

Embedded Systems Design: A Unified Hardware/Software Introduction

Chapter 5 Memory



Outline

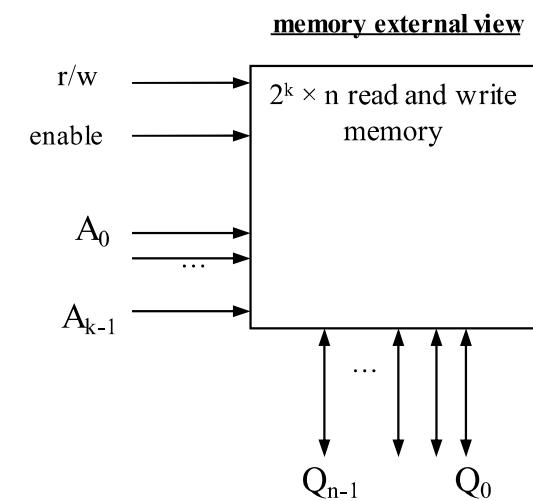
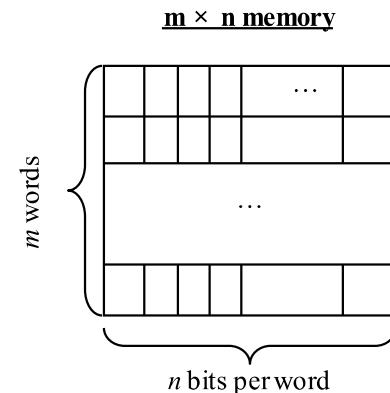
- Memory Write Ability and Storage Permanence
- Common Memory Types
- Composing Memory
- Memory Hierarchy and Cache
- Advanced RAM

Introduction

- Embedded system's functionality aspects
 - Processing
 - processors
 - transformation of data
 - Storage
 - memory
 - retention of data
 - Communication
 - buses
 - transfer of data

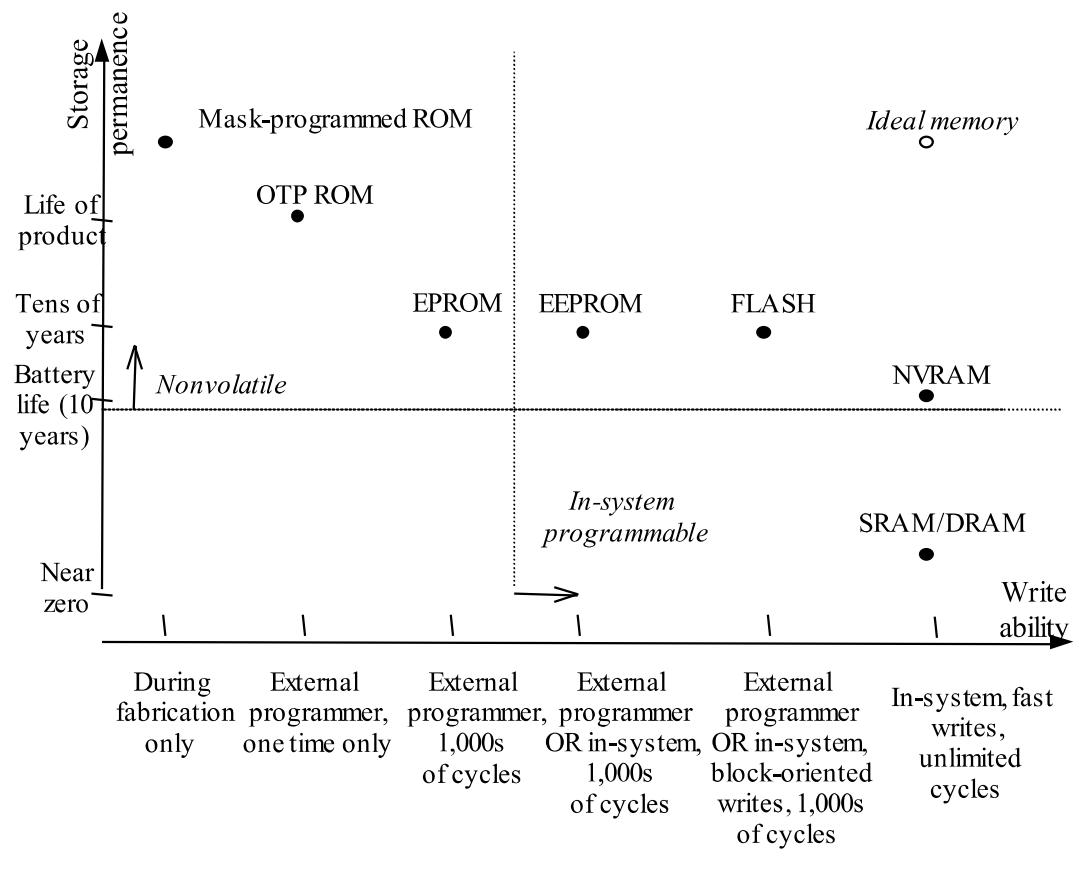
Memory: basic concepts

- Stores large number of bits
 - $m \times n$: m words of n bits each
 - $k = \log_2(m)$ address input signals
 - or $m = 2^k$ 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



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



Write ability and storage permanence of memories,
showing relative degrees along each axis (not to scale).

