1.   Verilog Testbench  (Coding is required):

      myf.v, cir.v & cir\_tb.v, demo.v & demo\_tb.v

     Task used in Verilog testbench;

     Be able to write data into file, and read data from file

     using Verilog Testbench. You need to able to write your Verilog testbench.

     Be able to use task and obtain displayed testing results

     using Verilog testbench.

    You should be able to derive the displayed results by reading

     testbench code too!

2.   Knowledge of Design Principles for Modern Computers

All instructions directly executed by hardware

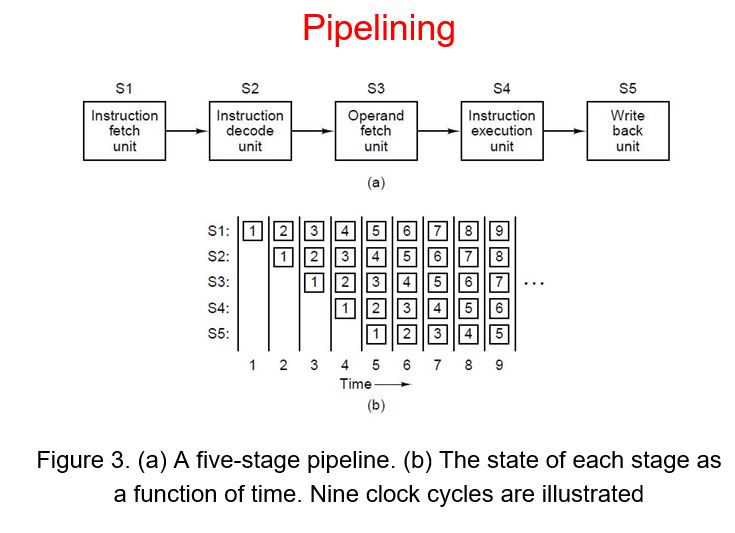
Maximize rate at which instructions are issued

Instructions should be easy to decode

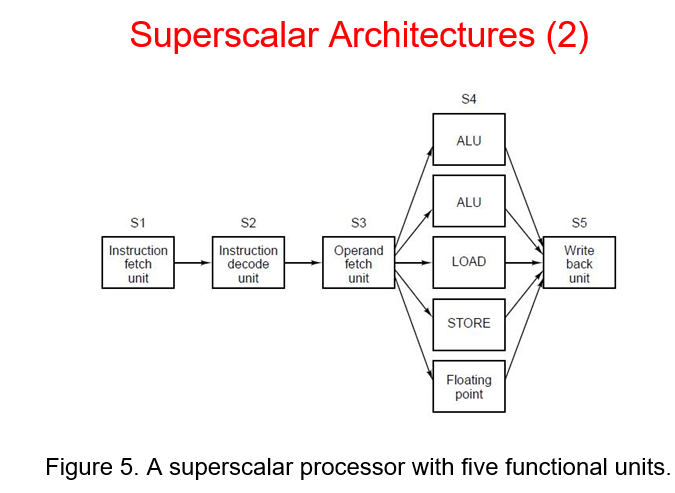
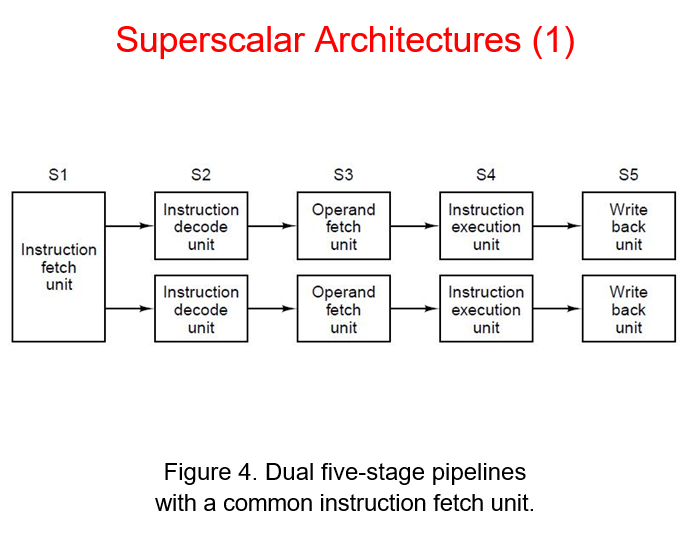
Only loads and stores should reference memory

Provide plenty of registers

3.   Knowledge of five stage of pipelining used in microprocessor design



4.   Knowledge of superscalar architectures

5.   knowledge of PCI bus width, bus speed, and maximum data transfer rates.



