

	PAGE
<b>Introduction</b> .....	2
<b>1.0 Glossary of Terms for the 82C59A</b> .....	2
1.1 Automatic End of Interrupt (AEOI) .....	2
1.2 Automatic Rotation .....	2
1.3 Buffered Mode .....	3
1.4 Cascade Mode .....	3
1.5 End of Interrupt (EOI) .....	3
1.6 Fully Nested Mode .....	4
1.7 Master .....	4
1.8 Slave .....	4
1.9 Special Fully Nested Mode .....	4
1.10 Special Mask Mode .....	4
1.11 Specific Rotation .....	4
<b>2.0 Initialization Control Words</b> .....	5
2.1 ICW1 .....	5
2.2 ICW2 .....	7
2.3 ICW3 .....	7
2.4 ICW4 .....	8
<b>3.0 Operation Command Words</b> .....	9
3.1 OCW1 .....	9
3.2 OCW2 .....	9
3.3 OCW3 .....	10
<b>4.0 Addressing the 82C59A</b> .....	11
<b>5.0 Programming the 82C59A</b> .....	12
5.1 Example 1: Single 82C59A .....	12
5.2 Example 2: Cascaded 82C59As .....	13
<b>6.0 Expansion Past 64 Interrupts</b> .....	14

## Introduction

The Intersil 82C59A is a CMOS Priority Interrupt Controller, designed to relieve the system CPU from the task of polling in a multi-level priority interrupt system. The 82C59A is compatible with microprocessors such as the 80C86, 80C88, 8086, 8088, 8080/85 and NSC800.

In the following discussion, we will look at the initialization and operation process for the 82C59A. We will focus our attention on 80C86/80C88-based systems. However, the information presented will also be applicable to use of the 82C59A in 8080 and 8085-based systems as well.

Let us look at the sequence of events that occur with the 82C59A during an interrupt request and service. In an 8080/85 based system:

1. One or more of the INTERRUPT REQUEST lines (IR0 - IR7) are raised high, setting the corresponding bits in the Interrupt Request Register (IRR).
2. The interrupt is evaluated in the priority resolver. If appropriate, an interrupt is sent to the CPU via the INT line (pin 17).
3. The CPU acknowledges the interrupt by sending a pulse on the  $\overline{\text{INTA}}$  line. Upon reception of this pulse, the 82C59A responds by forcing the opcode for a call instruction (0CDH) onto the data bus.
4. A second  $\overline{\text{INTA}}$  pulse is sent from the CPU. At this time, the device will respond by placing the lower byte of the address of the appropriate service routine onto the data bus. This address is derived from ICW1.
5. A final (third) pulse of  $\overline{\text{INTA}}$  occurs, and the 82C59A responds by placing the upper byte of the address onto the data bus. This address is taken from ICW2.
6. The three byte call instruction is then complete. If the AEOI mode has been chosen, the bit set during the first  $\overline{\text{INTA}}$  pulse in the ISR is reset at the end of the third  $\overline{\text{INTA}}$  pulse. Otherwise, it will not get reset until an appropriate EOI command is issued to the 82C59A.

For 80C86- and 80C88-based systems:

1. and 2. same as above.
2. The CPU responds to the interrupt request by pulsing the  $\overline{\text{INTA}}$  line twice. The first pulse sets the appropriate ISR bit and resets the IRR bit while the second pulse causes the interrupt vector to be placed on the data bus. This byte is composed of the interrupt number in bits 0 through 2, and bits 3 through 7 are taken from bits 3 - 7 of ICW2.
3. The interrupt sequence is complete. If using the AEOI mode, the bit set earlier in the ISR will be reset. Otherwise, the interrupt controller will await an appropriate EOI command at the end of the interrupt service routine.

## 1.0 Glossary of Terms for the 82C59A

### 1.1 Automatic End of Interrupt (AEOI)

When the 82C59A is programmed to operate in the Automatic EOI mode, the device will produce its own End-of-Interrupt (EOI) at the trailing edge of the last Interrupt

Acknowledge pulse ( $\overline{\text{INTA}}$ ) from the CPU. Using this mode of operation frees the software (service routines) from needing to send an EOI manually to the 82C59A.

However, using the Automatic EOI mode will upset the priority structure of the 82C59A. When the AEOI is generated, the bit that was set in the In-Service Register (ISR) to indicate which interrupt is being serviced, will be cleared. Because of this, while an interrupt is being serviced there will be no record in the ISR that it is being serviced. Unless interrupts are disabled by the CPU, there is a risk that interrupt requests of lower or equal priority will interrupt the current request being serviced. If this mode of operation is not desired, interrupts should not be re-enabled by the CPU when executing interrupt service routines.

### 1.2 Automatic Rotation

During normal operation of the 82C59A, we have an assigned order of priorities for the IR lines. There are however, instances when it might be useful to assign equal priorities to all interrupts. Once a particular interrupt has been serviced, all other equal priority interrupts should have an opportunity to be serviced before the original peripheral can be serviced again. This priority equalization can be achieved through Automatic Rotation of priorities.

Assume, for example, that the assigned priorities of interrupts has IR0 as the highest priority interrupt and IR7 as the lowest. Figure 1A shows interrupt requests occurring on IR7 as well as IR3. Because IR3 is of higher priority, it will be serviced first. Upon completion of the servicing of IR3, rotation occurs and IR3 then becomes the lowest priority interrupt. IR4 will now have the highest priority (See Figure 1B).

There are two methods in which Automatic Rotation can be implemented. First, if the 82C59A is operating in the AEOI mode as described above, the 82C59A can be programmed for "Rotate in Automatic EOI mode". This is done by writing a command word to OCW2. The second method occurs when using normal EOIs. When an EOI is issued by the service routine, the software can specify that rotation be performed.

	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
IRR STATUS	1	0	0	0	1	0	0	0
PRIORITY	7	6	5	4	3	2	1	0
	LOWEST PRIORITY							HIGHEST PRIORITY

FIGURE 1A. IR PRIORITIES (BEFORE ROTATION)

	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
IRR STATUS	1	0	0	0	0	0	0	0
PRIORITY	3	2	1	0	7	6	5	4
	HIGHEST PRIORITY							LOWEST PRIORITY

FIGURE 1B. IR PRIORITIES (AFTER ROTATION)



### 1.6 Fully Nested Mode

By default, the 82C59A operates in the Fully Nested Mode. It will remain in this mode until it is programmed otherwise. In the Fully Nested Mode, interrupts are ordered by priority from highest to lowest. Initially, the highest priority level is IR0 with IR7 having the lowest. This ordering can be changed through the use of priority rotation (see 1.2).

In the Fully Nested Mode, when an interrupt occurs, its corresponding bit will get set in the Interrupt Request Register (IRR). When the processor acknowledges the interrupt, the 82C59A will look to the IRR to determine the highest priority interrupt requesting service. The bit in the In-service Register (ISR) corresponding to this interrupt will then be set. This bit remains set until an EOI is sent to the 82C59A.

