Computer Architecture
Computer Components

Moniruzzaman

Adjunct Lecturer
North Western University, Khulna-9000

Confession

- Most of the materials have been collected from Internet.
- Images are taken from Internet.
- Various books are used to make these slides.
- Various slides are also used.
- References & credit:
  - Atanu Shome, Assistant Professor, CSE, KU.
  - Computer Organization and Design: the Hardware/Software Interface - Textbook by David A Patterson and John L. Hennessy.
  - Computer Organization and Architecture - Book by William Stallings

Programming

Program:
A set of Instructions

For each step, an arithmetic or logical operation is done

For each operation, a different set of control signals is needed

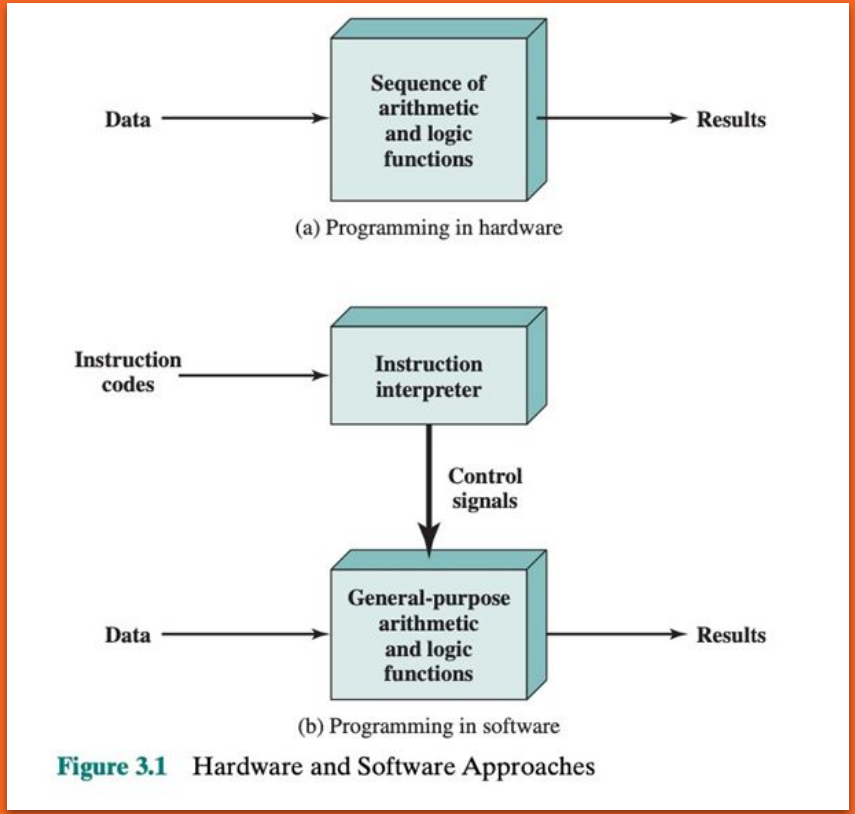
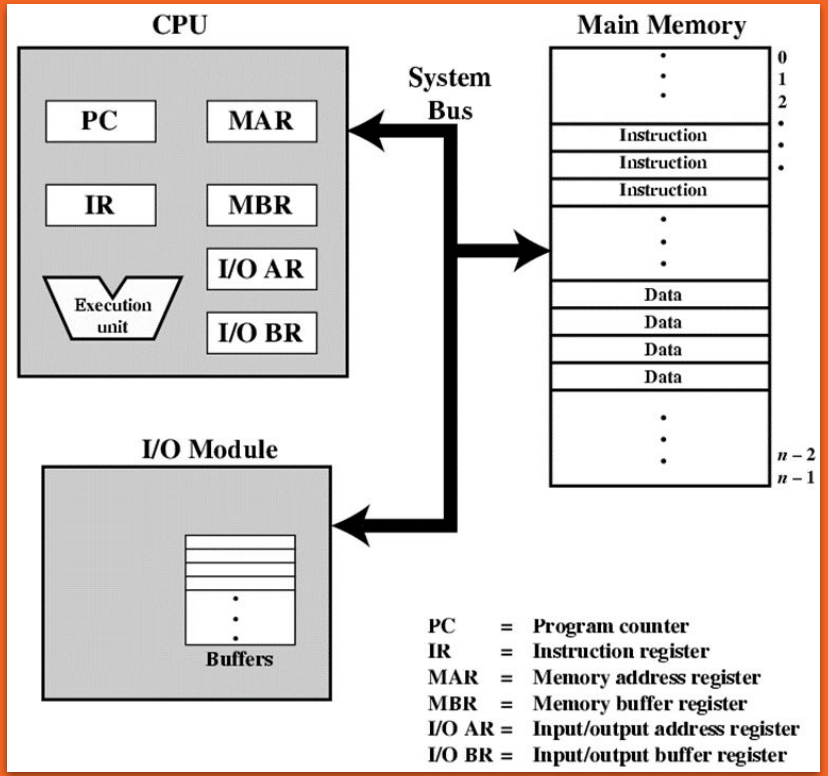
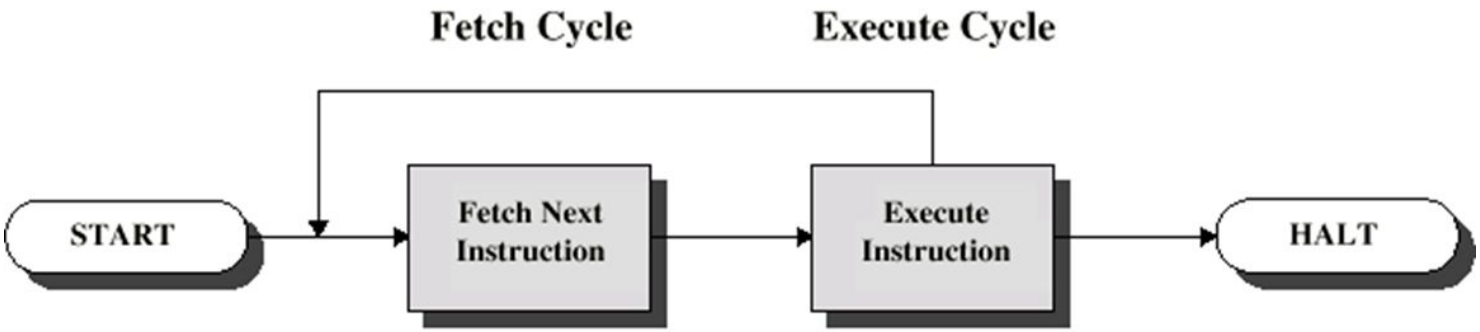


