



# UM82C59A-2

## CHMOS Programmable Interrupt Controller

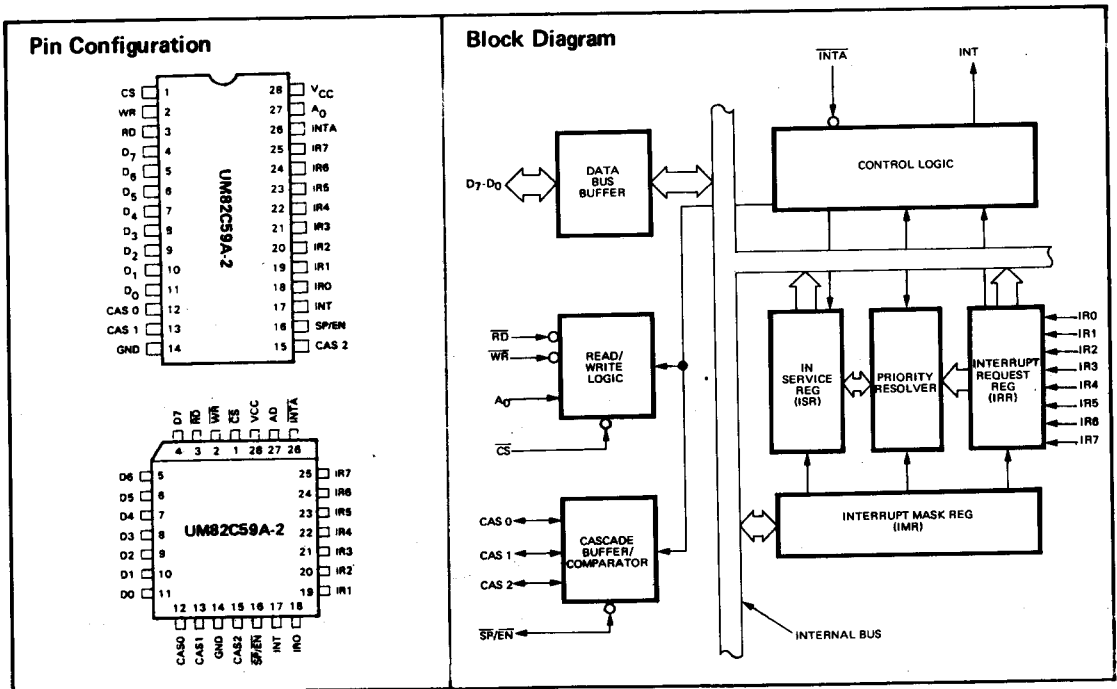
### Features

- Pin compatible with NMOS 8259A-2
- Eight-level priority controller
- Expandable to 64 levels
- Programmable interrupt modes
- Low standby power – 10  $\mu$ A
- Individual request mask capability
- 80C86/88 and 8080/85/86/88 compatible
- Fully static design
- Single 5V power supply
- Will be available in 28-lead plastic DIP and 28-lead PLCC packages

### General Description

The UM82C59A-2 is a high performance CHMOS Version of the NMOS UM82C59A-2 Priority Interrupt Controller. The UM82C59A-2 is designed to relieve the system CPU from the task of polling in a multi-level priority interrupt system. The high speed and industry standard configuration of the UM82C59A-2, make it compatible with micro-processors such as the 80C86/88, 8086/88 and 8080/85. The UM82C59A-2 can handle up to 8 vectored priority

interrupts for the CPU and is cascadable to 64 without additional circuitry. It is designed to minimize the software and real time overhead in handling multi-level priority interrupts. Two modes of operation make the UM82C59A-2 optimal for a variety of system requirements. Static CHMOS circuit design, requiring no clock input, insures low operating power. It is packaged in a 28-pin plastic DIP.



**Absolute Maximum Ratings\***

Ambient Temperature Under Bias . . . . .	0°C to 70°C
Storage Temperature . . . . .	-65°C to + 150°C
Supply Voltage (w.r.t. ground) . . . . .	-0.5 to 7.0V
Input Voltage (w.r.t. ground) . . . . .	-0.5 to $V_{CC} + 0.5V$
Output Voltage (w.r.t. ground) . . . . .	-0.5 to $V_{CC} + 0.5V$
Power Dissipation . . . . .	0.9 Watt

**\*Comments**

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.

Notice: Specifications contained within the following tables are subject to change.

**D.C. Characteristics** ( $T_A = 0^\circ C$  to  $70^\circ C$ ,  $V_{CC} = 5V \pm 10\%$ )

Symbol	Parameter	Min	Max	Units	Test Conditions
$I_{CCS}$	Standby Supply Current		10	$\mu A$	$V_{IN} = V_{CC}$ or GND All IR = $V_{CC}$ Outputs Unloaded $V_{CC} = 5.5V$
$I_{CC}$	Operating Supply Current		5	mA	(Note)
$V_{IH}$	Input High Voltage	2.2	$V_{CC} + 0.5$	V	
$V_{IL}$	Input Low Voltage	-0.5	0.8	V	
$V_{OL}$	Output Low Voltage		0.4	V	$I_{OL} = 2.5$ mA
$V_{OH}$	Output High Voltage	3.0 $V_{CC} - 0.4$		V	$I_{OH} = -2.5$ mA $I_{OH} = -100$ $\mu A$
$I_{LI}$	Input Leakage Current		$\pm 1.0$	$\mu A$	$0V \leq V_{IN} \leq V_{CC}$
$I_{LO}$	Output Leakage Current		$\pm 10$	$\mu A$	$0V \leq V_{OUT} \leq V_{CC}$
$I_{LIR}$	IR Input Leakage Current		-300 + 10	$\mu A$	$V_{IN} = 0$ $V_{IN} = V_{CC}$

Note: Repeated data input with 80C86-2 timings.

**Capacitance** ( $T_A = 25^\circ C$ ;  $V_{CC} = GND = 0V$ )

Symbol	Parameter	Min	Max	Units	Test Conditions
$C_{IN}$	Input Capacitance		7	pF	$f_c = 1$ MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins at GND
$C_{OUT}$	Output Capacitance		15	pF	

**TIMING REQUIREMENTS (Continue)**

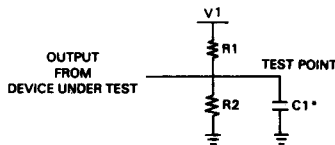
Symbol	Parameter	Min.	Max.	Units	Test Conditions
$T_{AHLW}$	Ao/ $\overline{CS}$ Setup to $\overline{WR}$	0		ns	
$T_{WHAX}$	Ao/ $\overline{CS}$ Hold after $\overline{WR}$	0		ns	
$T_{WLWH}$	$\overline{WR}$ Pulse Width	190		ns	
$T_{DVWH}$	Data Setup to $\overline{WR}$	160		ns	
$T_{WHDX}$	Data Hold after $\overline{WR}$	0		ns	
$T_{JLJH}$	Interrupt Request Width (Low)	100		ns	See Note 1
$T_{CVIAL}$	Cascade Setup to Second or Third INTA (Slave Only)	40		ns	
$T_{RHRL}$	End of $\overline{RD}$ to next $\overline{RD}$ ; End of INTA to next INTA within an INTA sequence only	160		ns	
$T_{WHWL}$	End of $\overline{WR}$ to next $\overline{WR}$	190		ns	
* $T_{CHCL}$	End of Command to next Command (Not same command type) End of INTA sequence to next INTA sequence	400		ns	

\*Worst case timing for  $T_{CHCL}$  in an actual microprocessor system is typically much greater than 400 ns (i.e. 8085A = 1.6 $\mu$ s, 8085A-2 = 1 $\mu$ s, 80C86 = 1 $\mu$ s).

