CMPE - 310

Lecture 13 – Interrupts and PIC (8259)

Outline

Interrupts – Concept, Implementation, Types
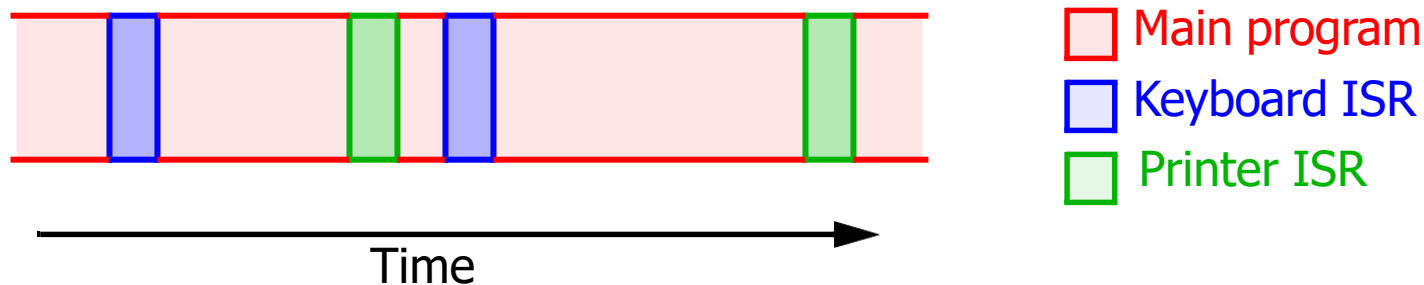
Programmable Interrupt Controller :8259

- Pinout
- Operation
- Interfacing
- Programing

Interrupts

Interrupt processing is an alternative to polling.

Executing task on the Microprocessor

□ Main program
□ Keyboard ISR
□ Printer ISR

The Intel microprocessors support hardware interrupts through:
o Two pins that allow interrupt requests, INTR and NMI
o One pin that acknowledges, $\overline{\text{INTA}}$, the interrupt requested on INTR.

And software interrupts through instructions:
o INT, INTO, INT 3, BOUND

Control is provided through
o IF and TF flag bits
o 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

Address	Vector	#
	32-255 User defined	
080H	14-31 Reserved	
040H	Coprocessor error	16
03CH	Unassigned	15
038H	Page fault	14
034H	General protection	13
030H	Stack seg overrun	12
02CH	Segment not present	11
028H	Invalid task state seg	10
024H	Coproc seg overrun	9
020H	Double fault	8
01CH	Coprocessor not avail	7
018H	Undefined Opcode	6
014H	Bound	5
010H	Overflow (INTO)	4
00CH	1-byte breakpoint	3
008H	NMI pin	2
004H	Single-step	1
000H	Divide error	0

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:

O Other instruction executions

O Single-step

O NMI

O Coprocessor segment overrun

O INTR

O INT

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

O FLAGS is pushed onto the stack

O Both the interrupt (IF) and trap (TF) flags are cleared, which disables the INTR pin and
  the trap or single-step feature.

O The CS and IP are pushed onto the stack.

O The interrupt vector contents are fetched and loaded into CS and IP and execution
  resumes in the ISR.

O 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:

O The next instruction.

O 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. More on this later in the course.

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

Present

Offset (A31-A16)	P	DPL	01110	00H	Segment Selector	Offset (A15-A0)
7	6		5	4	3 2	1 0
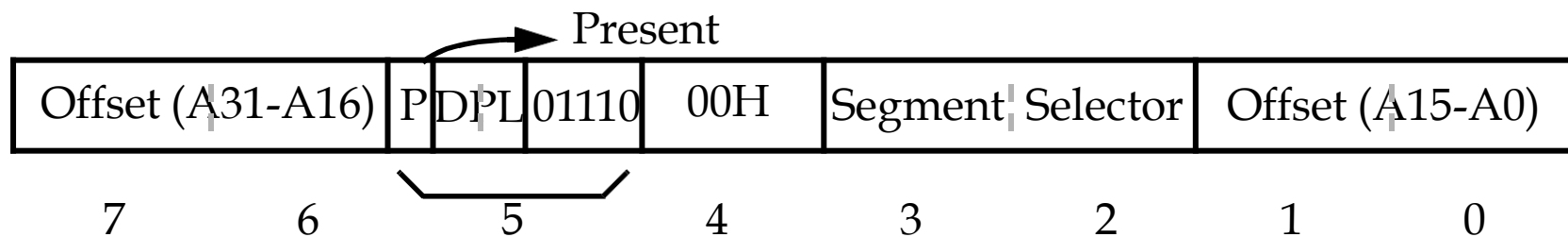
Hardware Interrupts

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.

INTR

$\overline{\text{LOCK}}$

$\overline{\text{INTA}}$

D7-D0

Vector number
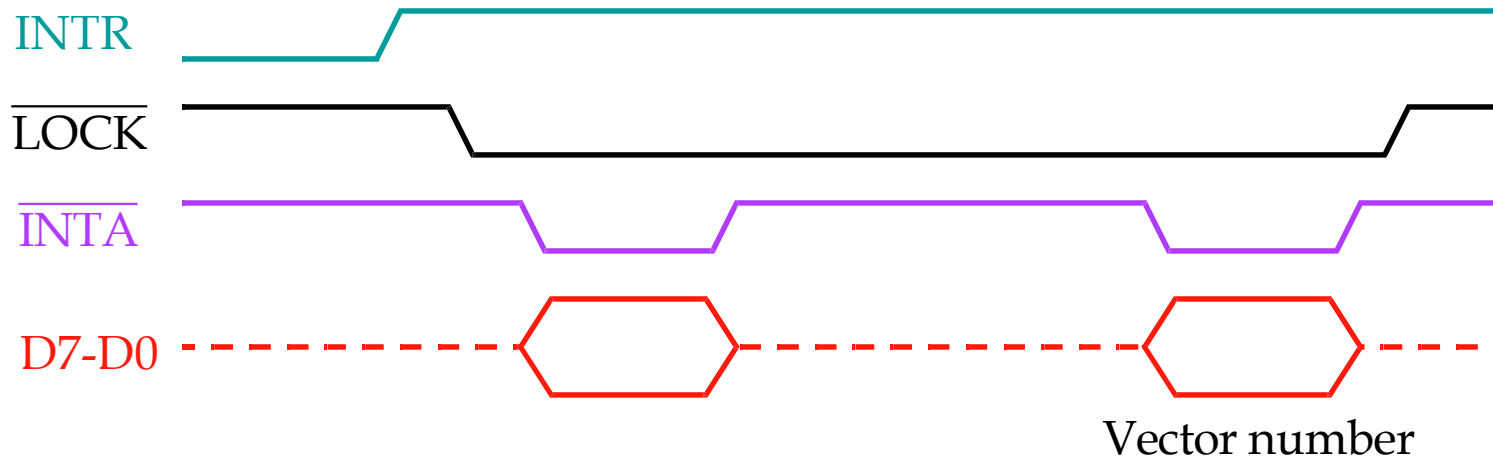
Hardware Interrupts

Simpliest method of generating an interrupt vector:

Always generates interrupt vector FFH in response to INTR.

VCC

27K

D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7

Low data bus

$\overline{INTA}$

no connection
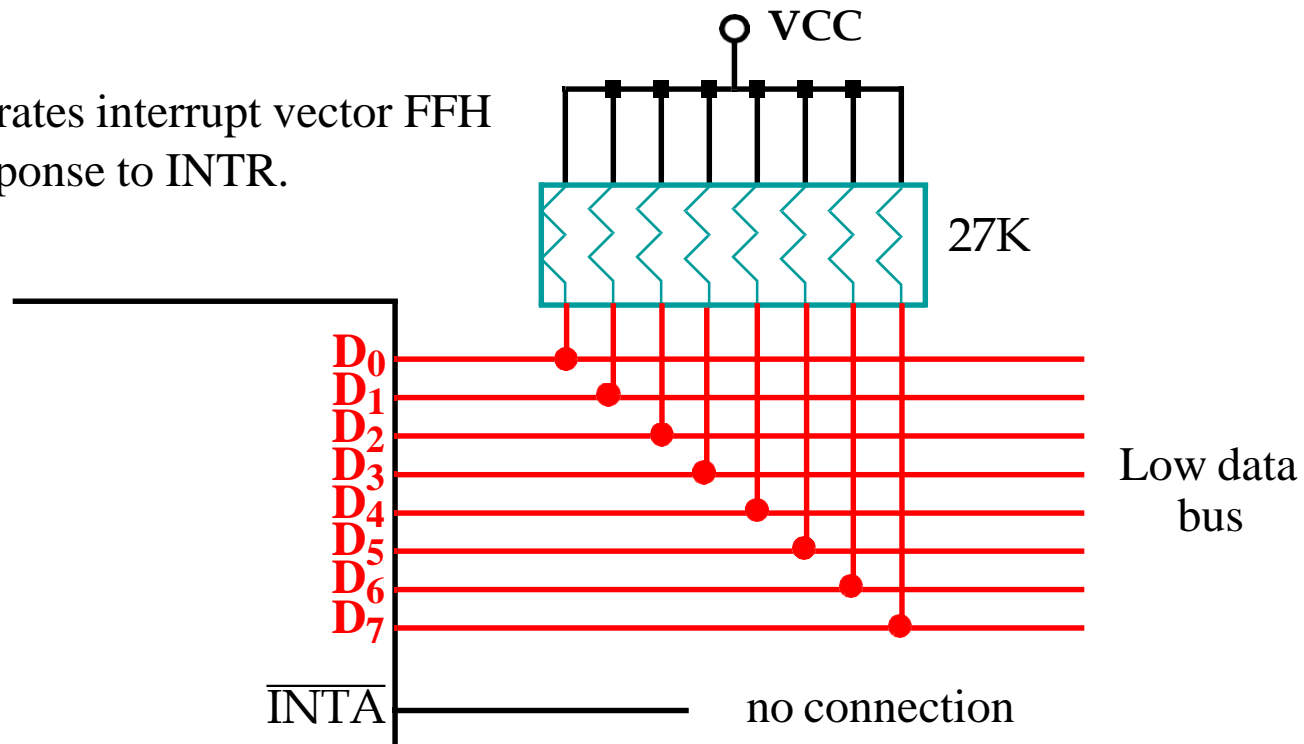
Tri-state Buffer for Generating the Interrupt Vector