Figure 3.1 Hardware and Software Approaches

Top-Level View



Instruction Cycle



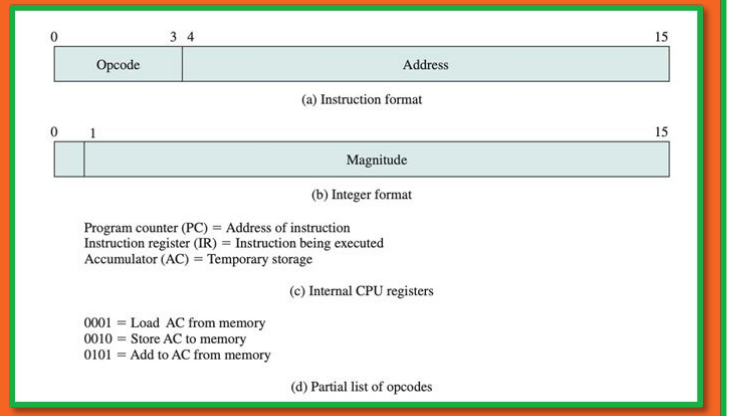
Fetch Cycle

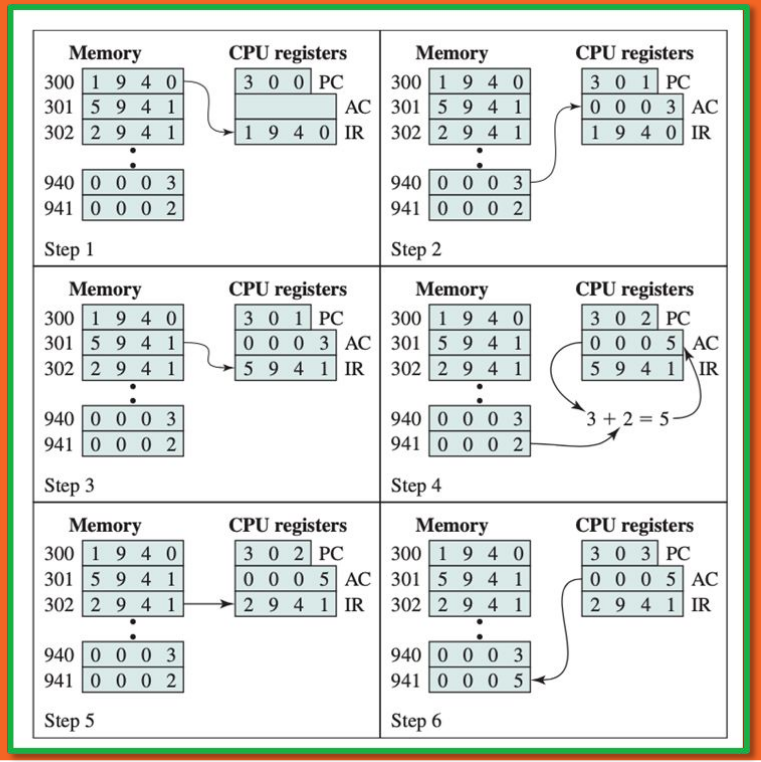
- Program Counter (PC) holds address of next instruction to fetch
- Processor fetches instruction from memory location pointed to by PC
- Increment PC
  - Unless told otherwise (loop, if)
- Instruction loaded into Instruction Register (IR)
- Processor interprets instruction and performs required actions

Execute Cycle

- Processor-memory
  - Data transfer between CPU and main memory
- Processor I/O
  - Data transfer between CPU and I/O module
- Data processing
  - Some arithmetic or logical operation on data
- Control
  - Alteration of sequence of operations
  - e.g. jump
- Combination of above

Example of Program Execution





Problem ???

The hypothetical machine has two I/O instructions:

0011 = Load AC from I/O

0111 = Store AC to I/O

In these cases, the 12-bit address identifies a particular I/O device. Show the program execution for the following program:

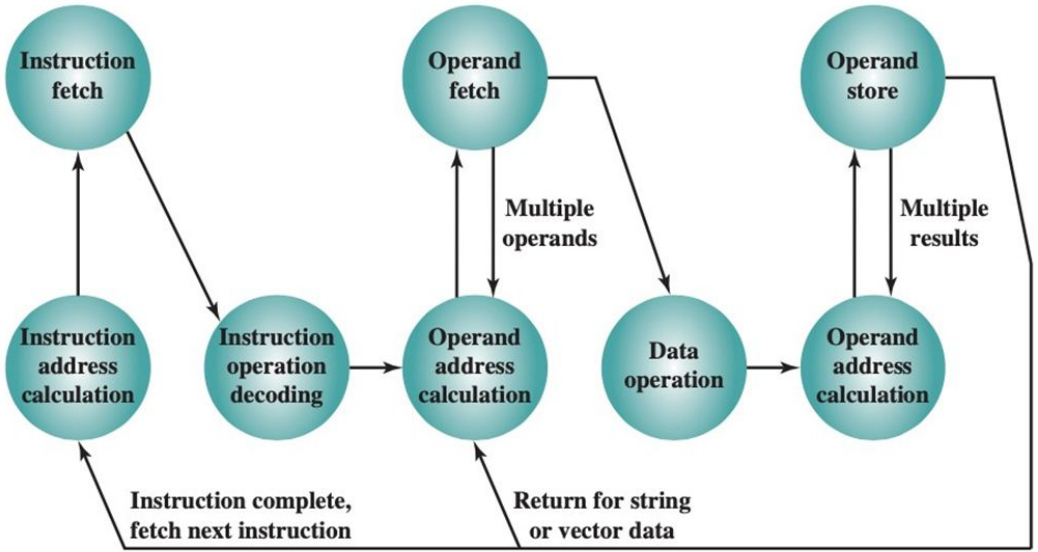
Load AC from device 5.

