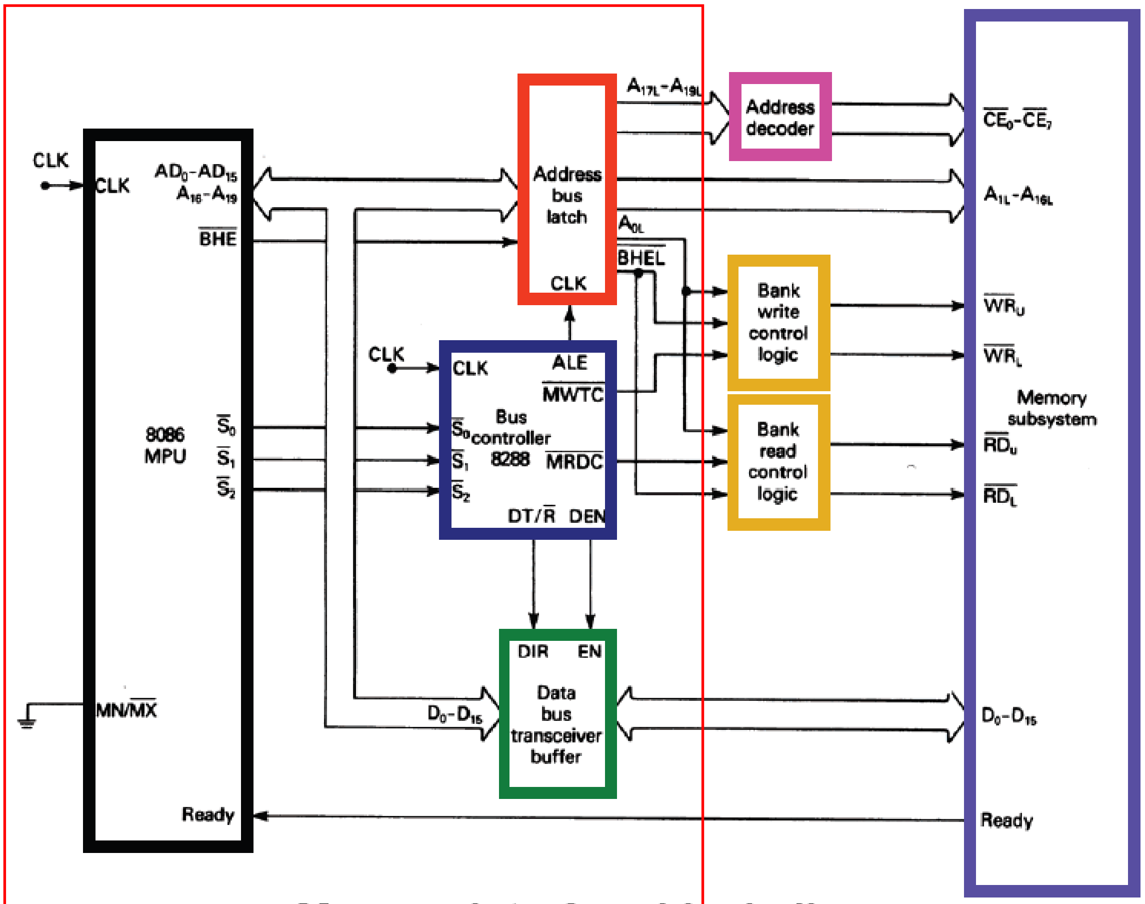
UNIT 4 I/O and Memory Interfacing Using 8085/8086

Lecture 1

Memory interfacing



Memory interface block diagram

Memory Interface Circuit

- Address bus latches and buffers
- Bank write and bank read control logic
- Data bus transceivers/buffers
- Address decoders

UNIT 4 I/O and Memory Interfacing Using 8085/8086

Lecture 2

Interrupts of 8085/8086 Microprocessors

Interrupt is a mechanism by which an I/O or an instruction can suspend the normal execution of processor and get itself serviced. Generally, a particular task is assigned to that interrupt signal. In the microprocessor based system the interrupts are used for data transfer between the peripheral devices and the microprocessor. The processor will check the interrupts always at the 2nd T-state of last machine cycle.

THE 8085 INTERRUPTS

Interrupt is a signal send by an external device to the processor, to the processor to perform a particular task or work. Mainly in the microprocessor based system the interrupts are used for data transfer between the peripheral and the microprocessor. When a peripheral is ready for data transfer, it interrupts the processor by sending an appropriate signal to the interrupt pin of the processor. If the processor accepts the interrupt then the processor suspends its current activity and executes an interrupt service subroutine to complete the data transfer between the peripheral and processor. After executing the interrupt service routine the processor resumes its current activity. This type of data transfer scheme is called interrupt driven data transfer scheme.

Types of Interrupts:- The interrupts are classified into software interrupts and hardware interrupts.

•The software interrupts are program instructions. These instructions are inserted at desired locations in a program. While running a program, lf a software interrupt instruction is encountered, then the processor executes an interrupt service routine (ISR).

•The hardware interrupts are initiated by an external device by placing an appropriate signal at the interrupt pin of the processor.

Interrupt Service Routine(ISR):-A small program or a routine that when executed services the corresponding interrupting source is called as an ISR.

Maskable/Non-Maskable Interrupt:-An interrupt that can be disabled by writing some instruction is known as Maskable Interrupt otherwise it is called Non-Maskable Interrupt.

There are two types of interrupts used in 8085 Microprocessor:

1.Hardware Interrupts

2.Software Interrupts

There are 6 pins available in 8085 for interrupt:

TRAP

RST 7.5

RST6.5

RST5.5

INTR

INTA

Execution of Interrupts:-

When there is an interrupt requests to the Microprocessor then after accepting the interrupts Microprocessor send the INTA (active low) signal to the peripheral. The vectored address of particular interrupt is stored in program counter. The processor executes an interrupt service routine (ISR) addressed in program counter.

8086 Interrupts

- **Interrupt Types**
  - **Hardware Interrupts: External event**
  - **Software Interrupts: Internal event (Software generated)**
  - **Maskable and non-maskable interrupts**
  - **Interrupt priority**
- **Interrupt Vectors and Interrupt Handlers**
- **Interrupt Controllers**

- **8086 Interrupt Pins**
  - **INTR: Interrupt Request. Activated by a peripheral device to interrupt the processor.**
    - **Level triggered. Activated with a logic 1.**
  - **/INTA: Interrupt Acknowledge. Activated by the processor to inform the interrupting device the the interrupt request (INTR) is accepted.**
    - **Level triggered. Activated with a logic 0.**
  - **NMI: Non-Maskable Interrupt. Used for major system faults such as parity errors and power failures.**
    - **Edge triggered. Activated with a positive edge (0 to 1) transition.**
    - **Must remain at logic 1, until it is accepted by the processor.**
    - **Before the 0 to 1 transition, NMI must be at logic 0 for at least 2 clock cycles.**

Interrupt Types

- **Type 0: Divide error – Division overflow or division by zero**
- **Type 1: Single step or Trap – After the execution of each instruction when trap flag set**
- **Type 2: NMI Hardware Interrupt – ‘1’ in the NMI pin**
- **Type 3: One-byte Interrupt – INT3 instruction (used for breakpoints)**
- **Type 4: Overflow – INTO instruction with an overflow flag**
- **Type 5: BOUND – Register contents out-of-bounds**
- **Type 6: Invalid Opcode – Undefined opcode occurred in program**
- **Type 7: Coprocessor not available – MSW indicates a coprocessor**

