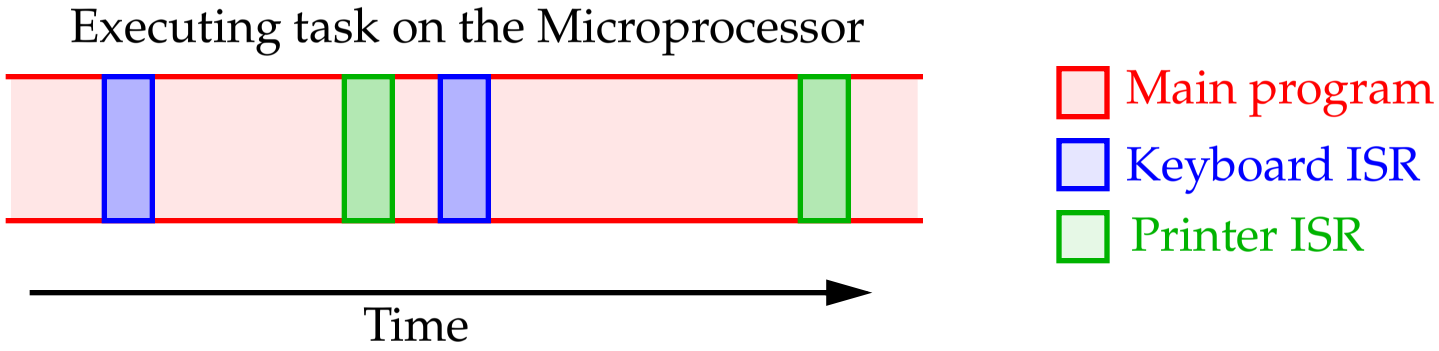
Interrupts

Interrupt processing is an alternative to polling.



The Intel microprocessors support hardware interrupts through:

- Two pins that allow interrupt requests, INTR and NMI
- One pin that acknowledges, $\overline{\text{INTA}}$, the interrupt requested on INTR.

And software interrupts through instructions:

- INT, INTO, INT 3, BOUND

Control is provided through

- IF and TF flag bits
- IRET and IRETD

Interrupt Vector Table

INT and INT3 behave in a similar way.

INT n:

Calls ISR located at vector n (n*4).

The INT instruction requires two bytes of memory, opcode plus n.

BOUND and INTO are both conditional.

BOUND:

```
BOUND AX, DATA      ;Compares AX with DATA
```

AX is compared with DATA and DATA+1, if less than an interrupt occurs.

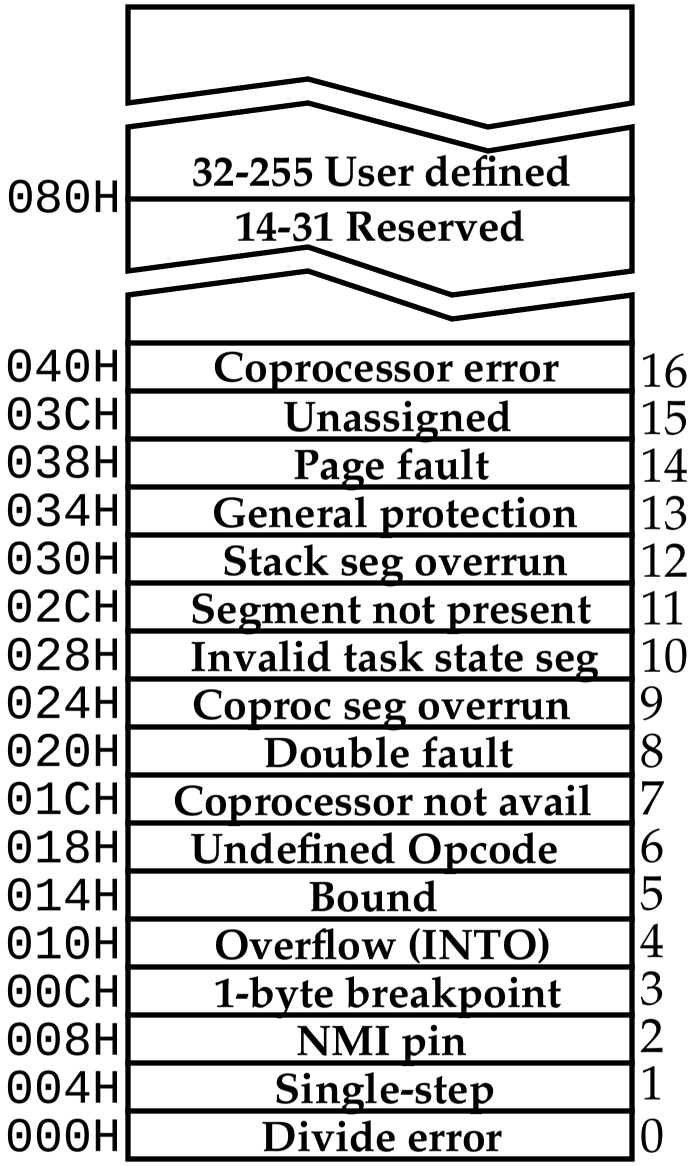
AX is compared with DATA+2 and DATA+3, if greater than an interrupt occurs.

INTO:

Checks the overflow flag (OF). If OF=1, the ISR is called.

IRET removes 6 bytes from the stack, 2 for IP, 2 for CS and 2 for FLAGS.

Interrupt Vector Table



The interrupt vector table is located in the first 1024 bytes of memory at addresses 000000H through 0003FFH.

There are 256 4-byte entries (segment and offset in real mode).

Seg high	Seg low	Offset high	Offset low
Byte 3	Byte 2	Byte 1	Byte 0

Real Mode Interrupts

After the execution of each instruction, the microprocessor determines whether an interrupt is active by checking, in order:

- Other instruction executions
- Single-step
- NMI
- Coprocessor segment overrun
- INTR
- INT

If one or more of these conditions are present, then:

- FLAGS is pushed onto the stack
- Both the interrupt (IF) and trap (TF) flags are cleared, which disables the INTR pin and the trap or single-step feature.
- The CS and IP are pushed onto the stack.
- The interrupt vector contents are fetched and loaded into CS and IP and execution resumes in the ISR.
- On IRET, CS, IP and FLAGS are popped.
  IF and TF are set to the state prior to the interrupt.

Real and Protected Mode Interrupts

The return address (CS/IP) is pushed onto the stack during the interrupt.

The return address can point to:

- The next instruction.
- The offending (current) instruction.

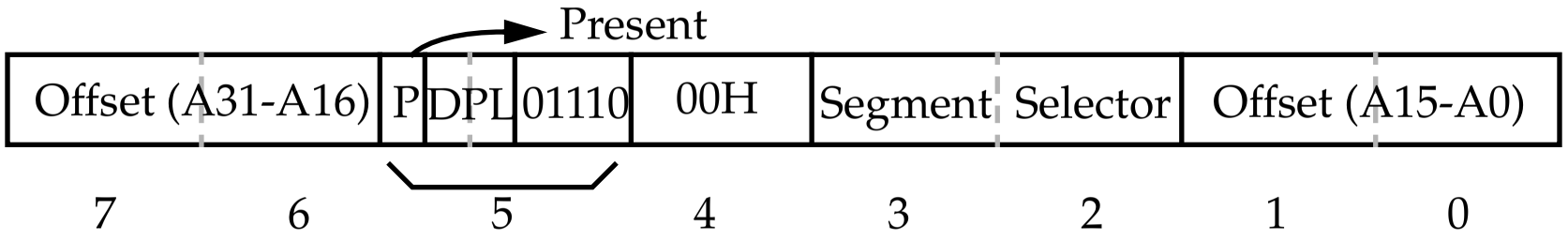
The latter case occurs for interrupts 0, 5, 6, 7, 8, 10, 11, 12 and 13.

This makes it possible to try the instruction again.

Protected Mode:

The same interrupt assignments are made and the same sequence of operations occurs in protected mode but the interrupt table is different.

Instead, 256 interrupt descriptors are used in the interrupt descriptor table (IDT).



Hardware Interrupts

The INTR pin must be externally decoded to select a vector.

