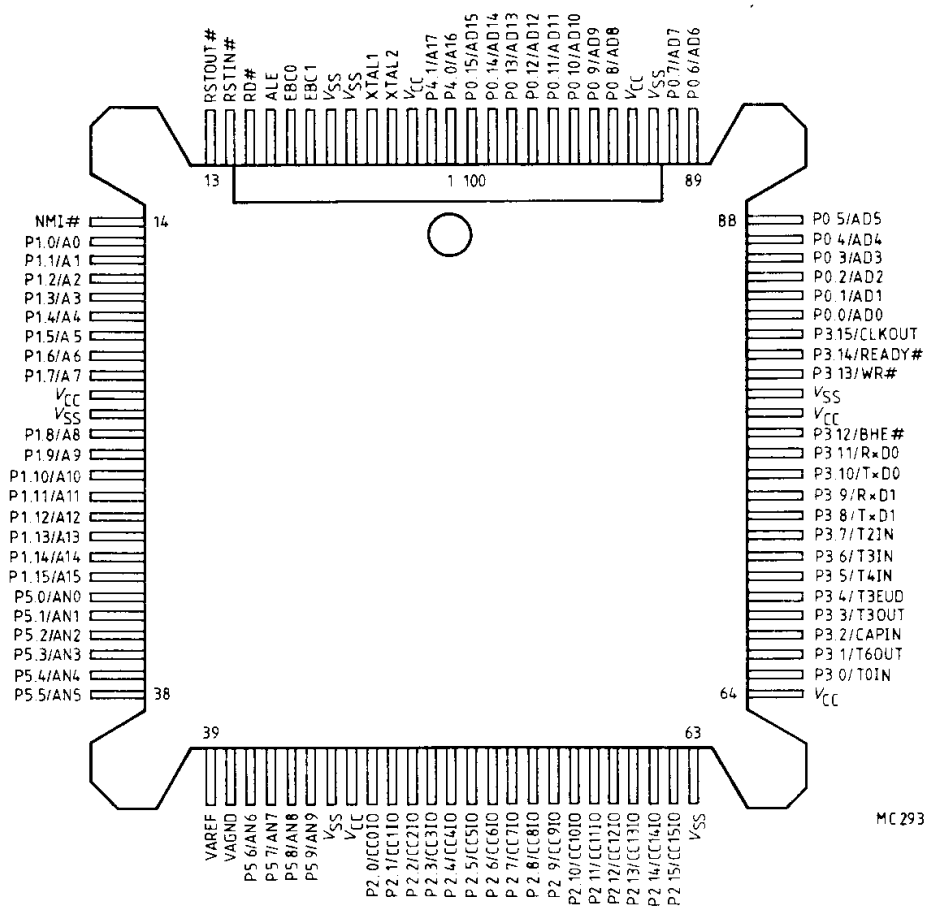
Figure 1
Logic Symbol



Ordering Information

Type	Ordering code	Package	Function
SAB 83C166-3S	Q67120-C552	P-QFP-100	16-bit microcontroller with 8K mask-programmable ROM
SAB 80C166-S	Q67120-C493	P-QFP-100	16-bit microcontroller for external program memory

Figure 2
Pin Configuration (Top View)



Pin Definitions and Functions

Symbol	Pin Number	Input (I) Output (O)	Function
P4.0 – P4.1	1,2	I/O	Port 4 is a 2-bit bidirectional I/O port. It is bit-wise programmable for input or output via a direction bit. For a pin configured as input, the output driver is put into high-impedance state. In case of an external bus configuration, Port 4 is used for output of the two segment address lines supposed that segmentation is enabled:
	1	O	P4.0 A16 Lower segment address line
	2	O	P4.1 A17 Higher segment address line
XTAL1	5	I	XTAL1: Input to the oscillator amplifier and input to the internal clock generator
XTAL2	4	O	XTAL2: Output of the oscillator amplifier circuits. To drive the device from an external source, XTAL1 should be driven while XTAL2 is left unconnected. Minimum and maximum high low and rise fall times specified in the AC Characteristics must be observed
EBC0 EBC1	9 8	I I	External Bus Configuration selection inputs These pins are sampled during reset and select either the single chip mode or one of the three external bus configurations, as follows:

EBC1	EBC0	Mode Bus Configuration
0	0	Single Chip Mode
0	1	16- 18-bit addresses, 8-bit data, multiplexed bus
1	0	16- 18-bit addresses, 16-bit data, multiplexed bus
1	1	16- 18-bit addresses, 16-bit data, non-multiplexed bus

RSTIN#	12	I	Reset Input with Schmitt-Trigger characteristics. A low level at this pin for a specified duration while the oscillator is running resets the SAB 80C166 83C166. An internal pullup resistor permits power-on reset using only a capacitor connected to V_{SS}.
RSTOUT#	13	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. RSTOUT# remains low until the EINIT (end of initialization) instruction is executed.
NMI#	14	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 NMI# pin must be low in order to force the SAB 80C166 83C166 to go into power down mode. If NMI# is high when PWRDN is executed, the part will continue to run in normal mode.

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input (I) Output (O)	Function
ALE	10	O	Address latch enable output. Can be used for latching the address into external memory or an address latch in the multiplexed bus modes.
RD#	11	O	External memory read control signal. RD# is activated for every external instruction or data read access.
P1.0 – P1.15	15 – 22 25 – 32	I/O	Port 1 is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via a direction bit. For a pin configured as input, the output driver is put into high-impedance state. Port 1 is also used as the 16-bit address bus (A) in the non-multiplexed bus mode.
	15 – 22 25 – 32	O O	16-bit address bus, non-multiplexed: P1.0 – P1.15 A0 – A15
P5.0 – P5.9	33 – 38 41 – 44	I I	Port 5 is a 10-bit input port with Schmitt-Trigger characteristics. The pins of Port 5 are also used as the analog input channels for the A/D converter as listed below:
	33 34 35 36 37 38 41 42 43 44	I I I I I I I I I I	P5.0 AN0 Analog input channel 0 P5.1 AN1 Analog input channel 1 P5.2 AN2 Analog input channel 2 P5.3 AN3 Analog input channel 3 P5.4 AN4 Analog input channel 4 P5.5 AN5 Analog input channel 5 P5.6 AN6 Analog input channel 6 P5.7 AN7 Analog input channel 7 P5.8 AN8 Analog input channel 8 P5.9 AN9 Analog input channel 9
P2.0 – P2.15	47 – 62	I/O	Port 2 is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via a direction bit. For a pin configured as input, the output driver is put into high-impedance state. The pins of Port 2 are also used as the capture inputs or compare outputs for the CAPCOM unit as listed in the following table:
	47 48 49 50 51	I/O I/O I/O I/O I/O	P2.0 CC0IO Register CC0 capture input/compare output P2.1 CC1IO Register CC1 capture input/compare output P2.2 CC2IO Register CC2 capture input/compare output P2.3 CC3IO Register CC3 capture input/compare output P2.4 CC4IO Register CC4 capture input/compare output

Pin Definitions and Functions (cont'd)

Symbol	Pin Number	Input (I) Output (O)	Function
P2.0 – P2.15 (cont'd)	52	I/O	P2.5 CC5IO Register CC5 capture input/compare output
	53	I/O	P2.6 CC6IO Register CC6 capture input/compare output
	54	I/O	P2.7 CC7IO Register CC7 capture input/compare output
	55	I/O	P2.8 CC8IO Register CC8 capture input/compare output
	56	I/O	P2.9 CC9IO Register CC9 capture input/compare output
	57	I/O	P2.10 CC10IO Register CC10 capture input/compare output
	58	I/O	P2.11 CC11IO Register CC11 capture input/compare output
	59	I/O	P2.12 CC12IO Register CC12 capture input/compare output
	60	I/O	P2.13 CC13IO Register CC13 capture input/compare output
	61	I/O	P2.14 CC14IO Register CC14 capture input/compare output
	62	I/O	P2.15 CC15IO Register CC15 capture input/compare output
P3.0 – P3.15	65 – 77	I/O	Port 3 is a 16-bit bidirectional I/O port. It is bit-wise programmable for input or output via a direction bit. For a pin configured as input, the output driver is put into high-impedance state. The pins of Port 3 are also used for various functions such as timer inputs and outputs and bus control signals. The alternate functions of the Port 3 pins are listed in the following:
	80 – 82	I/O	
	65	I	P3.0 T0IN Timer 0 count input
	66	O	P3.1 T6OUT Timer 6 toggle latch output
	67	I	P3.2 CAPIN CAPREL register capture input
	68	O	P3.3 T3OUT Timer 3 toggle latch output
	69	I	P3.4 T3EUD Timer 3 external up/down control input
	70	I	P3.5 T4IN Timer 4 count/gate/reload/capture input
	71	I	P3.6 T3IN Timer 3 count/gate input
	72	I	P3.7 T2IN Timer 2 count/gate/reload/capture input
	73	O	P3.8 TxD1 Serial Channel 1 clock output in synchronous mode; data output in asynchronous mode
	74	I/O	P3.9 RxD1 Serial Channel 1 data input/output in synchronous mode; data input in asynchronous mode
	75	O	P3.10 TxD0 Serial Channel 0 clock output in synchronous mode; data output in asynchronous mode
	76	I/O	P3.11 RxD0 Serial Channel 0 data input/output in synchronous mode; data input in asynchronous mode
	77	O	P3.12 BHE# External memory byte enable control signal
	80	O	P3.13 WR# External memory write control signal
	81	I	P3.14 READY# Ready input
	82	O	P3.15 CLKOUT System clock output (oscillator frequency/2)

Pin Definitions and Functions (cont'd)

