CPE 186

RVR 5022

[Vadhva@csus.edu](mailto:Vadhva@csus.edu)

10 Quizzes (lowest will be dropped)

1 Paper / 1 Project

Notes in the book

Intro to PCI

PCI bus is a local computer bus for attaching hardware devices in a computer

Supports same functions as processor bus but independent of processor.

PCI supports up to 80 PCI functions per PCI bus

Support for up to 256 PCI buses

Bus speed: 33MHz and 66 MHz

64 bit support

Access time of 60 nsec

Low Pin count: Target 47 and master 49 pins.

Three address spaces: Memory. IC. Configuration

Burst Transfer

Address phase transition type, data phase.

Initiator, Target and Agent

Single vs Multi Function PCI Devices (Each function contains own address space)

PCI Bus Clk

Must support 0 – 33Mhz Clk or 0 to 66Mhz Clk

9/8 Lecture

Find a book on cache memory, topic. Any book in computer Architecture.

Quiz – Describe things, (10-15 minute quiz)

Chapter 2 and 4. PCI Signal groups and Arbitration.

Signal Groups and Timing Diagram.

Two main groups:

Required and optional signals.

Categories of signals

AD/Command/ 64 bit support

Interface Control

CLK/System/Error reporting

Arbitration

Interrupt / JTAG

Power management

Anything beyond 4gb requires Extension of PCI Bus

Address is multiplex for 32 bit, 64 bit of addressable bits (2 data cycles)

><><>< (32 bits)

First address base, 2nd address base (2 address cycles)

><><><><><

Data bus is 64 bits (Data phase, full 64 bits of data, 2 clock cycles instead of 1)

Not 64 bit because some legacy devices might only be 32 and chose to use 2 addresses.

Frame is start of the cycle.

Par doesn't matter in address/data and command or interface

Target Ready is asserted by target (decodes and says yes this address is for me)

IRDY (initiator asserts, master in this case. Only asserts when ready to transfer)

STOP (target not capable)

DEVSEL(when target is ready issues command when ready. Device #4 located in 40005000, address is issued by master and once master issues address, all devices connected into address starts decoding. Surrogated might be decoded. 40005000 comes and says this device is for me.)

In the extension

(Extra bits)

AD[32-63) extension portion

Par64 (means another parity attached to extension portion)

Req64 and Ack64 – Devices comes around and master asks for its concern, all it knows are issue commands. It does not know what devices are where in system. Master job to manipulate and plant things

Master doesn't know if it is communicating with 16,32,64 bit device. Master issues req 64 and it must receive ack64 to know if one is 64 bit device. If ack64 is asserted then frame is 8bytes. Target must increment address by 8 every transfer. Otherwise incremented by 4.

Arbitration signals

In PCI bus, no one owns the bus. So a req must be made by master, device that wants to communicate, must receive a GNT. Is an advance information. Tells master (yes u will be next bus master but you have to monitor the bus to see when current bus master removes itself.)

GNT must be asserted, Frame (negated) and IRDY (negated) for it to take over the bus. The master has the right to take over and keep the bus for duration of time it needs. No one can take unless commands are done again.

GNT can be asserted for as long as it needs. GNT asserted all time means bus is asserted at this master.

Clock can be 0-33MHz. All inputs to PCI devices are sampled on the rising edge of Clk

All actions are synched to the PCI Clk

Clk frequency may be changed at any time as long as

Clk edge may remain clean

Min Clk hig and low times are not violated

There are no bus requests outstanding

LOCK# is not asserted.

DEVSEL#

Fast devices – asserted during 1st Clk following address phase

Med - 2nd Clk

Slow -3rd Clk

Legacy Bus Transactions (ISA) Subtractive decoding – If unclaimed is asserted during 4th clk following address phase. (Done by bridge itself)

Subtractive decoding

Address decoding

Method 1: target embedded within the bridge

Method 2: ??

Frame being asserted means burst data cycle. Doesn't know if it can accept burst data. Looks at data and tries to see if data is taken based on command it was issued. TRDY being negated means data 2 will not be taken. Once 2nd data is taken then master says i'm not able to transfer so it negates the transfer. Initiator repeats asserting. Frame identifies that the last data phase cab occur. When TRDY asserts itself, the last data phase occurs and at end of data cycle, master relinquishes the bus.

