Priority Interrupt

Polling

Daisy chaining

Subject: Computer Organization & Architecture

SAVIYA VARGHESE
Dept of BCA
2020-21

Priority Interrupt

- A priority interrupt is a system that establishes a priority over the various sources to determine which condition is to be serviced first when two or more requests arrive simultaneously.
- The system may also determine which conditions are permitted to interrupt the computer while another interrupt is being serviced.
- Devices with highspeed transfers such as magnetic disks are given high priority, and slow devices such as keyboards receive low priority.
- When two devices interrupt the computer at the same time, the computer services the device, with the higher priority first

Priority Interrupt

- In a typical application a number of IO devices are attached to the computer, with each device being able to originate an interrupt request.
- The first task of the interrupt system is to identify the source of the interrupt.
- Several sources may request service simultaneously. In this case the system must also decide which device to service first

Priority Interrupt

- 2 methods
  - Software → Polling
  - Hardware → Daisy Chaining, Parallel priority

Software Method

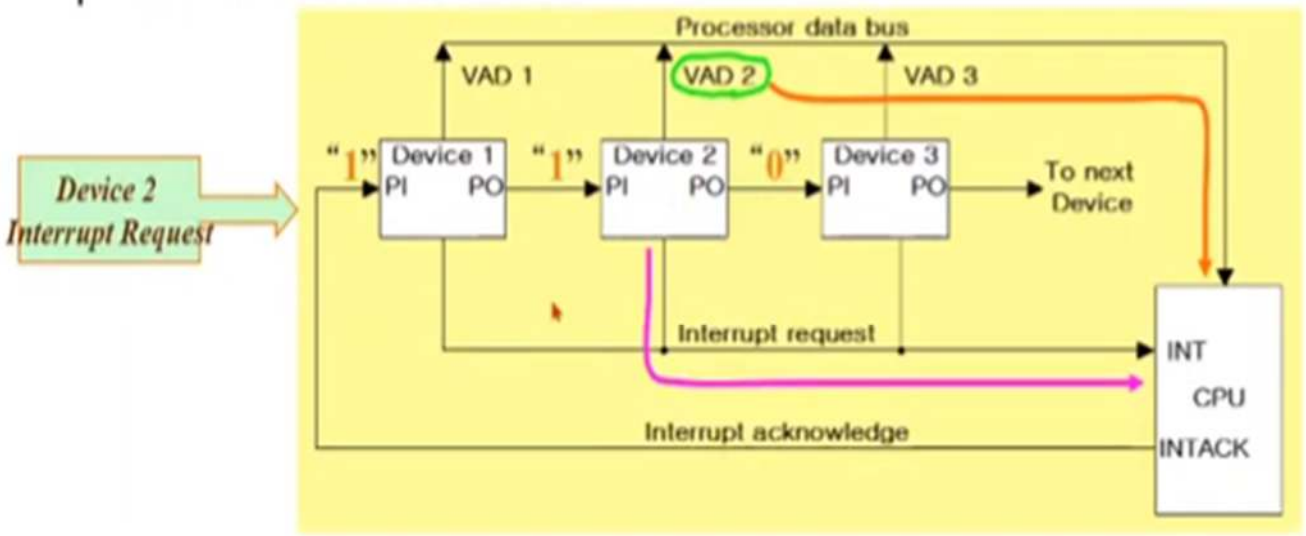
- A polling procedure is used to identify the highest-priority source by software means.
- In this method there is one common branch address for all interrupts.
- The interrupt handling program begins at the branch address and polls the interrupt sources in sequence.
- The order in which they are tested determines the priority of each interrupt.
- The highest-priority source is tested first, and if its interrupt signal is on, control branches to a service routine for this source.
- Otherwise, the next-lower-priority source is tested, and so on
- **Disadvantage**:If there are many interrupt sources, the time required to poll them can exceed the time available to service the I/O device

