Interrupts

Computer System Architecture
By
M. Morris Mano
Computer Organization and Architecture
by
William Stallings

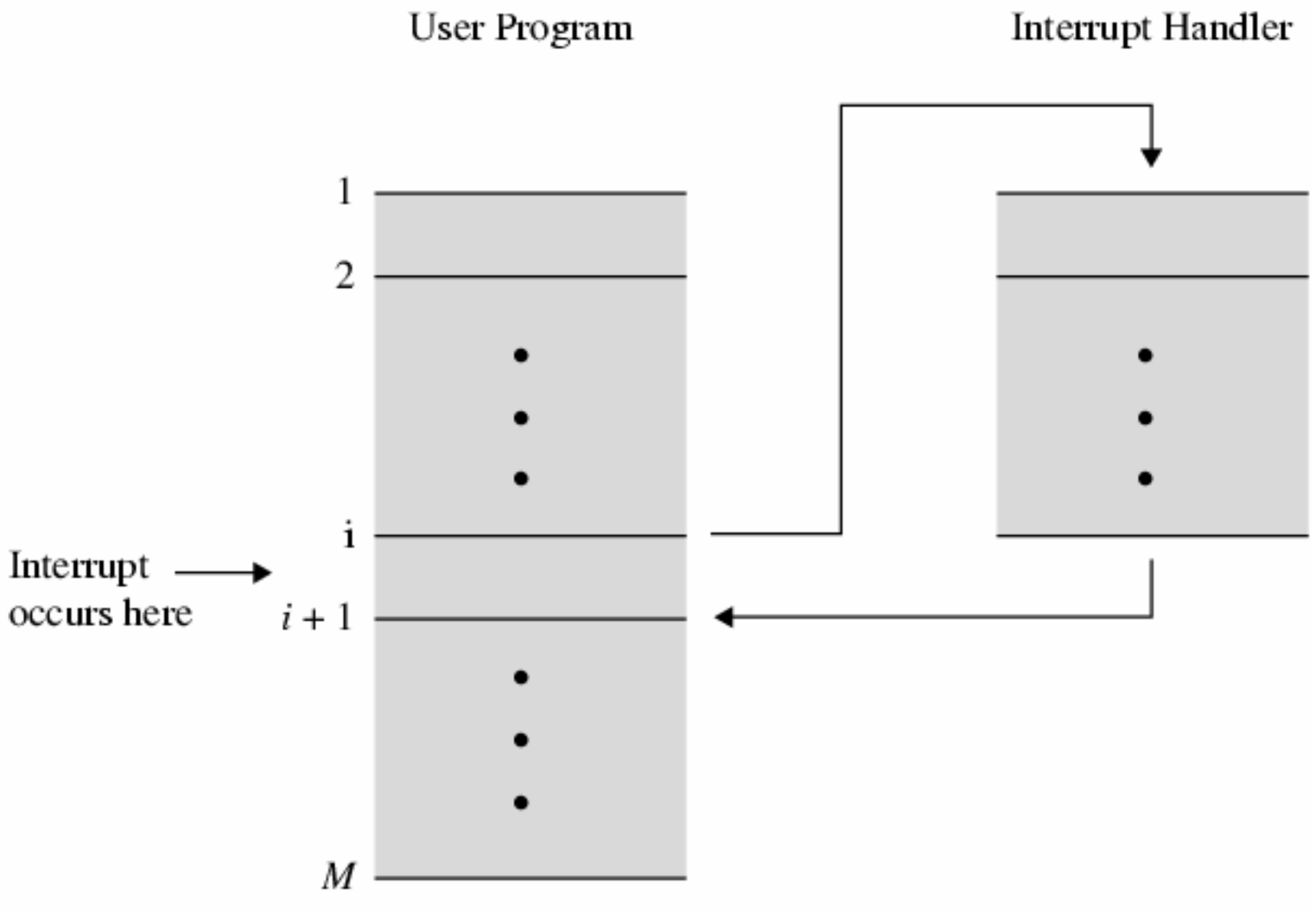
Interrupts

- Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing
- Program
  - Ex: overflow, division by zero
- Timer
  - Generated by internal processor timer
  - Used in pre-emptive multi-tasking
- I/O
  - From I/O controller
- Hardware failure
  - Ex: memory parity error

