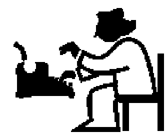
Administrative Stuff

- Homework 2 due NOW
  - No late homework accepted
- Homework 3 available on the www
  - Due Thursday 10/17
- Quiz 1 Tuesday 10/15
  - Cover
    - ESD Chapter 1-4
    - homework 1 and 2
    - Lecture 9/26-10/8
    - Lab tutorial 1,2 Lab 1-3
  - Extra credit questions
    - Take from student presentations



Memory

Outline

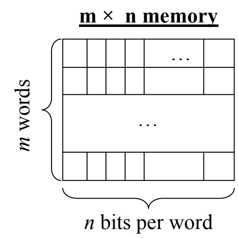
- Memory Write Ability and Storage Permanence
- Common Memory Types
- Composing Memory
- Memory Hierarchy and Cache
- Advanced RAM

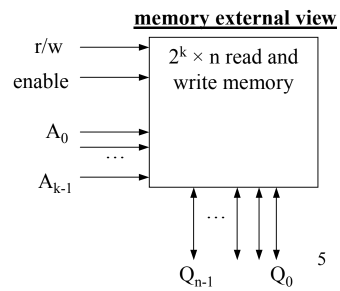
Embedded System Functionality

- Three aspects
  - Processing
    - Processors
    - Transformation of data
  - Storage
    - Memory
    - Retention of data
  - Communication
    - Buses
    - Transfer of data

Basic concepts

- Stores large number of bits
  - m x n: m words of n bits each
  - k = Log2(m) address input signals
  - m = 2^k words
  - e.g., 4,096 x 8 memory:
    - 32,768 bits
    - 12 address input signals
    - 8 input/output data signals
- Memory access
  - r/w: selects read or write
  - enable: read or write only when asserted
  - multiport: multiple accesses to different locations simultaneously





Trends

- Memory size double every 18 months
  - Following Moore's law
- Memory advances drive new products
  - Digital camera
    - Enable by fast CPU, A2D, and cheaper flash
- Tradition distinction for different types of memory
  - ROM: Read Only Memory
  - RAM: Random Access Memory

Write ability/ storage permanence

- Traditional ROM/RAM distinctions
  - ROM
    - read only, bits stored without power
  - RAM
    - read and write, lose stored bits without power
- Traditional distinctions blurred
  - Advanced ROMs can be written to
    - e.g., EEPROM
  - Advanced RAMs can hold bits without power
    - e.g., NVRAM
- Write ability
  - Manner and speed a memory can be written
- Storage permanence
  - Ability of memory to hold stored bits after they are written

7

CS122A: Embedded System Design, Fall 02

Write ability

- Ranges of write ability
  - High end
    - Processor writes to memory simply and quickly
    - E.g., RAM
  - Middle range

    Processor writes to memory, but slower
    - E.g., FLASH, EEPROM
  - Lower range
    - Special equipment, "programmer", must be used to write to memory
    - E.g., EPROM, OTP ROM
  - Low end
    - Bits stored only during fabrication
    - E.g., Mask-programmed ROM
- In-system programmable memory
  - Can be written to by processor in embedded system
  - Memories in high end and middle range of write ability

8

CS122A: Embedded System Design, Fall 02

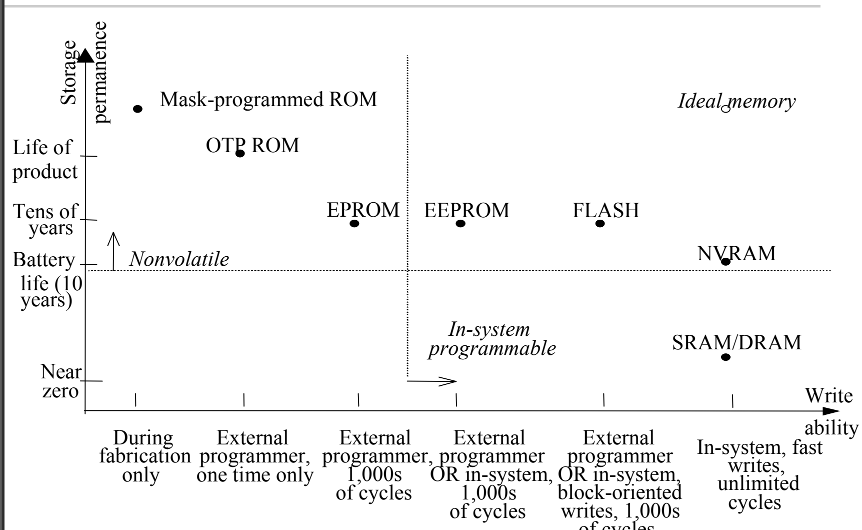
Storage permanence

- Range of storage permanence
  - High end
    - Essentially never loses bits
    - E.g., mask-programmed ROM
  - Middle range
    - Holds bits days, months, or years after memory's power source turned off
    - E.g., NVRAM
  - Lower range
    - Holds bits as long as power supplied to memory
    - E.g., SRAM
  - Low end
    - Begins to lose bits almost immediately after written
    - E.g., DRAM
- Nonvolatile memory
  - Holds bits after power is no longer supplied
  - High end and middle range of storage permanence