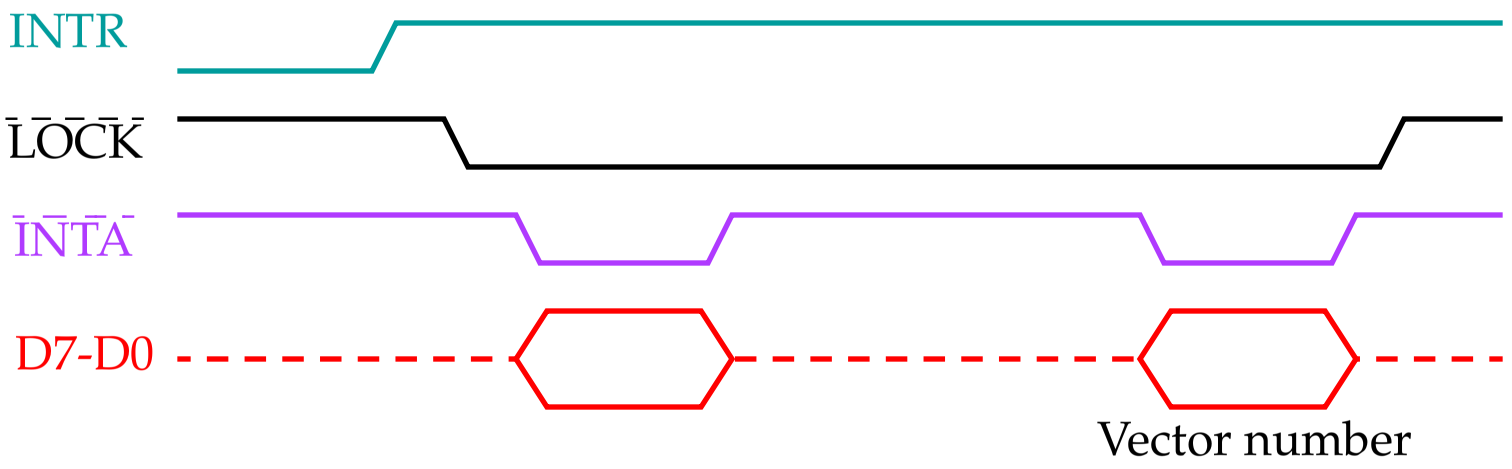
Any vector is possible, but the interrupt vectors between 20H and FFH are usually used (Intel reserves vectors between 00H and 1FH).

$\overline{\text{INTA}}$ is an output of the microprocessor to signal the external decoder to place the interrupt number on data bus connections D7-D0.

The INTR pin is set by an external device (8259A) and cleared in the ISR.

The input is automatically disabled by the microprocessor once it is recognized and re-enabled by IRET or IRETD instruction.

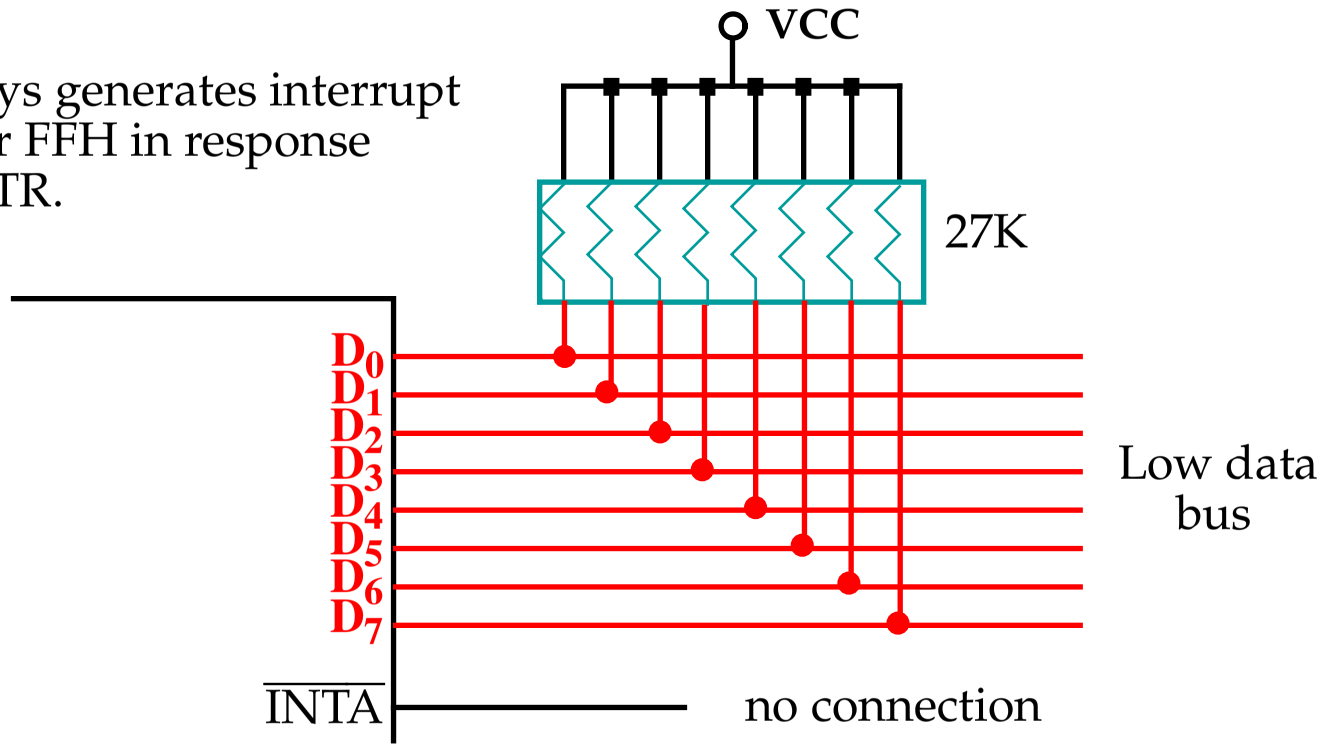
Timing diagram of the handshake.