While an interrupt is being serviced, only higher priority interrupts will be allowed to interrupt the current interrupt being serviced. However, lower priority interrupts can be allowed to interrupt higher priority requests if the 82C59A is programmed for operation in the Special Mask Mode.

When using the 82C59A in an 80C86- or 80C88-based system, interrupts will automatically be disabled when the processor begins servicing an interrupt request. The current address and the state of the flags in the processor will be pushed onto the stack. The interrupt-enable flag is then cleared. To allow interrupts to occur at this point, the STI instruction can be used. Upon exiting the service routine using the IRET instruction, execution of the program is resumed at the point where the interrupt occurred, and the flags are restored to their original values, thus re-enabling interrupts.

A configuration in which the Fully Nested structure is not preserved occurs when one or more of the following conditions occur:

- (a) The Automatic EOI mode is being used.
- (b) The Special Mask Mode is in use.
- (c) A slave 82C59A has a master that is not programmed to the Special Fully Nested Mode.

Cases (a) and (b) differ from case (c) in that the 82C59A would allow lower priority interrupt requests the opportunity to be serviced before higher priority interrupt requests.

### 1.7 Master

When using multiple 82C59As in a system, one 82C59A has control over all other 82C59As. This is known as the "master" interrupt controller. Communication between the master and the other (slave) 82C59As occurs via the CAS0 - 2 lines. These lines form a private bus between the multiple 82C59As. Also, the INT lines from the slaves are routed to the master's IR input pin(s). See Figure 2.

### 1.8 Slave

A "slave" 82C59A in a system is controlled by a master 82C59A. There is but one "master" in the system, but there can be up to 8 slave 82C59As. The INT outputs from the slaves act as inputs to the master through its IR inputs.

Communications between the master and slaves occurs via the CAS0 - 2 lines. See Figure 2.

### 1.9 Special Fully Nested Mode

The Special Fully Nested Mode (SFNM) is used in a system having multiple 82C59As where it is necessary to preserve the priority of interrupts within a slave 82C59A. Only the master is programmed for the Special Fully Nested Mode through ICW4. This mode is similar to the Fully Nested Mode with the following exceptions:

- (a) When an interrupt from a particular slave is being serviced, additional higher priority interrupts from that slave can cause an interrupt to the master. Normally, a slave is masked out when its request is in service.
- (b) When exiting the Interrupt Service routine, the software should first issue a non-specific EOI to the slave. The Inservice Register (ISR) should then be read and checked to see if its contents are zero. If the register is empty, the software should then write a non-specific EOI to the master. Otherwise, a second EOI need not be written because there are interrupts from that slave still being processed.

NOTE: Because the Master 82C59A and its slave 82C59As must be in Fully Nested Mode for this mode to be functional, we could not utilize Automatic EOIs. These would disturb the Fully Nested structure, as described in section 1.6.

### 1.10 Special Mask Mode

The Special Mask Mode is utilized in order to allow interrupts from all other levels (higher and lower as well) to interrupt the IR level that is currently being serviced. Invoking this mode of operation will disturb the fully nested priority structure.

Generally, the Special Mask Mode is selected during the servicing of an interrupt. The software should first set the bit corresponding to the IR level being serviced, in the Interrupt Mask Register (OCW1). The Special Mask Mode and interrupts should then be enabled. This will allow any of the IR levels except for those masked off by OCW1 to interrupt the IR level currently being serviced.

Because this disturbs the Fully Nested Structure, it is required that a Specific EOI be issued when servicing interrupts while the Special Mask Mode is in effect. Before exiting the original interrupt routine, the Special Mask Mode should be disabled.

### 1.11 Specific Rotation

By issuing the proper command word to OCW2, the priority structure of the 82C59A can be dynamically altered. The command word written to OCW2 would specify which is to be the lowest priority IR level.

This specific rotation can be accomplished one of two ways. The first is through a specific EOI. The software can specify that rotation is to be applied to the IR level provided with the EOI. The second method is a simple "set priority" command, in which the lowest priority level is specified with the command word.

## 2.0 Initialization Control Words

The following section gives a description of the Initialization Control Words (ICW) used for configuring the 82C59A Interrupt controller. There are four (4) control words used for initialization of the 82C59A. These ICWs must be programmed in the proper sequence beginning with ICW1. If at any time during the course of operation the configuration of the 82C59A needs to be changed, the user must again write out the control words to the device in their proper order. The initialization sequence is shown in Figure 3.

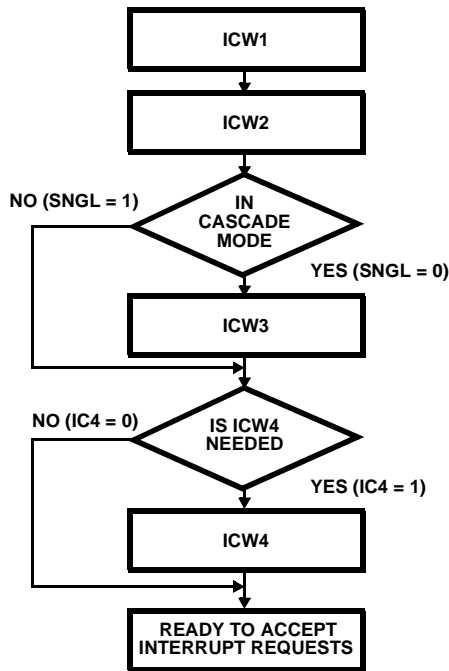


FIGURE 3. 82C59A INITIALIZATION SEQUENCE

**ICW1:** The 82C59A recognizes the first Initialization Control Word (ICW) written to it based on two criteria: (1) the A0 line from the address bus must be a zero, and (2) the D4 bit must be a one. If the D4 bit is set to a zero, we would be programming either OCW2 or OCW3 (these are explained later). The function of ICW1 is to tell the 82C59A how it is being used in the system (i.e. Single or cascaded, edge or level triggered interrupts etc.).

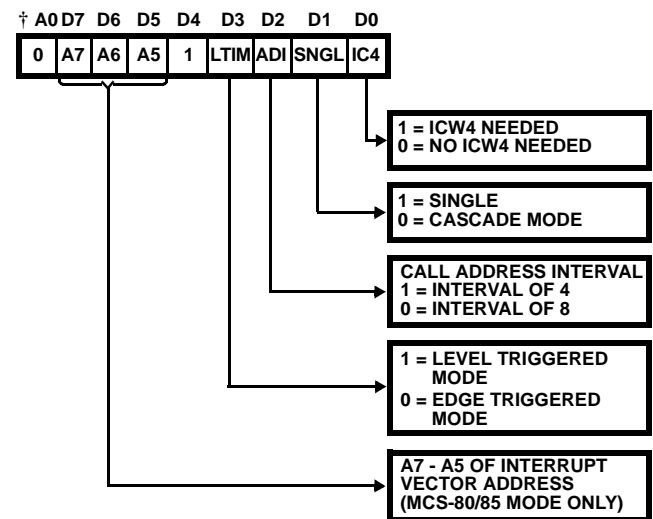
**ICW2:** This control word is always issued directly after ICW1. When addressing this ICW, the A0 line from the address bus must be a one (high). ICW2 is utilized in providing the CPU with information on where to vector to in memory when servicing an interrupt.

