Memory Basics

Lan-Da Van (范倫達), *Ph. D.*
Department of Computer Science
National Chiao Tung University
Taiwan, R.O.C.
Spring, 2011

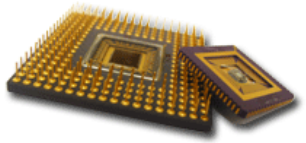
ldvan@cs.nctu.edu.tw

http://www.cs.nctu.edu.tw/~ldvan/

- **Source: M. Morris Mano and Charles R. Kime, *Logic and Computer Design Fundamentals*, 3rd Edition, 2004, Prentice Hall.**

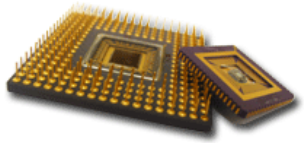
Outlines

- Memory Definitions
- Random-Access Memory
- SRAM Integrated Circuits
- Array of SRAM ICs
- DRAM Integrated Circuits
- DRAM Types
- Arrays of DRAM ICs
- Summary



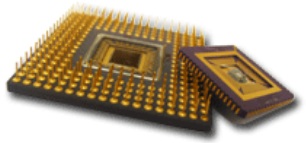
Memory Operation in Digital Computers

- Programs and data that cannot be altered are stored in ROM. Other large programs are maintained on magnetic disks.
- When power is turned on, the computer can use the programs from ROM. The other programs residing on a magnetic disk are then transferred into the computer RAM as needed.
- Before turning the power off, the binary information from the computer RAM is transferred into the disk for the information to be retained.



Memory Definitions

- Memory:
  - is a collection of cells capable of storing binary information
  - Binary storage cells + Associated circuits for storing and retrieving the information.
  - Operations:
    - *Write* op: storing new information in memory
    - *Read* op: transferring the stored information out of memory
  - Two types of memories used in digital systems:
    1. ROM: Read-only memory (only *Read* op) $\Rightarrow$ PLD
       - Stores data permanently: a suitable binary information is already stored inside the memory.
    2. RAM: Random-access memory (*Write* & *Read* ops)
       - Stores data temporarily.
       - Applications: cache, main memory

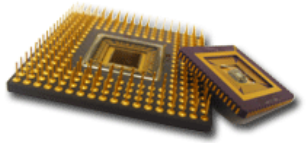


Memory Size

- *Word*:
  - an entity of bits that move in & out of memory as a unit
  - may represent a number, an instruction, one or more alphanumeric characters, or other binary-coded information
- *Byte*: a group of 8 bits
- *Capacity* of a memory unit:
  - total # of bytes it can store or # words, # bits/word

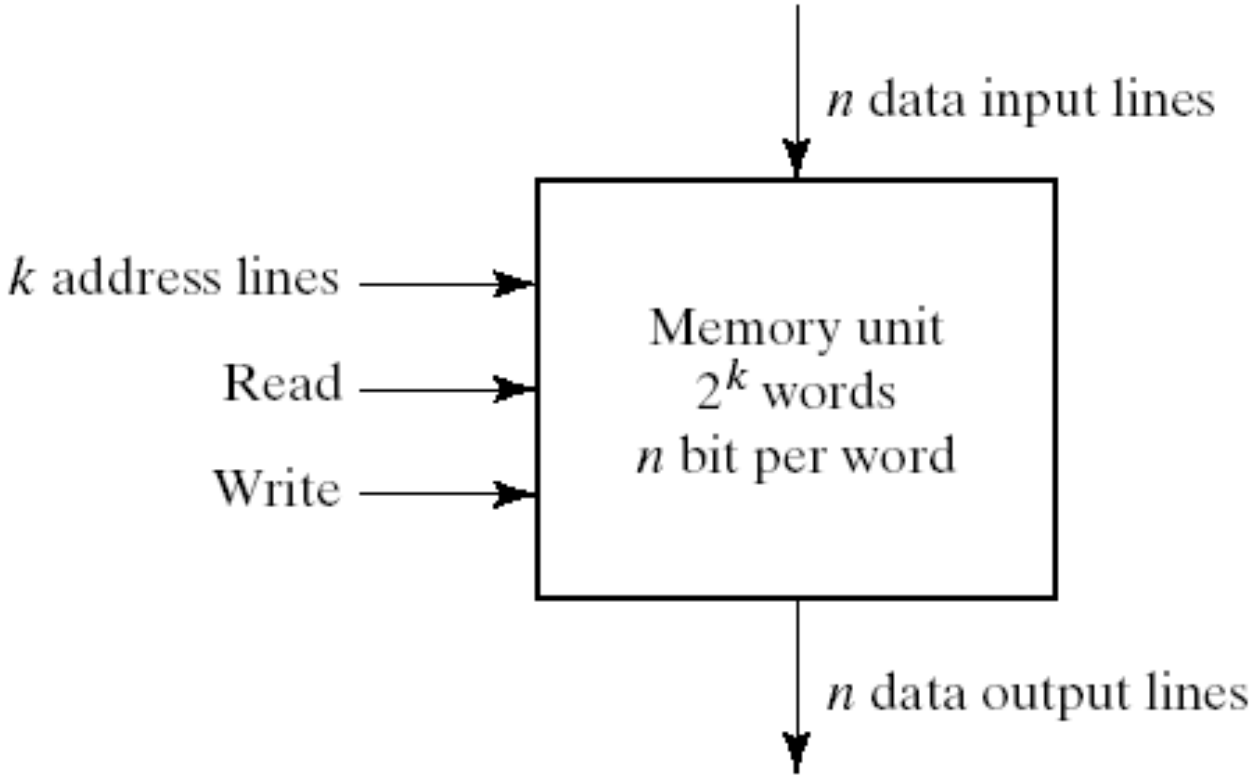
- Example: a 1024×16 memory $\Rightarrow$ $2^{10} \times 16$

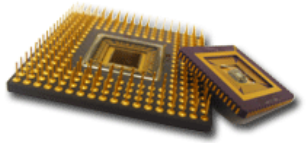
Memory address		Memory contest
Binary	decimal	
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
⋮	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1011111000100101



Memory Block Diagram

- Block diagram of a memory unit:

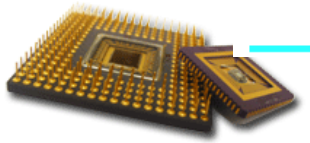




Write & Read Operations

- *Write* op: transfer-in
  - Step 1: Apply the binary addr of the desired word to the addr lines.
  - Step 2: Apply the data bits that must be stored in memory to the data input lines.
  - Step 3: Activate the *Write* input.
- *Read* op: transfer-out
  - Step1: Apply the binary addr of the desired word to the addr lines.
  - Step 2: Activate the *read* input.

Chip select CS	Read/$\overline{\text{Write}}$ R/$\overline{\text{W}}$	Memory operation
0	×	None
1	0	Write to selected word
1	1	Read from selected word

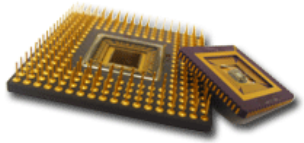


Timing Waveforms (1/2)

- Operation of a memory unit:
  - is controlled by an external device such as a CPU.
    - The CPU is usually synchronized by its own clock.
    - The memory does not employ an internal clock.

      Its read & write operations are specified by control inputs.

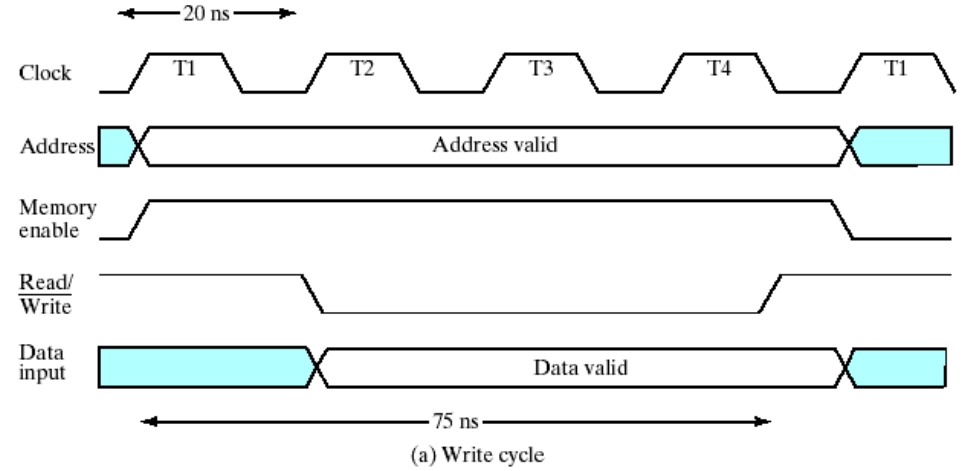
    ⇒ The CPU must provide the memory control signals to synchronize its internal clocked operations w/ the read and write operations of memory.
- Access time:
  - *read cycle time*: the max time from the application of the addr to the appearance of the data at the Data Output
  - *write cycle time*: the max time from the application of the addr to the completion of all internal memory ops required to store a word
  - The access time of the memory must be related within the CPU to a period equal to a fixed # of CPU clock cycles.