9

CS122A: Embedded System Design, Fall 02

Write ability/ storage permanence

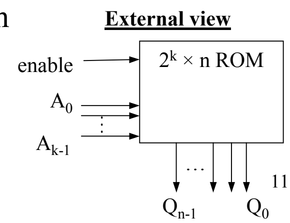


10

CS122A: Embedded System Design, Fall 02

ROM: “Read-Only” Memory

- Nonvolatile memory
- Can be read from but not written to
  - By a processor in an embedded system
- Traditionally written to through “programming”
  - Before inserting to embedded system
- Uses
  - Store software program for general-purpose processor
    - program instructions can be one or more ROM words
  - Store constant data needed by system
  - Implement combinational circuit

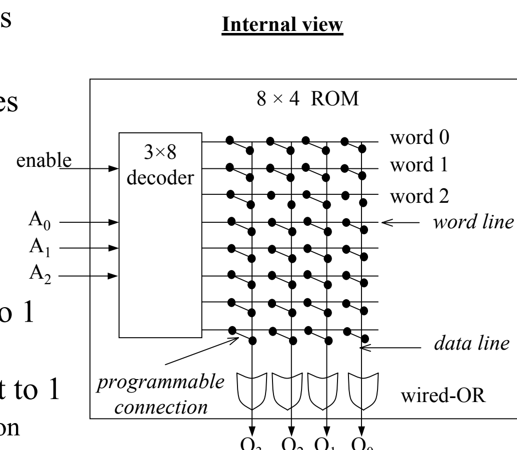


11

CS122A: Embedded System Design, Fall 02

8 x 4 ROM

- Horizontal lines = words
- Vertical lines = data
- Connected only at circles



- Decoder sets word 2’s to 1
  - if address input is 010
- Data lines Q_3, Q_1 are set to 1
  - “programmed” connection
- Word 2 not connected w/ Q_2 Q_0
- Output is 1010

12

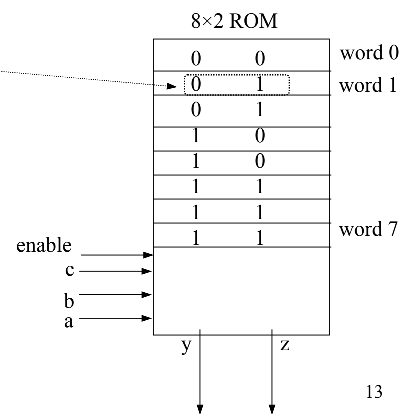
CS122A: Embedded System Design, Fall 02

Implementing combinational function

- Any combinational circuit of n functions of same k variables can be done with 2^k x n ROM

Truth table

Inputs (address)			Outputs	
a	b	c	y	z
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1



Types of ROM

- In order of increasing write ability
  - Mask-programmed ROM
  - One-time programmable ROM
  - Erasable programmable ROM
  - Electrically erasable programmable ROM
  - Flash

Mask-programmed ROM

- Connections "programmed" at fabrication
  - Set of masks
- Lowest write ability
  - Only once
- Highest storage permanence
  - Bits never change unless damaged
- Typically used for final design of high-volume systems

15

CS122A: Embedded System Design, Fall 02

One-time programmable ROM

- Connections "programmed" after manufacture by user
  - User provides file of desired contents of ROM
  - File input to "ROM programmer"
  - Each programmable connection is a fuse
  - ROM programmer blows fuses where connections should not exist
- Very low write ability
  - Typically written only once and requires ROM programmer device
- Very high storage permanence
  - bits don't change unless reconnected to programmer & more fuses blown
- Commonly used in prototype or even final products
  - Cheaper then other PROM
  - harder to inadvertently modify through radiation or time
  - Cheaper for low volume and faster time-to-market than Mask PROM

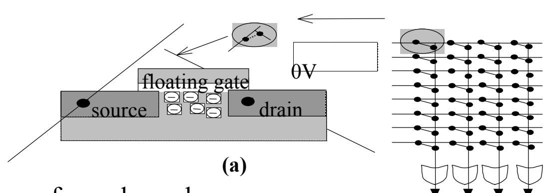
16

CS122A: Embedded System Design, Fall 02

EPROM: Erasable programmable ROM

- Programmable component is a MOS transistor
  - Transistor has “floating” gate surrounded by an insulator
- Better write ability
  - Erased/reprogrammed thousands of times
- Reading is much faster than writing
- Reduced storage permanence
  - Program lasts about 10 years
  - But is susceptible to radiation and electric noise
- Typically used during design development

