

## MSM80C88A-10RS/GS/JS

### 8-BIT CMOS MICROPROCESSOR

#### GENERAL DESCRIPTION

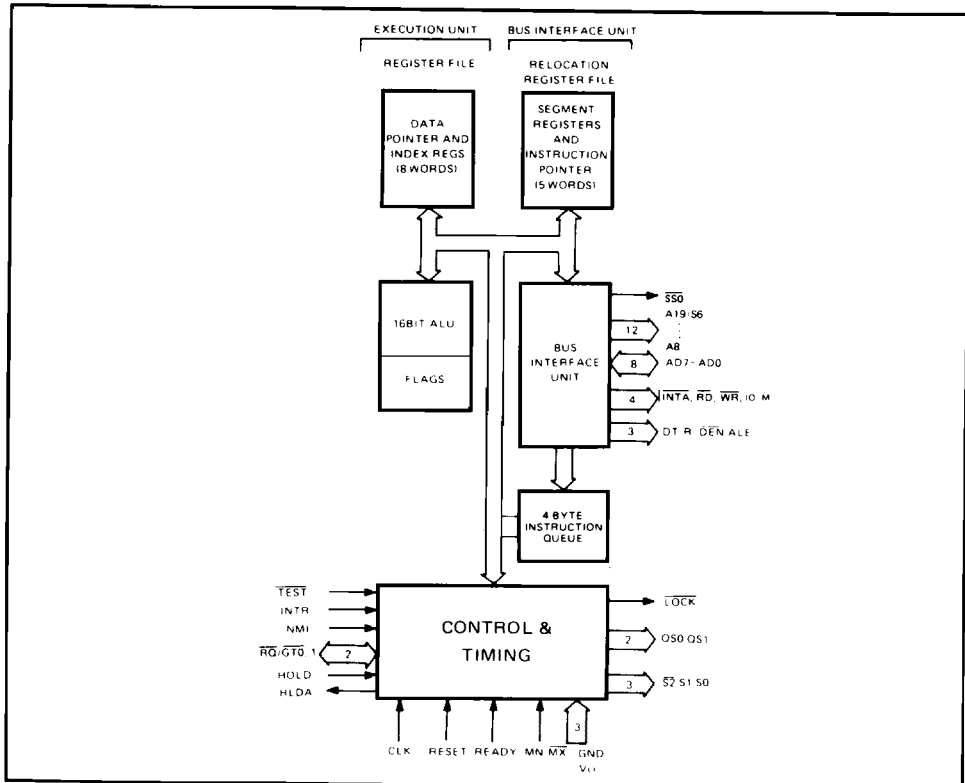
The MSM80C88A-10 are internal 16-bit CPUs with 8-bit interface implemented in Silicon Gate CMOS technology. They are designed with the same processing speed as the NMOS8088-1, but with considerably less power consumption.

The processor has attributes of both 8 and 16-bit microprocessor. It is directly compatible with MSM80C86A-10 software and MSM80C85A/MSM80C85A-2 hardware and peripherals.

#### FEATURES

- 8-Bit Data Bus Interface
- 16-Bit Internal Architecture
- 1 Mbyte Direct Addressable Memory Space
- Software Compatible with MSM80C86A
- Internal 14 Word by 16-bit Register Set
- 24 Operand Addressing Modes
- Bit, Byte, Word and String Operations
- 8 and 16-bit Signed and Unsigned Arithmetic Operation
- From DC to 10 MHz Clock Rate
- Low Power Dissipation (10 mA/MHz)
- Bus Hold Circuitry Eliminates Pull-Up Resistors
- 40 pin Plastic DIP (DIP40-P-600)
- 44 pin PLCC (QFJ44-P-S650)
- 56 pin(L)-V Plastic QFP (QFP56-P-1519-VK)

#### FUNCTIONAL BLOCK DIAGRAM





### ABSOLUTE MAXIMUM RATINGS

Parameter	Symbol	Limits			Unit	Conditions
		MSM80C88A-10RS	MSM80C88A-10GS	MSM80C88A-10JS		
Power Supply Voltage	V <sub>CC</sub>	-0.5 ~ +7			V	With respect to GND
Input Voltage	V <sub>IN</sub>	-0.5 ~ V <sub>CC</sub> +0.5			V	
Output Voltage	V <sub>OUT</sub>	-0.5 ~ V <sub>CC</sub> +0.5			V	
Storage Temperature	T <sub>stg</sub>	-65 ~ +150			°C	—
Power Dissipation	P <sub>D</sub>	1.0	0.7		W	T <sub>a</sub> = 25° C

### OPERATING RANGE

Parameter	Symbol	Limits	Unit
		MSM80C88A-10	
Power Supply Voltage	V <sub>CC</sub>	4.75 ~ 5.25	V
Operating Temperature	T <sub>OP</sub>	0 ~ +70	°C

### RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	MSM80C88A-10			Unit
		MIN	TYP	MAX	
Power Supply Voltage	V <sub>CC</sub>	4.75	5.0	5.25	V
Operating Temperature	T <sub>OP</sub>	0	+25	+70	°C
“L” Input Voltage	V <sub>IL</sub>	-0.5		+0.8	V
“H” Input Voltage	V <sub>IH</sub> (*1) (*2)	V <sub>CC</sub> - 0.8		V <sub>CC</sub> + 0.5	V
		2.0		V <sub>CC</sub> + 0.5	V

\*1 Only CLK, \*2 Except CLK

DC CHARACTERISTICS

(MSM80C88A-10:  $V_{CC} = 4.75$  to  $5.25V$ ,  $T_a = 0^{\circ}C$  to  $+70^{\circ}C$ )

Parameter	Symbol	MIN	TYP	MAX	Unit	Conditions
"L" Output Voltage	$V_{OL}$			0.4	V	$I_{OL} = 2.5$ mA
"H" Output Voltage	$V_{OH}$	3.0			V	$I_{OH} = -2.5$ mA
		$V_{CC}-0.4$				$I_{OH} = -100\mu A$
Input Leak Current	$I_{LI}$	-1.0		+1.0	$\mu A$	$0 < V_{IN} < V_{CC}$
Output Leak Current	$I_{LO}$	-10		+10	$\mu A$	$V_O = V_{CC}$ or GND
Input Leakage Current (Bus Hold Low)	$I_{BHL}$	50		400	$\mu A$	$V_{IN} = 0.8V$ *3
Input Leakage Current (Bus Hold High)	$I_{BHH}$	-50		-400	$\mu A$	$V_{IN} = 3.0V$ *4
Bus Hold Low Overdrive	$I_{BHLO}$			600	$\mu A$	*5
Bus Hold High Overdrive	$I_{BHHO}$			-600	$\mu A$	*6
Operating Power Supply Current	$I_{CC}$			10	mA/MHz	$V_{IL} = GND$ $V_{IH} = V_{CC}$
Standby Supply Current	$I_{CCS}$			500	$\mu A$	$V_{IN} = V_{CC}$ or GND Outputs Unloaded $CLK = GND$ or $V_{CC}$
Input Capacitance	$C_{in}$			10	pF	*7
Output Capacitance	$C_{out}$			15	pF	*7
I/O Capacitance	$C_{I/O}$			20	pF	*7

- \*3. Test conditions is to lower  $V_{IN}$  to GND and then raise  $V_{IN}$  to 0.8V on pins 2–16 and 35–39.
- \*4. Test condition is to raise  $V_{IN}$  to  $V_{CC}$  and then lower  $V_{IN}$  to 3.0V on pins 2–16, 26–32, and 34–39.
- \*5. An external driver must source at least  $I_{BHLO}$  to switch this node from LOW to HIGH.
- \*6. An external driver must sink at least  $I_{BHHO}$  to switch this node from HIGH to LOW.
- \*7. Test Conditions: a) Freq = 1 MHz.  
 b) Unmeasured Pins at GND.  
 c)  $V_{IN}$  at 5.0V or GND.