Hardware priority interrupt

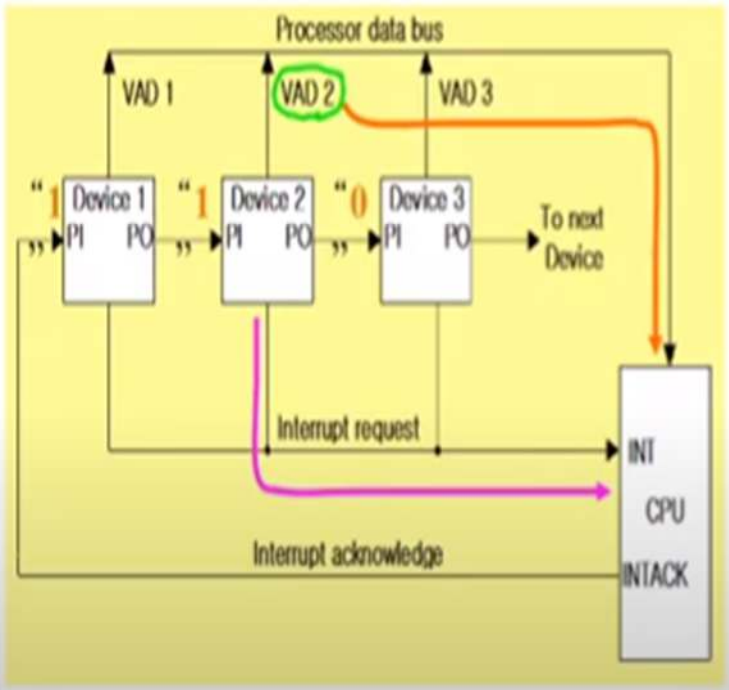
- It accepts interrupt requests from many sources, determines which of the incoming requests has the highest priority, and issues an interrupt request to the computer based on this determination.
- To speed up the operation, each interrupt source has its own interrupt vector to access its own service routine directly.
- The hardware priority function can be established by either a serial or a parallel connection of interrupt lines.
- The serial connection is also known as the **daisy chaining** method

Daisy Chaining Method

- It consists of a serial connection of all devices that request an interrupt.
- The device with the highest priority is placed in the first position, followed by lower-priority devices up to the device with the lowest priority, which is placed last in the chain