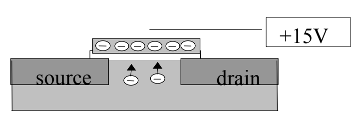
EPROM: Erasable programmable ROM

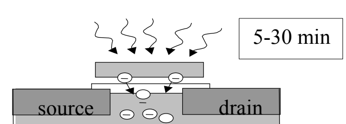


(a)

- (a) Negative charges form channel between source and drain
  - Storing logic 1
- (b) Negative charges to move out of channel and get trapped in floating gate
  - Storing logic 0
- (c) UV rays causes negative charges to return to channel from floating gate
  - Storing the logic 1
- (d) Quartz window through which UV light can pass

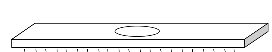
(b)





(c)

(d)



EEPROM: Electrically erasable programmable ROM

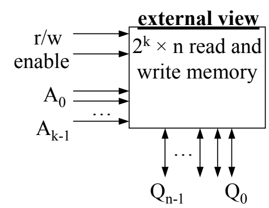
- Programmed and erased electronically
  - Typically by using higher than normal voltage
  - Can program and erase individual words
- Better write ability
  - In-system programmable
    - With built-in circuit to provide higher than normal voltage
    - Built-in memory controller used to hide details from memory user
  - Writes very slow due to erasing and programming
    - 10's microseconds versus 10's nanoseconds
    - "Busy" pin indicates to processor EEPROM still writing
  - Can be erased and programmed tens of thousands of times
- Similar storage permanence to EPROM (about 10 years)
- Far more convenient than EPROMs, but more expensive

Flash Memory

- Extension of EEPROM
  - Same floating gate principle
  - Same write ability and storage permanence
- Fast erase
  - Large blocks of memory erased at once, rather than one word at a time
  - Blocks typically several thousand bytes large
- Writes to single words may be slower
  - Entire block must be read, word updated, then entire block written back
- Used in systems storing large data items in NV memory
  - E.g., digital cameras, TV set-top boxes, cell phones

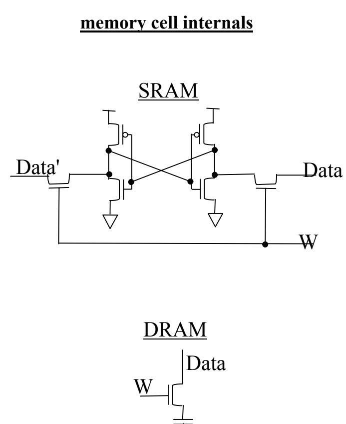
RAM: "Random-access" memory

- Typically volatile memory
  - bits are not held without power supply
- Read and written to easily by embedded system during execution
- Random-access, as opposed to:
  - Sequential access, like tape and disk



Basic types of RAM

- SRAM: Static RAM
  - Memory cell uses flip-flop to store bit
  - Requires 6 transistors
  - Holds data as long as power supplied
- DRAM: Dynamic RAM
  - Memory cell uses MOS transistor and capacitor to store bit
  - More compact than SRAM
  - "Refresh" required due to capacitor leakage
    - Cells refreshed when read
  - Typical refresh rate 15.625 microsec.
  - Slower to access than SRAM
  - Usually implemented on another IC



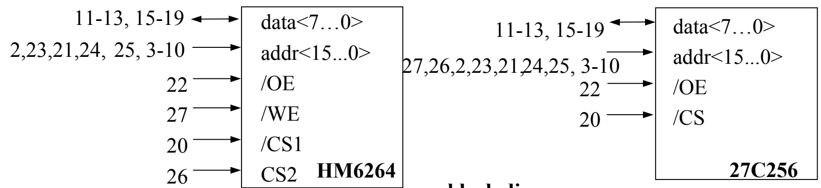
Ram variations

- PSRAM: Pseudo-static RAM
  - DRAM with built-in memory refresh controller
  - Popular low-cost high-density alternative to SRAM
- NVRAM: Nonvolatile RAM
  - Holds data after external power removed
  - Battery-backed RAM
    - SRAM with own permanently connected battery (lasting 10 years)
    - writes as fast as reads
    - no limit on number of writes unlike nonvolatile ROM-based memory
      - My $100 memory stick has a write cycle limit of 500,000!!! ☹
  - SRAM with EEPROM or flash
    - stores complete RAM contents on EEPROM or flash before power turned off

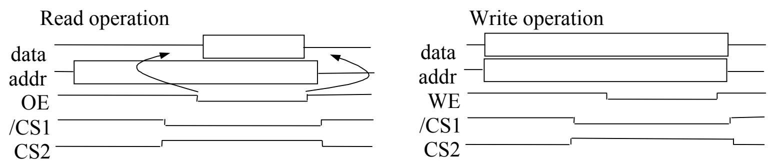
Example: HM6264 & 27C256 RAM/ROM devices