### A.C. CHARACTERISTICS

(MSM80C88A-10:  $V_{CC} = 4.75V$  to  $5.25V$ ,  $T_a = 0^{\circ}C$  to  $70^{\circ}C$ )

#### Minimum Mode System

#### Timing Requirements

Parameter	Symbol	MSM80C88A-10		Unit
		Min.	Max.	
CLK Cycle Period	TCLCL	100	DC	ns
CLK Low Time	TCLCH	46		ns
CLK High Time	TCHCL	44		ns
CLK Rise Time (From 1.0V to 3.5V)	TCH1CH2		10	ns
CLK Fall Time (From 3.5V to 1.0V)	TCL2CL1		10	ns
Data in Setup Time	TDVCL	20		ns
Data in Hold Time	TCLDX	10		ns
RDY Setup Time into MSM 82C84A (See Notes 1, 2)	TR1VCL	35		ns
RDY Hold Time into MSM 82C84A (See Notes 1, 2)	TCLR1X	0		ns
READY Setup Time into MSM 80C88A	TRYHCH	33		ns
READY Hold Time into MSM 80C88A	TCHRYX	20		ns
READY inactive to CLK (See Note 3)	TRYLCL	-8		ns
HOLD Setup Time	THVCH	20		ns
INTR, NMI, TEST Setup Time (See Note 2)	TINVCH	15		ns
Input Rise Time (Except CLK) (From 0.8V to 2.0V)	TILIH		15	ns
Input Fall Time (Except CLK) (From 2.0V to 0.8V)	TIHIL		15	ns

#### Timing Responses

Parameter	Symbol	MSM80C88A-10		Unit
		Min.	Max.	
Address Valid Delay	TCLAV	10	60	ns
Address Hold Time	TCLAX	10		ns
Address Float Delay	TCLAZ	TCLAX	50	ns
ALE Width	TLHLL	TCLCH-10		ns
ALE Active Delay	TCLLH		40	ns
ALE Inactive Delay	TCHLL		45	ns
Address Hold Time to ALE Inactive	TLLAX	TCHCL-10		ns
Data Valid Delay	TCLDV	10	60	ns
Data Hold Time	TCHDX	10		ns
Data Hold Time after $\overline{WR}$	TWHDX	TCLCH-25		ns
Control Active Delay 1	TCVCTV	10	55	ns
Control Active Delay 2	TCHCTV	10	50	ns
Control Inactive Delay	TCVCTX	10	55	ns
Address Float to $\overline{RD}$ Active	TAZRL	0		ns
$\overline{RD}$ Active Delay	TCLRL	10	70	ns

Parameter	Symbol	MSM80C88A-10		Unit
		Min.	Max.	
$\overline{RD}$ Inactive Delay	TCLR <sub>H</sub>	10	60	ns
$\overline{RD}$ Inactive to Next Address Active	TRH <sub>AV</sub>	TCLCL-35		ns
HLDA Valid Delay	TCLH <sub>AV</sub>	10	60	ns
$\overline{RD}$ Width	TRLR <sub>H</sub>	2TCLCL-40		ns
$\overline{WR}$ Width	TWLW <sub>H</sub>	2TCLCL-35		ns
Address Valid to ALE Low	TAVAL	TCLCH-35		ns
Output Rise Time (From 0.8V to 2.0)	TOLO <sub>H</sub>		15	ns
Output Fall Time (From 2.0V to 0.8V)	TOHO <sub>L</sub>		15	ns

- Notes:**
1. Signals at MSM82C84A shown for reference only.
  2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
  3. Applies only to T2 state. (8 ns into T3)

**Maximum Mode System (Using MSM 82C88 Bus Controller)**

**Timing Requirements**

Parameter	Symbol	MSM80C88A-10		Unit
		Min.	Max.	
CLK Cycle Period	TCLCL	100	DC	ns
CLK Low Time	TCLCH	46		ns
CLK High Time	TCHCL	44		ns
CLK Rise Time (From 1.0V to 3.5V)	TCH1CH2		10	ns
CLK Fall Time (From 3.5V to 1.0V)	TCL2CL1		10	ns
Data in Setup Time	TDVCL	20		ns
Data in Hold Time	TCLDX	10		ns
RDY Setup Time into MSM 82C84A (See Notes 1, 2)	TR1VCL	35		ns
RDY Hold Time into MSM 82C84A (See Notes 1, 2)	TCLR1X	0		ns
READY Setup Time into MSM 80C88A	TRYHCH	33		ns
READY Hold Time into MSM 80C88A	TCHRYX	20		ns
READY inactive to CLK (See Note 3)	TRYLCL	-8		ns
Set up Time for Recognition (NMI, INTR, TEST) (See Note 2)	TINVCH	15		ns
$\overline{RQ}/\overline{GT}$ Setup Time	TGVCH	15		ns
$\overline{RQ}$ Hold Time into MSM 80C88A	TCHGX	20		ns
Input Rise Time (Except CLK) (From 0.8V to 2.0V)	TILIH		15	ns
Input Fall Time (Except CLK) (From 2.0V to 0.8V)	TIHIL		15	ns