- **Type 8: Double Fault – Two separate interrupts occur during the same instruction**
- **Type 9: Coprocessor Segment Overrun – Coprocessor call operand exceeds FFFFH**
- **Type 10: Invalid Task State Segment – TSS invalid (probably not initialized)**
- **Type 11: Segment not present – Descriptor P bit indicates segment not present or invalid**
- **Type 12: Stack Segment Overrun – Stack segment not present or exceeded**
- **Type 13: General Protection – Protection violation in 286 (general protection fault)**
- **Type 14: Page Fault – 80386 and above**
- **Type 16: Coprocessor Error – ERROR′ = ‘0’ (80386 and above)**
- **Type 17: Alignment Check – Word/Doubleword data addressed at odd location (486 and above)**
- **Type 18: Machine Check – Memory Management interrupt (Pentium and above)**

Real Mode Interrupt

When current instruction execution completes, the processor checks:

1. Instruction executions
2. Single-step
3. NMI
4. Coprocessor segment overrun
5. INTR
6. INT instruction

When there is a pending interrupt:

7. The contents of the flag register are pushed onto the stack
8. IF and TF are cleared, disabling the INTR pin
9. CS is pushed to the stack
10. IP is pushed onto the stack
11. Interrupt Vector contents are fetched and placed into IP and CS, so the next instruction is the Interrupt Service Routine indicated by the Interrupt Vector

Protected Mode Interrupt

- **Exactly the same assignments as in Real Mode, but instead of Interrupt Vectors, there is an Interrupt Descriptor Table, located anywhere in memory (indicated by the IDTR)**

Interrupt Vectors

- **The *Interrupt Vector* contains the address of the interrupt service routine**
- **The *Interrupt Vector Table* is located in the first 1024 bytes of memory at address 000000H-0003FFH.**
- **It contains 256 different 4-byte interrupt vectors, grouped in 18 types**
  - **000H: Type 0 (Divide error)**
  - **004H: Type 1 (Single-step)**
  - **008H: Type 2 (NMI)**
  - **00CH: Type 3 (1-byte breakpoint)**

- **010H: Type 4 (Overflow)**
- **014H: Type 5 (BOUND)**
- **018H: Type 6 (Undefined opcode)**
- **01CH: Type 7 (Coprocessor not available)**
- **020H: Type 8 (Double fault)**
- **024H: Type 9 (Coprocessor segment overrun)**
- **028H: Type 10 (Invlid task state segment)**
- **02CH: Type 11 (Segment not present)**
- **030H: Type 12 (Stack segment overrun)**
- **034H: Type 13 (General protection)**
- **038H: Type 14 (Page fault)**
- **03CH: Type 15 (Unassigned)**
- **040H: Type 16 (Coprocessor error)**
- **044H-07CH: Type 14-31 (Reserved)**
- **080H: Type 32-255 (User)**

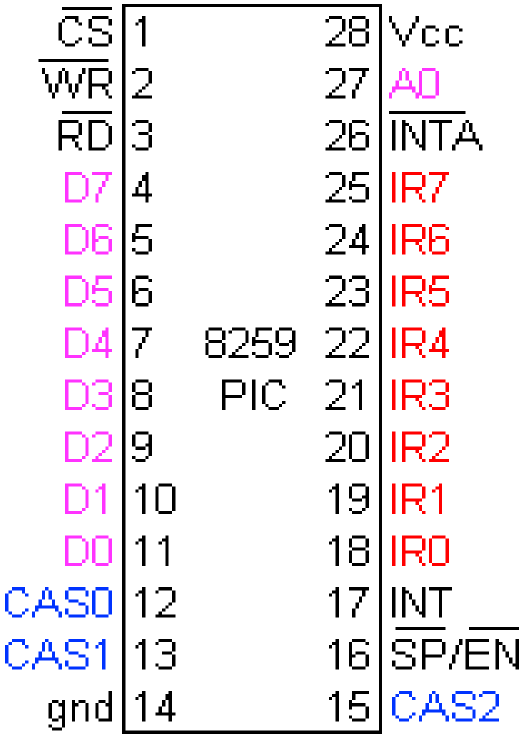
UNIT 4 I/O and Memory Interfacing Using 8085/8086

Lecture 3

8259A Programmable Interrupt Controller

Features

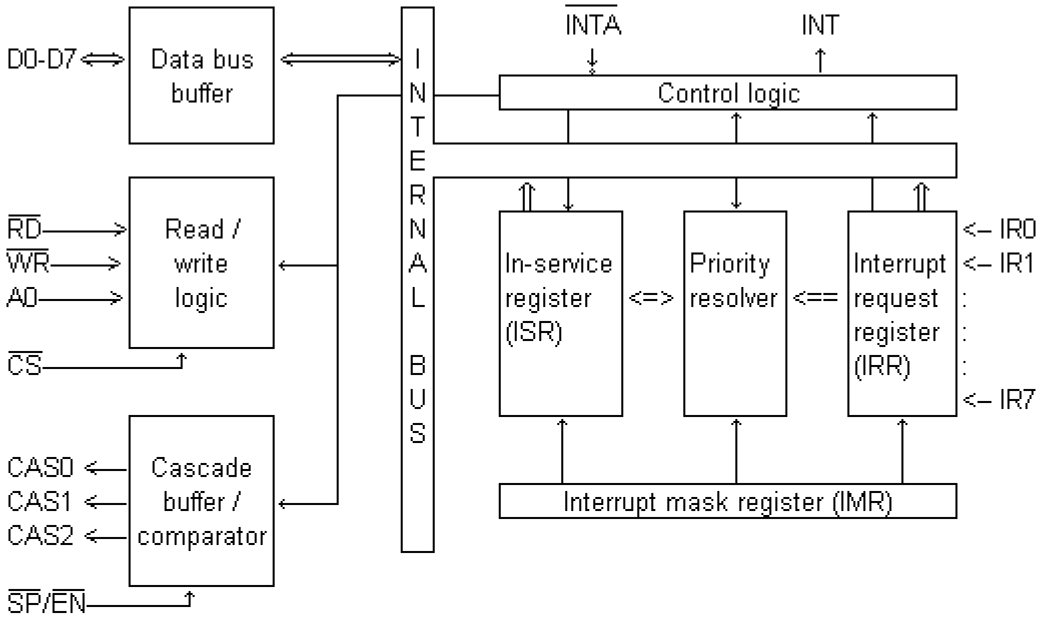
- 8 levels of interrupts.
- Can be cascaded in master-slave configuration to handle 64 levels of interrupts.
- Internal priority resolver.
- Fixed priority mode and rotating priority mode.
- Individually maskable interrupts.
- Modes and masks can be changed dynamically.
- Accepts IRQ, determines priority, checks whether incoming priority > current level being serviced, issues interrupt signal.
- In 8085 mode, provides 3 byte CALL instruction. In 8086 mode, provides 8 bit vector number.
- Polled and vectored mode.
- Starting address of ISR or vector number is programmable.
- No clock required.



PIN DIAGRAM OF 8259 PIC

D0-D7	Bi-directional, tristated, buffered data lines. Connected to data bus directly or through buffers
RD-bar	Active low read control
WR-bar	Active low write control
A0	Address input line, used to select control register
CS-bar	Active low chip select
CAS0-2	Bi-directional, 3 bit cascade lines. In master mode, PIC places slave ID no. on these lines. In slave mode, the PIC reads slave ID no. from master on these lines. It may be regarded as slave-select.
SP-bar / EN-bar	Slave program / enable. In non-buffered mode, it is SP-bar input, used to distinguish master/slave PIC. In buffered mode, it is output line used to enable buffers
INT	Interrupt line, connected to INTR of microprocessor
INTA-bar	Interrupt ack, received active low from microprocessor
IR0-7	Asynchronous IRQ input lines, generated by peripherals.

8259 internal block diagram



The 8259A 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 8259A's (up to 64 levels). It is programmed by the 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 8259A can be configured to match his 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 (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 being serviced.

PRIORITY RESOLVER

This logic block determines the priorites of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{\text{INTA}}$ pulse.

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 quality.

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 and 8086 input levels.