- Low-cost low-capacity memory devices
- Commonly used in 8-bit microcontroller-based embedded systems
- First two numeric digits indicate device type
  - RAM: 62
  - ROM: 27
- Subsequent digits indicate capacity in kilobits
- 27C256 is actually an EPROM

Example: HM6264 & 27C256 RAM/ROM devices



block diagrams



timing diagrams

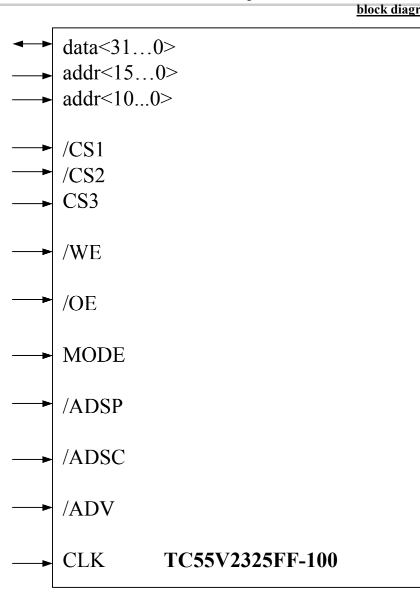
25

CS122A: Embedded System Design, Fall 02

Example: TC55V2325FF-100 memory device

- 2-megabit synchronous pipelined burst SRAM memory device
- Designed to be interfaced with 32-bit processors
- Fast single byte read/write
- Burst mode read
  - ADSP, ADSC, and ADV all active
  - Output sequential memory location at every clock
- Burst mode write

block diagram

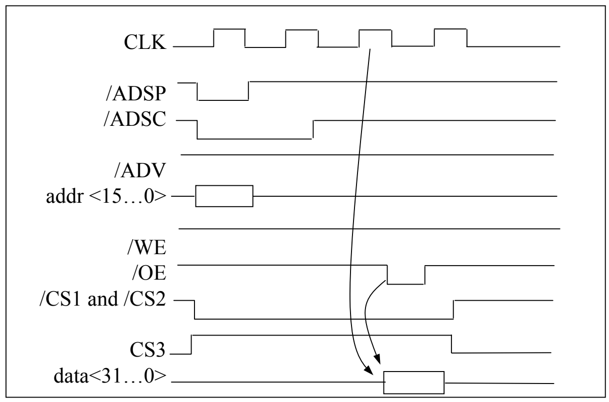


26

CS122A: Embedded System Design, Fall 02

Example: TC55V2325FF-100 memory device

A single read operation



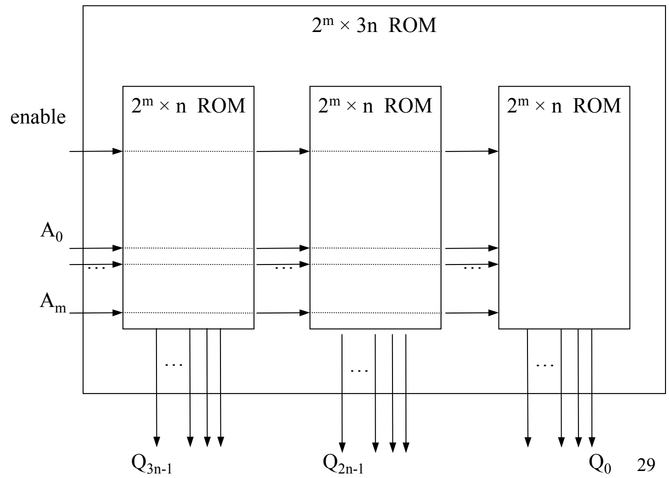
timing diagram

Composing memory

- Memory size needed often differs from size of readily available memories
- When available memory is larger, simply ignore unneeded high-order address bits and higher data lines
- When available memory is smaller, compose several smaller memories into one larger memory

Composing memory

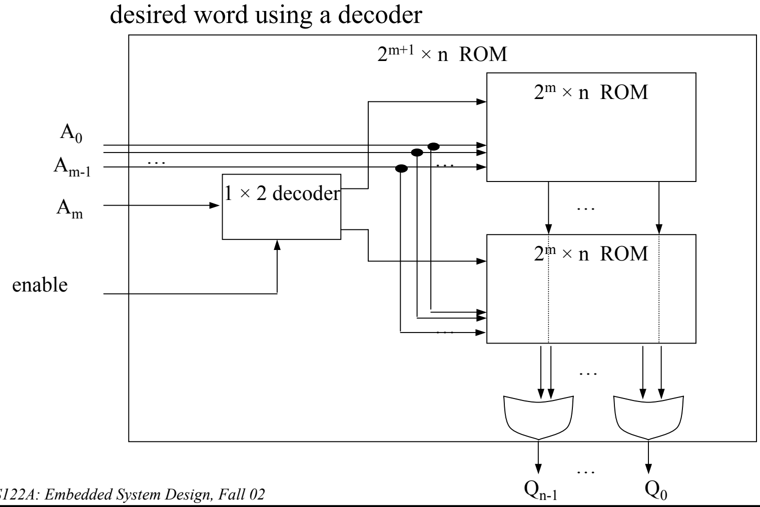
– Connect side-by-side to increase width of words



