Ch 12 : Interrupts (upto 475)

12.1 Basic Interrupt Processing

- *Purpose of Interrupts*
  - When interfacing I/O devices at relatively low data transfer rates
  - allows execution of other software while waiting for input
  - To avoid Polling
    - not good because has to individually check for each process
  - the process that needs CPU can interrupt on its own unlike polling
- *Interrupts*
  - INTR and NMI request interrupts
  - INTA overbar acknowledges interrupts
  - Software interrupts INT, INTO, INT3 and BOUND
  - IF(interrupt flag) and TF (trap flag)are also used with IRET
  - Interrupt Vectors:
    - Interrupt vector table is located in the first 1024 bytes of memory at addresses 000000H–0003FFH.
    - 256 different four-byte interrupt vectors.
    - contain the address (segment and offset) of the interrupt service procedure.
  - Dedicated Interrupts:
    - TYPE 0: divide error on result overflow from a division or divide by zero
    - TYPE 1: Single step or trap, occurring after execution of each instruction if the trap (TF)flag bit is set. TF cleared on accepting the interrupt
    - TYPE 2: non-maskable interrupt occurs when a logic 1 is placed on the NMI input pin to the microprocessor. This cannot be disabled.
    - *More types on pdf page 474-475*
- *Interrupt Instructions: BOUND, INTO, INT, INT3, IRET*
  - BOUND compares a register with two words of memory data
  - INTO tests the O flag (calls interrupt type number 4 if O=1)
  - INT n calls the ISP at address in vector number n
    - e.g. INT 80H or INT 128 calls ISP at 00200H - 00203H
    - Ch6 for rest
  - INT 3 is breakpoint interrupt. to debug
  - IRET is used to return
- *Operation of a Real Mode Interrupt(RMI)*
  - Interrupt's activeness is determined by checking the following in the order presented:
    - instruction executions,
    - single-step,
    - NMI,
    - coprocessor segment overrun,
    - INTR, and
    - INT instructions
  - If active, following happens:
    - push contents of flag register
    - IF and TF cleared to disable INTR pin
    - push contents of CS
    - push contents of IP
    - fetch interrupt vector contents and place into IP and CS
  - Interrupt type numbers 0, 5, 6,7, 8, 10, 11, 12, and 13 push a return address that points to the offending instruction, instead of to the next instruction in the program.
    - This may lead to recalling of that instruction
- *Operation of a Protected Mode Interrupt(PMI)*
  - The interrupt descriptor table is located at any memory location in the system by the interrupt descriptor table address register (IDTR).
  - IDT contains Segment Selector and OA and P bit(present) and DPL bits
  - RMI can be converted to PMI by copying interrupt procedure vectors from vector table and converting them to 32-bit OA
    - this descriptor will be in global descriptor table that identifies first 1M byte of memory
- *Interrupt Flag Bits*

    - When the IF bit is set, it allows the INTR pin to cause an interrupt;
        - when cleared, it prevents the INTR pin from causing an interrupt.
        - by STI and CLI
    - When TF = 1, it causes a trap interrupt (type number 1) to occur after each instruction executes.
        - This is why we often call trap a single-step.
        - When TF = 0, normal program execution occurs.
        - This flag bit allows debugging
        - no special instruction to set or clear
- *Storing an Interrupt Vector in the Vector Table*
    - hook: interrupt vector
    - installing requires addressing absolute memory
    - IRET instruction before ENDP is sometimes required because the assembler has no way of determining if the FAR procedure is an interrupt procedure.
    - Normal FAR procedures do not need a return instruction, but an interrupt procedure does need an IRET.
    - Interrupts must always be defined as FAR.

12.2 Hardware Interrupts

- NMI and INTR
- Whenever the NMI input is activated, a type 2 interrupt occurs because NMI is internally decoded.
- The INTR input must be externally decoded to select a vector.
- Any interrupt vector can be chosen for the INTR pin, usually between 20H and FFH
- interrupts 00H through 1FH reserved for internal and future expansion
- The INTA signal is also an interrupt pin on the microprocessor, but it is an output thatis used in response to the INTR input to apply a vector type number to the data bus connectionsD7–D0.
- NMI (Non-maskable Interrupt):
    - edge-triggered input that requests an interrupt on the positive edge (0-to-1 transition).
    - After a positive edge, the NMI pin must remain a logic 1 until it is recognized by the microprocessor
    - used for parity errors and other major system faults
        - Power failures are easily detected by monitoring the AC power line and causing an NMI interrupt whenever AC power drops out.
        - if this happens, the mp stores all of the internal register in a battery-backed-up memory or an EEPROM
- INTR and INTA:
    - INTR is level sensitive i.e. it must be held at logic 1 till recognised
    - set by an external event and cleared inside the ISP
    - automatically disabled after acceptance
    - re-enabled by IRET
    - mp responds by pulsing INTA output in anticipation of receiving an interrupt vector type number on data bus connections D7–D0.
    - *least expensive implementation on pdf page 481-482*
    - Can use a 3 state buffer for INTA
    - Can make the INTR Input Edge-Triggered by using a D-type flip-flop
- the 82C55 Keyboard Interrupt:
    - *pdf page 484*
    - The procedure is short because the microprocessor already knows that keyboard data are available when the procedure is called.
    - Data are input from the keyboard and then stored in the FIFO (first-in, first-out) buffer or queue.
    - *Full condition in Example 12-5*
    - *Removal of data in Example 12-6*