D_0
D_1
D_2
D_3
D_4
D_5
D_6
D_7

Low data bus

Applies interrupt vector 80H in response to INTA.

Y_1 ...
A_1 ...

GG 74ALS244

V_{CC}

$\overline{INTA}$
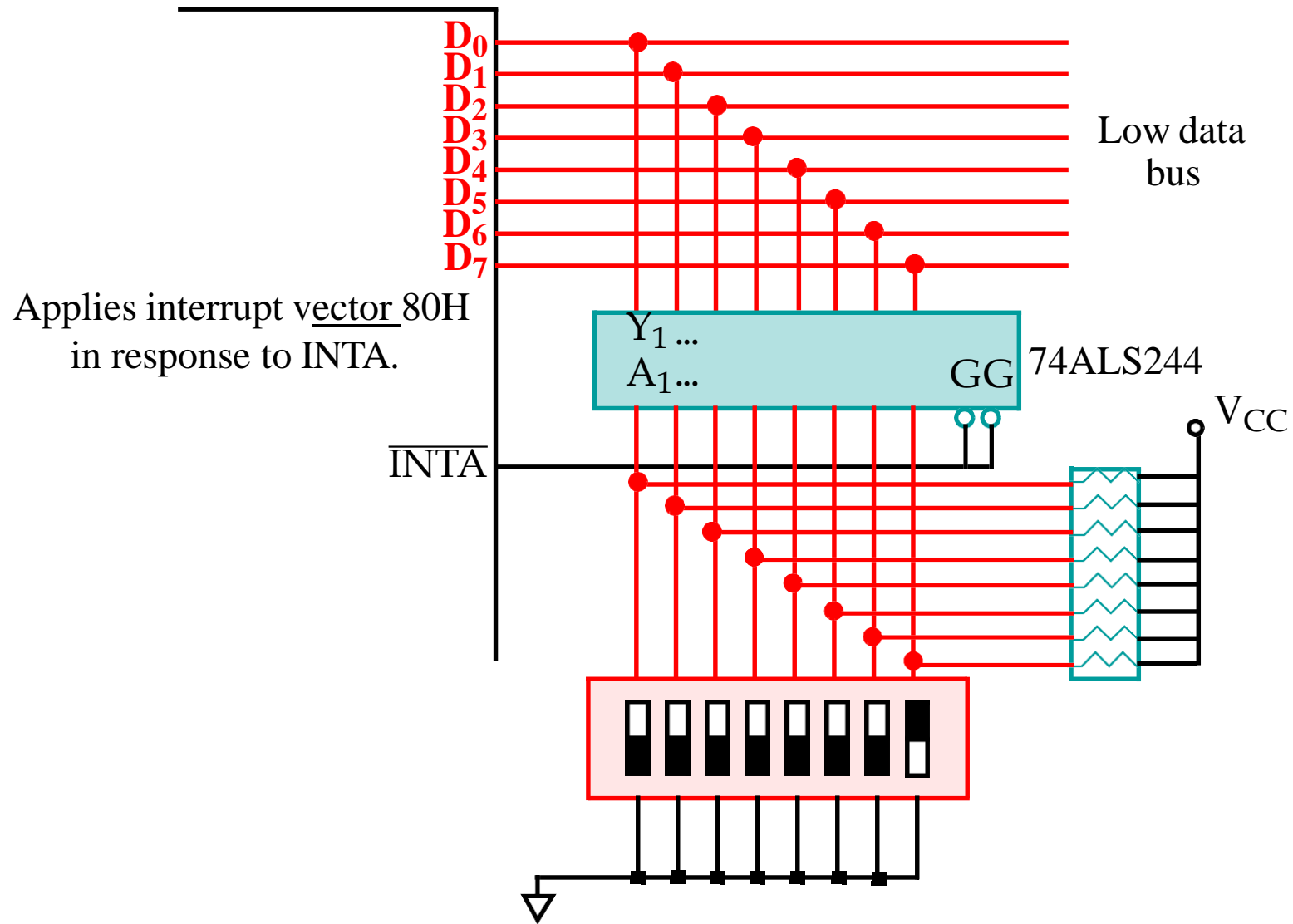
An Example 82C55 Interrupt Configuration