Note: This is the low time required to clear the input latch in the edge triggered mode.

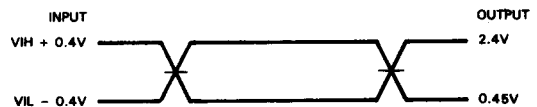
**TIMING RESPONSES**

Symbol	Parameter	Min.	Max.	Units	Test Conditions
$T_{RLDV}$	Data Valid from $\overline{RD}/\overline{INTA}$		120	ns	1
$T_{RHDZ}$	Data Float after $\overline{RD}/\overline{INTA}$	10	85	ns	2
$T_{JHIH}$	Interrupt Output Delay		300	ns	1
$T_{IALCV}$	Cascade Valid from First INTA (Master Only)		360	ns	1
$T_{RLEL}$	Enable Active from $\overline{RD}$ or $\overline{INTA}$		100	ns	1
$T_{RHEH}$	Enable Inactive from $\overline{RD}$ or $\overline{INTA}$		150	ns	1
$T_{AHDV}$	Data Valid from Stable Address		200	ns	1
$T_{CVDV}$	Cascade Valid to Valid Data		200	ns	1

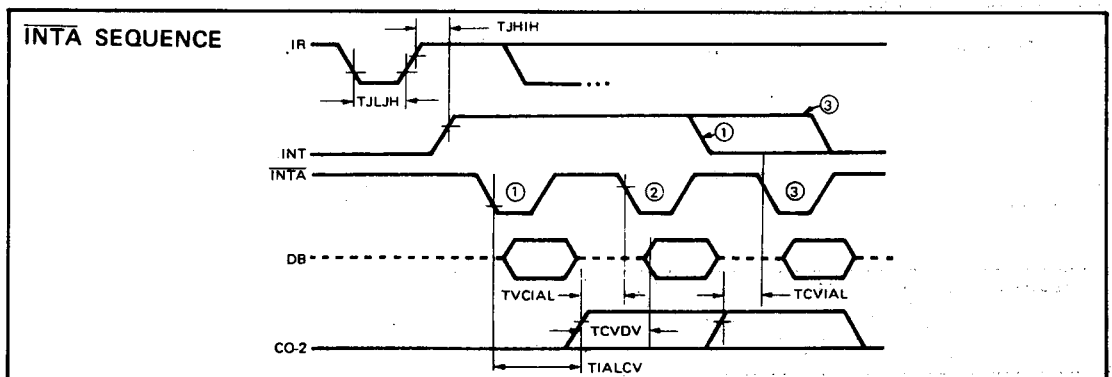
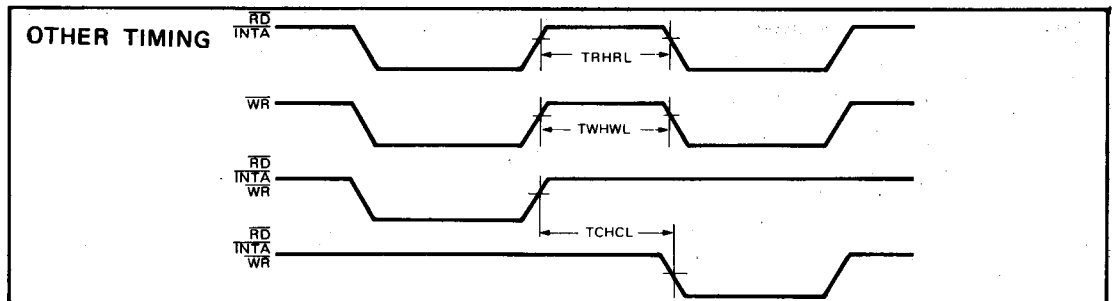
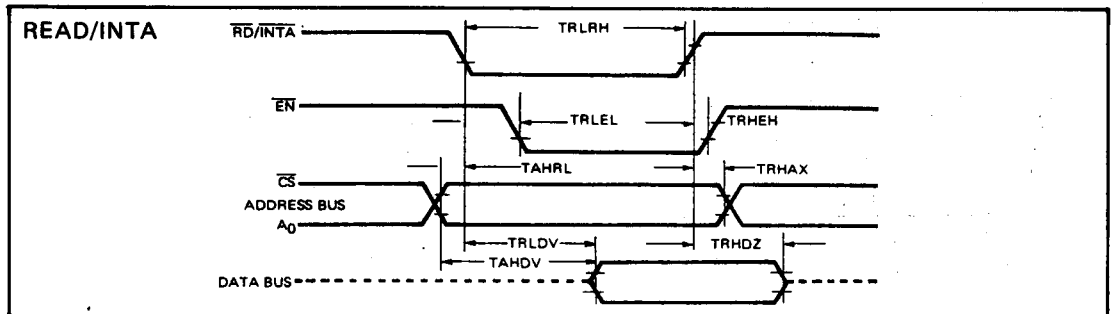
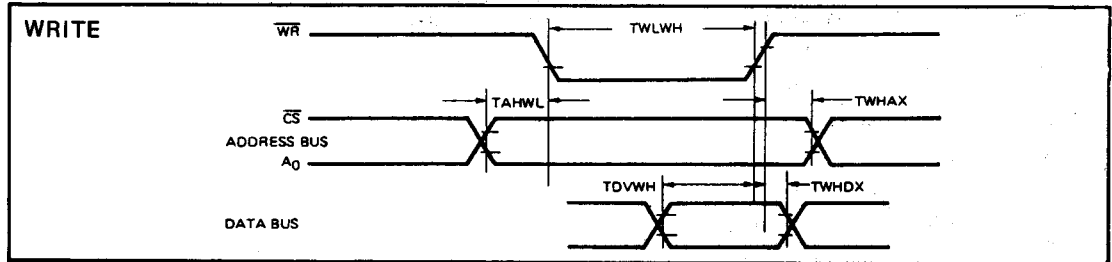
**A.C. Test Circuits**


\*Includes Stray and Jig Capacitance

Test Condition	V1	R1	R2	C2
1	1.7V	523 $\Omega$	Open	100 pF
2	4.5V	1.8K $\Omega$	1.8K $\Omega$	30 pF

**Test Condition Definition Table**
**A.C. Testing Input, Output Waveform**


A.C. Testing: All input signals must switch between  $V_{IL} - 0.4V$  and  $V_{IH} + 0.4V$ .  $T_R$  and  $T_F$  must be less than or equal to 15 ns.

**Waveforms**


**NOTES:**  
 Interrupt output must remain HIGH at least until leading edge of first INTA.  
 1. Cycle 1 in 80C86 and 80C88 systems, the Data Bus is not active.