Write ability

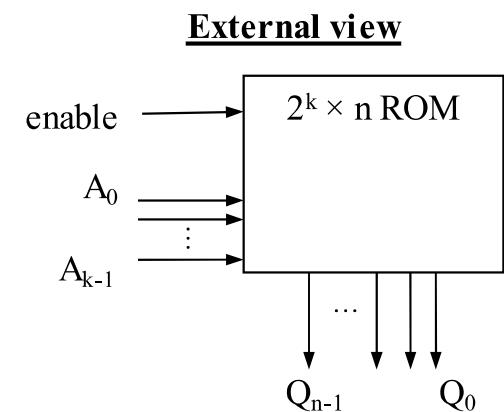
- Ranges of write ability
 - High end
 - processor writes to memory simply and quickly
 - e.g., RAM
 - Middle range
 - processor writes to memory, but slower
 - e.g., FLASH, EEPROM
 - Lower range
 - special equipment, “programmer”, must be used to write to memory
 - e.g., EPROM, OTP ROM
 - Low end
 - bits stored only during fabrication
 - e.g., Mask-programmed ROM
- In-system programmable memory
 - Can be written to by a processor in the embedded system using the memory
 - Memories in high end and middle range of write ability

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

ROM: “Read-Only” Memory

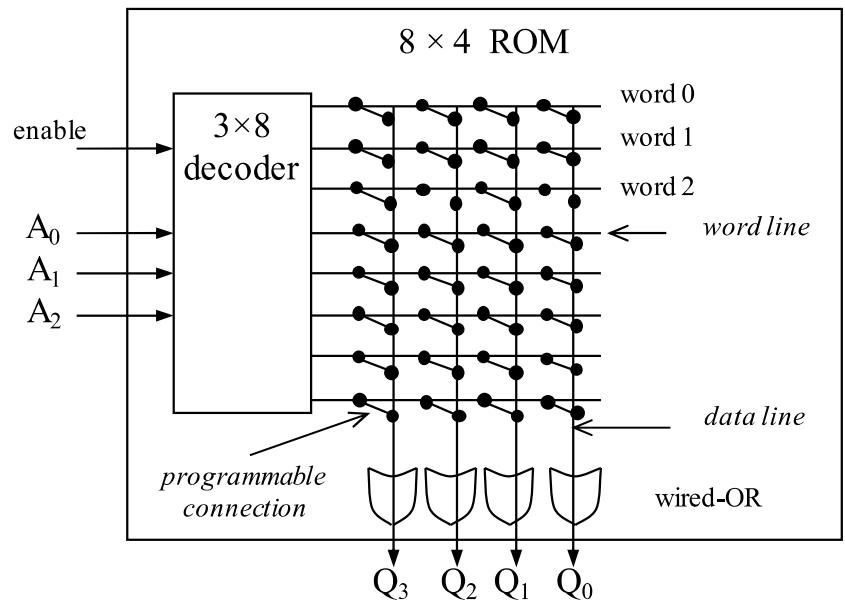
- Nonvolatile memory
- Can be read from but not written to, by a processor in an embedded system
- Traditionally written to, “programmed”, before inserting to embedded system
- Uses
 - Store software program for general-purpose processor
 - program instructions can be one or more ROM words
 - Store constant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



Example: 8 x 4 ROM

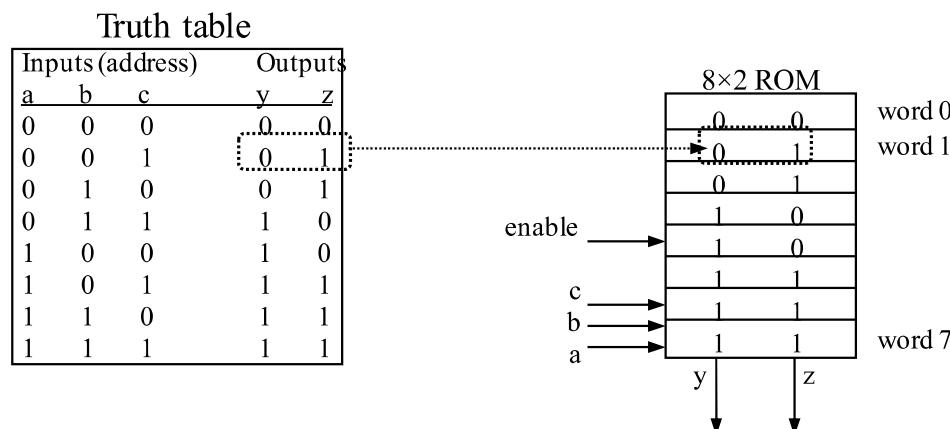
- Horizontal lines = words
- Vertical lines = data
- Lines connected only at circles
- Decoder sets word 2's line to 1 if address input is 010
- Data lines Q_3 and Q_1 are set to 1 because there is a “programmed” connection with word 2's line
- Word 2 is not connected with data lines Q_2 and Q_0
- Output is 1010