Handling more than 1 IRQ

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

IRQs

D_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7

V_{CC}

$Y_1 ...$
$A_1 ...$ 74ALS244 GG

$\overline{INTA}$

$\overline{INTR}$

$\overline{IRQ_0}$

$\overline{IRQ_6}$
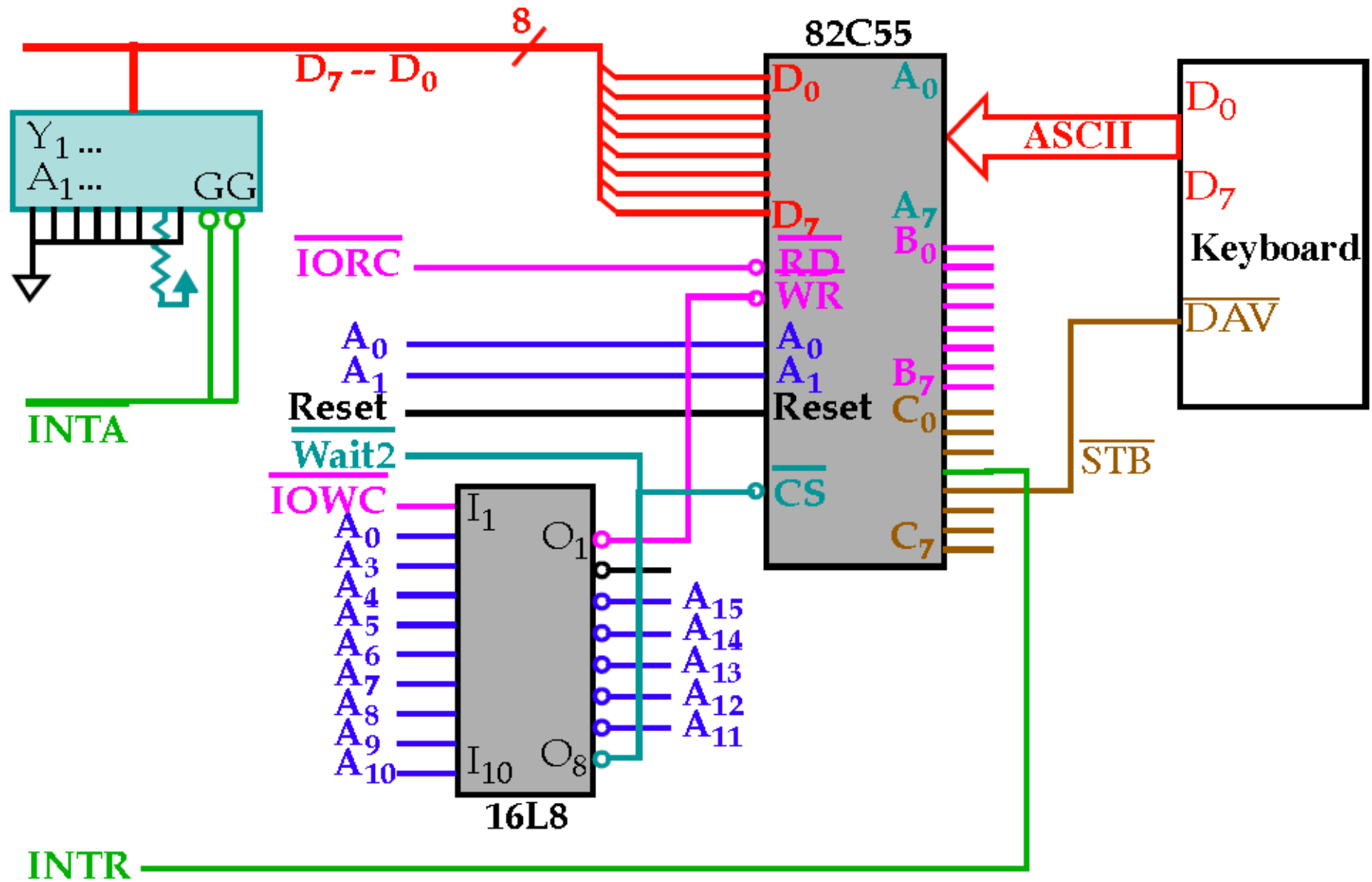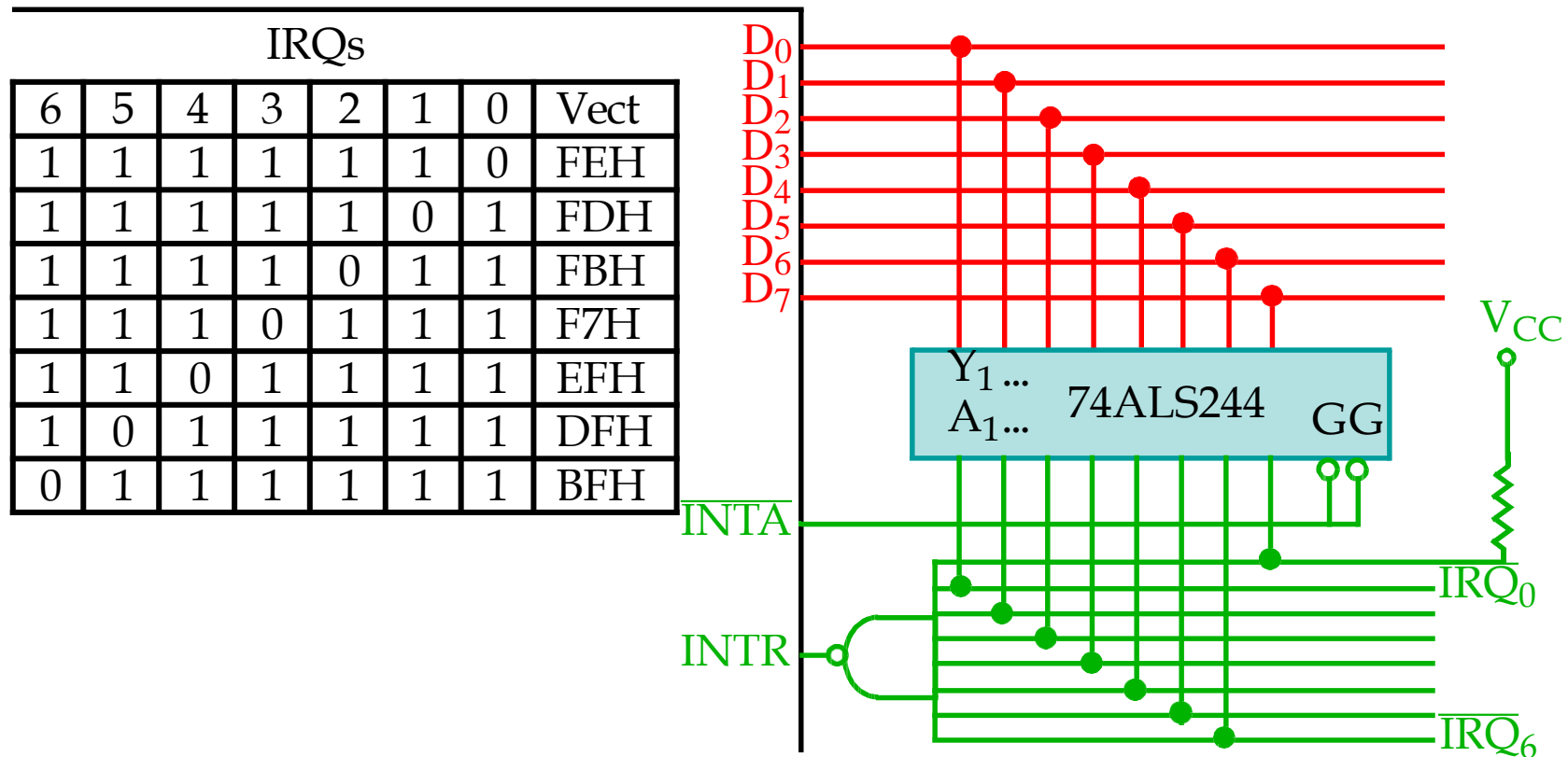
If any of $\overline{IRQ}_x$ goes low, the NAND goes low requesting an interrupt.

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

82C55

D_0 A_0

D_7 A_7

$\overline{RD}$ B_0

$\overline{WR}$

A_0 B_7

A_1 C_0

Reset

$\overline{CS}$

C_7

74ALS32

82C55

D_0 A_0

D_7 A_7

$\overline{RD}$ B_0

$\overline{WR}$

A_0 B_7

A_1 C_0

Reset

$\overline{CS}$

C_7

74ALS32

Only one interrupt vector required.

The scheme given earlier
that pulls the bus high (FFH)
can be used as the vector.

Priority is determined in
the ISR through polling
the individual 82C55s.
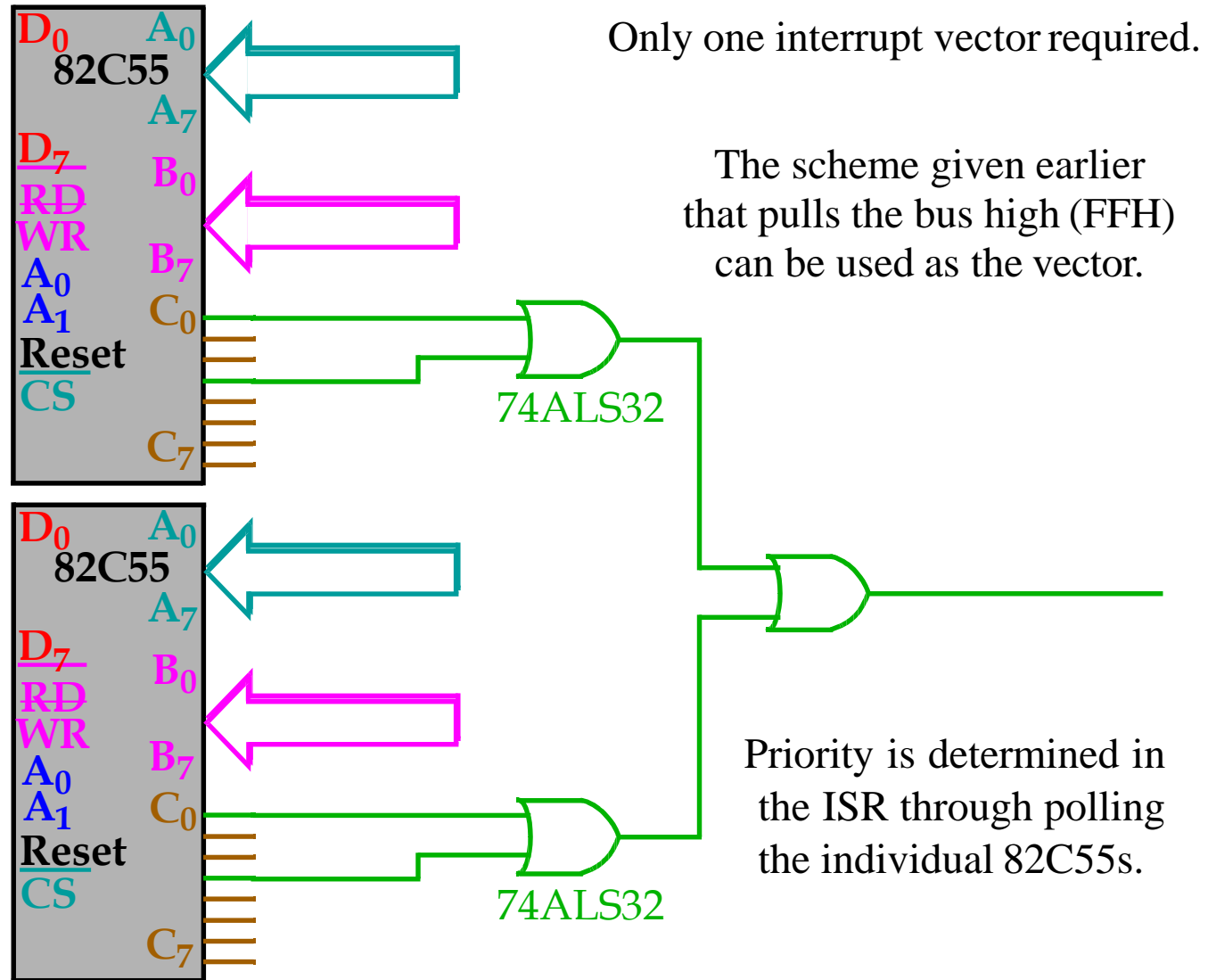
8259A Programmable Interrupt Controller

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.