$\overline{\text{INTA}}$ (INTERRUPT ACKNOWLEDGE)

$\overline{\text{INTA}}$ pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode (µPM) of the 8259A.

DATA BUS BUFFER

This 3-state, bidirectional 8-bit buffer is used to interface the 8259A 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 8259A to be transferred onto the Data Bus.

$\overline{\text{CS}}$ (CHIP SELECT)

A LOW on this input enables the 8259A. No reading or writing of the chip will occur unless the device is selected.

$\overline{\text{WR}}$ (WRITE)

A LOW on this input enables the CPU to write control words (ICWs and OCWs) to the 8259A.

$\overline{\text{RD}}$ (READ)

A LOW on this input enables the 8259A to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR), or the Interrupt level onto the Data Bus.

A_0

This input signal is used in conjunction with $\overline{\text{WR}}$ and $\overline{\text{RD}}$ 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.

UNIT 4 I/O and Memory Interfacing Using 8085/8086

Lecture4

Programmable peripheral Interface(8255)(RGPV DEC-2014)

Data Bus Buffer

This three-state bi-directional 8-bit buffer is used to interface the 8255 to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status informa-tion are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

(CS) Chip Select. A "low" on this input pin enables the communcation between the 8255 and the CPU.

(RD) Read. A "low" on this input pin enables 8255 to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255.

(WR) Write. A "low" on this input pin enables the CPU to write data or control words into the 8255.

(A0 and A1) Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word register. They are normally connected to the least significant bits of the address bus (A0 and A1).

(RESET) Reset. A "high" on this input initializes the control register to 9Bh and all ports (A, B, C) are set to the input mode.

A1	A0	SELECTION
0	0	PORT A
0	1	PORT B
1	0	PORT C
1	1	CONTROL

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255. Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

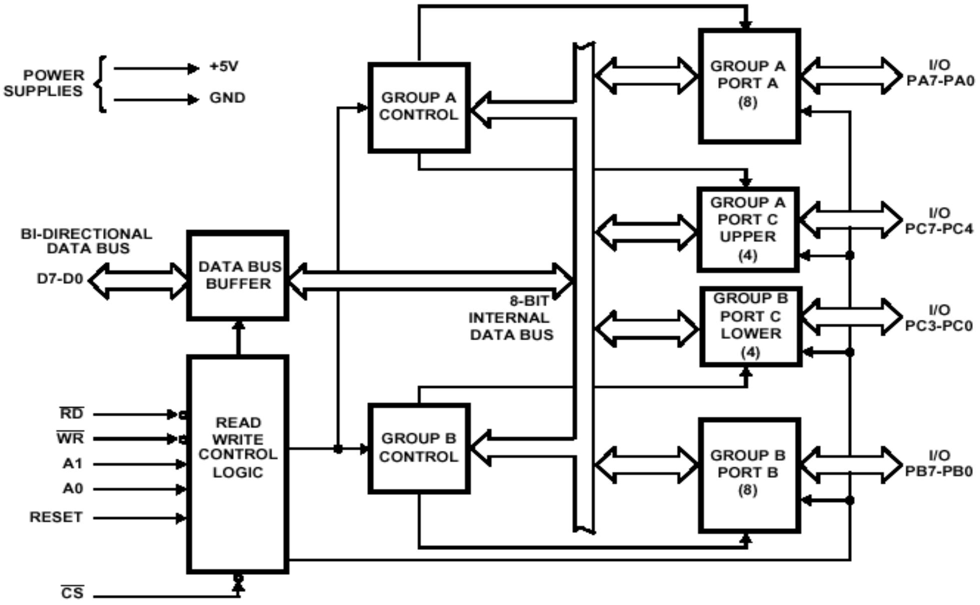
Ports A, B, and C

The 8255 contains three 8-bit ports (A, B, and C). All can be configured to a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255.

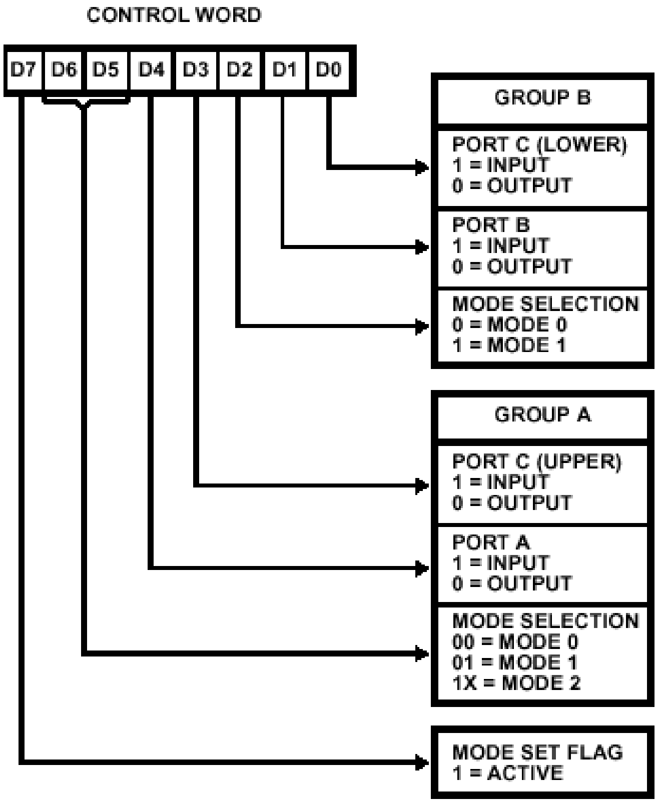
Port A One 8-bit data output latch/buffer and one 8-bit data input latch. Both "pull-up" and "pull-down" bus-hold devices are present on Port A.

Port B One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal output and status signal inputs in conjunction with ports A and B.



Block Diagram of the 8255 Programmable Peripheral Interface (PPI)



Mode Definition Format

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain the different modes of 8255	Dec 2014	7

Programmable Interval Timers (RGPV DEC-2014)

A **Programmable Interval Timer (PIT)** is a counter which triggers an interrupts when they reach their programmed count. The **8253** and **8254**microcontrollers are PITs avialble for the i86 architectures used as timer for i86-compatable systems.

These PICs include three timers that are used for different putposes. The first timer is useually used as the **System Clock**. Timer 2 was used for RAM refreshing, and timer 3 is connected to the PC speaker. We will see all of the connections a little later, so we won't go into much detail now.

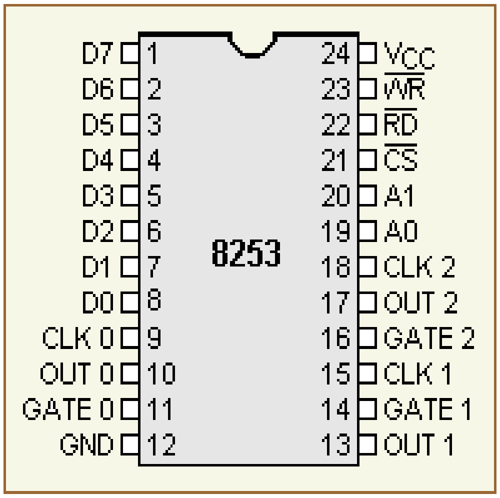
Instead, lets take a closer look at one of these famous PITs... The 8253 Microcontroller.

8253 Hardware

Before looking at the software side of things, it will be helpful to learn more about what we are actually programming. Because of this, we will look at the 8253 hardware first, and learn how it works and is connected to the rest of the PC. We will also be looking at internal registers, pin layout, command words, and more that will be needed for the software side of things.

8253 Hardware: Description

The 8253 PIT has a simple interface, and is not that hard to program.