**Timing Responses**

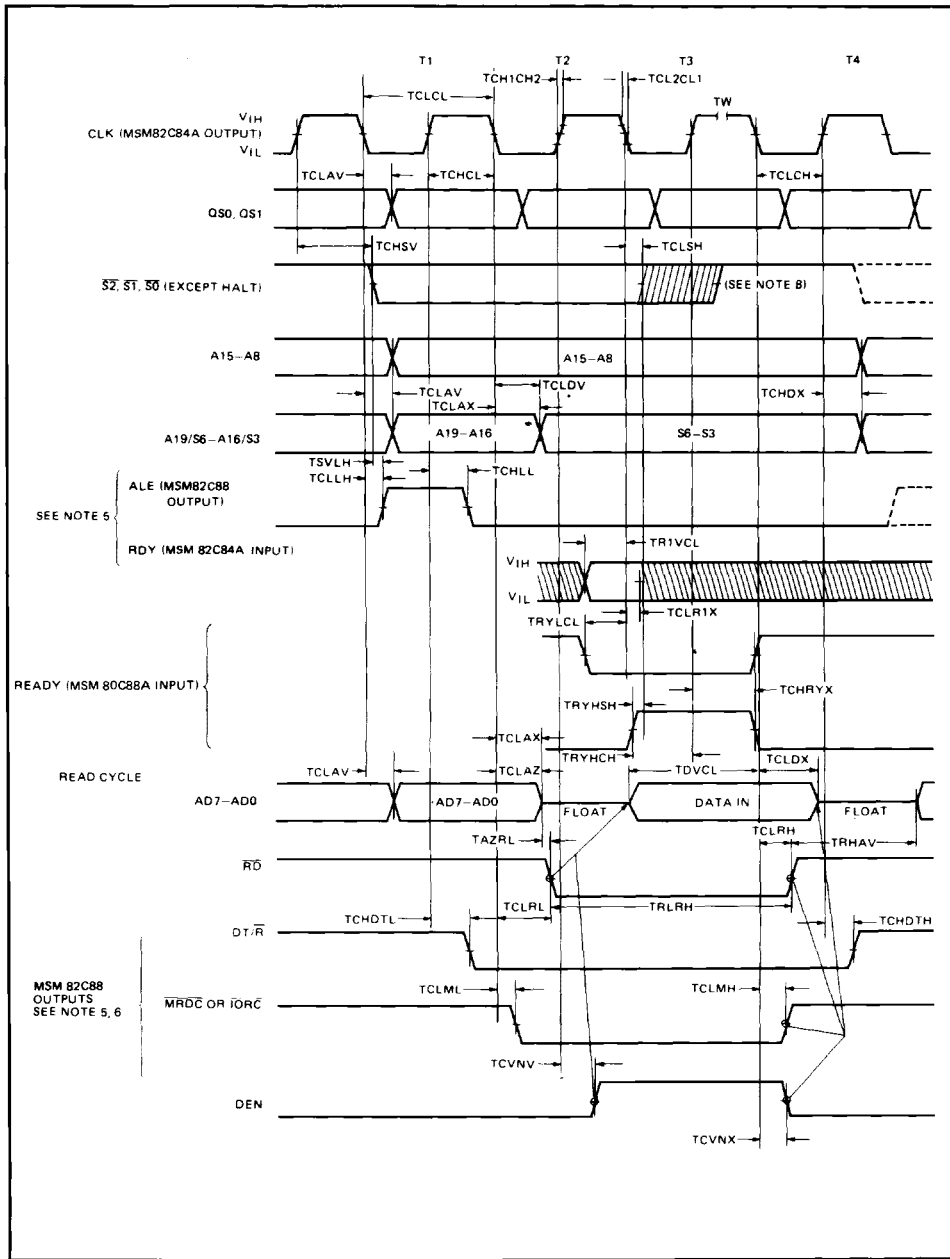
Parameter	Symbol	MSM80C88A-10		Unit
		Min.	Max.	
Command Active Delay (See Note 1)	TCLML	5	35	ns
Command Inactive Delay (See Note 1)	TCLMH	5	45	ns
READY Active to Status Passive (See Note 4)	TRYHSH		45	ns
Status Active Delay	TCHSV	10	45	ns
Status Inactive Delay	TCLSH	10	60	ns
Address Valid Delay	TCLAV	10	60	ns
Address Hold Time	TCLAX	10		ns
Address Float Delay	TCLAZ	TCLAX	50	ns
Status Valid to ALE High (See Note 1)	TSVLH		25	ns
Status Valid to MCE High (See Note 1)	TSVMCH		30	ns
CLK low to ALE Valid (See Note 1)	TCLLH		25	ns
CLK Low to MCE High (See Note 1)	TCLMCH		25	ns
ALE Inactive Delay (See Note 1)	TCHLL	4	25	ns
Data Valid Delay	TCLDV	10	60	ns
Data Hold Time	TCHDX	10		ns
Control Active Delay (See Note 1)	TCVNV	5	45	ns
Control Inactive Delay (See Note 1)	TCVNX	5	45	ns
Address Float to $\overline{RD}$ Active	TAZRL	0		ns
$\overline{RD}$ Active Delay	TCLRRL	10	70	ns
$\overline{RD}$ Inactive Delay	TCLRHL	10	60	ns
$\overline{RD}$ Inactive to Next Address Active	TRHAV	TCLCL-35		ns
Direction Control Active Delay (See Note 1)	TCHDTL		50	ns
Direction Control Inactive Delay (See Note 1)	TCHDTH		30	ns
$\overline{GT}$ Active Delay (See Note 5)	TCLGL	0	38	ns
$\overline{GT}$ Inactive Delay	TCLGH	0	45	ns
$\overline{RD}$ Width	TRLRH	2TCLCL-40		ns
Output Rise Time (From 0.8V to 2.0V)	TOLOH		15	ns
Output Fall Time (From 2.0V to 0.8V)	TOHOL		15	ns

- Notes:**
1. Signals at MSM 82C84A or MSM 82C88 are shown for reference only.
  2. Setup requirement for asynchronous signal only to guarantee recognition at next CLK.
  3. Applies only to T2 state (8 ns into T3)
  4. Applies only to T3 and wait states.
  5.  $C_L = 40\text{pF}$  ( $\overline{RQ}/\overline{GT}_0, \overline{RQ}/\overline{GT}_1$ )