Add contents of memory location 940.

Store AC to device 6.

Assume that the next value retrieved from device 5 is 3 and that location 940 contains a value of 2.

Instruction Cycle - State Diagram

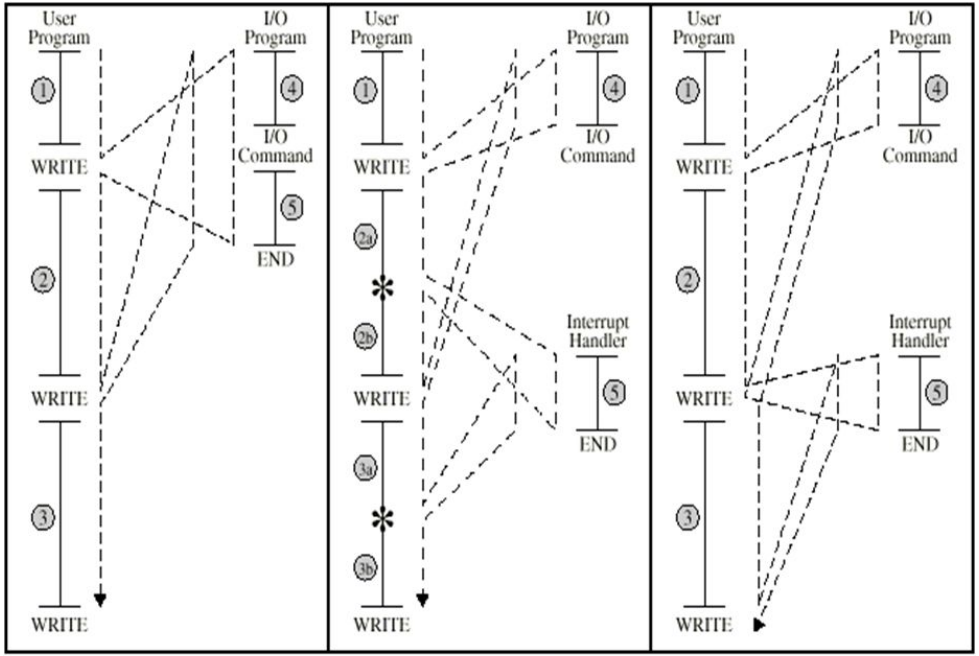


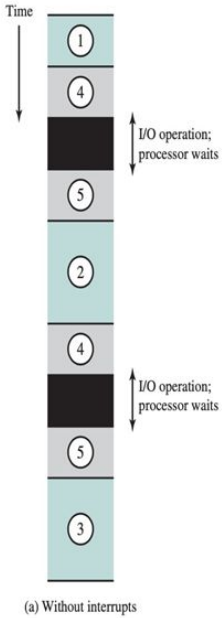
Interrupts

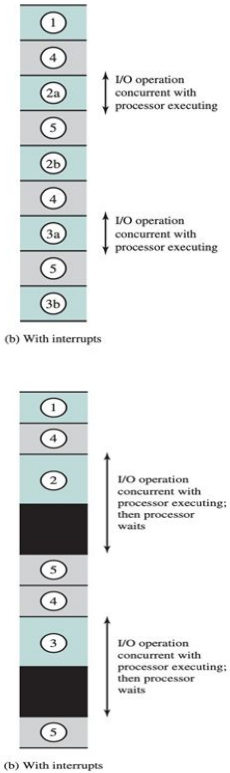
Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing

Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, or reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation, request service from the processor, or to signal a variety of error conditions.
Hardware Failure	Generated by a failure such as power failure or memory parity error.