- **D0...D7:** 8 bit data lines. This is connected to the data bus so we can read and send commands.
- **CLK 0, CLK 1, CLK 2:** Clock input pins. There are 3 pins for 3 separate counters
- **OUT 0, OUT 1, OUT 2:** Output data line. There are 3 pins for 3 separate counters
- **GATE 0, GATE 1, GATE 2:**Gate data line. There are 3 pins for 3 separate counters
- **GND:** Ground
- **Vcc:** Input voltage
- **WR:** Write enable. When this line is active, lets the 8253 that we are writing data
- **RD:** Read enable. When this line is active, lets the 8253 that we are reading data
- **CS:** Chip select signal
- **A0, A1:** Address lines. Used to determin what register we are accessing.

Not to bad. There are a couple of important pins here that we need to look at.

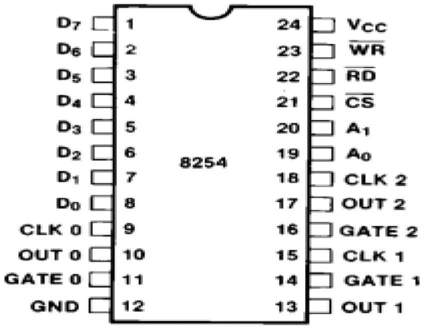
The **D0-D7** pins connect to the **systems data bus**. These pins carry our data when we are sending or reading data to the controller.

Vcc and **GND** complete the circuit (Voltage input, Ground output.)

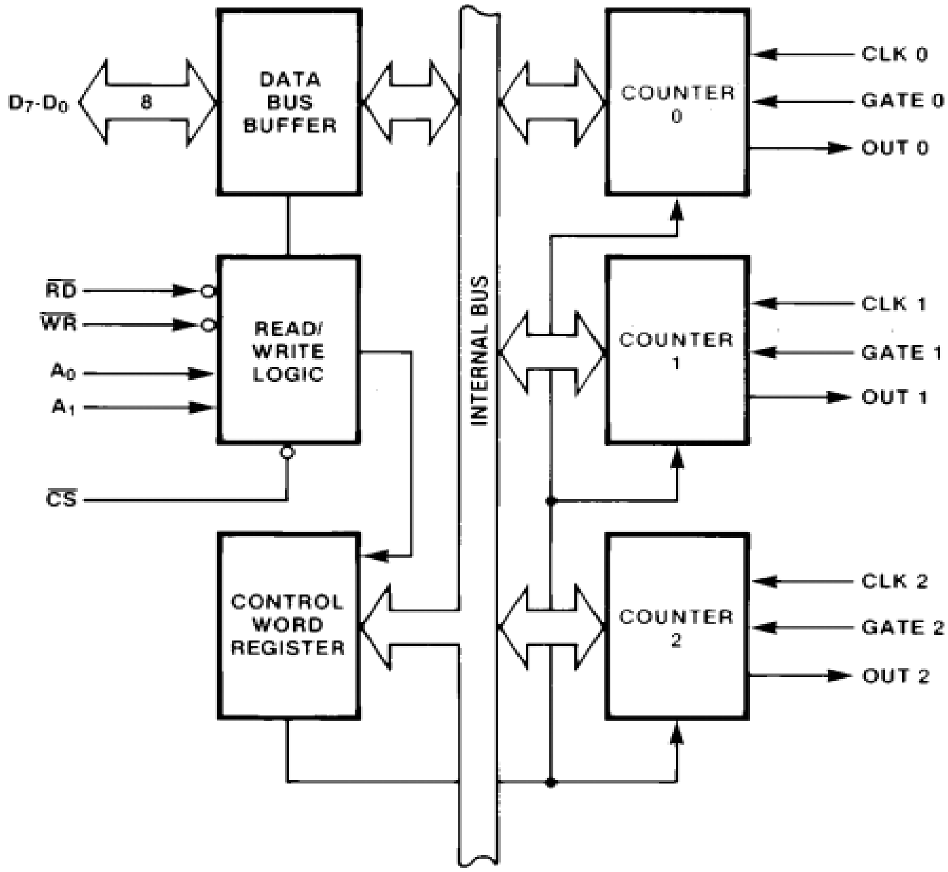
The **WR** pin tells the controller that we are writing (expect input on the data pins.) When the signal in this pin is "low", we are currently sending data. The **RD** pin is very simular in this manner, but it tells the controller we are reading data instead. The **CS** pin is a special pin that determins what the controller should do with the **RD** and **WR** pins. If the CS pin is "low", the controller will respond to the RD and WR pins. If CS is not, they are ignored. **WR** and **RD** connect to the **Systems Control Bus**. The **CS** pin connects to the systems **Address Bus** for port i/o operations.

The **A0** and **A1** pins are connected to the **Systems Address Bus**, and are used to determin what register we are accessing. These, in conjunction with **WR** and **RD** allows the controller to determin if are reading to or from a register.

Notice that there are three groups of the CLK, OUT, and GATE pins. Yes, there is indeed a reason for this: **The 8253/8254 microcontrollers contain 3 independent timers.**



Pin Configuration



8254 Block Diagram

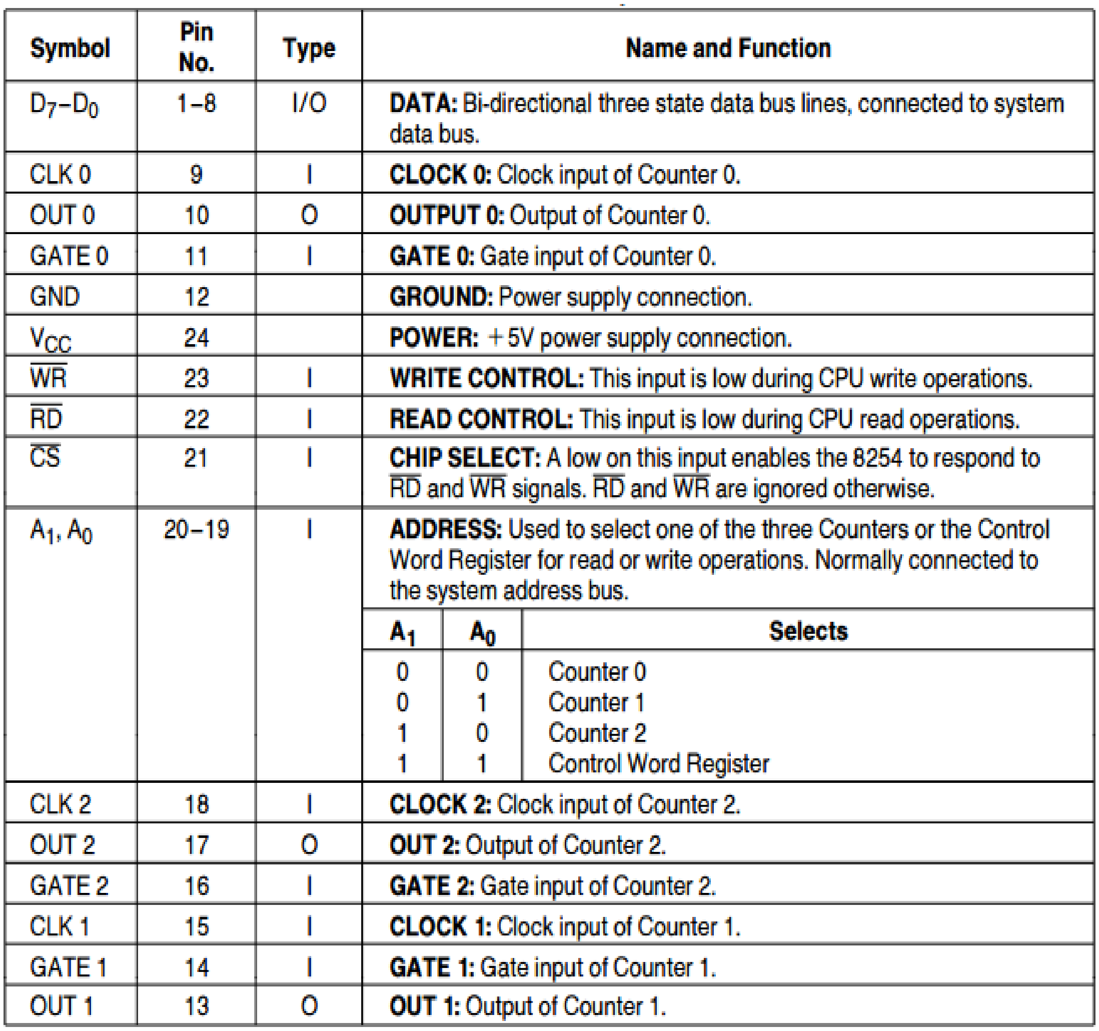
DATA BUS BUFFER:- This 3-state, bi-directional, 8-bit buffer is used to interface the 8254 to the system bus