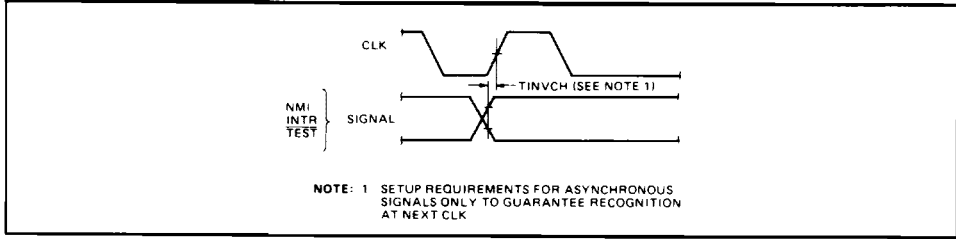


Maximum Mode

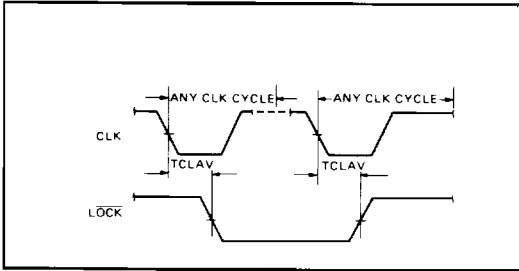




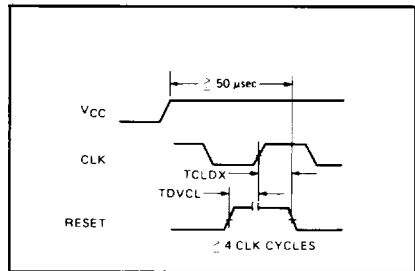
**Asynchronous Signal Recognition**



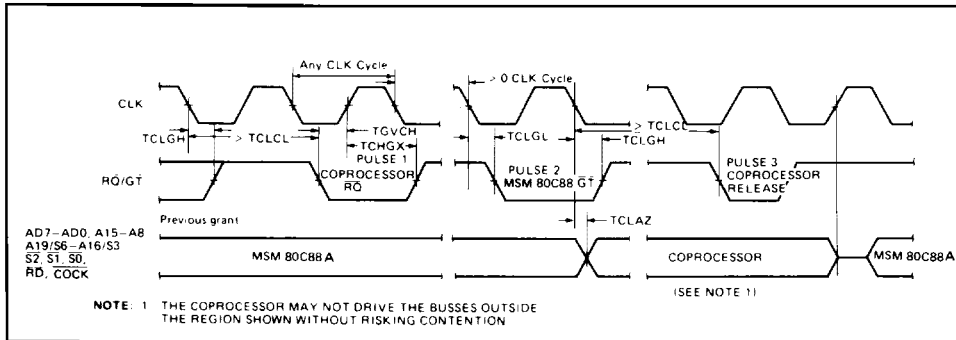
**Bus Lock Signal Timing (Maximum Mode Only)**



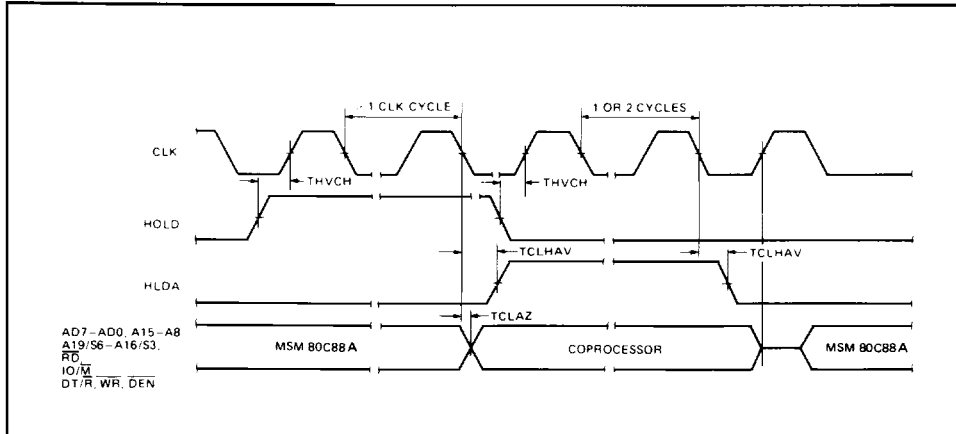
**Reset Timing**



**Request/Grant Sequence Timing (Maximum Mode Only)**



**Hold/Hold Acknowledge Timing (Minimum Mode Only)**



## PIN DESCRIPTION

### AD0-AD7

ADDRESS DATA BUS: Input/Output

These lines are the multiplexed address and data bus.

These are the address bus at T1 cycle and the data bus at T2, T3, TW and T4 cycle.

T2, T3, TW and T4 cycle.

These lines are high impedance during interrupt acknowledge and hold acknowledge.

### A8-A15

ADDRESS BUS: Output

These lines are the address bus bits 8 thru 15 at all cycles.

These lines do not have to be latched by an ALE signal.

These lines are high impedance during interrupt acknowledge and hold acknowledge.

### A16/S3, A17/S4, A18/S5, A19/S6

ADDRESS/STATUS: Output

These are the four most significant address as at the T1, cycle.

Accessing I/O port address, these are low at T1 Cycles.

These lines are Status lines at the T2, T3, TW and T4 Cycle.

S5 indicate interrupt enable Flag.

S3 and S4 are encoded as shown.

S3	S4	Characteristics
0	0	Alternate Data
1	0	Stack
0	1	Code or None
1	1	Data

These lines are high impedance during hold acknowledge.

### $\overline{RD}$

READ: Output

This line indicates that the CPU is in a memory or I/O read cycle.

This line is the read strobe signal when the CPU reads data from a memory or I/O device.

This line is active low.

This line is high impedance during hold acknowledge.

### READY

READY: Input

This line indicates to the CPU that an addressed memory or I/O device is ready to read or write.

This line is active high.

IF the setup and hold time are outof specification, an illegal operation will occur.

### INTR

INTERRUPT REQUEST: Input

This line is a level triggered interrupt request signal which is sampled during the last clock cycle of instruction and string manipulations.

It can be internally masked by software.

This signal is active high and internally synchronized.

### $\overline{TEST}$

TEST: Input

This line is examined by a "WAIT" instruction.

When  $\overline{TEST}$  is high, the CPU enters an idle cycle.

When  $\overline{TEST}$  is low, the CPU exits an idle cycle.

### NMI

NON MASKABLE INTERRUPT: Input

This line causes a type 2 interrupt.

NMI is not maskable.

This signal is internally synchronized and needs a 2 clock cycle pulse width.

### RESET

RESET: Input

This signal causes the CPU to initialize immediately.

This signal is active high and must be at least four clock cycles.

### CLK

CLOCK: Input

This signal provide the basic timing for an internal circuit.

### $\overline{MN/MX}$

MINIMUM/MAXIMUM: Input

This signal selects the CPU's operate mode.

When  $V_{CC}$  is connected, the CPU operates in minimum mode.

When GND is connected, the CPU operates in maximum mode.

### $V_{CC}$

$V_{CC}$  +5V supplied.

### GND

GROUND

The following pin function descriptions are for maximum mode only.

Other pin functions are already described.

**S0, S1, S2**

STATUS: Output

These lines indicate bus status and they are used by the MSM82C88 Bus Controller to generate all memory and I/O access control signals.

These lines are high impedance during hold acknowledge.

These status lines are encoded as shown.

S2	S1	S0	Characteristics
0 (LOW)	0	0	Interrupt acknowledge
0	0	1	Read I/O Port
0	1	0	Write I/O Port
0	1	1	Halt
1 (HIGH)	0	0	Code Access
1	0	1	Read Memory
1	1	0	Write Memory
1	1	1	Passive

**RQ/GT0**

**RQ/GT1**

REQUEST/GRANT: Input/Output

These lines are used for Bus Request from other devices and Bus GRANT to other devices.

These lines are bidirectional and active low.

**LOCK**

LOCK: Output

This line is active low.

When this line is low, other devices can not gain control of the bus.

This line is high impedance during hold acknowledge.

**QS0/QS1**

QUEUE STATUS: Output

These are Queue Status Lines that indicate internal instruction queue status.

QS1	QS0	Characteristics
0 (LOW)	0	No Operation
0	1	First Byte of Op Code from Queue
1 (HIGH)	0	Empty the Queue
1	1	Subsequent Byte from Queue

The following pin function descriptions are minimum mode only. Other pin functions are already described.

**IO/M**

STATUS: Output

This line selects memory address space or I/O address space.

When this line is low, the CPU selects memory address space and when it is high, the CPU selects I/O address space.