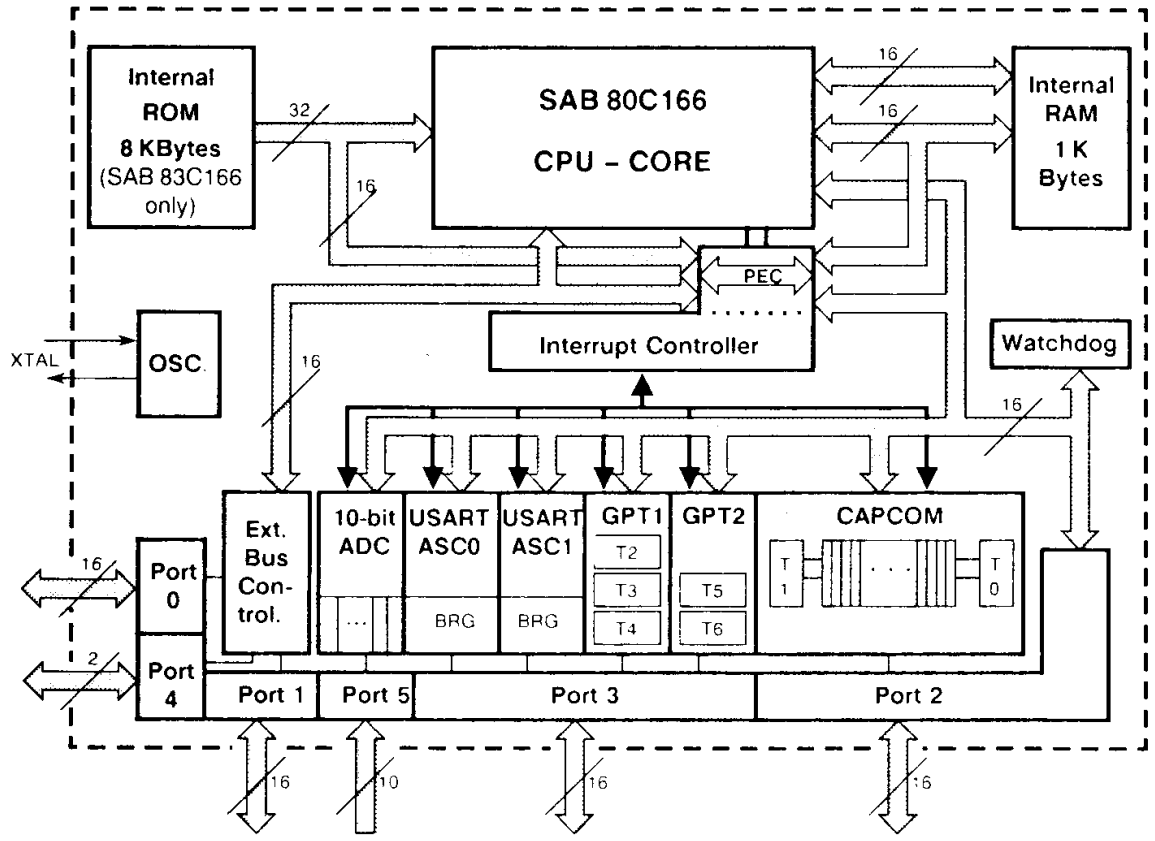
Symbol	Pin Number	Input (I) Output (O)	Function
P0.0 – P0.15	83 – 90 93 – 100	I/O O	Port 0 is a 16-bit bidirectional I/O port. In case of the Single Chip Mode, Port 0 is bit-wise programmable for input or output via a direction bit. For a pin configured as input, the output driver is put into high-impedance state. In case of an external bus configuration, Port 0 serves as the address (A) and address/data (AD) bus in the multiplexed bus modes and as the data (D) bus in the non-multiplexed bus mode. 16-/18-bit addresses, 8-bit data, multiplexed bus: P0.0 – P0.7 AD0 – AD7 P0.8 – P0.15 A8 – A15 16-/18-bit addresses, 16-bit data, multiplexed bus: P0.0 – P0.15 AD0 – AD15 16-/18-bit addresses, 16-bit data, non-multiplexed bus: P0.0 – P0.15 D0 – D15
	83 – 90 93 – 100	I/O O	
	83 – 90 93 – 100	I/O	
	83 – 90 93 – 100	I/O	
V_{CC}	3, 23, 46, 64, 78, 92		Digital supply voltage: + 5 V during normal operation and idle mode ≥ 2.5 V during power down mode
V_{SS}	6, 7, 24, 45, 63, 79, 91		Digital ground
V_{AREF}	39		Reference voltage for the A/D converter
V_{AGND}	40		Reference ground for the A/D converter

Functional Description

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

Note: In this description, any reference to the SAB 80C166 also applies to the SAB 83C166 unless otherwise noted. All time specifications refer to a CPU clock of 20 MHz which means an oscillator frequency (f_{OSC}) of 40 MHz.

Figure 3
Block Diagram



Memory Organization

The memory space of the SAB 80C166 is configured in a Von Neumann architecture which means that code memory, data memory, registers and I/O ports are organized within the same linear address space which currently includes 256 Kbytes. Address space expansion to 16 Mbytes is provided for future versions. The entire memory space can be accessed bytewise or wordwise. Particular portions of the on-chip memory have additionally been made directly bit addressable.

The SAB 83C166 contains 8 Kbytes of a mask-programmable on-chip ROM for code or constant data.

A large dual port RAM of 1 Kbyte is contained on both the SAB 80C166 and the SAB 83C166. This internal 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 (SFR) 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 to either the Single Chip Mode when no external memory is required, or to one of three different external memory access modes, which are as follows:

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

In the non-multiplexed bus mode, Port 1 is used as an output for addresses and Port 0 is used as an input output for data. In the multiplexed bus modes, just one of the two 16-bit ports, Port 0, is used as an input output for both addresses and data.

Important timing characteristics of the external bus interface (Memory Cycle Time, Memory Tri-State Time and Read write Delay) have been made programmable to allow the user the adaption of a wide range of different types of memories. Access to very slow memories is supported via a particular 'Ready' function.

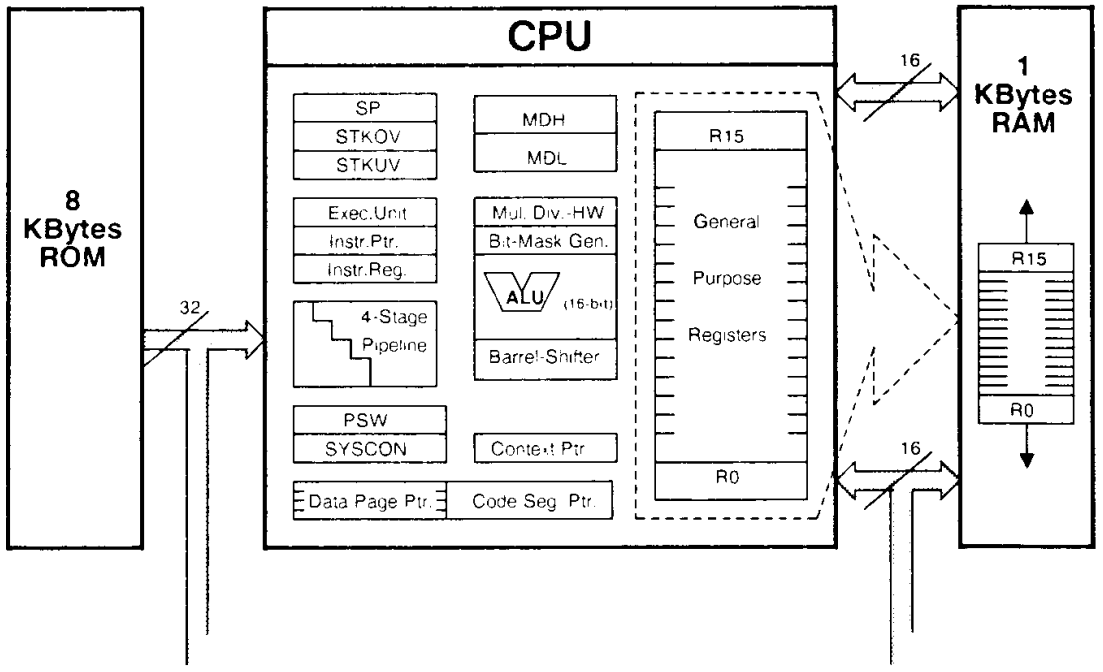
For applications which require less than 64 Kbytes of memory space, a non-segmented memory model can be selected. In this case, all memory locations can be addressed by 16 bits, and thus Port 4 is not needed as an output for the two most significant address bits (A17 and A16), as is the case when using the segmented memory model.

Central Processing Unit (CPU)

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

Based on these hardware provisions, most of the SAB 80C166's instructions can be executed in just one machine cycle which requires 100 ns at 20 MHz CPU clock. For example, shift and rotate instructions are always processed during one machine cycle independent of the number of bits to be shifted. All multiple-cycle instructions have been optimized so that they can be executed very fast as well: A 32- 16 bit division in 1µs, a 16*16 bit multiplication in 0.5 µs, and branches in 200 ns. Another pipeline optimization, the so called 'Jump Cache', allows reducing the execution time of repeatedly performed jumps in a loop from 200ns to100ns.

Figure 4
CPU Block Diagram



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

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 functional SAB 80C166 instruction set which includes the following instruction classes:

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

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

Interrupt System

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

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

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

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

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

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

Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
CAPCOM Register 0	CC0IR	CC0IE	CC0INT	40h	10h
CAPCOM Register 1	CC1IR	CC1IE	CC1INT	44h	11h
CAPCOM Register 2	CC2IR	CC2IE	CC2INT	48h	12h
CAPCOM Register 3	CC3IR	CC3IE	CC3INT	4Ch	13h
CAPCOM Register 4	CC4IR	CC4IE	CC4INT	50h	14h
CAPCOM Register 5	CC5IR	CC5IE	CC5INT	54h	15h
CAPCOM Register 6	CC6IR	CC6IE	CC6INT	58h	16h
CAPCOM Register 7	CC7IR	CC7IE	CC7INT	5Ch	17h
CAPCOM Register 8	CC8IR	CC8IE	CC8INT	60h	18h
CAPCOM Register 9	CC9IR	CC9IE	CC9INT	64h	19h
CAPCOM Register 10	CC10IR	CC10IE	CC10INT	68h	1Ah
CAPCOM Register 11	CC11IR	CC11IE	CC11INT	6Ch	1Bh
CAPCOM Register 12	CC12IR	CC12IE	CC12INT	70h	1Ch
CAPCOM Register 13	CC13IR	CC13IE	CC13INT	74h	1Dh
CAPCOM Register 14	CC14IR	CC14IE	CC14INT	78h	1Eh
CAPCOM Register 15	CC15IR	CC15IE	CC15INT	7Ch	1Fh
CAPCOM Timer 0	T0IR	T0IE	T0INT	80h	20h
CAPCOM Timer 1	T1IR	T1IE	T1INT	84h	21h
GPT 1 Timer 2	T2IR	T2IE	T2INT	88h	22h
GPT 1 Timer 3	T3IR	T3IE	T3INT	8Ch	23h
GPT 1 Timer 4	T4IR	T4IE	T4INT	90h	24h
GPT 2 Timer 5	T5IR	T5IE	T5INT	94h	25h
GPT 2 Timer 6	T6IR	T6IE	T6INT	98h	26h
GPT 2 CAPREL Register	CRIR	CRIE	CRINT	9Ch	27h
A D Conversion Complete	ADCIR	ADCIE	ADCINT	A0h	28h
A D Overrun Error	ADEIR	ADEIE	ADEINT	A4h	29h
Serial Channel 0 Transmit	S0TIR	S0TIE	S0TINT	A8h	2Ah
Serial Channel 0 Receive	S0RIR	S0RIE	S0RINT	ACh	2Bh
Serial Channel 0 Error	S0EIR	S0EIE	S0EINT	B0h	2Ch
Serial Channel 1 Transmit	S1TIR	S1TIE	S1TINT	B4h	2Dh
Serial Channel 1 Receive	S1RIR	S1RIE	S1RINT	B8h	2Eh
Serial Channel 1 Error	S1EIR	S1EIE	S1EINT	BCh	2Fh

The SAB 80C166 also provides an excellent mechanism to identify and to process exceptions or error conditions that arise during run-time, so called 'Hardware Traps'. Hardware traps cause immediate non-maskable system reaction which is similiar to a standard interrupt service (branching to a dedicated vector table location). The occurrence of a hardware trap is additionally signified by an individual bit in the trap flag register (TFR). Except another higher prioritized trap service being 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	0h	0h	III
Software Reset		RESET	0h	0h	III
Watchdog Timer Overflow		RESET	0h	0h	III
Class A Hardware Traps:					
Non-Maskable Interrupt	NMI	NMITRAP	08h	2h	II
Stack Overflow	STKOF	STOTRAP	10h	4h	II
Stack Underflow	STKUF	STUTRAP	18h	6h	II
Class B Hardware Traps:					
Undefined Opcode	UNDOPC	BTRAP	28h	Ah	I
Protected Instruction Fault	PRTFLT	BTRAP	28h	Ah	I
Illegal Word Operand Access	ILLOPA	BTRAP	28h	Ah	I
Illegal Instruction Access	ILLINA	BTRAP	28h	Ah	I
Illegal External Bus Access	ILLBUS	BTRAP	28h	Ah	I
Reserved			[2Ch – 3Ch]	[8h – Fh]	
Software Traps TRAP Instruction			Any [0h - 1FCh] in steps of 4H	Any [0h - 7Fh]	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. The CAPCOM unit is typically used to handle high speed I/O tasks such as pulse and waveform generation, pulse width modulation (PWM), 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.