**Pin Description**

Symbol	Pin No.	Type	Name and Function
$V_{CC}$	28	I	SUPPLY: +5V Supply.
GND	14	I	GROUND.
$\overline{CS}$	1	I	CHIP SELECT: A low on this pin enables $\overline{RD}$ and $\overline{WR}$ communication between the CPU and the UM82C59A-2. INTA functions are independent of CS.
$\overline{WR}$	2	I	WRITE: A low on this pin when $\overline{CS}$ is low enables the UM82C59A-2 to accept command words from the CPU.
$\overline{RD}$	3	I	READ: A low on this pin when $\overline{CS}$ is low enables the UM82C59A-2 to release status onto the data bus for the CPU.
$D_7-D_0$	4-11	I/O	BIDIRECTIONAL DATA BUS: Control, status and interrupt-vector information is transferred via this bus.
CAS0-CAS2	12, 13, 15	I/O	CASCADE LINES: The CAS lines form a private UM82C59A-2 bus to control a multiple UM82C59A-2 structure. These pins are outputs for a master UM82C59A-2 and inputs for a slave UM82C59A-2.
$\overline{SP/EN}$	16	I/O	SLAVE PROGRAM/ENABLE BUFFER: This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	INTERRUPT: This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR0-IR7	18-25	I	INTERRUPT REQUESTS: Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode). Internal pull-up resistors are implemented on IR0-7.
$\overline{INTA}$	26	I	INTERRUPT ACKNOWLEDGE: This pin is used to enable UM82C59A-2 interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
$A_0$	27	I	$A_0$ ADDRESS LINE: This pin acts in conjunction with the $\overline{CS}$ , $\overline{WR}$ , and $\overline{RD}$ pins. It is used by the 82C59A-2 to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU $A_0$ address line ( $A_1$ for 80C86, 80C88).

**Functional Description**
**Interrupts in Microcomputer Systems**

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the Polled approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuous polling cycle and that such a method would have a serious, detrimental effect on system through put, thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desirable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete, however, the processor would resume exactly where it left off.

This method is called interrupt. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. This "pointer" is an address in a vectoring table and will often be referred to, in this document, as vectoring data.

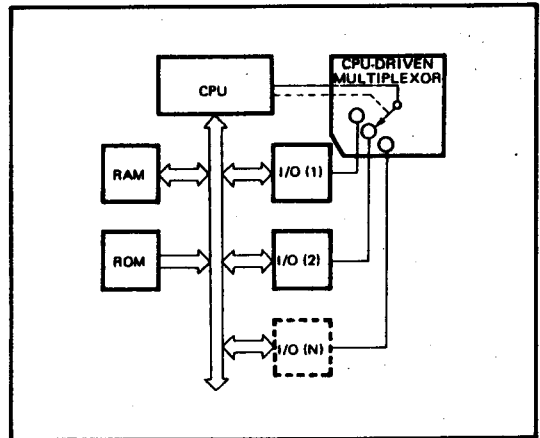
### The UM82C59A-2

The UM82C59A-2 is a device specifically designed for use in real time, interrupt driven microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other UM82C59A-2's (up to 64 levels). It is programmed by system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the manner in which the requests are processed by the UM82C59A-2 can be configured to match system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

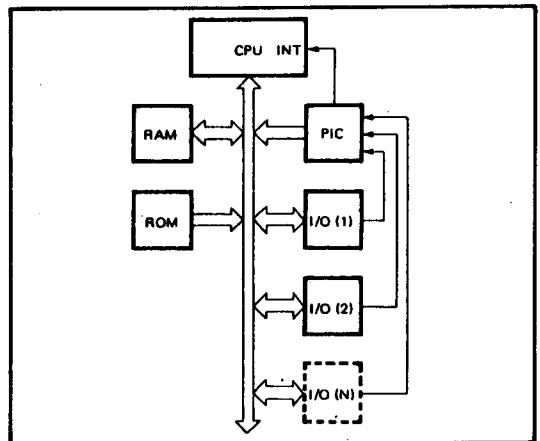
### Interrupt Request Register (IRR) and In-Service Register (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR)

and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service, and the ISR is used to store all the interrupt levels which are currently being serviced.



**Figure a. Polled Method**



**Figure b. Interrupt Method**

### Priority Resolver

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during INTA sequence.

### Interrupt Mask Register (IMR)

The IMR stores the bits which mask the interrupt lines to be masked. The IMR operates on the IRR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

**INT (Interrupt)**

This output goes directly to the CPU interrupt input. The  $V_{OH}$  level on this line is designed to be fully compatible with the 8080A, 8085A, 80C88 and 80C86 input levels.

**INTA (Interrupt Acknowledge)**

INTA pulses will cause the 82C59A-2 to release vectoring information onto the data bus. The format of this data depends on the system mode ( $\mu$ PM) of the UM82C59A-2.

**Data Bus Buffer**

This 3-state, bidirectional 8-bit buffer is used to interface the UM82C59A-2 to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

**Read/Write Control Logic**

The function of this block is to accept Output commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the UM82C59A-2 to be transferred onto the Data Bus.

 **$\overline{CS}$  (Chip Select)**

A LOW on this input enables the UM82C59A-2. No

reading or writing of the device will occur unless the device is selected.

**WR (Write)**

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the UM82C59A-2.

 **$\overline{RD}$  (Read)**

A LOW on this input enables the UM82C59A-2 to send the status of the Interrupt Request Register (IRR). In Service Register (ISR), the Interrupt Mask Register (IMR), or the interrupt level (in the poll mode) onto the Data Bus.

 **$A_0$** 

This input signal is used in conjunction with  $\overline{WR}$  and  $\overline{RD}$  signals to write commands into the various command registers, as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

**The Cascade Buffer/Comparator**

This function block stores and compares the IDs of all UM82C59A-2's used in the system. The associated three I/O pins (CAS0-2) are outputs when the UM82C59A-2 is used as a master and are inputs when the UM82C59A-2

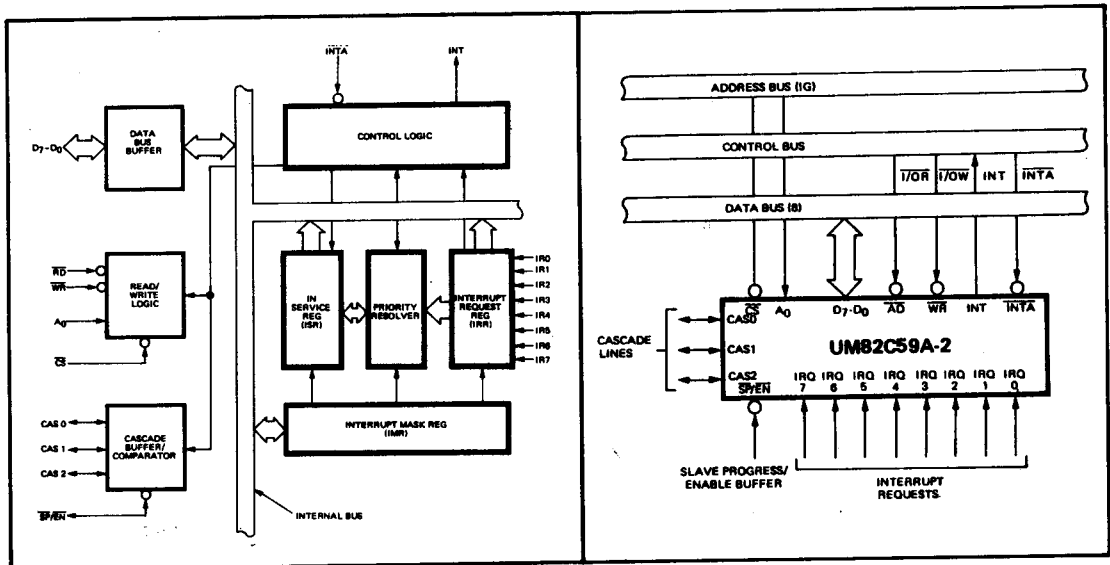


Figure 4. UM82C59A-2 Block Diagram

Figure 5. UM82C59A-2 Interface to Standard System Bus

is used as a slave. As a master, the UM82C59A-2 sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive  $\overline{INTA}$  pulses. (See section "Cascading the UM82C59A-2".)

