CSCI 4717/5717
Computer Architecture

Topic: Buses

Reading: Stallings, Sections 3.4, 3.5, and 7.7

Buses – Common Characteristics

- Multiple devices communicating over a single set of wires
- Only one device can talk at a time or the message is garbled
- Each line or wire of a bus can at any one time contain a single binary digit. Over time, however, a sequence of binary digits may be transferred
- These lines may and often do send information in parallel
- A computer system may contain a number of different buses

Buses – Structure

- Serial versus parallel
- Around 50-100 lines although it's possible to have as few as 3 or 4
- Lines can be classified into one of four groups
  - Data lines
  - Address Lines
  - Control Lines
  - Power

Buses – Structure (continued)

- Bus lines (parallel)
  - Data
  - Address
  - Control
  - Power
- Bus lines (serial)
  - Data, address, and control are sequentially sent down single wire
  - There may be additional control lines
  - Power

Buses – Structure (continued)

- Data Lines
  - Passes data back and forth
  - Number of lines represents width
- Address lines
  - Designates location of source or destination
  - Width of address bus specifies maximum memory capacity
  - High order selects module and low order selects a location within the module

Bus Structure – Control lines

- Because multiple devices communicate on a line, control is needed
- Timing
- Typical lines include:
  - Memory Read or Write
  - I/O Read or Write
  - Transfer ACK
  - Bus request
  - Bus grant
  - Interrupt request
  - Interrupt acknowledgement
  - Clock
  - Reset

Operation – Sending Data

- Obtain the use of the bus
- Transfer the data via the bus
- Possible acknowledgement

Operation – Requesting Data

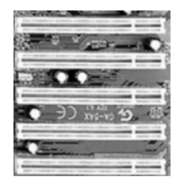
- Obtain the use of the bus
- Transfer the data request via the bus
- Wait for other module to send data
- Possible acknowledgement

Classic Bus Arrangement

- All components attached to bus (STD bus)
- Due to Moore's law, more and more functionality exists on a single board, so major components are now on same board or even the same chip

Physical Implementations

- Parallel lines on circuit boards (ISA or PCI)
- Ribbon cables (IDE)



Physical Implementations (continued)

- Strip connectors on mother boards (PC104)
- External cabling (USB or Firewire)

Single Bus Problems

Lots of devices on one bus leads to:

- Physically long buses
  - Propagation delays – Long data paths mean that co-ordination of bus use can adversely affect performance
  - Reflections/termination problems
- Aggregate data transfer approaches bus capacity
- Slower devices dictate the maximum bus speed

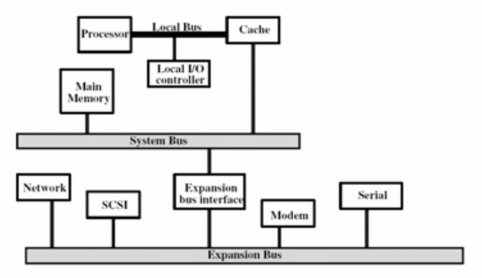
Multiple Buses

- Most systems use multiple buses to overcome these problems
- Requires bridge to buffer (FIFO) data due to differences in bus speeds
- Sometimes I/O devices also contain buffering (FIFO)

Multiple Buses – Benefits

- Isolate processor-to-memory traffic from I/O traffic
- Support wider variety of interfaces
- Processor has bus that connects as direct interface to chip, then an expansion bus interface interfaces it to external devices (ISA)
- Cache (if it exists) may act as the interface to system bus