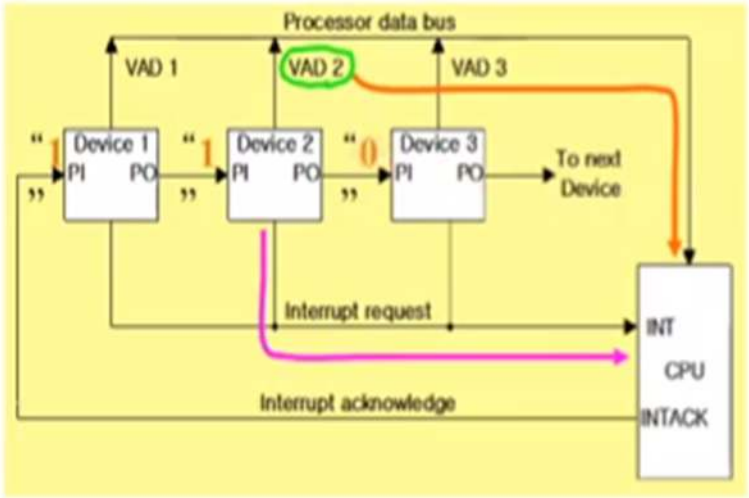


Daisy Chaining Method



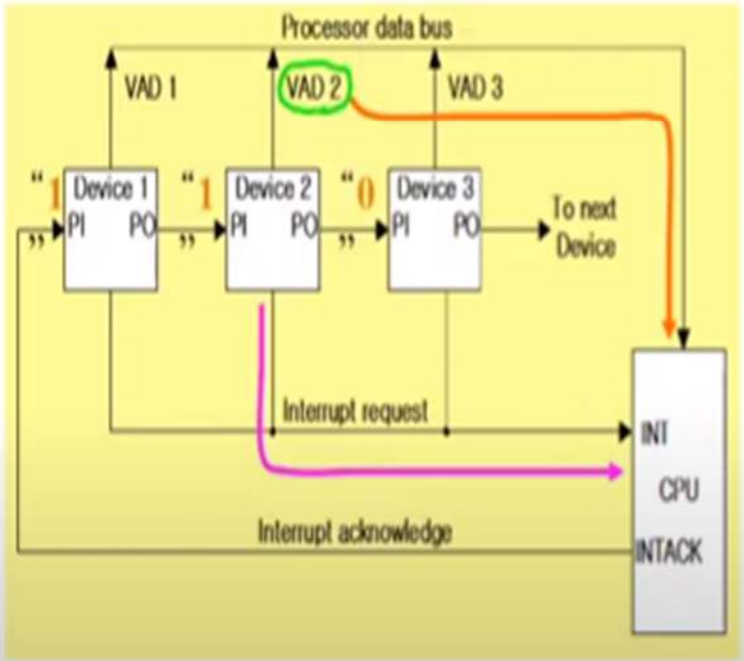
- The interrupt request line is common to all devices and forms a wired logic connection.
- If any device has its interrupt signal in the low-level state, the interrupt line goes to the low-level state and enables the interrupt input in the CPU.
- When no interrupts are pending, the interrupt line stays in the high-level state and no interrupts are recognized by the CPU.

Daisy Chaining Method



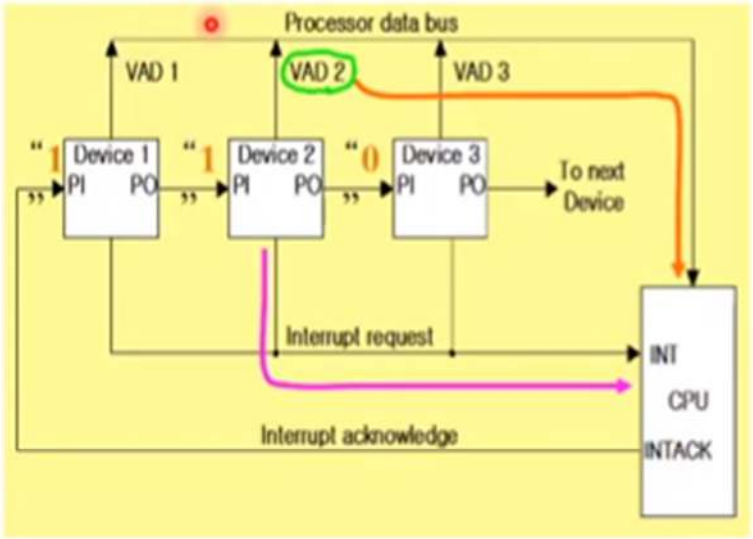
- The CPU responds to an interrupt request by enabling the interrupt acknowledge line.
- This signal is received by device 1 at its PI (priority in) input.
- The acknowledge signal passes on to the next device through the PO (priority out) output only if device 1 is not requesting an interrupt.

Daisy Chaining Method



- If device 1 has a pending interrupt, it blocks the acknowledge signal from the next device by placing a 0 in the PO output.
- It then proceeds to insert its own interrupt vector address (VAD) into the data bus for the CPU to use during the interrupt cycle.

Daisy Chaining Method



- A device with a 0 in its PI input generates a 0 in its PO output to inform the next-lower-priority device that the acknowledge signal has been blocked.
- A device that is requesting an interrupt and has a 1 in its PI input will intercept the acknowledge signal by placing a 0 in its PO output.
- If the device does not have pending interrupts, it transmits the acknowledge signal to the next device