Lectures 16: Buses

1. Buses
   - 1.1. Key characteristic = shared transmission medium. So the first thing a module must do in any operation is it must obtain use of the bus.
   - 1.2. Three functional groups
     - 1.2.1. Data lines
     - 1.2.2. Address lines. Typically high-order bits select the module, and low-order bits select a memory location or I/O port.
     - 1.2.3. Control lines: memory write, memory read, I/O write, I/O read, transfer ACK, bus request, bus grant, interrupt request, interrupt ACK, clock, reset.
   - 1.3. Multiple-bus hierarchies to reduce length of bus, and reduce aggregate data transfer demand.
     - 1.3.1. Traditional: CPU↔local bus ↔cache↔system bus↔Main memory & expansion bus interface.
       - 1.3.1.1. Expansion bus interface ↔expansion bus↔network, SCSI, modem, serial.
     - 1.3.2. High performance: CPU↔local bus ↔cache/bridge↔system bus and high-speed bus
       - 1.3.2.1. System bus↔Main memory
       - 1.3.2.2. High-speed bus↔SCSI, FireWire, graphic, video, LAN, & expansion bus interface.
       - 1.3.2.3. Expansion bus interface ↔FAX, Modem, Serial.
   - 1.4. Two types of buses
     - 1.4.1. Dedicated= permanently assigned to a physical subset of computer components, e.g. local bus, or to one function, e.g. address bus vs data bus.
       - 1.4.1.1. Physical dedication has high throughput, but adds size and cost to system
     - 1.4.2. Time Multiplexed=using bus for multiple functions based on clock, e.g. address then data.
   - 1.5. Methods of arbitration
     - 1.5.1. Centralized = bus controller allocates time on the bus.
       - 1.5.1.1. Bus Request Line is an OR of requests from all devices on the bus.
       - 1.5.1.2. Bus Grant Line is daisy chained through the devices. The order of devices on this line determines their priority.
     - 1.5.2. Distributed = each module contains access control logic and the modules act together to share the bus.
       - 1.5.2.1. Arbitration Line = indicates the bus is being granted.
       - 1.5.2.2. Busy Line = indicates whether the bus is being used.
       - 1.5.2.3. Bus Request = indicates whether another device has made a request. Conflicts settled by priority or, if no priorities, then each device must wait a random length of time before attempting to request the bus again.
   - 1.6. Types of Timing
     - 1.6.1. Synchronous = proscribed actions take place at a given clock cycle.
       - 1.6.1.1. Example for CPU read request:
         - 1.6.1.1.1. $T_{1:}$ CPU writes MAR to on address bus, and issues read command on status lines. After address stabilizes CPU enables Address Enable line. Once Address Enable is high, Memory reads address and begins to get the corresponding data.
         - 1.6.1.1.2. T_2: Address Enable disabled. MDR set to read. Memory is getting the data.
         - 1.6.1.1.3. T_3: Memory puts the data on the data bus. After reading data, MDR set to not read, CPU drops read command on status lines.
       - 1.6.1.2. Example for CPU write request:
         - 1.6.1.2.1. T_1: Not used
         - 1.6.1.2.2. T_2: CPU sets MDR and MAR to write to their buses. Once they stabilize the CPU issues a write command
         - 1.6.1.2.3. T_3: Memory writes the data to the address specified. At the end, CPU drops write command.
       - 1.6.1.3. Simpler to implement and test, but is less flexible
     - 1.6.2. Asynchronous = occurrence of one event follows and depends on the occurrence of the previous event.
       - 1.6.2.1. For example, CPU waits for acknowledgement from memory before sending next command.
       - 1.6.2.2. Allows mixture of slow and fast devices. Can take advantage of newer, faster technologies.