Arbitration (Ch 5)

Arbiter is a circuitry in system, external to bus master. Arbiter looks at few things, looks at priority of device. Might have higher priority. Some have lower priority. Short time slices given vs a long time slice.Input to the master and output to the arbiter. Arbiter's job is to make decision. Not everyone in PCI bus is created equal. Group of masters that are Master A, master B.

State Machines

A,B,X,A,B,Y,A,B,Z (Make a state machine for next time)

Highest priority A, B 2nd, C is lowest.

A,B,A,B,C,A,B,A,B,C

9/17

State machine review & ch 6 stuff

9/24

Should not be susceptible to the noise in the environment. Pollution (noise)

PCI Master and Target Latency.

Latency – set of rules in a way

PCI Agent latencies

Mandatory Delay before the first transactionsBus Access Latency

Arbitration latency – Grant might be issued but another bus master has the bus.

Bus Acquisition Latency

Initiator & Target Latency-Target might insert multiple wait states, bus master can insert multiple wait states.

Latencies – possible delays

Dat phase: Master: IRDY

Final Data phase Master: IRDY

Data Phase:Target: TRDY#

Subsequent data Phase: Target

Rules:

Master must transfer data within 8Clks

IRDY# must be de-asserted after final data

Making sure one master does not monopolize the Bus

Latency Timer (LT)

How it works – once LT expires, the master only has 1 more transfer allowed. Must relinquish power after.

Is implementation of LT mandatory?

Can LT Value be hardwired

Time Slice.

Treatment of Memory Write and invalidate.

Target must transfer data expeditiously

The first data phase rule

(Target issues retry)

Masters response to the retry

Sometimes Target can't transfer first data within 16Clks

Target frequently can't transfer first data within 16 Clks

Two exception to the first data phase rule

Target could read 16 Clk rule

The subsequent data phase rule

In data phase & cannot transfer data in 8 Clks

Method 1: Disconnect without Data

Method 2: Target Abort

Ok in this data phase but can't meet rule in next data phase: Disconnect with data

Master's response to a disconnect

Target's subsequent latency and Master's latency

Two exceptions to first data phase rule

During system initialization time

Host/PCI Bridge that is snooping

Is permitted to exceed 16 clk limit to maximum 32 Clks

Delayed Transactions:

The problem of delayed transactionsbuThe solution for delayed transactions

Information memorized

Master and target actions during Delayed transactions

Commands that can use delayed transactions

Interrupt acknowledge, I/O, Read/Wire

Request not completed and targeted again

special cycle monitoring while processing request

Discard or delayed requests

Multiple delayed requests from Same Master

Request queuing in target

Discard f delayed completions

Read from preferable memory

Main problem: issues we know about, so much delay once request is made til bus is acquired til data is transferred (Think of taking bus, or show up 10-15 mins before appointment, get your information, then wait then called, take some of your vitals, then wait another 30 mins for doctor. Thats how latency works

Cache controllers, DMA (Read if we have time about it)

Next topic (PCI Commands) Chapter 7 of the book

PCI commands

Different commands and stuff.

PCI Bus arbitration:

When PCI Bus master requires use of PCI bus to perform data transfer it must first request the use of the bus from the arbiter.

At any given time, one or more PCI bus master devices may require use of the PCI bus to perform a data transfer with another PCI Device. Each request master asserts its REQ# output to inform the bus arbiter of its pending request for the use of the bus.

In order to grant the PCI bus to a bus master, the arbiter asserts the devices respective GNT signal. This grants the bus to the master for one transactionsIF a master generates a request, is subsequently granted the bus and does not initiate a transaction (assert Frame#) within 16 PCI clocks after the bus goes idle.

Assume the following conditions:

Master A is the next to receive the bus in the first group.

Master X is the next to receive it in the second group.

A master in the first group is the next to receive the bus.

All masters are asserting the REQ# and wish to perform multiple transactions.

