Short Notes Chapter 8 (Lecture 22-23):

Interrupts are asynchronous and unpredictable.

Asynchronous means that the interrupts occur, independent of the working of the processor, i.e. independent of the instruction currently executing.

Synchronous events are those that occur side by side with another activity.

Why interrupts must be asynchronous?

Interrupts must be asynchronous as they are generated by the external world which is unaware of the happenings inside the processor. True interrupts that occur in real time are asynchronous with the execution. Also, it is unpredictable at which time an interrupt will come.

Unpredictable means the time at which an interrupt will come cannot be predicted, while asynchronous means that the interrupt has nothing to do with the currently executing instruction and the current state of the processor.

The 8088 processor divides interrupts into two classes:

- Software interrupts and
- Hardware interrupts

Hardware interrupts are the real interrupts generated by the external world while Software interrupts on the contrary are not generated from outside the processor

The interrupt call loads new values in **CS, IP, and FLAGS.**

How an interrupt is generated?

To generate an interrupt the INT instruction is used. The routine that executes in response to an INT instruction is called the interrupt service routine (ISR) or the interrupt handler.

What is hooking and interrupt?

The software interrupt mechanism in 8088 uses vectored interrupts meaning that the address of the interrupt routine is not directly mentioned in an interrupt call, rather the address is lookup up from a table. 8088 provides a mechanism for mapping interrupts to interrupt handlers. Introducing a new entry in this mapping table is **called hooking an interrupt.**

Syntax of the INT instruction: is very simple. It takes a single byte argument varying from 0-255.

A few interrupt numbers in the start are reserved and we generally do not use them. They are related to the processor working.

For example: INT 0 is the divide by zero interrupt.

Interrupt Vector table:

The correlation process from the interrupt number to the interrupt handler uses a table **called interrupt vector table**.

Location & Structure of IVT:

Its location is fixed to physical memory address zero. Each entry of the table is four bytes long containing the segment and offset of the interrupt routine for the corresponding interrupt number. The first two bytes in the entry contain the offset and the next two bytes contain the segment.

INT 1 vector occupies location 4, 5, 6, and 7 and similarly vector for INT 2 occupies locations 8, 9, 10, and 11

Reserved Interrupts:

INT 0, Division by zero

Meaning the quotient did not fit in the destination register. INT 0 is automatically generated by a DIV when a specific situation arises, there is no INT 0 instruction.

INT 1, Trap, Single step Interrupt

This interrupt is used in debugging with the trap flag. If the trap flag is set the Single Step Interrupt is generated after every instruction.

By hooking this interrupt, a debugger can get control after every instruction and display the registers etc. 8088 was the first processor that has this ability to support debugging.

INT 2, NMI-Non Maskable Interrupt

Real interrupts come from outside the processor. INT 0 is not real as it is generated from inside. For real interrupts there are two pins in the processor, the INT pin and the NMI pin. The processor can be directed to listen or not to listen to the INT pin.

The NMI pin which informs about fatal hardware failures in the system and is tied to interrupt 2. INT pin can be masked but NMI cannot be masked.

INT 3, Debug Interrupt

The only special thing about this interrupt is that it has a single byte opcode and not a two-byte combination where the second byte tells the interrupt number. This allows it to replace any instruction whatsoever. It is also used by the debugger.

INT 4, Arithmetic Overflow, change of sign bit

The overflow flag is set if the sign bit unexpectedly changes as a result of a mathematical or logical instruction.

However, the overflow flag signals a real overflow only if the numbers in question are treated as signed numbers.

So, this interrupt is **not automatically generated** but as a result of a special instruction INTO (interrupt on overflow) if the overflow flag is set. Otherwise the INTO instruction behaves like a NOP (no operation).

BIOS AND DOS INTERRUPTS

Since the manufacturer knows the hardware, it burns the software to control its hardware in ROM. Such software is called firmware and access to this firmware is provided through specified interrupts. This basic interface to the hardware is called **BIOS (basic input output services).**

BIOS has the responsibility of testing the basic hardware including video, keyboard, floppy drive, hard drive etc. and a special program to bootstrap.

Bootstrap means to load OS from hard disk and from there OS takes control and proceeds to load its components and display a command prompt in the end.

There are two important programs;

- BIOS and
- OS

BIOS AND OS COMPARISON:

- OS services are high level and build upon the BIOS services.
- BIOS services are very low level.
- BIOS services provide a hardware independent layer above the hardware and OS services provide another higher-level layer over the BIOS services.
- The layer of BIOS provides services like display a character, clear the screen, etc.
- The video services are accessible through INT 10. Similarly keyboard services are available at INT 16
- BIOS exports its various services through different interrupts.
- Keyboard services are exported through INT 16,
- Parallel port services through INT 17
- DOS has a single-entry point through INT 21 just like a pin hole camera, this single-entry point leads to a number of DOS services.

How one interrupt provides a number of different services? OR

What is the concept of service number?

A concept of service number is a defecto standard in providing multiple services through an interrupt.

For example:

INT 10 is for video services and each of character printing service, screen clearing service, cursor movement service etc. has a service number associated to it. So, we say INT 10 service 0 is used for this purpose and INT 10 service 1 is used for that purpose etc. Service numbers for different standard services are also fixed for every IBM compatible.

The service number is usually given in the AH register, while the sub-service number is in AL or BL.