Bus width - determines how much data can be transmitted at one time

Bus speed - refers to how much data can move across the bus simultaneously

MB/sec – megabit per second

6.   Understand the meaning of the following PCI signals:

     AD, C/BE, PAR, FRAME#, TRDY#, IRDY#, STOP#, DEVSEL#, CLK, RST#

CLK - Clock provides timing for all transactions on PCI and is an input to every PCI device. All other PCI signals, except RST#, INTA#, INTB#, INTC#, and INTD#, are sampled on the rising edge of CLK and all other timing parameters are defined with respect to this edge. PCI operates up to 33 MHz (refer to Chapter 4) or 66 MHz (refer to Chapter 7) and, in general, the minimum frequency is DC (0 Hz); however, component-specific permissions are described in Chapter 4 (refer to Section 4.2.3.1.).

RST# - Reset is used to bring PCI-specific registers, sequencers, and signals to a consistent state

AD - Address and Data are multiplexed on the same PCI pins. A bus transaction consists of an address2 phase followed by one or more data phases. PCI supports both read and write bursts. The address phase is the first clock cycle in which FRAME# is asserted. During the address phase, AD[31::00] contain a physical address (32 bits). For I/O, this is a byte address; for configuration and memory, it is a DWORD address. During data phases, AD[07::00] contain the least significant byte (lsb) and AD[31::24] contain the most significant byte (msb). Write data is stable and valid when IRDY# is asserted; read data is stable and valid when TRDY# is asserted. Data is transferred during those clocks where both IRDY# and TRDY# are asserted.

C/BE - Bus Command and Byte Enables are multiplexed on the same PCI pins. During the address phase of a transaction, C/BE[3::0]# define the bus command (refer to Section 3.1. for bus command definitions). During the data phase, C/BE[3::0]# are used as Byte Enables. The Byte Enables are valid for the entire data phase and determine which byte lanes carry meaningful data. C/BE[0]# applies to byte 0 (lsb) and C/BE[3]# applies to byte 3 (msb)

PAR - Parity is even parity across AD[31::00] and C/BE[3::0]#. Parity generation is required by all PCI agents. PAR is stable and valid one clock after each address phase. For data phases, PAR is stable and valid one clock after either IRDY# is asserted on a write transaction or TRDY# is asserted on a read transaction. Once PAR is valid, it remains valid until one clock after the completion of the current data phase. (PAR has the same timing as AD[31::00], but it is delayed by one clock.) The master drives PAR for address and write data phases; the target drives PAR for read data phases

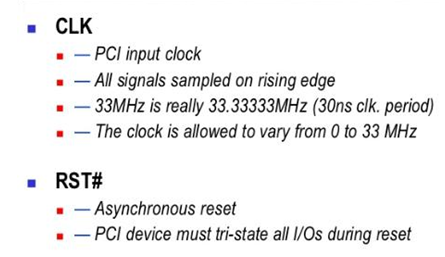
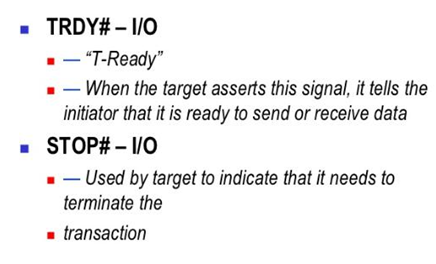
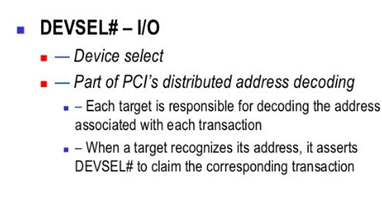
FRAME# - Cycle Frame is driven by the current master to indicate the beginning and duration of an access. FRAME# is asserted to indicate a bus transaction is beginning. While FRAME# is asserted, data transfers continue. When FRAME# is deasserted, the transaction is in the final data phase or has completed