### Interrupt Sequence

The powerful features of the UM82C59A-2 in a micro-computer system are its programmability and the interrupt routine addressing capability. The latter allows direct or indirect jumping to the specified interrupt routine requested without any polling of the interrupting devices. The normal sequence of events during an interrupt depends on the type of CPU being used.

The events occur as follows in an MCS-80/85 system:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The UM82C59A-2 evaluates these requests, and sends an INT to the CPU, if appropriate.
3. The CPU acknowledges the INT and responds with an  $\overline{INTA}$  pulse.
4. Upon receiving an  $\overline{INTA}$  from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The UM82C59A-2 will also release a CALL instruction code (11001101) onto the 8-bit data bus through D<sub>0</sub>-D<sub>7</sub> pin.
5. This CALL instruction will initiate two more  $\overline{INTA}$  pulses to be sent to the UM82C59A-2 from the CPU group.
6. These two  $\overline{INTA}$  pulses allow the UM82C59A-2 to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first  $\overline{INTA}$  pulse and the higher 8-bit address is released at the second  $\overline{INTA}$  pulse.
7. This completes the 3-byte CALL instruction released by the UM82C59A-2. In the AEOI mode, the ISR bit is reset at the end of the third  $\overline{INTA}$  pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt sequence.

The events occurring in an 80C86 system are the same until step 4.

4. Upon receiving an  $\overline{INTA}$  from the CPU group, the highest priority ISR bit is set and the corresponding IRR bit is reset. The UM82C59A-2 does not drive the data bus during this cycle.
5. The 80C86 will initiate a second  $\overline{INTA}$  pulse. During this pulse, the UM82C59A-2 releases an 8-bit pointer onto the Data Bus where it is read by the CPU.
6. This completes the interrupt cycle. In the AEOI mode,

the ISR bit is reset at the end of the second  $\overline{INTA}$  pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

If no interrupt is present at step 4 of either sequence (i.e., the request was too short in duration), the UM82C59A-2 will issue an interrupt level 7. Both the vectoring bytes and the CAS lines will look like an interrupt level 7 was requested.

### Interrupt Sequence Outputs

#### MCS<sup>®</sup>-80, MCS-85

This sequence is timed by three  $\overline{INTA}$  pulses. During the first  $\overline{INTA}$  pulse, the CALL opcode is enabled onto the data bus.

#### Content of First Interrupt Vector Byte

	D7	D6	D5	D4	D3	D2	D1	D0
CALL CODE	1	1	0	0	1	1	0	1

During the second  $\overline{INTA}$  pulse, the lower address of the appropriate service routine is enabled onto the Data Bus. When interval = 4 bits, A<sub>5</sub>-A<sub>7</sub> are programmed, while A<sub>0</sub>-A<sub>4</sub> are automatically inserted by the UM82C59A-2. When interval = 8, only A<sub>4</sub> and A<sub>7</sub> are programmed, while A<sub>0</sub>-A<sub>5</sub> are automatically inserted.

#### Content of Second Interrupt Vector Byte

IR	Interval = 4							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	A5	1	1	1	0	0
6	A7	A6	A5	1	1	0	0	0
5	A7	A6	A5	1	0	1	0	0
4	A7	A6	A5	1	0	0	0	0
3	A7	A6	A5	0	1	1	0	0
2	A7	A6	A5	0	1	0	0	0
1	A7	A6	A5	0	0	0	0	0
0	A7	A6	A5	0	0	0	0	0

IR	Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0
7	A7	A6	1	1	1	0	0	0
6	A7	A6	1	1	0	0	0	0
5	A7	A6	1	0	1	0	0	0
4	A7	A6	1	0	0	0	0	0
3	A7	A6	0	1	1	0	0	0
2	A7	A6	0	1	0	0	0	0
1	A7	A6	0	0	1	0	0	0
0	A7	A6	0	0	0	0	0	0

During the third  $\overline{INTA}$  pulse, the higher address of the appropriate service routine, which was programmed as byte 2 of the initialization sequence ( $A_5-A_{15}$ ), is enabled onto the bus.

### Content of Third Interrupt Vector Byte

D7	D6	D5	D4	D3	D2	D1	D0
A15	A14	A13	A12	A11	A10	A9	A8

### 80C86, 80C88

80C86, 80C88 mode is similar to MCS-80 mode except that only two interrupt Acknowledge cycles are issued by the processor and no CALL opcode is sent to the processor. The first interrupt acknowledge cycle is similar to that of MCS-80,85 systems in that the UM82C59A-2 uses it to internally freeze the state of the interrupts for priority resolution and, as a master, it issues the interrupt code on the cascade lines at the end of the  $\overline{INTA}$  pulse. On this first cycle, it does not issue any data to the processor and leaves its Data Bus buffers disabled. On the second interrupt acknowledge cycle in 80C86, 80C88 mode, the master (or slave if so programmed) will send a byte of data to the processor with the acknowledged interrupt code composed as follows (note the state of the ADI mode control is ignored and  $A_5-A_{11}$  are unused in 80C86, 80C88 mode.):

**Content of Interrupt Vector Byte for  
80C86, 80C88 System Mode**

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

### Programming the UM82C59A-2

The UM82C59A-2 accepts two types of command words generated by the CPU:

1. Initialization Command Words (ICWs): Before normal operation can begin, each UM82C59A-2 in the system must be brought to a starting point — by a sequence of 2 to 4 bytes timed by  $\overline{WR}$  pulses.
2. Operation Command Words (OCWs): These are the

command words which command the UM82C59A-2 to operate in various interrupt modes. These modes are:

- a. Fully nested mode
- b. Rotating priority mode
- c. Special mask mode
- d. Polled mode

The OCWs can be written into the UM82C59A-2 anytime after initialization.

### Initialization Command Words (ICWS)

#### General

Whenever a command is issued with  $A_0 = 0$  and  $D_4 = 1$ , this is interpreted as Initialization Command Word 1 (ICW1). ICW1 starts the initialization sequence during which the following automatically occur.

- a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low-to-high transition to generate an interrupt.
- b. The Interrupt Mask Register is cleared.
- c. IR7 input is assigned priority 7.
- d. The slave mode address is set to 7.
- e. Special Mask Mode is cleared and Status Read is set to IRR.
- f. If IC4 = 0, then all functions selected in ICW4 are set to zero. (Non-Buffered mode\*, no Auto-EOI, MCS-80, 85 system).

\*NOTE: Master/Slave in ICW4 is only used in the buffered mode.

#### Initialization Command Words 1 and 2 (ICW1, ICW2)

$A_5-A_{15}$ : Page starting address of service routines. In an MCS-80,85 system, the 8 request levels will generate CALLS to 8 locations equally spaced in memory. These can be programmed to be spaced at intervals of 4 or 8 memory locations, thus the 8 routines will occupy a page of 32 or 64 bytes, respectively.