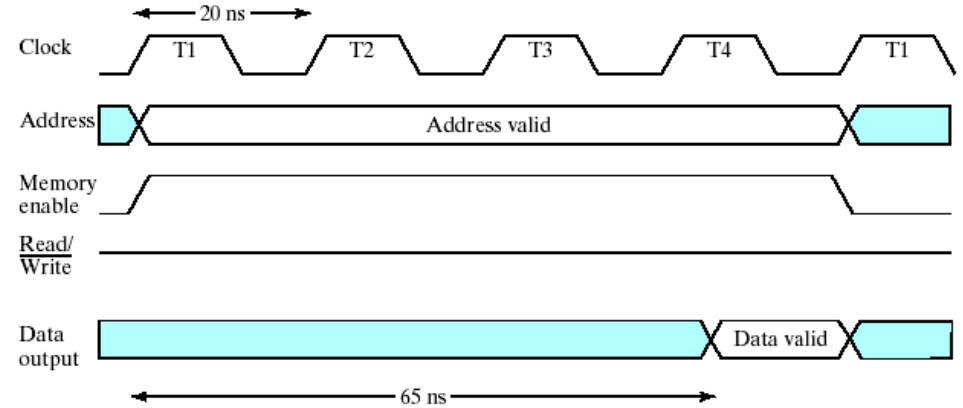
Timing Waveforms (2/2)

- Example: CPU – 50MHz clock frequency (20 ns)

  Memory – write time = 75 ns & read time = 65 ns



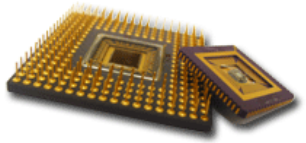
(a) Write cycle



(b) Read cycle

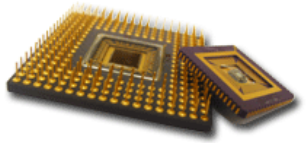
Volatile vs. Nonvolatile Memory

- Volatile:
  - Lose stored information when power is turned off.
  - Memory cells can be accessed to transfer information to or from any desired location, w/ the access taking the same time regardless of the location.
  - E.g.: static RAM, dynamic RAMs
- Nonvolatile:
  - Retains the stored information after the removal of power
  - takes different lengths of time to access information, depending on where the desired location is relative of the current physical position of the disk or tape
  - E.g.: ROM, magnetic disk, tape



Volatile Memory

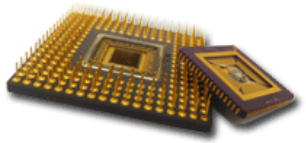
- Operating modes of RAMs:
  1. Static RAM: SRAM
     - Consists of internal latches that store the binary information.
     - The stored information remains valid as long as power is applied to the RAM.
     - Advs.: easier to use, shorter read and write cycles, & no refresh
  2. Dynamic RAM: DRAM
     - Stores the binary information in the form of electric charges on capacitors.
     - The capacitors must be periodically recharged by *refreshing* the DRAM (every few milliseconds).
     - Advs.: reduced power consumption & larger storage capacity



SRAM Integrated Circuits

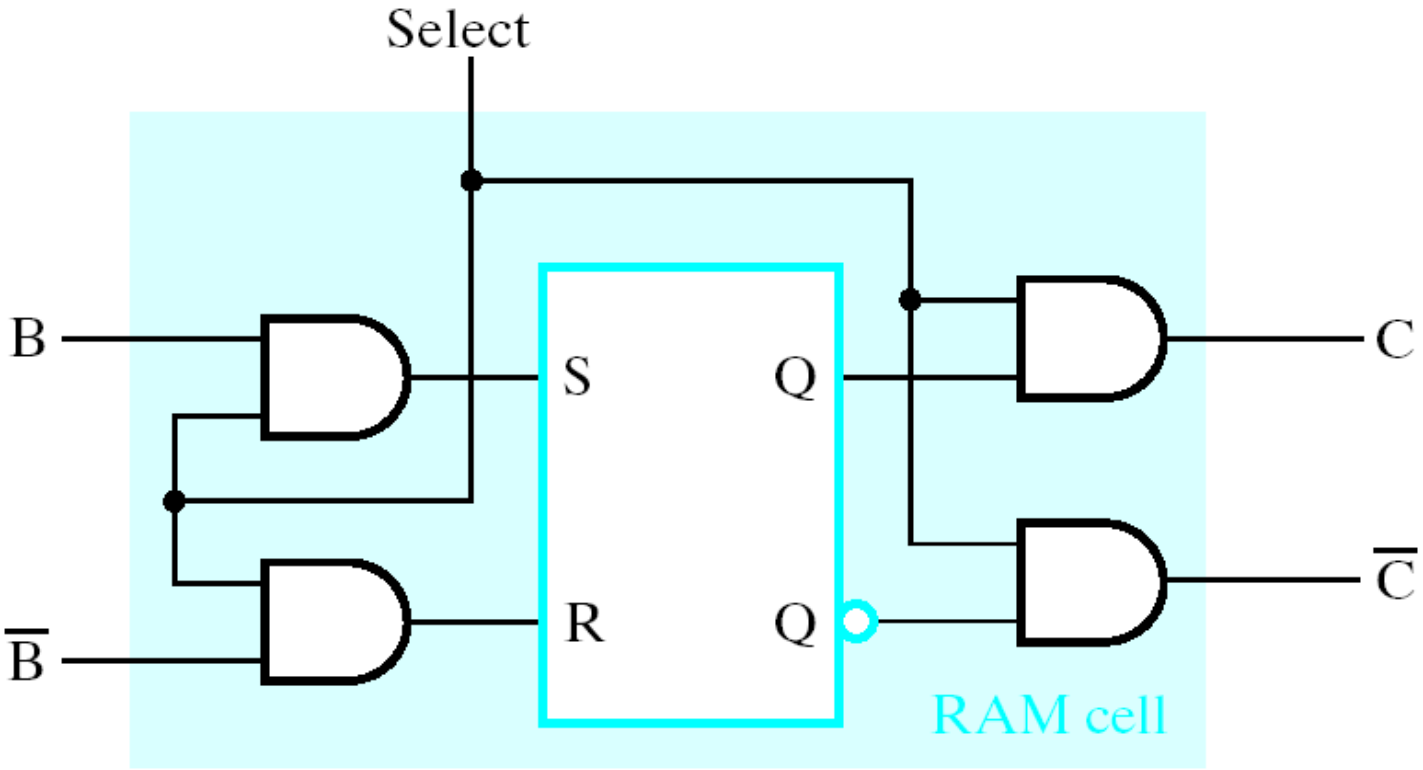
- Memory unit: storage components + decoding ckts
  - Decoding ckts: select the memory word specified by the input address
- Internal structure of a RAM chip: *m* words, *n* bits/word

  $\Rightarrow$ $m \times n$ binary storage cells + associated decoding ckts
  - *RAM cell*: the basic binary storage cell used in a RAM chip



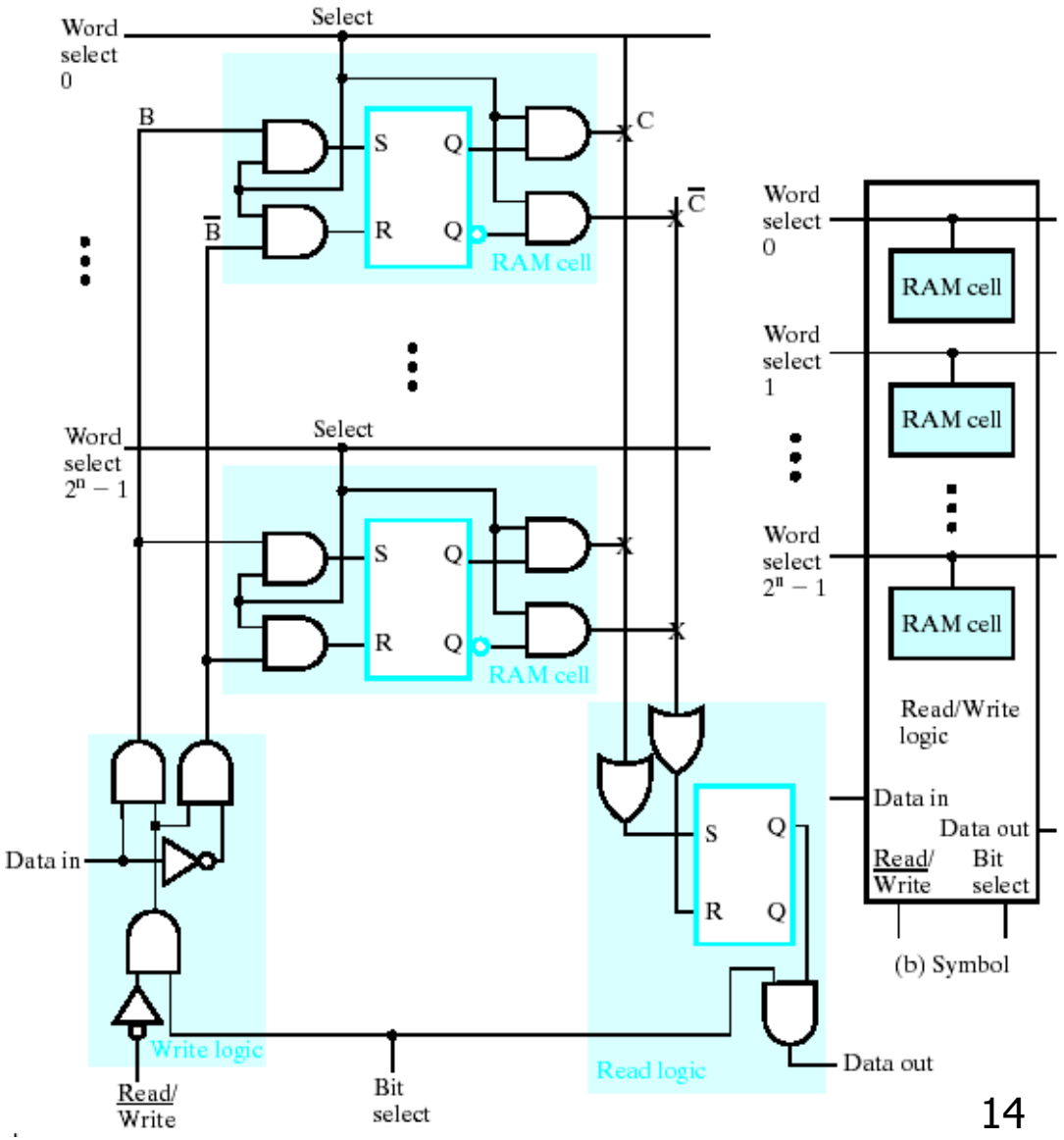
SRAM Cell

- SRAM cell: stores one bit information



SRAM Bit Slice

- SRAM bit slice:
  - Contains all the circuitry associated w/ a single bit position of a set of RAM words