Internal view



Implementing combinational function

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



Mask-programmed ROM

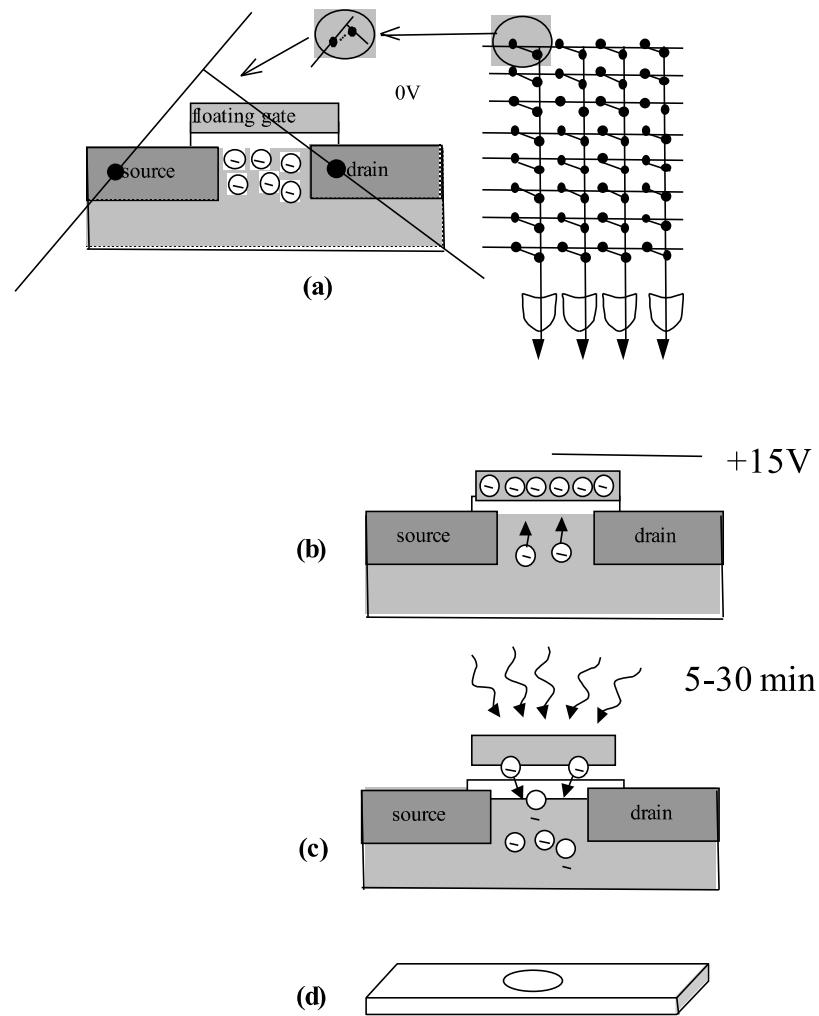
- Connections “programmed” at fabrication
 - set of masks
- Lowest write ability
 - only once
- Highest storage permanence
 - bits never change unless damaged
- Typically used for final design of high-volume systems
 - spread out NRE cost for a low unit cost

OTP ROM: One-time programmable ROM

- Connections “programmed” after manufacture by user
 - user provides file of desired contents of ROM
 - file input to machine called 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 and more fuses blown
- Commonly used in final products
 - cheaper, harder to inadvertently modify

EPROM: Erasable programmable ROM

- **Programmable component is a MOS transistor**
 - Transistor has “floating” gate surrounded by an insulator
 - (a) Negative charges form a channel between source and drain storing a logic 1
 - (b) Large positive voltage at gate causes negative charges to move out of channel and get trapped in floating gate storing a logic 0
 - (c) (Erase) Shining UV rays on surface of floating-gate causes negative charges to return to channel from floating gate restoring the logic 1
 - (d) An EPROM package showing quartz window through which UV light can pass
- **Better write ability**
 - can be erased and reprogrammed thousands of times
- **Reduced storage permanence**
 - program lasts about 10 years but is susceptible to radiation and electric noise
- **Typically used during design development**



EEPROM: Electrically erasable programmable ROM

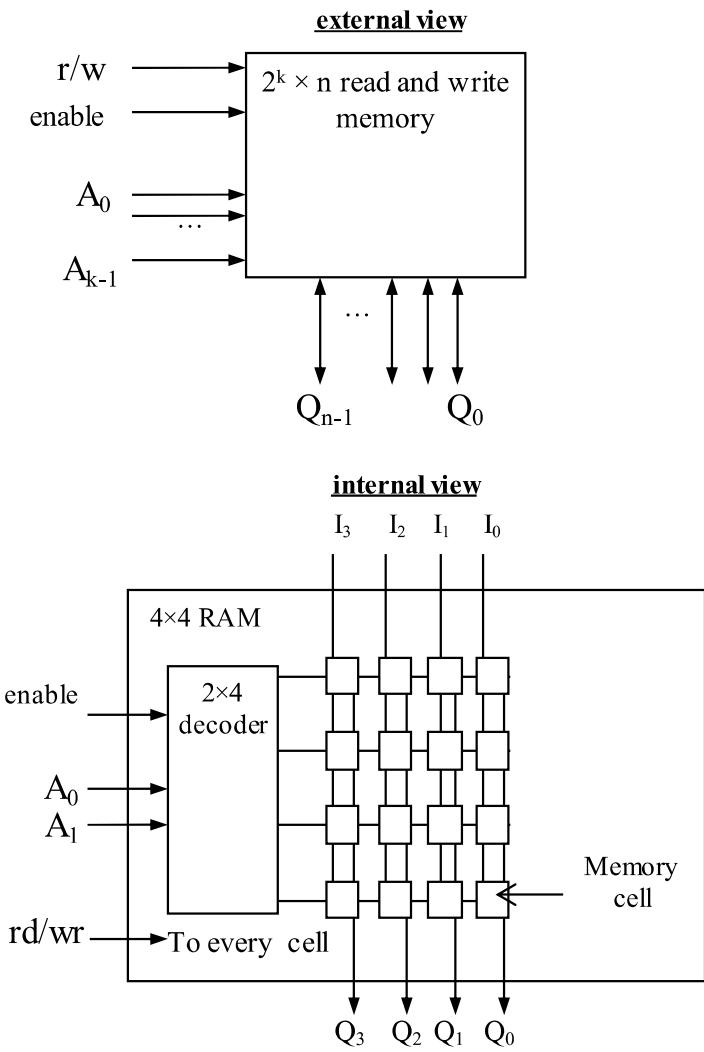
- Programmed and erased electronically
 - typically by using higher than normal voltage
 - can program and erase individual words
 - Better write ability
 - can be in-system programmable with built-in circuit to provide higher than normal voltage
 - built-in memory controller commonly used to hide details from memory user
 - writes very slow due to erasing and programming
 - “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 with embedded systems storing large data items in nonvolatile memory
 - e.g., digital cameras, TV set-top boxes, cell phones