**ICW3:** This control word is issued only if the SNGL (D1) bit of the ICW1 has been programmed with a zero. When addressing this word, the A0 line from the CPU must be high (1). This control word is for cascaded 82C59As. It allows the master and slave 82C59As to communicate via the CAS0-2 lines. With the master, this word indicates which IR lines have slaves connected to them. For the slave 82C59A(s), this word indicates to which IR line on the master it is connected.

**ICW4:** Issuance of this ICW is selectable through the IC4 (D0) bit of ICW1. If ICW4 is to be written to the 82C59A, A0 from the CPU must be high (1) when writing to it. This word needs to be written only when the 82C59A is operating in modes other than the default modes. Instances when we would want to write to ICW4 are one or more of the following: An 80C86(80C88) processor is being used, buffered outputs (D0 - D7) are to be used, Automatic EOIs are desired, or the Special Fully Nested mode is to be used.

### 2.1 ICW1

ICW1 is the first control word that is written to the 82C59A during the initialization process. To access this word, the value of A0 must be a zero (0) in the addressing, and bit D4 of ICW1 must be a one (1). The format of the command word is as follows:



† A0 is an address bit, and not part of the ICW.

FIGURE 4. ICW1 FORMAT

**D7 thru D5 - A7, A6, A5:** These bits are used in the 8080/85 mode to form a portion of the low byte call address. When using the 4 byte address interval, all 3 bits are utilized. When using the 8 byte interval, only bits A7 and A6 are used. Bit A5 becomes a "don't care" bit. If using an 80C86(80C88) system, the value of these bits can be set to either a one or zero.

#### D3 - LTIM:

- 0: The 82C59A will operate in an edge triggered mode. An interrupt request on one of the IR lines (IR0 - IR7) is recognized by a low to high transition on the pin. The IR signal must remain high at least until the falling edge of the first  $\overline{INTA}$  pulse. Subsequent interrupts on the IR pin(s) will not occur until another low-to-high transition occurs.
- 1: Sets up the 82C59A to operate in the level triggered mode. Interrupts occur when a "high" level is detected on one or more of the IR pins. The interrupt request must be removed from this pin before the EOI command is issued by the CPU. Otherwise, the 82C59A will see the IR line still in a high state, and consider this to be another interrupt request.