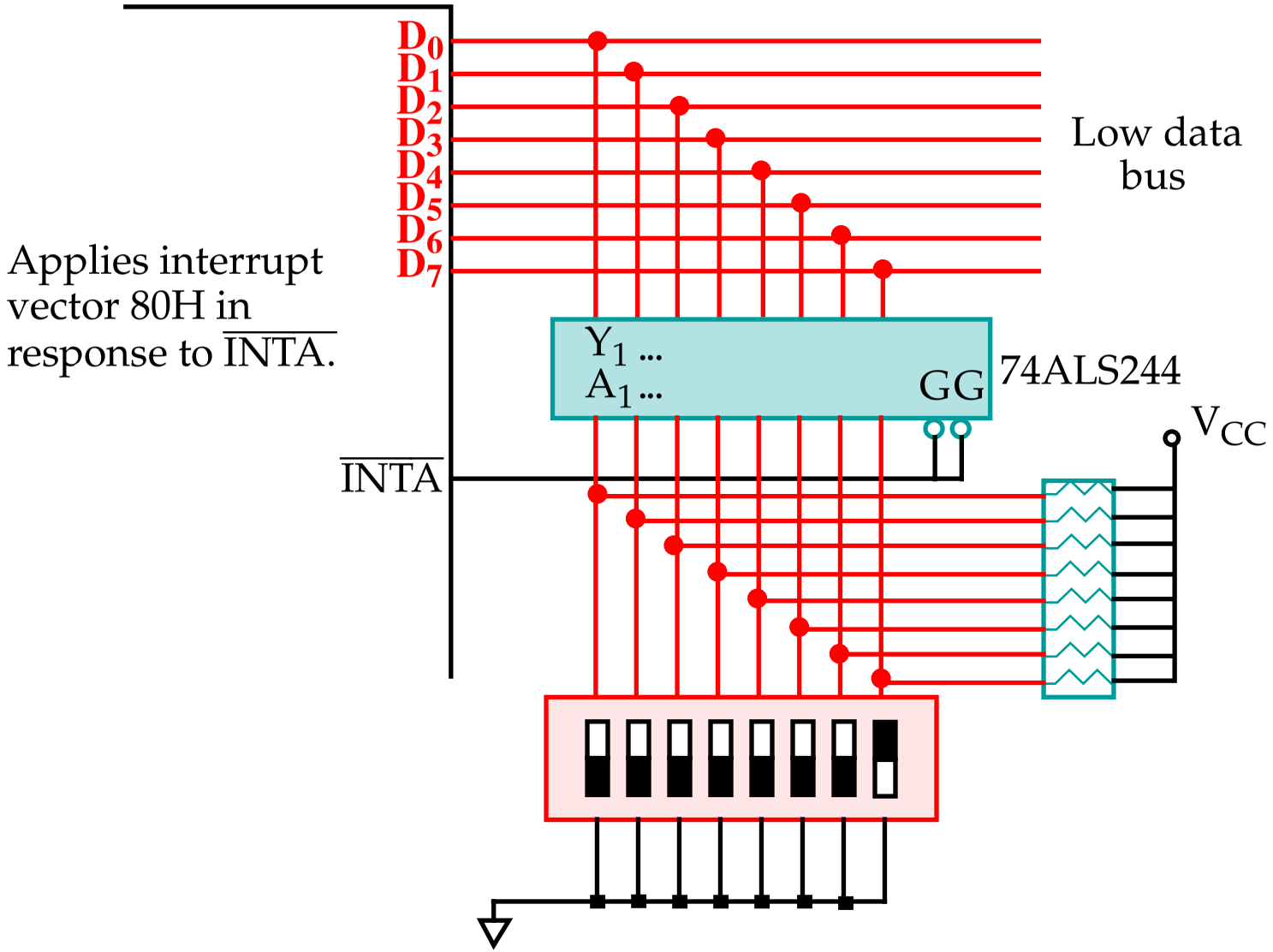
Hardware Interrupts

Simpliest method of generating an interrupt vector:

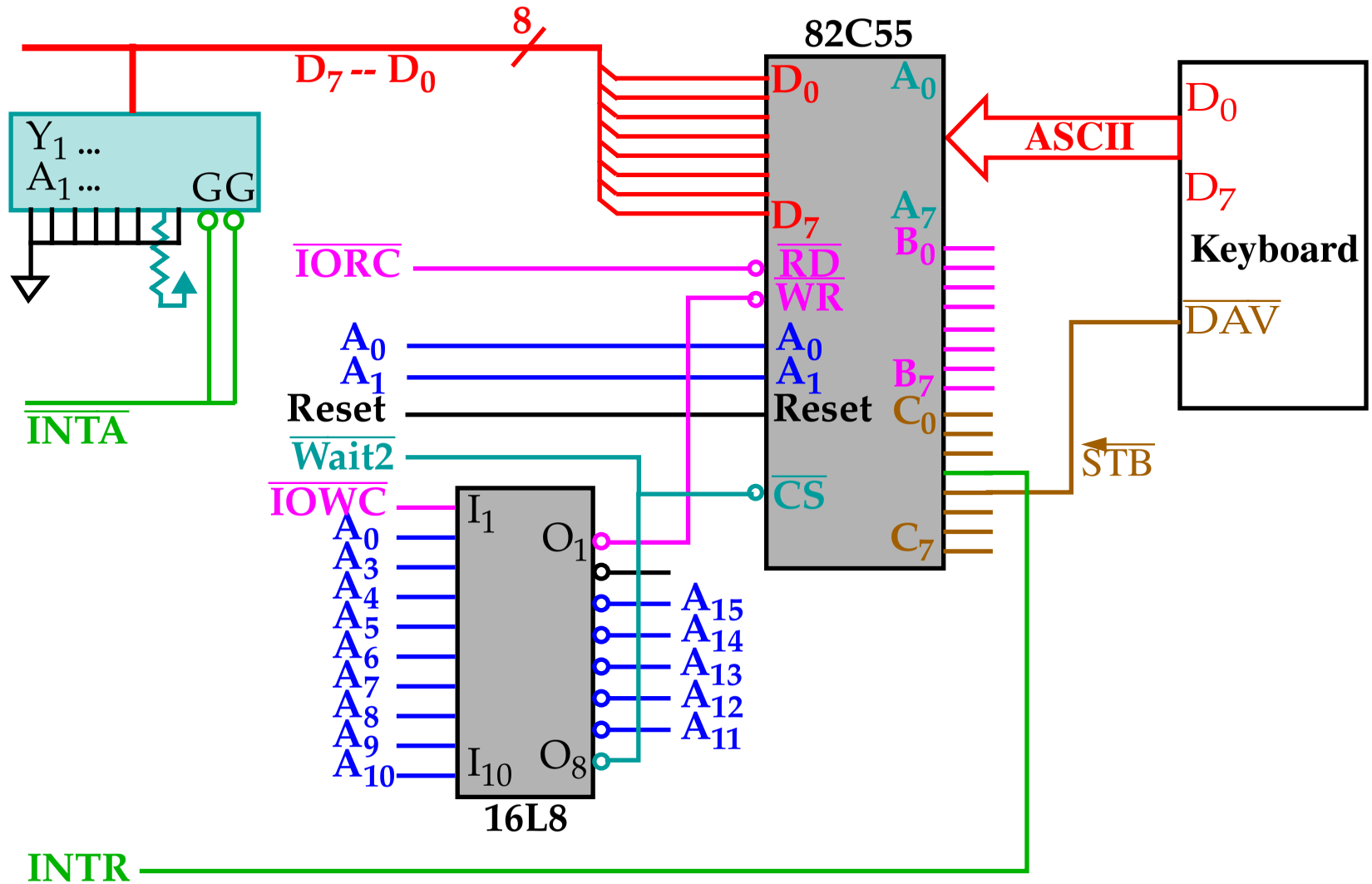
Always generates interrupt vector FFH in response to INTR.



Tri-state Buffer for Generating the Interrupt Vector

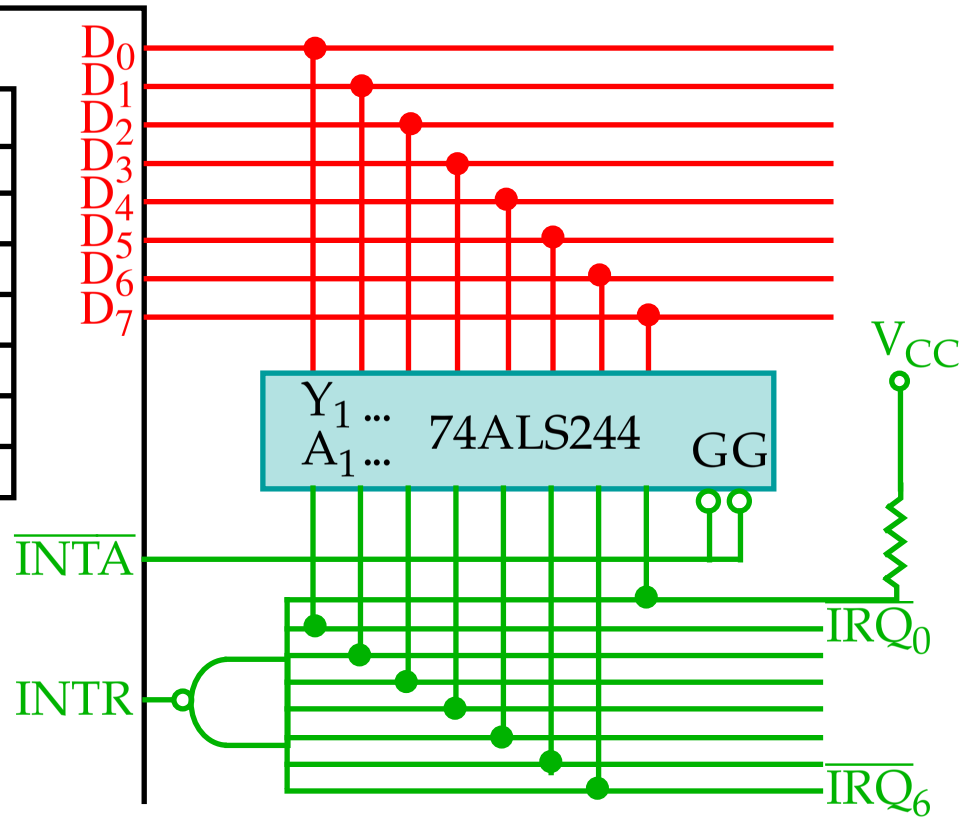


An Example 82C55 Interrupt Configuration



Handling more than 1 IRQ

IRQs							
6	5	4	3	2	1	0	Vect
1	1	1	1	1	1	0	FEH
1	1	1	1	1	0	1	FDH
1	1	1	1	0	1	1	FBH
1	1	1	0	1	1	1	F7H
1	1	0	1	1	1	1	EFH
1	0	1	1	1	1	1	DFH
0	1	1	1	1	1	1	BFH