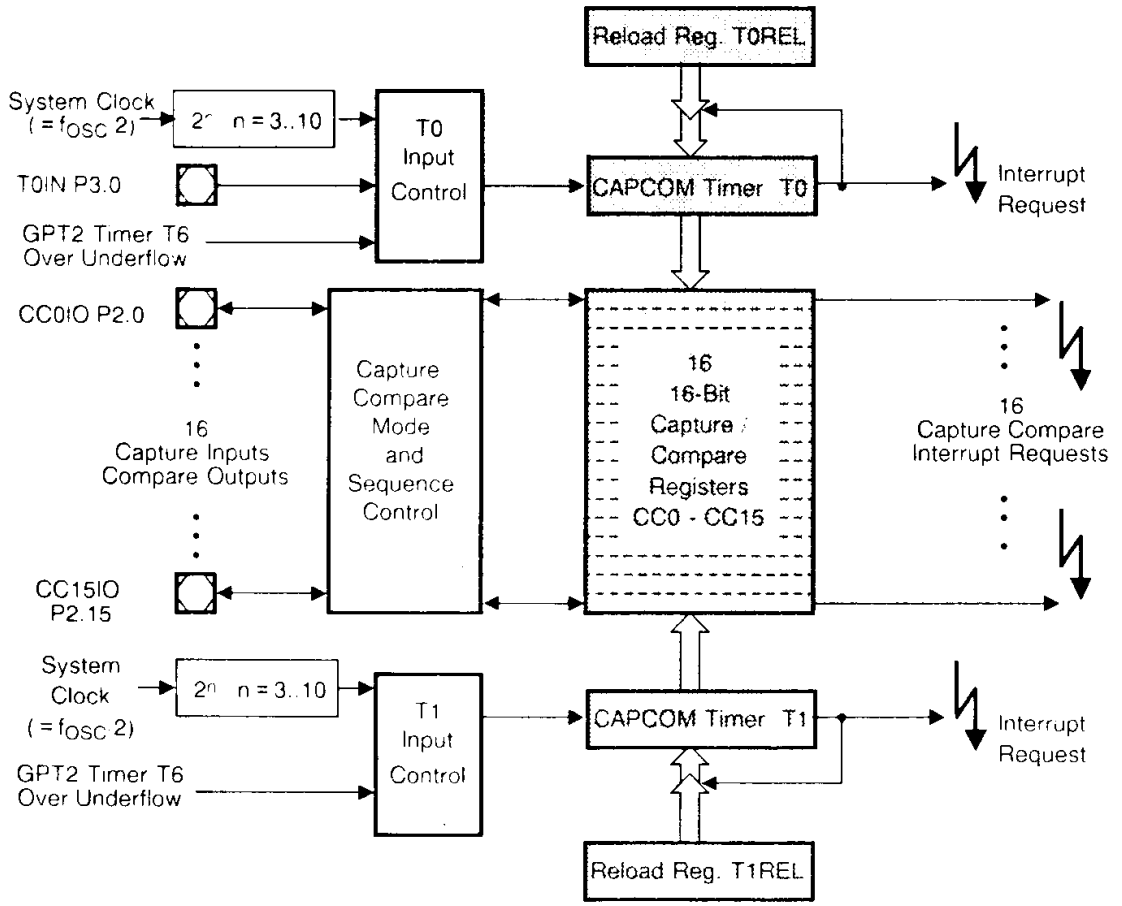
The input clock for the timers is programmable to several prescaled values of the internal system clock, or may be derived from an overflow/underflow of timer T6 in module GPT2. This provides a wide range of variation for the timer period and resolution and allows precise adjustment to the application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

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

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

Compare Modes	Function
Mode 0	Interrupt-only compare mode: several compare interrupts per timer period are possible
Mode 1	Pin toggles on each compare match; several compare events per timer period are possible
Mode 2	Interrupt-only compare mode: only one compare interrupt per timer period is generated
Mode 3	Pin set to '1' on match; pin reset to '0' on compare timer 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 measurement, 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 the GPT1 module 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 internal system 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 (T2IN, T3IN, T4IN) which serves as gate or clock input. The maximum resolution of the timers in the GPT1 module is 400 ns (@ f_{OSC} = 40 MHz).

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 time out monitoring of external hardware components, or may be used internally to clock timers T2 and T4 for measuring long time periods with high resolution.

In addition to their basic operating modes, timers T2 and T4 may be configured as reload or capture registers for timer T3. When used as capture or reload registers, timers T2 and T4 are stopped. The contents of timer T3 are captured into T2 or T4 in response to a signal at their associated input pins (T2IN, T4IN). Timer T3 is reloaded with the contents of T2 or T4 either by an external signal or by a selectable state transition of its toggle latch T3OTL. When both T2 and T4 are configured to alternately reload T3 on opposite state transitions of T3OTL with the low and high times of a PWM signal, this signal can be constantly generated without software intervention.

With its maximum resolution of 200 ns (@ f_{OSC} = 40 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 independently count up or down, clocked with an input clock which is derived from a programmable prescaler. Concatenation of the timers is supported via the output toggle latch (T6OTL) of timer T6, which changes its state on each timer overflow/underflow.

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

Figure 6
Block Diagram of GPT1

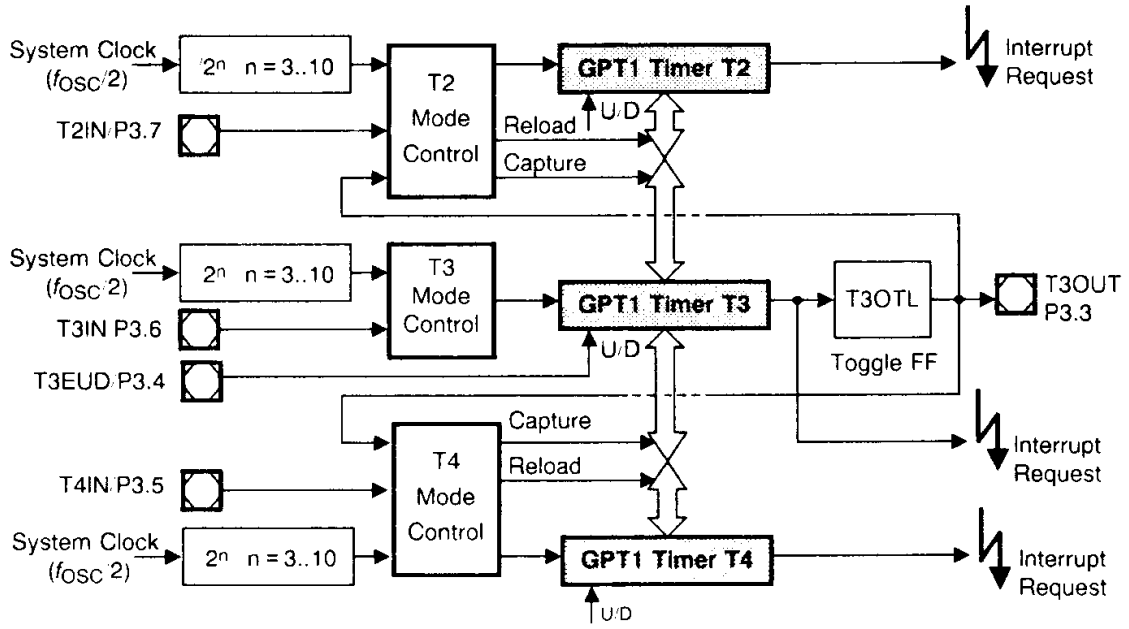
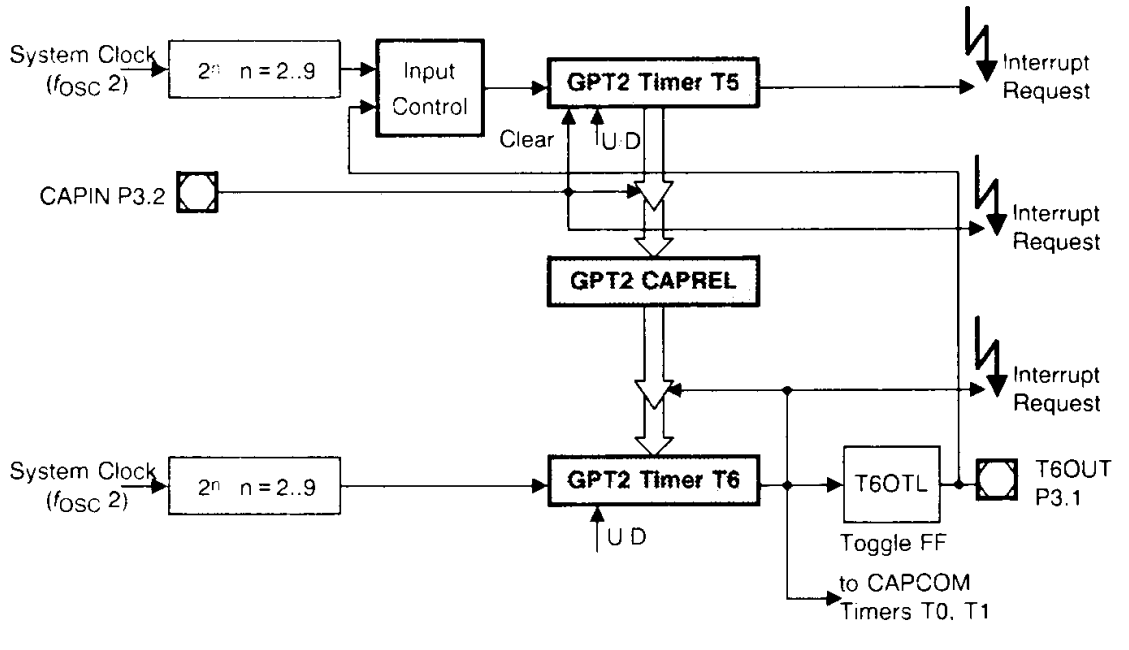


Figure 7
Block Diagram of GPT2



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 which returns the conversion result for an analog channel within 9.75 µs (@ f_{OSC} = 40 MHz).

Overrun error detection capability 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 channels can be used as digital input port pins.

The A/D converter of the SAB 80C166 supports four different conversion modes. In the standard Single Channel conversion mode, the analog level on a specified channel is once sampled and converted into a digital result. In the Single Channel Continous mode, the analog level 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.

Serial Channels

Serial communication with other microcontrollers, processors, terminals, or external peripheral components is provided by two serial interfaces with identical functionality, Serial Channel 0 (ASC0) and Serial Channel 1 (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.

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 the synchronous mode, one data byte is transmitted or received synchronously to a shift clock which is generated by the SAB 80C166. In the asynchronous mode, an 8- or 9-bit data frame is 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), and 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 reception of a new character is complete.

Watchdog Timer

The Watchdog Timer of the SAB 80C166 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 of the SAB 80C166 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. When the software has been designed to service the Watchdog Timer before it overflows, the Watchdog Timer times out if the program does not progress properly due to hardware or software related failures. When the Watchdog Timer overflows, it generates an internal hardware reset and pulls the RSTOUT# pin low in order to allow external hardware components to reset.

