Northern India Engineering College, Delhi

(GGSIP University)

SUBJECT – MICROPROCESSOR SYSTEMS SUBJECT CODE- ETCS-302

PAPER –I

Q1.Explain IVT?

ANS. interrupt vector table is a memory space for storing starting addresses of all the interrupt service routine. It stores CS:IP PAIR corresponding to each ISR. An **interrupt vector** is the memory address of an interrupt handler, and it is also an index into an array called an interrupt vector table that contains the memory addresses of interrupt handlers. When an interrupt is generated, the Operating System saves its execution state via a context switch, and begins execution of the interrupt handler at the interrupt vector.

Q2. What are the various interrupts in 8086? Explain.

Ans. Maskable interrupts, Non-Maskable interrupts.

i) An interrupt that can be turned off by the programmer is known as Maskable interrupt.

ii) An interrupt which can be never be turned off (ie.disabled) by the programmer is known as Non-Maskable interrupt.

Q3. What is an instruction queue? Explain?

Ans. This is introduced in 8086 processor. This queue is in the BIU and is used for storing the instructions. This will overlap the fetching and execution cycle. The EU will take the instructions from the queue for decoding and execution.

Q4. What is REP prefix? How it functions for string instructions?

Ans. This REP prefix is used for repeating. The instruction with REP prefix will execute repeatedly till the count in the cx register will be zero. This can be used in with some of the string handling instructions.

Q.5What is the difference between minimum mode and maximum mode of 8086 ?

Maximum mode	**Minimum Mode**
When MN/MX(bar) low 8086 is in maximum mode.	When MN/MX(bar) high 8086 is in minimum mode.
In maximum mode 8086 generates QS1,QS0,S0(bar),S1(bar),S2(bar), LOCK(bar),RQ(bar)/GT1,RQ(bar)/GT0 control signal	In minimum mode 8086 generates INTA(bar), ALE, DEN(bar), DT/R(bar), M/IO(bar), HLDA,HOLD and WR(bar) control signals.
So clearly there are multiple processors in the system.	There is only one processor in the system minimum mode.
Whereas in maximum mode interfacing, master/slave and multiplexing and several such control signals are required	In minimum mode no interfacing or master/slave signals is required.
In maximum mode a bus controller is required to produce control signals. This bus controller produces MEMRDC, MEMWRC, IORDC, IOWRC, ALE, DEN,	In minimum mode direct RD WR signals can be used. No bus controller required. A simple demultiplexer would do the job. of producing the control signals. This

DT/R control signals.	demultiplexer produces MEMRD, MEMWR, IORD, IOWR control signals.

Q6. Explain the instructions (i) LDS (ii) PUSHF (iii) TEST Ans.

i) LDS : load pointer to DS
Move a 32 bit content from the memory given as source to 16 bit destination register specified and to DS register.

ii) PUSHF : push the flag
After the execution the content of the flag register will be pushed to the stack. The higher byte to sp-1 and lower to sp-2

iii) TEST : logical comparison
This will compare the source and the destination specified. The result will be reflected only in the flag registers.

iv) CLD: this will clear the direction flag.

Q7. What is stack? Explain the use and operation of stack and stack pointer?

Ans. A stack is a portion of the memory used for the temporary storage. A stack is a last In first Out memory. A stack grows in the decreasing order. A stack will hold the temporary information’s push and pop are the instructions used for storing and accessing data from the stack. Contents can be moved as 16 bit only using push and pop instructions.

Q8. Explain different types of registers in 8086 microprocessor architecture.

Ans. Most of the registers contain data/instruction offsets within 64 KB memory segment. There are four different 64 KB segments for instructions, stack, data and extra data. To specify where in 1 MB of processor memory these 4 segments are located the processor uses four segment registers:

Code segment (CS) is a 16-bit register containing address of 64 KB segment with processor instructions. The processor uses CS segment for all accesses to instructions referenced by instruction pointer (IP) register. CS register cannot be changed directly. The CS register is automatically updated during far jump, far call and far return instructions.

Stack segment (SS) is a 16-bit register containing address of 64KB segment with program stack. By default, the processor assumes that all data referenced by the stack pointer (SP) and base pointer (BP) registers is located in the stack segment. SS register can be changed directly using POP instruction.

Data segment (DS) is a 16-bit register containing address of 64KB segment with program data. By default, the processor assumes that all data referenced by general registers (AX, BX, CX, DX) and index register (SI, DI) is located in the data segment. DS register can be changed directly using POP and LDS instructions.

Extra segment (ES) is a 16-bit register containing address of 64KB segment, usually with program data. By default, the processor assumes that the DI register references the ES segment in string manipulation instructions. ES register can be changed directly using POP and LES instructions.