12.3 Expanding the Interrupt Structure

- *Using the 74ALS244 to expand interrupts*
    - diagram explained at pdf page 485-486
- *Daisy-Chained interrupts*
    - better than previous because it requires only one interrupt vector
    - priority determined by ISP
        - requires additional software time
    - fig 12-14 on pdf page 487

12.4 8259A Programmable Interrupt Controller

- Adds eight vectored priority encoded interrupts to the mp
- can be expanded, without additional hardware, to accept up to 64 interrupt requests.
- *General Description*
  - Easy to connect to mp because direct pin connection except CS(to be decoded) and WR (to have I/O bank write pulse)
  - Pin descriptions :
    - D0–D7: The bidirectional data connections are normally connected to the data bus on the microprocessor.
    - IR0–IR7:Interrupt request inputs are used to request an interrupt and to connect to a slave in a system with multiple 8259As.
    - WR: The write input connects to write strobe signal (IOWC overbar) on the mp.
    - RD: The read input connects to the IORC overbar signal.
    - INT: The interrupt output connects to the INTR pin on the microprocessor from the master and is connected to a master IR pin on a slave.
    - INTA: Interrupt acknowledge is an input that connects to the INTA signal on the system. In a system with a master and slaves, only the master INTA signal is connected.
    - A0: The A0 address input selects different command words within the 8259A.
    - CS: Chip select enables the 8259A for programming and control.
    - SP/EN: Slave program/enable buffer is a dual-function pin. When the 8259A is in buffered mode, this is an output that controls the data bus transceivers in a large microprocessor-based system. When the 8259A is not in the buffered mode, this pin programs the device as a master (1) or a slave (0).
    - CAS0–CAS2: The cascade lines are used as outputs from the master to the slaves for cascading multiple 8259As in a system.
- *Connecting to a single 8259A*
  - pdf page 489
- *Cascading multiple 8259As*
  - pdf page 489
- *Programming the 8259A*
  - programmed by initialization and operation command words.
  - Initialization command words(ICWs) are programmed before the 8259A is able to function in the system and dictate the basic operation of the 8259A.
  - Operation command words(OCWs) are programmed during the normal course of operation. They control the operation of the 8259A.
  - ICWs:
    - 4 are selected when A0 is 1
    - ICW1, 2 and 4 sent on power up
    - if in cascade mode then also ICW3
    - ICW1: programs basic operation of 8259A and selects single or cascade operation by programming the SNGL bit.
    - ICW2: Selects the vector number used with the interrupt request inputs
    - ICW3: Only used when ICW1 indicates that the system is operated in cascade mode. It indicates where the slave is connected to the master
    - ICW4: in 8086-Pentium 4.
      - rightmost bit selects operation (must be 1)
      - SFNM Selects the special fully nested mode of operation. allows priority requests
      - BUF and M/S are used together to select buffered operation or non buffered operation
      - AEOI Selects automatic or normal end of interrupt
        - The EOI commands of OCW2 are used only if the AEOI mode is not selected by ICW4.
        - If selected, the interrupt automatically resets the interrupt request bit and does not modify priority. This is the preferred mode of operation and reduces the length of the ISP
  - OCWs:
    - used to direct the operation of the 8259A once it is programmed with the ICW.
    - OCW1: Used to set and read the interrupt mask register. must be programmed after programming the ICW upon initialization.
    - OCW2: Programmed only when the AEOI mode is not selected for the 8259A. selects the way that the 8259A responds to an interrupt. Modes are:
      - Non specific end of interrupt(NSEOI): A command sent by the interrupt service procedure to signal the end of the interrupt.
        - automatically determines which interrupt level was active and resets the correct bit of the interrupt status register.
        - Resetting the status bit allows the interrupt to take action again or a lower priority interrupt to take effect.
      - Specific End of Interrupt(SEOI): command that allows a specific interrupt request to be reset.
      - Rotate-on-Nonspecific EOI(RoNSEOI): A command that functions exactly like the NSEOI except that it rotates interrupt

priorities after resetting the interrupt status register bit.
        - The level reset by this command becomes the lowest priority interrupt.
    - Rotate-on-Automatic EOI(RoAEOI)—A command that selects automatic EOI with rotating priority.
    - Rotate-on-Specific EOI (RoSEOI) —Functions as the specific EOI, except that it selects rotating priority.
    - Set priority—Allows the programmer to set the lowest priority interrupt input
- OCW3: Selects the register to be read, the operation of the special mask register, and the poll command.
    - The rightmost three bits of the poll word indicate the active interrupt request with the highest priority.
    - The leftmost bit indicates whether there is an interrupt and must be checked to determine whether the rightmost three bits contain valid information.
    - Bit positions D0 and D1of OCW3 select which register (IRR or ISR) is read when A0= 0.

- Status Register:
    - interrupt request register(IRR): 8 bit. indicates which interrupt request inputs are active. read by OCW3(A0=0)
    - In-service register(ISR): 8 bit. contains the level of interrupt being serviced. read by OCW3(A0=0)
    - Interrupt mask register(IMR): 8 bit. holds the interrupt mask bits and indicates which interrupts are masked off. read by OCW1(A0=1)