The order in which the masters would receive access to the bus is:

1. Master A
2. Master B
3. Master X
4. Master A
5. Master B
6. Master Y
7. Master A
8. Master B
9. Master Z
10. Master A
11. Master B
12. Master X

Master and Target Latency

Describes the rules governing how much time a device may hold the bus in wait states during any given data phase. IT describes how soon after reset is removed the first transaction may be initiated and how soon after reset is removed a target device may use to meet the latency rules are described: Delayed Transactions and posting of memory writes

There is a mandatory delay before first transaction is initiated. The 2.2 spec mandates that the system must guarantee that the first transaction will not be initiated on the PCI bus for at least five PCI clock cycles after RST# is de asserted. This value is referred to as Trhff in the spec (Time from Reset High to First Frame assertion)

When a bus master wishes to transfer a block of one or more data items between itself and a target PCI device, it must request the use of the bus from the bus arbiter. Bus access latency is 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.

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 arbitration algorithm, the master's priority and whether any other masters are requesting access to the bus.

Bus Acquisition: 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 or the longest latency to first data phase completion in the system (limited to 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 how 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.

Master asserts REQ# -------> Master receives GNT# ---------> Master asserts Frame# ------> Initiator

Arbitration Latency Bus acquisition Latency Target and initiator latency

asserts IRDY# and Target asserts TRDY#

Difference between read/write, Changes direction, there is a turnaround cycle if read, no turnaround cycle if write.

Turn around cycle - there as terminal bus has to change direction. Data is coming to the unit, has to go bi directional, needs time to change buffers.

Single data phase is just one file transfer Burst data phase read is multiple transfers.

Reason for turn around cycle is to allow time for output buffers to change direction.

If he asks about timing diagram they are basically the same thing.

Computer Organization ECE 232 University of Massachusetts

System Layers, application software, system software: Compiler translates HLL code to machine code.

Operating system: Service code, handling input/output, managing memory and storage

Hardware: processor, memory, I/O Controllers.

High level language

Processors – [control unit, arithmetic logical unit, registers][main memory][dusk][printer]

Computer System Organization

All figures from Structured ComputeR Organization (5th edition)

Quiz Thursday. Chapters 2 and 4 from book.

Everything must tell the computer the work to do. Signal must be given to particular register. Very dependent on device being used

We need to look at specs and satisfying setup to store data. Increase clock speed for improved performance. Makes transistors smaller, work harder (like OC'ing a cpu)

Pipelining – assembly language.

Branch always moves forward unless stated.

Keeping cache filled. Pipelining increases performance, fast and large amounts of cache are needed.

Cache Memory Concepts:

The memory of the system is Drams, Srams require more space and power.

Problem and Solution

Speed of UP and DRAM

SRAM can probive 0-ws performance

But – expensive, larger in size, consumes more power, generates more heat.

Temporal Locality

Same instruction to be fetched frequently

Older information less likely to be used

Programs that don't access same information more than once; no improvement

Spatial locality

Programs and data reside in consecutive memory locations

Programs need data/code close/ adjacent to locations already in use.

Performance

Cache performance – (cache hits/total memory requests) \* 100%

Fast access to memory subsystems

Improves bus utilization (% of time Bus Master occupies system bus)

Components of cache subsystems

\*Cache memory (Data cache RAM)

Cache management logic

compares mem - addr with mem directory locations

Enables cache memory to output info from addressed location.

Cache memory directory (called tag: RAM)

has listing of all memory addresses that have copies in cache memory

Each location set has corresponding entry in cache directory

Example

size of cache 1k

Size of main memory is 1Mb, how big is main memory when compared to cache, 1000x

You can divide main memory into number of pages.

Memory location of 100 of same memory cuz we brought in 100 pages 2^10 = 1024

Direct map concept

correspondance 1 to 1 from cache to dividing segments.

In direct map what has to be done, when 5 is accessed, throw out page 0 then throw out page 5 and replace it with page 0

Too much overhead.

Write paper group of 2 or by yourself, requirements are the same. Proposal needed, review of it. Should be related to computer architecture. New processors? Different types of busses? 15 Pages. Computer hardware security issues.

Do not cut and paste, will be checked on turnitin.

4 weeks (before thanksgiving) Thursday 19th, November.

Proposal 2-3 References. Next Thursday.