It is possible to change default segments used by general and index registers by prefixing instructions with a CS, SS, DS or ES prefix.

All general registers of the 8086 microprocessor can be used for arithmetic and logic operations. The general registers are:

Accumulator register consists of 2 8-bit registers AL and AH, which can be combined together and used as a 16-bit register AX. AL in this case contains the low-order byte of the word, and AH contains the high-order byte. Accumulator can be used for I/O operations and string manipulation.

Base register consists of 2 8-bit registers BL and BH, which can be combined together and used as a 16-bit register BX. BL in this case contains the low-order byte of the word, and BH contains the high-order byte. BX register usually contains a data pointer used for based, based indexed or register indirect addressing.

Count register consists of 2 8-bit registers CL and CH, which can be combined together and used as a 16-bit register CX. When combined, CL register contains the low-order byte of the word, and CH contains the high-order byte. Count register can be used as a counter in string manipulation and shift/rotate instructions.

Data register consists of 2 8-bit registers DL and DH, which can be combined together and used as a 16-bit register DX. When combined, DL register contains the low-order byte of the word, and DH contains the high-order byte. Data register can be used as a port number in I/O operations. In integer 32-bit multiply and divide instruction the DX register contains high-order word of the initial or resulting number.

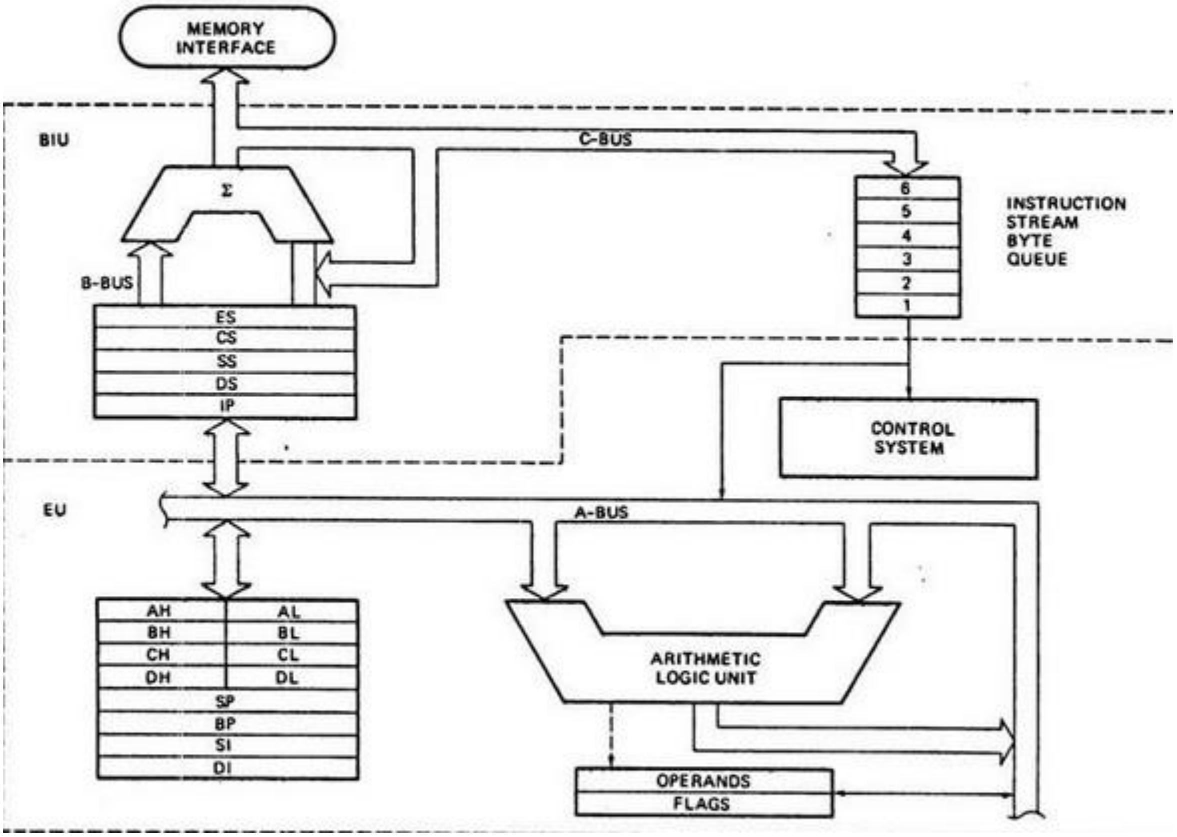
The following registers are both general and index registers:

Stack Pointer (SP) is a 16-bit register pointing to program stack.

Base Pointer (BP) is a 16-bit register pointing to data in stack segment. BP register is usually used for based, based indexed or register indirect addressing.

