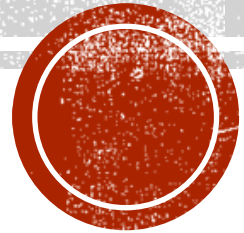
CHAPTER 5
INTERRUPTS



INTERRUPTS AND ITS NEED

- The processor can be interrupted in the following ways
  1. By an external signal generated by peripheral
  2. By an internal signal generated by a special instruction in the program
  3. By an internal signal generated due to an exceptional condition which occurs while executing an instruction
- In general, the process of interrupting the normal program execution to carry out a specific task/work is referred to as interrupt

- The interrupt is initiated by a signal generated by an external device or by a signal generated internal to the processor
- When a microprocessor receives an interrupt signal it stops executing current normal program, saves the status of various registers in stack and then the processor executes a subroutine/procedure in order to perform a specific task/work requested by an interrupt.
- The subroutine/procedure that is executed in response to an interrupt is also called Interrupt Service Subroutine (ISS)
- At the end of the ISS, the stored status of registers in stack are restored to respective registers, and the processor resumes the normal program execution from the point where it was interrupted

- The external interrupts are used to implement the interrupt driven data transfer scheme.
- The interrupts are generated by special instructions are called software interrupts and they are used to implement system service/calls.
- The system/monitor services are procedures developed by the system designer for various operations and stored in memory
- The user can call these services through software interrupts.
- The Interrupts generated by exceptional conditions are used to implement error conditions in the system

INTERRUPT DRIVEN DATA TRANSFER SCHEME

- The interrupts are useful for efficient data transfer between the processor and the peripheral
- When a peripheral is ready for data transfer, it interrupts the processor by sending an appropriate signal.
- Upon receiving an interrupt signal, the processor suspends the current program execution, saves the status in stack and executes an ISS to perform the data transfer between the peripheral and processor

- At the end of ISS the processor status is restored from stack and processor resumes its normal program execution.
- This type of data transfer scheme is called interrupt driven data transfer scheme
- The data transfer between the processor and the peripheral devices can be implements either by polling technique or by interrupt method

CLASSIFICATION OF INTERRUPTS

- Interrupts can be classified in the following three ways
    1. Hardware and software interrupts
    2. Vectored and non-vectored interrupts
    3. Maskable and non-maskable interrupts
- The interrupts initiated by external hardware by sending an appropriate signal to the interrupt pin of the processor is called hardware interrupt
- The 8086 processor has two interrupt pins: INTR and NMI
- The interrupts initiated by applying appropriate signal to these pins are called hardware interrupts of 8086

- Software interrupts are program instructions
- These instructions are inserted at desired locations in a program
- While running a program, if a software interrupt instruction is encountered, then the processor initiates an interrupt.
- The 8086 processor has 256 types of software interrupts
- The software interrupt instruction is INT n, where n is the type number in the range 255

- When an interrupt signal is accepted by the processor, if the program control automatically branches to a specific address(called vector address), then the interrupt is called vectored interrupt
- The automatic branching to a vector address is predefined by the manufacturer of processors.
- In non vectored interrupts, the interrupting device should supply the address of the ISS to be executed in response to the interrupt
- All the 8086 interrupts are vectored interrupts
- The vector address for an 8086 interrupt is obtained from a vector table implemented in the first 1Kb memory space

- The processors have the facility for accepting or rejecting hardware interrupts
- Programming the processor to reject an interrupt is referred to as masking and programming the processor to accept an interrupt is referred to as unmasking or enabling.
- In 8086, the Interrupt Flag (IF) can be set to one to unmask or enable all hardware interrupts and IF is cleared to zero to mask or disable all hardware interrupts except NMI

- The interrupts whose request can be either accepted or rejected by the processor are called maskable interrupts
- The interrupts whose request has to be definitely accepted by the processor are called non-maskable interrupts
- Whenever a request is made by non-maskable interrupt, the processor has to definitely accept that request and service that interrupt by suspending its current program and executing an ISS
- In an 8086 processor, all the hardware interrupts initiated through INTR pin are maskable by clearing Interrupt Flag (IF).
- The interrupt initiated through NMI pin and all software interrupts are non-maskable.