RAM: “Random-access” memory

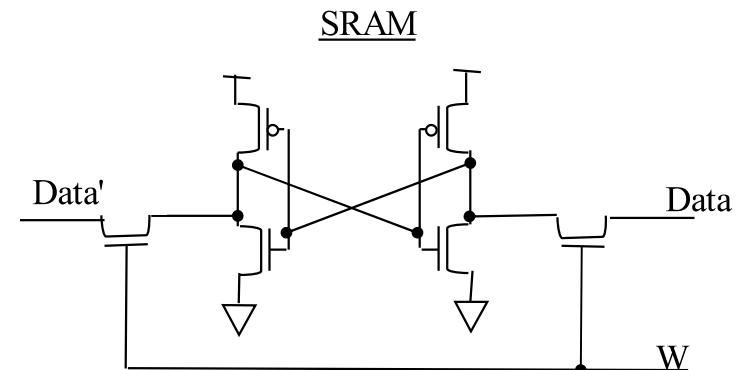
- **Typically volatile memory**
 - bits are not held without power supply
- **Read and written to easily by embedded system during execution**
- **Internal structure more complex than ROM**
 - a word consists of several memory cells, each storing 1 bit
 - each input and output data line connects to each cell in its column
 - rd/wr connected to every cell
 - when row is enabled by decoder, each cell has logic that stores input data bit when rd/wr indicates write or outputs stored bit when rd/wr indicates read



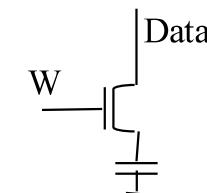
Basic types of RAM

- 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 leak
 - word’s cells refreshed when read
 - Typical refresh rate 15.625 microsec.
 - Slower to access than SRAM

memory cell internals



DRAM



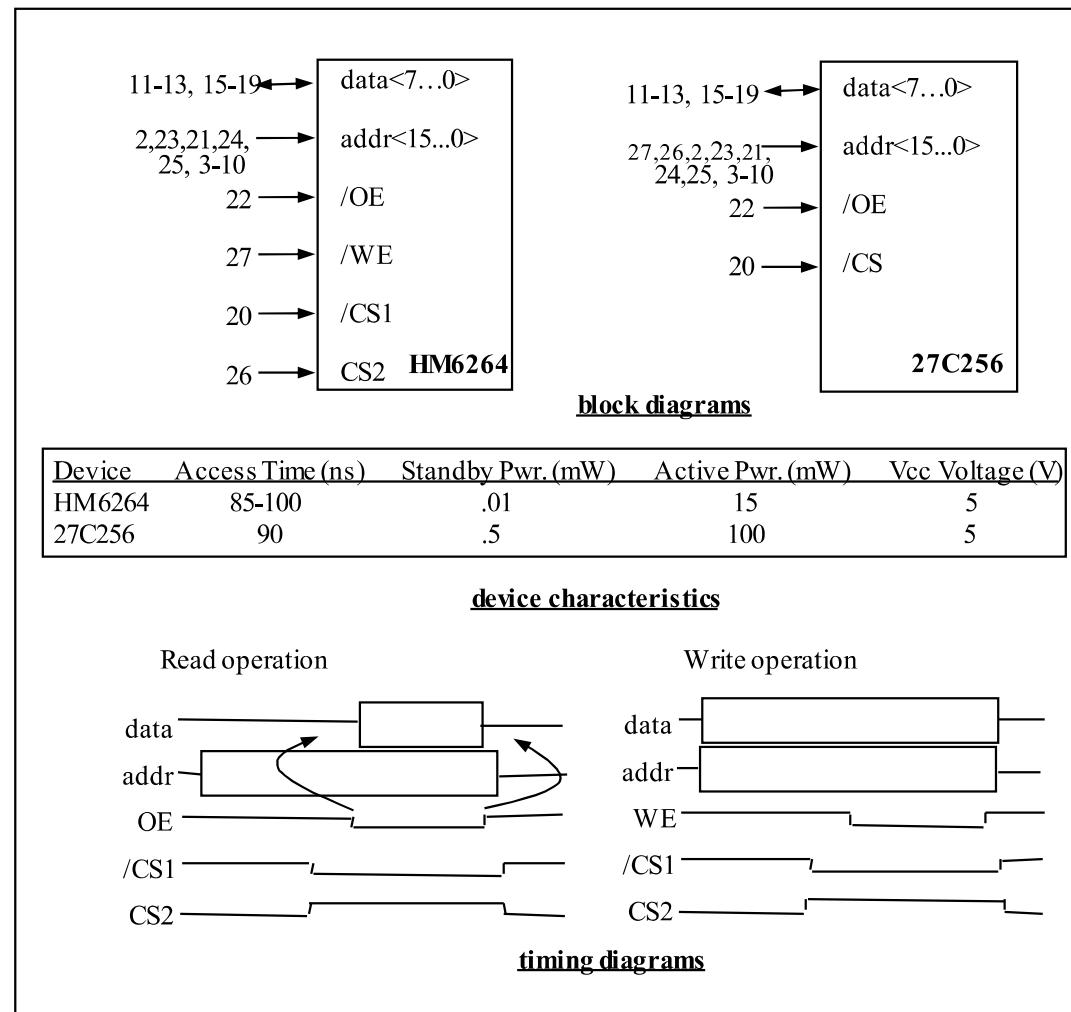
Ram variations

- 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
 - writes as fast as reads
 - no limit on number of writes unlike nonvolatile ROM-based memory
 - SRAM with EEPROM or flash
 - stores complete RAM contents on EEPROM or flash before power turned off

Example:

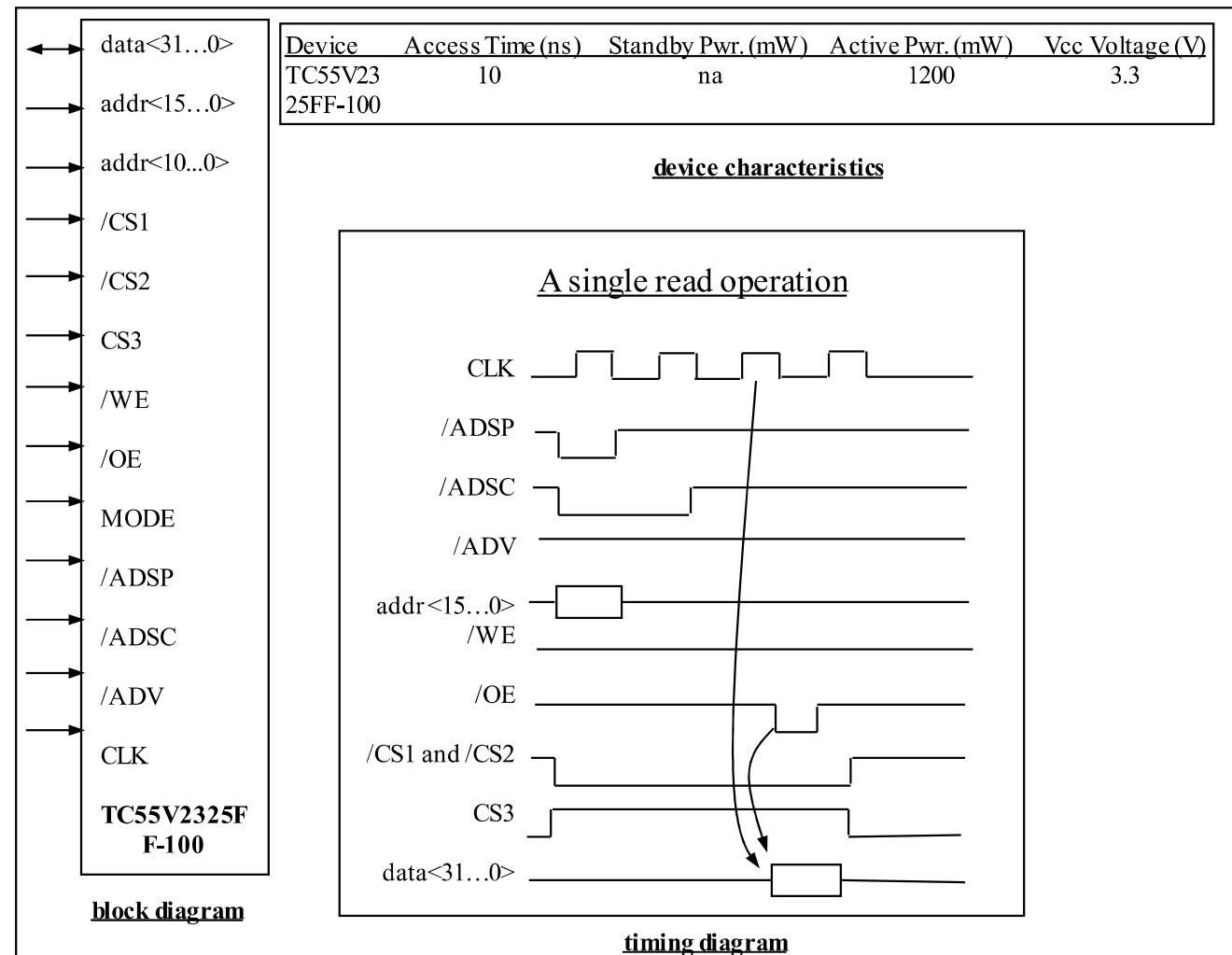
HM6264 & 27C256 RAM/ROM devices

- 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



Example: TC55V2325FF-100 memory device

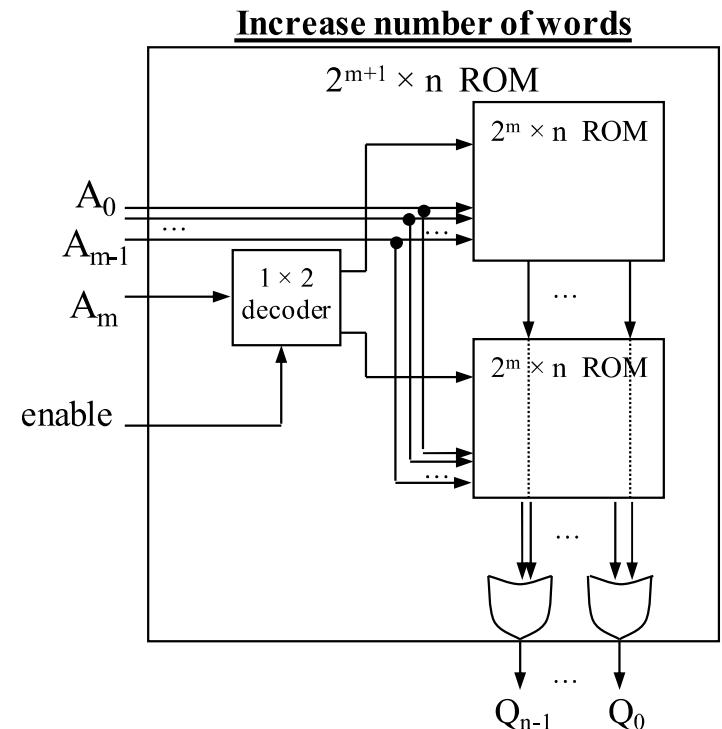
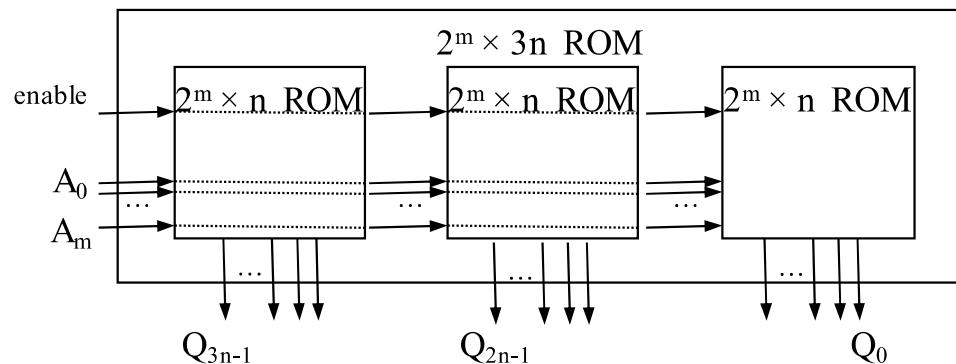
- 2-megabit synchronous pipelined burst SRAM memory device
- Designed to be interfaced with 32-bit processors
- Capable of fast sequential reads and writes as well as single byte I/O



