EECS 150 Spring 2004

Lab Lecture 11

Checkpoint4

4/9/2004

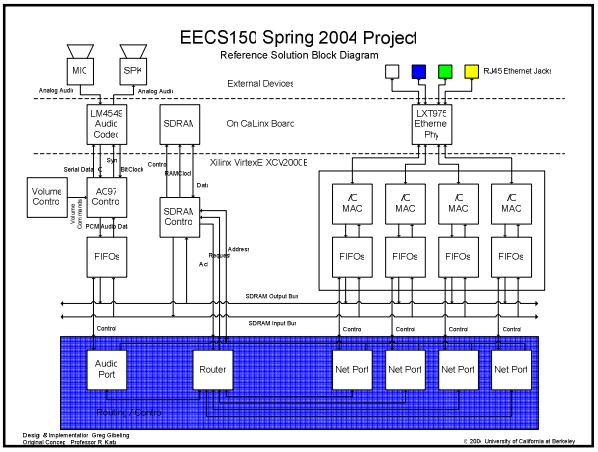
Greg Gibeling

Today

- The Project
- Checkpoint4
- Announcements
- SDRAM FIFOs
- SDRAM Arbiter
- Extra Credit

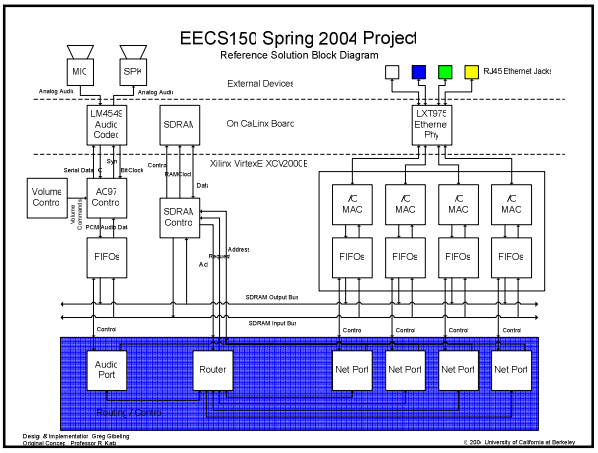
The Project (1)

- Multimedia Network Switch
  - Receive and Transmit PCM Audio
  - Perform basic routing/switching
- Details
  - 4 Ethernet Ports
  - 1 AC97 Audio Port
  - Static Routing Table
  - SDRAM Based FIFO Buffers



The Project (2)

- I/O MAC
  - EthTx - Transmit Ethernet Packets
  - EthRx – Receive Ethernet Packets
- FIFOs
  - 32b Async FIFOs (31 words deep)
- "Ports"
  - Maintain SDRAM Based FIFO
  - More later…

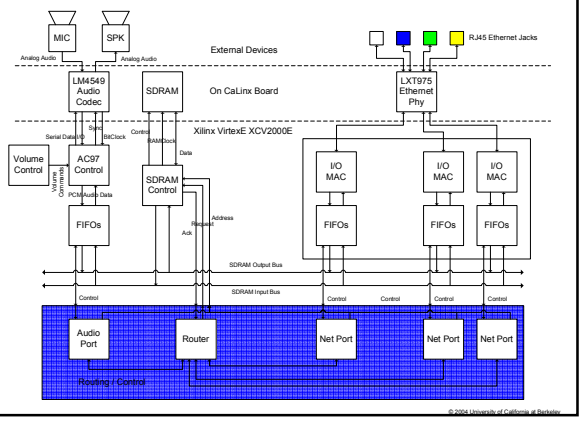


The Project (3)

- SDRAM Control
  - SDRAM_Control/SDRAM_Top
  - SDRAM Access Arbiter
- Router
  - A Layer of Address Indirection
  - Selects the correct SDRAM write address

Checkpoint4

- Minimal Network Switch
  - Checkpoint3
  - Ethernet Passthrough
- Receive Audio from the White Network
  - But move the data through SDRAM
- Connect Green -> Yellow Network
  - Store and Forward through SDRAM



Announcements (1)

- Only 2 Homeworks Left
- Midterm Regrades
  - Please look carefully
  - They're due a week from yesterday
- Problem 3
  - You solution must stop after 4 cycles
  - We will not regrade this