READ/WRITE LOGIC:- The Read/Write Logic accepts inputs from the system bus and generates control signals for the other functional blocks of the 8254. A1 and A0 select one of the three counters or the Control Word Register to be read from/written into. A ‘‘low’’ on the RD input tells the 8254 that the CPU is reading one of the counters. A ‘‘low’’ on the WR input tells the 8254 that the CPU is writing either a Control Word or an initial count. Both RD and WR are qualified by CS; RD and WR are ignored unless the 8254 has been selected by holding CS low.

CONTROL WORD REGISTER:- The Control Word Register is selected by the Read/Write Logic when A1,A0 e 11. If the CPU then does a write operation to the 8254, the data is stored in the Control Word Register and is interpreted as a Control Word used to define the operation of the Counters. The Control Word Register can only be written to; status information is available with the Read-Back Command.

COUNTER 0, COUNTER 1, COUNTER 2 :-These three functional blocks are identical in operation, so only a single Counter will be described. The Counters are fully independent. Each Counter may operate in a different Mode. The Control Word Register is shown in the figure; it is not part of the Counter itself, but its contents determine how the Counter operates. The status register when latched, contains the current contents of the Control Word Register and status of the output and null count flag. The actual counter is labelled CE (for ‘‘Counting Element’’). It is a 16-bit presettable synchronous down counter. OLM and OLL are two 8-bit latches. OL stands for ‘‘Output Latch’’; the subscripts M and L stand for ‘‘Most significant byte’’ and ‘‘Least significant byte’’respectively.

Both are normally referred to as one unit and called just OL. These latches normally ‘‘follow’’ the CE, but if a suitable Counter Latch Command is sent to the 8254, the latches ‘‘latch’’ the present count until read by the CPU and then return to ‘‘following’’ the CE. One latch at a time is enabled by the counter's Control Logic to drive the internal bus. This is how the 16-bit Counter communicates over the 8-bit internal bus. Note that the CE itself cannot be read; whenever you read the count, it is the OL that is being read. Similarly, there are two 8-bit registers called CRM and CRL (for ‘‘Count Register’’). Both are normally referred to as one unit and called just CR. When a new count is written to the Counter, the count is stored in the CR and later transferred to the CE. The Control Logic allows one register at a time to be loaded from the internal bus. Both bytes are transferred to the CE simultaneously. CRM and CRL are cleared when the Counter is programmed. In this way, if the Counter has been programmed for one byte counts (either most significant byte only or least significant byte only) the other byte will be zero. Note that the CE cannot be written into; whenever a count is written, it is written into the CR. The Control Logic is also shown in the diagram. CLK n, GATE n, and OUT n are all connected to the outside world through the Control Logic.



Symbol	Pin No.	Type	Name and Function
D_7–D_0	1–8	I/O	**DATA:** Bi-directional three state data bus lines, connected to system data bus.
CLK 0	9	I	**CLOCK 0:** Clock input of Counter 0.
OUT 0	10	O	**OUTPUT 0:** Output of Counter 0.
GATE 0	11	I	**GATE 0:** Gate input of Counter 0.
GND	12		**GROUND:** Power supply connection.
V_{CC}	24		**POWER:** +5V power supply connection.
$\overline{WR}$	23	I	**WRITE CONTROL:** This input is low during CPU write operations.
$\overline{RD}$	22	I	**READ CONTROL:** This input is low during CPU read operations.
$\overline{CS}$	21	I	**CHIP SELECT:** A low on this input enables the 8254 to respond to $\overline{RD}$ and $\overline{WR}$ signals. $\overline{RD}$ and $\overline{WR}$ are ignored otherwise.
A_1, A_0	20–19	I	**ADDRESS:** Used to select one of the three Counters or the Control Word Register for read or write operations. Normally connected to the system address bus. **A_1 A_0 Selects** 0 0 Counter 0 0 1 Counter 1 1 0 Counter 2 1 1 Control Word Register
CLK 2	18	I	**CLOCK 2:** Clock input of Counter 2.
OUT 2	17	O	**OUT 2:** Output of Counter 2.
GATE 2	16	I	**GATE 2:** Gate input of Counter 2.
CLK 1	15	I	**CLOCK 1:** Clock input of Counter 1.
GATE 1	14	I	**GATE 1:** Gate input of Counter 1.
OUT 1	13	O	**OUT 1:** Output of Counter 1.

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain the various operating modes of 8253 programmable interval timer	Dec 2014	7

UNIT 4 I/O and Memory Interfacing Using 8085/8086

Lecture 7

DMA Controller 8257 (RGPV DEC-2014)

<u>Features:</u>

It is a 4-channel DMA.

- **So 4 I/O devices can be interfaced to DMA**
- **It is designed by Intel**
- **Each channel have 16-bit address and 14 bit counter**
- **It provides chip priority resolver that resolves priority of channels in fixed or rotating mode.**

It provide on chip channel inhibit logic

- **It generates a TC signal to indicate the peripheral that the programmed number of data bytes have been transferred.**
- **It generates MARK signal to indicate the peripheral that 128 bytes have been transferred.**
- **It requires single phase clock.**
- **The maximum frequency is 3Mhz and minimum frequency is 250 Hz.**
- **It execute 3 DMA cycles**

1.DMA read
2.DMA write
3.DMA verify.

- **It provide AEN signal that can be used to isolate CPU and other devices from the system bus.**
- **It is operate in two modes.**

1.Master Mode
2.Slave Mode

Pin Diagram of DMA controller

- **<u>D0-D7:</u>**
- **it is a bidirectional ,tri state ,Buffered ,Multiplexed data (D0-D7)and (A8-A15).**
- **In the slave mode it is a bidirectional (Data is moving).**
- **In the Master mode it is a unidirectional (Address is moving).**
- **<u>IOR:</u>**
- **It is active low ,tristate ,buffered ,Bidirectional lines.**
- **In the slave mode it function as a input line. IOR signal is generated by microprocessor to read the contents 8257 registers.**
- **In the master mode it function as a output line. IOR signal is generated by 8257 during write cycle**
- **<u>IOW:</u>**
- **It is active low ,tristate ,buffered ,Bidirectional control lines.**
- **In the slave mode it function as a input line. IOR signal is generated by microprocessor to**

write the contents 8257 registers.

- **In the master mode it function as a output line. IOR signal is generated by 8257 during read cycle**

CLK:

- **It is the input line ,connected with TTL clock generator.**
- **This signal is ignored in slave mode.**

RESET:

- **Used to clear mode set registers and status registers**

A0-A3:

These are the tristate, buffer, bidirectional address lines.

In slave mode ,these lines are used as address inputs lines and internally decoded to access the internal registers.

In master mode, these lines are used as address outputs lines,A0-A3 bits of memory address on the lines.

CS:

- **It is active low, Chip select input line.**
- **In the slave mode, it is used to select the chip.**
- **In the master mode, it is ignored.**

A4-A7:

These are the tristate, buffer, output address lines.