GPU architecture, difference between blower style vs acx 2.0 vs water cooling

Skylake vs Haswell vs Broadwell

Direct mapped. One way set associative

Cache controller identifies location of information in cche based on, the location occupies in cache, page number contained in cache directory, lower portion of address identifies set: which directory entry much be checked. Upper portion of address is compared with directory entry to match

Two Way set Associative (Cache system)

Compares page number given by page address by location of the cache and sets up comparison. Valid bit must be set and both are comparing things with page number in the address. Either Cache hit or miss.

No hit in the cache at all which means that the only copy resides in the main memory, nowhere else. The new access is requested so you should go and bring it to one of the sets.

Cache controller has choice of 2 cache banks to store data, each has own cache directory, pages are now have the page (direct mapped)

4 way set associative

Now cache controller has choice of 4 cache banks to store data. (4 way) possible to have a cache miss, how do you replace it? We have to use 2 bits now.

LR

11 ---> 00 (A-hit) 01 00 (a hit) 00 (a hit) 00 (a hit)

01 ----> 10 00 (B hit) 01 10 11

10 ----> 11 11 11 11 11

00---> 01 10 11 11 11

Each cache has down directory, pages are now 1.4 of page size in directed mapped, same size of cache

Requires 4 entries to be checked

Look up penalty increases

Multiple way comparator used to compare target page address to all selected tags simultaneously (4 way compares target page number to 4 selected directory entries)

Increasing ways eventually leads to fully associative structure

Block/Line

Right policies of cache

Cache is very important

Comparison of 4 way set associative must be done at once.

LRU (Last register used)

Cache and main memory are consistent and same memory exists in both

Disadvantage: Slow, every time we modify something we need to go to the main memory. (Right Through)

Make it a little faster, cache controller, creates a buffer that writes into the cache and buffer at the same time.

Problem, buffer is full, needs to have signal that buffer is empty, data was successfully written in main memory, disadvantage breaks down if it has a single buffer, still breaks down with more buffers, (Buffer right through)

Cache updated while memory is not

memory updated which cache is not

Handled through write policies

Write thro

Buffered write thro

Write back

Writes back to memory cached data status

Write through

Pass each write to main memory, write through cache generally uses this, main memory has latest data always

Simple effective but poor performance (slow access to main memory)

Buffered write through

posted write to main memory

fools processor by 0-ws writes

cache controller stores entire write into buffer, completes it later.

For B2B writes, WS inserted in second write until 1st one is done, 2nd write is then posted and UP's ready# asserted.

Other masters not permitted to use bus till write through is completed.

Poor bus utilization

need latest data only when someone access it

unnecessary to update memory each time

Write-Back

Updates memory only when needed

Better bus utilization

Cache locations marked modified when updated.

Snooping required for reads/writes to memory location

Complicated and costly

Snooping, watch the cache to see if it has something I have or I want something they have. Watches traffic only.

MESI

M – Modified (means that the only copy is only present in particular cache, no one else has it)

E - Exclusive

S – shared

I – invalid

CPU 1 CPU 2

MESI MESI

1011 1101

1101 (Sends SHR) 0011

1110 (makes it invalid) 1101

1101

Bus based shared memory organiztion

CPU/cache ------x3 times --- shared memory all connected via shared bus

Bus is simple physical connection (wires)

Bus bandwidth limits no. of CPU's

could be multiple memory elements

Problem of memory Coherence

Assume just single level cache and main memory

Processor writes to location in its cache

other caches may hold shared copies these, will be out of date.

Updating main memory alone is not enough.

Example

Newest copy is only available in processor 1.

Takes into action Bus Snooping.

Scheme where every cpu has a copy of its cached data is too far too complex.

Write invalidate:

CPU wanting to write to an address, grabs a bus cycle and sends a write invalidate message. All snooping caches invalidate their copy of appropriate cache line

CPU writes to its cached copy.

Any shared read in other CPU's will not miss in cache and re-fetch new data.

ARTY# (Address retry)

<----->

<----->

SHR# (Shared)

Invalid means cache doesn't have right value. Period.

Write-miss - Data is not in cache to start with but they want to write something.