The Watchdog Timer of the SAB 80C166 is a 16-bit timer which can either be clocked with $f_{OSC}/4$ or $f_{OSC}/256$. 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 (@ f_{OSC} = 40 MHz). The default Watchdog Timer interval after reset is 6.55 ms.

Parallel Ports

The SAB 80C166 provides 76 I/O lines which are organized into four 16-bit I/O ports (Port 0 through 3), one 2-bit I/O port (Port 4), and one 10-bit input port (Port 5). All port lines are bit addressable, and all lines of Port 0 through 4 are individually bit-wise programmable as inputs or outputs via direction registers. The I/O ports are true bidirectional ports which are switched to the high impedance state when configured as inputs. During the internal reset, all port pins are configured as inputs.

Each port line has one programmable alternate input or output function associated with it. Ports 0 and 1 may be used as address and data lines when accessing external memory, while Port 4 outputs the additional segment address bits A16 and A17 in systems where segmentation is enabled to access more than 64 Kbytes of memory. Port 2 is associated with the capture inputs/compare outputs of the CAPCOM unit, and Port 3 includes alternate functions of timers, serial interfaces, optional bus control signals (WR#, BHE#, READY#), and the system clock output (CLKOUT). Port 5 is used for the analog input channels to the A/D converter. When anyone of these alternate functions is not used, the respective port line may be used as general purpose I/O line.

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

Instruction Set Summary

Mnemonic		Description	Bytes
Arithmetic operations (cont'd)			
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
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

Instruction Set Summary

Mnemonic		Description	Bytes
Arithmetic operations (cont'd)			
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
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	Unigned 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

Instruction Set Summary

Mnemonic		Description	Bytes
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
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

Instruction Set Summary

Mnemonic		Description	Bytes
Logical Instructions (cont'd)			
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

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

Instruction Set Summary

Mnemonic		Description	Bytes
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
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

Instruction Set Summary

Mnemonic		Description	Bytes
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
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

Instruction Set Summary

Mnemonic		Description	Bytes
Data Movement (cont'd)			
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
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, #data16	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

Instruction Set Summary

Mnemonic		Description	Bytes
Data Movement (cont'd)			
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
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

Instruction Set Summary

Mnemonic		Description	Bytes
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 and update register with immediate data	4
SCXT	reg, mem	Push direct word register onto system stack and update register with direct memory	4
Return operations			
RET		Return from intra-segment subroutine	2
RETS		Return from inter-segment subroutine	2
RETP	reg	Return from intra-segment subroutine and pop direct word register from system stack	2
RETI		Return from interrupt service subroutine	2
System Control			
SRST		Software Reset	4
IDLE		Enter Idle Mode	4
PWRDN		Enter Power Down Mode (supposes NMI#-Pin being low)	4
SRVWDT		Service Watchdog Timer	4
DISWDT		Disable Watchdog Timer	4
EINIT		Signify End-of-Initialization on RSTOUT-pin	4
Miscellaneous			
NOP		Null operation	2

Instruction Set Summary

Notes

Data addressing modes

Rw: – Word GPR (R0, R1. ..., R15)

Rb: – Byte GPR (RL0, RH0. ..., RL7, RH7)

reg: – SFR or GPR
(in case of a byte operation on an SFR, only the low byte can be accessed via 'reg')

mem: – Direct word or byte memory location

[...]: – Indirect word or byte memory location
(Any word GPR can be used as indirect address pointer, except for the arithmetic, logical and compare instructions, where only R0 to R3 are allowed)

bitaddr: – Direct bit in the bit-addressable memory area.

bitoff: – Direct word in the bit-addressable memory area.

#data$_i$: – Immediate constant
(The number of significant bits which can be specified by the user is represented by the respective index 'i')

#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

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, #data16
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
33	4	SUBCB	reg, mem
34	4	SUBC	mem, reg
35	4	SUBCB	mem, reg

For notes see page 37

Instruction Op Codes in Hexadecimal Order (cont'd)

Hex-code	Number of bytes	Mnemonic	Operands
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

Hex-code	Number of bytes	Mnemonic	Operands
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
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

For notes see page 37

Instruction Op Codes in Hexadecimal Order (cont'd)

Hex-code	Number of bytes	Mnemonic	Operands
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
A1	2	NEGB	Rb
A2	4	CMPD1	Rw, mem
A3	-	-----	-----
A4	4	MOVB	[Rw], mem
A5	4	DISWDT	

For notes see page 37

Instruction Op Codes in Hexadecimal Order (cont'd)

Hex-code	Number of bytes	Mnemonic	Operands	Hex-code	Number of bytes	Mnemonic	Operands
A6	4	CMPD1	Rw, #data16	C6	4	SCXT	reg, #data16
A7	4	SRVWDT		C7	-	-----	-----
A8	2	MOV	Rw, [Rw]	C8	2	MOV	[Rw], [Rw]
A9	2	MOVB	Rb, [Rw]	C9	2	MOVB	[Rw], [Rw]
AA	4	JBC	bitaddr, rel	CA	4	CALLA	cc, caddr
AB	2	CALLI	cc, [Rw]	CB	2	RET	
AC	2	ASHR	Rw, Rw	CC	2	NOP	
AD	2	JMPR	cc_SGT, rel	CD	2	JMPR	cc_SLT, rel
AE	2	BCLR	bitoff.10	CE	2	BCLR	bitoff.12
AF	2	BSET	bitoff.10	CF	2	BSET	bitoff.12
B0	2	CMPD2	Rw, #data4	D0	2	MOVBS	Rw, Rb
B1	2	CPLB	Rb	D1	-	-----	-----
B2	4	CMPD2	Rw, mem	D2	4	MOVBS	reg, mem
B3	-	-----	-----	D3	-	-----	-----
B4	4	MOVB	mem, [Rw]	D4	4	MOV	Rw, [Rw + #data16]
B5	4	EINIT		D5	4	MOVBS	mem, reg
B6	4	CMPD2	Rw, #data16	D6	4	SCXT	reg, mem
B7	4	SRST		D7	-	-----	-----
B8	2	MOV	[Rw], Rw	D8	2	MOV	[Rw +], [Rw]
B9	2	MOVB	[Rw], Rb	D9	2	MOVB	[Rw +], [Rw]
BA	4	JNBS	bitaddr, rel	DA	4	CALLS	seg, caddr
BB	2	CALLR	rel	DB	2	RETS	
BC	2	ASHR	Rw, #data4	DC	-	-----	-----
BD	2	JMPR	cc_SLE, rel	DD	2	JMPR	cc_SGE, rel
BE	2	BCLR	bitoff.11	DE	2	BCLR	bitoff.13
BF	2	BSET	bitoff.11	DF	2	BSET	bitoff.13
C0	2	MOVBZ	Rw, Rb	E0	2	MOV	Rw, #data4
C1	-	-----	-----	E1	2	MOVB	Rb, #data4
C2	4	MOVBZ	reg, mem	E2	4	PCALL	reg, caddr
C3	-	-----	-----	E3	-	-----	-----
C4	4	MOV	[Rw + #data16], Rw	E4	4	MOVB	[Rw + #data16], Rb
C5	4	MOVBZ	mem, reg	E5	-	-----	-----

For notes see page 37

Instruction Op Codes in Hexadecimal Order (cont'd)

Hex-code	Num-ber of bytes	Mnemonic	Operands
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 format. Thus, these instructions can be differentiated by means of the second byte, as follows:

x0h - x7h :	Rw, #data3	or	Rb, #data3
x8h - xBh :	Rw, [Rw]	or	Rb, [Rw]
xCh - xFh :	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-adressable 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 80C166 in alphabetical order. Bit-addressable SFRs are marked with the letter "b" in column "Name". An SFR can be specified via its individual mnemonic name. Depending on the selected addressing mode, an SFR can be accessed via its physical address (using the Data Page Pointers), or via its short 8-bit address (without using the Data Page Pointers).

Name	Physical Address	8-bit Address	Description	Reset Value
ADCIC b	FF98h	CCh	A D Converter End of Conversion Interrupt Control Register	0000h
ADCON b	FFA0h	D0h	A D Converter Control Register	0000h
ADDAT	FEA0h	50h	A D Converter Result Register	0000h
ADEIC b	FF9Ah	CDh	A D Converter Overrun Error Interrupt Control Register	0000h
CAPREL	FE4Ah	25h	GPT2 Capture Reload Register	0000h
CC0	FE80h	40h	CAPCOM Register 0	0000h
CC0IC b	FF78h	BCh	CAPCOM Register 0 Interrupt Control Register	0000h
CC1	FE82h	41h	CAPCOM Register 1	0000h
CC1IC b	FF7Ah	BDh	CAPCOM Register 1 Interrupt Control Register	0000h
CC2	FE84h	42h	CAPCOM Register 2	0000h
CC2IC b	FF7Ch	BEh	CAPCOM Register 2 Interrupt Control Register	0000h
CC3	FE86h	43h	CAPCOM Register 3	0000h
CC3IC b	FF7Eh	BFh	CAPCOM Register 3 Interrupt Control Register	0000h
CC4	FE88h	44h	CAPCOM Register 4	0000h
CC4IC b	FF80h	C0h	CAPCOM Register 4 Interrupt Control Register	0000h
CC5	FE8Ah	45h	CAPCOM Register 5	0000h
CC5IC b	FF82h	C1h	CAPCOM Register 5 Interrupt Control Register	0000h
CC6	FE8Ch	46h	CAPCOM Register 6	0000h
CC6IC b	FF84h	C2h	CAPCOM Register 6 Interrupt Control Register	0000h
CC7	FE8Eh	47h	CAPCOM Register 7	0000h
CC7IC b	FF86h	C3h	CAPCOM Register 7 Interrupt Control Register	0000h
CC8	FE90h	48h	CAPCOM Register 8	0000h
CC8IC b	FF88h	C4h	CAPCOM Register 8 Interrupt Control Register	0000h
CC9	FE92h	49h	CAPCOM Register 9	0000h
CC9IC b	FF8Ah	C5h	CAPCOM Register 9 Interrupt Control Register	0000h
CC10	FE94h	4Ah	CAPCOM Register 10	0000h
CC10IC b	FF8Ch	C6h	CAPCOM Register 10 Interrupt Control Register	0000h

Special Function Registers Overview (cont'd)