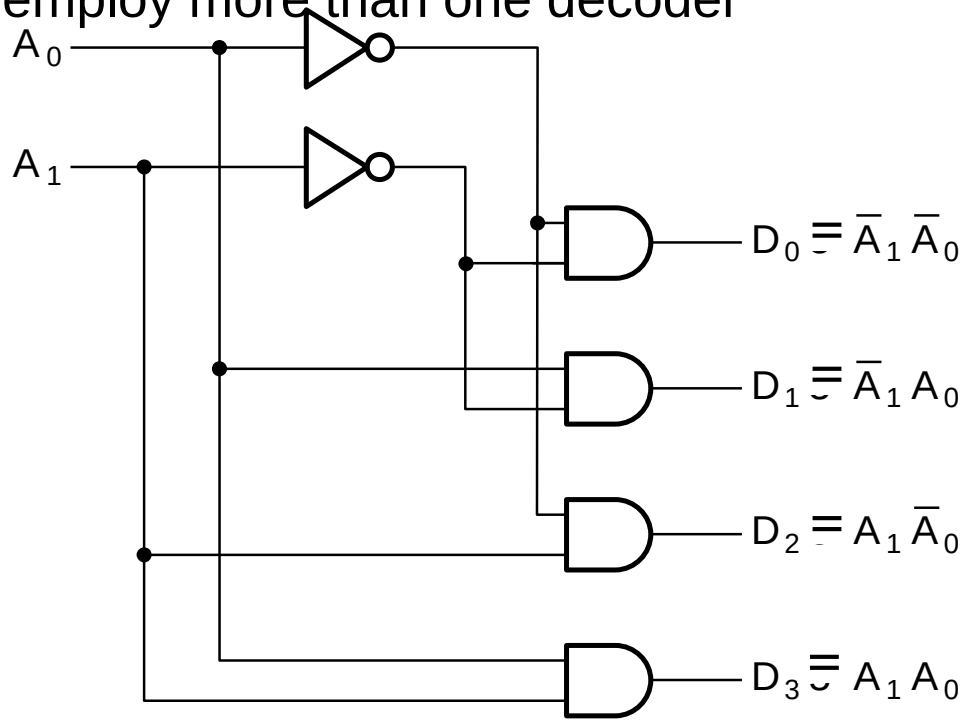


Decoding Circuits

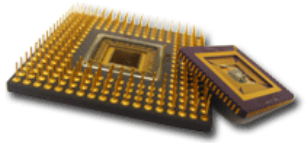
- Logical construction of a RAM:
  - 2^k words $\times$ n bits/word needs $2^k \times n$ binary storage cells + associated decoding ckts with k addr lines.
- Decoding ckts:
  - Linear decoding: employ a $k \times 2^k$ decoder
  - Coincident decoding: employ more than one decoder

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

(a)

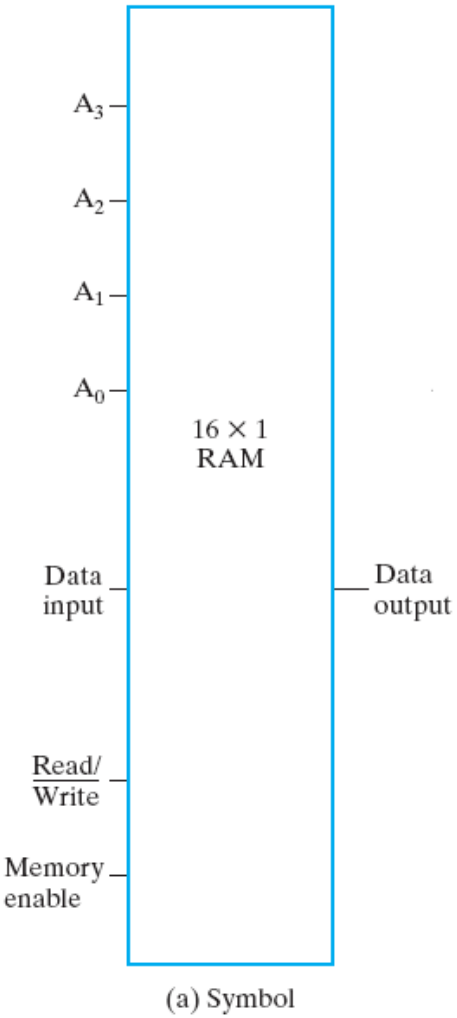


(b)