The address format is 2 bytes long ( $A_0-A_{15}$ ). When the routine interval is 4,  $A_0-A_4$  are automatically inserted by the UM82C59A-2, while  $A_5-A_{15}$  are programmed externally. When the routine interval is 8,  $A_0-A_5$  are automatically inserted by the UM82C59A-2 while  $A_6-A_{15}$  are programmed externally.

The 8-byte interval will maintain compatibility with current software, while the 4-byte interval is best for a compact jump table.

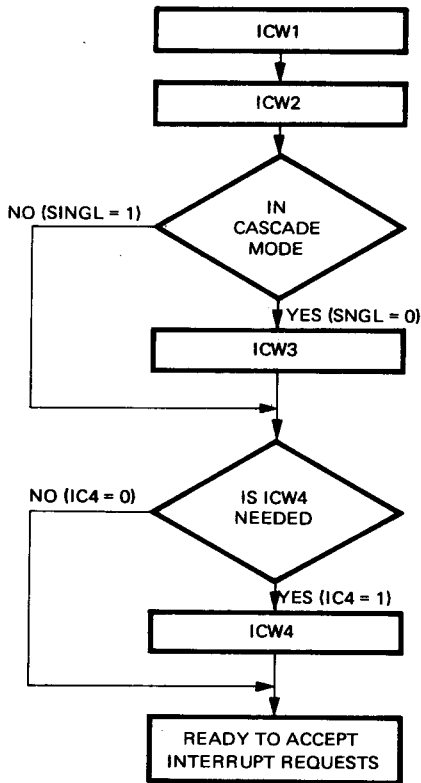
In an 80C86, 80C88 system,  $A_{15}$ - $A_{11}$  are inserted in the five most significant bits of the vectoring byte and the UM82C59A-2 sets the three least significant bits according to the interrupt level.  $A_{10}$ - $A_5$  are ignored and ADI (Address interval) has no effect.

**LTIM:** If  $LTIM = 1$ , then the UM82C59A-2 will operate in the level interrupt mode. Edge detect logic on the interrupt inputs will be disabled.

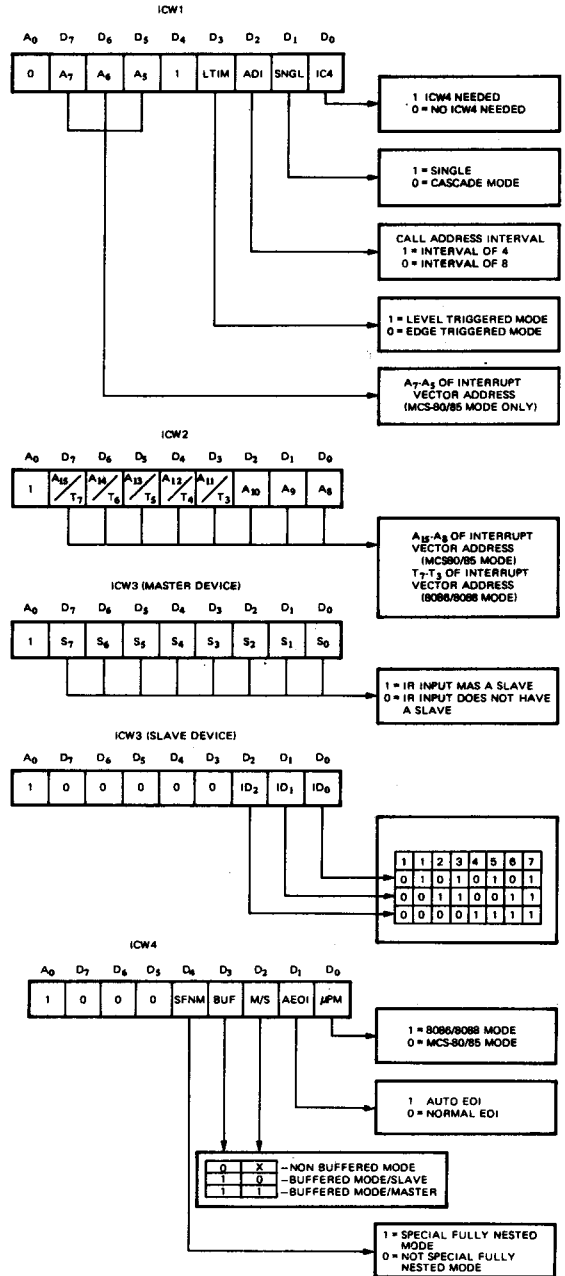
**ADI:** CALL address interval.  $ADI = 1$  then interval = 4;  $ADI = 0$  then interval = 8.

**SNGL:** Single. Means that this is the only UM82C59A-2 in the system. If  $SNGL = 1$ , no ICW3 will be issued.

**IC4:** If this bit is set – ICW4 has to be issued. If ICW4 is not needed, set  $IC4 = 0$ .



**Figure 4. Initialization Sequence**



NOTE: Slave ID is equal to the corresponding master IR input

**Figure 7. Initialization Command Word Format**

**Initialization Command Word 3 (ICW3)**

This word is read only when there is more than one UM82C59A-2 in the system and cascading is used, in which case SNGL = 0. It will load the 8-bit slave register. The functions of this register are:

- a. In the master mode (either when SP = 1, or in buffered mode when M/S = 1 in ICW4), a "1" is set for each slave in the system. The master then will release byte 1 of the call sequence (for MCS80/85 system) and will enable the corresponding slave to release bytes 2 and 3 (for 80C86, 80C88 only byte 2) through the cascade lines.
- b. In the slave mode (either when SP = 0, or if BUF = 1 and M/S = 0 in ICW4), bits 2-0 identify the slave. The slave compares its cascade input with these bits and if they are equal, bytes 2 and 3 of the call sequence (or just byte 2 for 80C86, 80C88) are released by it on the Data Bus.

**Initialization Command Word 4 (ICW4)**

**SFNM:** If SFNM = 1, the special fully nested mode is programmed.

**BUF:** If BUF = 1, the buffered mode is programmed. In buffered mode, SP/EN becomes an enable output and the master/slave determination is by M/S.

**M/S:** If buffered mode is selected: M/S = 1 means the UM82C59A-2 is programmed to be a master, M/S = 0 means the UM82C59A-2 is programmed to be a slave. If BUF = 0, M/S has no function.

**AEOI:** If AEOI = 1, the automatic and of interrupt mode is programmed.

**μPM:** Microprocessor mode: μPM = 0 sets the UM82C59A-2 for MCS-80,85 system operation, μPM = 1 sets the UM82C59A-2 for 80C86 system operation.

**Operation Command Words (OCWs)**

After the initialization Command Words (ICWs) are programmed into the UM82C59A-2, the device is ready to accept interrupt requests at its input lines. However, during the UM82C59A-2 operation, a selection of algorithms can command the UM82C59A-2 to operate in various modes through the Operation Command Words (OCWs).