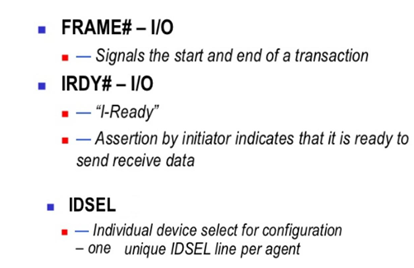
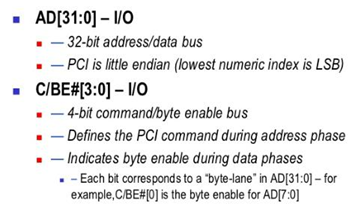
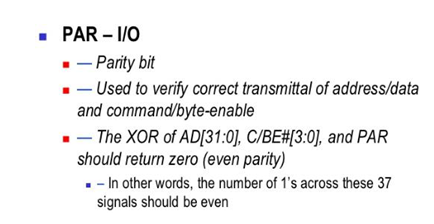
IRDY# - Initiator Ready indicates the initiating agent’s (bus master’s) ability to complete the current data phase of the transaction. IRDY# is used in conjunction with TRDY#. A data phase is completed on any clock both IRDY# and TRDY# are asserted. During a write, IRDY# indicates that valid data is present on AD[31::00]. During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together.

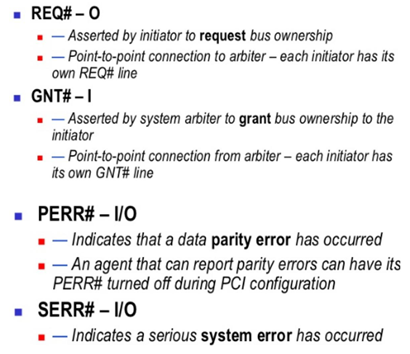
TRDY# - Target Ready indicates the target agent’s (selected device’s) ability to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is completed on any clock both TRDY# and IRDY# are asserted. During a read, TRDY# indicates that valid data is present on AD[31::00]. During a write, it indicates the target is prepared to accept data. Wait cycles are inserted until both IRDY# and TRDY# are asserted together.

STOP# - Stop indicates the current target is requesting the master to stop the current transaction.

DEVSEL# - Device Select, when actively driven, indicates the driving device has decoded its address as the target of the current access. As an input, DEVSEL# indicates whether any device on the bus has been selected.