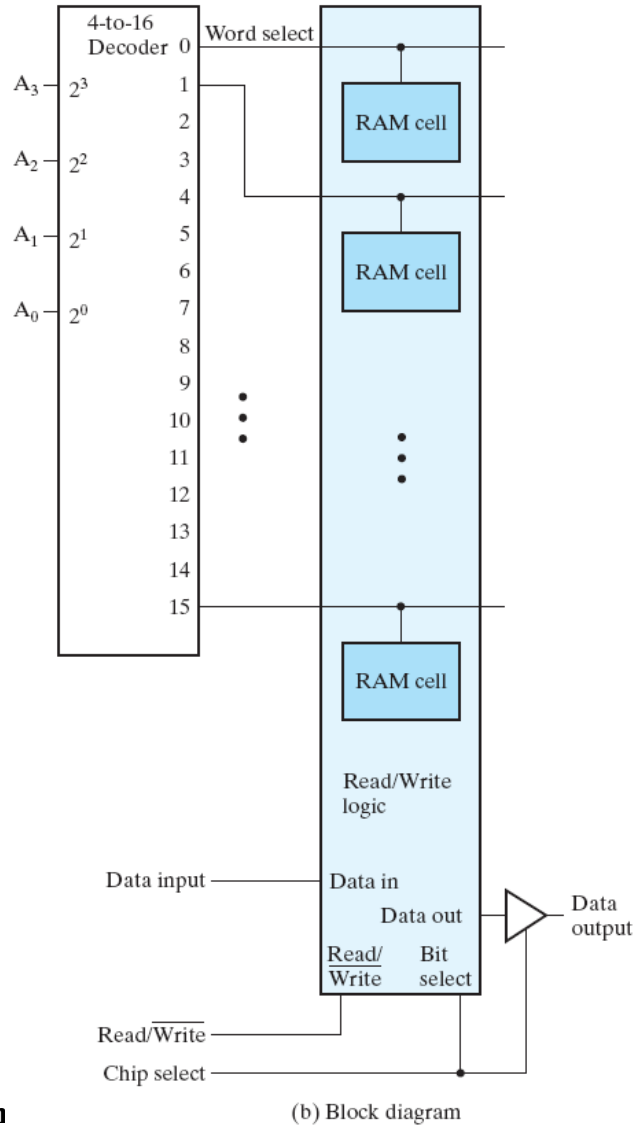


Linear Decoding

- E.g.: 16 ×1 SRAM chip



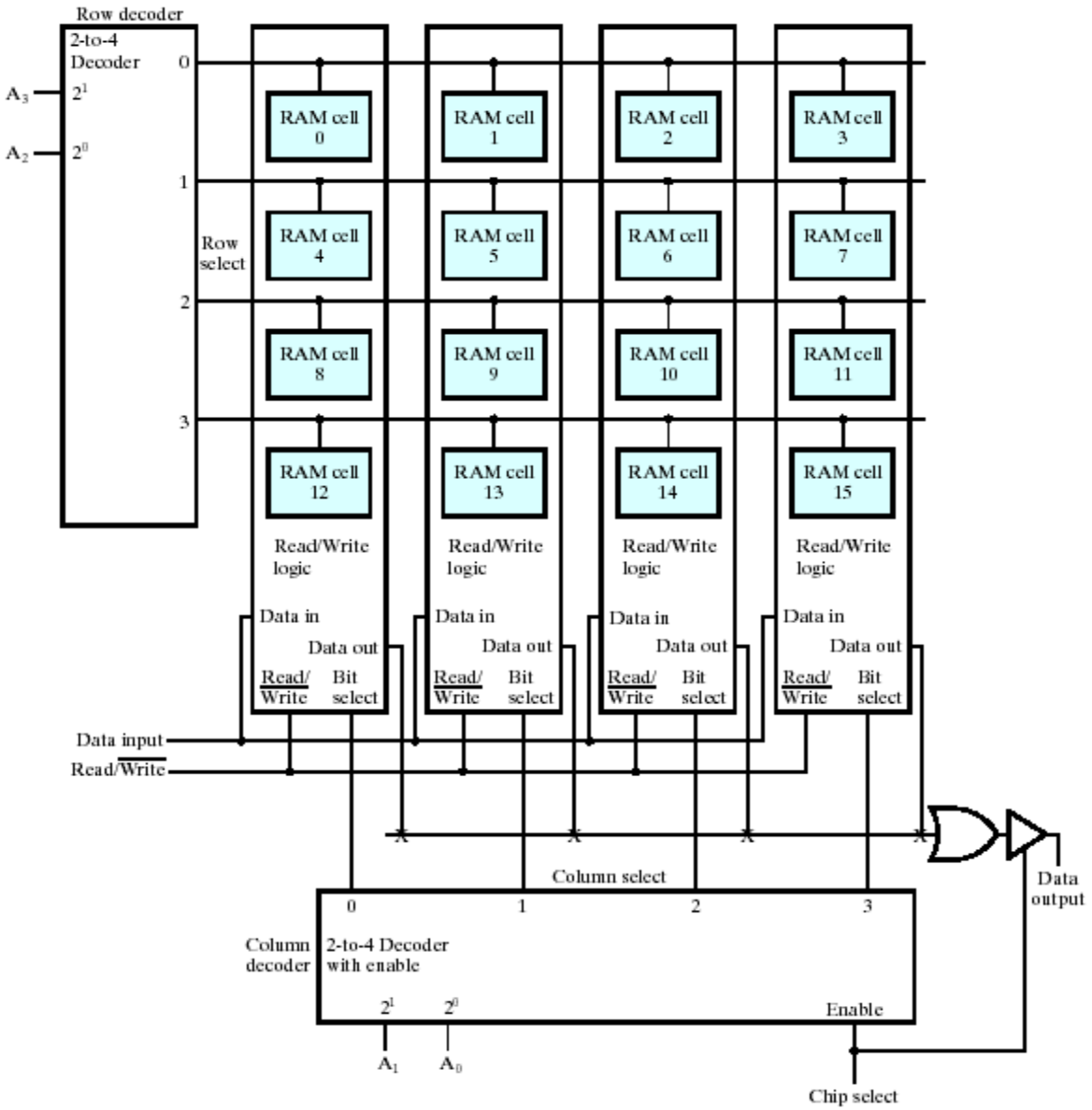
(a) Symbol

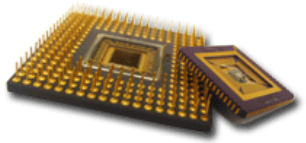


(b) Block diagram

Coincident Decoding (1/2)

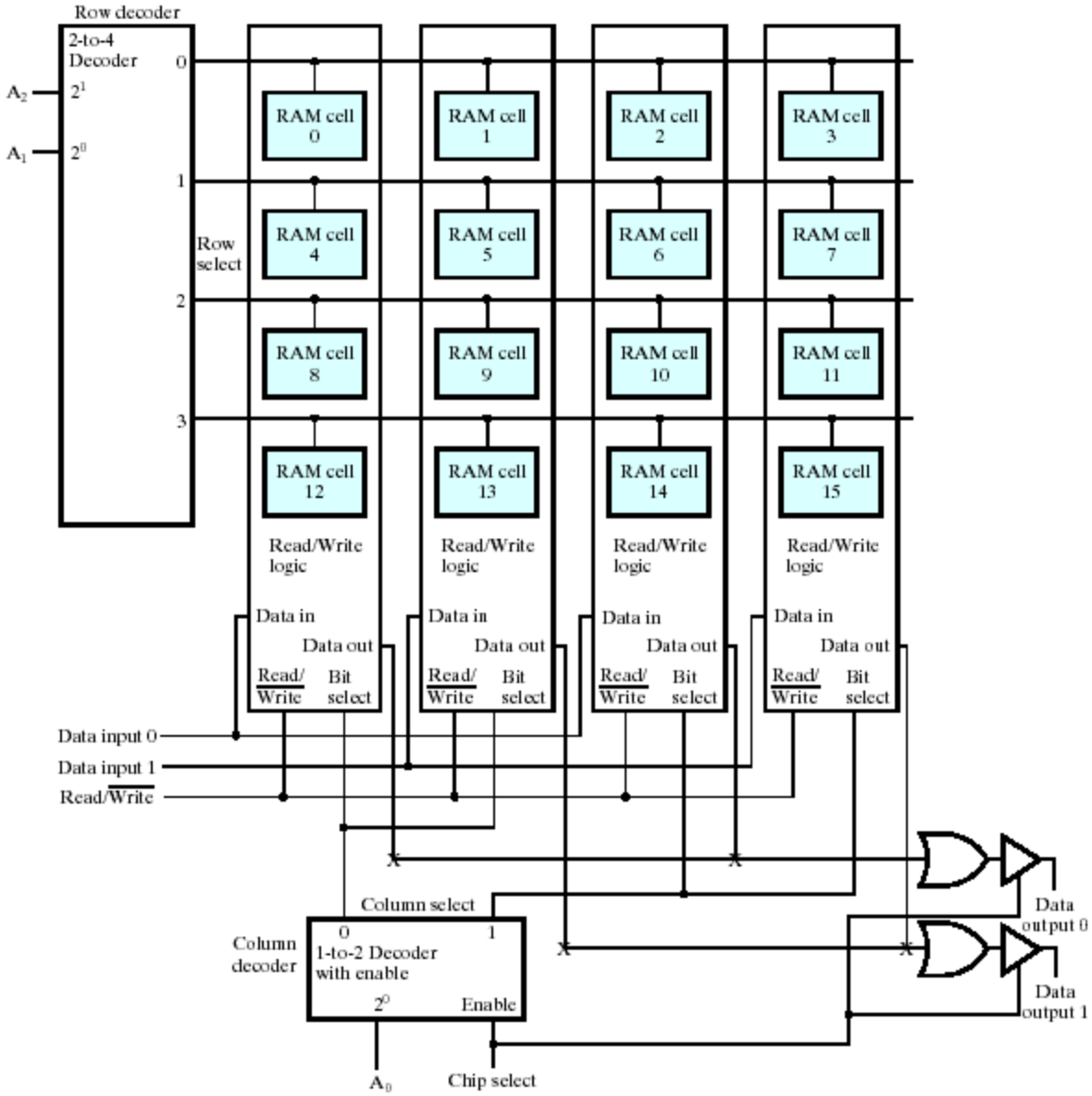
- E.g.: a 16x1 SRAM using a 4x4 SRAM cell array

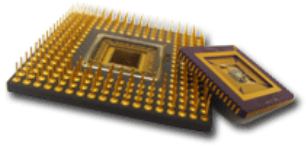




Coincident Decoding (2/2)

- E.g.: a 8x2 SRAM using a 4x4 SRAM cell array

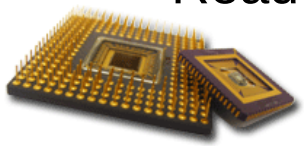




Linear Decoding vs Coincident Decoding (1/2)

- 2^k words
  - Linear decoding:
    - Employ a $k \times 2^k$ decoder
  - Coincident decoding:
    - Employ two decoders in a two-dimensional selection scheme
    - Basic idea: arrange the memory cells in an array that is close as possible to **square**
    - E.g.: $\Rightarrow$ two $k/2$-input decoders (row selection & column selection)

	Linear	**Coincident (2D)**
Decoder	1 $k \times 2^k$	2 $k/2 \times 2^{k/2}$
# AND gates	2^k	$2 \times 2^{k/2}$ $(= 2^{k/2+1})$
# inputs/gates	k	$k/2$
Read & write times	longer	shorter



Linear Decoding vs Coincident Decoding (2/2)

- E.g.: For a 32K×8 RAM
  - Linear selection scheme:
    - A single 15-to-2^{15} line decoder:

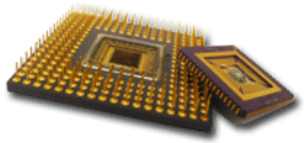
      $\Rightarrow$ 32,768 AND gates w/ 15 inputs in each
  - Coincident selection:
    - Make the # of rows and columns in the array equal:

      $32K \times 8 = 256K \text{ bits} = 2^{18} \text{ bits} = 2^9 \times 2^9$

      $= 2^9 \times (2^6 \times 8) \text{ bits}$

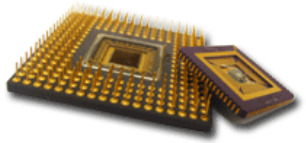
      $\Rightarrow$ a 9-to-512 line decoder (row) & a 6-to-64 line decoder (column)