Write Through:

Pass each write to main memory

Write through cache generally use this

Main memory always has latest data

Simple, effective but poor performance (Slow access to main memory)

Buffered Write Through:

Posted write to main memory

Fools processor by 0-ws writes

Cache controller stores entire write into buffer, completes it later.

For B2B writes, WS inserted in second write until 1st one is done, 2nd write is then posted and UP's READY# asserted

Other masters not permitted to use bus till write through is completed

Poor bus utilization

Needs latest data only when someone accesses it

Unnecessary to update memory every time

Write-Back

Updates memory only when needed

Better bus utilization

Cache locations marked modified when updated.

Snooping required for reads/writes to memory locations

Complicated and costly

LRU

Keeps track of cache locations within set that have at least been least recently used (temporal locality; info not recently used is less likely to be used)

Direct mapped (One way set associative)

Cache controller identifies location of information in cache based on:

Set position that location occupies in cache (Set where mem location comes from within a page)

Page Number contained in cache directory (Page number the location comes from in the main memory)

Lower portion of address identifies set: which directory entry must be checked

Upper portion of address is compared with directory entry to match

Two Way Set associative

Cache controller has choice of 2 cache banks to store data (2 ways)

Each has own cache directory

Pages are not half of page size in direct mapped

4 way set associative

Cache controller has choice of 4 cache banks to store data (4 ways)

Each has own cache directory

Pages are now ¼ of page size in direct mapped, same size as cache

Requires 4 entries to be checked

Loop up penalty increases

Multiple way comparator used to compare target page address to all selected tags simultaneously (4 way compares target page number to 4 selected directory entries)

Increasing ways eventually leads to fully associative structure

Fully associative

optimum structure for ensuring hits

Highest performance

views memory as single page/block regardless of its size

Any memory location can be copied to any cache location: set

Cache directory stores entire addressableRequest memory location must be in only one cache location

Each memory access requires that address be compared against all entires of cache directory

High look up penalty

Small size (<= KB) due to high lookup penalty

Cache line

Line size from 4 bytes to 128 bytes

Example: Cache to handle 4G of address

4kB page size

Number of pages: 1M

Cache directory: 20 bits (to address 1M pages)

Each set stores 4 bytes: total 4KB

Total Tag-ram size needed: sets\*directory size (20bits)

Tag ram is considered overhead on 4KB cache

Data bus width limits cache line size. Larger cache line doesn't mean better performance (2 memory cycles if data bus 32 bytes, but line is 64 bytes)

Storing data that may not be useful

ADV: smaller directory overhead

Block (sectored cache)

Consecutive addressed lines that share same cache directory entry

Each sector in sectored cache consists of 1 or more lines

Lines comes from adjacent locations in main memory (should have same page number in cache directory)

Still have state bits for each line: line is still the smallest unit that cache can track)

Bus snooping:

Bus master(BM) writes location in main memory

Cache and memory not coherent

Write through:

invalidates line in cache

Miss on next access, data fetched from main memory

Write back:

Push back entire line to memory

Forces BM to abort bus cycle

Modified data is in cache updates main memory with data in cache

BM then writes its changes to main memory

Bus snooping required to check if BM is writing to memory: if snoop hit, then it captures data from bus, data is updated in both cache and main memory (both write back and write through)

Bus snooping required for BM

Location read has latest data in cache and not in memory

Snoop hit on modified line detected: BM cycle suspended(back off) till cache updates memory

BM cycle now allowed to access memory.

Lecture (11/3)

Cover MESI proto calls a little bit more today. (Look at notes he provided. Added stuff he said verbally)

Bus based shared memory organization

Arbitrations involved with it also requires time on the bus is also a negative.

Organization. Increase bus bandwidth, don't have to use it as much and if constantly using the bus, system slows down.

Problems of memory coherence

Snooping mechanism, watch traffic over the bus.

Data becomes invalid data.

When they all change memory, the shared memory should update their cache

Back to back writes is a problem with a buffered write through

Bus may not be available.