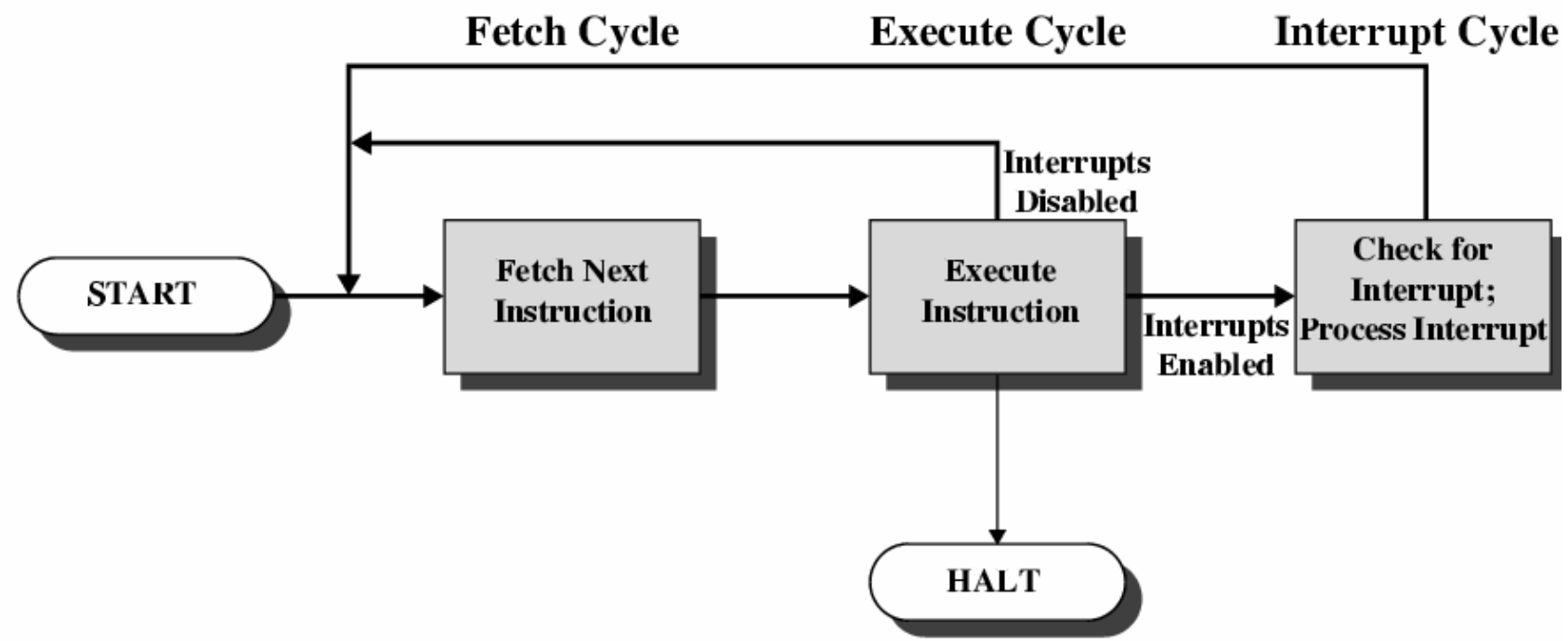
Program Interrupt

- Transfer of program control from a currently running program to another service program as a result of an external or internal generated request.
- Control returns to the original program after the service program is executed.
- It is a way to improve processing efficiency.
- Ex: most external devices are much slower than the processor.