8259A

D0
D1
D2
D3
D4
D5
D6
D7
A0
$\overline{CS}$
$\overline{RD}$
WR
SP/$\overline{EN}$
INT
INTA

IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7

CAS0
CAS1
CAS2

Connect either to devices
or to upstream slave 8259As
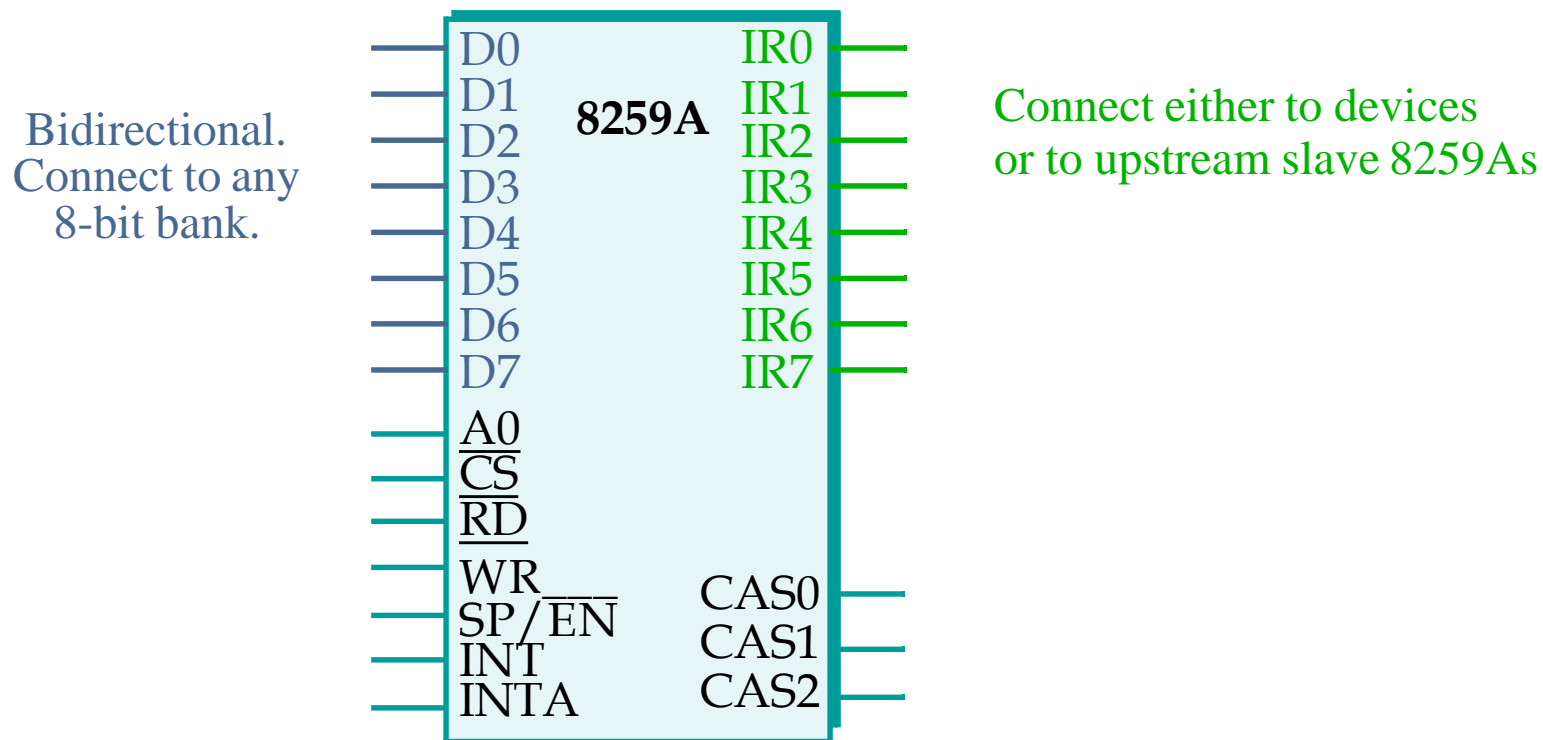
$\overline{CS}$ and $\overline{WR}$ must be decoded. Other connections are direct to microprocessor.

8259A Programmable Interrupt Controller

The meaning of the other connections:

O $\overline{WR}$

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

O $\overline{RD}$

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

O INT

Connects to the INTR pin on the microprocessor.

O $\overline{INTA}$

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

O A0

Selects different command words in the 8259A.

O $\overline{CS}$

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

O SP/$\overline{EN}$

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

O 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.

D7-D0

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

VCC

10K

A1
$\overline{\text{WAIT2}}$

A0
$\overline{\text{CS}}$

IORC
IOWC

$\overline{\text{RD}}$
$\overline{\text{WR}}$

M/$\overline{\text{IO}}$

I1   O1
A0
A2   I2   O2
A3   I3   A15
A4   I4   I16 A14
A5   I5   I15 A13
A6   I6   I14 A12
A7   I7   I13 A11
A8   I8   I12 A10
A9   I9   I11
      I10

INTR
INTA

SP/$\overline{\text{EN}}$ CAS0
INT   CAS1
INTA   CAS2

High indicates master mode.