Name	Physical Address	8-bit Address	Description	Reset Value
CC11	FE96h	4Bh	CAPCOM Register 11	0000h
CC11IC b	FF8Eh	C7h	CAPCOM Register 11 Interrupt Control Register	0000h
CC12	FE98h	4Ch	CAPCOM Register 12	0000h
CC12IC b	FF90h	C8h	CAPCOM Register 12 Interrupt Control Register	0000h
CC13	FE9Ah	4Dh	CAPCOM Register 13	0000h
CC13IC b	FF92h	C9h	CAPCOM Register 13 Interrupt Control Register	0000h
CC14	FE9Ch	4Eh	CAPCOM Register 14	0000h
CC14IC b	FF94h	CAh	CAPCOM Register 14 Interrupt Control Register	0000h
CC15	FE9Eh	4Fh	CAPCOM Register 15	0000h
CC15IC b	FF96h	CBh	CAPCOM Register 15 Interrupt Control Register	0000h
CCM0 b	FF52h	A9h	CAPCOM Mode Control Register 0	0000h
CCM1 b	FF54h	AAh	CAPCOM Mode Control Register 1	0000h
CCM2 b	FF56h	ABh	CAPCOM Mode Control Register 2	0000h
CCM3 b	FF58h	ACh	CAPCOM Mode Control Register 3	0000h
CP	FE10h	08h	CPU Context Pointer Register	FC00h
CRIC b	FF6Ah	B5h	GPT2 CAPREL Interrupt Control Register	0000h
CSP	FE08h	04h	CPU Code Segment Pointer Register (2 bits, read only)	0000h
DP0 b	FF02h	81h	Port 0 Direction Control Register	0000h
DP1 b	FF06h	83h	Port 1 Direction Control Register	0000h
DP2 b	FFC2h	E1h	Port 2 Direction Control Register	0000h
DP3 b	FFC6h	E3h	Port 3 Direction Control Register	0000h
DP4 b	FF0Ah	85h	Port 4 Direction Control Register (2 bits)	0000h
DPP0	FE00h	00h	CPU Data Page Pointer 0 Register (4 bits)	0000h
DPP1	FE02h	01h	CPU Data Page Pointer 1 Register (4 bits)	0001h
DPP2	FE04h	02h	CPU Data Page Pointer 2 Register (4 bits)	0002h
DPP3	FE06h	03h	CPU Data Page Pointer 3 Register (4 bits)	0003h
MDC b	FF0Eh	87h	CPU Multiply Divide Control Register	0000h
MDH	FE0Ch	06h	CPU Multiply Divide Register – High Word	0000h
MDL	FE0Eh	07h	CPU Multiply Divide Register – Low Word	0000h
ONES	FF1Eh	8Fh	Constant Value 1's Register (read only)	FFFFh

Special Function Registers Overview (cont'd)

Name	Physical Address	8-bit Address	Description	Reset Value
P0 b	FF00h	80h	Port 0 Register	0000h
P1 b	FF04h	82h	Port 1 Register	0000h
P2 b	FFC0h	E0h	Port 2 Register	0000h
P3 b	FFC4h	E2h	Port 3 Register	0000h
P4 b	FF08h	84h	Port 4 Register (2 bits)	0000h
P5 b	FFA2h	D1h	Port 5 Register (10 bits, read only)	XXXXh
PECC0	FEC0h	60h	PEC Channel 0 Control Register	0000h
PECC1	FEC2h	61h	PEC Channel 1 Control Register	0000h
PECC2	FEC4h	62h	PEC Channel 2 Control Register	0000h
PECC3	FEC6h	63h	PEC Channel 3 Control Register	0000h
PECC4	FEC8h	64h	PEC Channel 4 Control Register	0000h
PECC5	FECAh	65h	PEC Channel 5 Control Register	0000h
PECC6	FECCh	66h	PEC Channel 6 Control Register	0000h
PECC7	FECEh	67h	PEC Channel 7 Control Register	0000h
PSW b	FF10h	88h	CPU Program Status Word	0000h
S0BG	FEB4h	5Ah	Serial Channel 0 Baud Rate Generator Reload Register	0000h
S0CON b	FFB0h	D8h	Serial Channel 0 Control Register	0000h
S0EIC b	FF70h	B8h	Serial Channel 0 Error Interrupt Control Register	0000h
S0RBUF	FEB2h	59	Serial Channel 0 Receive Buffer Register (read only)	XXXXh
S0RIC b	FF6Eh	B7h	Serial Channel 0 Receive Interrupt Control Register	0000h
S0TBUF	FEB0h	58h	Serial Channel 0 Transmit Buffer Register (write only)	0000h
S0TIC b	FF6Ch	B6h	Serial Channel 0 Transmit Interrupt Control Register	0000h
S1BG	FEBCh	5Eh	Serial Channel 1 Baud Rate Generator Reload Register	0000h
S1CON b	FFB8h	DCh	Serial Channel 1 Control Register	0000h
S1EIC b	FF76h	BBh	Serial Channel 1 Error Interrupt Control Register	0000h
S1RBUF	FEBAh	5Dh	Serial Channel 1 Receive Buffer Register (read only)	XXXXh
S1RIC b	FF74h	BAh	Serial Channel 1 Receive Interrupt Control Register	0000h
S1TBUF	FEB8h	5Ch	Serial Channel 1 Transmit Buffer Register (write only)	0000h
S1TIC b	FF72h	B9h	Serial Channel 1 Transmit Interrupt Control Register	0000h
SP	FE12h	09h	CPU System Stack Pointer Register	FC00h

Special Function Registers Overview (cont'd)

Name	Physical Address	8-bit Address	Description	Reset Value
STKOV	FE14h	0Ah	CPU Stack Overflow Pointer Register	FA00h
STKUN	FE16h	0Bh	CPU Stack Underflow Pointer Register	FC00h
SYSCON b	FF0Ch	86h	CPU System Configuration Register	0XX0h*)
T0	FE50h	28h	CAPCOM Timer 0 Register	0000h
T01CON b	FF50h	A8h	CAPCOM Timer 0 and Timer 1 Control Register	0000h
T0IC b	FF9Ch	CEh	CAPCOM Timer 0 Interrupt Control Register	0000h
T0REL	FE54h	2Ah	CAPCOM Timer 0 Reload Register	0000h
T1	FE52h	29h	CAPCOM Timer 1 Register	0000h
T1IC b	FF9Eh	CFh	CAPCOM Timer 1 Interrupt Control Register	0000h
T1REL	FE56h	2Bh	CAPCOM Timer 1 Reload Register	0000h
T2	FE40h	20h	GPT1 Timer 2 Register	0000h
T2CON b	FF40h	A0h	GPT1 Timer 2 Control Register	0000h
T2IC b	FF60h	B0h	GPT1 Timer 2 Interrupt Control Register	0000h
T3	FE42h	21h	GPT1 Timer 3 Register	0000h
T3CON b	FF42h	A1h	GPT1 Timer 3 Control Register	0000h
T3IC b	FF62h	B1h	GPT1 Timer 3 Interrupt Control Register	0000h
T4	FE44h	22h	GPT1 Timer 4 Register	0000h
T4CON b	FF44h	A2h	GPT1 Timer 4 Control Register	0000h
T4IC b	FF64h	B2h	GPT1 Timer 4 Interrupt Control Register	0000h
T5	FE46h	23h	GPT2 Timer 5 Register	0000h
T5CON b	FF46h	A3h	GPT2 Timer 5 Control Register	0000h
T5IC b	FF66h	B3h	GPT2 Timer 5 Interrupt Control Register	0000h
T6	FE48h	24h	GPT2 Timer 6 Register	0000h
T6CON b	FF48h	A4h	GPT2 Timer 6 Control Register	0000h
T6IC b	FF68h	B4h	GPT2 Timer 6 Interrupt Control Register	0000h
TFR b	FFACh	D6h	Trap Flag Register	0000h
WDT	FEAEh	57h	Watchdog Timer Register (read only)	0000h
WDTCON	FFAEh	D7h	Watchdog Timer Control Register	0000h
ZEROS b	FF1Ch	8Eh	Constant Value 0's Register (read only)	0000h

*) system configuration selected during reset

Absolute Maximum Ratings

Ambient temperature under bias (T_A) . . . . . 0 to + 70 °C
Storage temperature (T_{ST}) . . . . . – 65 to + 150 °C
Supply Voltage (V_{CC}) . . . . . + 6.5 V
Input voltage (V_{IN} min. = – 3.0 V for pulse width less than 15 ns) . – 0.5 to V_{CC} + 0.5 V
Power dissipation . . . . . tbd

Notes *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.*

The SAB 80C166 will also be offered in the temperature ranges - 40 to + 110 °C and - 40 to + 85 °C.

All of the following time specifications refer to a CPU clock of 20 MHz which is identical to an oscillator frequency (f_{OSC}) of 40MHz.

DC Characteristics

T_A = 0 to + 70 °C; V_{CC} = 5 V ± 10 %; V_{SS} = 0 V

Parameter	Symbol	Limit values		Unit	Test condition
		min.	max.		
Input Low Voltage	V_{IL}	– 0.5	0.2 V_{CC} – 0.1	V	–
Input High Voltage (all except RSTIN# and XTAL1)	V_{IH}	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V	–
Input High Voltage RSTIN#	V_{IH1}	0.6 V_{CC}	V_{CC} + 0.5	V	–
Input High Voltage XTAL1	V_{IH2}	0.7 V_{CC}	V_{CC} + 0.5	V	–
Output Low Voltage (Ports 0, 1, 4, ALE, RD#, WR#, BHE#, CLKOUT, RSTOUT#)	V_{OL}	–	0.4	V	I_{OL} = 2.4 mA
Output Low Voltage (all other outputs)	V_{OL1}	–	0.4	V	I_{OL1} = 1.6 mA
Output High Voltage (Ports 0, 1, 4, ALE, RD#, WR#, BHE#, CLKOUT, RSTOUT#)	V_{OH}	0.9 V_{CC} 2.4	–	V	I_{OH} = – 100 µA I_{OH} = – 2.4 mA
Output High Voltage (all other outputs)	V_{OH1}	0.9 V_{CC} 2.4	–	V V	I_{OH} = – 50 µA I_{OH} = – 1.6 mA
Input Leakage Current (Ports 0, 1, 2, 3, 4, NMI#, EBC0, EBC1)	I_{OZ}	–	± 1	µA	0 V < V_{in} < V_{CC}

DC Characteristics (cont'd)

Parameter	Symbol	Limit values		Unit	Test condition
		min.	max.		
Reset Pullup Resistor	R_{RST}	50	150	kΩ	–
XTAL1 Input Current	I_{IL}	–	tbd	µA	$0\ V < V_{in} < V_{CC}$
Pin Capacitance (digital inputs/outputs)	C_{IO}	–	10	pF	$f = 1$ MHz $T_A = 25°C$
Power Supply Current	I_{CC}	–	180	mA	1/TCL = 40 MHz
Idle Mode Supply Current	I_{ID}	–	20	mA	1/TCL = 40 MHz
Power Down Mode Supply Current	I_{PD}	–	100	µA	V_{CC} = 2.5 V 1)

A/D Converter Characteristics

T_A = 0 to +70 °C; V_{CC} = 5 V ± 10 %; V_{SS} = 0 V; $V_{AREF} = V_{CC} \pm 0.2$ V; $V_{AGND} = V_{SS} \pm 0.2$ V

Parameter	Symbol	Limit values		Unit	Test condition
		min.	max.		
Analog Input Voltage	V_{AIN}	$V_{SS} - 0.2$	$V_{CC} + 0.2$	V	–
Analog Input Capacitance	C_I	–	70	pF	–
Sample Time	t_S	–	63 TCL		2)
Conversion Time	t_C	–	390 TCL		3)
Total Unadjusted Error	TUE	–	± 2	LSB	–
V_{AREF} Supply Current	I_{REF}	–	5	mA	4)
Analog Input Current	I_{AIN}	–	± 500	nA	5)

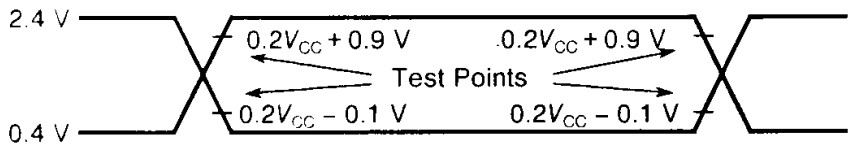
Notes

1) 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_{AREF} = 0 V, all outputs (including pins configured as outputs) disconnected.
2) This parameter specifies the time during which the input capacitance C_I can be charged/decharged by the external source. It must be guaranteed, that the input capacitance C_I is fully loaded within these 63 TCLs. 63 TCL is 1.575 µs at 20 MHz CPU clock. After the end of the sample time t_S, changes of the analog input voltage have no effect on the conversion result.
3) This parameter includes the sample time TS. 390 TCL is 9.75 µs at 20 MHz CPU clock.
4) I_{REF} in Power Down Mode: TBD
5) This parameter specifies the static input current for an analog input channel, e.g. when the channel is not selected for conversion.