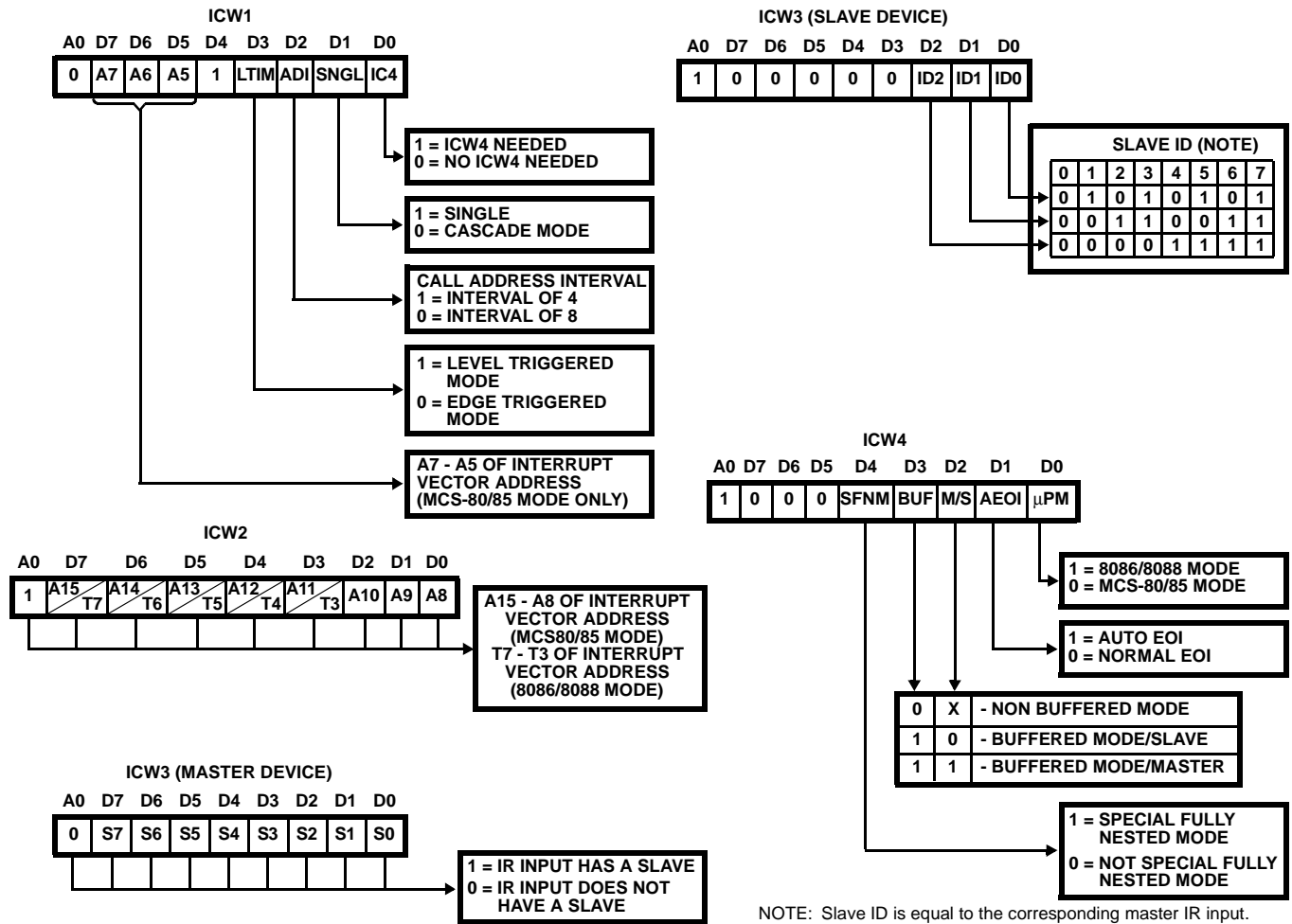


FIGURE 5. 82C59A INITIALIZATION COMMAND WORD FORMAT

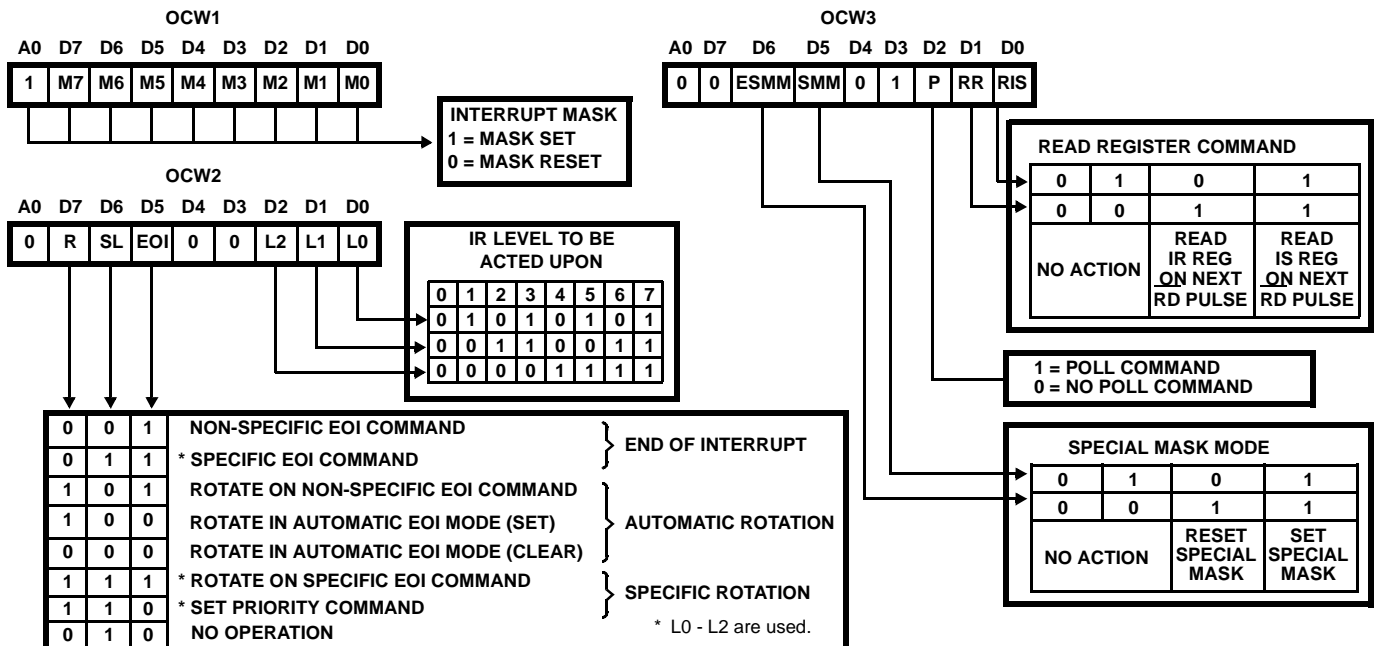


FIGURE 6. 82C59A OPERATION COMMAND WORD FORMAT

**D2 - ADI:** Call Address Interval (for 8080/8085 use only). If using the 82C59A in an 80C86/88 based system, the value of this bit can be either a 0 or 1.

- 0: The address interval generated by the 82C59A is 8 bytes. This option provides compatibility with the RST interrupt vectoring in 8080/8085 systems since the vector locations are 8 bytes apart. This vector will be combined with the values specified in bits D7 and D6 of ICW1. The addresses generated are shown in Table 1.

**TABLE 1. ADDRESS INTERVAL (8 BYTES)**

D7	D6	D5	D4	D3	D2	D1	D0	
A7	A6	1	1	1	0	0	0	IR7
A7	A6	1	1	0	0	0	0	IR6
A7	A6	1	0	1	0	0	0	IR5
A7	A6	1	0	0	0	0	0	IR4
A7	A6	0	1	1	0	0	0	IR3
A7	A6	0	1	0	0	0	0	IR2
A7	A6	0	0	1	0	0	0	IR1
A7	A6	0	0	0	0	0	0	IR0

- 1: The address interval generated by the interrupt controller will be 4 bytes. This provides the user with a compact jump table for 8080/8085 systems. The interrupt number is effectively multiplied by four and combined with bits D7, D6, and D5 to form the lower byte of the call instruction generated and sent to the 8080 and 8085. Table 2 shows how these addresses are generated for the various interrupt request (IR) levels.

**TABLE 2. ADDRESS INTERVAL (4 BYTES)**

D7	D6	D5	D4	D3	D2	D1	D0	
A7	A6	A5	1	1	1	0	0	IR7
A7	A6	A5	1	1	0	0	0	IR6
A7	A6	A5	1	0	1	0	0	IR5
A7	A6	A5	1	0	0	0	0	IR4
A7	A6	A5	0	1	1	0	0	IR3
A7	A6	A5	0	1	0	0	0	IR2
A7	A6	A5	0	0	1	0	0	IR1
A7	A6	A5	0	0	0	0	0	IR0

## D1 - SNGL:

- 0: This tells the 82C59A that more than one 82C59A is being used in the system, and it should expect to receive ICW3 following ICW2. How the particular 82C59A is being used in the system will be determined either through ICW4 for buffered mode, or through the  $\overline{SP/EN}$  pin for non-buffered mode operation.
- 1: Tells the 82C59A that it is being used alone in the system. Therefore, there will be no need to issue ICW3 to the device.

**D0 - IC4:** Specifies to the 82C59A whether or not it can expect to receive ICW4. If this device is being used in an 80C86/80C88 system, ICW4 must be issued.

- 0: ICW4 will not be issued. Therefore, all of the parameters associated with ICW4 will default to the zero (0) state. This should only be done when using the 82C59A in an 8080 or 8085 based system.

- 1: ICW4 will be issued to the 82C59A.

## 2.2 ICW2

ICW2 is the second control word that must be sent to the 82C59A. This byte is used in one of two ways by the 82C59A, depending on whether it is being used in an 8080/85 or an 80C86/88 based system.

When used in conjunction with the 8080/85 microprocessor, the value given to this register is taken as being the high byte of the address in the CALL instruction sent to the CPU.

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

**FIGURE 7. ICW2 FORMAT**

In an 80C86- or 80C88-based system, ICW2 is used to send the processor an interrupt vector. This vector is formed by taking the value of bits D7 through D3 and combining them with the interrupt request level to get an eight bit number. The processor will multiply this number by four and go to that absolute location in memory to find a starting address for the interrupt service routine corresponding to the interrupt request.

For example, if we set ICW2 to "00011000" and an interrupt is recognized on IR1, the vector sent to the 80C86(80C88) will be 00011001 (19H). The processor will then look to the memory location 64H to find the starting address of the corresponding interrupt service routine. It is the responsibility of the software to provide this address in the interrupt table.

D7	D6	D5	D4	D3	D2	D1	D0
A7	A6	A5	A4	A3	X	X	X

**FIGURE 8. ICW2 FORMAT (80C86 MODE)**

## 2.3 ICW3

ICW3 is only issued when the SNGL bit in ICW1 has been set to zero. If not set, the next word written to the 82C59A will be interpreted as ICW4 if A0 = 1 and IC4 from ICW4 was set to one, or it could see it as one of the Operation Command Words based upon the state of the A0 line.

