Arbitration
and
Communication Protocols

Arbitration:

- It is the method to arbitrate among some contending issues such as when two or more peripherals request service simultaneously.
- Several situations exist in which multiple peripherals request service might form a single resource. For example:
- Peripherals might share a single microprocessor that services their interrupt requests.
- Multiple peripherals might share a single DMA controller.

In such situation, two or more peripherals may request service simultaneously. We therefore must decide which one of the contending (competing) peripherals gets service and thus which peripherals need to wait.

Several methods exist to resolve such situation called as arbitration methods.

1. Priority Based Arbitration
2. Daisy-Chain Arbitration
3. Network-Oriented Arbitration

Priority Based Arbitration:

- Single purpose processor ,arbitrating among multiple peripherals using vectored interrupt to request servicing from the microprocessor.
- Each peripheral makes request to arbiter, arbiter makes request to resource i.e. microprocessor.
- Arbiter after receiving acknowledgement, provides the acknowledgement to exactly one peripheral, which permits the peripheral to put its interrupt vector address on the data bus.

Arbiter connected to system bus for configuration only (for configuring registers within the arbiter to set the priority schemes and/or the relative priorities of the devices).

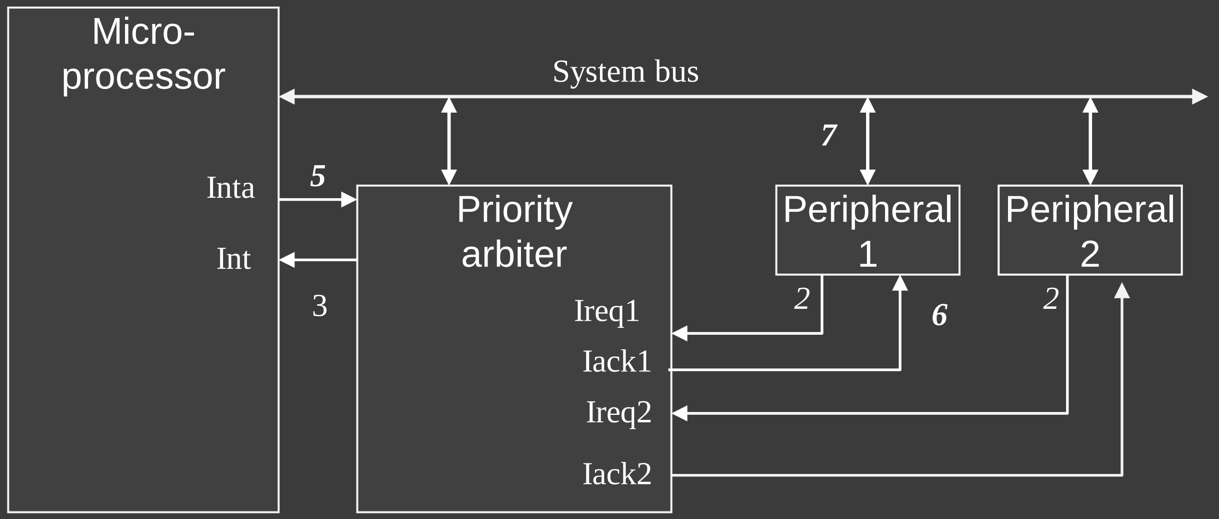


Fig: Arbitration using a priority arbiter

1. Microprocessor is executing its program.
2. Peripheral1 needs servicing so asserts *Ireq1*. Peripheral2 also needs servicing so asserts *Ireq2*.
3. Priority arbiter sees at least one *Ireq* input asserted, so asserts *Int*.
4. Microprocessor stops executing its program and stores its state.
5. Microprocessor asserts *Inta*.

6. Priority arbiter asserts *Iack1* to acknowledge Peripheral1.

7. Peripheral1 puts its interrupt address vector on the system bus

8. Microprocessor jumps to the address of ISR read from data bus, ISR executes and returns (and completes handshake with arbiter).