When all written, main memory seems to also have right memory

Snooping watches the data, grab the data for yourself and make sure you wright thru policies,

Once CPU 2 grabs the data, then CPU1 changes the data, CPU2 wants it, CPU1 sends a signal Address retry, what u are trying to get is not available to it. CPU2 has to wait, CPU1 dumps the cache in the main memory, it knows CPU2 will get the data and makes it as shared. Important to invalidate the data.

MESI Protocol: Modified, Exclusive, Shared and Invalid.

Exclusive: Cache line is the same as main memory and is the only cached copy.

Shared means that everyone has same value of data

MESI used in write through.

Invalid – line data is not valid.

Basically matching addresses, looking at them. Cache line is usually in one of those states.

Read hit – data is in cache

Read Miss – read data and it's a miss (Options not bringing it to cache and just reading it.

Write Hit – Once ur writing to a hit it is changed to modified

Write miss- Data is not in cache already and want to write to particular place. Not bringing it to cache.

Mesi Local Read Miss – No other copy in cache, Processor makes bus request to memory, possible data was modified in other cache.

Read with intent to modify:

What happens if you have cache when implementing a cache read with intent to modify:

Other caches know that even though at this moment one of the CPU's will modify the data, with that type of instruction they will try and invalidate the cache. It will read from memory to local cache – bus transaction marked RWITM.

November 12th, Santa Clara at 12:00pm

Early Transaction: Chapter 12 of PCI

Master initiated termination

4 scenarios:

1. Transaction completed normally
2. Time Slice expire and it is preempted
3. Preempted during time slice, then uses up its allotted time slice.
4. No target has responded

Preemption during time slice.

Master pre-empted

Master abort

Target does not claim transactions

Target issues disconnect, target does not support burst mode.

Memory sequencing is not allowed.

Quiz 11/12 review

MESI review

Invalid data contained is garbage and cannot be used

Modified – Indicates that its only copy of data in whole data, copy in main memory is stale

Exclusive – As modified state, only copy in data. Clean copy, same data as in main memory

Shared – indicates that multiple copies and clean, same data as main memory.

Core 0 asking for L1

I-Read going from either E (No other copies only in main memory) or S (if exist other copies)

If trying to write, the line will be brought into modified state. Core has written new value.

PCI Parity

Status bit name change

Intro to PCI parity

PERR# Signal

Data Parity

Data Parity reporting:

General

Master can choose not to assert PERR#

Parity error during Read

Parity error during write

PCI Device's Configuration command register

15-10 Reserved

9 = Fast back to back enable

8 = SERR# Enable

7 = Stepping Control

6 = Parity Error response

5 = Palette Snoop enable

4 = Memory write and invalidate enable

3 = Special Cycle

2 = Bus Master

1 = Memory Space

0 = IO Space

PCI Device's configuration status register

3-0 Reserved

4 = 2.2, Capabilities list, new in 2.2

5 = 66MHz-Capable

6 = 2.2 Was UDF Supported now reversed in 2.2

7 = Fast back to back capable

8 = Master data parity error

10 = DEVSEL Timing

11 = Signal target abort

12 = Recieved target abort

13 = Received master abort

14 = Signaled System error

15 =Detected Parity error

Data Parity Error reporting

Recovery by the master

Recovery by the device driver

Recovery by the operation system

Special case: Data parity error during special transactionsDevices excluded from PERR# Requirement

* + Chipset
  + Devices that don't with OS/application program or data

SERR#

Address Phase parity

Address phase parity Generation & Checking

Address phase parity error reporting

Method 1: Assert DEVSEL# and complete transaction normally

Method 2: assert DEVSEL# and do Target abort

Method 3: Do not assert DEVSEL# and let master do a master abort

Address Phase Parity

CLK – targets calculate expected parity during clock 2

Frame – line drops halfway through clk 1 and back up ¼ of new clock |

[AD 31:0] rectangular hexagon | Targets latch to the 3

C/BE#[3:0] – rectangular hexagon |

PAR – line halfway through first clock then rectangular hexagon into 2nd and 3rd Targets latch parity and compare to expected parity

SERR# - Drops halfway into clock 3 and then slow slope up (If parity error detected, target(s) assert SERR# for one clock. Pullup then returns it high-bandwidth

IRDY# has circle arrow drops halfway 2-3 then back up

System Errors:

General.

