Bus Systems

Common connection between the CPU, the memory, and the peripheral devices.

One device issues a request for information over the bus, and another device sends the information in response.
Bus Lines

Control signals
- Several lines.
- Processor uses them to control communication.
- Other devices may make requests to the processor.

Data/Address
- May be 8, 16 or some other width.
- May use shared data and address lines (cheaper) or separate (faster).

Types of Busses

- Processor-Memory (Local Bus): Short, transfer at memory speed. Connect CPU, memory, and maybe a few fast devices.
- I/O Busses: Long, slow. SCSI, IDE
- Backplane Busses: Keep everyone happy. PCI

*Distinctions are not always clear.*
Bus Arrangements

Synchronous Busses

- All connections to the bus share the same clock and clock periods.
- Each device knows what it may do during each period.
- Bus length is limited. Clock signals don’t travel well over long wires.
- Faster than asynchronous.

Synchronous/Asynchronous Busses

- Synchronous: Bus has a clock and devices are synchronized.
- Asynchronous: Bus had no clock and devices must handshake.

Asynchronous Busses

- Each communicating device has own clock.
- Devices use special signals to keep track of each other and know what to do next.
- Works over longer distances.
- Devices may be added easily.
- Slower than synchronous.
**Handshaking Protocol**

1. ReadReq
2. Data
3. Ack
4. DataRdy
5. ReadReq
6. Data
7. Ack

**Performance Parameters**

- Data bus width. *More wires mean more bits at a time.*
- Separate data/address lines.
- Block transfers.

**State Machines for Handshaking**

**The Bus Master**

The device which initiates transfers on the bus.

- Simplest: Only the CPU
- Faster: Multiple bus masters.

*If the CPU is the only bus master, it must be involved in every transfer. It is better to relieve the CPU of this work.*
Bus Arbitration

Can’t all talk at once. Who gets to be bus master?

- Daisy Chain. *Requests go through each device in increasing priority.*

- Centralized. *All requests go through a centralized arbiter.*

- Distributed by self-selection. *Each device knows all pending requests and yields to the highest priority.*

- Distributed by collision-detection. *Try it; if it collided, try it again.*

Centralized

![Centralized Diagram]

Daisy Chain

![Daisy Chain Diagram]

Commanding Devices

- The CPU writes data interpreted as command codes.

- The CPU reads status information.

- Memory-mapped: Devices are given addresses in real address space.

- Dedicated I/O instructions.
User Programs Need Not Apply

Only the O/S is allowed to communicate with devices.

- Memory-mapped device addresses are not part of user space.
- I/O instructions are not permitted in privileged mode.

Polling v. Interrupts

- Polling: CPU repeatedly reads the status.
- Interrupts: The device informs the CPU when something happens.

*Polling is simpler, interrupts are more efficient.*

O/S Enforces Security

- Reading keyboards.
- File permissions.
- File system integrity.
- Network packets for other processes.

*User programs could bypass the O/S security systems and data structures if they could make direct access to devices.*

Data Transfer

- Processor can direct each transfer.
- A separate device can direct block transfers. *Direct Memory Access (DMA).*
Direct Memory Access

- CPU sends instructions to DMA.
- DMA controller moves a block of data from the “other device” to the memory.
- CPU does something else in the mean time.
- DMA controller interrupts CPU when done.