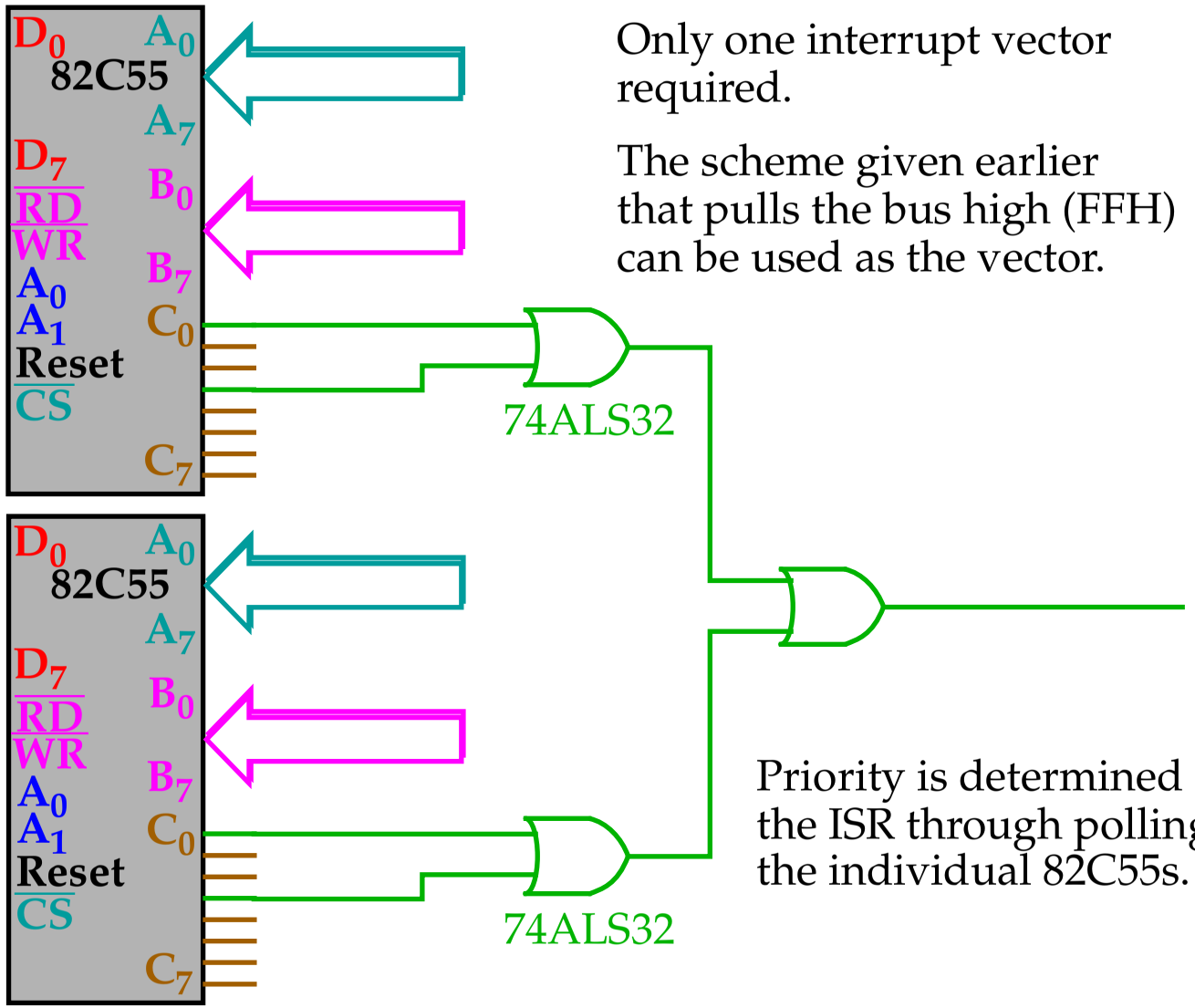


If any of $\overline{IRQ}_x$ goes low, the NAND goes low requesting an interrupt.

Note that if more than one IRQ goes low, a unique interrupt vector is generated and an interrupt priority needs to be defined.

The Interrupt Vector table must be expanded to accommodate this.

Daisy-Chained Mechanism for Multiple IRQs

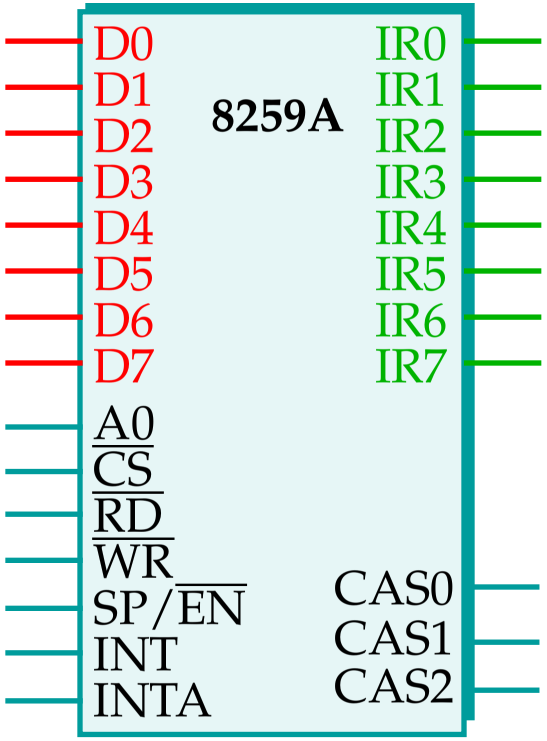


8259A Programmable Interrupt Controller

The 8259A adds 8 vectored priority encoded interrupts to the microprocessor.

It can be expanded to 64 interrupt requests by using one master 8259A and 8 slave units.

Bidirectional. Connect to any 8-bit bank.



Connect either to devices or to upstream slave 8259As

$\overline{CS}$ and $\overline{WR}$ must be decoded. Other connections are direct to micro.

8259A Programmable Interrupt Controller

The meaning of the other connections:

- $\overline{WR}$

  Connects to a write strobe signal (one of 8 for the Pentium).

- $\overline{RD}$

  Connects to the $\overline{IORC}$ signal.

- INT

  Connects to the INTR pin on the microprocessor.

- $\overline{INTA}$

  Connects to the $\overline{INTA}$ pin on the microprocessor.

- A0

  Selects different command words in the 8259A.

- $\overline{CS}$

  Chip select - enables the 8259A for programming and control.