This line is high impedance during hold acknowledge.

**WR**

WRITE: Output

This line indicates that the CPU is in a memory or I/O write cycle.

This line is a write strobe signal when the CPU writes data to memory or an I/O device.

This line is active low.

This line is high impedance during hold acknowledge.

**INTA**

INTERRUPT ACKNOWLEDGE: Output

This line is a read strobe signal for the interrupt acknowledge cycle.

This line is active low.

**ALE**

ADDRESS LATCH ENABLE: Output

This line is used for latching an address into the MSM82C12 address latch it is a positive pulse and the trailing edge is used to strobe the address. This line in never floated.

**DT/R**

DATA TRANSMIT/RECEIVE: Output

This line is used to control the direction of the bus transceiver.

When this line is high, the CPU transmits data, and when it is low, the CPU receive data.

This line is high impedance during hold acknowledge.

**DEN**

DATA ENABLE: Output

This line is used to control the output enable of the bus transceiver.

This line is active low. This line is high impedance during hold acknowledge.

**HOLD**

HOLD REQUEST: Input

This line is used for a Bus Request from an other device.

This line is active high.

**HLDA**

HOLD ACKNOWLEDGE: Output

This line is used for a Bus Grant to an other device.

This line is active high.

**SS0**

STATUS: Output

This line is logically equivalent to S0 in the maximum mode.

## STATIC OPERATION

All MSM80C88A circuitry is of static design. Internal registers, counters and latches are static and require no refresh as with dynamic circuit design. This eliminates the minimum operating frequency restriction placed on other microprocessors. The MSM80C88A can operate from DC to the appropriate upper frequency limit. The processor clock may be stopped in either state (high/low) and held there indefinitely. This type of operation is especially useful for system debug or power critical applications.

The MSM80C88A can be signal stepped using only the CPU clock. This state can be maintained as long as is necessary. Single step clock operation allows simple interface circuitry to provide critical information for bringing up your system.

Static design also allows very low frequency operation (down to DC). In a power critical situation, this can provide extremely low power operation since 80C88A power dissipation is directly related to operating frequency. As the system frequency is reduced, so is the operating power until, ultimately, at a DC input frequency, the MSM80C88A power requirement is the standby current (500  $\mu$ A maximum).

## FUNCTIONAL DESCRIPTION

### GENERAL OPERATION

The internal function of the MSM80C88A consist of a Bus Interface Unit (BIU) and an Execution Unit (EU). These units operate mutually but perform as separate processors.

The BIU performs instruction fetch and queuing, operand fetch, DATA read and write address relocation and basic bus control. By performing instruction pre-fetch while waiting for decoding and execution of instruction, the CPU's performance is increased. Up to 4-bytes fo instruction stream can be queued.

EU receives pre-fetched instructions from the BIU queue, decodes and executes instructions and provides an un-relocated operand address to the BIU.

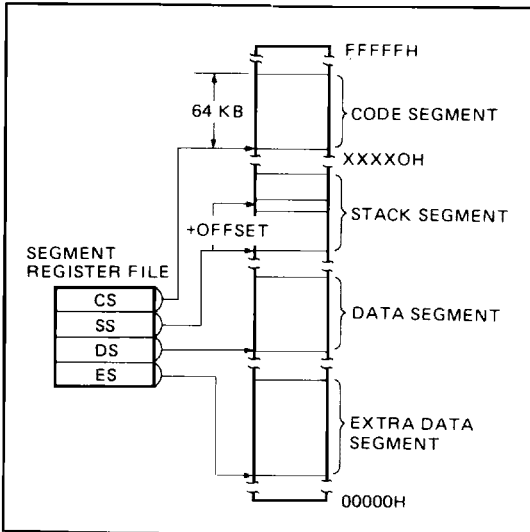
### MEMORY ORGANIZATION

The MSM80C88A has a 20-bit address to memory. Each address has 8-bit data width. Memory is organized 00000H to FFFFFH and is logically divided into four segments: code, data, extra data and stack segment. Each segment contains up to 64 Kbytes and locates on a 16-byte boundary. (Fig. 3a)

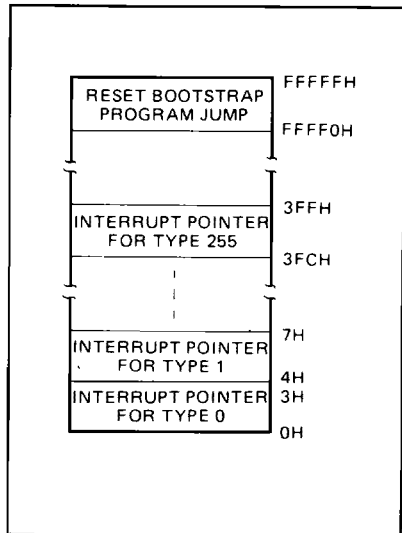
All memory references are made relative to a segment register according to a select rule. Memory location FFFF0H is the start address after reset, and 00000H through 003FFH are reserved as an interrupt pointer. There are 256 types of interrupt pointer;

Each interrupt type has a 4-byte pointer element consisting of a 16-bit segment address and a 16-bit offset address.

Memory Organization



Reserved Memory Locations



Memory Reference Need	Segment Register Used	Segment Selection Rule
Instructions	CODE (CS)	Automatic with all instruction prefetch.
Stack	STACK (SS)	All stack pushes and pops. Memory references relative to BP base register except data references.
Local Data	DATA (DS)	Data references when relative to stack, destination of string operation, or explicitly overridden.
External (Global) Data	EXTRA (ES)	Destination of string operations: Explicitly selected using a segment override.

**MINIMUM AND MAXIMUM MODES**

The MSM80C88A has two system modes: minimum and maximum. When using the maximum mode, it is easy to organize a multiple-CPU system with the MSM 82C88 Bus Controller which generates the bus control signal.

When using the minimum mode, it is easy to organize a simple system by generating the bus control signal itself. MN/MX is the mode select pin. Definition of 24–31, 34 pin changes depends on the MN/MX pin.

**BUS OPERATION**

The MSM80C88A has a time multiplexed address and data bus. If a non-multiplexed bus is desired for the system, it is only needed to add the address latch.

A CPU bus cycle consists of at least four clock cycles: T1, T2, T3 and T4. (Fig. 4)

The address output occurs during T1, and data transfer occurs during T3 and T4. T2 is used for changing the direction of the bus during read operation. When the device which is accessed by the CPU is not ready to data transfer and send to the CPU "NOT READY" is indicated TW cycles are inserted between T3 and T4.

When a bus cycle is not needed, T1 cycles are inserted between the bus cycles for internal execution. At the T1 cycle an ALE signal is output from the CPU or the MSM82C88 depending in MN/MX. at the trailing edge of an ALE, a valid address may be latched. Status bits S0, S1 and S2 are uased, in maximum mode, by the bus controller to recognize the type of bus operation according to the following table.