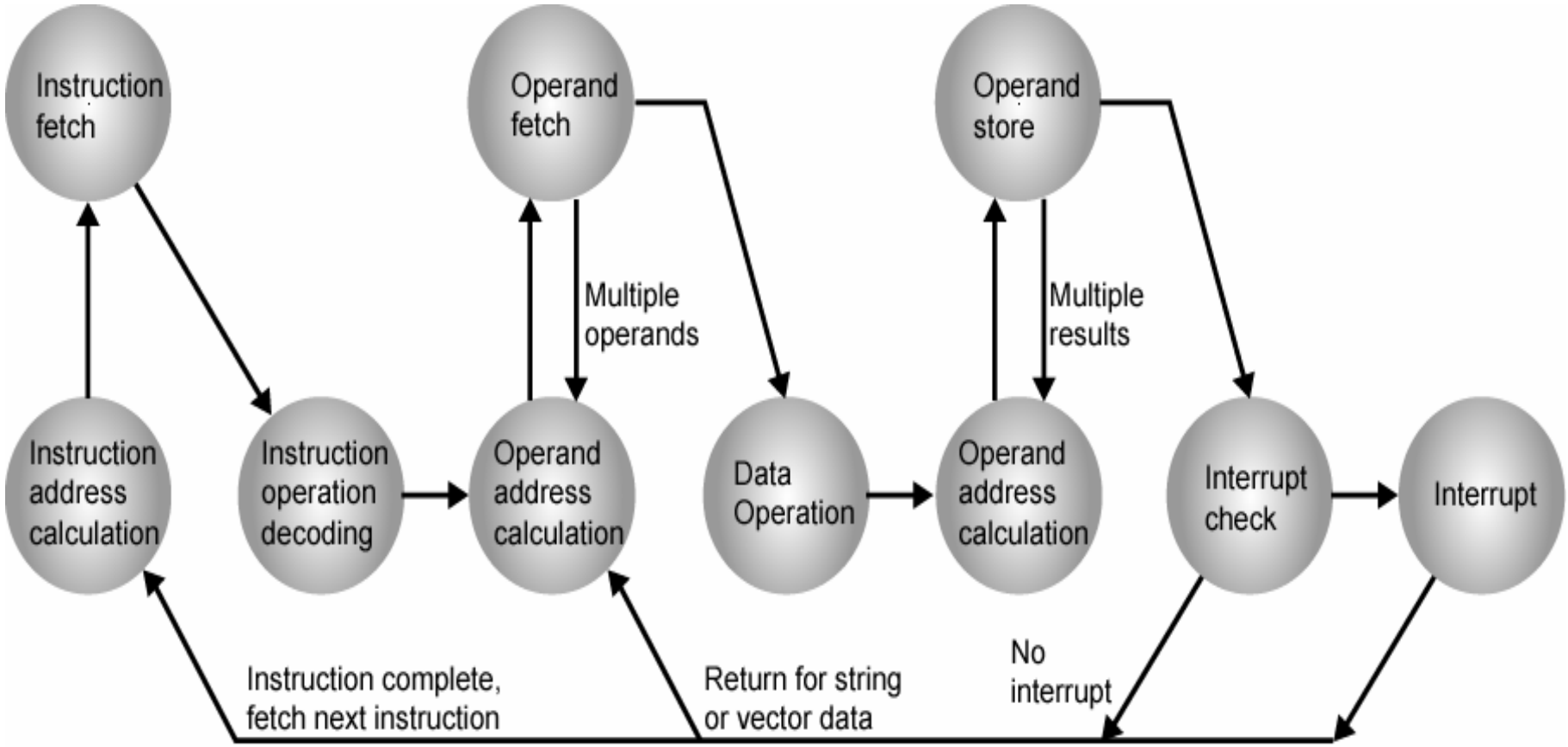
Transfer of Control via Interrupts



Instruction Cycle with Interrupts