- SP/$\overline{EN}$

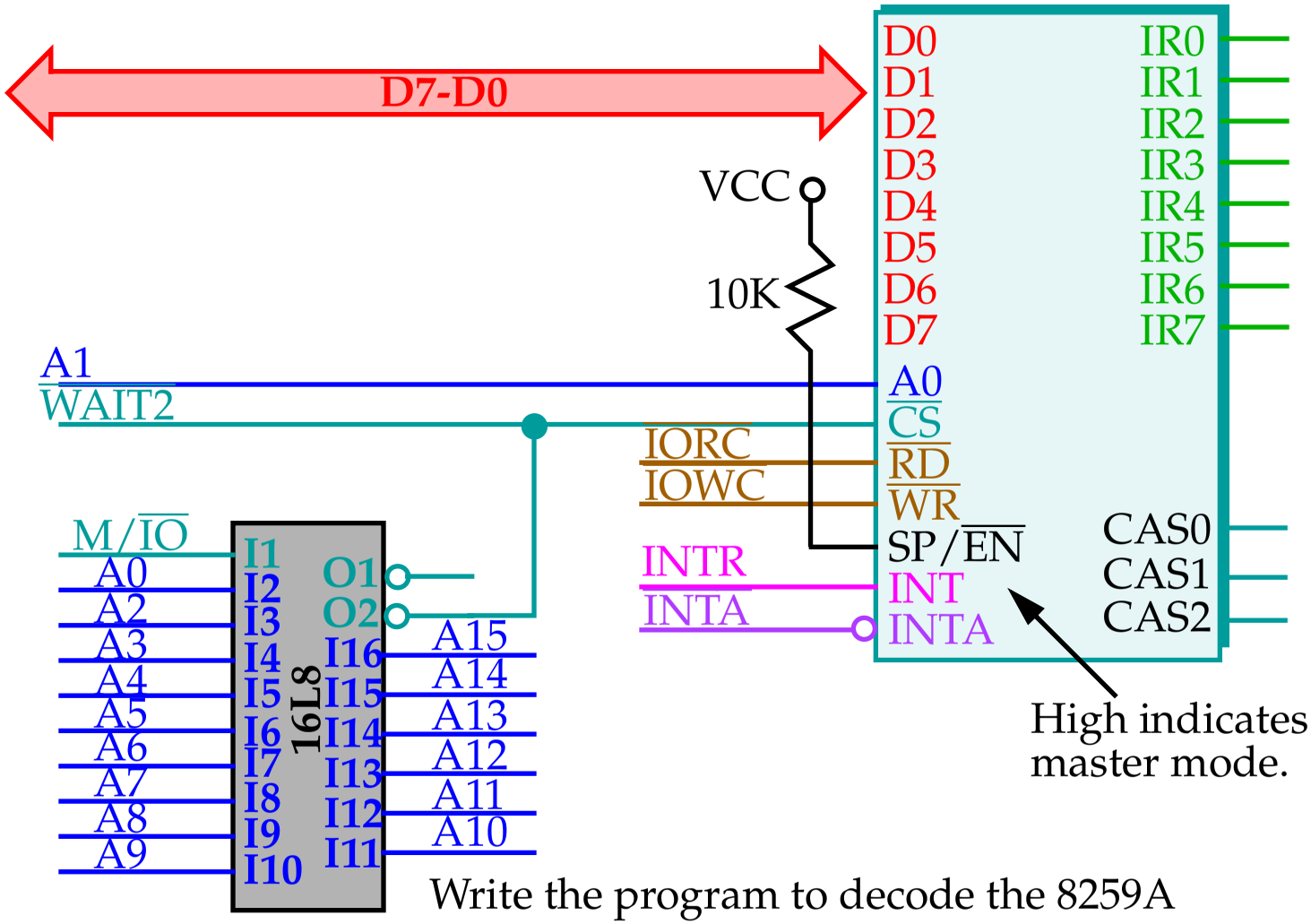
  Slave Program (1 for master, 0 for slave)/Enable Buffer (controls the data bus transievers when in buffered mode).

- CAS2-CAS0

  Used as outputs from the master to the slaves in cascaded systems.

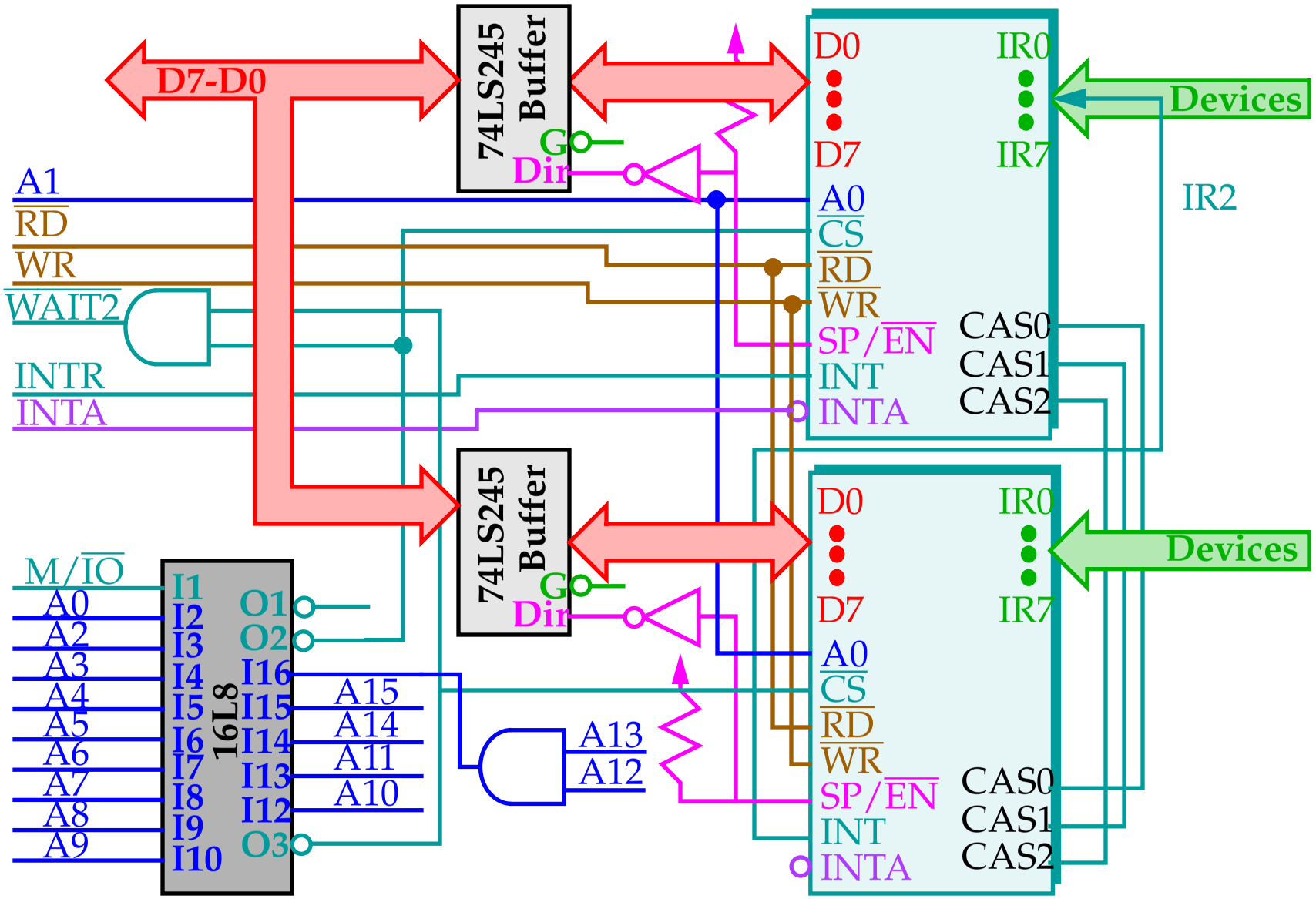
8259A Programmable Interrupt Controller

A single 8259A connected in the 8086.



Write the program to decode the 8259A at ports 0400H and 0402H.

8259A Programmable Interrupt Controller



Programming the 8259A

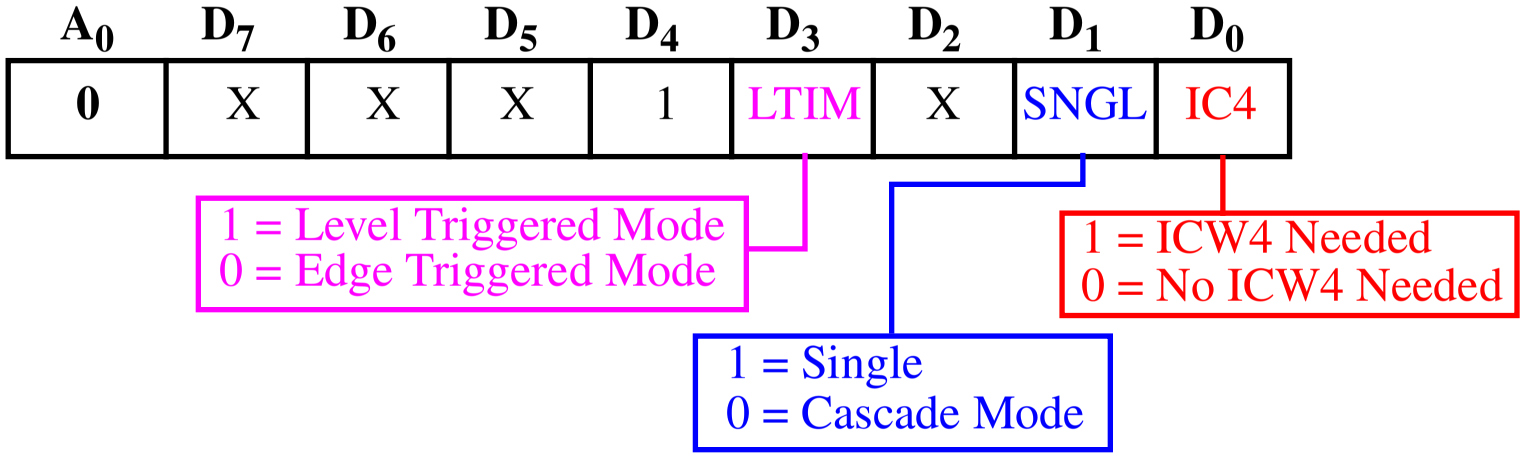
Programmed by Initialization (ICWs) and Operation (OCWs) Command Words.