Program Flow Control

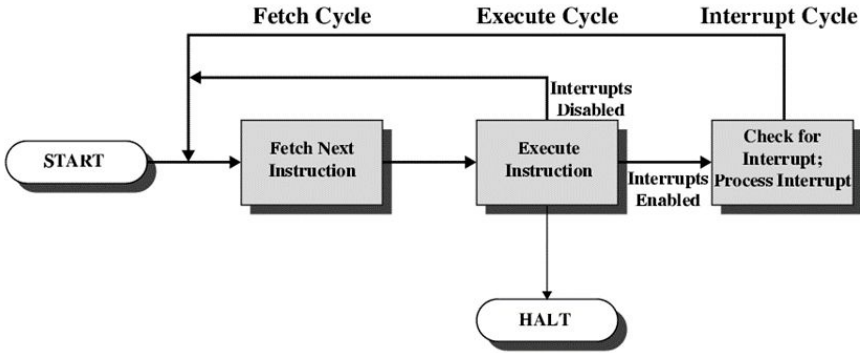


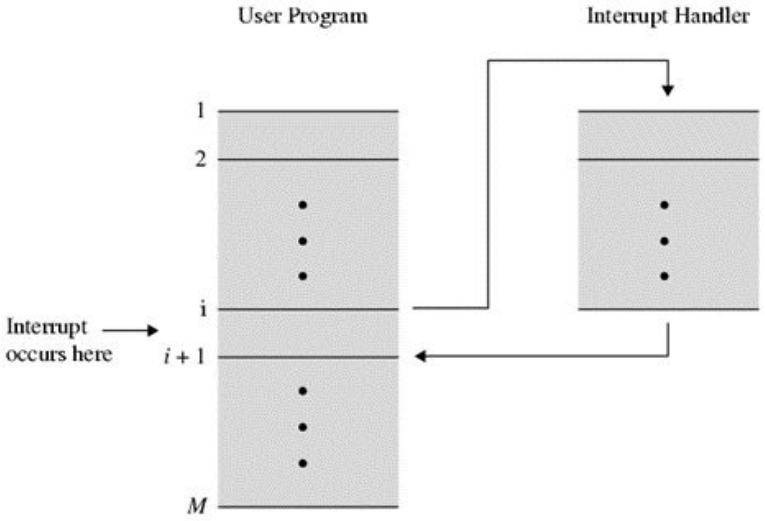




Transfer of Control via Interrupts