      $\Rightarrow 2^9+2^6=576$ AND gates



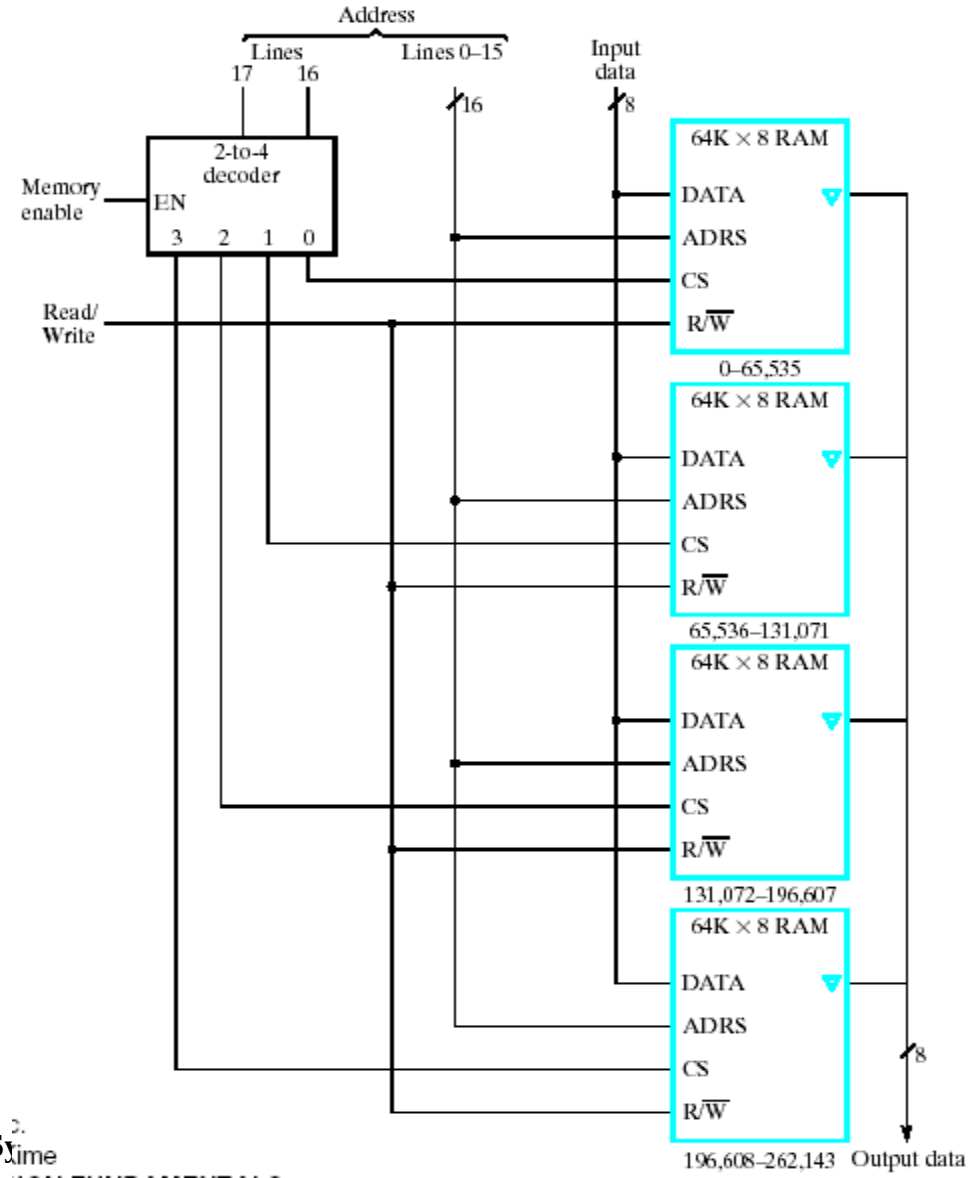
Array of SRAM ICs (1/3)

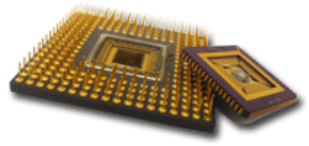
- Memory unit:
  - Capacity: # words X # bits/word
- Symbol for a RAM chip:
  - E.g.: 64K × 8 RAM



Array of SRAM ICs (2/3)

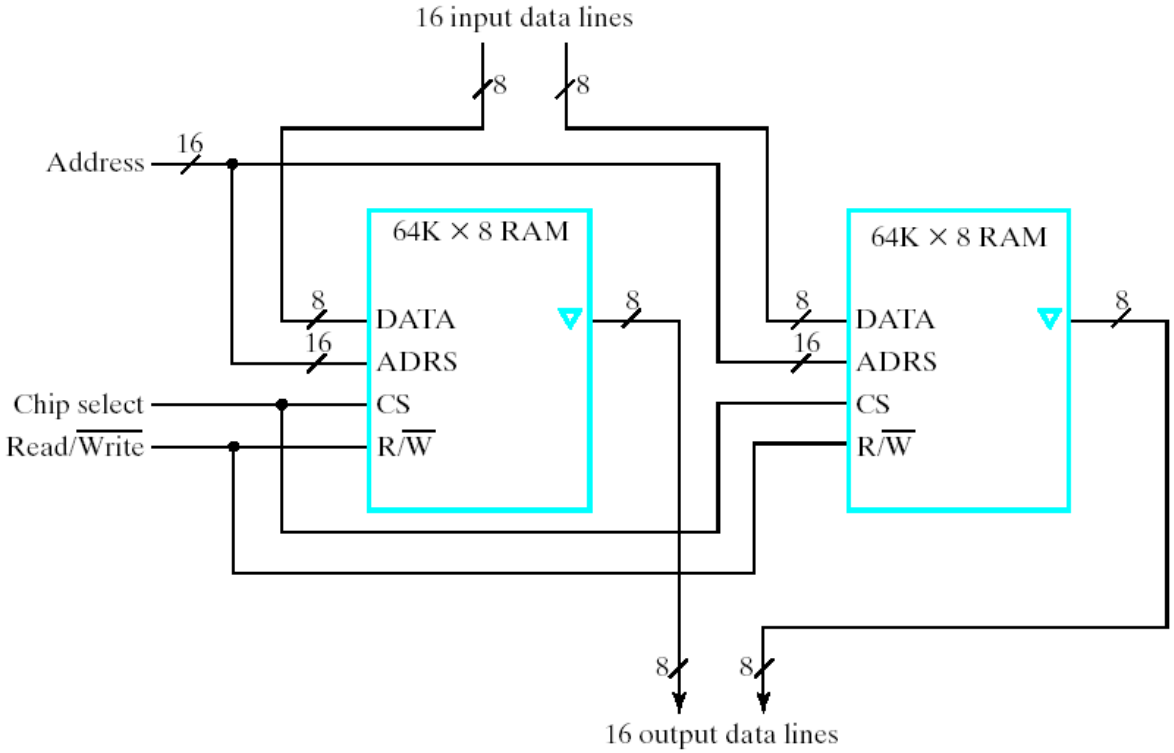
- Increasing # of Words in the Memory:
  - E.g.: Construct a 256K×8 SRAM by using 64K×8 SRAM Ics
  - Ans: 256K×8 RAM = $2^2 \times 2^{16} \times 8$
  - => four 64K×8 RAM ICs





Array of SRAM ICs (3/3)

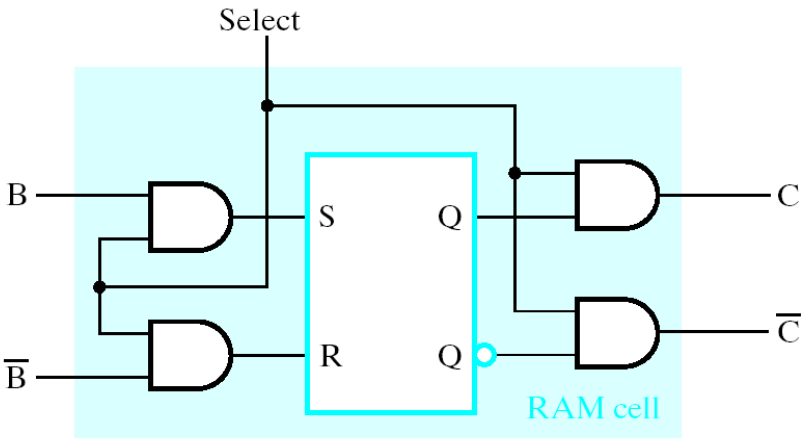
- Increasing # Bits/Word in the Memory
  - E.g.: Construct a 64K×16 SRAM by using 64K×8 SRAM Ics
  - Ans: 64K×16 SRAM ⇐ two 64K×8 SRAM ICs



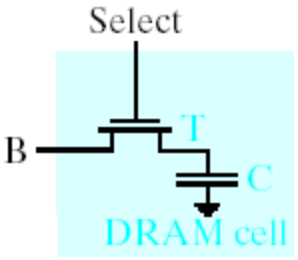
DRAM Cell (1/2)

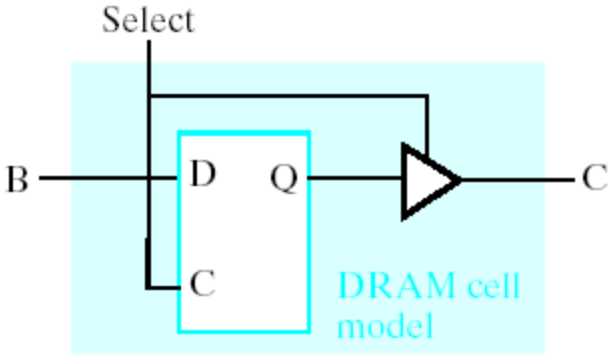
- SRAM cell vs. DRAM cell:

SRAM cell



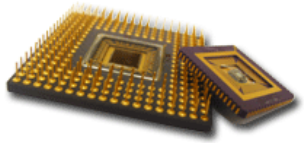
DRAM cell



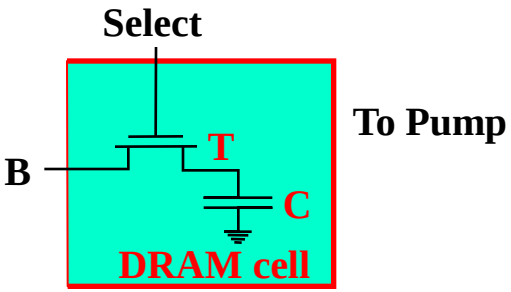


- 6 transistors

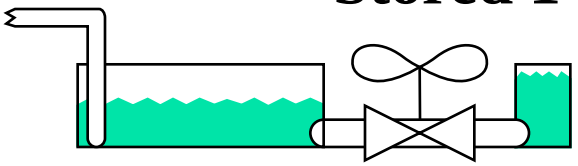
- 1 transistor & 1 capacitor
- Destructive read ⇒ Restore
- Leaks ⇒ Refresh