In slave mode ,these lines are used as address outputs lines.

In master mode, these lines are used as address outputs lines,A0-A3 bits of memory address on the lines.

READY:

- **It is a asynchronous input line.**
- **In master mode,**
- **When ready is high it is received the signal.**
- **When ready is low, it adds wait state between S1 and S3**
- **In slave mode ,this signal is ignored.**

HRQ:

- **It is used to receiving the hold request signal from the output device.**

HLDA:

- **It is acknowledgment signal from microprocessor.**

MEMR:

- **It is active low ,tristate ,Buffered control output line.**
- **In slave mode, it is tristated.**
- **In master mode ,it activated during DMA read cycle.**

MEMW:

- **It is active low ,tristate ,Buffered control input line.**
- **In slave mode, it is tristated.**
- **In master mode ,it activated during DMA write cycle.**

AEN (Address enable):

- **It is a control output line.**
- **In master mode ,it is high**

- In slave mode ,it is low
- Used it isolate the system address ,data ,and control lines.

<u>ADSTB: (Address Strobe)</u>

- It is a control output line.
- Used to split data and address line.
- It is working in master mode only.
- In slave mode it is ignore.

<u>TC (Terminal Count):</u>

- It is a status of output line.
- It is activated in master mode only.
- It is high ,it selected the peripheral.
- It is low ,it free and looking for a new peripheral.

MARK:

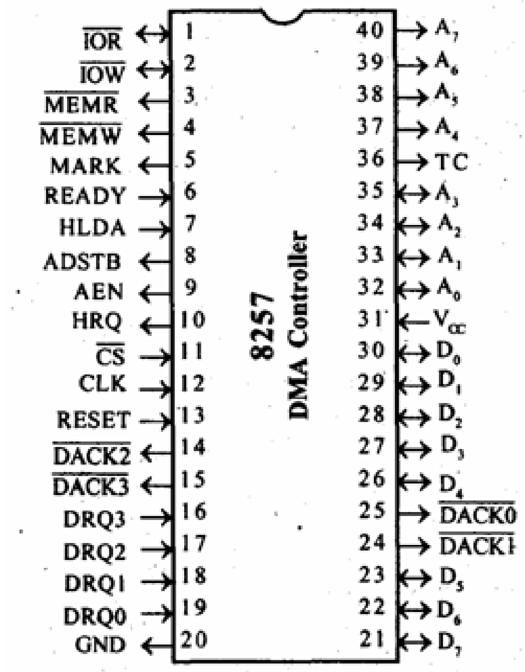
- It is a modulo 128 MARK output line.
- It is activated in master mode only.
- It goes high ,after transferring every 128 bytes of data block.

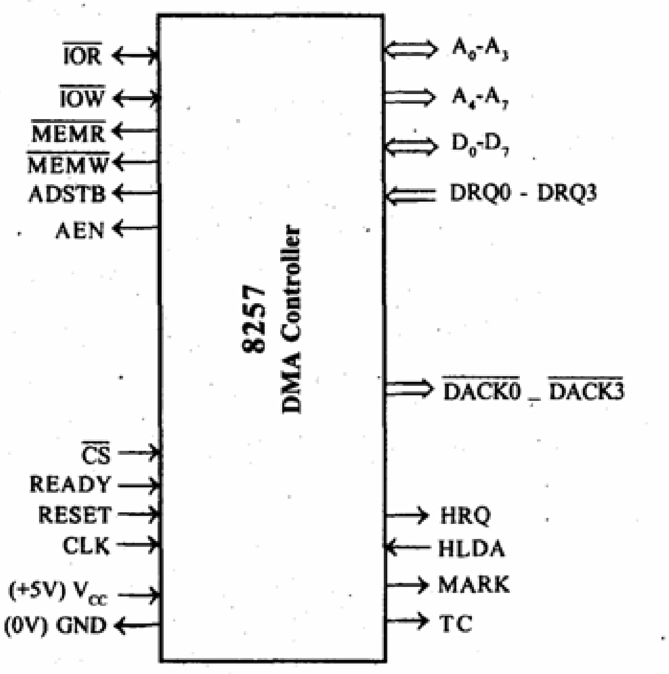
<u>DRQ0-DRQ3(DMA Request):</u>

- These are the asynchronous peripheral request input signal.
- The request signals is generated by external peripheral device.

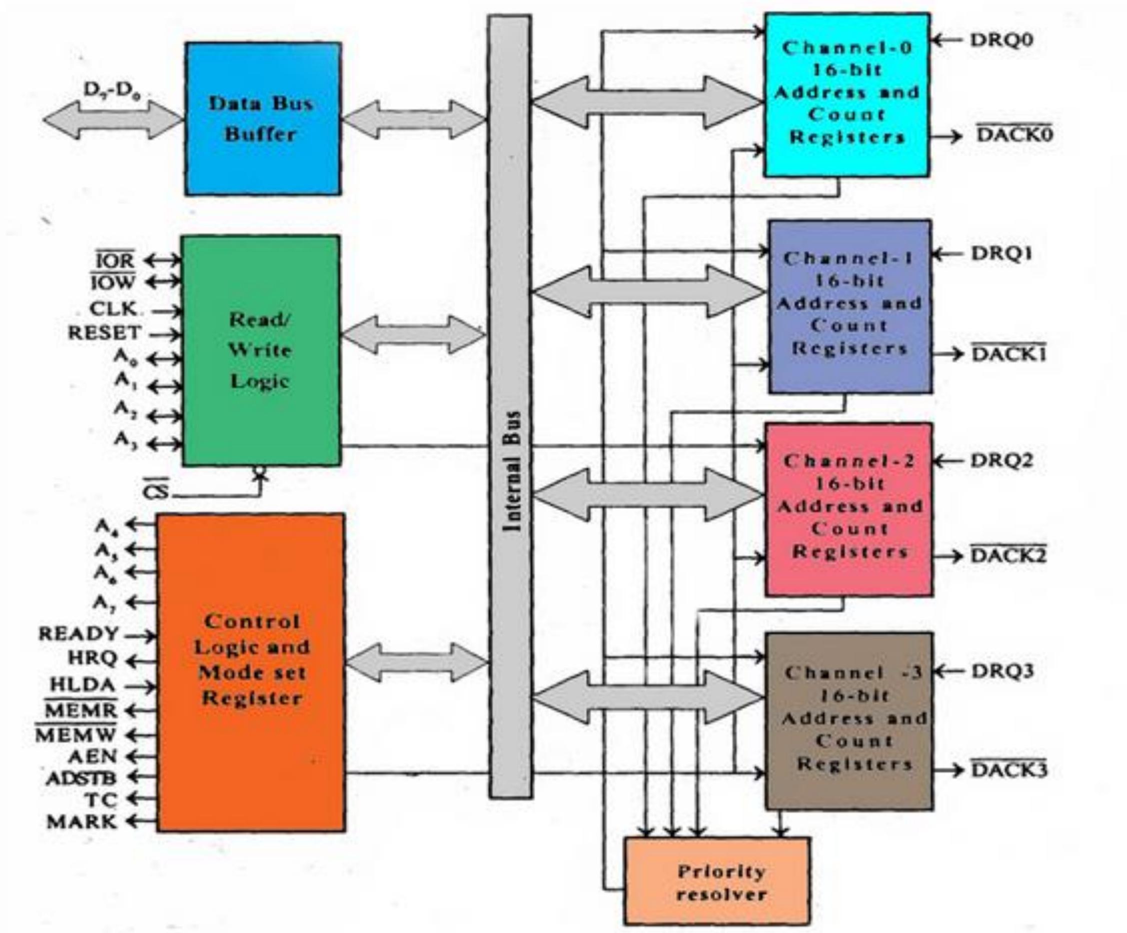
<u>DACK0-DACK3:</u>

- These are the active low DMA acknowledge output lines.
- Low level indicate that ,peripheral is selected for giving the information (DMA cycle).
- In master mode it is used for chip select.