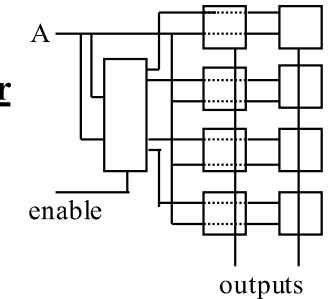
Composing memory

- 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
 - Connect side-by-side to increase width of words
 - Connect top to bottom to increase number of words
 - added high-order address line selects smaller memory containing desired word using a decoder
 - Combine techniques to increase number and width of words

Increase width of words

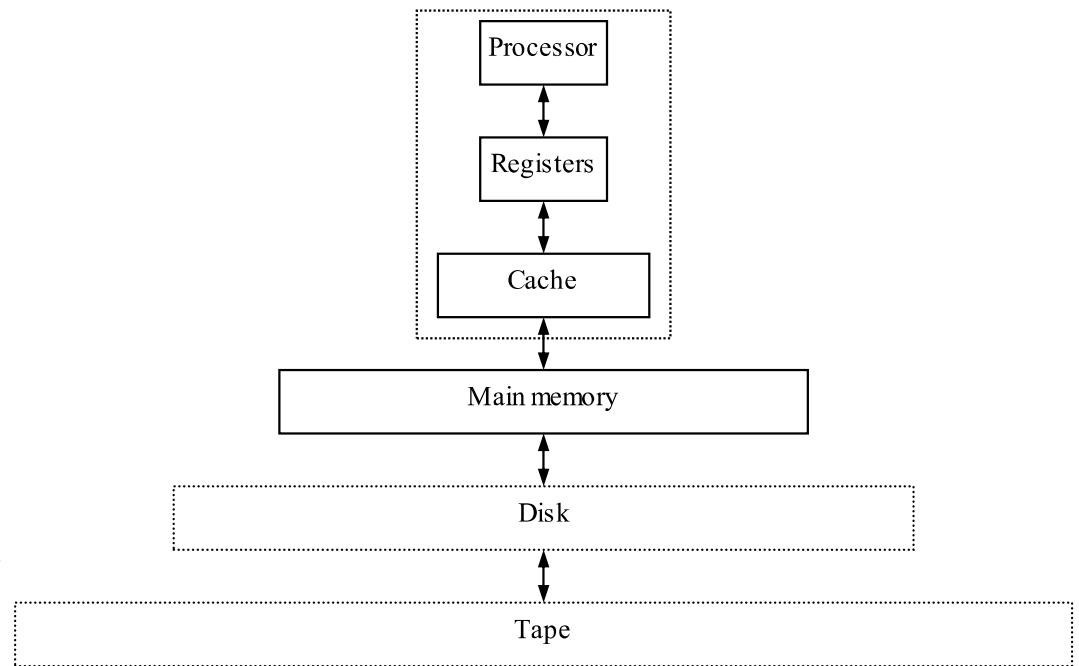


Increase number and width of words



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



Cache

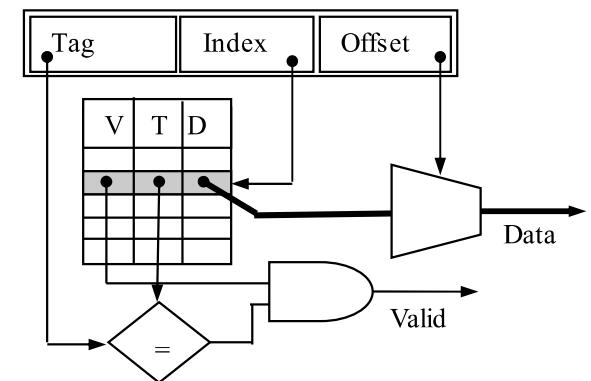
- **Usually designed with SRAM**
 - faster but more expensive than DRAM
- **Usually on same chip as processor**
 - space limited, so much smaller than off-chip main memory
 - faster access (1 cycle vs. several cycles for main memory)
- **Cache operation:**
 - Request for main memory access (read or write)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

Cache mapping

- Far fewer number of available cache addresses
- Are address' contents in cache?
- Cache mapping used to assign main memory address to cache address and determine hit or miss
- 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