29

Composing memory

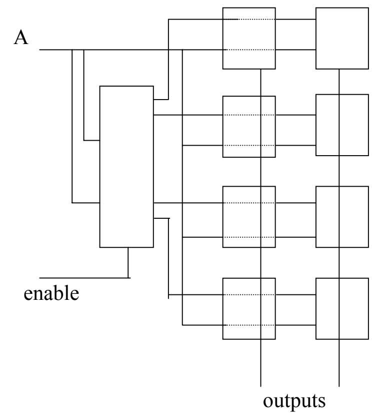
– Connect top to bottom to increase number of words
  - added high-order address line selects smaller memory containing desired word using a decoder



30

Composing memory

– Combine techniques to increase number and width of words

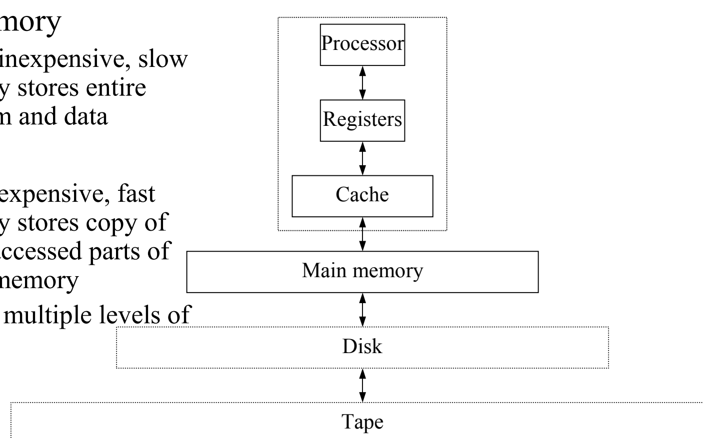


31

CS122A: Embedded System Design, Fall 02

Memory hierarchy

- Want inexpensive, fast memory
- Main memory
  - Large, inexpensive, slow memory stores entire program and data
- Cache
  - Small, expensive, fast memory stores copy of likely accessed parts of larger memory
  - Can be multiple levels of cache



32

CS122A: Embedded System Design, Fall 02

Cache

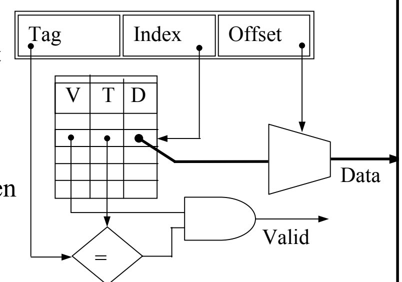
- Usually designed with SRAM
  - faster but more expensive than DRAM
- Usually on same chip as processor
  - space limited, so much smaller than off-chip main memory
  - faster access (1 cycle vs. several cycles for main memory)
- Cache operation:
  - Request for main memory access (read or write)
  - First, check cache for copy
    - cache hit: copy is in cache, quick access
    - cache miss: copy not in cache, read from main memory into cache
- Several cache design choices
  - cache mapping, replacement policies, and write techniques

Cache mapping

- Used to assign main memory locations to cache locations and determine hit or miss
  - Much less cache locations than main memory locations
- Three basic techniques:
  - Direct mapping
  - Fully associative mapping
  - Set-associative mapping
- Caches partitioned into indivisible blocks or lines of adjacent memory addresses
  - usually 4 or 8 addresses per line

Direct mapping

- Main memory address divided into 2 fields
  - Index
    - cache address
    - number of bits determined by cache size
  - Tag
    - compared with tag stored in cache at address indicated by index
    - if tags match, check valid bit
- Valid bit
  - indicates whether data in slot has been loaded from memory
- Offset
  - used to find particular word in cache line

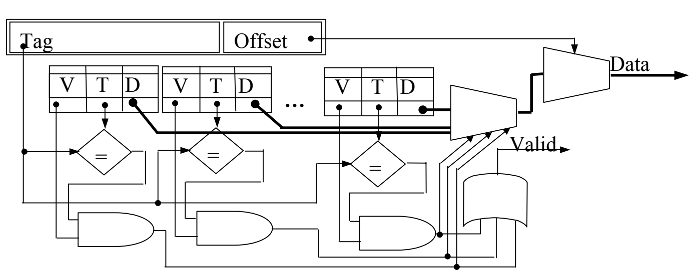


35

CS122A: Embedded System Design, Fall 02

Fully associative mapping

- All addresses stored in cache simultaneously compared with desired address

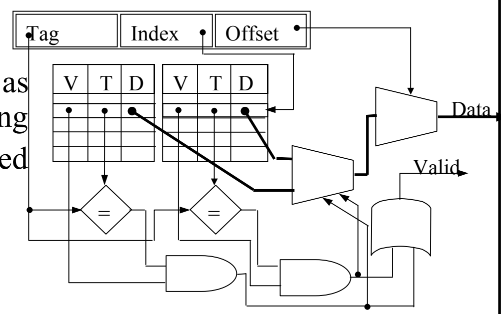


36

CS122A: Embedded System Design, Fall 02

Set-associative mapping

- Compromise btw direct & fully associative mapping
- Index as in direct mapping
- Each cache address contains content and tags of 2 or more memory address locations
- Tags of that set simultaneously compared as in fully associative mapping
- Cache with set size N called N-way set-associative
  - 2-way, 4-way, 8-way are common