BLOCKDIAGRAM OF 8257



Description

- **It containing Five main Blocks.**

1. **Data bus buffer**
2. **Read/Control logic**
3. **Control logic block**
4. **Priority resolver**
5. **DMA channels.**

DATA BUS BUFFER:

- **It contain tristate ,8 bit bi-directional buffer.**
- **Slave mode ,it transfer data between microprocessor and internal data bus.**

- Master mode ,the outputs A8-A15 bits of memory address on data lines (Unidirectional).

READ/CONTROL LOGIC:

- It control all internal Read/Write operation.
- Slave mode ,it accepts address bits and control signal from microprocessor.
- Master mode ,it generate address bits and control signal.

Control logic block:

- It contains ,

1. Control logic
2. Mode set register and
3. Status Register.

CONTROL LOGIC:

- Master mode ,It control the sequence of DMA operation during all DMA cycles.
- It generates address and control signals.
- It increments 16 bit address and decrement 14 bit counter registers.
- It activate a HRQ signal on DMA channel Request.
- Slave ,mode it is disabled.

MODE SET REGISTERS:

- It is a write only registers.
- It is used to set the operating modes.
- This registers is programmed after initialization of DMA channel.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
AL	TCS	EW	RP	EN_3	EN_2	EN_1	EN_0

- AL=1=Auto load mode

AL=0=Rotating mode

- TCS=1=Stop after TC (Disable Channel)
- TCS=0=Start after TC (Enable Channel)
- EW=1=Extended write mode
- EW=0=normal mode.
- RP=1=Rotating priority
- RP=0=Fixed priority.
- EN_3=1=Enable DMA CH-3
- EN_3=0=Disable DMA CH-3
- EN_2=1=Enable DMA CH-2
- EN_2=0=Disable DMA CH-2
- EN_1=1=Enable DMA CH-1
- EN_1=0=Disable DMA CH-1
- EN_0=1=Enable DMA CH-0
- EN_0=0=Disable DMA CH-0

STATUS REGISTERS:

- It is read only registers.
- It is tell the status of DMA channels

- **TC status bits are set when TC signal is activated for that channel.**
- **Update flag is not affected during read operation.**
- **The UP bit is set during update cycle . It is cleared after completion of update cycle.**

- **UP=Update flag**
- **UP=1=8257 executing update cycle**
- **UP=0=8257 executing DMA cycle**
- **TC_3=1=TC activated CH-3**
- **TC_3=0=TC activated CH-3**
- **TC_2=1=TC activated CH-2**
- **TC_2=0=TC activated CH-2**
- **TC_1=1=TC activated CH-1**
- **TC_1=0=TC activated CH-1**
- **TC_0=1=TC activated CH-0**
- **TC_0=0=TC activated CH-0**
- **The address of status register is $A_3A_2A_1A_0$=1000.**

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain with a neat diagram the operation of 8257 DMA controller.	Dec 2014	7

UNIT 4 I/O and Memory Interfacing Using 8085/8086

Lecture 8

8279-Programmable Keyboard and Display I/O Interface(RGPV JUNE-2013)

Intel's 8279 is a general purpose Keyboard Display controller that simultaneously drives the display of a system and interfaces a Keyboard with the CPU. The Keyboard Display interface scans the Keyboard to identify if any key has been pressed and sends the code of the pressed key to the CPU. It also transmits the data received from the CPU, to the display device.

Both of these functions are performed by the controller in repetitive fashion without involving the CPU. The Keyboard is interfaced either in the interrupt or the polled mode.In the interrupt mode, the processor is requested service only if any key is pressed, otherwise the CPU can proceed with its main task.

In the polled mode, the CPU periodically reads an internal flag of 8279 to check for a key pressure. The Keyboard section can interface an array of a maximum of 64 keys with the CPU. The Keyboard entries (key codes) are debounced and stored in an 8-byte FIFO RAM, that is further accessed by the CPU to read the key codes. If more than eight characters are entered in the FIFO (i.e. more that eight keys are pressed), before any FIFO read operation, the overrun status is set. If a FIFO contains a valid key entry, the CPU is interrupted (in interrupt mode) or the CPU checks the status (in polling) to read the entry.

Once the CPU reads a key entry, the FIFO is updated, i.e. the key entry is pushed out of the FIFO to generate space for new entries. The 8279 normally provides a maximum of sixteen 7-seg display interface with CPU It contains a 16-byte display RAM that can be used either as an integrated block of 16x8-bits or two 16x4-bit block of RAM. The data entry to RAM block is controlled by CPU using the command words of the 8279.

Architecture and Signal Descriptions of 8279

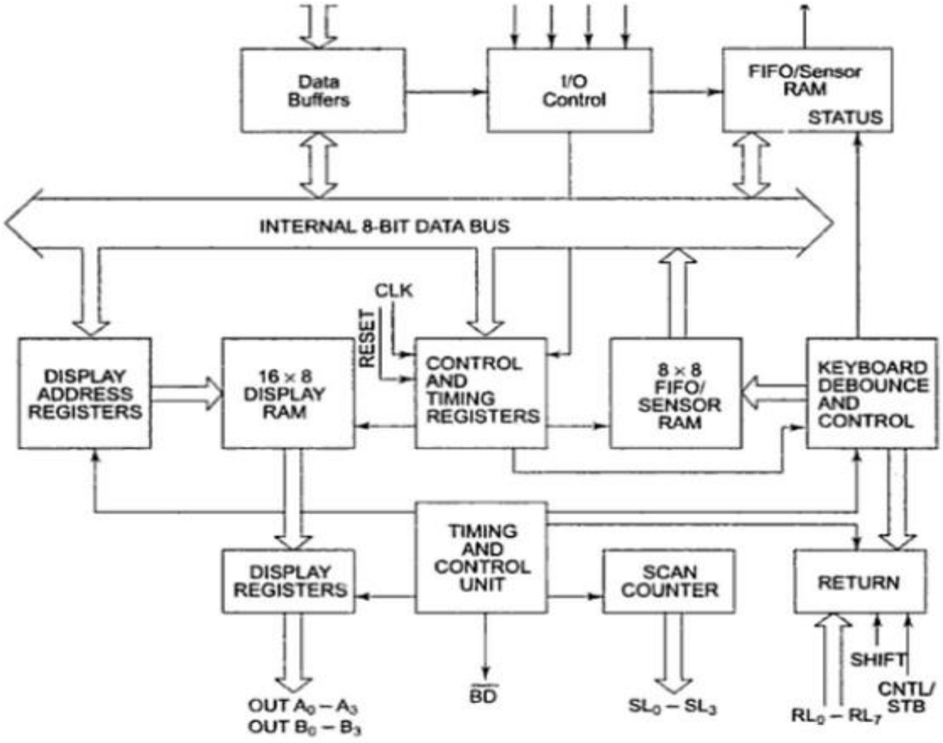
The Keyboard display controller chip 8279 provides

1. A set of four scan lines and eight return lines for interfacing keyboards.
2. A set of eight output lines for interfacing display.

I/O Control and Data Buffer

The I/O control section controls the flow of data to/from the 8279. The data buffer interface the external bus of the system with internal bus of 8279. the I/O section is enabled only if D is low.

8279 Internal Architecture



The pin Ao, RD and WR select the command, status or data read/write operations carried out by the CPU with 8279.

Control and Timing Register and Timing Control:-

These registers store the keyboard and display modes and other operating conditions programmed by CPU. The registers are written with Ao=1 and WR =0. The timing and control unit controls the basic timings for the operation of the circuit. Scan Counter divide down the operating frequency of 8279 to derive scan keyboard and scan display frequencies.

