Overview

- Memory Interfacing
  - Memory Type
  - Memory Decoding
  - D-RAM Access
  - Making DRAM Access fast

Review: Buses in a PC: Connect a few devices

- Data rates
  - Memory: 133 MHz, 8 bytes ⇒ 1064 MB/s (peak)
  - PCI: 33 MHz, 8 bytes wide ⇒ 264 MB/s (peak)
  - SCSI: “Ultra3” (80 MHz), “Wide” (2 bytes) ⇒ 160 MB/s (peak)
  - Ethernet: 12.5 MB/s (peak)

Review: Computers with Memory Mapped I/O

- I/O devices Accessed like memory
Big Picture: A System on a Chip

Integration of Core Processor and many subsystem micro-cells
- ARM7TDMI core
- Cache RAM
- Embedded Co-processors
- External Mem Interface
- Low bandwidth I/O devices
- Timers
- I/O ports

ARM System Architecture

Need a Mechanism to access various memory units and I/O devices, uniquely, to avoid access conflicts

ARM System Architecture with Multiple Masters

Need a Mechanism to allow various Processing units to access the Memory Bus without causing conflict

ARM Core Interface Signals

Memory Interface
ARM Core Memory Interface Signals

- Internal clock is mclk AND wait
- 32 bit address A[31:0]
- 32 Bi-directional Data D[31:0]
- Separate Data in and out Din[31:0] & Dout[31:0]
- Bidirectional Data bus D[31:0]
- nmreq and seq for requesting memory access
- nr/w for read/write indication
- mas[1:0] for data size identification: word 10, half-word 01 and byte 00.
- All activities controlled by mclk.

Simple Memory Interface

- 4 SRAMs
  - write enabled separately
  - Read enabled together
- 4 ROMs
  - No write enable
  - Read enabled together
- SRAM Size: $2^n \times 32$
- ROM Size: $2^m \times 32$

Simple Memory Decoder Control

- Controls the Activation of RAM and ROM
  - a[31]: 0 → ROM
  - a[31]: 1 → RAM
- It controls the byte write enables during write
  - mas[1:0]: 00: Byte, 01: H-word, 10: Word
- It ensures that data is ready before processor continues.

SRAM/ROM Memory Timing

- Address should be stable during the falling edge
- SRAM is fast, ROM is slow
  - ROM needs more time. Slows the system
- Solutions?
  - Slow down the MCLK clock; loose performance
  - Use Wait states; more complex control
**ROM Wait Control State Transition**

- ROM access requires 4 clock cycles
- RAM access is fast

**Timing Diagram for ROM Wait States**

**Improving Performance**

- Processor internal operations cycles do not need access to memory
  - Mem. Access is much slower than internal operations.
  - Use wait states for mem Accesses
- $mreq = 1$ internal operation
- $mreq = 0$ memory access

**DRAM Interface**

- Dynamic RAM Features:
  - much cheaper than SRAM
  - more capacity than SRAM
  - slower than SRAM
- Widely used in Computer Systems
**DRAM Organisation**

- Two dimensional matrix
- Bits are accesses by:
  - Accepting row and column addresses down the same multiplexed address bus
  - First Row address is presented and latched by ras signal
  - Next column address is presented and latched by cas signal

**Making DRAM Access Fast**

- Accessing data in the same row using cas-only access is 2 – 3 times faster
  - cas-only access does not activate the cell matrix
  - If next accesses is within the same row, a new column address may be presented just by applying a cas-only access.
- Fact: Most processor addresses are sequential (75%)
- If we had a way of knowing that the next address is sequential with respect with the current address (current address + 4), then we could only assert cas and make DRAM access fast
- Difficulty?
  - Detecting early in memory access cycle that the next address is in the same row.

**ARM Solution to cas-only Access**

- ARM address register Instruction:
  - 75% of next addresses are current address +4.
  - Sequential addresses flagged by seq signal
  - The external mem device checks previous address and row boundaries to issue cas only or ras-cas

**Revised State Transition Diagram**

- seq = 1: sequential address
- seq = 0: non-sequential
- mreq = 1 internal operation
- mreg = 0 memory access
DRAM Timing Diagram

- Notice the pipelined memory access
  - Address is presented 1/2 cycle earlier

DRAM Timing Diagram after an Internal Cycle

- During internal operations cycles, a memory access cycle can be set up in advance.
  - This eliminates the wait (New Cycle) state

Memory Access Timing Summary

- Notice the pipelined memory access
  - Address is presented 1/2 cycle earlier

Reading Material

Conclusion

° Memory interfacing can degrade performance

° Can improve performance by increasing the clock frequency and allocating differing clock cycles for each memory access type

° cas-only accesses in DRAM are 2 to 3 times faster than ras – cas accesses.