S2	S1	S0	Characteristics
0 (LOW)	0	0	Interrupt acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bit S3 through S6 are multiplexed with A16–A19, and therefore they are valid during T2 through T4. S3 and S4 indicate which segment register was selected on the bus cycle, according to the following table.

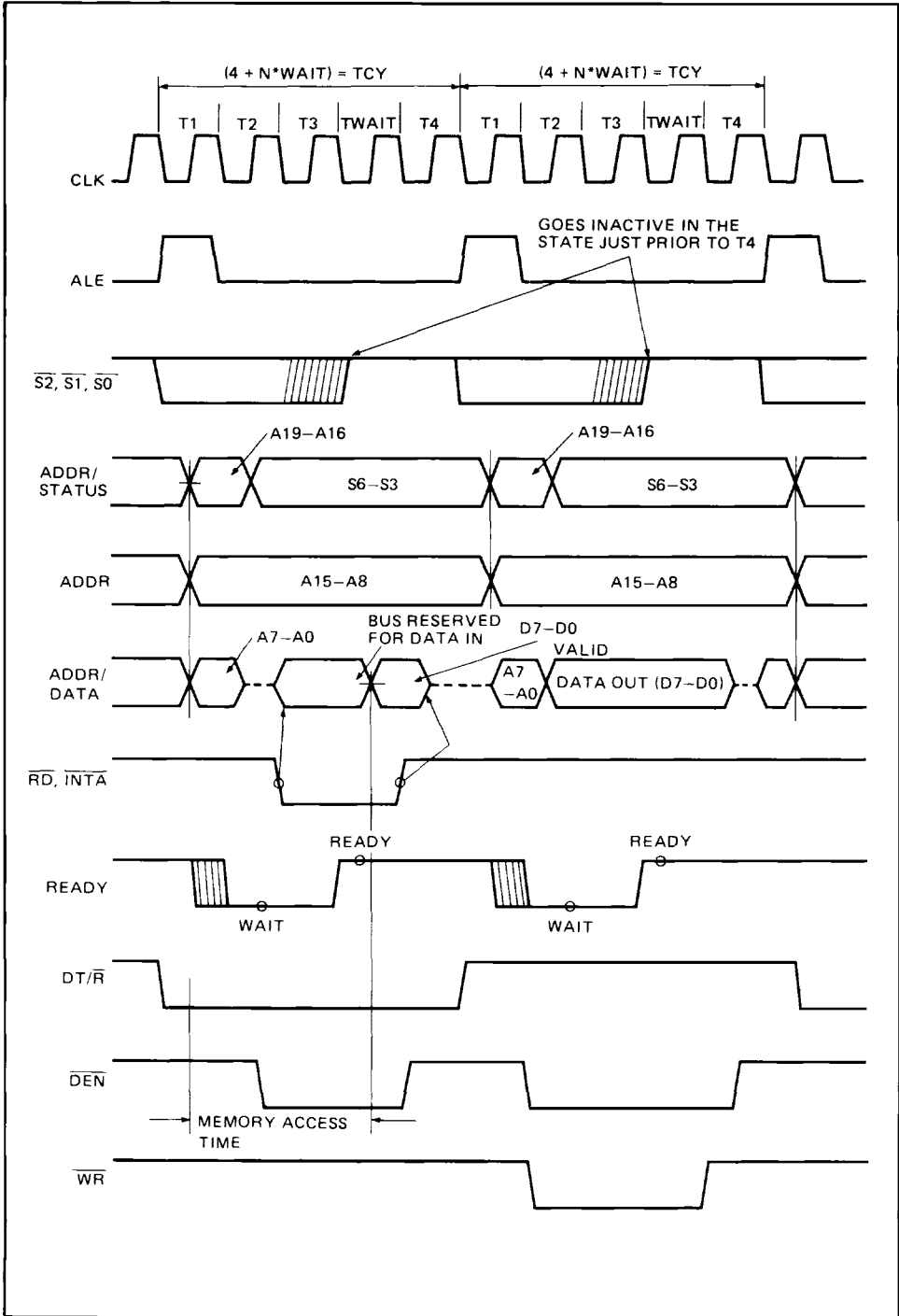
S4	S3	Characteristics
0 (LOW)	0	Alternate Data (Extra Segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data

S5 indicates interrupt enable Flag.

**I/O ADDRESSING**

The MSM80C88A has a 64 Kbyte I/O. When the CPU accesses an I/O device, address A0–A15 are in same format as a memory access, and A16–A19 are low. I/O ports address are same as four memory.

Basic System Timing



## EXTERNAL INTERFACE

### RESET

CPU initialization is executed by the RESET pin. The MSM80C88A's RESET High signal is required for greater than 4 clock cycles.

The rising edge of RESET terminates the present operation immediately. The falling edge of RESET triggers an internal reset sequence for approximately 10 clock cycles. After internal reset sequence is finished, normal operation begins from absolute location FFFF0H.

### INTERRUPT OPERATIONS

The interrupt operation is classified as software or hardware, and hardware interrupt is classified as non-maskable or maskable.

An interrupt causes a new program location which is defined by the interrupt pointer table, according to the interrupt type. Absolute location 00000H through 003FFH is reserved for the interrupt pointer table. The interrupt pointer table consists of 256-elements. Each element is 4 bytes in size and corresponds to an 8-bit type number which is sent from an interrupt request device during the interrupt acknowledge cycle.

### NON-MASKABLE INTERRUPT (NMI)

The MSM80C88A has a non-maskable Interrupt (NM1) which is of higher priority than a maskable interrupt request (INTR).

An NM1 request pulse width needs minimum of 2 clock cycles. The NM1 will be serviced at the end of the current instruction or between string manipulations.

### MASKABLE INTERRUPT (INTR)

The MSM80C88A provides another interrupt request (INTR) which can be masked by software. INTR is level triggered, so it must be held until interrupt request is acknowledged.

The INTR will be serviced at the end of the current instruction or between string manipulations.

### INTERRUPT ACKNOWLEDGE

During the interrupt acknowledge sequence, further interrupts are disabled. The interrupt enable bit is reset by any interrupt, after which the Flag register is automatically pushed onto the stack. During an acknowledge sequence, the CPU emits the lock signal from T2 of first bus cycle to T2 of second bus cycle. At the second bus cycle, a byte is fetched from the external device as a vector which identifies the type of interrupt. This vector is multiplied by four and used as an interrupt pointer address (INTR only).

The Interrupt Return (IRET) instruction includes a Flag pop operation which returns the original interrupt enable bit when it restores the Flag.

### HALT

When a Halt instruction is executed, the CPU enter Halt state. An interrupt request or RESET will force the MSM80C88A out of the Halt state.