Cache-replacement policy

- Technique for choosing which block to replace
  - when fully associative cache is full
  - when set-associative cache's set is full
- Direct mapped cache has no choice
- Random
  - replace block chosen at random
- LRU: least-recently used
  - replace block not accessed for longest time
- FIFO: first-in-first-out
  - push block onto queue when accessed
  - choose block to replace by popping queue

Cache write techniques

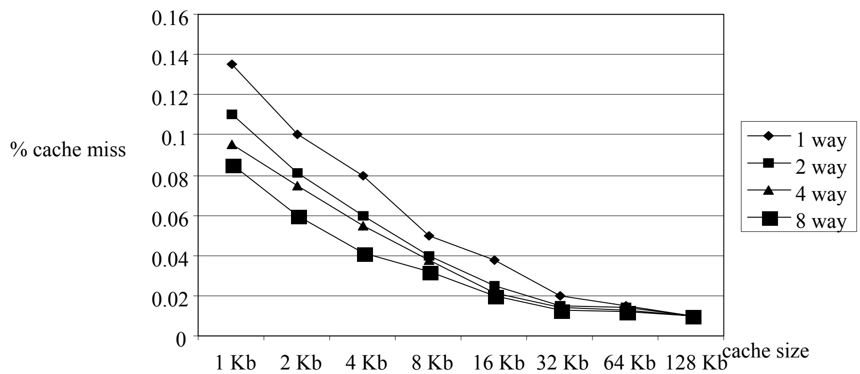
- When written, data cache must update main memory
- Write-through
  - write to main memory whenever cache is written to
  - easiest to implement
  - processor must wait for slower main memory write
  - potential for unnecessary writes
- Write-back
  - main memory only written when "dirty" block replaced
  - extra dirty bit for each block set when cache block written to
  - reduces number of slow main memory writes

Cache impact on system performance

- Most important parameters in terms of performance:
  - Total size of cache
    - total number of data bytes cache can hold
    - tag, valid and other house keeping bits not included in total
  - Degree of associativity
  - Data block size
- Larger caches achieve lower miss rates but higher access cost
  - 2 Kbyte cache: miss rate = 15%, hit = 2 cycles, miss = 20 cycles
    - avg. cost of memory access = (0.85 * 2) + (0.15 * 20) = 4.7 cycles
  - 4 Kbyte cache: miss rate = 6.5%, hit = 3 cycles, miss = 20 cycles
    - avg. cost of memory access = (0.935 * 3) + (0.065 * 20) = 4.105 cycles (improve)
  - 8 Kbyte cache: miss rate = 5.565%, hit = 4 cycles, miss = 20 cycles
    - avg. cost of mem access = (0.94435 * 4) + (0.05565 * 20) = 4.8904 cycles (worsen)

Cache performance trade-offs

- Improving cache hit rate without increasing size
  - Increase/Decrease set-associativity



41

CS122A: Embedded System Design, Fall 02

Advanced RAM

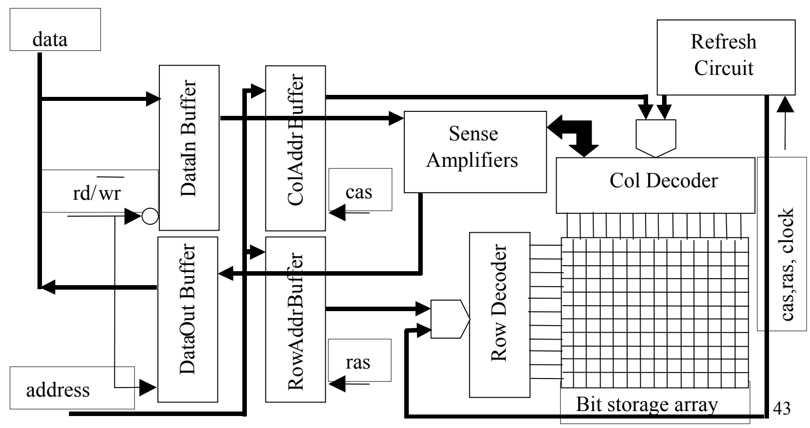
- DRAMs commonly used as main memory in processor based embedded systems
  - high capacity, low cost
- Many variations of DRAMs proposed
  - need to keep pace with processor speeds
  - FPM DRAM: fast page mode DRAM
  - EDO DRAM: extended data out DRAM
  - SDRAM/ESDRAM: synchronous and enhanced synchronous DRAM
  - RDRAM: Rambus DRAM