Like ICW2, this control word can be interpreted in two ways by the 82C59A. However the interpretation of this word depends on whether the 82C59A is being used as a "master" or "slave" in the system. The definition of the particular device's role in the system is assigned through ICW4 (which

will be discussed later), or through the state of the  $\overline{SP/EN}$  pin (pin 16).

## 82C59A AS A MASTER

If a given 82C59A is being used as a master, the eight (8) bits in this command word are used to indicate which of the IR lines are being driven by a slave 82C59A.

D7	D6	D5	D4	D3	D2	D1	D0
S7	S6	S5	S4	S3	S2	S1	S0

FIGURE 9. ICW3 FORMAT (MASTER)

### D7 THRU D0:

- 0: The corresponding IR line to this bit is not being driven by a slave 82C59A. This line can however then be connected to the interrupt output of another interrupting device such as a UART. If there are unused bits in this byte because not all eight of the IR lines are used, set them to zero.
- 1: The corresponding IR line to this bit is being driven by a slave 82C59A.

The bits in this command word are directly related to the IR lines. For example, to tell the 82C59A that there is a slave device connected to IR5 (pin 23), bit D5 of the command word should be set to a one (1).

## 82C59A AS A SLAVE DEVICE

When the device is being used as a slave device, we must use ICW3 to inform itself as to which IR line it will be connected to in the master. Therefore, only the three (3) least significant bits of ICW3 will be used to specify this value.

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	ID2	ID1	ID0

FIGURE 10. ICW3 FORMAT (SLAVE)

These bits are coded as follows:

TABLE 3. SLAVE IDENTIFICATION WITH ICW3

MASTER IR NUMBER	ID2	ID1	ID0
IR7	1	1	1
IR6	1	1	0
IR5	1	0	1
IR4	1	0	0
IR3	0	1	1
IR2	0	1	0
IR1	0	0	1
IR0	0	0	0

For example, if the INT output of a "slave" 82C59A is connected to the input pin IR5 on the "master" 82C59A, ICW3 of the "slave" would be programmed with the value

00000101b, or 05H. This informs the "slave" as to which priority level it holds with the "master".

**D7 thru D3:** These bits must be set to zeros (0) for proper operation of the device.

## 2.4 ICW4

This control register is written to only when the IC4 bit is set in ICW1. The purpose of this command word is to set up the 82C59A to operate in a mode other than the default mode of operation. The default mode of operation is the same as if a value of 00H were to be written to ICW4 (i.e. all bits set to zero).

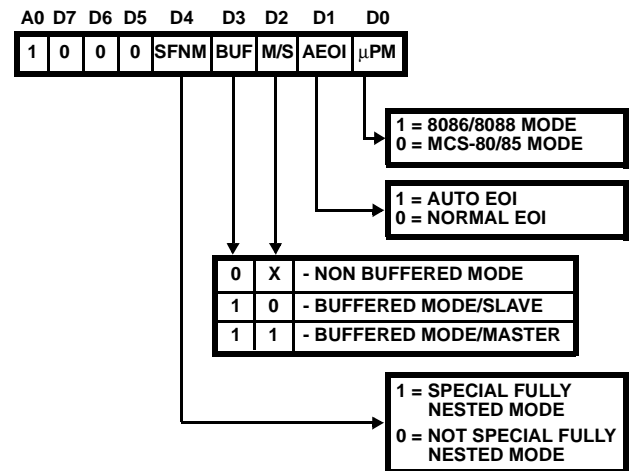


FIGURE 11. ICW4 FORMAT

**D7 thru D5:** These bits must be set to zero for proper operation.

**D4-SFNM:** This bit is used in the selection of the Special Fully Nested Mode (SFNM) of operation. This mode should only be used when multiple 82C59As are cascaded in a system. It needs only to be programmed in the Master 82C59A in the system.

0: Special Fully Nested Mode is not selected.

1: Special Fully Nested Mode is selected.

**D3 - BUF:** This bit tells the 82C59A whether or not the outputs from the data pins (D0 - D7) will be buffered. If they are buffered, this bit will cause the  $\overline{SP/EN}$  pin to become an output signal that can be used to control the "enable" pin on a buffering device(s).

0: The device will be used in a non-buffered mode.

Therefore, (1) the M/S bit in ICW4 is a don't care, and (2) the  $\overline{SP/EN}$  pin becomes an input pin telling the device if it is being used as a master (pin 16 = High) or a slave (pin 16 = low). For systems using a single 82C59A, the  $\overline{SP/EN}$  input should be tied high.

1: The device is used in buffered mode. An enable output signal will be generated on pin 16, and the M/S bit will be

used for determining whether the particular 82C59A is a “master” or a “slave”.

**D2 - M/S:** This bit is of significance only when the BUF bit is set (BUF =1). The purpose of this bit is to determine whether the particular 82C59A is being used as a “master” or a “slave” in the target system.

0: The 82C59A is being used as slave.

1: The 82C59A is the master interrupt controller in the system.

**D1 - AEOI:** This bit is used to tell the 82C59A to automatically perform a non-specific End-of-Interrupt on the trailing edge of the last Interrupt Acknowledge pulse. Users should note that when this is selected, the nested priority interrupt structure is lost.

0: Automatic End-of-Interrupt will not be generated.

1: Automatic End-of-Interrupt will be generated on the trailing edge of the last Interrupt Acknowledge pulse.

**D0 -  $\mu$ PM:** This bit tells the Interrupt Controller which microprocessor is being used in the system. An 8080/8085, or an 80C86/80C88.

0: The 82C59A will be used in an 8080/8085 based system.

1: 82C59A to be used in the 80C86/88 mode of operation.

## 3.0 Operation Command Words

Once the Initialization Command Words, described in the previous section, have been written to the 82C59A, the device is ready to accept interrupt requests. While the 82C59A is operating, we have the ability to select various options that will put the device in different operating modes, by writing Operation Command Words (OCWs) to the 82C59A. These OCWs can be sent at any time after the device has been initialized and in any order. These words can be changed at any time as well. Note: If A0 = 0 and D4 of the command word = 1, the 82C59A will begin the ICW initialization sequence.

There are three different OCWs for the 82C59A. Each has a different purpose. The first control word (OCW1) is used for masking out interrupt lines that are to be inactive or ignored during operation. OCW2 is used to select from various priority resolution algorithms in the device. Finally, OCW3 is used for (1) controlling the Special Mask Mode, and (2) telling the 82C59A which Register will be read on the next RD pulse; the ISR (In-service Register) or the IRR (Interrupt Request Register).

### 3.1 OCW1

This control word is used to set or clear the masking of the eight (8) interrupt lines input to the 82C59A. This control word performs this function via the Interrupt Mask Register (IMR). In its initial state, the value of this register is 00H. In other words, all of the interrupt lines are enabled. Therefore,

we need only write this control word when we wish to disable specific interrupt lines.