### SYSTEM TIMING—MINIMUM MODE

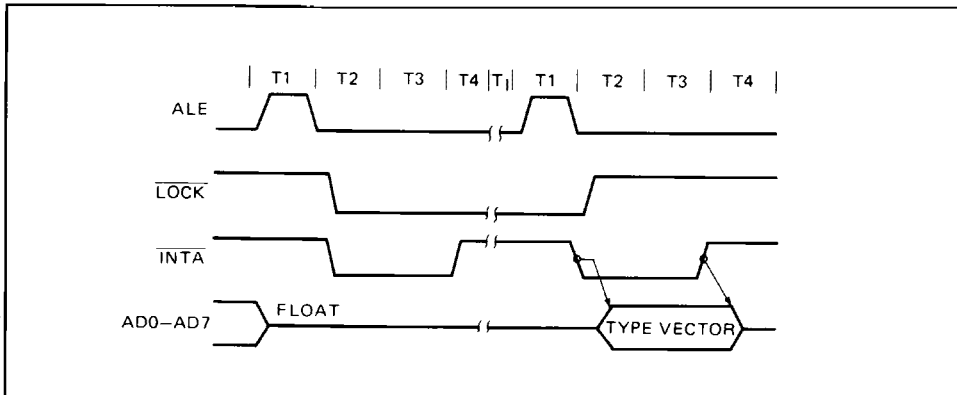
A bus cycle begins at T1 with an ALE signal. The trailing edge of ALE signal is used to latch the address. From T1 to T4 the IO/M signal indicates a memory or I/O operation. From T2 to T4, the address data bus changes the address bus to the data bus.

The read (RD), write (WR), and interrupt acknowledge (INTA) signals caused the addressed device to enable the data bus. These signals become active at the beginning of T2 and inactive at the beginning of T4.

### SYSTEM TIMING—MAXIMUM MODE

In maximum mode, the MSM82C88 Bus Controller is added to system. The CPU sends status information to the Bus Controller. Bus timing signals are generated by the Bus Controller. Bus timing is almost the same as in minimum mode.

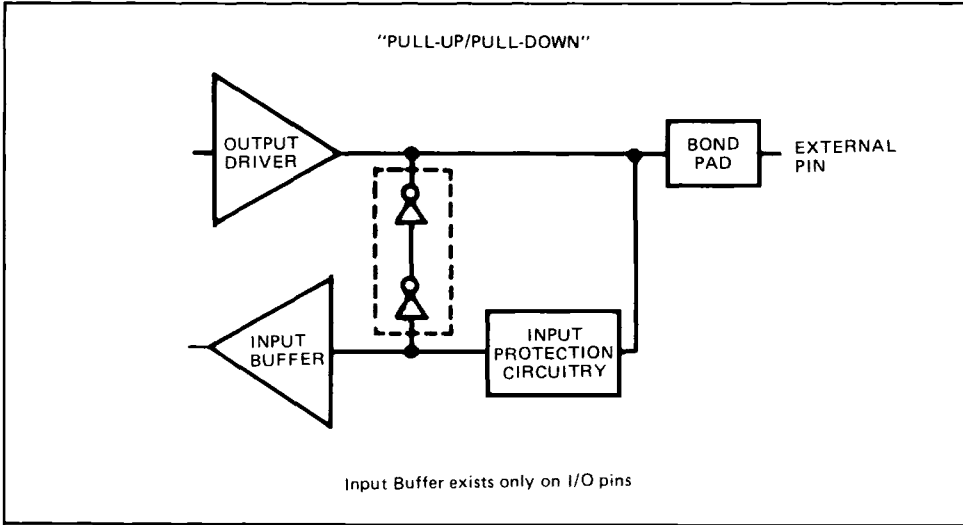
### Interrupt Acknowledge Sequence



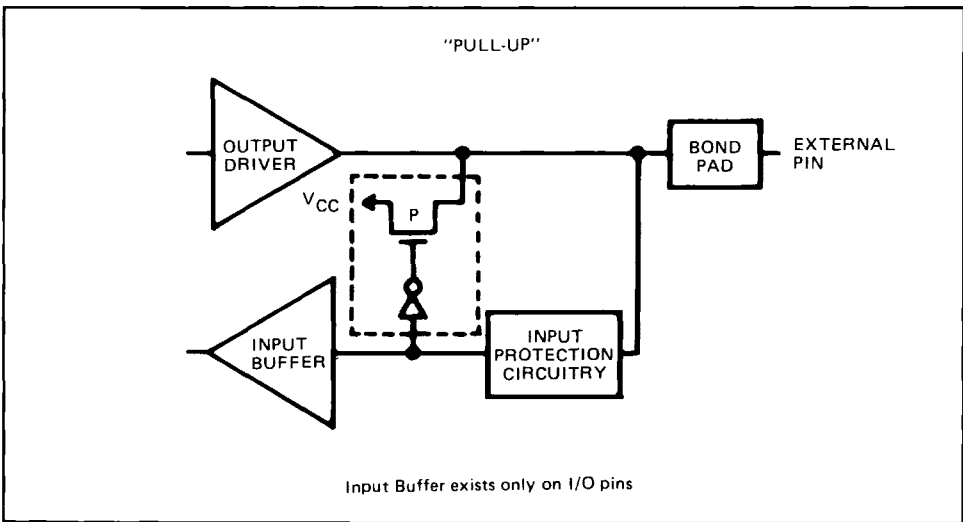
**BUS HOLD CIRCUITRY**

To avoid high current conditions caused by floating inputs to CMOS devices, and to eliminate the need for pull-up/down resistors, "bus-hold" circuitry has been used on 80C86 pins 2-16, 26-32, and 34-39 (Figures 6a, 6b). These circuits will maintain the last valid logic state if no driving source is present (i.e. an unconnected pin or a driving source which goes to a high impedance state). To overdrive the "bus

hold" circuits, an external driver must be capable of supplying approximately 400  $\mu$ A minimum sink or source current at valid input voltage levels. Since this "bus hold" circuitry is active and not a "resistive" type element, the associated power supply current is negligible and power dissipation is significantly reduced when compared to the use of passive pull-up resistors.



**Figure 6a. Bus hold circuitry pin 2-16, 35-39.**



**Figure 6b. Bus hold circuitry pin 26-32, 34**



**ARITHMETIC**

<p><b>ADD = Add:</b> Reg./memory with register to either register/memory Immediate to accumulator</p>	<p>0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0</p>	<p>mod mod 0 0 0 mod 0 0 0</p>	<p>r/m reg data</p>	<p>data if w = 1 data if s:w = 01</p>
<p><b>ADC = Add with carry:</b> Reg./memory with register to either register/memory Immediate to accumulator</p>	<p>0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0</p>	<p>mod mod 0 1 0 mod 0 1 0</p>	<p>r/m reg data</p>	<p>data if w = 1 data if s:w = 01</p>
<p><b>INC = Increment:</b> Register/memory Register AAA = ASCII adjust for add DAA = Decimal adjust for add</p>	<p>1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 0 1 0 0 1 1 1</p>	<p>mod 0 0 0 0 0 mod 0 0 0 0 0</p>	<p>r/m reg data</p>	<p>data if s:w = 01</p>
<p><b>SUB = subtract:</b> Reg./memory and register to either register/memory Immediate from accumulator</p>	<p>0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0</p>	<p>mod 1 0 1 mod 1 0 1</p>	<p>r/m reg data</p>	<p>data if s:w = 01</p>
<p><b>SBB = Subtract with borrow:</b> Reg./memory and register to either register/memory Immediate from accumulator</p>	<p>0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0</p>	<p>mod 0 1 1 mod 0 1 1</p>	<p>r/m reg data</p>	<p>data if s:w = 01</p>
<p><b>DEC = Decrement:</b> Register/memory Register NEG = Change sign</p>	<p>1 1 1 1 1 1 1 1 0 1 0 0 1 0 0 1 1 1 1 1 0 1 1 0</p>	<p>mod 0 0 1 mod 0 1 1</p>	<p>r/m reg data</p>	<p>data if s:w = 01</p>
<p><b>CMP = Compare:</b> Register/memory and register Immediate with register/memory Immediate with accumulator AAS = ASCII adjust for subtract</p>	<p>0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 1 1</p>	<p>mod 1 1 1 mod 1 1 1</p>	<p>r/m reg data</p>	<p>data if s:w = 01</p>