Source Index (SI) is a 16-bit register. SI is used for indexed, based indexed and register indirect addressing, as well as a source data addresses in string manipulation instructions.

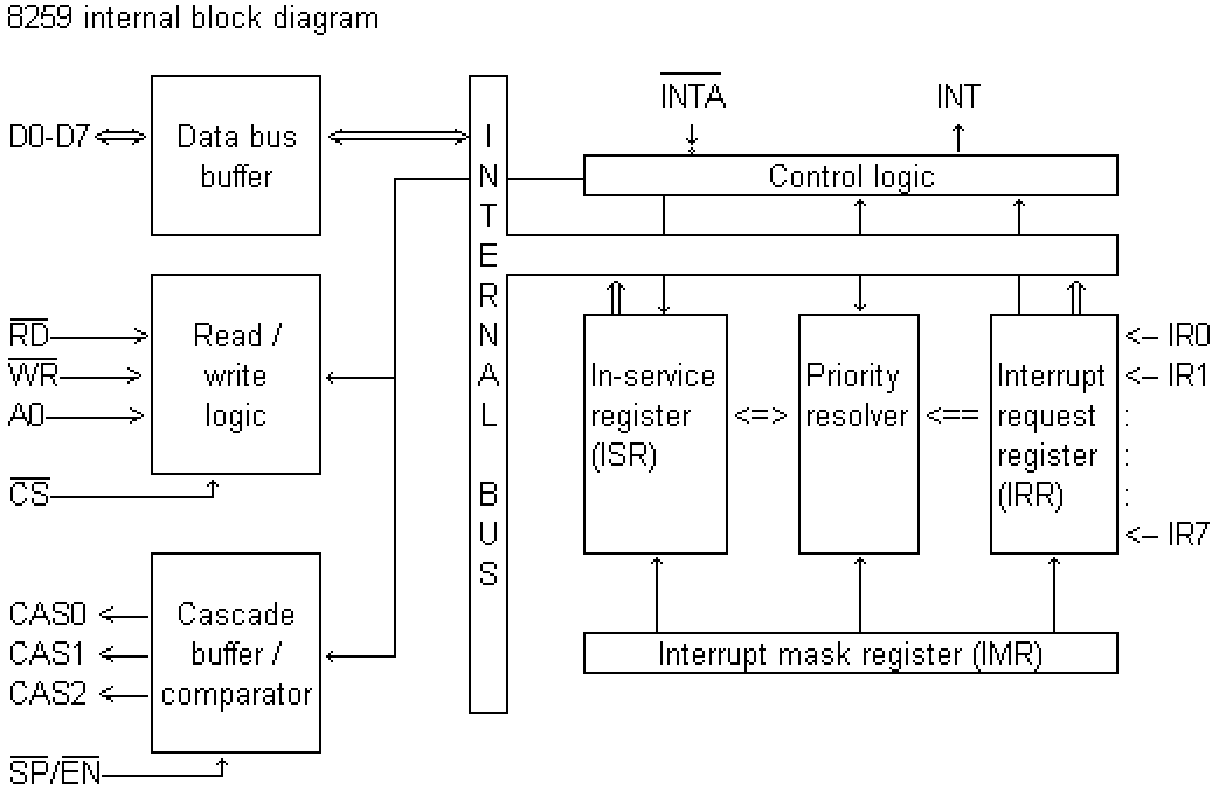
Destination Index (DI) is a 16-bit register. DI is used for indexed, based indexed and register indirect addressing, as well as a destination data addresses in string manipulation instructions.



Architecture of 8086

Q9) Explain architecture of Intel 8259 PIC.

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.


8259 internal block diagram
D0-D7
Data bus buffer
RD
WR
A0
Read / write logic
CS
CAS0
CAS1
CAS2
Cascade buffer / comparator
SP/EN
INTERNAL BUS
INTA
INT
Control logic
In-service register (ISR)
<=>
Priority resolver
<==
Interrupt request register (IRR)
<– IR0
<– IR1
<– IR7
Interrupt mask register (IMR)

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 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 VOH level on this line is designed to be fully compatible with the 8080A, 8085A and 8086 input levels.

INTA (Interrupt Acknowledge)

INTA pulses will cause the 8259A to release vectoring information onto the data bus. The format of this data depends on the system mode 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.

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.

WR (Write)

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

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.

A0

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

The Cascade Buffer/Comparator

This function block stores and compares the IDs of all 8259A's used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259A is used as a master and are inputs when the 8259A is used as a slave. As a master, the 8259A sends the ID of the interrupting slave device onto the CAS0±2 lines. The slave thus selected will send its preprogrammed subroutine address onto the Data Bus during the next one or two consecutive INTA pulses.