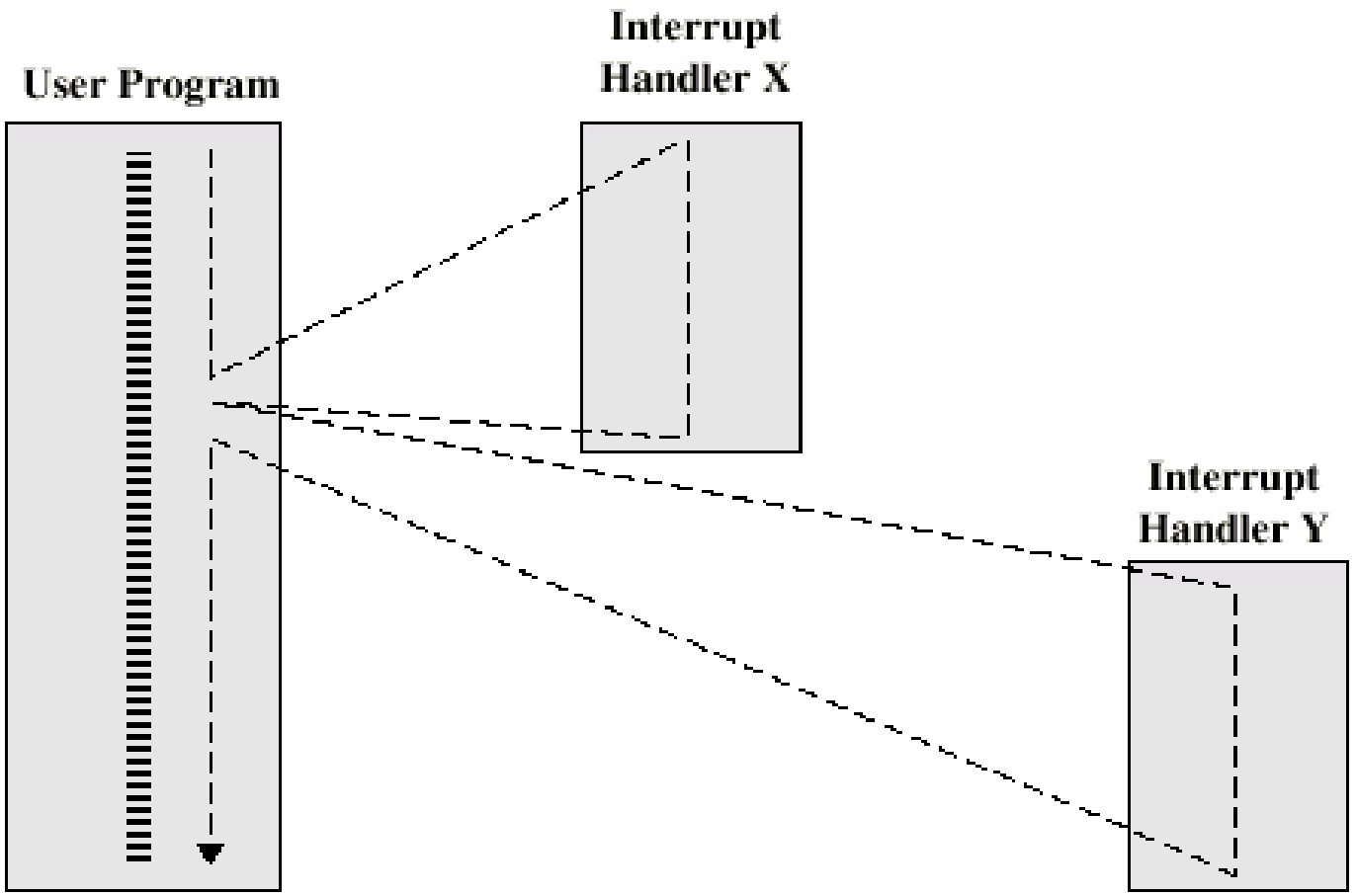
Instruction Cycle (with Interrupts) - State Diagram