Expansion Bus Example

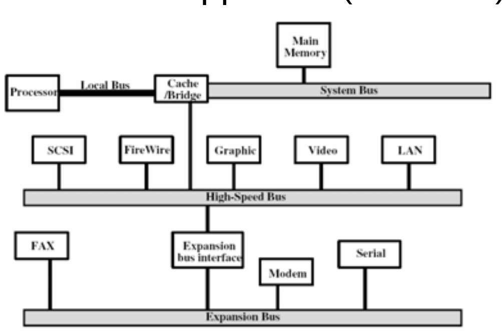


(a) Traditional Bus Architecture

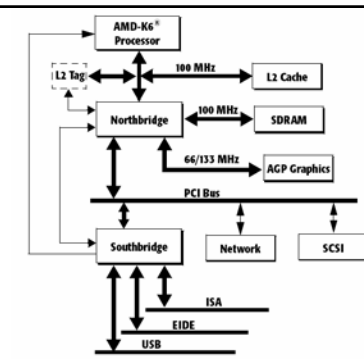
Mezzanine Approach

- Differences in I/O speeds demands separating devices.
- Separate items that are high-speed and those that are not
- An additional high-speed bus is added to communicate with the faster devices and also the slower expansion bus
- Advantage is that high-speed devices are brought closer to processor

Mezzanine Approach (continued)



Pentium Example



(Source: http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/21644.pdf)

Bus Types
Dedicated vs. Time Multiplexed

- Dedicated
  – Separate data & address lines
- Time multiplexed
  – Shared lines
  – Address valid or data valid control line
  – Advantage - fewer lines
  – Disadvantages
    - More complex control
    - Degradation of performance

Bus Types
Physical Dedication

– Physically separating buses and controlling them with a "channel changer“
– Advantages – faster
– Disadvantages – physically larger

Bus Arbitration

- Listening to the bus is not usually a problem
- Talking on the bus is a problem – need arbitration to allow more than one module to control the bus at one time
- Arbitration may be centralised or distributed

Centralised vs. Distributed Arbitration

- Centralised Arbitration
  – Single hardware device controlling bus access – Bus Controller/Arbiter
  – May be part of CPU or separate
- Distributed Arbitration
  – Each module may claim the bus
  – Access control logic is on all modules
  – Modules work together to control bus

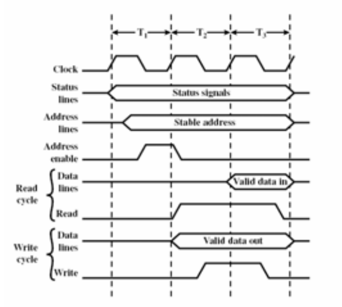
Bus Timing

- Co-ordination of events on bus
- Synchronous – controlled by a clock
- Asynchronous – timing is handled by well-defined specifications, i.e., a response is delivered within a specified time after a request

Synchronous Bus Timing

- Events determined by clock signals
- Control Bus includes clock line
- A single 1-0 cycle is a bus cycle
- All devices can read clock line
- Usually sync on leading/rising edge
- Usually a single cycle for an event
- Analogy – Orchestra conductor with baton
- Usually stricter in terms of its timing requirements

Synchronous Bus Timing



Asynchronous Timing

- Devices must have certain tolerances to provide responses to signal stimuli
- More flexible allowing slower devices to communicate on same bus with faster devices.
- Performance of faster devices, however, is limited to speed of bus

Asynchronous Timing – Read

Asynchronous Timing – Write

Bus Width

- Wider the bus the better the data transfer rate or the wider the addressable memory space
- Serial “width” is determined by length/duration of frame

Peripheral Component Interconnection (PCI) Bus

Brief history

- Original PC came out with 8-bit ISA bus which was slow, but had enormous amount of existing equipment.
- For AT, IBM expanded ISA bus to 16-bit by adding connector
- Many PC board manufacturers started making higher speed, proprietary buses
- Intel released the patents to its PCI and this soon took over as the standard

PCI Bus (continued)

Brief list of PCI 2.2 characteristics

- General purpose
- Mezzanine or peripheral bus
- Supports single- and multi-processor architectures
- 32 or 64 bit – multiplexed address and data
- Synchronous timing
- Centralized arbitration (requires bus controller)
- 49 mandatory lines (see Table 3.3)