Instruction Cycle with Interrupts





Instruction Cycle with Interrupts

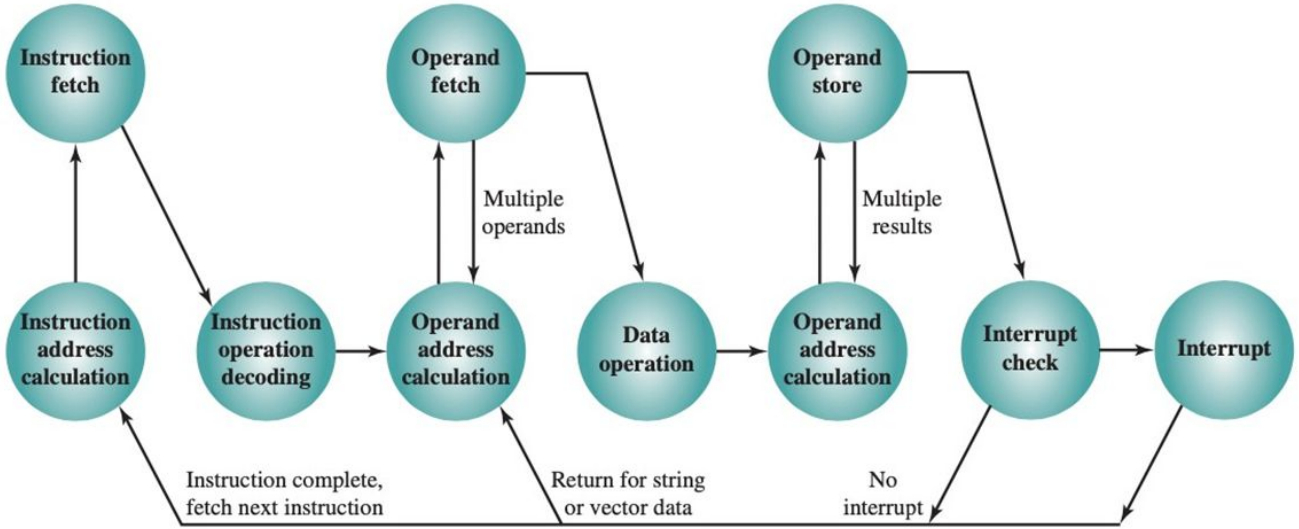
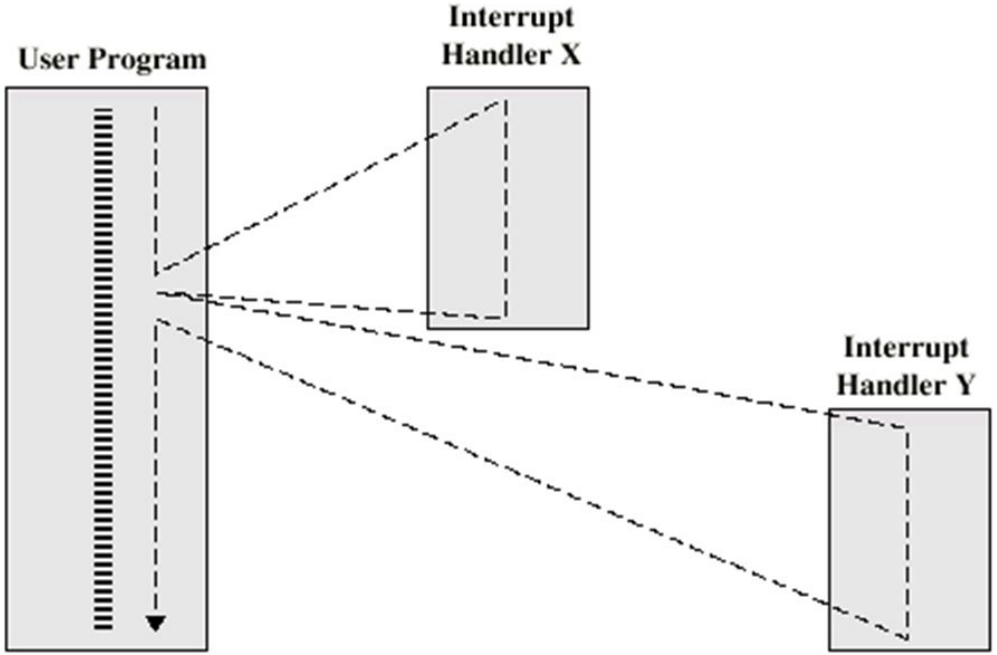