AC Characteristics

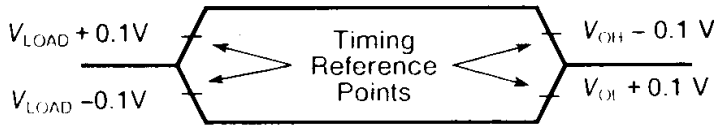
Testing Waveforms

Figure 8
Input Output 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
Float Waveforms

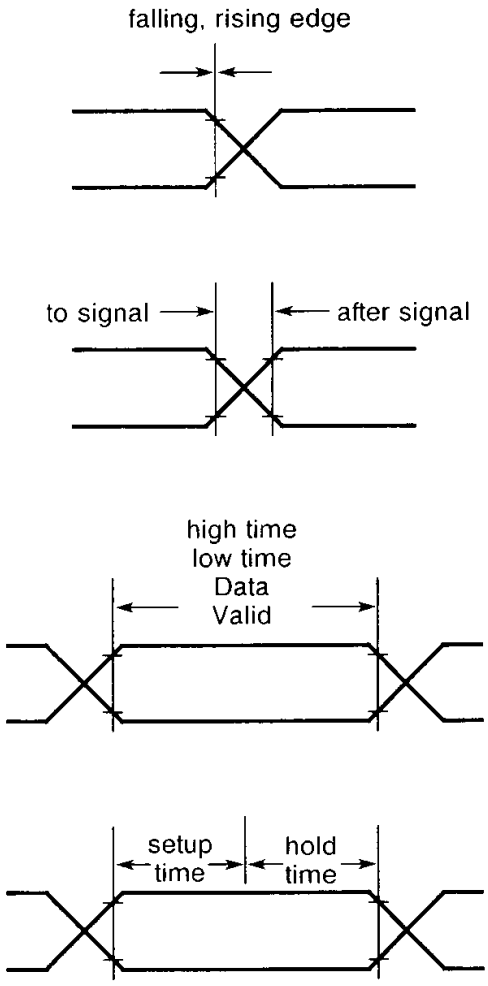


For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs, but begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs (I_{OH}/I_{OL} = 20 mA).

AC Characteristics (cont'd)

In the AC Characteristics waveforms, the mid-point of a signal transition is mostly used as the timing reference point. If not specifically specified in the drawings, the exact timing reference points are given by the parameter description according to the following figures (test voltage levels and float state references shown on previous page):

Figure 10
Timing Reference Points



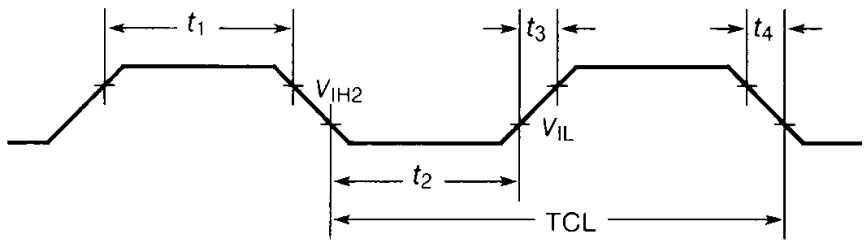
AC Characteristics (cont'd)

External Clock Drive XTAL1

T_A = 0 to + 70 °C; V_{CC} = 5 V ± 10 %; V_{SS} = 0 V

Parameter	Symbol	CPU Clock 20 MHz		Variable Timing 1/TCL = 2 to 40 MHz		Unit
		min.	max.	min.	max.	
Oscillator Period	TCL	25	25	25	500	ns
High Time	t_1	6	–	6	–	ns
Low Time	t_2	6	–	6	–	ns
Rise Time	t_3	–	5	–	5	ns
Fall Time	t_4	–	5	–	5	ns

Figure 10
External Clock Drive XTAL1



AC Characteristics (cont'd)

Multiplexed Bus with Read/Write Delay

T_A = 0 to + 70 °C; V_{CC} = 5 V ± 10 %; V_{SS} = 0 V;
C_L (for Ports 0, 1 and 4, ALE, RD#, WR#, BHE#, CLKOUT) = 100 pF;
ALE cycle time = 6 TCL (150 ns at 20 MHz CPU clock)

Parameter	Symbol	CPU Clock 20 MHz		Variable Timing 1/TCL = 2 to 40 MHz		Unit
		min.	max.	min.	max.	
ALE High Time	t_5	15	–	TCL – 10	–	ns
Address Setup to ALE	t_6	10	–	TCL – 15	–	ns
Address Hold after ALE	t_7	15	–	TCL – 10	–	ns
ALE Falling Edge to RD#, WR#	t_8	15	–	TCL – 10	–	ns
Address Float after RD#, WR#	t_{10}	–	5	–	5	ns
RD#, WR# Low Time	t_{12}	40	–	2TCL – 10	–	ns
RD# to Valid Data In	t_{14}	–	35	–	2TCL – 15	ns
ALE Low to Valid Data In	t_{16}	–	60	–	3TCL – 15	ns
Address to Valid Data In	t_{17}	–	75	–	4TCL – 25	ns
Data Hold after RD# Rising Edge	t_{18}	0	–	0	–	ns
Data Float after RD#	t_{19}	–	35	–	2TCL – 15	ns
Data Valid to WR#	t_{22}	35	–	2TCL – 15	–	ns
Data Hold after WR#	t_{23}	35	–	2TCL – 15	–	ns
ALE rising edge after RD#, WR#	t_{25}	35	–	2TCL – 15	–	ns
Address Hold after RD#, WR#	t_{27}	35	–	2TCL – 15	–	ns

Figure 11
External Memory Read Cycle

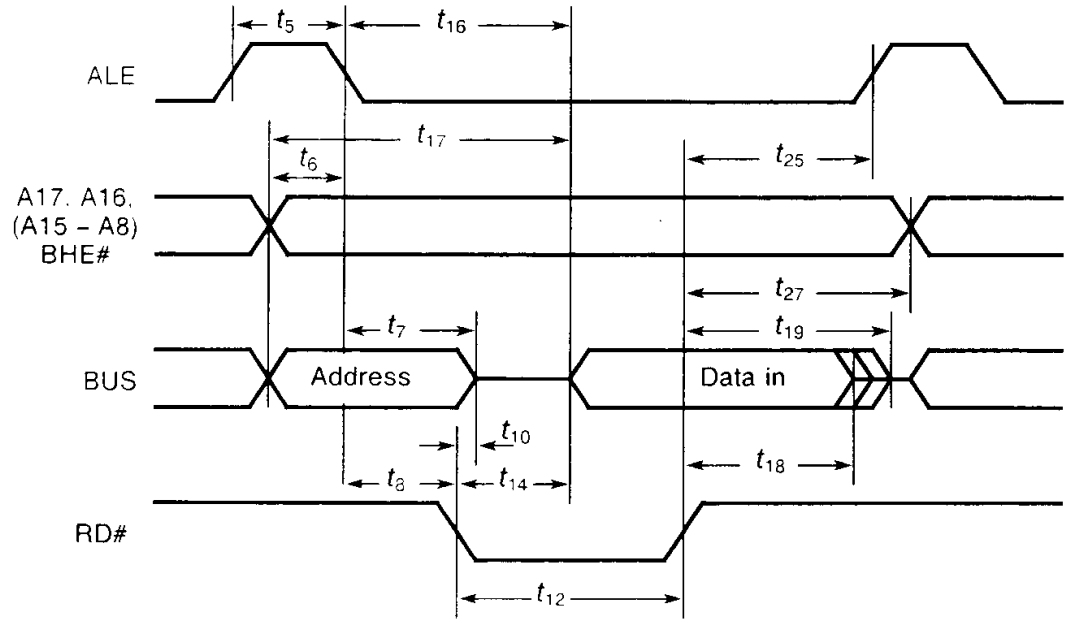
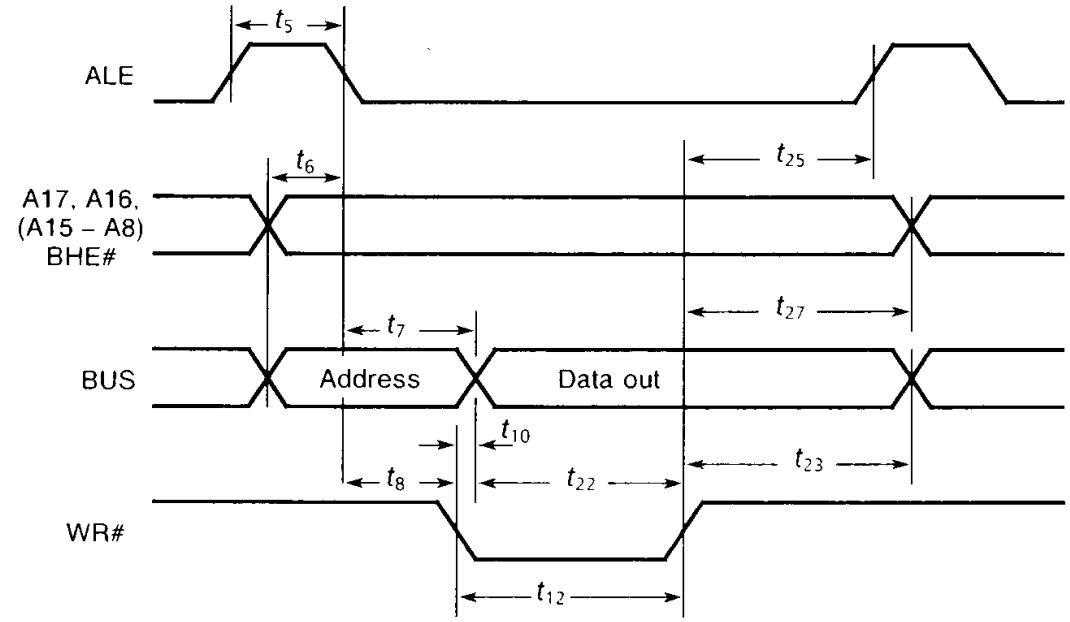


Figure 12
External Memory Write Cycle



AC Characteristics (cont'd)

Multiplexed Bus without Read/Write Delay

T_A = 0 to +70 °C; V_{CC} = 5 V ±10 %; V_{SS} = 0 V;
C_L (for Ports 0, 1 and 4, ALE, RD#, WR#, BHE#, CLKOUT) = 100 pF
ALE cycle time = 6 TCL (150 ns at 20 MHz CPU clock)