A direct mapping occurs between the bits in this control word and the actual interrupt pins on the device. For example bit 7 (D7) controls interrupt line IR7 (pin 25), bit 6 controls IR6, and so on.

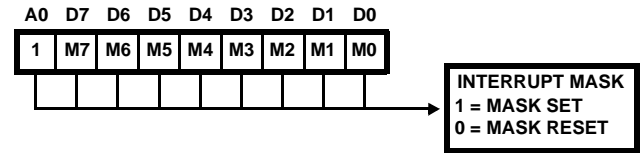


FIGURE 12. OCW1 FORMAT

Even though the user can mask off any of the IR lines, any interrupt occurring during that time will not be lost. The request for an interrupt is retained in the IRR; therefore when that IR is unmasked by issuing a new mask value to OCW1, the interrupt will be generated when it becomes the highest requesting priority.

### D7 THRU D0:

0: When any of the bits in the control word are reset (0), the corresponding interrupt is enabled.

1: By setting a bit(s) to a one in the control word, the corresponding interrupt line(s) is disabled.

For example, if the value 34H (00110100b) were written to OCW1, interrupts would be disabled from being serviced on lines IR2, IR4, and IR5.

### 3.2 OCW2

In ICW4 bit D1 was used to specify whether the 82C59A should wait for an EOI (End of Interrupt) from the CPU, or generate its own EOI (Automatic EOI). If bit D1 of ICW4 had been programmed to be zero, OCW2 would be used for sending the EOI to the 82C59A. Conversely, if this bit had been set to a one, OCW2 would be used for specifying whether or not the 82C59A should perform a priority rotation on the interrupts when the AEOI is detected.

OCW2 has several EOI options. The EOI issued can be either specific or non-specific. For each of these EOIs, the user can specify whether or not priority rotation should be performed.

D7	D6	D5	D4	D3	D2	D1	D0
R	SL	EOI	0	0	L2	L1	L0

FIGURE 13.

### R, SL, AND EOI:

These three bits are used for specifying how the device should handle AEOIs, or for issuing one of several different EOIs. They are programmed as shown in the following table.

TABLE 4. ROTATE AND EOI MODES

R	SL	EOI	
0	0	1	Non-specific EOI command
0	1	1	*Specific EOI command
1	0	1	Rotate on non-specific EOI command
1	0	0	Rotate in Automatic EOI mode (set)
0	0	0	Rotate in Automatic EOI mode (clear)
1	1	1	*Rotate on specific EOI command
1	1	0	*Set priority command
0	1	0	No operation

\* L0 - L2 are used.

### L2, L1, AND L0:

These three bits of the control word are used in conjunction with the issuance of specific EOIs or when specifically establishing a different priority structure. The bits tell the 82C59A which interrupt level is to be acted upon. Therefore, the software needs to know which interrupt is being serviced by the 82C59A

TABLE 5. INTERRUPT LEVEL TO ACT UPON

L2	L1	L0	
0	0	0	IR level 0
0	0	1	IR level 1
0	1	0	IR level 2
0	1	1	IR level 3
1	0	0	IR level 4
1	0	1	IR level 5
1	1	0	IR level 6
1	1	1	IR level 7

### 3.3 OCW3

There are two main functions that OCW3 controls: (1) Interrupt Status, and (2) Interrupt Masking. Interrupt status can be checked by looking at the ISR or IRR registers, or by issuing a Poll Command to manually identify the highest priority interrupt requesting service.

D7	D6	D5	D4	D3	D2	D1	D0
0	ESMM	SMM	0	1	P	RR	RIS

FIGURE 14.

**D7:** Must be set to zero for proper operation of the 82C59A.

**D6 - ESMM:** Enable Special Mask Mode - The ESMM bit when enabled allows the SMM bit to set or clear the Special Mask Mode. When disabled this bit causes the SMM bit to have no effect on the 82C59A.

0: Disables the effect of the SMM bit.

1: Enable the SMM bit to control the Special Mask Mode.

**D5 - SMM:** Special Mask Mode - The SMM bit is used to enable or disable the Special Mask Mode. This bit will only affect the 82C59A when the ESMM bit is set to 1.

0: Disable the Special Mask Mode.

1: Put the 82C59A into the Special Mask Mode.

**D4, D3:** These bits are used to differentiate between OCW2, OCW3, and ICW1. To properly select OCW3, D4 must be set to zero and D3 must be set to one.

**D2 - P:** Poll Command - This bit is used to issue the poll command to the 82C59A. The next read of the 82C59A will cause a poll word to be returned which tells if an interrupt is pending, and if so, which is the highest requesting level.

NOTE: The poll command must be issued each time the poll operation is desired.

0: No poll command issued to the 82C59A.

1: Issue the poll command.

**D1 - RR:** Read Register - This bit is used to execute the "read register" command. When this bit is set, the 82C59A will look at the RIS bit to determine whether the ISR or IRR register is to be read. When issuing this command, the next instruction executed by the CPU should be an input from this same port to get the contents of the specified register.

0: No "Read Register" command will be performed.

1: The next input instruction by the CPU will read either the contents of the ISR or the IRR as specified by the RIS bit.

**D0 - RIS:** This bit is used in conjunction with the RR bit to select which register is to be read when the "Read Register" command is issued.

0: The next input instruction will read the contents of the Interrupt Request Register (IRR).

1: The next input instruction will read the contents of the In-Service Register (ISR).

The two registers that can be accessed through the Read Register command are used to determine which interrupts are requesting service, and which one(s) are currently being serviced.

The IRR bits get set when corresponding Interrupt requests are received. For instance, when IR4 is detected, bit D4 of the IRR will get set. When an interrupt acknowledge comes back from the CPU, the priority resolution logic will determine which interrupt request will be serviced. The corresponding bit in the In-service Register (ISR) will then be set. Clearing of the correct bits in the ISR occurs through out use of the AEOL, or by issuing an EOI to the device.

## 4.0 Addressing the 82C59A

There are two factors that must be taken into account when addressing the 82C59A in a system. To begin with, the 82C59A is accessed only when the  $\overline{CS}$  pin (chip select) sees an active signal (low). This signal is generated using control circuitry in the system. Secondly, the various registers within the 82C59A are selected based upon the state of the A0 (address pin) as well as specific bits in the command words (i.e. for ICW1, OCW2, and OCW3 A0 must be a zero).

The circuit in Figure 15 shows that the  $\overline{CS}$  signal is generated using an HPL-82C338 Programmable Chip Select Decoder (PCSD). This device is being used as a 3-to-8 decoder. Note that the G1 input is active high and G2 thru G5 have been programmed to be active low. The A, B, and C inputs to the 82C338 correspond to address lines AD2, AD3, and AD4 respectively, from the 80C88. The A0 input to the 82C59A is also taken from the CPU's address bus: AD0 is

used. It should be noted that address line AD1 from the 80C88 is not being used in the addressing of this particular peripheral. This is done to allow other peripheral devices that require two address inputs for internal register selection, to use address lines AD0 and AD1 from the processor.