There are 4 ICWs.

At power-up, ICW1, ICW2 and ICW4 must be sent.

If ICW1 indicates cascade mode, then ICW3 must also be sent.

- ICW1:

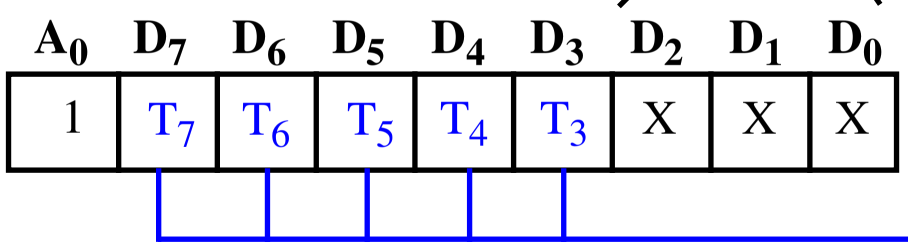


LTIM indicates if IRQ lines are positive edge-triggered or level-triggered.

Programming the 8259A

- ICW2:

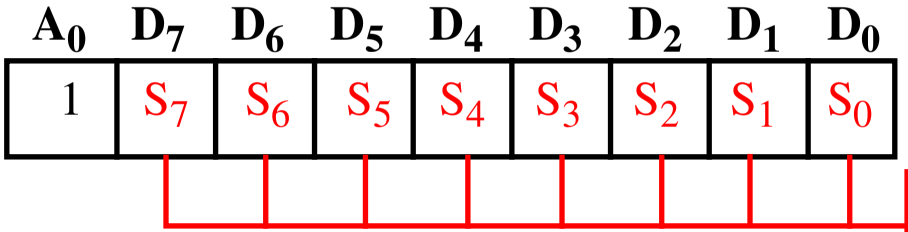
Low order bits are 0 since there are 8 interrupts.



T7-T3 of Interrupt Vector Address (8086/8088 Mode)

These bits determine the vector numbers used with the IRQ inputs. For example, if programmed to generate vectors 08H-0FH, 08H is placed into these bit positions.

- ICW3:

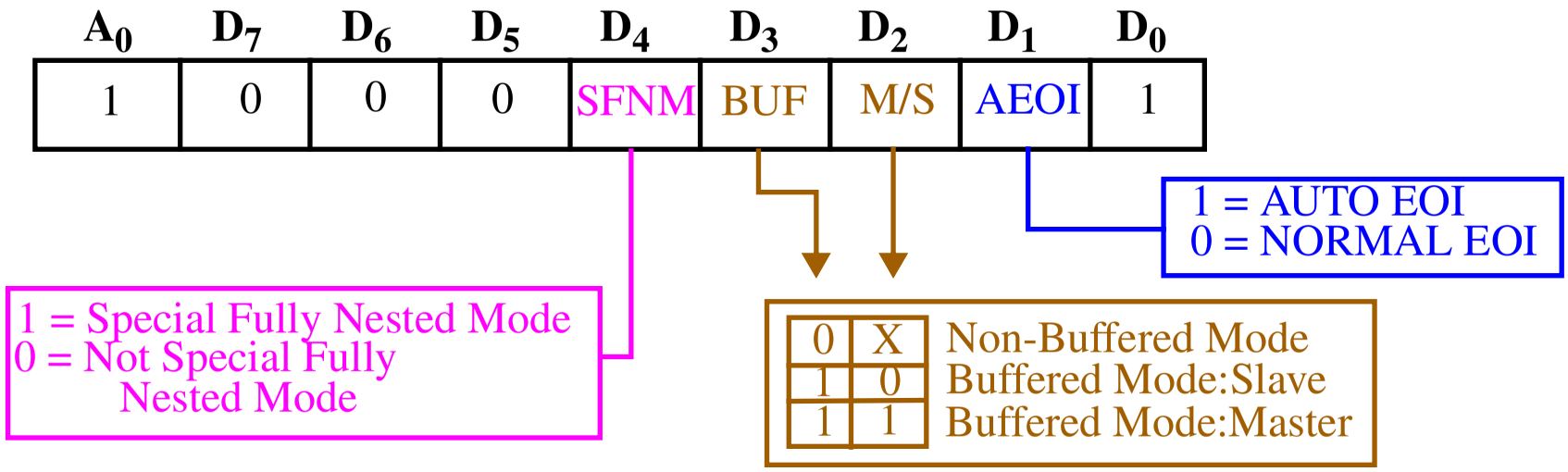


This register is treated as a mask, with 1's indicating the IRQ channels connected to master/slave 8259As.

0 = IR Input has a slave
1 = IR Input does not have a slave

Programming the 8259A

- ICW4:



Fully nested mode allows the highest-priority interrupt request from a slave to be recognized by the master while it is processing another interrupt from a slave.

AEOI, if 1, indicates that an interrupt automatically resets the interrupt request bit, otherwise OCW2 is consulted for EOI processing.

Programming the 8259A

The Operation Command Words (OCWs) are used to direct the operation of the 8259A.

- OCW1:

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0

INTERRUPT MASK
0 = Mask Reset
1 = Mask Set

OCW1 is used to read or set the interrupt mask register.

If a bit is set, it will turn off (mask) the corresponding interrupt input.

- OCW2:

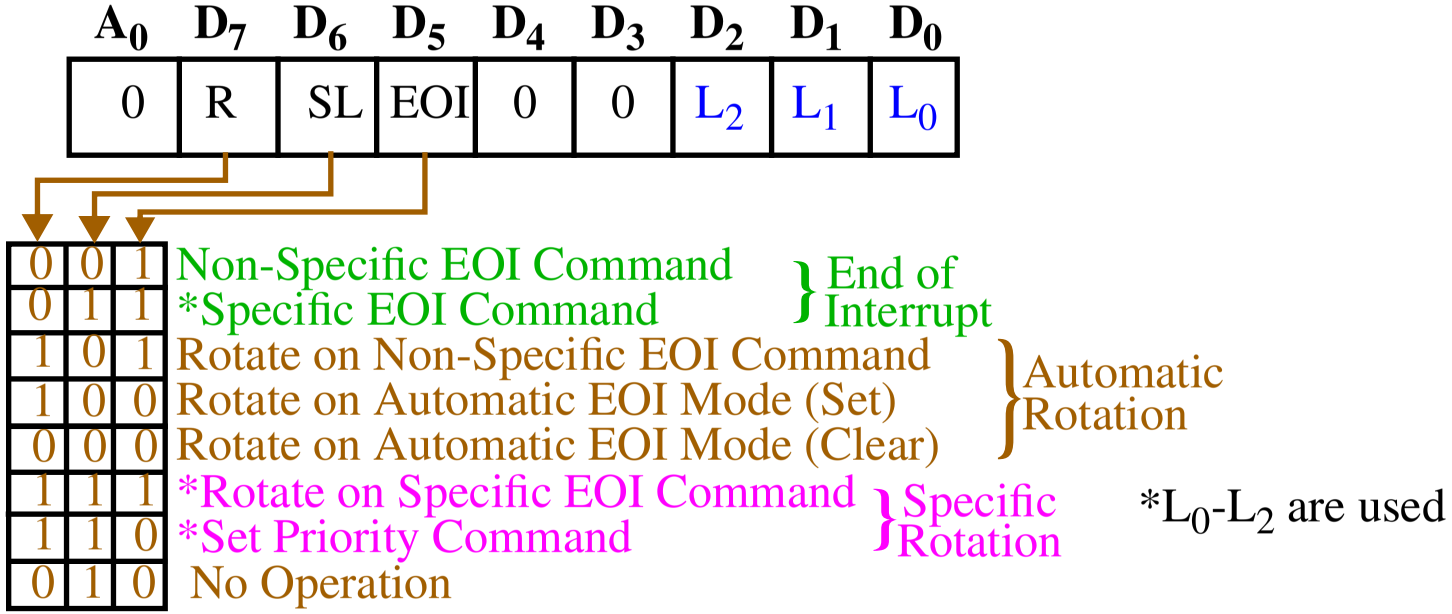
A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	R	SL	EOI	0	0	L_2	L_1	L_0

Only programmed when the AEOI mode in ICW4 is 0.