**Operation Control Words (OCWs)**

A0 D7 D6 D5 D4 D3 D2 D1 D0

**OCW1**

1	M7	M6	M5	M4	M3	M2	M1	M0
---	----	----	----	----	----	----	----	----

**OCW2**

0	R	SL	EOI	0	0	L2	L1	L0
---	---	----	-----	---	---	----	----	----

**OCW3**

0	0	ESMM	SMM	0	1	P	RR	RIS
---	---	------	-----	---	---	---	----	-----

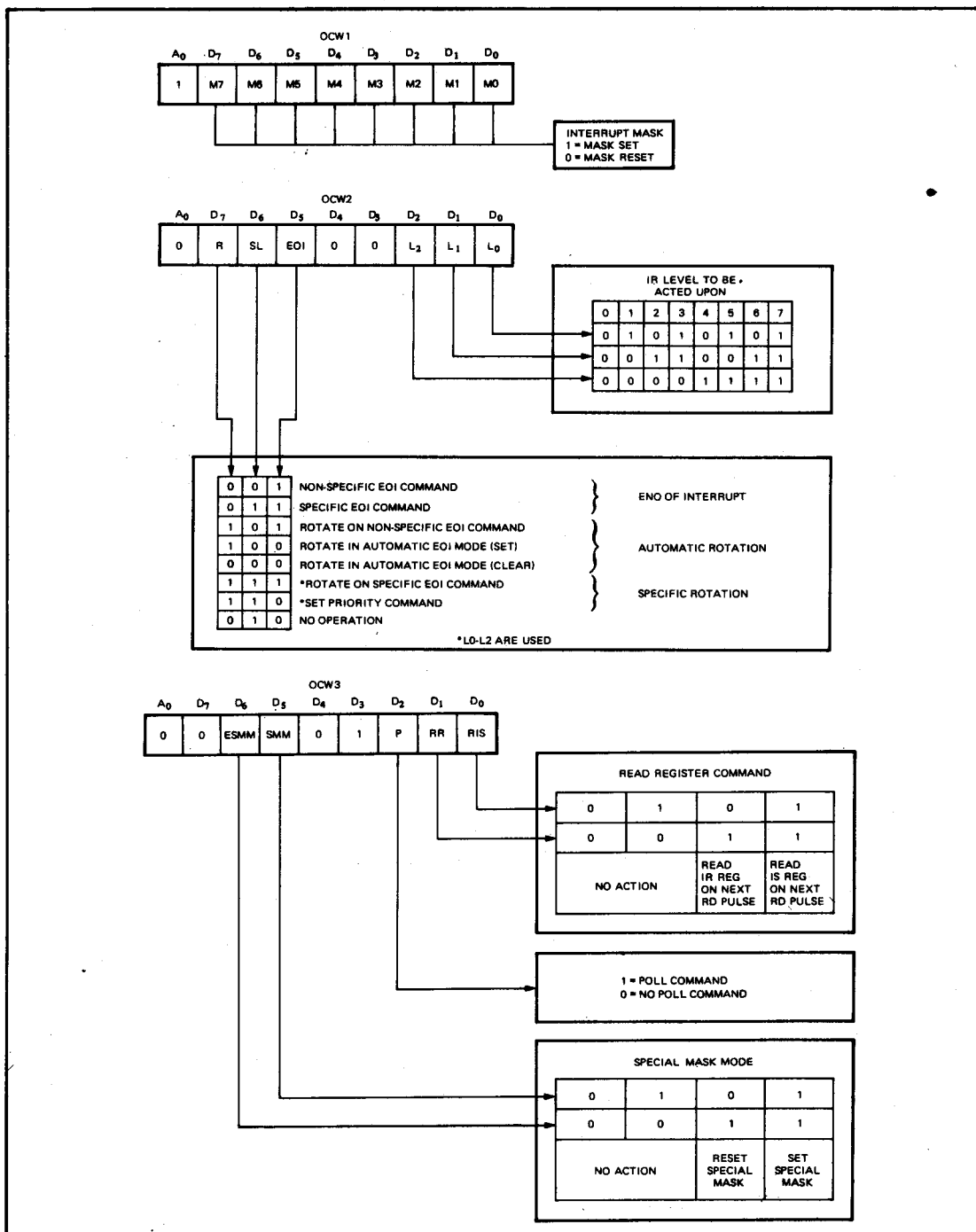
**Operation Control Word 1 (OCW1)**

OCW1 sets and clears the mask bits in the interrupt Mask Register (IMR). M<sub>7</sub>-M<sub>0</sub> represent the eight mask bits. M = 1 indicates the channel is masked (inhibited), M = 0 indicates the channel is enabled.

**Operation Control Word 2 (OCW2)**

R, SL, EDI – These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations can be found on the Operation Command Word Format.

L<sub>2</sub>, L<sub>1</sub>, L<sub>0</sub> – These bits determine the interrupt level acted upon when the SL bit is active.



**Figure 8. Operation Command Word Format**

### Operation Control Word 3 (OCW3)

ESMM — Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0, the SMM bit becomes a “don’t care”.

SMM — Special Mask Mode. If ESMM = 1 and SMM = 1, the UM82C59A-2 will enter Special Mask Mode. If ESMM = 1 and SMM = 0, the UM82C59A-2 will revert to normal mask mode. When ESMM = 0, SMM has no effect.

### Fully Nested Mode

This mode is entered after initialization unless another mode is programmed. The interrupt requests are ordered in priority from 0 through 7 (0 highest). When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus. Additionally, a bit of the Interrupt Service register (IS0-7) is set. This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine, or if AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA. While the IS bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels will generate an interrupt (which will be acknowledged only if the microprocessor internal interrupt enable flip-flop has been re-enabled through software).

After the initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained, in the rotating priority mode.

### End Of Interrupt (EOI)

The In Service (IS) bit can be reset either automatically following the trailing edge of the last in sequence INTA pulse (when AEOI bit in ICW4 is set) or by a command word that must be issued to the UM82C59A-2 before returning from a service routine (EOI command). An EOI command must be issued twice if in the Cascade mode, once for the master and once for the corresponding slave.

There are two forms of EOI command: Specific and Non-Specific. When the UM82C59A-2 is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a Non-Specific EOI command is issued the UM82C59A-2 will automatically reset the highest IS bit of those that are

set, since in the fully nested mode the highest IS level was necessarily the last level acknowledged and serviced. A non-specific EOI can be issued with OCW2 (EOI = 1, SL = 0, R = 0).

When a mode is used which may disturb the fully nested structure, the UM82C59A-2 may no longer be able to determine the last level acknowledged. In this case a Specific End of Interrupt must be issued which includes as part of the command the IS level to be reset. A specific EOI can be issued with OCW2 (EOI = 1, SL = 1, R = 0, and L0-L2 is the binary level of the IS bit to be reset).

It should be noted that an IS bit that is masked by an IMR bit will not be cleared by a non-specific EOI if the UM82C59A-2 is in the Special Mask Mode.

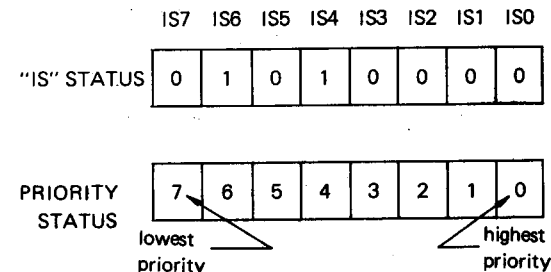
### Automatic End of Interrupt (AEOI) Mode

If AEOI = 1 in ICW4, then the UM82C59A-2 will operate in AEOI mode continuously until reprogrammed by ICW4. In this mode the UM82C59A-2 will automatically perform a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse (third pulse in MCS-80/85, second in 80C86/88). Notes that from a system standpoint, this mode should be used only when a nested multi-level interrupt structure is not required within a single UM82C59A-2.

