ECE 280 / CSE 280
Digital Design Laboratory
Lecture 4

Bus Design
Bus Structure
Buses

- There are a number of possible interconnection systems
- Single and multiple bus structures are most common
- Examples
  - ISA (Early PCs)
  - PCI (PCs - 1993 and later)
  - PCMCIA (Laptops)
  - NuBus (MacII series)
  - MicroChannel (IBM PS/2)
  - Sbus (Sun)
Buses

• Advantages
  – Allows movement of peripheral devices from one system to another
  – Easy expandability of system
  – Cheaper than point to point connections
    • Sharing of multiple wires
Buses

• Disadvantages
  – Communication bottleneck
  – Throughput limited by
    • Length of bus
    • Number of devices on bus
    • Slowest device on bus (e.g. PCI-X)
Master versus Slave

- A bus transaction includes two parts:
  - Issuing the command (and address) – request
  - Transferring the data – action

- Master (Initiator) is the one who starts the bus transaction by:
  - issuing the command (and address)

- Slave (Target) is the one who responds to the address by:
  - Sending data to the master if the master ask for data
  - Receiving data from the master if the master wants to send data
Bus Design

• Components
  – Data Bus
  – Address Bus
  – Control Signals
  – Power/Ground
Single Bus Problems

• Lots of devices on one bus leads to:
  – Propagation delays
    • Long data paths mean that co-ordination of bus use can adversely affect performance
    • If aggregate data transfer approaches bus capacity
  – Multiple bus masters can cause contention delays

• Most systems use multiple buses to overcome these problems
Traditional Bus Architecture
High Performance Bus
Bus Types

• Processor-Memory Bus (design specific)
  – Short and high speed
  – Only need to match the memory system
    • Maximize memory-to-processor bandwidth
  – Connects directly to the processor
  – Optimized for cache block transfers

• I/O Bus (industry standard)
  – Usually is lengthy and slower
  – Need to match a wide range of I/O devices
  – Connects to the processor-memory bus or backplane bus
North/South Bridge architectures: separate busses

- Separate sets of pins for different functions
  - Memory bus
  - Caches
  - Graphics bus (for fast frame buffer)
  - I/O busses are connected to the backplane bus
- Advantage:
  - Busses can run at different speeds
  - Much less overall loading!
Northbridge and Southbridge
Bus Arbitration

- Only a single device may “drive” the bus
- Many devices can “listen”
- Must request the bus
- Wait until the request is granted
Bus Arbitration

- One of the most important issues in bus design:
  - How is the bus reserved by a device that wishes to use it?

- Chaos is avoided by a master-slave arrangement:
  - Only the bus master can control access to the bus:
    - It initiates and controls all bus requests
  - A slave responds to read and write requests

- The simplest system:
  - Processor is the only bus master
  - All bus requests must be controlled by the processor
  - Major drawback: the processor is involved in every transaction
Bus Arbitration

• Bus arbitration scheme:
  – A bus master wanting to use the bus asserts the bus request
  – A bus master cannot use the bus until its request is granted
  – A bus master must signal to the arbiter the end of the bus utilization

• Bus arbitration schemes usually try to balance two factors:
  – Bus priority: the highest priority device should be serviced first
  – Fairness: Even the lowest priority device should never be completely locked out from the bus
Bus Arbitration

• Bus arbitration schemes can be divided into four broad classes:
  – Daisy chain arbitration
  – Centralized, parallel arbitration
  – Distributed arbitration by self-selection: each device wanting the bus places a code indicating its identity on the bus.
  – Distributed arbitration by collision detection:
    - Each device just “goes for it”. Problems found after the fact.
Daisy Chain Arbitration

- Advantage: simple

- Disadvantages:
  - Cannot assure fairness: A low-priority device may be locked out indefinitely
  - The use of the daisy chain grant signal also limits the bus speed
Centralized Arbitration

• Single hardware device controlling bus access
  – Bus Controller
  – Arbiter

• May be part of CPU or separate
Centralized Arbitration

Bus Arbiter

Device 1

Device 2

\ldots

Device N

Grant

Req
Distributed Arbitration

• Each module may claim the bus
• Control logic on all modules
Bus Transaction Protocol

• Simplest Paradigm

• All agents operate synchronously

• All can source / sink data at same rate

• => simple protocol
  – just manage the source and target
Bus Transaction Protocol

- What if the target takes longer to respond?
Bus Transaction Protocol

- Slave indicates when it is ready to transfer data
- Data transfer proceeds at bus rate
Bus Design

- Separate versus multiplexed address and data lines:
  - Address and data can be transmitted in one bus cycle if separate address and data lines are available
  - Cost: (a) more bus lines vs. increased complexity