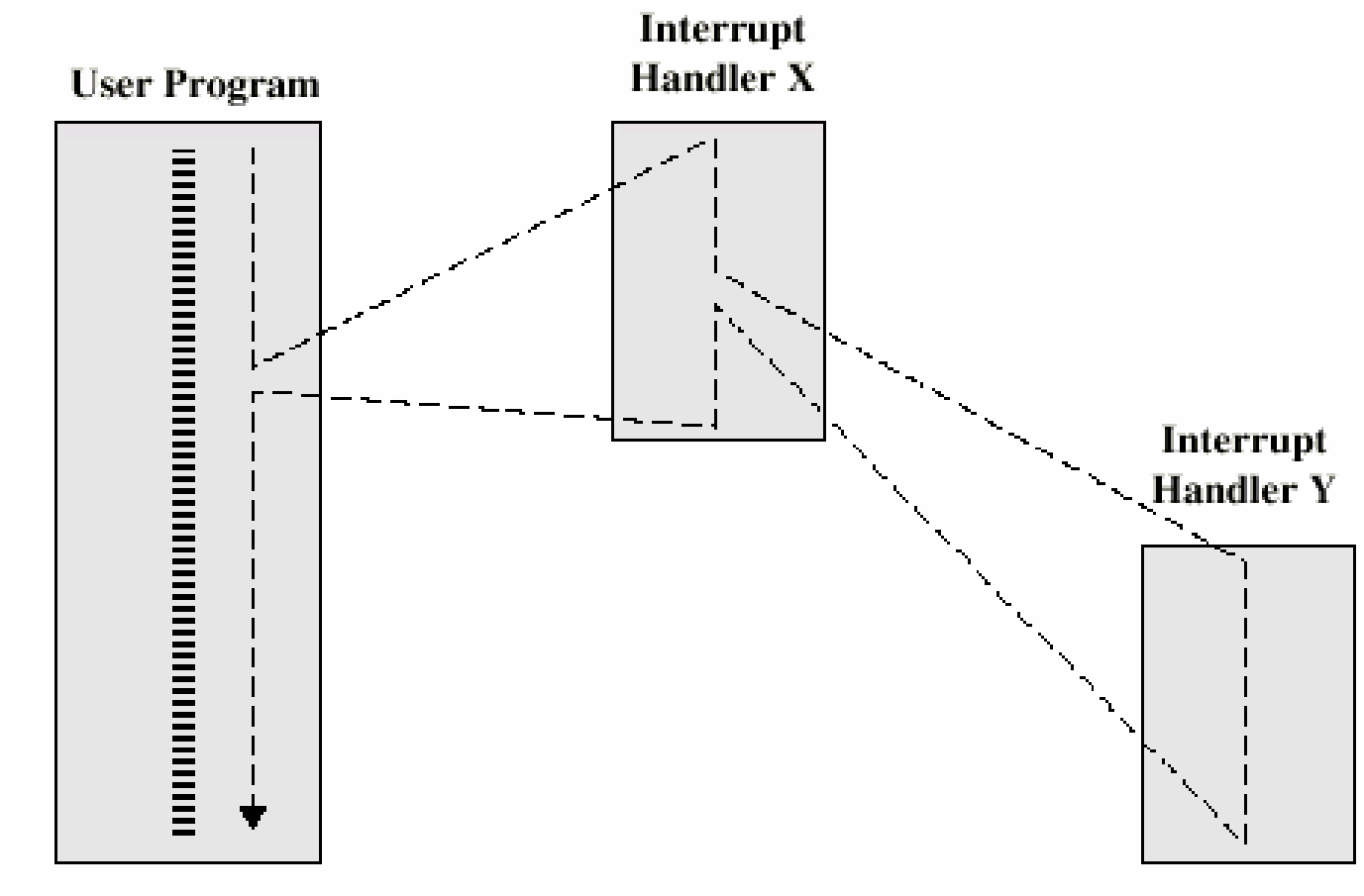
Multiple Interrupts

- **Disable interrupts**
  - Processor will ignore further interrupts while it is 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.