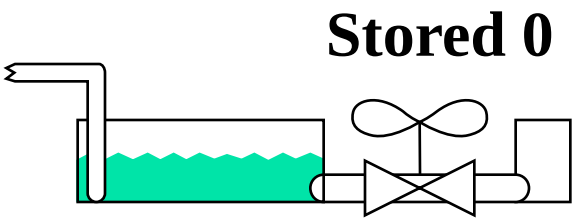
DRAM Cell (2/2)



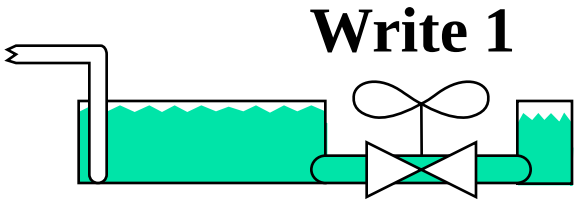
(a)



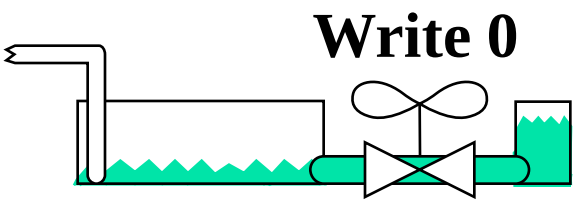
(b)



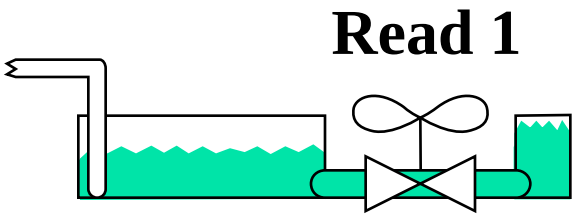
(c)



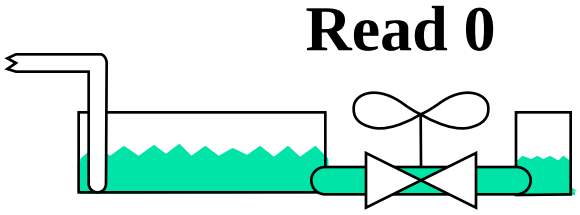
(d)



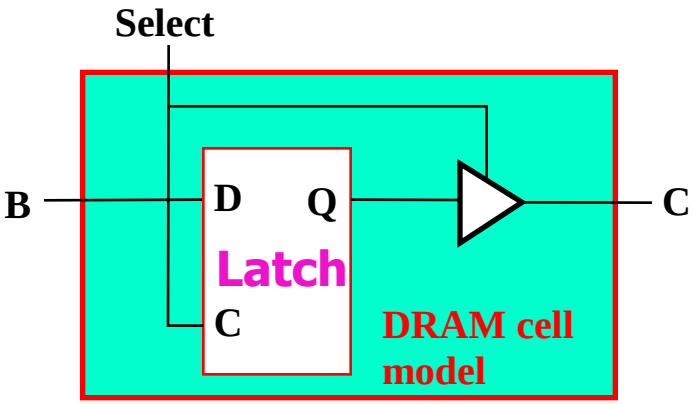
(e)



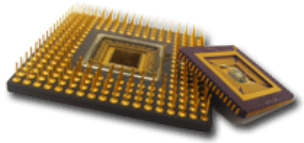
(f)



(g)



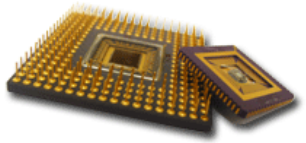
(h)



DRAM Property

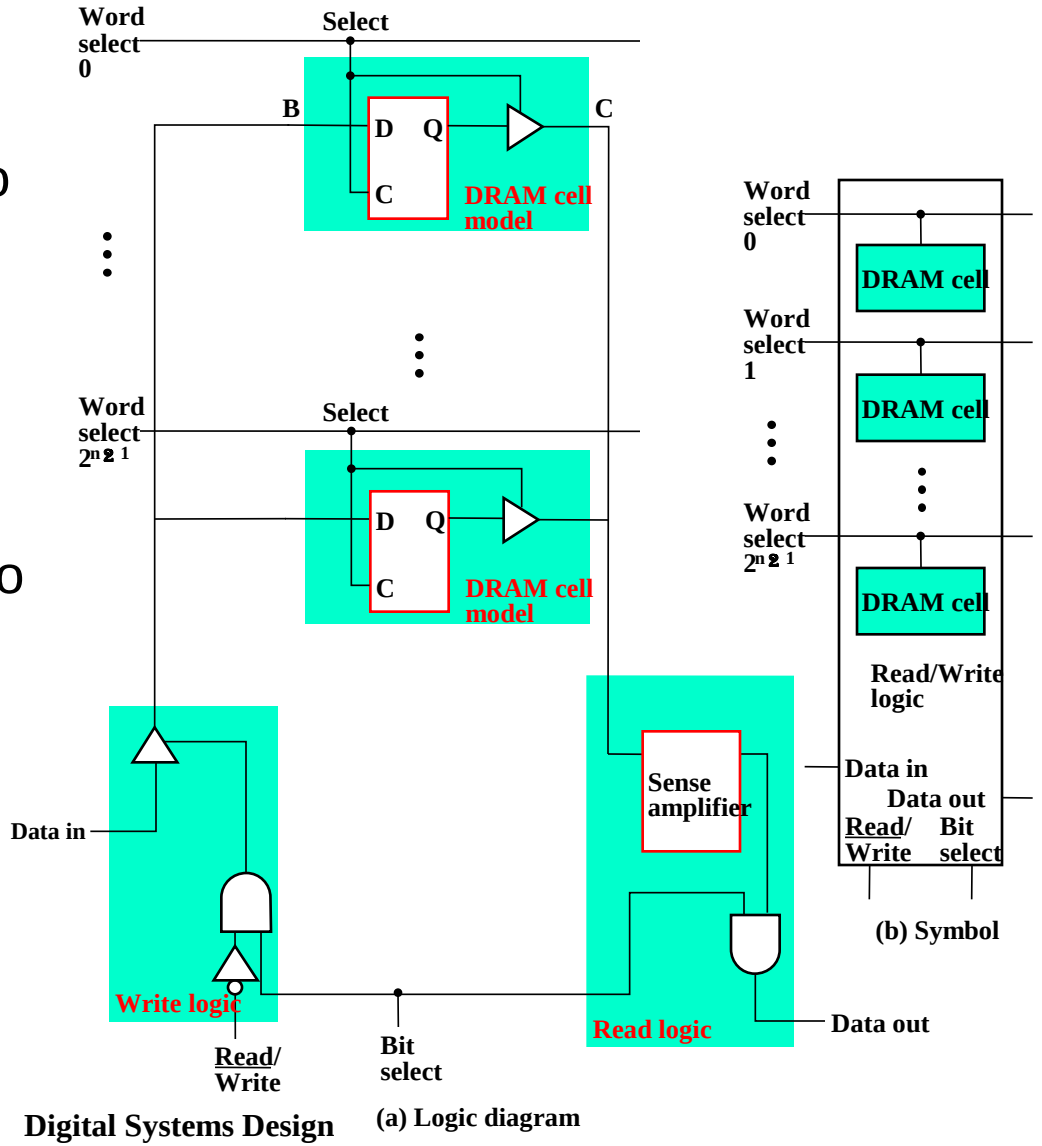
- Advs. of DRAM:
  - Density: have 4 times the density of SRAM.
  - Cost/bit: 3 to 4 times less than SRAM.
  - Power: lower power requirement

  $\Rightarrow$ DRAM is the preferred technology for large memories (e.g.: main memory).
- Disadv. of DRAM:
  - Its electronic design is considerably more challenging.