- Data bus width:
  - By increasing the width of the data bus, transfers of multiple words require fewer bus cycles
  - Example: SPARCstation 20’s memory bus is 128 bit wide
  - Cost: more bus lines
Bus Design

• Block transfers:
  – Allow the bus to transfer multiple words in back-to-back bus cycles
  – Only one address needs to be sent at the beginning
  – The bus is not released until the last word is transferred
  – Cost: (a) increased complexity
    (b) decreased response time for request
Bus Performance

- Overlapped arbitration
  - perform arbitration for next transaction during current transaction

- Bus parking
  - master holds onto bus and performs multiple transactions as long as no other master makes request

- Overlapped address / data phases
  - requires one of the above techniques
Bus Performance

• Split-phase (or packet switched) bus
  – completely separate address and data phases
  – arbitrate separately for each
  – address phase yield a tag which is matched with data phase

• Most modern memory and I/O buses use one or more of these techniques
PCI Bus

- Peripheral Component Interconnection
- Intel released to public domain
- 32 or 64 bit
- 49 lines
PCI Bus
PCI Bus Lines (required)

- Systems lines
  - Including clock and reset
- Address & Data
  - 32 time mux lines for address/data
  - Interrupt & validate lines
- Interface Control
- Arbitration
  - Not shared
  - Direct connection to PCI bus arbiter
- Error lines
PCI Commands

- 4 bits of control signals encode the command
  - I/O Read and Write
  - Memory Read and Write
  - Memory Read Multiple
  - Configuration Read and Write
  - Interrupt Acknowledge
PCI Protocol

• Basic bus transfer mechanism is a burst
• A burst is one address phase followed by one or more data phases
• Three primary signals
  – FRAME# - driven by master to indicate start of transaction
  – IRDY# - driven by master when ready to transfer data
  – TRDY# - driven by target when ready to transfer data
PCI Read Timing Diagram
PCI Write Transaction
PCI Bus Arbitration

- Access-Based
- Central Arbitration
- Fairness in Arbitration
  - Each requester is guaranteed access
  - But not necessarily equal access
- Master requests bus by asserting REQ# signal
- Arbiter grants bus by asserting GNT# signal
PCI Bus Arbitration
PCI Bus Parking

- Master can leave REQ# signal asserted
- Arbiter can leave GNT# signal asserted
- Avoids bus arbitration cycle
- In most cases, there is only a single master so there is no need to do arbitration
On-Chip Buses

- Used in
  - FPGAs
  - SOCs
- AMBA (Advanced Microprocessor Bus Architecture)
- CoreConnect
- Wishbone
On-Chip Buses

• How do you connect from the FPGA to the outside world?

• Interface logic
  – Glue logic that connects your system to the external bus - PCI, ISA, etc.
  – Requires that your circuitry understands the external bus
On-Chip Buses

• Bridge logic
  – Provides a mapping from internal bus (AMBA, Wishbone, etc.) to external bus
  – Less efficient in terms of area
  – Allows internal logic to use a higher performance bus
VHDL for Buses

- Opencores.org has a Wishbone to PCI bridge controller
- CoreConnect available as an IP core for Xilinx devices
- Altera uses AMBA bus in Excalibur processor core
Lab 4

entity bus280module
port ( clk : in std_logic;
    reset : in std_logic;
    addr : inout std_logic_vector(31 downto 0);
    data : inout std_logic_vector(31 downto 0);
    ireq : out std_logic;
    ignt : in std_logic;
    irdy_i : in std_logic;
    irdy_o : out std_logic;
    trdy_o : out std_logic;
    trdy_i : in std_logic;
    tselect: in std_logic;
    write : inout std_logic;
);

ECE 280 Digital Design Laboratory
Fall 2005 - Lecture 4 - 09/22/2005
Lab 4

entity bus280controller
port ( clk : in std_logic;
      reset : in std_logic;
      addr : inout std_logic_vector(31 downto 0);
      data : inout std_logic_vector(31 downto 0);
      ireq : in std_logic_vector(N-1 downto 0);
      ignt : out std_logic_vector(N-1 downto 0);
      irdy_o : in std_logic_vector(N-1 downto 0);
      irdy : out std_logic;
      tselect : in std_logic_vector(N-1 downto 0);
      trdy_o : in std_logic_vector(N-1 downto 0);
      trdy : out std_logic;
      write : in std_logic
    );
Lab 4

CLK
IREQ0
IREQ1
IGNTO
IGNT1
IRDY0
IRDY1
ADDR
DATA
WRITE