**LOGIC**

NOT = Invert	1 1 1 0 0 1 1	w	mod	0 1 0	r/m	
SHL/SAL = Shift logical/arithmetic left	1 1 0 1 0 0 v	w	mod	1 0 0	r/m	
SHR = Shift logical right	1 1 0 1 0 0 v	w	mod	1 0 1	r/m	
SAR = Shift arithmetic right	1 1 0 1 0 0 v	w	mod	1 1 1	r/m	
ROL = Rotate left	1 1 0 1 0 0 v	w	mod	0 0 0	r/m	
ROR = Rotate right	1 1 0 1 0 0 v	w	mod	0 0 1	r/m	
RCL = Rotate left through carry	1 1 0 1 0 0 v	w	mod	0 1 0	r/m	
RCR = Rotate right through carry	1 1 0 1 0 0 v	w	mod	0 1 1	r/m	
AND = And:						
Reg./memory and register to either	0 0 1 0 0 0 d	w	mod	reg	r/m	data if w = 1
Immediate to register/memory	1 0 0 0 0 0 w	w	mod	1 0 0	r/m	data if w = 1
Immediate to accumulator	0 0 1 0 0 1 0 w	w		data		
TEST = And function to flags, no result:						
Register/memory and register	1 0 0 0 0 1 0 w	w	mod	reg	r/m	data if w = 1
Immediate data and register/memory	1 1 1 0 1 1 w	w	mod	0 0 0	r/m	data if w = 1
Immediate data and accumulator	1 0 1 0 1 0 0 w	w		data		
OR = Or:						
Reg./memory and register to either	0 0 0 0 1 0 d	w	mod	reg	r/m	data if w = 1
Immediate to register/memory	1 0 0 0 0 0 w	w	mod	0 1	r/m	data if w = 1
Immediate to accumulator	0 0 0 0 1 1 0 w	w		data		
XOR = Exclusive or:						
Reg./memory and register to either	0 0 1 1 0 0 d	w	mod	reg	r/m	data if w = 1
Immediate to register/memory	1 0 0 0 0 0 w	w	mod	1 1 0	r/m	data if w = 1
Immediate to accumulator	0 0 1 1 0 1 0 w	w		data		

**STRING MANIPULATION**

REP = Repeat	1 1 1 1 0 0 1 z					
MOVS = Move byte/word	1 0 1 0 0 1 0 w					
CMPS = Compare byte/word	1 0 1 0 0 1 1 w					
SCAS = Scan byte/word	1 0 1 0 1 1 w					
LODS = Load byte/word to AL/AX	1 0 1 0 1 1 0 w					
STOS = Store byte/word from AL/AX	1 0 1 0 1 0 1 w					



**CONTROL TRANSFER**

CALL = Call:	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Direct within segment	1 1 1 0 1 0 0 0	disp-low		disp-high
Indirect within segment	1 1 1 1 1 1 1 1	mod 0 1 0	r/m	
Direct intersegment	1 0 0 1 1 0 1 0	offset-low		offset-high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 0 1 1	r/m	seg-high
JMP = Unconditional Jump:				
Direct within segment	1 1 1 0 1 0 0 1	disp-low		disp-high
Direct within segment-short	1 1 1 0 1 0 1 1	disp		
Indirect within segment	1 1 1 1 1 1 1 1	mod 1 0 0	r/m	
Direct intersegment	1 1 1 0 1 0 1 0	offset-low		offset-high
Indirect intersegment	1 1 1 1 1 1 1 1	mod 1 0 1	r/m	seg-high
RET = Return from CALL:				
Within segment	1 1 0 0 0 0 1 1			
Within seg. adding immediate to SP	1 1 0 0 0 0 1 0	data-low		data-high
Intersegment	1 1 0 0 1 0 1 1			
Intersegment adding immediate to SP	1 1 0 0 1 0 1 0	data-low		data-high

**Footnotes:**

AL = 8-bit accumulator  
 AX = 18-bit accumulator  
 CX = Count register  
 DS = Data segment  
 ES = Extra segment

Above/below refers to unsigned value

Greater = more positive

Less = less positive (more negative) signed value

If d = 1 then "to" reg: If d = 0 then "from" reg.

If w = 1 then word instruction: If w = 0 then byte instruction

If mod = 11 then r/m is treated as a REG field

If mod = 00 then DISP = 0\*, disp-low and disp-high are absent

If mod = 01 then DISP = disp-low sign-extended to 16-bits, disp-high is absent

If mod = 10 then DISP = disp-high: disp-low

If r/m = 000 then EA = (BX) + (SI) + DISP

If r/m = 001 then EA = (BX) + (DI) + DISP

If r/m = 010 then EA = (BP) + (SI) + DISP

If r/m = 011 then EA = (BP) + (DI) + DISP

If r/m = 100 then EA = (SI) + DISP

If r/m = 101 then EA = (DI) + DISP

If r/m = 110 then EA = (BP) + DISP\*

If r/m = 111 then EA = (BX) + DISP

DISP follows 2nd byte of instruction (before data if required)

\* except if mod = 00 and r/m = 110 then EA-disp-high: disp-low

If s:w = 01 then 16 bits of immediate data form the operand

If s:w = 11 then an immediate data byte is sign extended to form the 16-bit operand

If v = 0 then "count" = 1: If v = 1 then "count" in (CL)

x = don't care

z is used for string primitives for comparison with ZF FLAG

**SEGMENT OVERRIDE PREFIX**

001 reg 110

REG is assigned according to the following table:

16-Bit (w = 1)	8-Bit (w = 0)	Segment
000 AX	000 AL	00 ES
001 CX	001 CL	01 CS
010 DX	010 DL	10 SS
011 BX	011 BL	11 DS
100 SP	100 AH	
101 BP	101 CH	
110 SI	110 DH	
111 DI	111 BH	

Instructions which reference the flag register file as a 16-bit object use the symbol FLAGS to represent the file:

FLAGS = x:x:x:(OF):(DF):(IF):(TF):(SF):(ZF):X:(AF):X:(PF):X:(CF)