SOURCES OF INTERRUPTS IN 8086

- An Interrupt in 8086, can come from one of the following 3 sources
  - One source is from an external signal applied to NMI/INTR input pin of the processor. The interrupts initiated by applying appropriate signals to these input are called hardware interrupts
  - A Second source of an interrupt is execution of the interrupt instruction "INT n", where n is the type number. The interrupts initiated by 'INT n' instructions are called software interrupts
  - The third source of an interrupt is from some condition produced in the 8086 by the execution of an instruction. Divide by zero is an example. Such conditional interrupts are also known as exceptions

INTERRUPTS OF 8086

- 8086 has 256 interrupts, numbered 0-255
- The interrupts can be initiated wither by executing 'INT n' or can be initiated by sending an appropriate signal to INTR input pin of the processor
- When the interrupt is initiated through INTR pin, then the processor runs an interrupt acknowledge cycle to get the type number
- INTEL has defined the functions of the first 5 interrupts

INTEL PREDEFINED INTERRUPTS

- Division by zero(Type-0) interrupt.
- Single step(Type-1) interrupt.
- Non-maskable interrupt, NMI(Type-2) interrupt.
- Breakpoint(Type-3) interrupt.
- Interrupt on overflow(Type-4) interrupt.

- The predefined interrupts are only defined by INTEL and INTEL has not provided any subroutine or procedure to be executed for these interrupts.
- To use the predefined interrupts the user/system designer has to write Interrupt Service Subroutine (ISS) for each interrupt and store them in memory.
- The corresponding address of the ISS should be stored in interrupt vector table.
- If a predefined interrupt is not used in a system then the user may assign some other functions to these Interrupts.

DIVIDE BY ZERO(TYPE-0) INTERRUPT

- Type-0 interrupt is implemented by INTEL as a part of the execution of the divide instruction.
- The 8086 will automatically do a type-0 interrupt if the result of a division operation is too large to fit in the destination register and this interrupt is non-maskable.
- Since the type-0 interrupt cannot be disabled in any way, we have to account for it in the programs using divide instructions.
- To account for this, we have to write an ISS which takes the desired action or indicate error condition when an invalid division occurs.
- The ISS should be stored in memory and the address of ISS is stored in interrupt vector table.

SINGLE STEP(TYPE-1) INTERRUPT

- When the Trap/Trace Flag (TF) is set to one, the 8086 processor will automatically generate a type-1 interrupt after execution of each instruction.
- The user can write an ISS for type-1 interrupt to halt the processor temporarily and return the control to the user so that after execution of each instruction, the processor status (content of register/memory) can be verified.
- If they are correct then we can proceed to execute the next instruction.
- Execution of one instruction by one instruction is known as single step and this feature will be useful to debug a program.

NON-MASKABLE(TYPE-2) INTERRUPT

- The 8086 processor will automatically generate a type-2 interrupt when it receives a low-to-high transition on its NMI input pin.
- This interrupt cannot be disabled or masked.
- Usually, the type-2 interrupt is used to save program data or processor status in case of system AC power failure.

BREAKPOINT(TYPE-3) INTERRUPT

- Type-3 interrupt is used to implement a breakpoint function, which executes a program partly or up to the desired point and then return the control to the user.

  The breakpoint interrupt is initiated by the execution of "INT 3" instructions.
- To implement the breakpoint function the system designer has to write an ISS for type-3, which takes care of displaying a message and return the control to the user whenever type-3 interrupt is initiated.
- This interrupt will be useful to debug a program by executing the program part by part.

OVERFLOW (TYPE-4) INTERRUPT

- In the 8086 processor, the Overflow Flag (OF) will be set if the signed arithmetic operation generates such a result whose size is larger than the size of the destination register/memory.
- During conditions, the type-4 interrupt can be used to indicate an error condition.
- The type-4 interrupt is initiated by "INTO" instruction.
- One way of detecting the overflow error is to put the "INTO" instruction immediately after the arithmetic instruction in the program.
- After arithmetic operation if the overflow flag is not set then the processor will consider "INTO" instruction as NOP (No operation).
- However, if the overflow flag is set then the 8086 will generate a type-4 interrupt, which executes an-ISS to indicate overflow condition.