Required PCI Bus Lines (Table 3.3)

- Systems lines – clock and reset
- Address & Data
  - 32 time multiplexed lines for address/data
  - Parity lines
- Interface Control
  - Hand shaking lines between bus controller and devices
  - Selects devices
  - Allows devices to indicates when they are ready

Required PCI Bus Lines (continued)

- Arbitration
  - Not shared
  - Direct connection to PCI bus arbiter
- Error lines – parity and critical/system

Optional PCI Bus Lines

There are 51 optional PCI 2.2 bus lines

- Interrupt lines
  - Not shared
  - Multiple lines for multiple interrupts on a single device
- Cache support
- 64-bit Bus Extension
  - Additional 32 lines
  - Time multiplexed
  - 2 lines to enable devices to agree to use 64-bit transfer
- JTAG/Boundary Scan – For testing procedures

PCI Commands

- Transaction between initiator (master) and target
- Master claims bus
- During address phase of write, 4 C/BE lines signal the transaction type
- One or more data phases

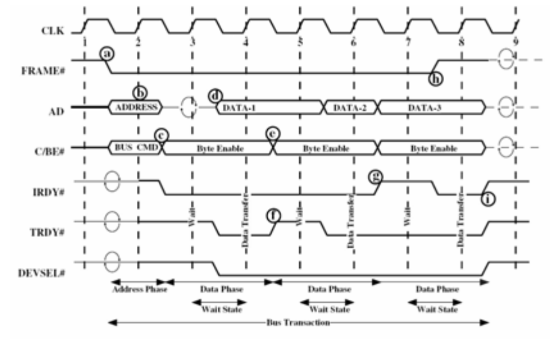
PCI Transaction Types

- Interrupt acknowledge – prompts identification from interrupting device
- Special cycle – message broadcast
- I/O read – read to I/O address space
- I/O write – write to I/O address space
- Memory read – 1 or 2 data transfer cycles
- Memory read line – 3 to 12 data transfer cycles
- Memory read multiple – more than 12 data transfers

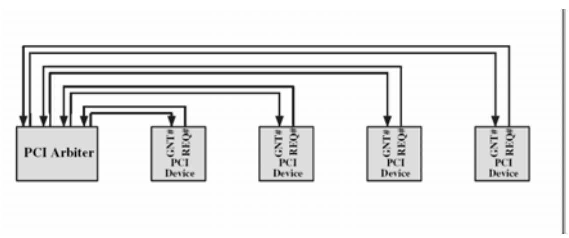
PCI Transaction Types (continued)

- Memory write – writing 1 or more cycles to memory
- Memory write and invalidate – writing 1 or more cycles to memory allowing for cache write-back policy
- Configuration read – reading PCI device's configuration (up to 256 configuration registers per device)
- Configuration write – writing PCI device's configuration (up to 256 configuration registers per device)
- Dual address cycle – indication of 64-bit addressing on 32 bit lines

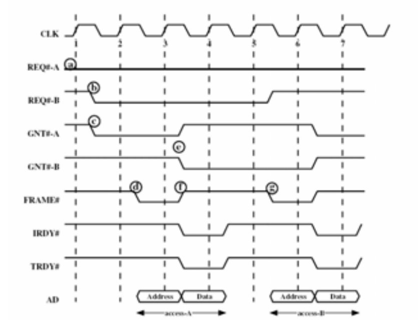
PCI Read Timing Diagram



PCI Bus Arbiter



PCI Bus Arbitration Between Two Masters



Higher Performance External Buses

- Historically, parallel has been used for high speed peripherals (e.g., SCSI, parallel port zip drives rather than serial port). High speed serial, however, has begun to replace this need
- Serial communication also used to be restricted to point-to-point communications. Now there's an increasing prevalence of multipoint