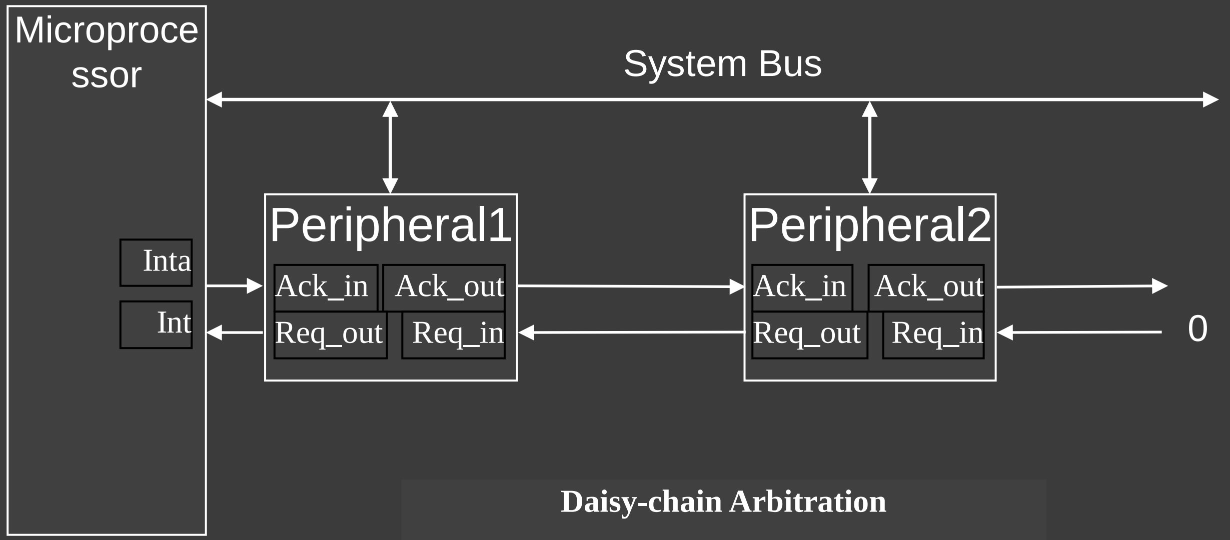
9. Microprocessor resumes executing its program

Types of Priority:

1. Fixed Priority:
   - Each peripheral has unique rank.
   - Highest rank chosen first with simultaneous requests.
   - preferred when clear difference in rank between peripherals.
2. Rotating Priority:
   - priority changed based on history of servicing
   - better distribution of servicing especially among peripherals with similar priority demands

Daisy-Chain Arbitration

- Arbitration is done by peripherals themselves.



Daisy-chain Arbitration

- Each peripherals has a request input and a request output plus an acknowledge input and an acknowledge output.

- Peripherals are connected to each other in daisy-chain manner.
  - One peripheral connected to resource, all others connected “up stream”
  - Peripheral’s *req* flows “downstream” to resource, if it requires servicing .
  - resource’s *ack* flows “upstream” to the next requesting peripheral, if doesn’t need service.
  - Closest peripheral to the microprocessor (μp) has highest priority.

Network-Oriented Arbitration

- When multiple microprocessors share a bus (called a network), arbitration is typically built into bus protocols since bus serves as the only connection among the ups.
- Separate processors may try to write simultaneously causing collisions, so processor detects this collisions and;
  - Stop transmitting their data
  - Wait for some time and transmit again
  - Don’t start sending again at same time or must at least use statistical methods

Advanced Communication Principles

Serial Communication:

- a single wire bus, carries one bit at a time along with a control and possible power lines.
- Capable of higher throughputs than parallel buses when use to connect two physically distinct object.
  - Less average capacitance an more bits per unit time
- Cables used are cheaper but with complex interfacing logic and communication protocol.
- On the sending side, a TX decompose data words into bits and on receiving side; RX must compose bits into words.
- START and STOP bits are used rather than extra control signals(as read, write)

Parallel Communication:

- Requires the multiple wire bus, capable of carrying multiple bits at a time with extra wires(control and power wires).
- It has advantages of high data throughput if short length bus but large capacitance for long wire.
- Small variations in length of individual wire of a parallel bus can cause the received bits of the data word to arrive at different times.
- More costly to construct and may be bulky due to insulation from each other to prevent noise.
- It is used when devices reside on same IC or on same circuit board.

Wireless Communication:

It eliminates the need for devices to be physically connected in order to communicate. Physical layer may be either an infrared (IR) channel or a radio frequency (RF) channel.

Infrared(IR):

- Uses EM wave frequencies just below visible light spectrum
- These waves are generated using an IR diode and detected by using IR transistor.
- IR transistor is that which conducts when exposed to IR light.
- A simple transmitter can send 1s by turning on its IR diode and send 0s by turning off its IR diode.
- Cheap to build IR transmitters and receivers

Radio Frequency(RF):

- Uses EM wave frequencies in the radio spectrum
- A transmitter will need to use analog circuitry with an antenna to transmit data.
- Longer distance communication is possible