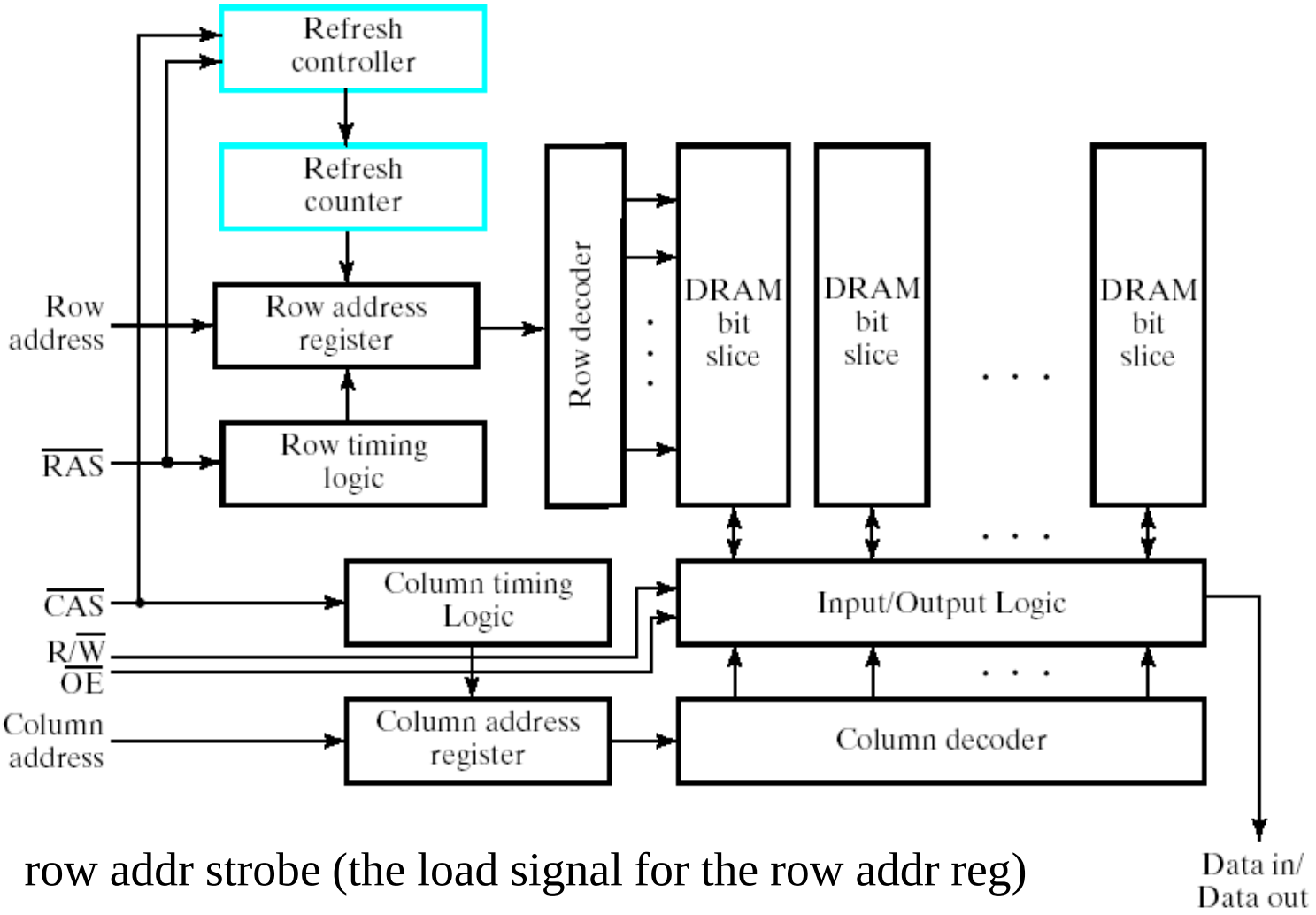
DRAM - Bit Slice

- C is driven by 3-state drivers
- Sense amplifier is used to change the small voltage change on C into H or L
- In the electronics, B, C, and the sense amplifier output are connected to make destructive read into non-destructive read



(a) Logic diagram

Block Diagram of a DRAM



$\overline{RAS}$: row addr strobe (the load signal for the row addr reg)

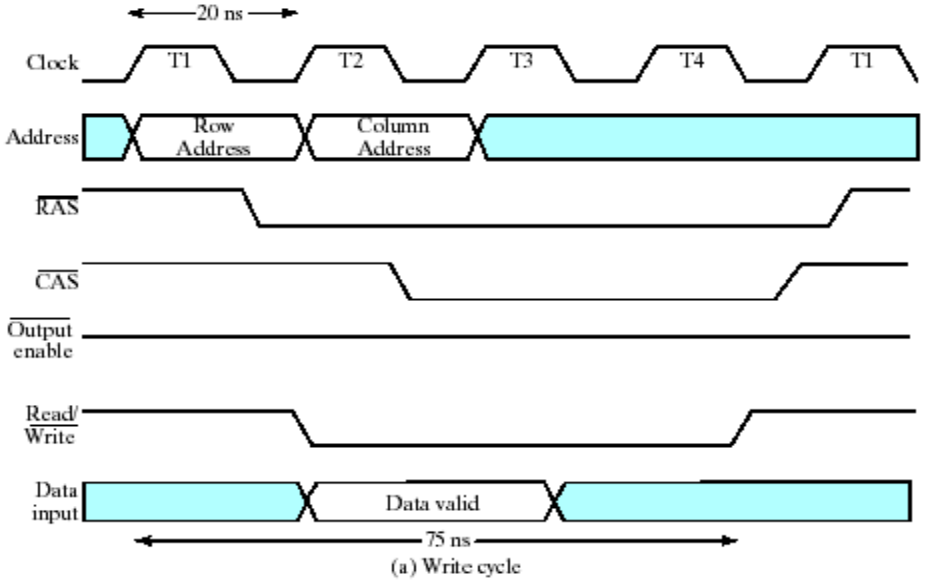
$\overline{CAS}$: column addr strobe (the load signal for the column addr reg)

Timing Waveform

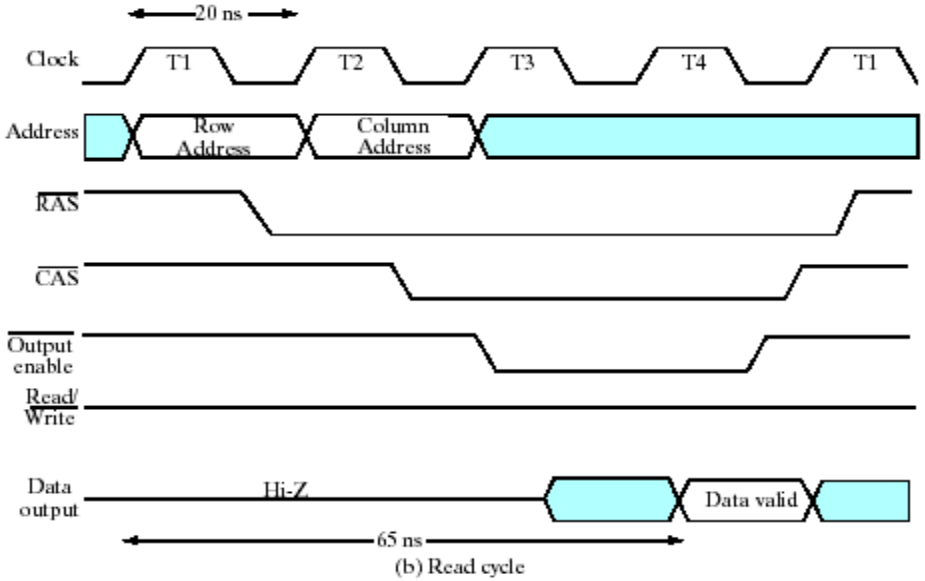
- The initiation of a refresh is controlled externally by using the RAS and CAS signals.
- During any referesh cycle, no DRAM reads or writes can occur.
- Types of refresh:
  - Distributed refresh (✓)
  - Burst refresh

$\overline{\text{RAS}}$: row addr strobe

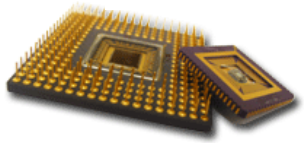
$\overline{\text{CAS}}$: column addr strobe