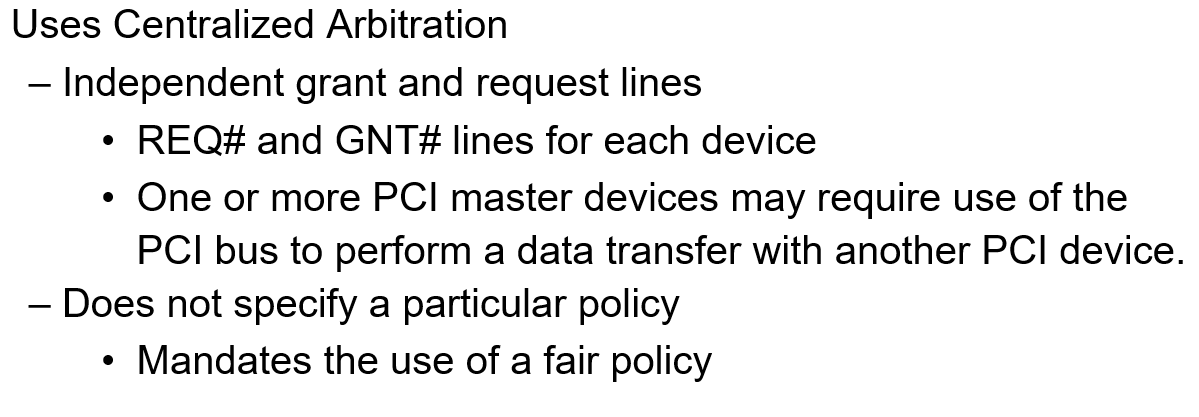
  



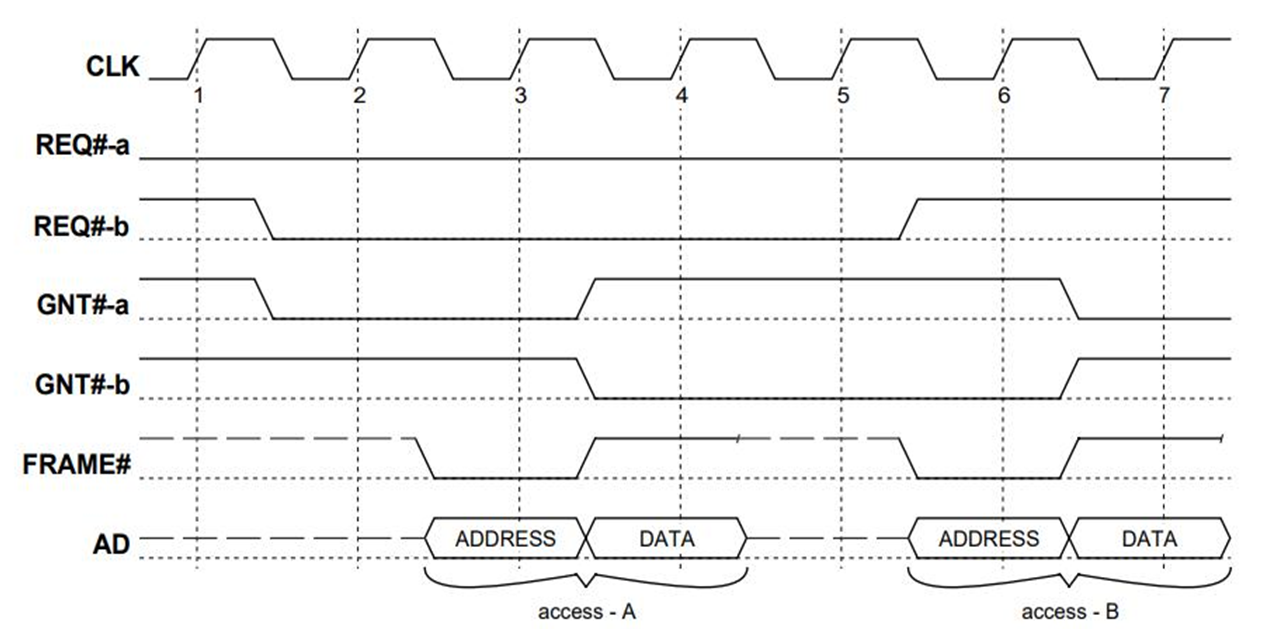
7.  Detailed Knowledge of PCI Bus Arbitration:

    Explain what centralized arbitration is;

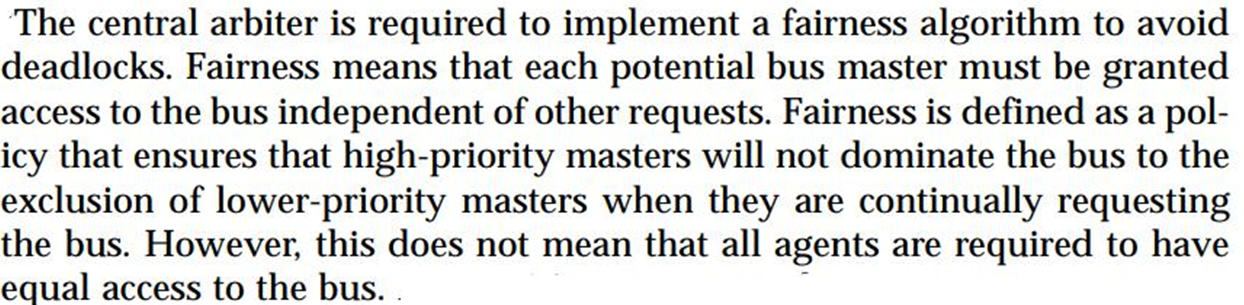


Centralized arbitration - A single bus arbiter performs the required arbitration.

    Be able to explain Basic PCI arbitration form;

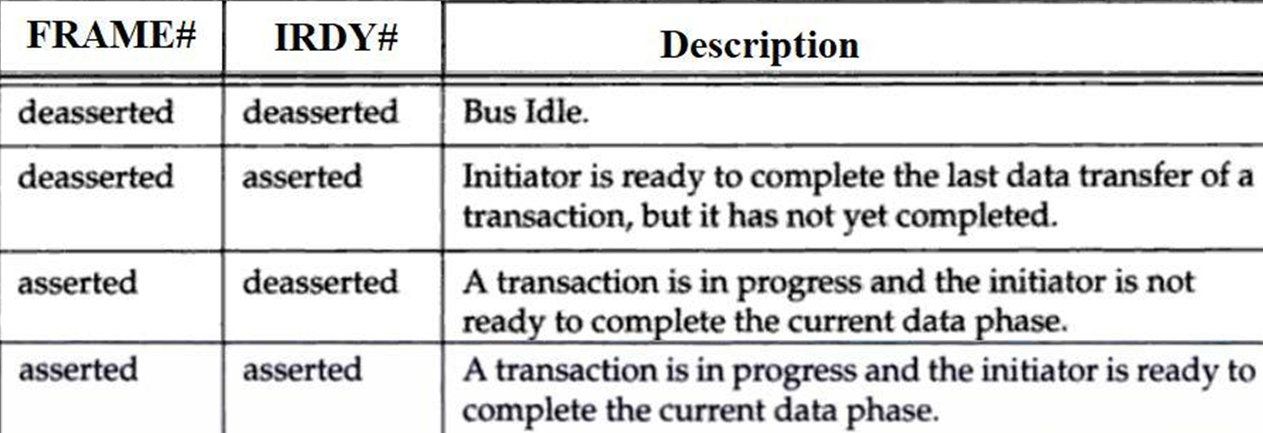


    Be able to explain what fairness means for PCI arbitration;