Allows you to control priorities after each interrupt is processed.

Programming the 8259A

- OCW2:



Non-specific EOI: Here, the ISR sets this bit to indicate EOI. The 8259A automatically determines which interrupt was active and re-enables it and lower priority interrupts.

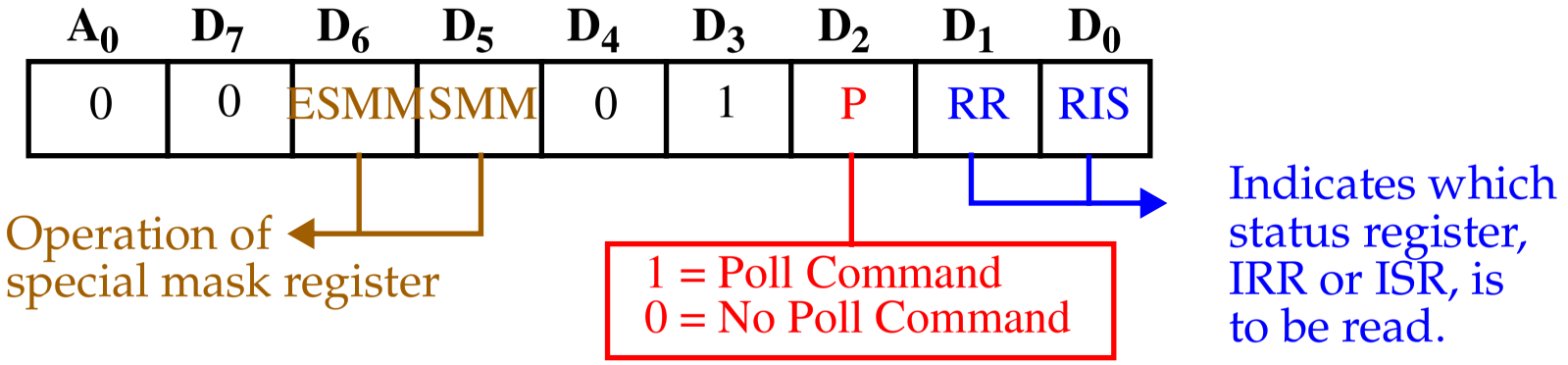
Specific EOI: ISR resets a specific interrupt request given by L_2-L_0.

Rotate commands cause priority to be rotated w.r.t. the current one being processed.

Set priority: allows the setting of the lowest priority interrupt (L_2-L_0).

Programming the 8259A

- OCW3:



If polling is set, the next read operation will read the poll word.

If the leftmost bit is set in the poll word, the rightmost 3 bits indicate the active interrupt request with highest priority.

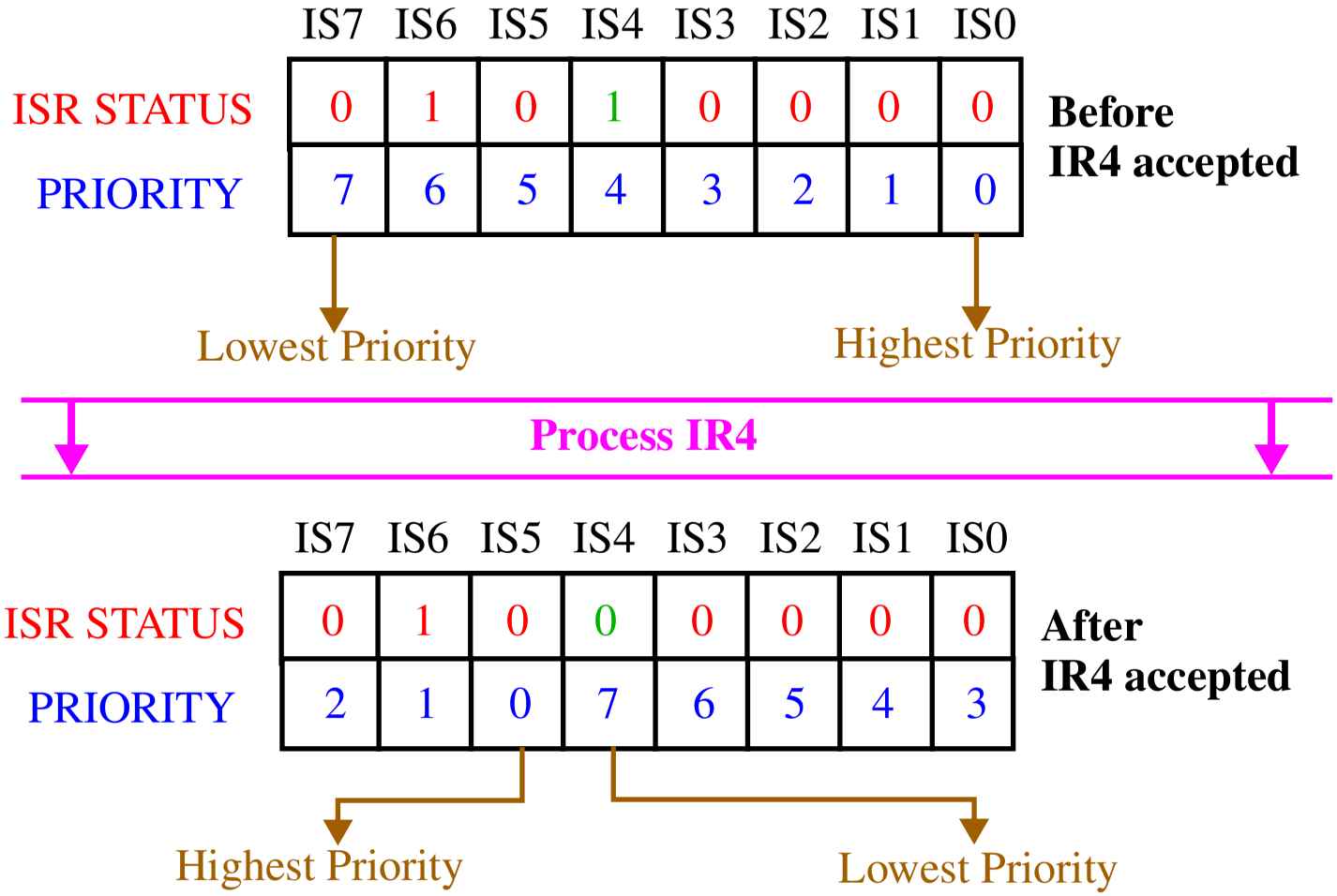
Allows ISR to service highest priority interrupt.

There are three status registers, Interrupt Request Register (IRR), In-Service Register (ISR) and Interrupt Mask Register (IMR).

- IRR: Indicates which interrupt request lines are active.
- ISR: Level of the interrupt being serviced.
- IMR: A mask that indicates which interrupts are on/off.

Programming the 8259A

ISR update procedure with rotating priority configured.



Interfacing 16550 UART using 8259A

In the following configuration the 16550 is connected to the 8259A through IR0.

An interrupt is generated, if enabled through the interrupt control register, when either:

- The transmitter is ready to send another character.
- The receiver has received a character.
- An error is detected while receiving data.
- A modem interrupt occurs.