Write the program to decode the 8259A
at ports 0400H and 0402H.
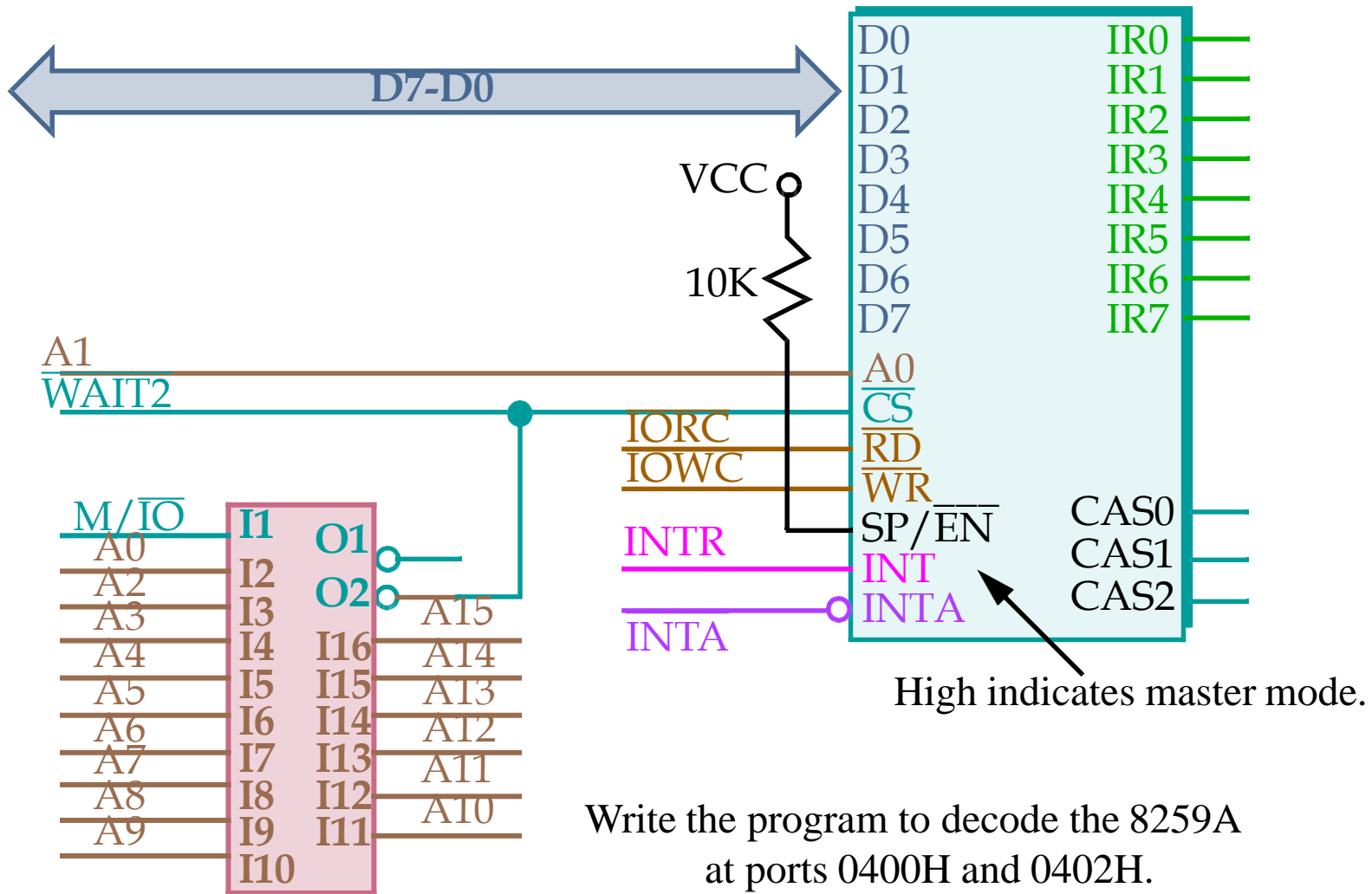
8259A Programmable Interrupt Controller
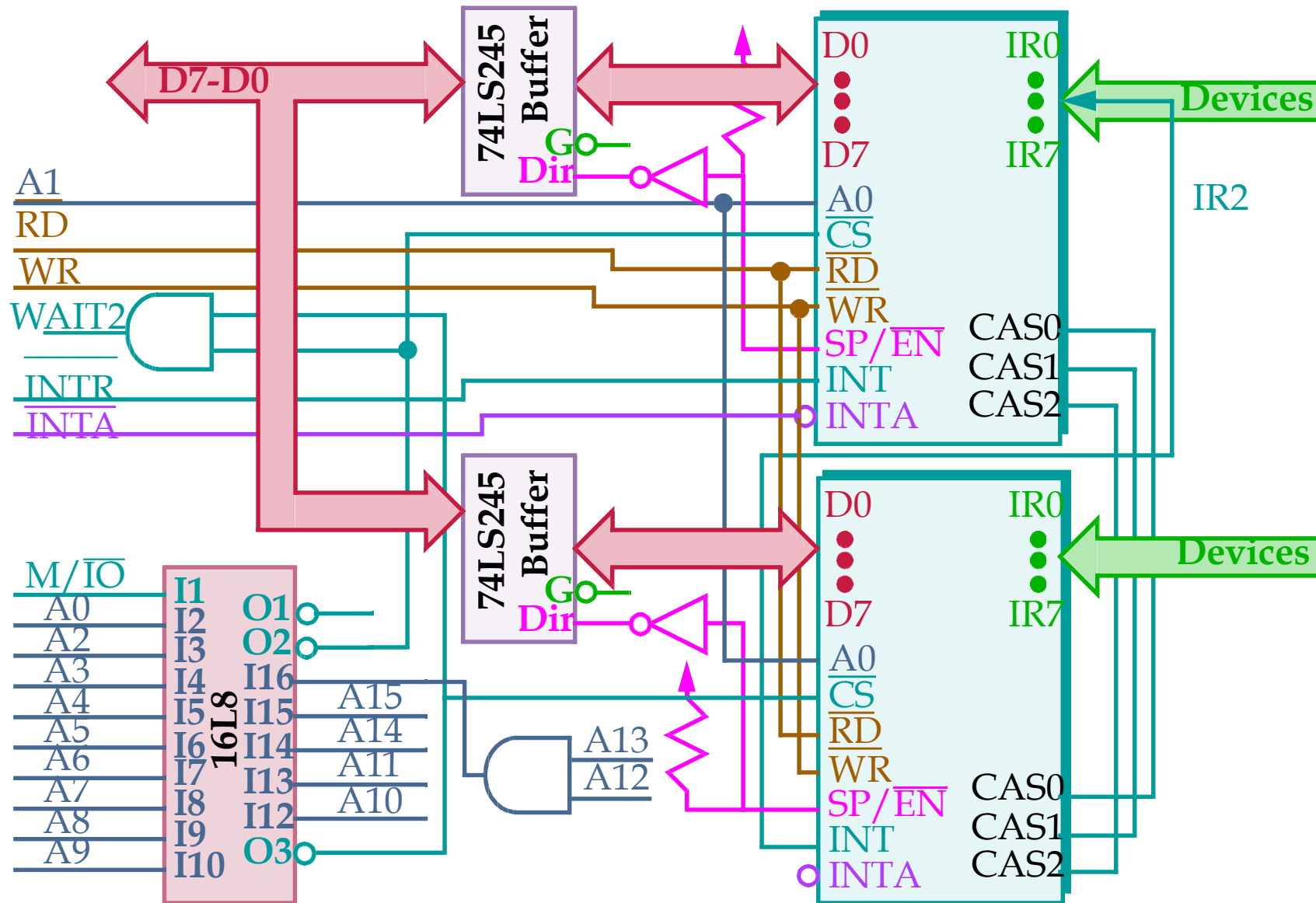
Programming the 8259A

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.

o ICW1:

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	X	X	X	1	LTIM	X	SNGL	IC4

1 = Level Triggered Mode
0 = Edge Triggered Mode

1 = ICW4 Needed
0 = No ICW4 Needed

1 = Single
0 = Cascade Mode
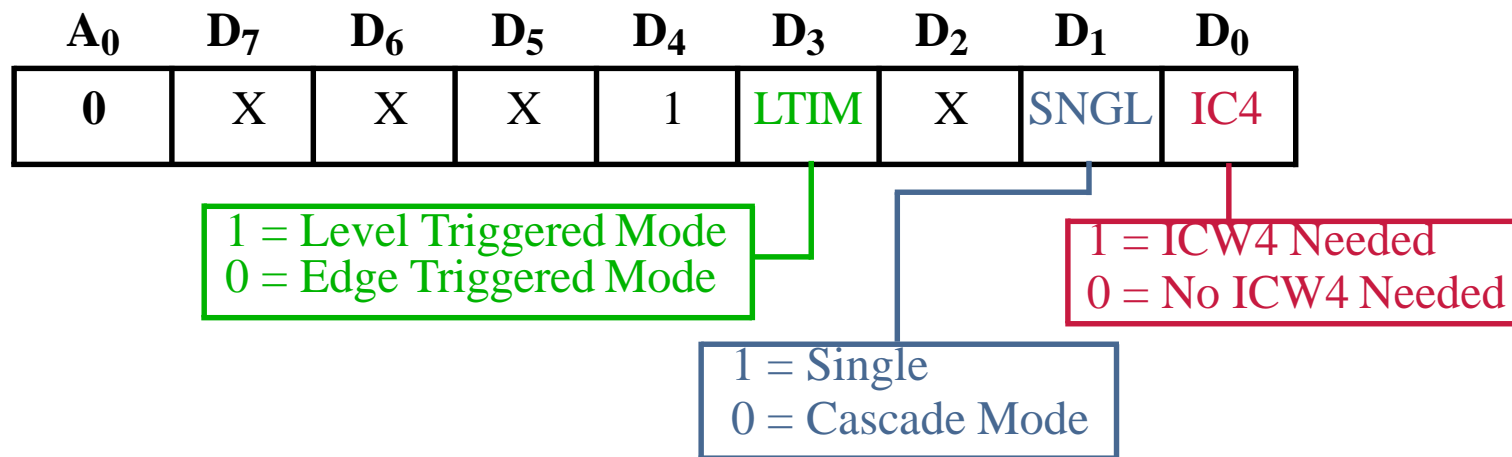
LTIM indicates if IRQ lines are positive edge-triggered or level-triggered.

Programming the 8259A

O ICW2:

Low order bits are 0 since there are 8 interrupts.

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	T_7	T_6	T_5	T_4	T_3	X	X	X

T7-T3 of Interrupt Vector Address (8086/8088 Mode)
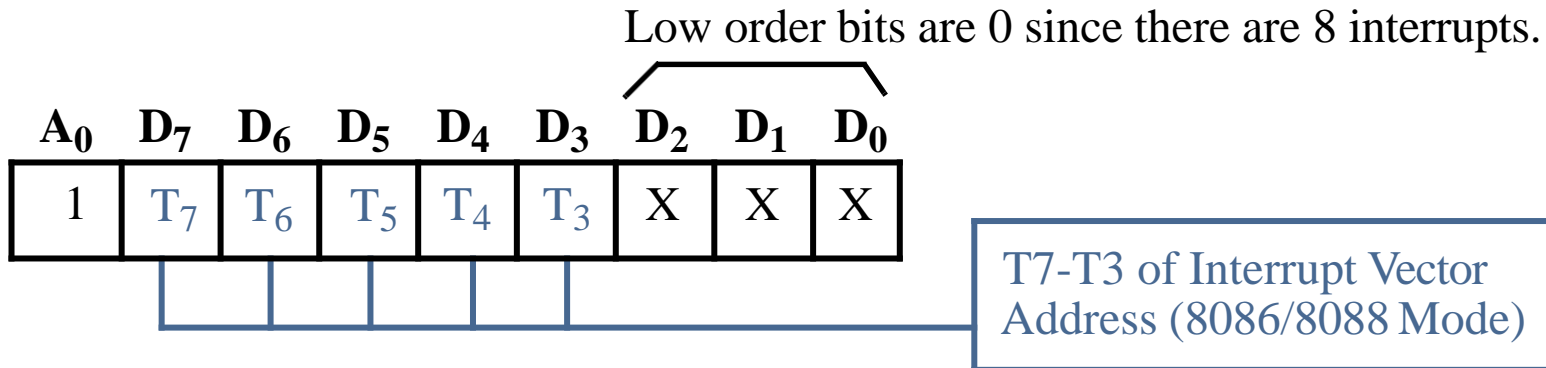
These bits determine the vector numbers used with the IRQ inputs.
For example, if programmed to generate vectors 08H-0FH, 08H is placed into these bit positions.