Announcements (2)

- Early Checkoff on April 26th
  - Two weeks from Monday
- Regular Checkoff on May 3rd
  - Three weeks from Monday
- Final Report due 4pm May 7th
- Checkpoint2 Solution is posted

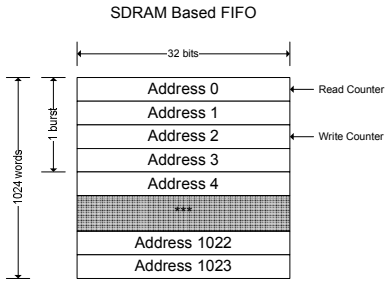
Testbenches

- Absolutely vital to the project
  - Bottom up testing will save you **DAYS**
  - A module isn't finished until it's tested
  - Synthesis takes longer
  - Harder to see most bugs
- Project Solution
  - 6000 lines of Verilog
  - 4000 lines of Testbenches
  - **40% of our verilog is for testing only**

SDRAM FIFO (1)

- 2 Counters
  - One counter for the read address
  - One counter for the write address
- Full/Empty
  - What if the counters are equal?
  - What operation did you do last?
  - Read -> Empty, Write -> Full

SDRAM FIFO (2)



SDRAM FIFO (3)

- FIFO Was Empty
- We have written 4 words
  - To address 0, 1, 2 and 3
- We have done a single write burst

SDRAM FIFO (4)

- Generating SDRAM Addresses
  - {FIFO_ID, Read/Write Address, 2'b00}
  - FIFO_ID is just some fixed number
  - This ensures that the SDRAM FIFOs don't overlap
- When do you count?
  - At the end of a burst

SDRAM Arbiter (1)

- The new SDRAM_Top
- Takes in Multiple Read/Write Requests
- Generates Control Signals
  - When is data needed/ready
  - Which port is actually writing/reading
- Works with the router
  - What address should you be writing to?

SDRAM Arbiter (2)

- Round Robin Scheduling
  - One state per port
  - If current port isn't ready move to next
  - If current port is ready, do a read/write
  - This ensures FIFOs don't fill up
- When do you move to the next port?

After 4 Bursts

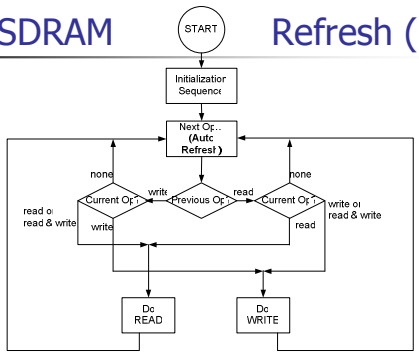
SDRAM Arbiter (3)

- Write addressing…
  - Port A is writing
  - Where is it writing to? Port B
  - So we get the Write Counter from Port B
- This is where the router comes into play
  - Give it a port/Ethernet address to write to
  - It returns the SDRAM address to use

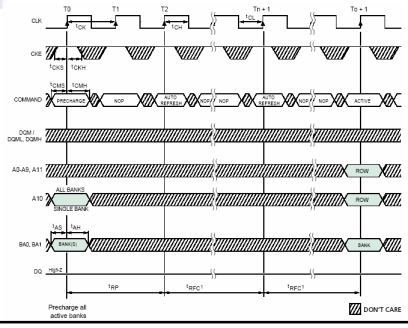
SDRAM Refresh (1)

- You must have SDRAM Refresh
  - It would have made checkpoint1 pointless
  - But without it your project will not work
- A New SDRAM_Control Operation
  - Read Burst
  - Write Burst
  - Auto Refresh

SDRAM Refresh (2)



SDRAM Refresh (3)



Extra Credit Options (1)

- Remote Control & Balance (8%)
  - Volume/etc control by remote
  - Balance Controls
- Early Checkoff (10%)
- Mix with Broadcast Audio (10%)
  - Must hear P2P and Broadcast
- Dynamic Routing (15%)
  - The Rolls Royce Option

Extra Credit Options (2)

- Maximum of Total 20% Extra Credit
- Extra Credit is on top of:
  - Checkpoints
  - Final Checkoff
- You still have to do the project report!