Parameter	Symbol	CPU Clock 20 MHz		Variable Timing 1/TCL = 2 to 40 MHz		Unit
		min.	max.	min.	max.	
ALE High Time	t_5	15	–	TCL – 10	–	ns
Address Setup to ALE	t_6	10	–	TCL – 15	–	ns
Address Hold after ALE	t_7	15	–	TCL – 10	–	ns
ALE Falling Edge to RD#, WR#	t_9	– 10	–	– 10	–	ns
Address Float after RD#, WR#	t_{11}	–	30	–	TCL + 5	ns
RD#, WR# Low Time	t_{13}	65	–	3TCL – 10		ns
RD# to Valid Data In	t_{15}	–	60	–	3TCL – 15	ns
ALE Low to Valid Data In	t_{16}	–	60	–	3TCL – 15	ns
Address to Valid Data In	t_{17}	–	75	–	4TCL – 25	ns
Data Hold after RD# Rising Edge	t_{18}	0	–	0	–	ns
Data Float after RD#	t_{19}	–	35	–	2TCL – 15	ns
Data Valid to WR#	t_{22}	35	–	2TCL – 15	–	ns
Data Hold after WR#	t_{23}	35	–	2TCL – 15	–	ns
ALE rising edge after RD#, WR#	t_{25}	35	–	2TCL – 15	–	ns
Address Hold after RD#, WR#	t_{27}	35	–	2TCL – 15	–	ns

Figure 13
External Memory Read Cycle

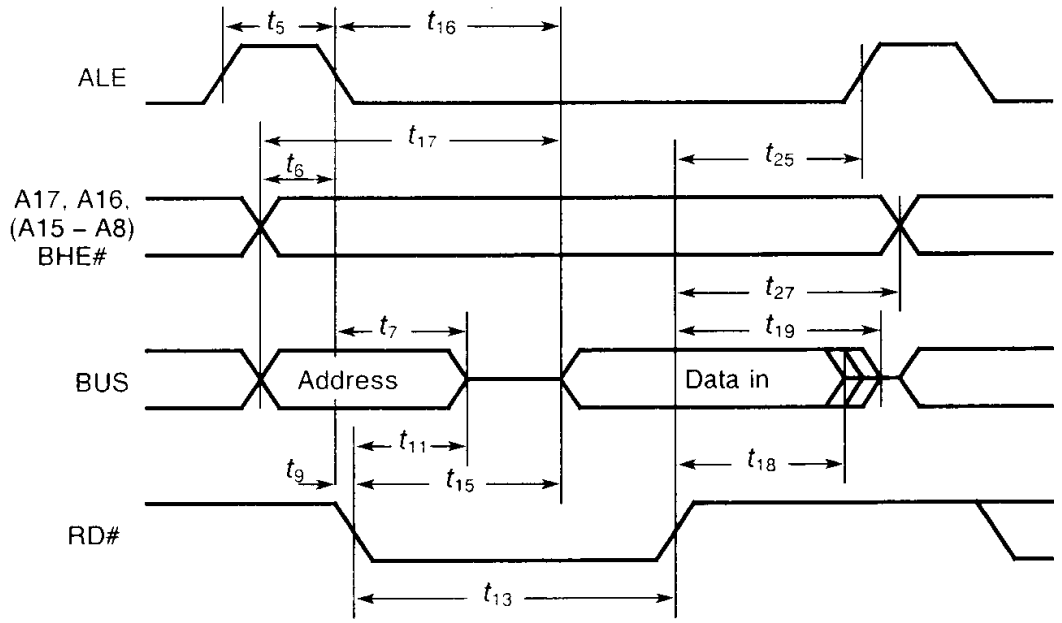
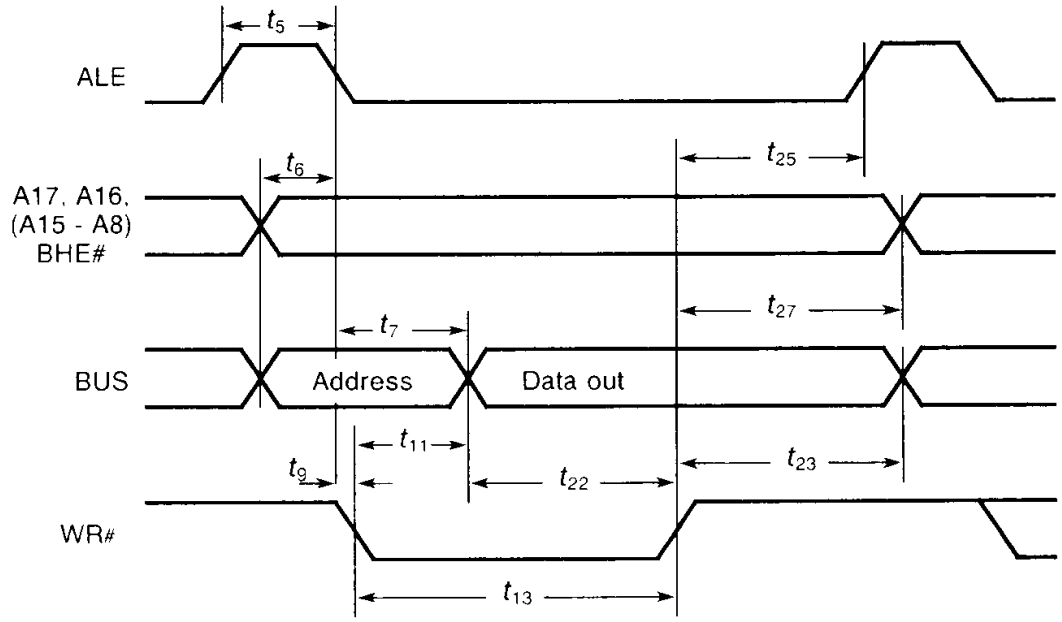


Figure 14
External Memory Write Cycle



AC Characteristics (cont'd)

Non-Multiplexed Bus with Read/Write Delay

T_A = 0 to +70 °C; V_{CC} = 5 V ±10 %; V_{SS} = 0 V;
C_L (for Ports 0, 1 and 4, ALE, RD#, WR#, BHE#, CLKOUT) = 100 pF
ALE cycle time = 4 TCL (150 ns at 20 MHz CPU clock)

Parameter	Symbol	CPU Clock 20 MHz		Variable Timing 1/TCL = 2 to 40 MHz		Unit
		min.	max.	min.	max.	
ALE High Time	t_5	15	–	TCL – 10	–	ns
Address Setup to ALE	t_6	10	–	TCL – 15	–	ns
ALE Falling Edge to RD#, WR#	t_8	15	–	TCL – 10	–	ns
RD#, WR# Low Time	t_{12}	40	–	2TCL – 10	–	ns
RD# to Valid Data In	t_{14}	–	35	–	2TCL – 15	ns
ALE Low to Valid Data In	t_{16}	–	60	–	3TCL – 15	ns
Address to Valid Data In	t_{17}	–	75	–	4TCL – 25	ns
Data Hold after RD# Rising Edge	t_{18}	0	–	0	–	ns
Data Float after RD# *)	t_{20}	–	35	–	2TCL – 15	ns
Data Valid to WR#	t_{22}	35	–	2TCL – 15	–	ns
Data Hold after WR#	t_{24}	15	–	TCL – 10	–	ns
ALE rising edge after RD#, WR#	t_{26}	– 10	–	– 10	–	ns
Address Hold after RD#, WR#	t_{28}	0	–	0	–	ns

*) This time may be longer if no external bus conflict can occur. For example, this requirement is always met if only code but no data are accessed externally.

Figure 15
External Memory Read Cycle

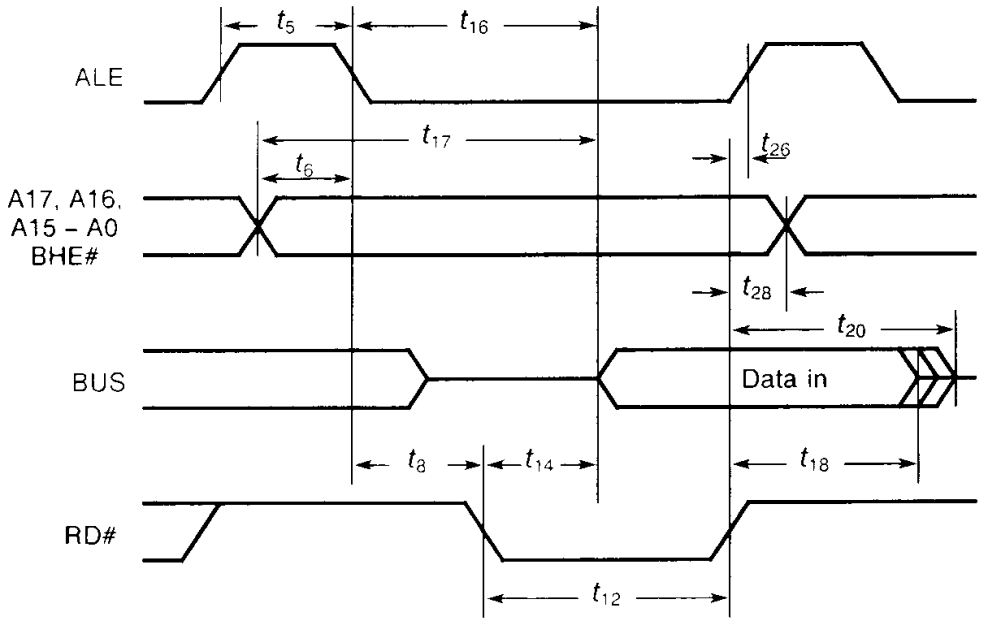
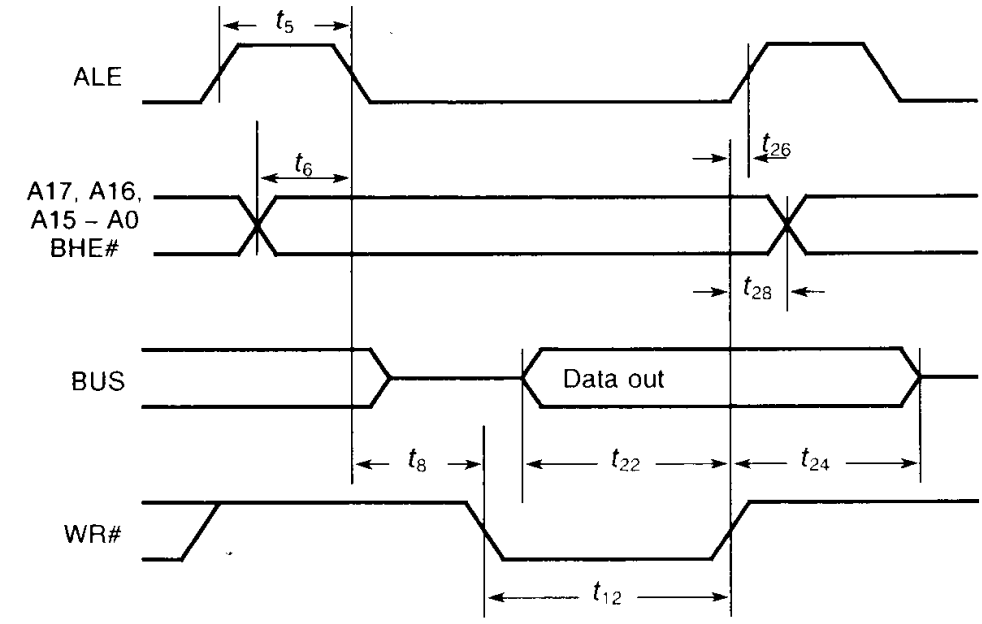


Figure 16
External Memory Write Cycle



AC Characteristics (cont'd)

Non-Multiplexed Bus without Read/Write Delay

T_A = 0 to +70 °C; V_{CC} = 5 V ± 10 %; V_{SS} = 0 V;
C_L (for Ports 0, 1 and 4, ALE, RD#, WR#, BHE#, CLKOUT) = 100 pF
ALE cycle time = 4 TCL (100 ns at 20 MHz CPU clock)

Parameter	Symbol	CPU Clock 20 MHz		Variable Timing 1/TCL = 2 to 40 MHz		Unit
		min.	max.	min.	max.	
ALE High Time	t_5	15	–	TCL – 10	–	ns
Address Setup to ALE	t_6	10	–	TCL – 15	–	ns
ALE Falling Edge to RD#, WR#	t_9	– 10	–	– 10	–	ns
RD#, WR# Low Time	t_{13}	65	–	3TCL – 10	–	ns
RD# to Valid Data In	t_{15}	–	60	–	3TCL – 15	ns
ALE Low to Valid Data In	t_{16}	–	60	–	3TCL – 15	ns
Address to Valid Data In	t_{17}	–	75	–	4TCL – 25	ns
Data Hold after RD# Rising Edge	t_{18}	0	–	0	–	ns
Data Float after RD# *)	t_{21}	–	15	–	TCL – 10	ns
Data Valid to WR#	t_{22}	35	–	2TCL – 15	–	ns
Data Hold after WR#	t_{24}	15	–	TCL – 10	–	ns
ALE rising edge after RD#, WR#	t_{26}	– 10	–	– 10	–	ns
Address Hold after RD#, WR#	t_{28}	0	–	0	–	ns

*) This time may be longer if no external bus conflict can occur. For example, this requirement is always met if only code but no data are accessed externally.