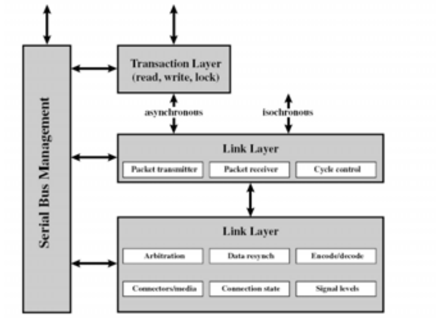
IEEE 1394 FireWire

- Inexpensive alternative needed for SCSI
- High performance serial bus
- Serial implies cheaper cabling (fewer wires, less shielding, less synchronization)
- Small connectors for smaller devices
- Fast
- Low cost
- Easy to implement
- Also being used in digital cameras, VCRs and TVs

FireWire Configuration

- Daisy chain/tree structure (Mac O/S Help indicates that daisy chain is preference for up to 16 devices)
- Up to 63 devices on single port – really 64 of which one is the interface itself
- Up to 1022 buses can be connected with bridges
- Automatic configuration for addressing
- No bus terminators
- Hot swappable

FireWire 3 Layer Stack



FireWire 3 Layer Stack
Physical Layer

- Transmission medium, electrical and signaling characteristics
- Up to 400 Mbps
- Arbitration – basic form
  - Fair arbitration
  - Urgent arbitration
- Link layer packet transmission
  - Asynchronous
  - Isochronous

Arbitration – Basic form

- Upon automatic configuration, each tree designates a root
- Parent/child relationship forms tree topology
- Root acts as central arbitrator
- Requests are first-come-first-serve
- Simultaneous requests are granted first to the closest node to the root and second to the lower ID number
- Two additional functions are used to best allocate the use of the bus
  - Fair arbitration
  - Urgent arbitration

Fair arbitration

- Keeps one device from monopolizing the bus by allowing only one request during a set *fairness interval*
- At beginning of interval, all devices set arbitration_enable flag
- Each device may compete for bus access
- If bus access is granted, arbitration_enable flag is cleared prohibiting bus access until next fairness interval

Urgent arbitration

- Allows overriding of fairness interval by nodes configured as having an *urgent priority*
- Provides support for devices with severe latency requirements or high throughput such as video
- They may use the bus up to 75% of the time, i.e., for each non-urgent packet, three urgent packets may be sent

FireWire 3 Layer Stack
Link Layer

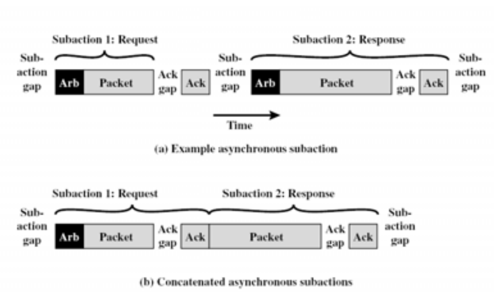
- Transmission of data in packets
- Two types of transmission supported:
  - Asynchronous
  - Isochronous

Asynchronous

Packets contain

- Variable amount of data
- Several bytes of transaction layer information
- Addressing information
- Acknowledgement

Asynchronous Sequence



(a) Example asynchronous subaction

(b) Concatenated asynchronous subactions

Asynchronous Sequence (continued)

- Arbitration sequence – gives one device control of the bus
- Packet transmission – packet contains source and destination IDs, type, CRC, and possible data
- Acknowledgement gap – allows destination to receive and process message
- Acknowledgement – destination sends packet containing source and destination IDs and code indicating action taken
- Subaction gap – idle period between packets to avoid bus contention

Isochronous

- Fixed packet size w/variable amount of data
- Simplified addressing
- No acknowledgement
- Periodically, cycle_start packet is sent indicating period where only isochronous packets are transmitted
- Transaction – Request-response protocol (This is what your code sees. It separates the programmer from the packet-level stuff)

Isochronous Sequence