42

CS122A: Embedded System Design, Fall 02

Basic DRAM

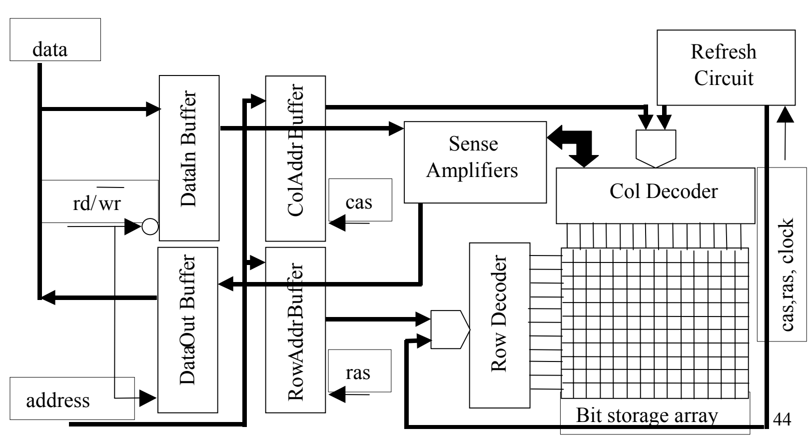
- Address bus multiplexed between row and column components



CS122A: Embedded System Design, Fall 02 43

Basic DRAM

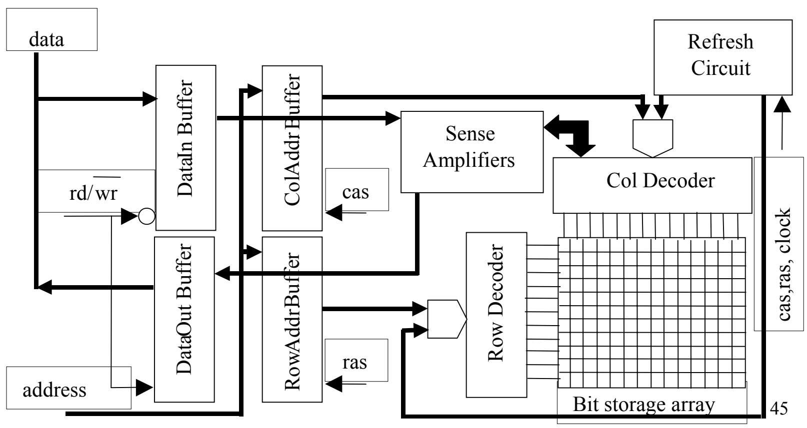
- Row and column addresses are latched in, sequentially, by strobing *ras* and *cas* signals, respectively



CS122A: Embedded System Design, Fall 02 44

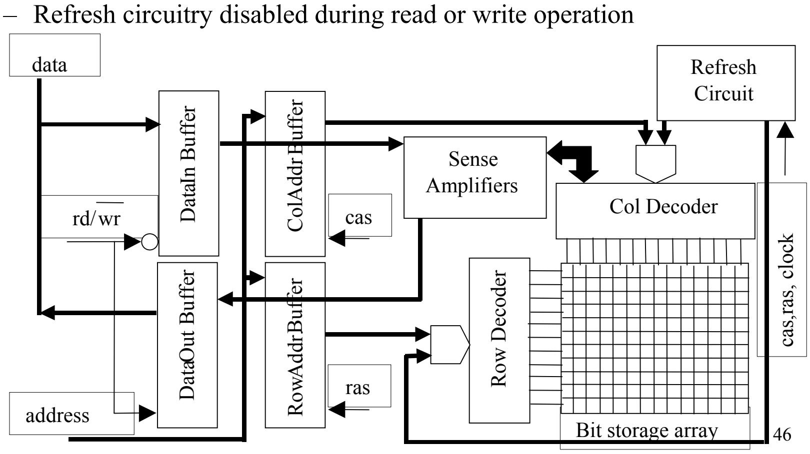
Basic DRAM

- Sense Amps “detect” the logic levels and output them



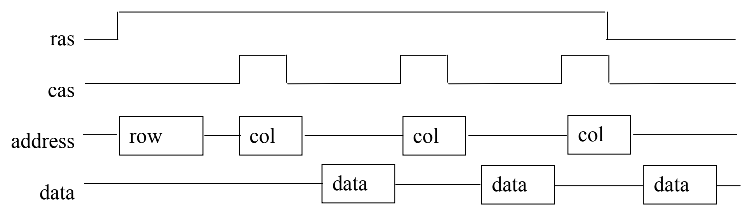
Basic DRAM

- Refresh circuitry can be external or internal to DRAM device
  - strobes consecutive memory address periodically causing memory content to be refreshed
  - Refresh circuitry disabled during read or write operation