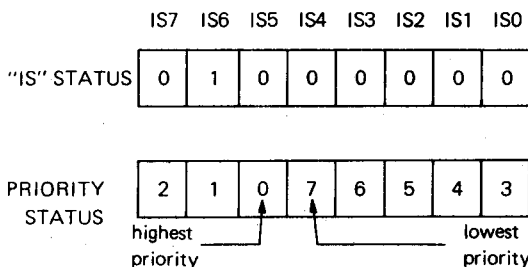
### Automatic Rotation (Equal Priority Devices)

In some applications there are a number of interrupting devices of equal priority. In this mode a device, after being serviced, receives the lowest priority, so a device requesting an interrupt will have to wait, in the worst case until each of 7 other devices are serviced at most once. For example, if the priority and “in service” status is:

Before Rotate (IR4 the highest priority requiring service)



After Rotate (IR4 was serviced, all other priorities rotated correspondingly)



There are two ways to accomplish Automatic Rotation using OCW2, the Rotation on Non-Specific EOI Command (R = 1 SL = 0, EOI = 1) and the Rotate in Automatic EOI Mode which is set by (R = 1, SL = 0, EOI = 0) and cleared by (R = 0, SL = 0, EOI = 0).

### Specific Rotation (Specific Priority)

The programmer can change priorities by programming the bottom priority and thus fixing all other priorities; i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

The Set Priority command is issued in OCW2 where: R = 1, SL = 1; L0-L2 is the binary priority level code of the bottom priority device.

Observe that in this mode internal status is updated by software control during OCW2. However, it is independent of the End of Interrupt (EOI) command (also executed by OCW2). Priority changes can be executed during an EOI command by using the Rotate on Specific EOI command in OCW2 (R = 1, SL = 1, EOI = 1 and L0-L2 = IR level to receive bottom priority).

### Interrupt Masks

Each Interrupt Request input can be masked individually by the Interrupt Mask Register (IMR) programmed through OCW1. Each bit in the IMR masks one interrupt channel if it is set (1). Bit 0 masks IR0, Bit 1 masks IR1 and so forth. Masking an IR channel does not affect the other channels operation.

### Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for

a portion of its execution but enable some of them for another portion.

The difficulty here is that if an Interrupt Request is acknowledged and an End of Interrupt command did not reset its IS bit (i.e., while executing a service routine), the UM82C59A-2 would have inhibited all lower priority requests with no easy way for the routine to enable them.

That is where the Special Mask Mode comes in. In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.

Thus, any interrupts may be selectivity enabled by loading the mask register.

The special Mask Mode is set by OCW3 where: SSMM = 1, SMM = 1, and cleared where SSMM = 1, SMM = 0.

### Poll Command

In this mode, the INT output is not used or the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

The Poll command is issued by setting P = "1" in OCW3. The UM82C59A-2 treats the next  $\overline{RD}$  pulse to the UM82C59A-2 (i.e.,  $\overline{RD} = 0$ ,  $\overline{CS} = 0$ ) as an interrupt acknowledge, sets the appropriate IS bit if there is a request, and reads the priority level. Interrupt is frozen from  $\overline{WR}$  to  $\overline{RD}$ .

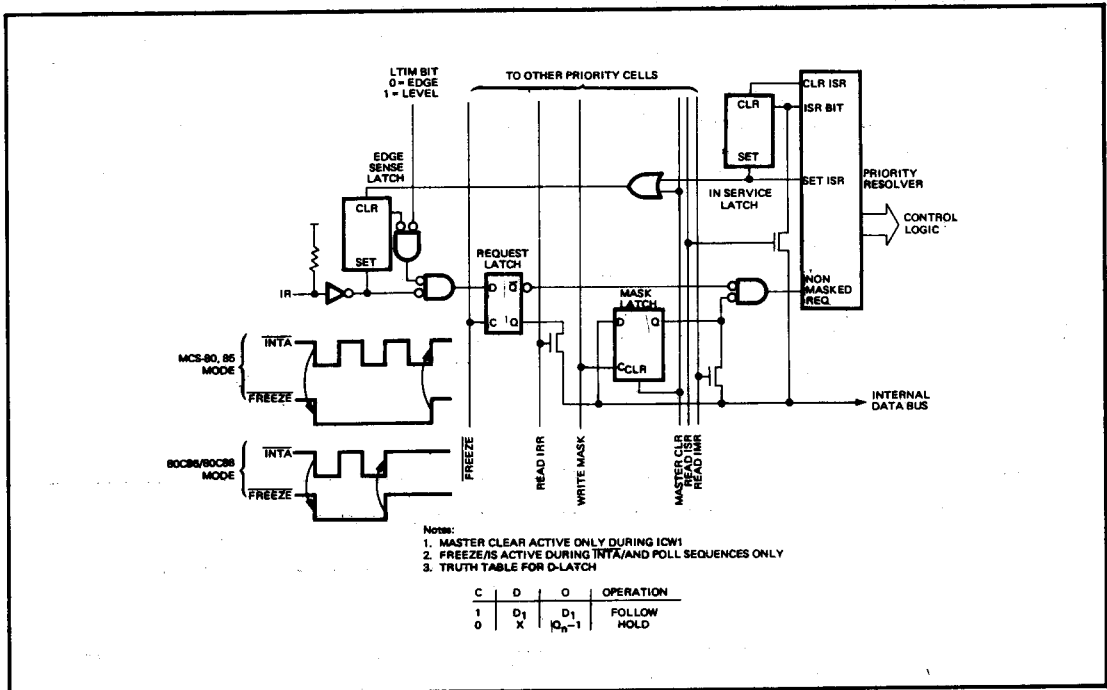
The word enabled onto the data bus during  $\overline{RD}$  is:

D7	D6	D5	D4	D3	D2	D1	D0
1	-	-	-	-	W2	W1	W0

W0-W2: Binary code of the highest priority level requesting service.

1: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine command common to several levels so that the  $\overline{INTA}$  sequence is not needed (saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.



**Figure 9. Priority Cell – Simplified Logic Diagram**

### Reading the UM82C59A-2 Status

The input status of several internal register can be read to update the user information on the system. The following registers can be read via OCW3 (IRR and ISR or OCW1 (IMR)).

**Interrupt Request Register (IRR):** 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

**In-Service Register (ISR):** 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt Command is issued.

**Interrupt Mask Register:** 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when prior to the  $\overline{RD}$  pulse, a Read Register Command is issued with OCW3 (RR = 1, RIS = 1).

There is no need to write an OCW3 before every status read operation, as long as the status read corresponds with the previous one; i.e., the UM82C59A-2 "remembers" whether the IRR or ISR has been previously selected by the OCW3. This is not true when poll is used. After initialization, the UM82C59A-2 is set to IRR.

For reading the IMR, no OCW3 is needed. The output data bus will contain the IMR whenever  $\overline{RD}$  is active and A0 = 1 (OCW1). Polling overrides status read when P = 1, RR = 1 in OCW3.

### Edge and Level Triggered Modes

This mode is programmed using bit 3 in ICW1.

If LTIM = '0', an interrupt request will be recognized by a low to high transition on an IR input. The IR input can remain high without generating another interrupt.