What causes system Errors

Address phase parity error

Data parity during special cycle

Master of MSI (Message Signal Interrupt) receives an error

Target Abort Detection

Other possible causes for system error

Interrupts

Three ways to deliver interrupts

Method 1: Legacy Method

Method 2: Multiprocessor System

Method 3: Message Signal Interrupts

Using Pins vs Using MSI

Single Function PCI Device

Interrupt signal bonded to | Value hardwired in pin register

Device doesn't generate interrupts | 00h

INTA# pin | 01h

INTB# pin | 02h

INTC# pin | 03h

INTD# pin | 04h

Multi-function PCI Device

Connection to INT x# Pins to system board traces

Interrupt routing

Routing recommendations

BIOS knowns interrupt trace layout

Well designed chipset has programmable interupt router

Interrupt routing information

Interrupt routing table

Interrupts – PCI interrupts are sharable, Hooking the interrupt

Interrupt Chaining

Step 1: Intialize all entries to point to dummy handler

Step 2: Initialize all entries for embedded Devices

Step 3: Hook entries for embedded device BIOS routines

Step 4: Perform expansion Bus ROM scan

Step 5: Perform PCI Device scan

Step 6: Load OS

Step 7: OS Loads and call driver's initialization code

Linked List has been built for each interrupt Level

Servicing shared interrupts

Example scenario

Both devices simultaneously generate requests

Processor interrupted and requests vector

First handler executed

Jump to next driver in linked list

Jump to dummy handler: Control passed back to interrupted program

implied priority scheme

Message Signaled interrupts

Introduction

Method 1: Interrupt PIN

Method 2: MSI

Advantages of MSI interrupts

Basics of MSI Configuration

Basics of generating an MSI Interrupt Request

MSI

How is the memory write treated by bridges

Memory write synced when interrupt handler entered.

-The problem

Old way of solving the problem

How MSI solves the problem

Interrupt latency

MSI are non shared

MSI capability register

Description of capability register

Capability ID

Pointer to next new capability

Message control register

Memory address register

Memory data register

Message write can have bad ending

Retry or disconnect

Master or target abort

Write results in data parity error

Electromagnetic interface (EMI)

Systems face problems with noise

Should take advantage of this course.

Within the equipment, there are oscilators, amplifiers, Mixers, Amplifier, Detector, Audio amplifiers

Noise can multiply itself

Noise has a path, it has to have a path to go through. Long cables are like antennas

External source of noise

Power lines with radio on

Motors, old tv's and radios, Vaccum cleaner next to radio.

External source can come from nature too, not necessarily man made.

Electromagnetic compatability (EMC)

Is the ability of the system to function properly in its intended electromagnetic environment, not be a source of pollution to that environment.

Emission/Susceptibility.

Emission means It creates noise for test of system. Not self controlled.

Susceptibility, put card into system, it doesn't work. Outside chasis it works well, inside it doesn't. It is not able to function in its intended environment.

Equipment development cycle

Regulations

FCC Regulations

Part 15 for radio frequency devices, Any electromagnetic energy in the frequency range of 10Khz 3GHZ

Part 18: Operational conditions for industrial, scientific, and medical equipment

Part 15: Subpart J regulations to control digital electronics and computing devices

Military Standard MIL-STD 461B and MIL STD-462

Digital Equipment Regulations

Class A: A computing device that is marketed for use in commercial industrial or business environment

Class B: A computing device that is marketed for use in residential environment anything but military

Class b device are more likely to be located in closest proximity to radio and television, that emission limits are about 10db more restrictive than class A devices.

Open field test site for FCC emission test

Noise Path-----

Three elements necessary to produce a noise problem

Noise source

Receptor circuit

Coupling channel to transmit the noise from the source to the reciever

Break the noise path

Noise can be suppressed at the source

The receptor can be made insensitive to the noise

Transmission through the coupling channel can be minimized

Noise path

Emission tests \_\_\_\_\_\_

Conducted – Antenna terminal, control and signal leads, lower peads

Radiated 0 Magnetic fields/ electric and electromagnetics

Susceptibility tests\_\_\_\_

Conducted – Power leads (RF/Spikes)

Reciever, antenna terminals

Noise paths are coupling channels.