O ICW3:

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0

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

O ICW4:

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	SFNM	BUF	M/S	AEOI	1

1 = AUTO EOI
0 = NORMAL EOI

1 = Special Fully Nested Mode
0 = Not Special Fully
    Nested Mode

0	X	Non-Buffered Mode
1	0	Buffered Mode:Slave
1	1	Buffered Mode:Master
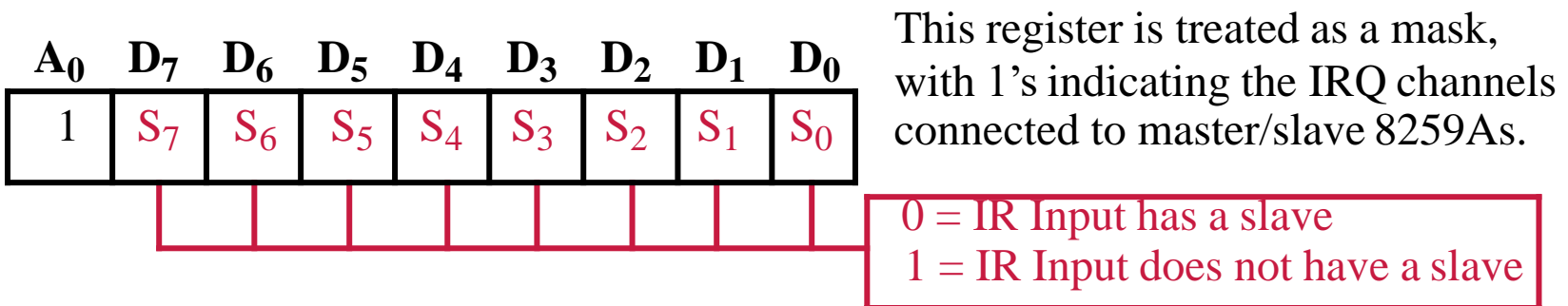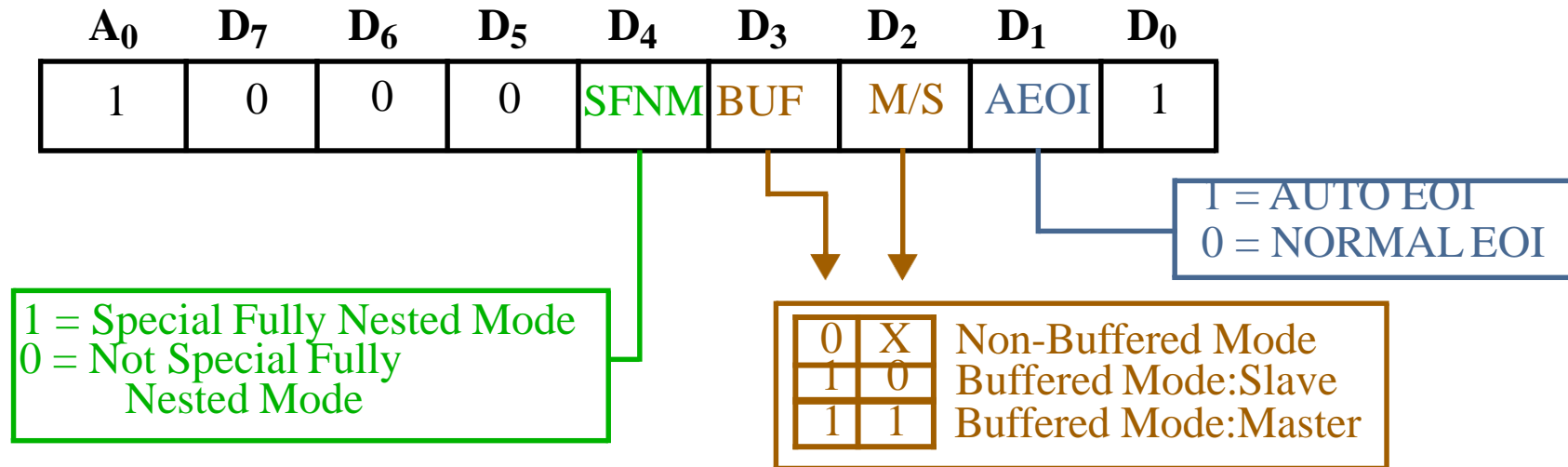
Fully nested mode allows the highest-priority interrupt request from a slave to be rec-
ognized by the master while it is processing another interrupt from a slave.
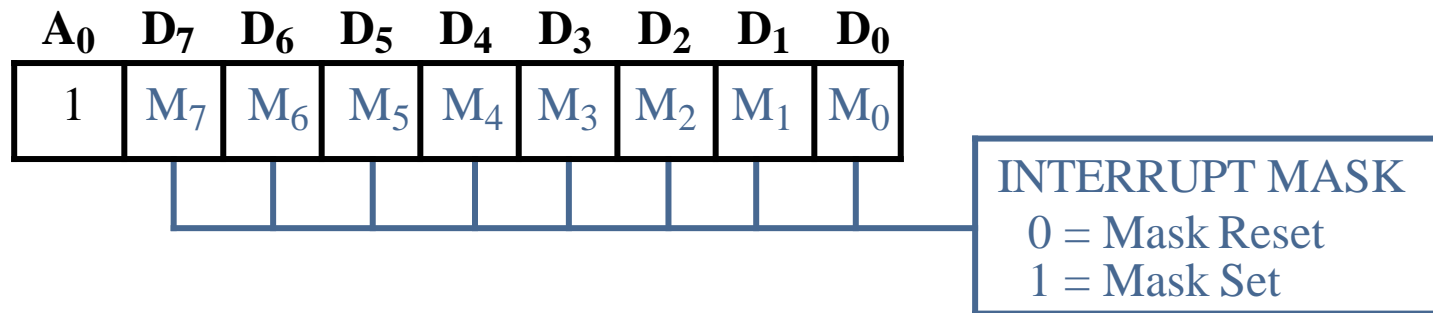
AEOI, if 1, indicates that an interrupt automatically resets the interrupt request bit,
otherwise OCW2 is consulted for EOI processing.

Programming the 8259A

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

O 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
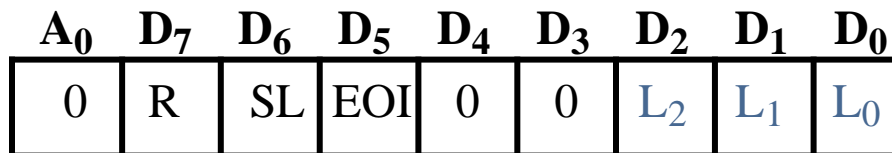
Only programmed when the AEOI mode in ICW4 is 0.
Allows you to control priorities after each interrupt is processed.