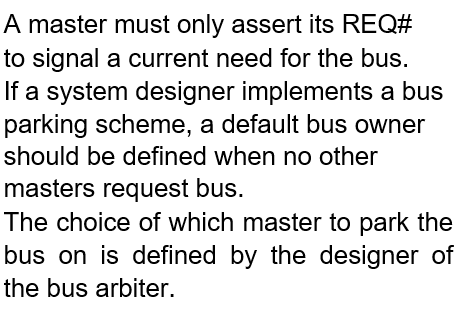
    Four different PCI bus states can be defined based on FRAME# and IRDY#

    values. What are they? What does each state mean?



    Be able to understand PCI Bus arbitration waveforms among multiple masters.

    Understand the meaning of PCI Bus Parking.



    Understand the PCI arbitrator request/grant timing requirement (Textbook page 67).

    Understand the meaning of hidden arbitration.

The PCI scheme allows bus arbitration to take place while the current initiator is preforming a data transfer. If the arbiter decides to grant ownership of the bus for the next transaction to a master other than the initiator of the current transaction, it removes the GNT# from the current initiator and issues GNT# to the next owner of the bus. The next owner cannot assume bus

8.  Understand the definition of the following latency:

    Bus access latency, arbitration latency, bus acquisition latency,

    initiator and target latency.

Bus Access Latency – Defined as the amount of time that expires from the moment a bus master requests the use of the PCI bus until it completes the first data transfer of the transaction. In other words, it is the sum of arbitration, bus acquisition and target latency.

Arbitration Latency – Defined as the period of time from the bus master’s assertion of REQ# until the bus arbiter asserts the bus master’s GNT#. This period is a function of the algorithm, the master’s priority and whether any other masters are requesting access to the bus.

Bus Acquisition Latency – Defined as the period time from the reception of GNT# by the requesting bus master until the current bus master surrenders the bus. The requesting bus master can then initiate its transaction by asserting FRAME#. The duration of this period is a function of how long the current bus master’s transaction in progress takes to complete. This parameter is the larger of either the current master’s LT value (in other words, its time slice) or the longest latency to first data phase completion in the custom (which limited to a maximum of 16 clocks).

Initiator and Target Latency – Defined as the period of time from the start of a transaction until the master and the currently-addressed target are ready to complete the first data transfer of the transaction. This period is a function of hoe fast the master is able to transfer the first data item, as well as the access time for the currently-addressed target device (and is limited to a maximum of 8 clocks for the master and 16 clocks for the target).

    What is the timing requirement to prevent master from monopolizing the bus?

    What is Master Latency Timer (LT)?

    Explain how Master latency timer (LT) works.

    Can LT Value Be Hardwired? (Textbook Page 80).

    Yes, for a master that performs one or two data phases per transaction, but the

    hardwired value may not exceed 16 (and it could be zero). Please refer to the

    previous sections regarding the implication if you choose to hardwire a value of

    zero.

    What is the timing requirement to prevent target from monopolizing the bus?

9.  Be able to explain single data phase read transaction waveform;

    Be able to explain burst data phase read transaction waveform;

    Be able to explain single data phase write transaction waveform;

    Be able to explain burst data phase write transaction waveform;

    Be able to discuss performance during read transaction:

    1). back-to-back single data phase read transfers

    2). the achievable burst transfer rate during the second through the last data read phases

    Be able to discuss performance during write transaction:

    1). back-to-back single data phase write transfers

    2). the achievable burst transfer rate during the second through the last data write phases

10. Be able to draw single data phase read transaction waveform;

    Be able to draw burst data phase read transaction waveform with wait states inserted;

    Be able to draw optimized burst data phase read transaction waveform without wait states inserted;

    Be able to draw single data phase write transaction waveform;

    Be able to draw burst data phase write transaction waveform with wait states inserted;

    Be able to draw optimized burst data phase write transaction waveform without wait states inserted;