Figure 3.12 Instruction Cycle State Diagram, with Interrupts

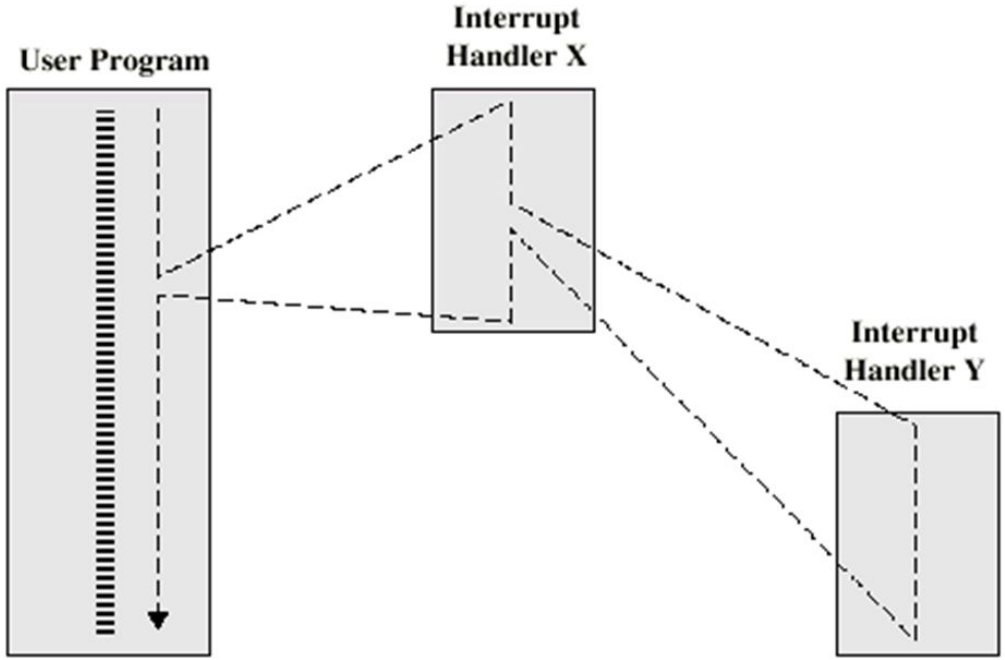
Multiple Interrupts

- **Disable interrupts**
  - Processor will ignore further interrupts whilst processing one interrupt
  - Interrupts remain pending and are checked after first interrupt
  - has been processed
  - Interrupts handled in sequence as they occur
- **Define priorities**
  - Low priority interrupts can be interrupted by higher priority interrupts
  - When higher priority interrupt has been processed, processor returns to previous interrupt

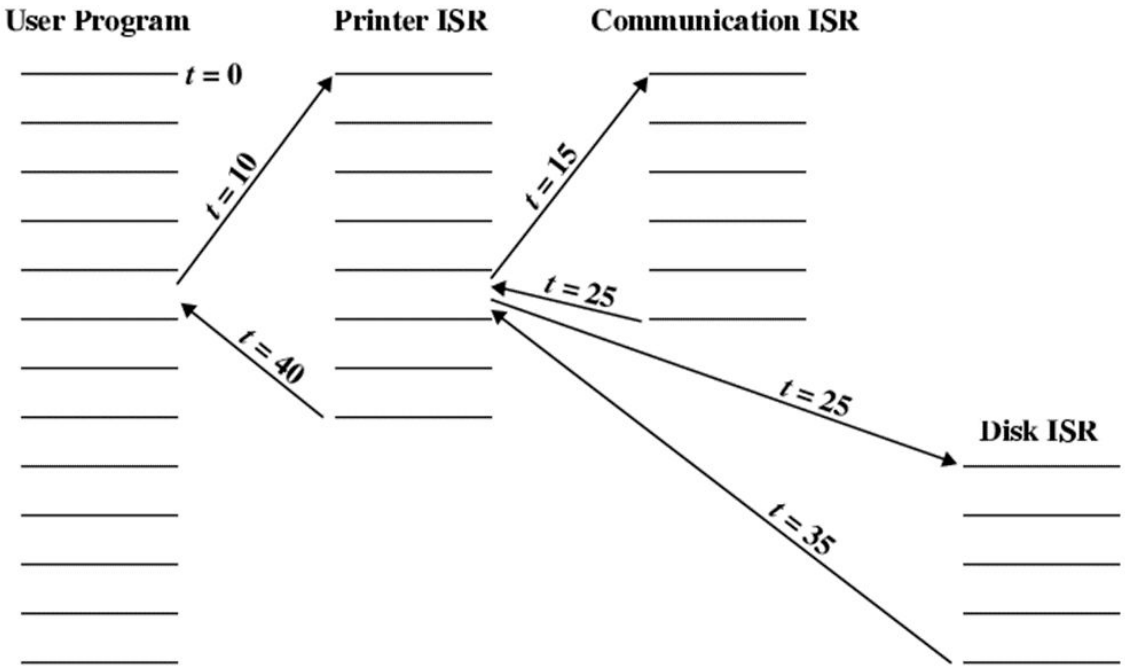
Multiple Interrupts - Sequential



Multiple Interrupts – Nested



Time Sequence of Multiple Interrupts



Thank You