Programming the 8259A

O OCW2:

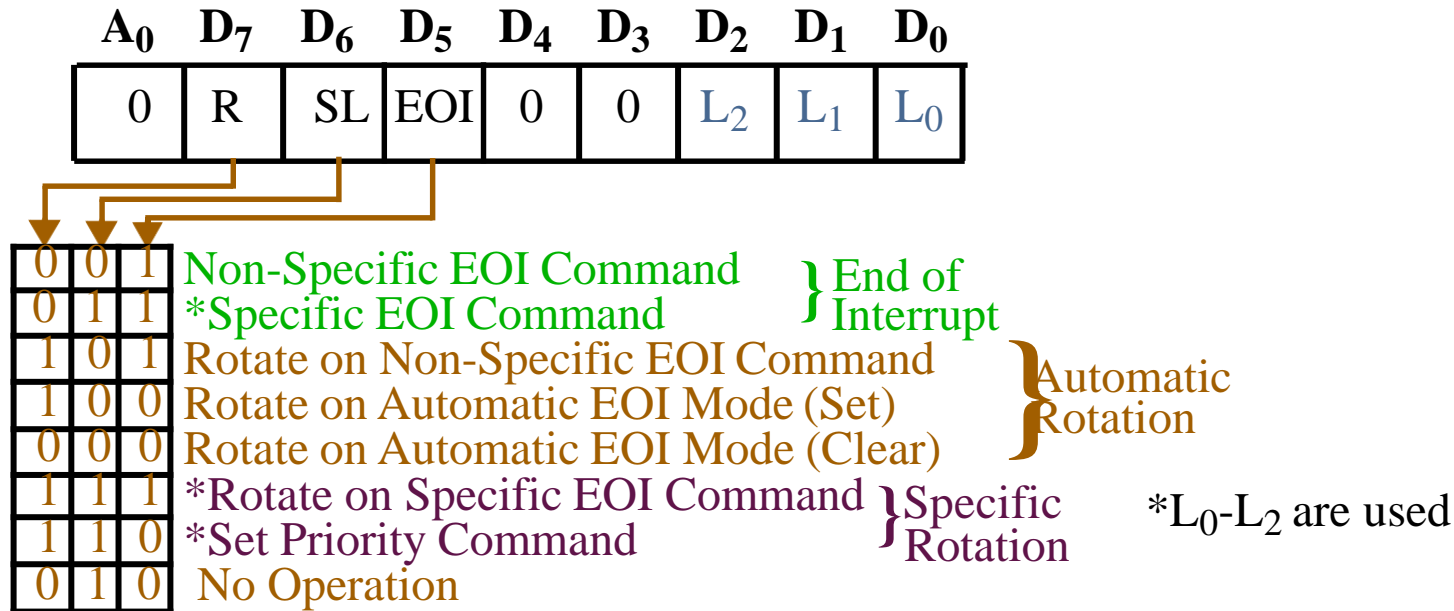| A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	R	SL	EOI	0	0	L_2	L_1	L_0

0	0	1	Non-Specific EOI Command	} End of Interrupt
0	1	1	*Specific EOI Command	
1	0	1	Rotate on Non-Specific EOI Command	} Automatic Rotation
1	0	0	Rotate on Automatic EOI Mode (Set)	
0	0	0	Rotate on Automatic EOI Mode (Clear)	
1	1	1	*Rotate on Specific EOI Command	} Specific Rotation
1	1	0	*Set Priority Command	
0	1	0	No Operation	

*L_0-L_2 are used

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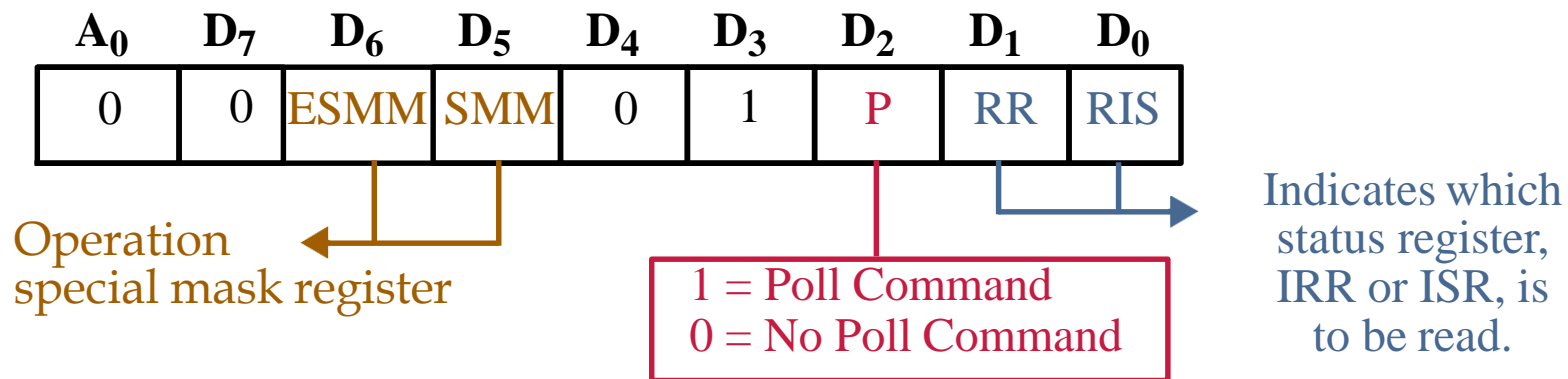
Specific EOI: ISR resets a specific interrupt request given by L_2-L_0.

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

O OCW3:

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	ESMM	SMM	0	1	P	RR	RIS

Operation
special mask register

1 = Poll Command
0 = No Poll Command

Indicates which
status register,
IRR or ISR, is
to be read.

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
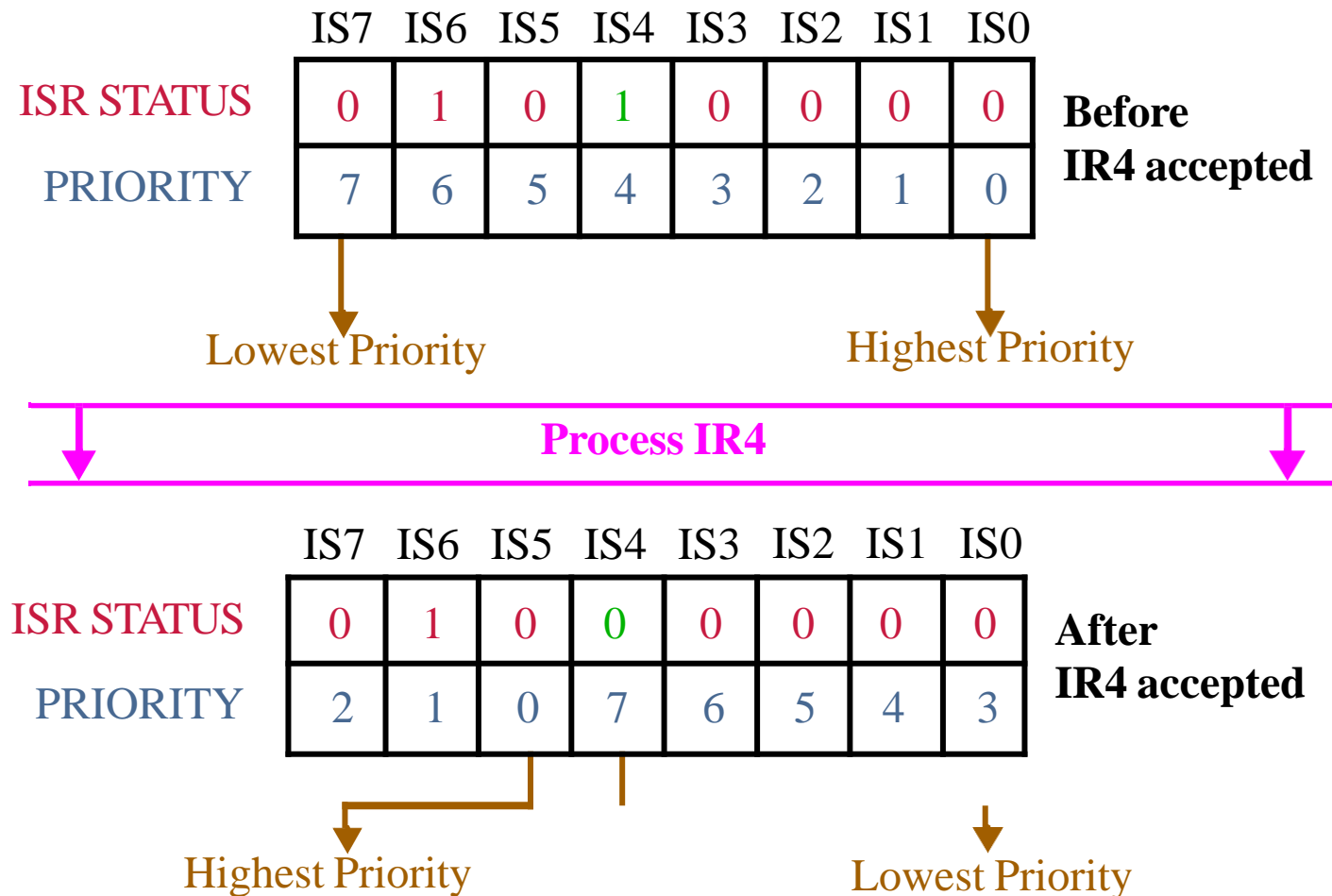interrupt request with highest priority.

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.