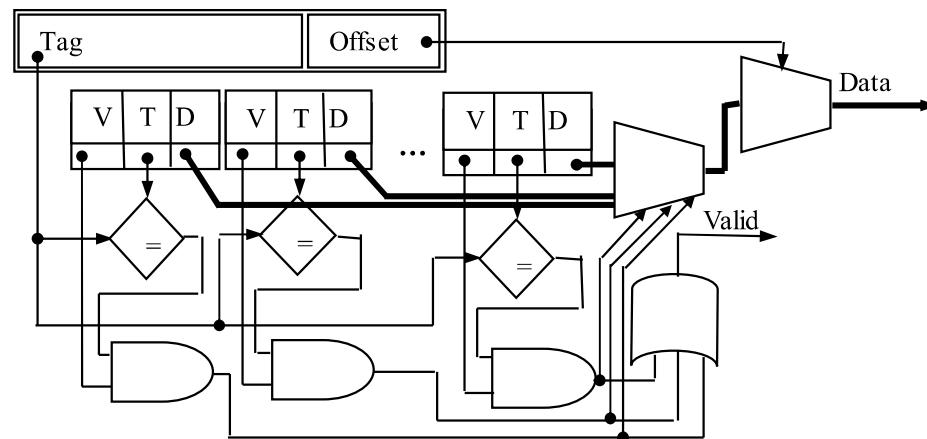
Direct mapping

- 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



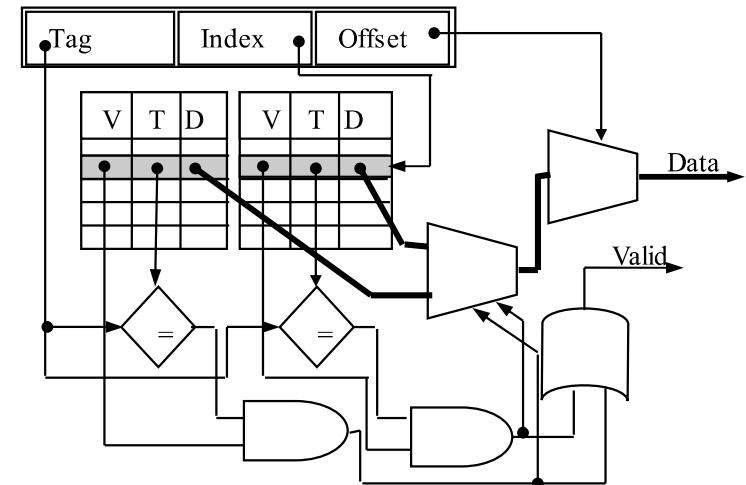
Fully associative mapping

- Complete main memory address stored in each cache address
- All addresses stored in cache simultaneously compared with desired address
- Valid bit and offset same as direct mapping



Set-associative mapping

- Compromise between direct mapping and fully associative mapping
- Index same as in direct mapping
- But, 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 line 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

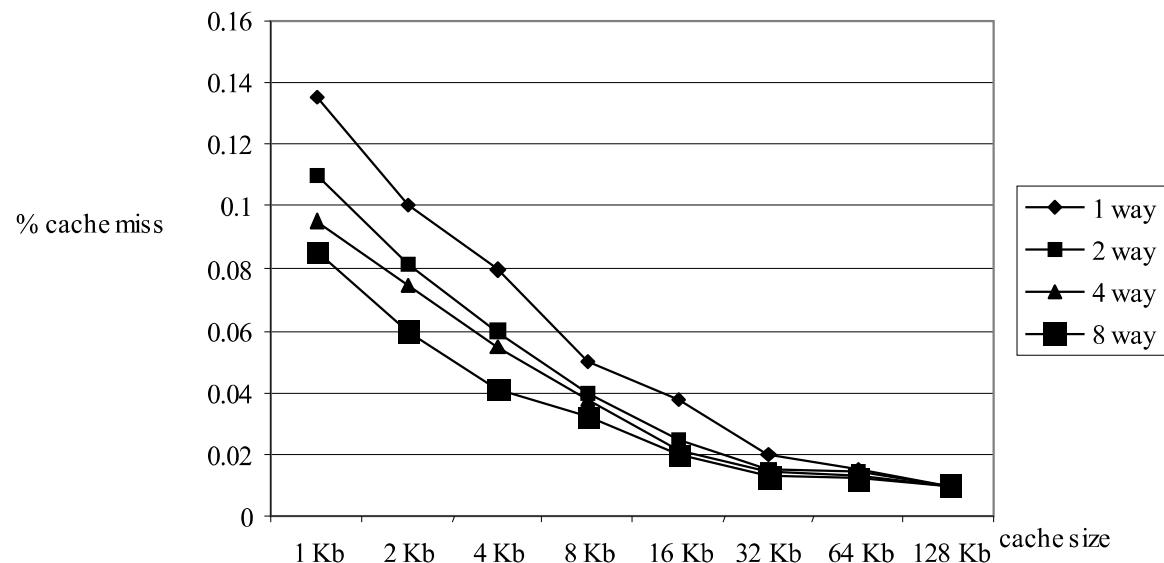
- When written, data cache must update main memory
- Write-through
 - write to main memory whenever cache is written to
 - easiest to implement
 - processor must wait for slower main memory write
 - potential for unnecessary writes
- Write-back
 - main memory only written when “dirty” block replaced
 - 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
 - e.g.,
 - 2 Kbyte cache: miss rate = 15%, hit cost = 2 cycles, miss cost = 20 cycles
 - avg. cost of memory access = $(0.85 * 2) + (0.15 * 20) = 4.7$ cycles
 - 4 Kbyte cache: miss rate = 6.5%, hit cost = 3 cycles, miss cost will not change
 - avg. cost of memory access = $(0.935 * 3) + (0.065 * 20) = 4.105$ cycles **(improvement)**
 - 8 Kbyte cache: miss rate = 5.565%, hit cost = 4 cycles, miss cost will not change
 - avg. cost of memory access = $(0.94435 * 4) + (0.05565 * 20) = 4.8904$ cycles **(worse)**

Cache performance trade-offs

- Improving cache hit rate without increasing size
 - Increase line size
 - Change set-associativity