If LTIM = '1', an interrupt request will be recognized by a 'high' level on IR Input, and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued or the CPU

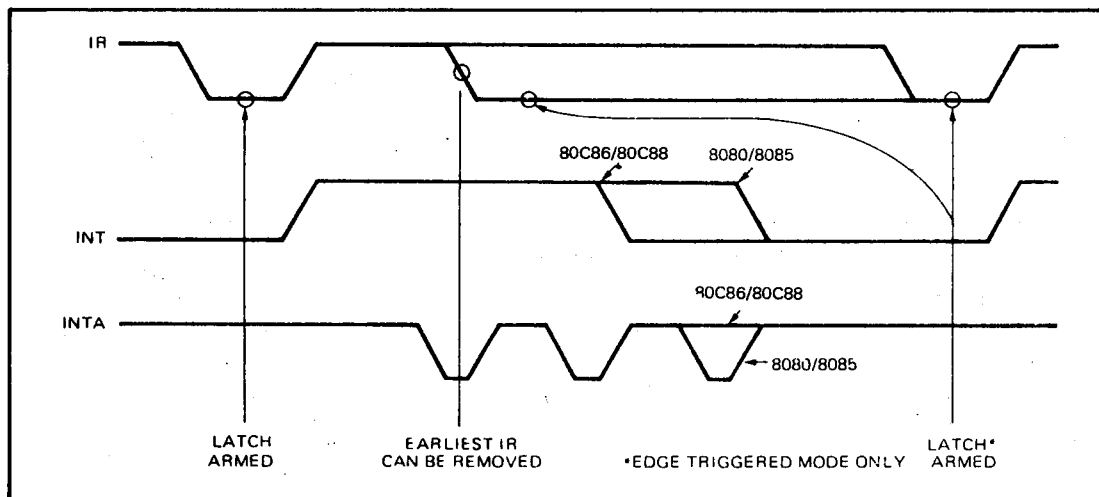
interrupt is enabled to prevent a second interrupt from occurring.

The priority cell diagram shows a conceptual circuit of the level sensitive and edge sensitive input circuitry of the UM82C59A-2. Be sure to note that the request latch is a transparent D type latch.

In both the edge and level triggered modes the IR inputs must remain high until after the falling edge of the first INTA. If the IR input goes low before this time a DEFAULT IR7 will occur when the CPU acknowledges the interrupt. This can be a useful safeguard for detecting interrupts caused by spurious noise glitches on the IR inputs. To implement this feature the IR7 routine is

used for "clean up" simply executing a return instruction, thus ignoring the interrupt. If IR7 is needed for other purposes a default IR7 can still be detected by reading the ISR. A normal IR7 interrupt will set the corresponding ISR bit, a default IR7 won't. If a default IR7 routine occurs during a normal IR7 routine, however, the ISR will remain set. In this case it is necessary to keep track of whether or not the IR7 routine was previously entered. If another IR7 occurs it is a default.

In power sensitive applications, it is advisable to place the UM82C59A-2 in the edge-triggered mode. This will minimize the current through the pull-up resistors on the IR pins.



**Figure 10. IR Triggering Timing Requirements**

### The Special Fully Nested Mode

This mode will be used in the case of a big system where cascading is used, and the priority has to be conserved within each slave. In this case the fully nested mode will be programmed to the master (using ICW4). This mode is similar to the normal nested mode with the following exceptions:

- When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic and further interrupt requests from higher priority IR's within the slave will be recognized by

the master and will initiate interrupts to the processor. (In the normal nested mode a slave is masked out when its request is in service and no higher requests from the same slave can be serviced.)

- When exiting the Interrupt Service routine the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a non-specific End of Interrupt (EOI) command to the slave and then reading its In-Service register and checking for zero. If it is empty, a non-specified EOI can be sent to the master, too. If not, no EOI should be sent.

**Buffered Mode**

When the UM82C59A-2 is used in a large system where bus driving buffers are required on the data bus and the cascading mode is used, there exists the problem of enabling buffers.

The buffered mode will structure the UM82C59A-2 to send an enable signal on  $\overline{SP/EN}$  to enable the buffers. In this mode, whenever the UM82C59A-2's data bus outputs are enabled, the  $\overline{SP/EN}$  output becomes active.

This modification forces the use of software programming to determine whether the UM82C59A-2 is a master or a slave. Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW3 determines whether it is a master or a slave.

**Cascade Mode**

The UM82C59A-2 can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

The master controls the slaves through the 3 line cascade bus. The cascade bus acts like chip selects to the slaves during the  $\overline{INTA}$  sequence.

In a cascade configuration, the slave interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and afterwards acknowledged, the master will enable the corresponding slave to release the device routine address during bytes 2 and 3 of  $\overline{INTA}$ . (Byte 2 only for 80C86/8088).

The cascade bus lines are normally low and will contain the slave address code from the trailing edge of the first  $\overline{INTA}$  pulse to the trailing edge of the third pulse. Each UM82C59A-2 in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: Once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select (CS) input of each UM82C59A-2.

The cascade lines of the Master UM82C59A-2 are activated only for slave inputs, non slave inputs leave the cascade line inactive (low).

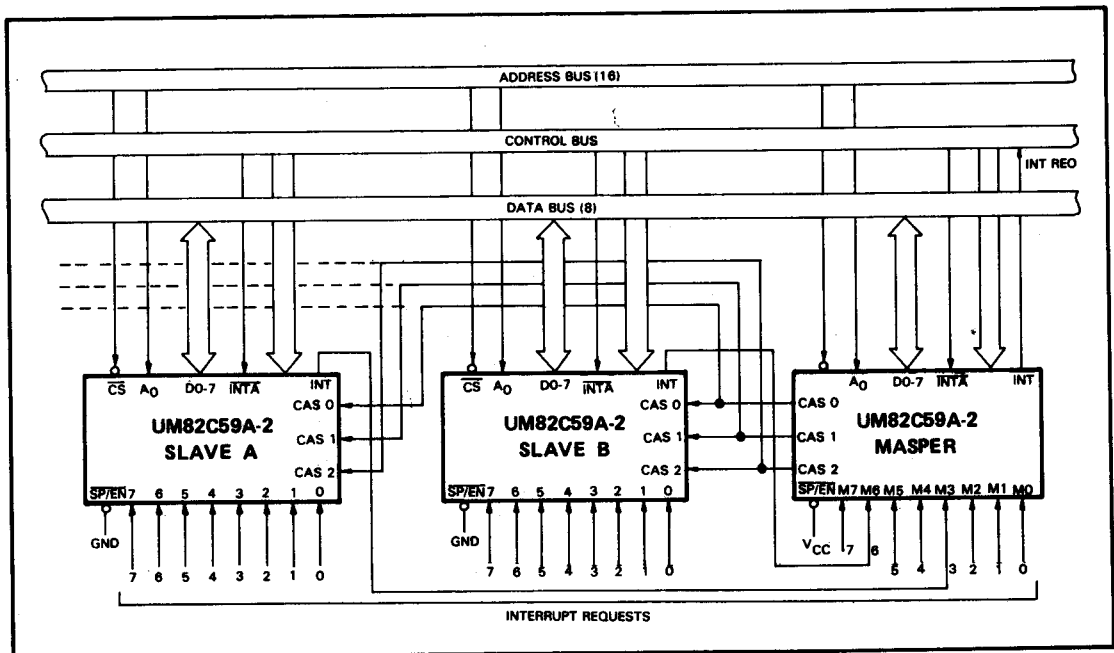


Figure 11. Cascading the UM82C59A-2