Because the AD1 address line from the 80C88 is not being used, the 82C59A will be addressed regardless of whether AD1 is high or low (1 or 0). The remainder of the address lines from the 80C88 can either be a zero or one when addressing the 82C59A. For the examples to be presented, it can be assumed that all unused address lines will be set to zero when addressing the 82C59A.

In Figure 15, output  $\overline{Y6}$  from the HPL-82C338 is being used as the  $\overline{CS}$  input to the 82C59A. This line is enabled when the inputs on A, B, and C are: A = 0, B = 1, and C = 1. Combining this with the A0 input to the 82C59A, we get the addresses 18H and 19H for accessing the 82C59A.

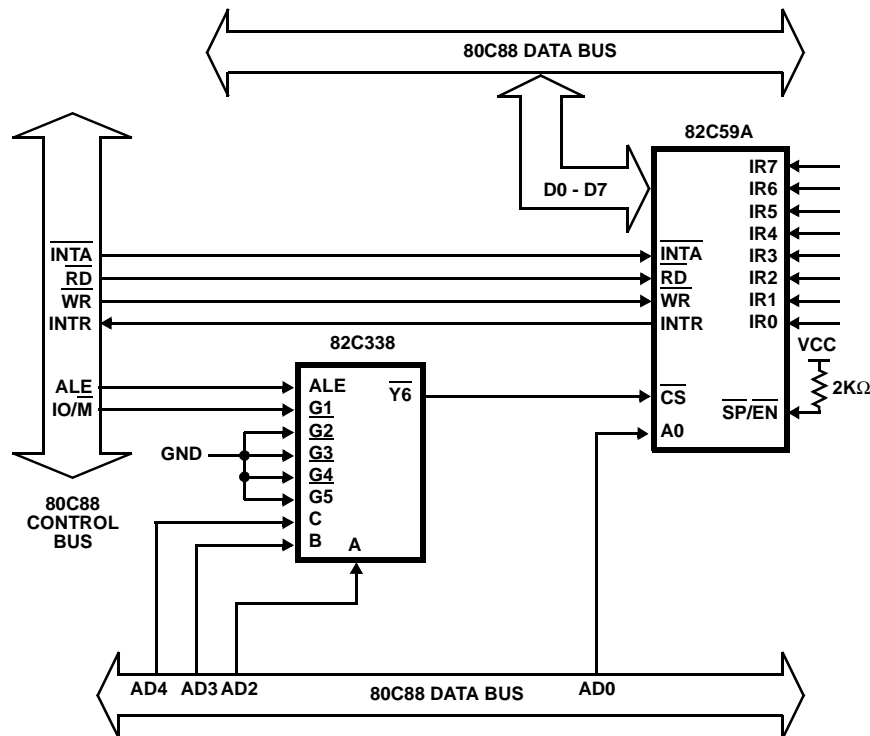


FIGURE 15. ADDRESSING THE 82C59A

## 5.0 Programming the 82C59A

As described earlier, there are two different types of command words that are used for controlling 82C59A operation; the Initialization Command Words (ICWs) and the Operation Command Words (OCWs). To properly program the 82C59A, it is essential that the ICWs be written first. When writing the ICWs to the 82C59A, they must be written in the following sequence:

1. Write ICW1 to the 82C59A, A0 = 0.
2. Write ICW2 to the 82C59A, A0 = 1.
3. If using cascaded 82C59As in system, write ICW3 to the 82C59A, A0 = 1.
4. If IC4 bit was set in ICW1, write ICW4 to the 82C59A.

NOTE: When using multiple 82C59As in the system (cascaded), each one must be initialized following the above sequence.

Once the 82C59A(s) has been configured through the ICWs, the OCWs can be used to select from the various operation mode options. These include: masking of interrupt lines,

selection of priority rotation, issuance of EOIs, reading of the ISR and/or IRR etc. These OCWs can be written to the 82C59A at any time during operation of the 82C59A. The various command words are identified by the state of selected bits in the words, rather than by the sequence that they are written to the 82C59A; as with the ICWs. Therefore, it is imperative that the fixed bit values in the command words be written as such to insure proper operation of the device(s).

### 5.1 Example 1: Single 82C59A

In Example 1, we are using a single 82C59A in a system to handle the interrupts caused by an 82C52 Programmable UART. The system is driven using a 80C86 microprocessor. The system configuration is shown in Figure 16.

Interrupts are initiated by the 82C52 anytime it receives data on its Serial Data In pin (SDI), or when it is ready to transmit more data via its Serial Data Out pin (SDO).

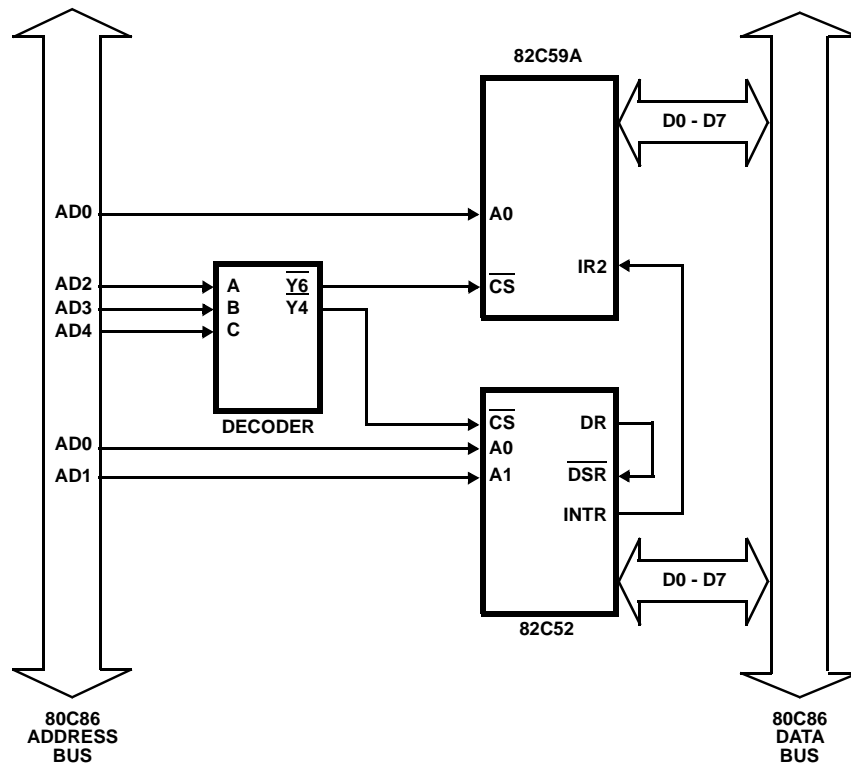


FIGURE 16. EXAMPLE 1: SINGLE 82C59A

## 5.2 Example 2: Cascaded 82C59As

Example 2 illustrates how we can use multiple 82C59As in Cascade Mode. Figure 17 shows the interconnections between the master and slave interrupt controllers. In this

example, only one interrupt can occur. This is generated by the 82C52 UART. Except for the fact that this system is configured with a Master-Slave interrupt scheme, it is the same as that in Example 1.