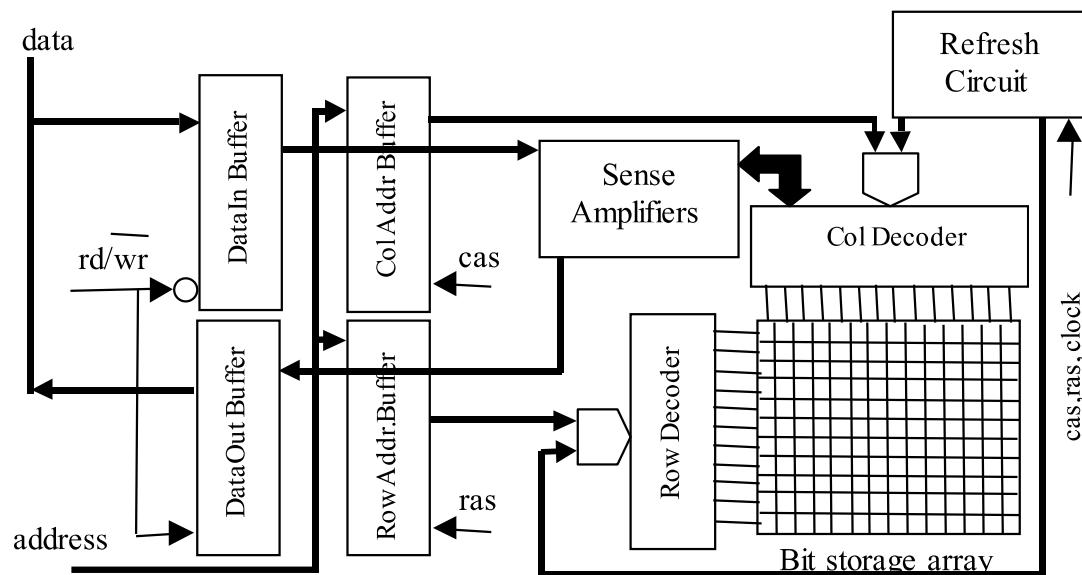


Advanced RAM

- 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

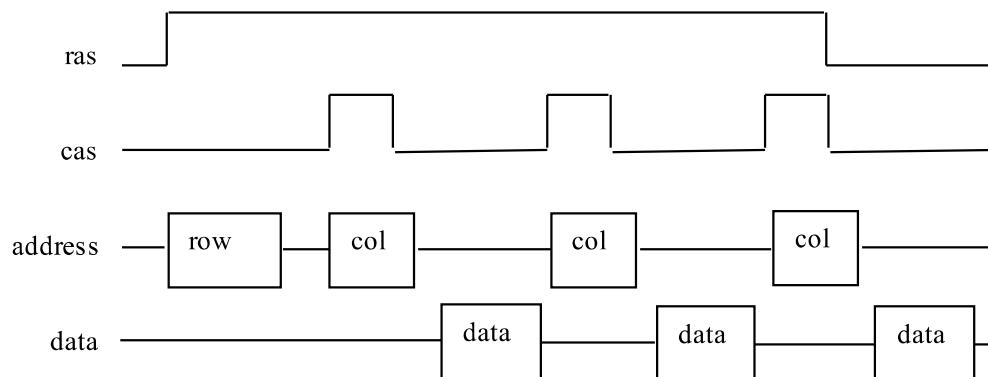
Basic DRAM

- Address bus multiplexed between row and column components
- Row and column addresses are latched in, sequentially, by strobing *ras* and *cas* signals, respectively
- Refresh circuitry can be external or internal to DRAM device
 - strobos consecutive memory address periodically causing memory content to be refreshed
 - Refresh circuitry disabled during read or write operation



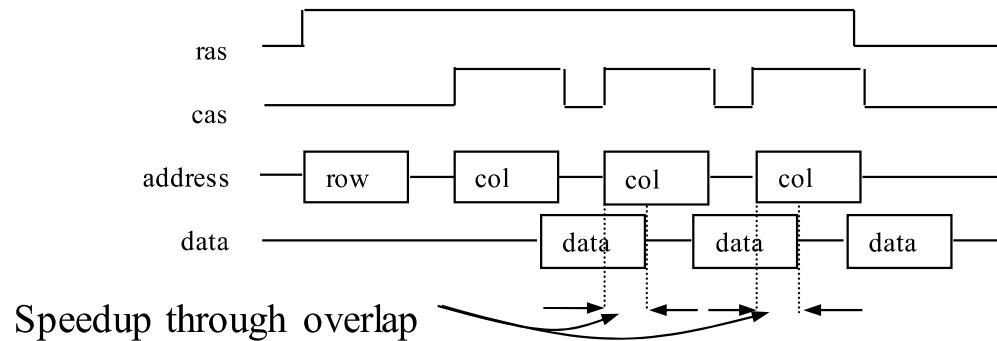
Fast Page Mode DRAM (FPM DRAM)

- 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



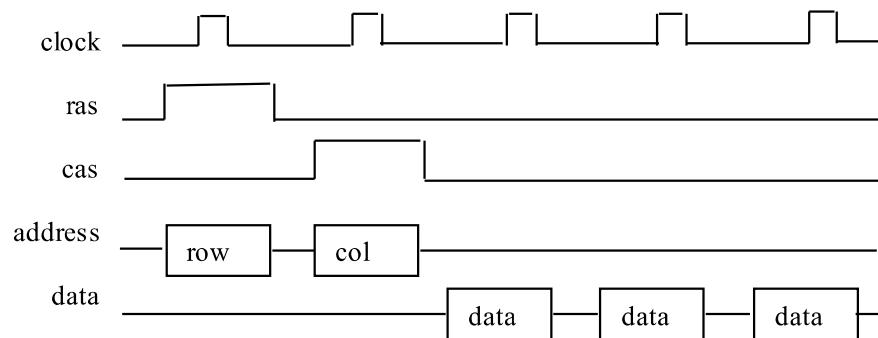
Extended data out DRAM (EDO DRAM)

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



(S)ynchronous and Enhanced Synchronous (ES) 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
 - ESDRAM improves SDRAM
 - added buffers enable overlapping of column addressing
 - faster clocking and lower read/write latency possible



Rambus DRAM (RDRAM)

- More of a bus interface architecture than DRAM architecture
- Data is latched on both rising and falling edge of clock
- Broken into 4 banks each with own row decoder
 - can have 4 pages open at a time
- Capable of very high throughput

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

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