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0	
ISR STATUS	0	1	0	1	0	0	0	0	**Before**
PRIORITY	7	6	5	4	3	2	1	0	**IR4 accepted**

Lowest Priority   Highest Priority

Process IR4

	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0	
ISR STATUS	0	1	0	0	0	0	0	0	**After**
PRIORITY	2	1	0	7	6	5	4	3	**IR4 accepted**

Highest Priority   Lowest Priority

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:
O The transmitter is ready to send another character.
O The receiver has received a character.
O An error is detected while receiving data.
O A modem interrupt occurs.
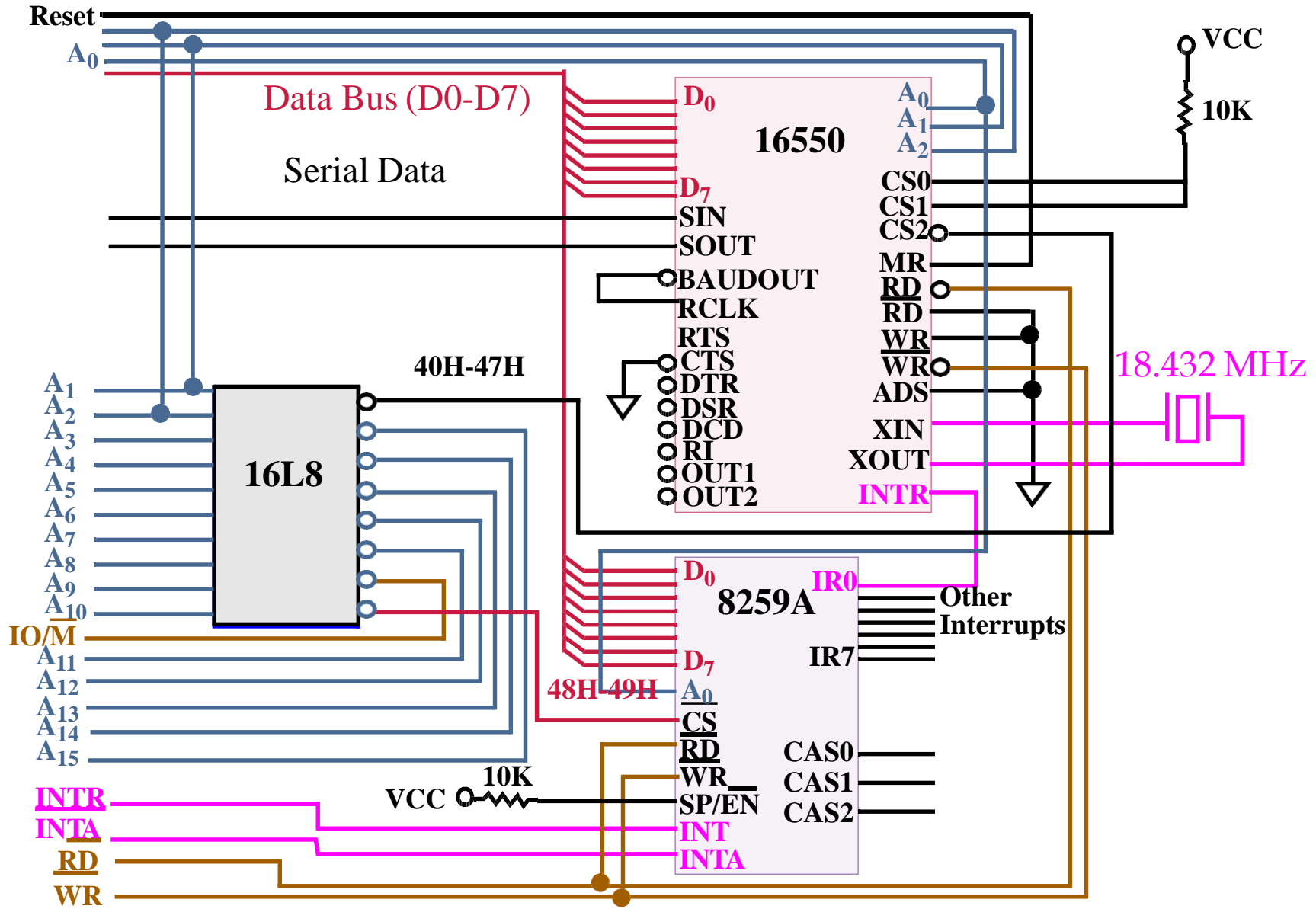
The16550 is decoded at 40H and 47H.
The 8259A is decoded at 48H and 49H.

Program in text shows the steps involved in programming both devices.
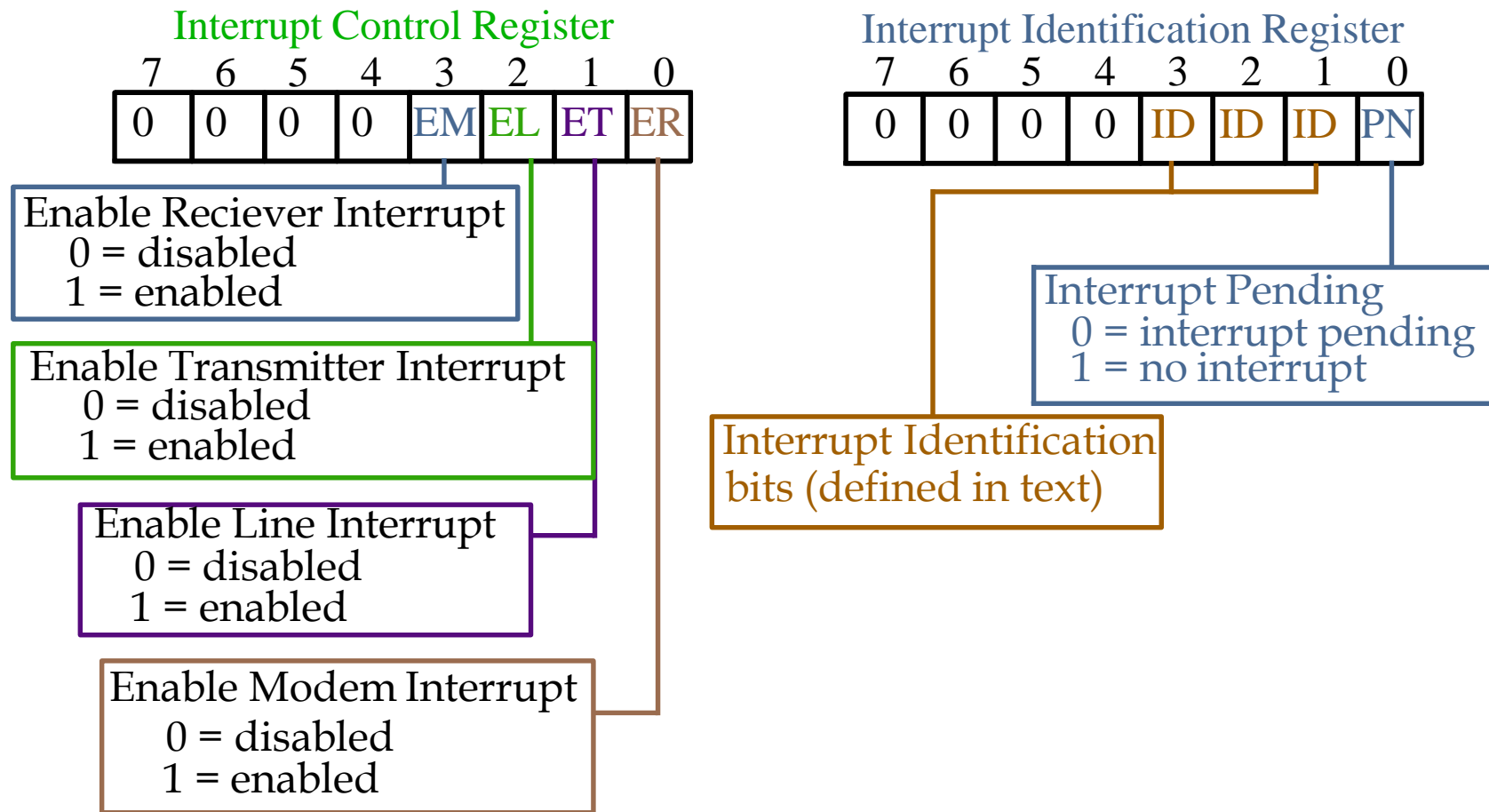Since the 16550 generates only one interrupt request for each of the above interrupts, the 16550 must be polled.

Remember the interrupt identification register of the 16550?

Interfacing 16550 UART using 8259A

16550 UART Interrupts

Interrupt Control Register

7	6	5	4	3	2	1	0
0	0	0	0	EM	EL	ET	ER

Enable Reciever Interrupt
   0 = disabled
   1 = enabled

Enable Transmitter Interrupt
   0 = disabled
   1 = enabled

Enable Line Interrupt
   0 = disabled
   1 = enabled

Enable Modem Interrupt
   0 = disabled
   1 = enabled

Interrupt Identification Register

7	6	5	4	3	2	1	0
0	0	0	0	ID	ID	ID	PN

Interrupt Pending
   0 = interrupt pending
   1 = no interrupt

Interrupt Identification
bits (defined in text)

Text gives ISR programming examples that show initialization and operation.