Fast Page Mode DRAM (FPM DRAM)

- Each row of memory bit array is viewed as a page
- Page contains multiple words
- Individual words addressed by column address
- Timing diagram:
  - row (page) address sent
  - 3 words read consecutively by sending column address for each
- Extra cycle eliminated on each read/write of words from same page

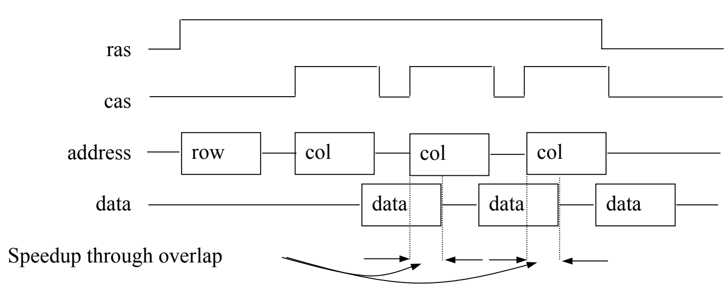


47

CS122A: Embedded System Design, Fall 02

Extended data out DRAM (EDO DRAM)

- Improvement of FPM DRAM
- Extra latch to buffer output
  - allows strobing of cas before data read operation completed
- Reduces read/write latency

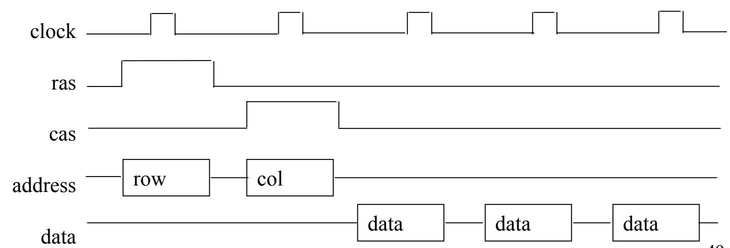


48

CS122A: Embedded System Design, Fall 02

Synchronous DRAM

- SDRAM latches data on active edge of clock
- Eliminates time to detect ras/cas and rd/wr signals
- A counter is initialized to column address then incremented on active edge of clock to access consecutive memory locations



Rambus in 1988

→Is said to, "…adding complexity to increase bandwidth"

→ Would cost more, at least in the short run, and would also take longer to retrieve the first word of data in order to decrease the time to get the rest of the data.

→Why was it controversial at the time?

→ Adopting this technology would entail many changes, for instance, chip testing, packaging, and motherboard design.

→Breaks: 1995 Silicon Graphics use RDRAM for high-end workstations and later on in its graphics system for the N64, and 1996 when Intel developed Rambus-based chipsets for Pentium 3 processors.

Rambus Inc.

- → They do not make their own chips, instead they license their designs to manufacturers and collect royalties on products that emerge.
- → Rambus Inc. currently holds 104 U.S. patents.
- → Other company's started off this way.
  - →Adobe Systems Inc. with its Postscript code
  - → Dolby Laboratories Inc.
- → Having no capital expenditures, they became profitable only after 5 years.
  - →Stock was up to $127 a share
  - →But now only $4 …WOW! What a drop!

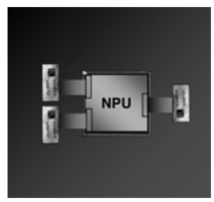
- →Rambus technology was shipping in over 250 products.
  - → In workstations, PC's, game consoles, High-Def TV

Most notable product…

How Do Rambus DRAM's work?

→Specifications:

→ 16 bit wide, 800 MHz channel with special memory chips (RDRAM's) mounted on Rambus in-line memory modules, a.k.a. RIMM's, and an interface that allows the memory controller to talk to the RDRAM's.



RDRAM

53

DRAM integration problem

- SRAM easily integrated on same chip as processor
- DRAM more difficult
  - Different chip making process between DRAM and conventional logic
  - Goal of conventional logic (IC) designers:
    - minimize parasitic capacitance to reduce signal propagation delays and power consumption
  - Goal of DRAM designers:
    - create capacitor cells to retain stored information
  - Integration processes beginning to appear

54

Memory Management Unit (MMU)

- Duties of MMU
  - Handles DRAM refresh, bus interface and arbitration
  - Takes care of memory sharing among multiple processors
  - Translates logic memory addresses from processor to physical memory addresses of DRAM
- Modern CPUs often come with MMU built-in
- Single-purpose processors can be used

Administrative Stuff

- Homework 2 due 60 minutes ago
  - No late homework accepted
- Homework 3 available on the www
  - Due Thursday 10/17
- Quiz 1 Tuesday 10/15
  - Cover
    - ESD Chapter 1-4
    - homework 1 and 2
    - Lecture 9/26-10/8
    - Lab tutorial 1,2 Lab 1-3
  - Extra credit questions
    - Take from student presentations