PRIORITIES OF INTERRUPTS OF 8086

Interrupt	Priority
Divide error, INT n, INTO	highest
NMI	.
INTR	.
SINGLE STEP	lowest

IMPLEMENTING INTERRUPT SCHEME IN 8086

- 8086 has 256 types of interrupts and these can be implemented either as hardware or software
- Except some of the INTEL predefined interrupts, for all other interrupts, the system designer has to decide on the method of initiating the interrupts selected to implement on a system
- The interrupts can be initiated either by external hardware or internally by software instructions 'INT n'

INTERRUPT VECTOR TABLE

- For each and every interrupt, the system designer has to write an Interrupt Service Subroutine (ISS) and store them in memory.
- Then the system designer has to create an interrupt vector table in the first 1kb memory space.
- In this vector table, the 16-bit offset address and 16-bit segment base address of each ISS are stored in four consecutive memory locations.
- The address stored in this table are called vector address

- The memory address for storing the vector address for an interrupt is given by multiplying the type number by four and sign extending it to 20 bit
- The vector address is stored in four consecutive memory location starting from this 20-bit address
- The first two locations are used to store the low byte and high byte of offset address, and next two locations are used to store low byte and high byte of segment base address of ISS to be executed for an interrupt.

SERVICING AN INTERRUPT BY 8086

- The 8086 processor checks for interrupt request at the end of each instruction cycle.
- If an interrupt is detected, it performs the following actions

1. The SP is decremented by two and the content of flag register is pushed to stack memory
2. The interrupt system is disabled by clearing Interrupt Flag (IF)
3. The single step trap flag is disabled by clearing Trap Flag (TF)

4. The stack pointer is decremented by two and the content of CS register is pushed to stack memory
5. Again, the stack pointer is decremented by two and the content of IP is pushed to stack memory
6. In case of hardware interrupt through INTR, the processor runs an interrupt acknowledge cycle to get the interrupt type number. For software interrupt, the type number is specified in the instruction itself. For NMI and exceptions the type number is defined by INTEL
7. The processor generates a 20-bit memory address by multiplying the type number by four and sign extending it to 20 bit. This memory address is the address of the IVT, where the vector address of the ISS is stored by the system designer

8. The first word pointed by vector table address is loaded in IP and the next word is loaded in CS register. Now the content of the IP is the offset address and the content of the CS register is the segment base address of the ISS to be executed
9. The 20 bit physical memory address of ISS is calculated by multiplying the content of CS register by 16 and adding it to the content of IP
10. The processor executes the ISS to service the interrupt
11. The ISS will be terminated by the IRET instruction. When this instruction is executed, the top of stack is popped to IP, CS and flag register one word by one word. After every pop operation, the SP is incremented by two
12. Thus, at the end of ISS, the previous status of the processor is restored and so the processor will resume the execution of normal program from the instruction where it was suspended

INTR AND ITS EXPANSION

- The hardware interrupt INTR can be used by any external device to interrupt the processor.
- When an interrupt request is made through the INTR, interrupt is enabled/unmasked, then the processor will run an interrupt acknowledge cycle.
- During this cycle, the processor asserts INTA' signal twice.
- The first INTA' signal is to inform the interrupting device about the acceptance of the interrupt.
- The second time, INTA' is asserted to request the interrupting device to supply the type number and to read the type number from the low order data bus.
- Therefore, the processor expects a type number on the low order data bus whenever it is interrupted through the INTR input pin

PROGRAMMABLE INTERRUPT CONTROLLER-INTEL 8259

- It is used to expand the interrupts of the 8086 processor
- One 8259 can accept eight interrupt requests and allow one by one to the processor INTR pin.
- The interrupt controller can be used in cascaded mode in a system to expand the interrupts up to 64

FEATURES OF 8259

- It is programmed to work either 8085 or 8086 processor
- It manages 8 interrupts according to the instructions written into its control registers
- The interrupt vector address is programmable
- The priorities of the interrupts are programmable.
- The different operating modes which decide the priorities in automatic rotation mode, specific rotation mode and fully nested mode

- The interrupts can be masked or unmasked individually
- 8259 is programmed to accept either level triggered interrupt signal or edge triggered interrupt signal
- 8259 provides the status of the pending requests, masked interrupts and the interrupt being serviced
- 8259 can be cascaded to accept a maximum of 64 interrupts