Figure 17
External Memory Read Cycle

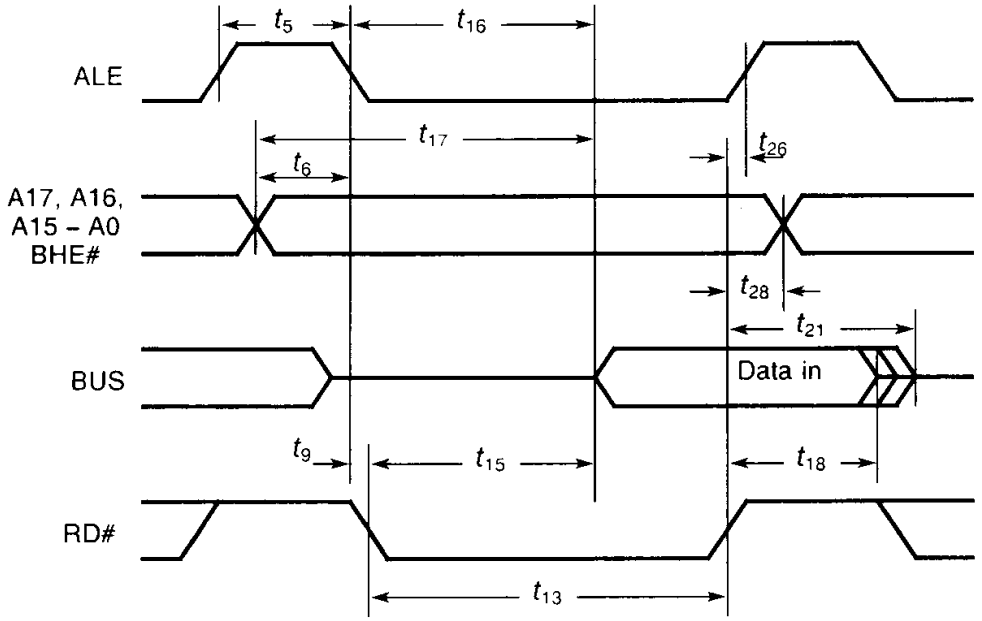
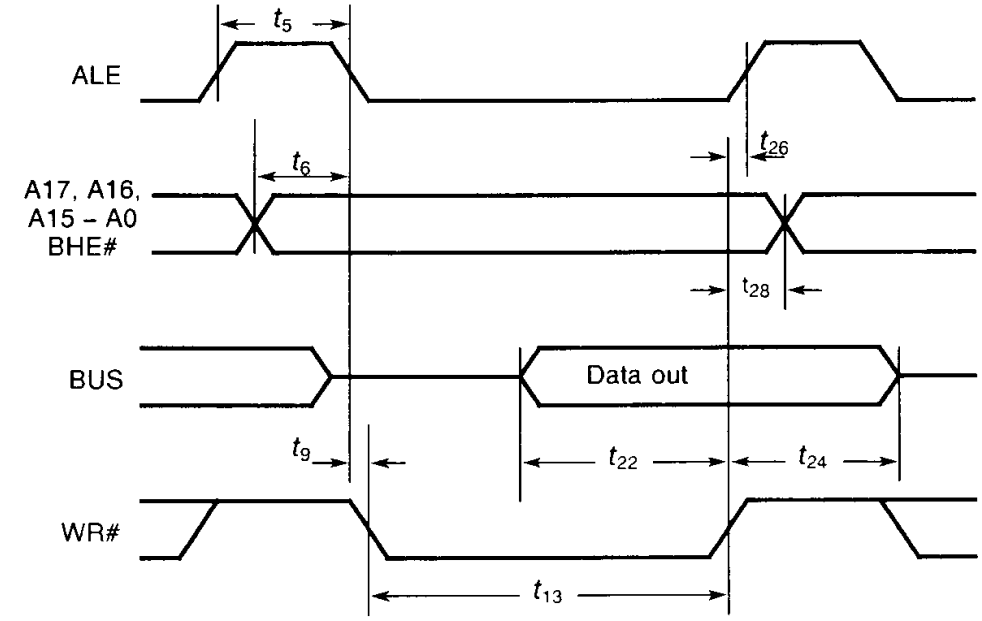


Figure 18
External Memory Write Cycle



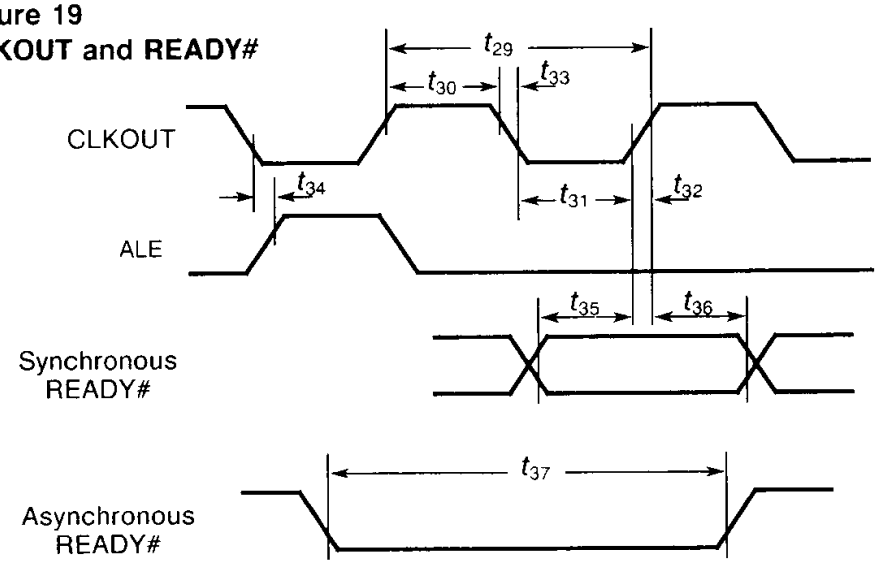
AC Characteristics (cont'd)

CLKOUT and READY#

T_A = 0 to +70 °C; V_{CC} = 5 V ± 10 %; V_{SS} = 0 V;
C_L (for Ports 0, 1 and 4, ALE, RD#, WR#, BHE#, CLKOUT) = 100 pF

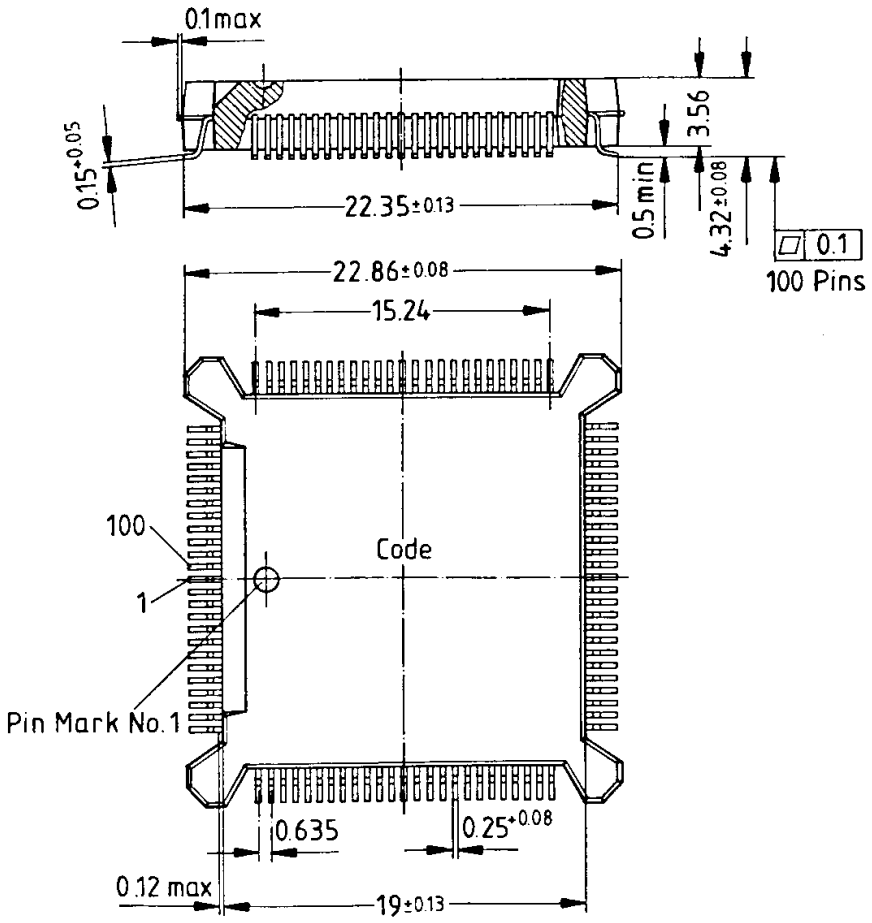
Parameter	Symbol	CPU Clock 20 MHz		Variable Timing 1/TCL = 2 to 40 MHz		Unit
		min.	max.	min.	max.	
CLKOUT Cycle Time	t_{29}	50	50	2TCL	2TCL	ns
CLKOUT High Time	t_{30}	15	–	TCL – 10	–	ns
CLKOUT Low Time	t_{31}	15	–	TCL – 10	–	ns
CLKOUT Rise Time	t_{32}	–	5	–	5	ns
CLKOUT Fall Time	t_{33}	–	5	–	5	ns
ALE Rising to CLKOUT Falling Edge	t_{34}	0	10	0	10	ns
Synchronous READY# Setup Time to CLKOUT	t_{35}	10	–	10	–	ns
Synchronous READY# Hold Time after CLKOUT	t_{36}	10	–	10	–	ns
Asynchronous READY# Hold Time	t_{37}	65	–	2TCL + 15	–	ns

Figure 19
CLKOUT and READY#



Package Outlines

Figure 20
Plastic Package, P-QFP-100 (SMD)
(Plastic Quad-Flat-Pack)



Dimensions in mm