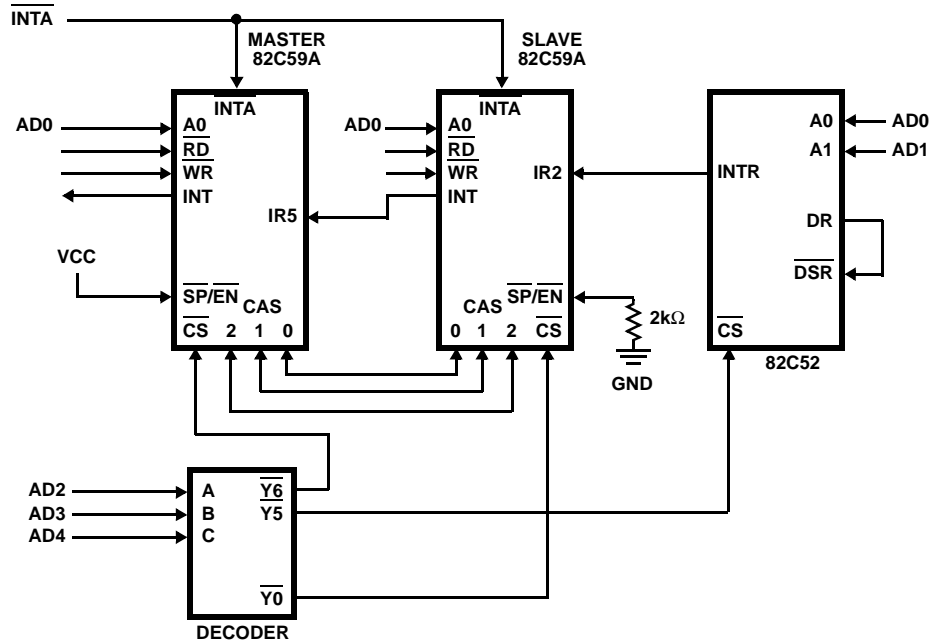


FIGURE 17. EXAMPLE 2: CASCADED 82C59As

## 6.0 Expansion Past 64 Interrupts

In some instances, it may be desirable to expand the number of available interrupts in a system past the maximum of 64 imposed when using cascaded 82C59As. The easiest way to accomplish this is through the use of the Poll command with the 82C59A. Figure 18 illustrates one example of how this expansion can be accomplished. Notice that we are using two 3 - to - 8 decoders to address up to 16 82C59As. Selection of which decoder is active takes place using the  $\overline{OE}$  pin driven by AD5 from the CPU's address bus.

With this type of interrupt structure, we are not using the INT and INTA lines from our processor (80C88 for this example). Because of this, no interrupts will break execution of the

system software. Therefore, it is the task of the software to poll the various 82C59As in the system to see if any interrupts are pending. Once it has been established which interrupt requires servicing, the software can take appropriate action.

There are disadvantages to using the poll mode for the systems interrupt structure: (1) the overhead of polling each of the 82C59As reduces the systems efficiency, and (2) realtime interrupt servicing cannot be guaranteed.

There are several advantages to using the poll mode in this manner: (1) there can be more than 64 priority interrupts in the system, and (2) memory in the system is freed because no interrupt vector table is required.

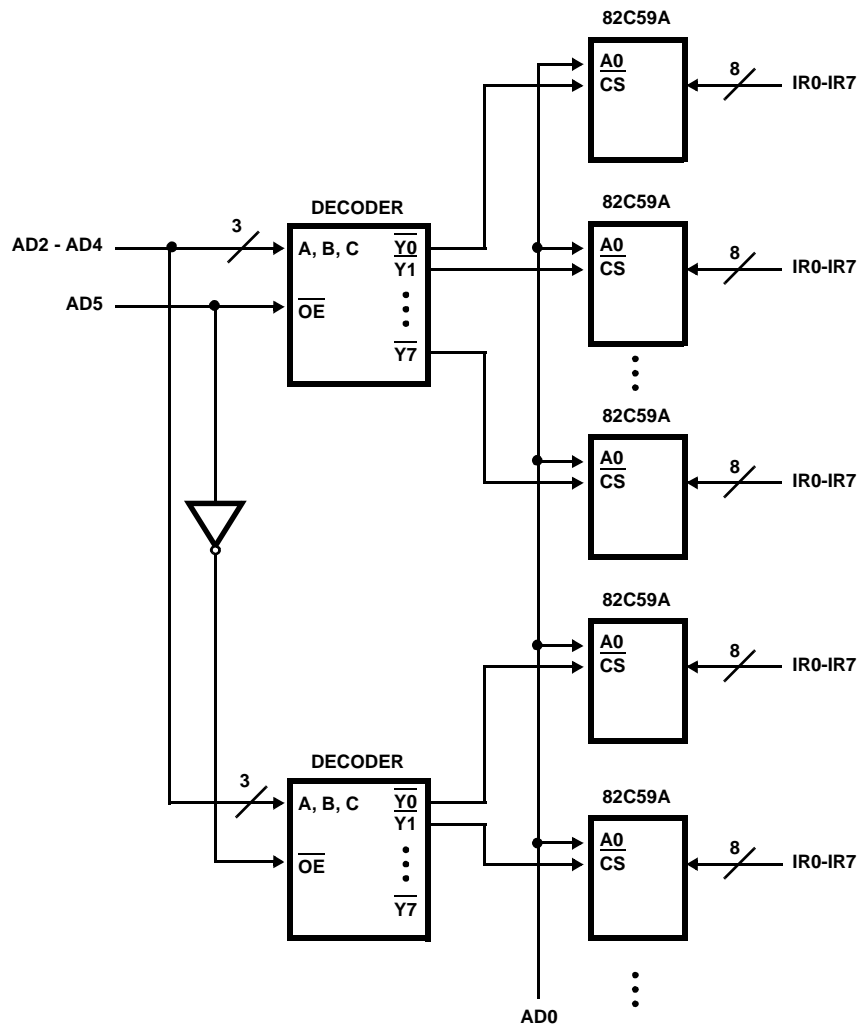


FIGURE 18. EXPANDING PAST 64 INTERRUPTS

All Intersil U.S. products are manufactured, assembled and tested utilizing ISO9000 quality systems.

Intersil Corporation's quality certifications can be viewed at [www.intersil.com/design/quality](http://www.intersil.com/design/quality)

*Intersil products are sold by description only. Intersil Corporation reserves the right to make changes in circuit design, software and/or specifications at any time without notice. Accordingly, the reader is cautioned to verify that data sheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.*

For information regarding Intersil Corporation and its products, see [www.intersil.com](http://www.intersil.com)