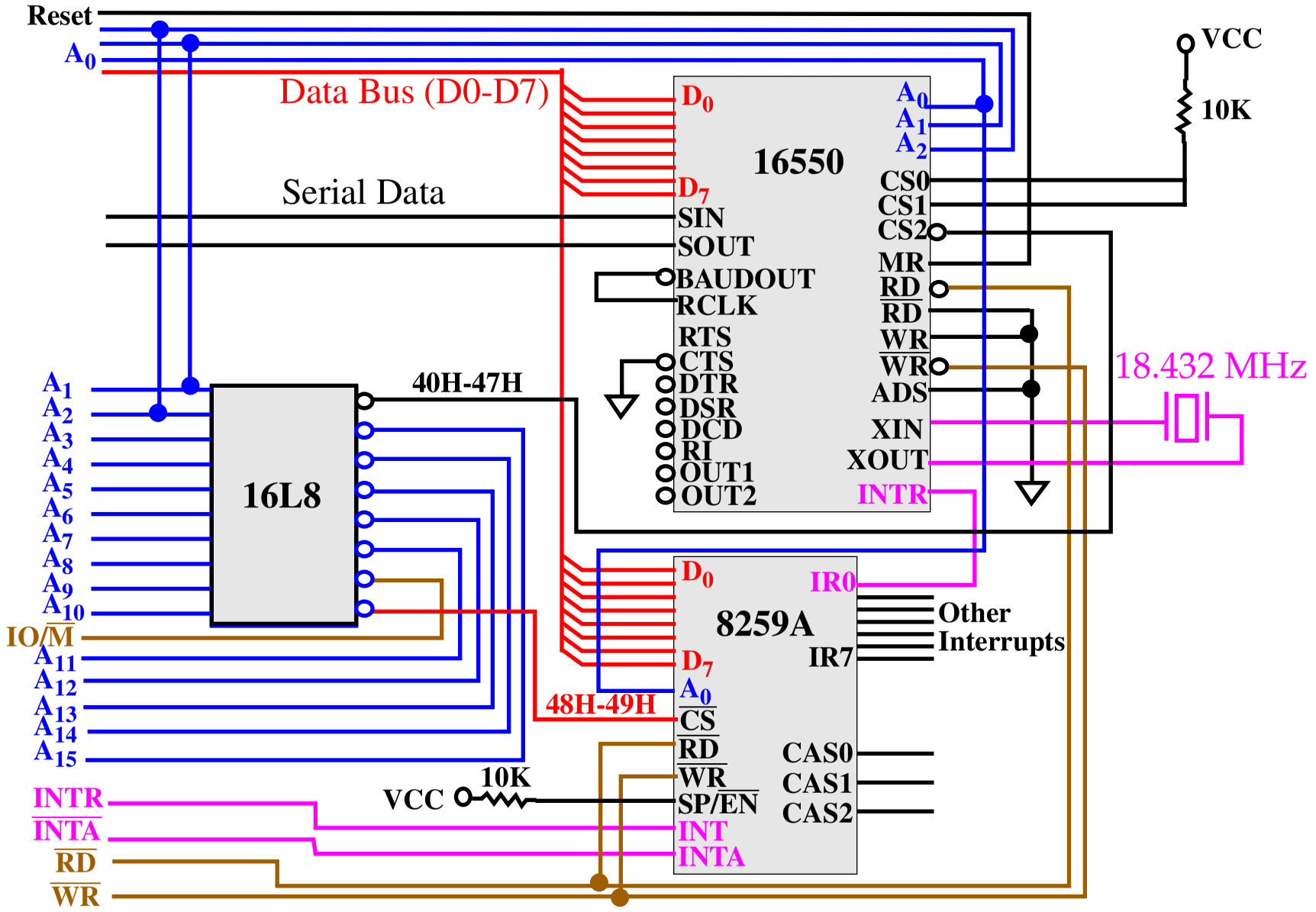
The16550 is decoded at 40H and 47H.
The 8259A is decoded at 48H and 49H.

Program in text shows the steps involved in programming both devices.

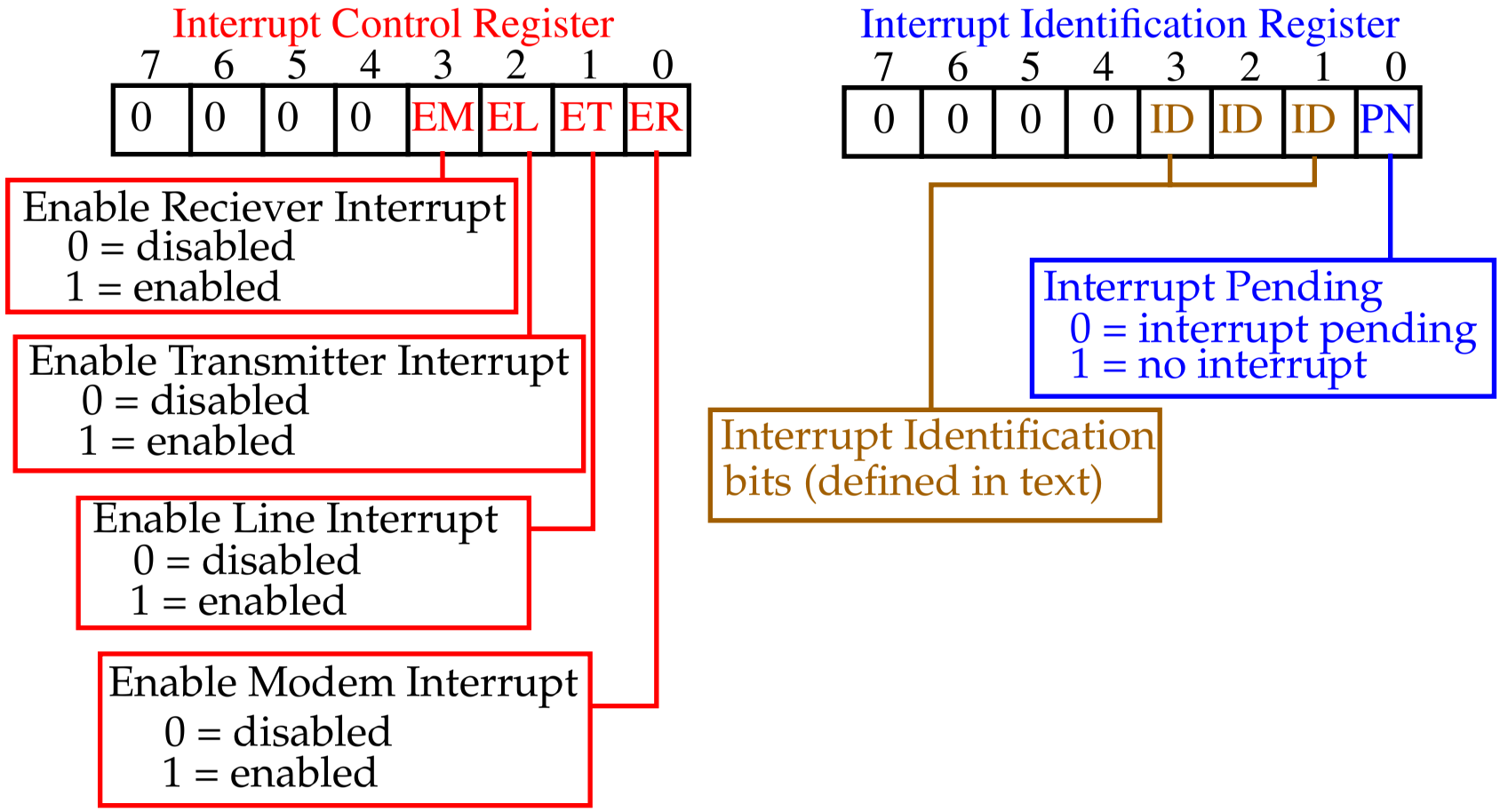
Since the 16550 generates only one interrupt request for each of the above interrupts, the 16550 must be polled.

Remember the interrupt identification register of the 16550?

Interfacing 16550 UART using 8259A



16550 UART Interrupts



Text gives ISR programming examples that show initialization and operation.