Scan Counter:-

The Scan Counter has two modes to scan the key matrix and refresh the display.In the Encoded mode, the counter provides a binary count that is to be externally decoded to provide the scan lines for keyboard and display (four externally decoded scan lines may drive up to 16 displays).

In the decoded scan mode, the counter internally decodes the least significant 2 bits and provides a decoded 1 out of 4 scan on SL0-SL3 (four internally decoded scan lines may drive up to 4 Displays). The Keyboard and Display both are in the same mode at a time.

Return Buffers and Keyboard Debounce and Control:-

This section scans for a Key closure row-wise. If it is detected, the Keyboard debounce unit debounces the key entry (i.e. wait for 10 ms). After the debounce period, if the key continues to be detected. The code of the Key is directly transferred to the sensor RAM along with SHIFT and CONTROL key status.

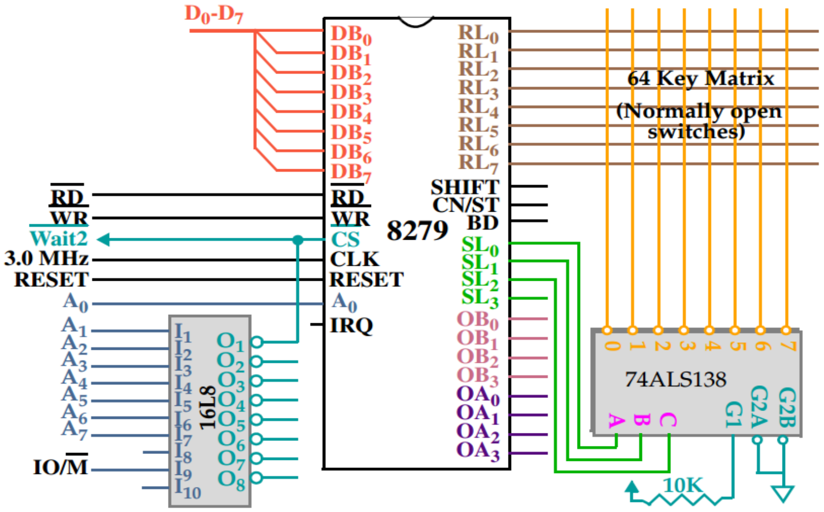
FIFO/Sensor RAM and Status Logic:-In Keyboard or strobed input mode, this block acts as 8-byte first-in-first-out (FIFO) RAM. Each key code of the pressed key is entered in the order of the entry, and in the meantime, read by the CPU, till the RAM becomes empty. The status logic generates an interrupt request after each FIFO read operation till the FIFO is empty.

In scanned sensor matrix mode, this unit acts as sensor RAM. Each row of the sensor RAM is loaded with the status of the corresponding row of sensors in the matrix. If a sensor changes its state, the IRQ line goes high to interrupt the CPU.

Display Address Registers and Display RAM.

The Display address registers hold the addresses of the word currently being written or read by the CPU to or from the display RAM. The contents of the registers are automatically updated by 8279 to accept the next data entry by CPU. The 16-byte display RAM contains the 16-byte of data to be displayed on the sixteen 7-seg displays in the encoded scan mode.

S.NO	RGPV QUESTIONS	Year	Marks
Q.1	Explain the different commands of 8279 in brief.	JUNE 2013	10



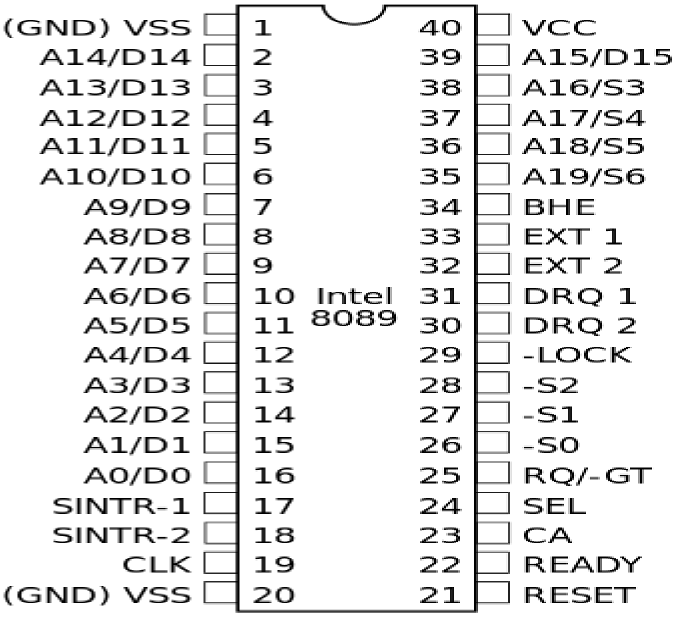
UNIT 4 I/O and Memory Interfacing Using 8085/8086

Lecture 9

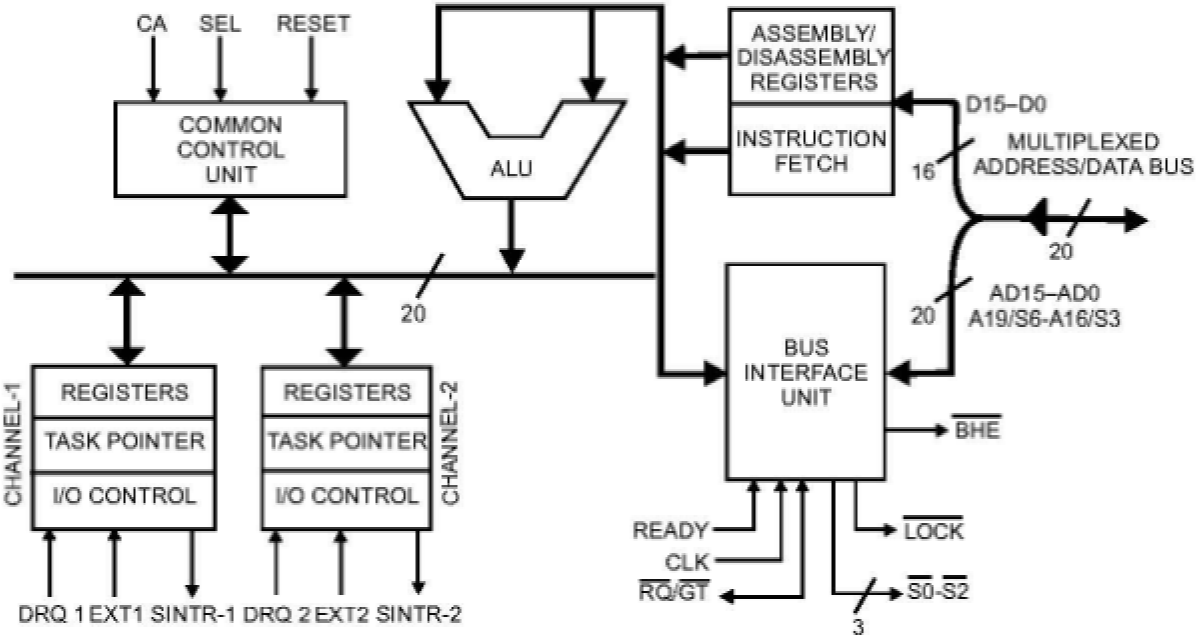
8089 I/O processor

Features of 8089

1. Very high speed DMA capability—I/O to memory, memory to I/O, memory to memory and I/O to I/O.
2. 1 MB address capability.
3. Supports local mode and remote mode I/O processing.
4. Allows mixed interface of 8-and 16-bit peripherals, to 8-and 16-bit processor buses.
5. Multibus compatible system interface.
6. Memory based communications with CPU.
7. Flexible, intelligent DMA functions, including translation, search, word assembly/disassembly.
8. Supports two I/O channels.



Pin diagram of 8089



8089 block diagram