More noise stuff, you can reduce the noise but you cannot eliminate it. There is interference that occurs.

Chapter 16, 17, 18, and the Bridge

Ch 17 Configuration Space

Power up process

Discovering devices on the system

Scanning the bus, walking the bus, probing the bus, discovery process.

Program that performs the PCI bus scan is referred to as Bus enumerator.

PCI Device vs PCI Function

PCI device may contain a single function or multiple function

Bit in one of a function's register's configuration register defines weather the package contains one function or more than one function.

Bus enumeration

Configuration software reads subset of configuration registers to determine

Presence of functions and types

Determine blocks of memory or I/O space required

Determine interrupt capabilities and requirements

Mastering capabilities, how often require bus access, what arbitration priority it requires

How long to maintain the bus

Three addressing spaces

Memory address space

I/O Address space

Configuration space

Quiz next tuesday ch 15/17 64 bit extension

Pci to pci bridge

Scalable Bus structure

Too many devices in the system

Performance might be a problem

Each master needs to grab device

Problem can be solved by adding additional pci buses to redistribute the device popularion

PCI to PCI bridge

forms a bridge to connect

Each bridge is connected to 2 pci buses

Primary/secondry bus

Terminologies used

Downstream

Downstraim

Primary

Secondary

Subordinate Bus

Functions of PCI bridge: Traffic Director

Detects different transactions on different buses, primary/secondary

Good to learn system verillog

Quiz stuff -------------------

64 bit extension signals

REQ64# is asserted by a 64 bit bus master to indiciate that it would like to perform 64 bit data transfers. REQ64# has the same timing and duration as the FRAME# signal. The REQ64# signal line must be supplied with a pullup resistor.

ACK64# is asserted by a target in response to REQ64# assertion by the master.

AD[63:32] comprise the upper four address/data paths

C/BE#[7:4] comprise the upper four command/byte enable signals.

PAR64 is the parity bit that provides even parity for the upper four AD paths and upper four C/BE signal lines.

64 bit card in 32 bit add in connector automatically only uses the lower half of the bus

64 bit extension is not in use when

PCI bus is idle

A 32 bit bus master is performing a transaction with a 32 bit target

a 32 bit bus master is performing transaction with a 64 bit target

A 64 bit bus master addresses a target to perform 32 bit data transfers.

A 64 bit bus master attempts a 64 bit dara transfer with a 32 bit memory target that resides below the 4GB boundary.

When 64 bit device is installed in a 32 bit PCI expansion slot the system board pullups on AD[63:32], C/BE#[7:4]. And PAR64 are not available to the add in card.

32 bit connector is difference from 64 bit connector as 64 bit has an extra portion of a bus slot.

Chapter 17

physical pci device may contain one or more separate PCI functions

Single function is a device that contains only one function

Multi function device is a pci device that contains more than one function.

The function contained in a single function device must respond as function zero when addressed n a type 0 PCI configuration read or write transaction.

In a multi function device, the first function must be designed to respond to configuration accesses as function zero.

Too many devices on the bus can cause problems like load issues and performance.

PCI bridges can solve the problem by redistributing the population.

PCI bus masters use PCI IO and memory transactions to access PCI IO and memory locations. A third access type, the configuration access is used to access a device's configuration registers. A PCI memory space is either 4GB or 2^64 locations in size. IF 64 bit addressing is utilized.

Tuesday December 8th Ch 15, and bridges. Quiz with chart

PCI Express Architectural perspective

Third generation high performance I/o Bus

PCI, AGP, PCI-X

PCI express way

A point to point interconnect.

Differential signaling.

Each same number of bits can be transferred simultaneously.

Impedance has to be matched with the driver when it is DC.

Low cost due to low pin count.

If bridge, has to go through multiple bridges to complete operation.

PCI Express

Packet Based protocol

Quality of Service (Qos) Capability of routing packers from different applications

Traffic Classes(TCs) numbers between 0 and 7 assigned by the device driver.

End pointsL Devices connected that can do something, memory device/ IO device.

PCI Express Transactions:

Memory

I/o

Configuration

Message

Acknowledge is last.

Transaction Layer packets- TLP