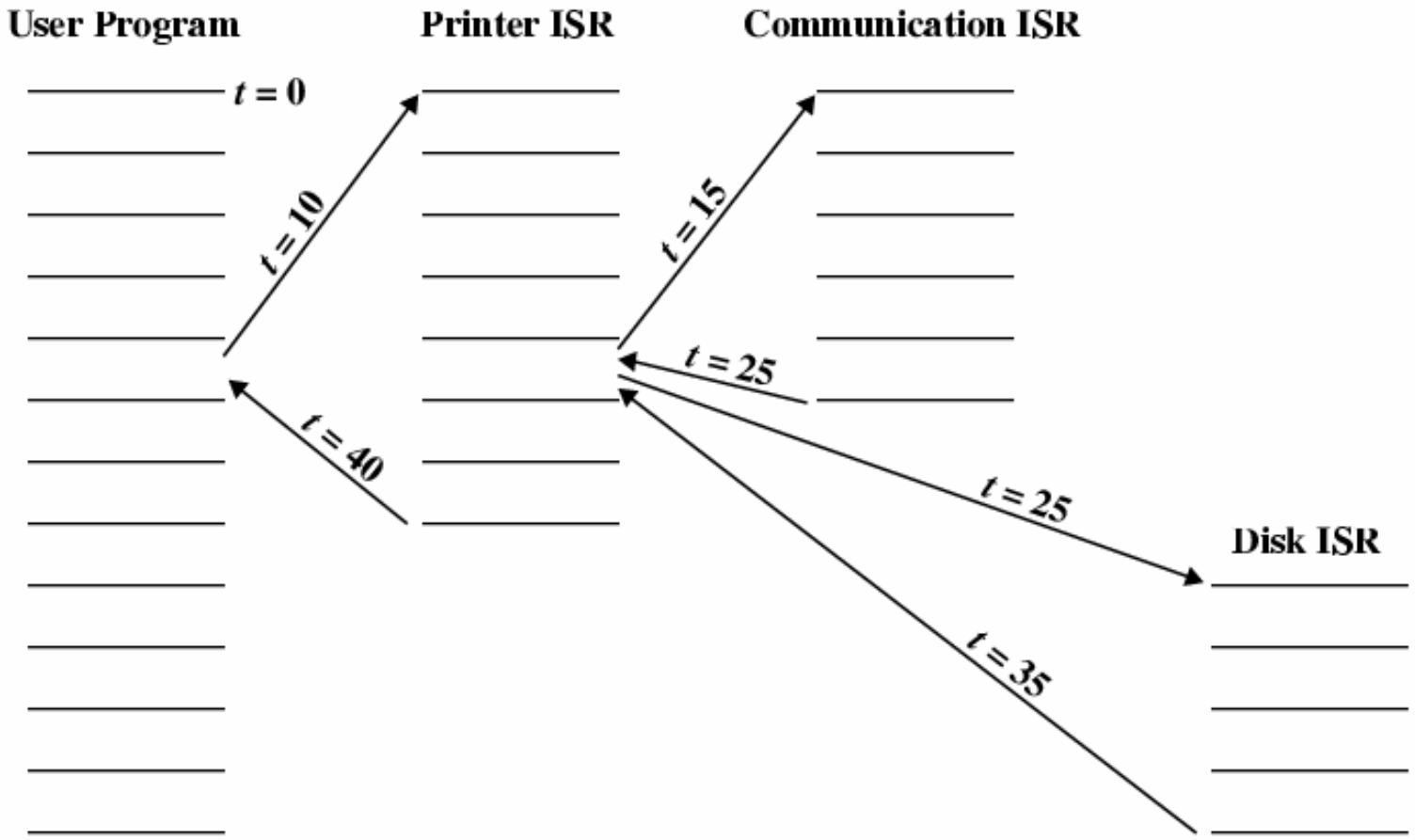
Multiple Interrupts - Sequential



Multiple Interrupts – Nested



Time Sequence of Multiple Interrupts



Interrupt procedure

- Similar to Subroutine Call except for 3 variation

1. Interrupt is initiated by an internal or external signal (*except for software interrupt*)
    - A subroutine call is initiated from the execution of an instruction (CALL).
2. Address of the interrupt service program is determined by the hardware
    - Address of the subroutine call is determined from the address field of an instruction
3. Interrupt procedure stores all the information necessary to define the state of the CPU.
    - A subroutine call stores only the program counter (*Return address*)

PSW and Operating mode

- Program Status Word (PSW)
  - Collection of all status bit conditions in the CPU is called a PSW
  - It is stored in a hardware register.
  - Contains status information that characterizes state of CPU.
- Two CPU Operating Modes
  - Supervisor (*System*) Mode: Privileged Instruction
    - CPU is executing a program that is part of the operating system.
  - User Mode: User program

Interrupt procedure cont..

- Some computer store only PC.
- Few computers store both PC and all status and register content.
- Most computers store PC and PSW.

Types of Interrupts

1. External Interrupts
2. Internal Interrupts or TRAP
3. Software Interrupts

1. External Interrupts

- Come from
  - I/O devices,
  - a timing device,
  - a circuit monitoring the power supply, or
  - any other external source
- Ex:
  - I/O device request- transfer of data
  - I/O device finished transfer of data.
  - Elapsed time of event.
  - Power failure.

2. Internal Interrupts or TRAP

- Arise from illegal use of an inst. or data.
- Caused by internal error conditions
  - register overflow,
  - attempt to divide by zero,
  - an invalid operation code,
  - stack overflow, and
  - protection violation

Software Interrupts

- Initiated by executing an instruction (**INT** or **RST**)
- Used by the programmer to initiate an interrupt procedure at any desired point in the program

Procedure for processing an interrupt

1. State of CPU is pushed into stack.
2. Beginning address of the service routine is transferred to the PC.
3. Service routine determine what caused the interrupt and proceed to service it.
4. Last inst. in the service routine is a **return from interrupt.**
   - Pop old PSW and return the address.
5. State of CPU is restored and the original program can continue.

Procedure for executing ISR