(a) Write cycle

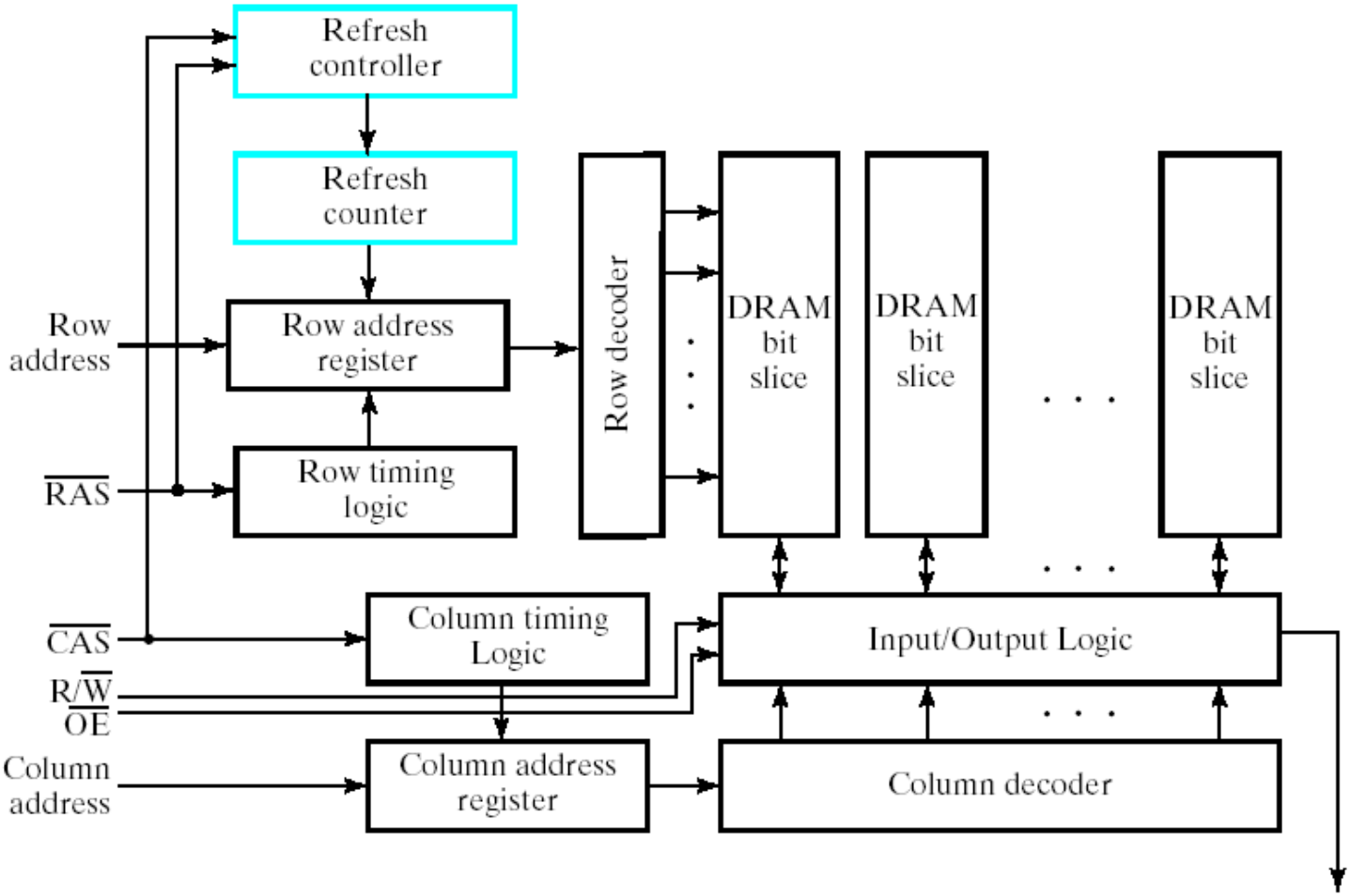


(b) Read cycle



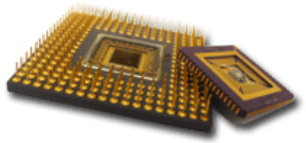
Refreshing

- Refreshing:



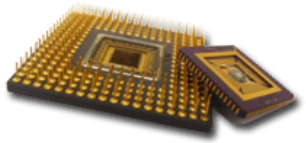
DRAM Types

Type	Abbreviation	Description
Fast Page Mode DRAM	FPM DRAM	Takes advantage of the fact that, when a row is accessed, all of the row values are available to be read out. By changing the column address, data from different addresses can be read out without reapplying the row address and waiting for the delay associated with reading out the row cells to pass if the row portion of the addresses match.
Extended Data Out-put DRAM	EDO DRAM	Extends the length of time that the DRAM holds the data values on its output, permitting the CPU to perform other tasks during the access since it knows the data will still be available.
Synchronous DRAM	SDRAM	Operates with a clock rather than being asyn-chronous. This permits a tighter interaction between memory and CPU, since the CPU knows exactly when the data will be available. SDRAM also takes advantage of the row value availability and divides memory into distinct banks, permitting overlapped accesses.
Double Data Rate Synchronous DRAM	DDR SDRAM	The same as SDRAM except that data output is provided on both the negative and the positive clock edges.
Rambus DRAM	RDRAM	A proprietary technology that provides very high memory access rates using a relatively narrow bus.
Error-Correcting Code	ECC	May be applied to most of the DRAM types above to correct single bit data errors and often detect double errors.



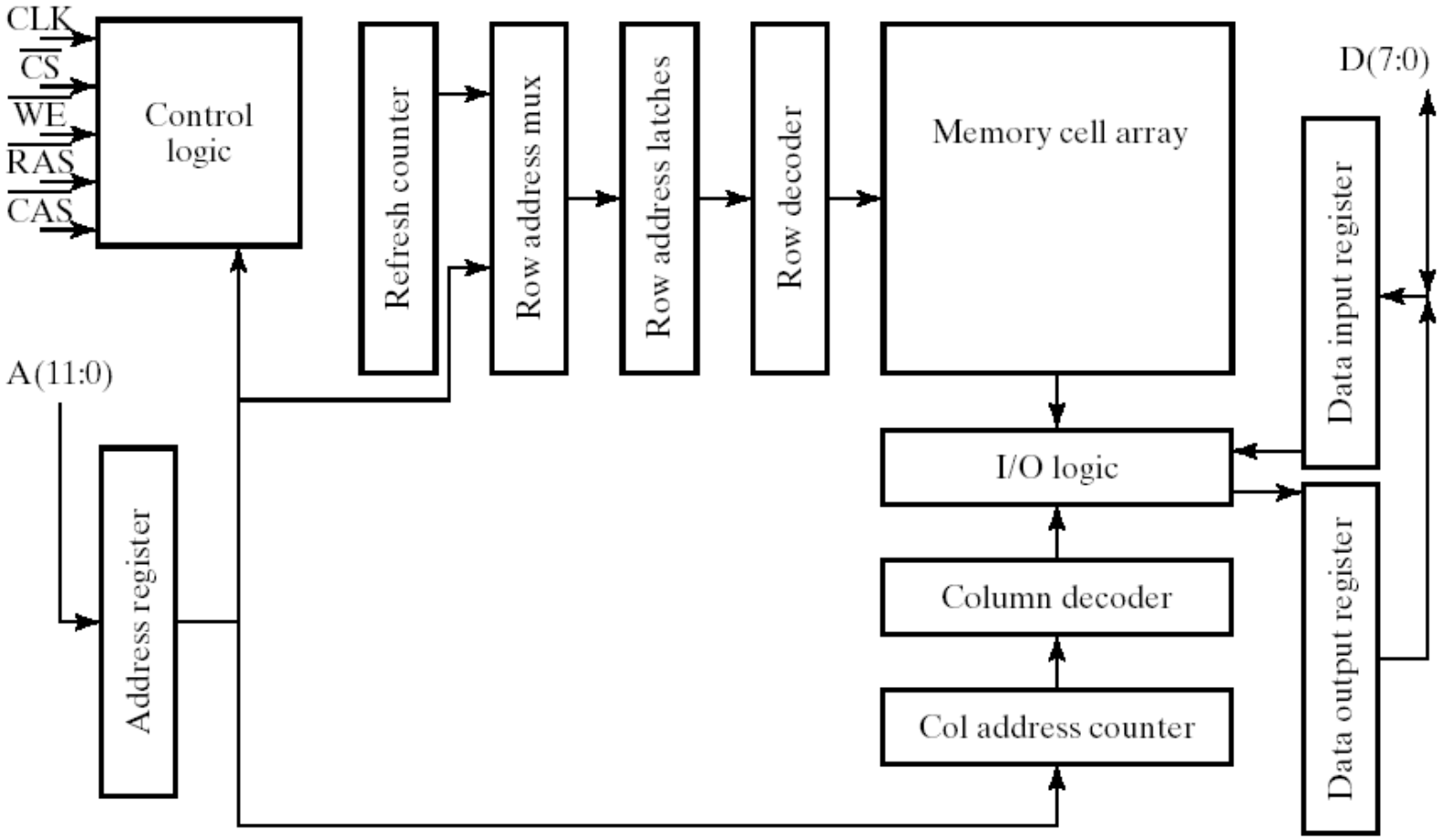
Synchronous DRAM (1/3)

- Transfers to and from the DRAM are synchronized with a clock
- Synchronous registers appear on:
  - Address input
  - Data input
  - Data output
- Column address counter
  - for addressing internal data to be transferred on each clock cycle
  - beginning with the column address counts up to column address + burst size – 1
- Before performing an actual read op from a specified column addr, the entire row specified by the applied row addr is read out internally and stored in the I/O logic.
- Example: Memory data path width: 1 word = 4 bytes
  Burst size: 8 words = 32 bytes
  Memory clock frequency: 5 ns
  Latency time (from application of row address until first word available): 4 clock cycles
  Read cycle time: (4 + 8) x 5 ns = 60 ns
  Memory Bandwidth: $32/(60 \times 10^{-9}) = 533$ Mbytes/sec



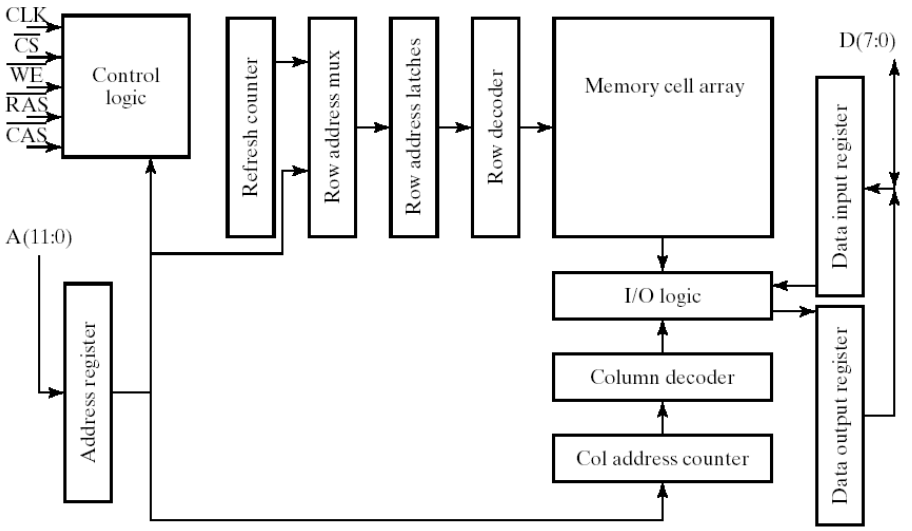
SDRAM (2/3)

- SDRAM: clocked transfer



SDRAM (3/3)

- E.g.: 16Mbyte SDRAM

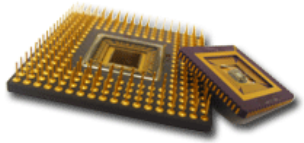


– 16MB = $2^{24} \times 2^3$ bits = 2^{27} bits = $2^{13} \times 2^{14}$ bits

= $2^{13} \times (2^{11} \times 8)$ bits

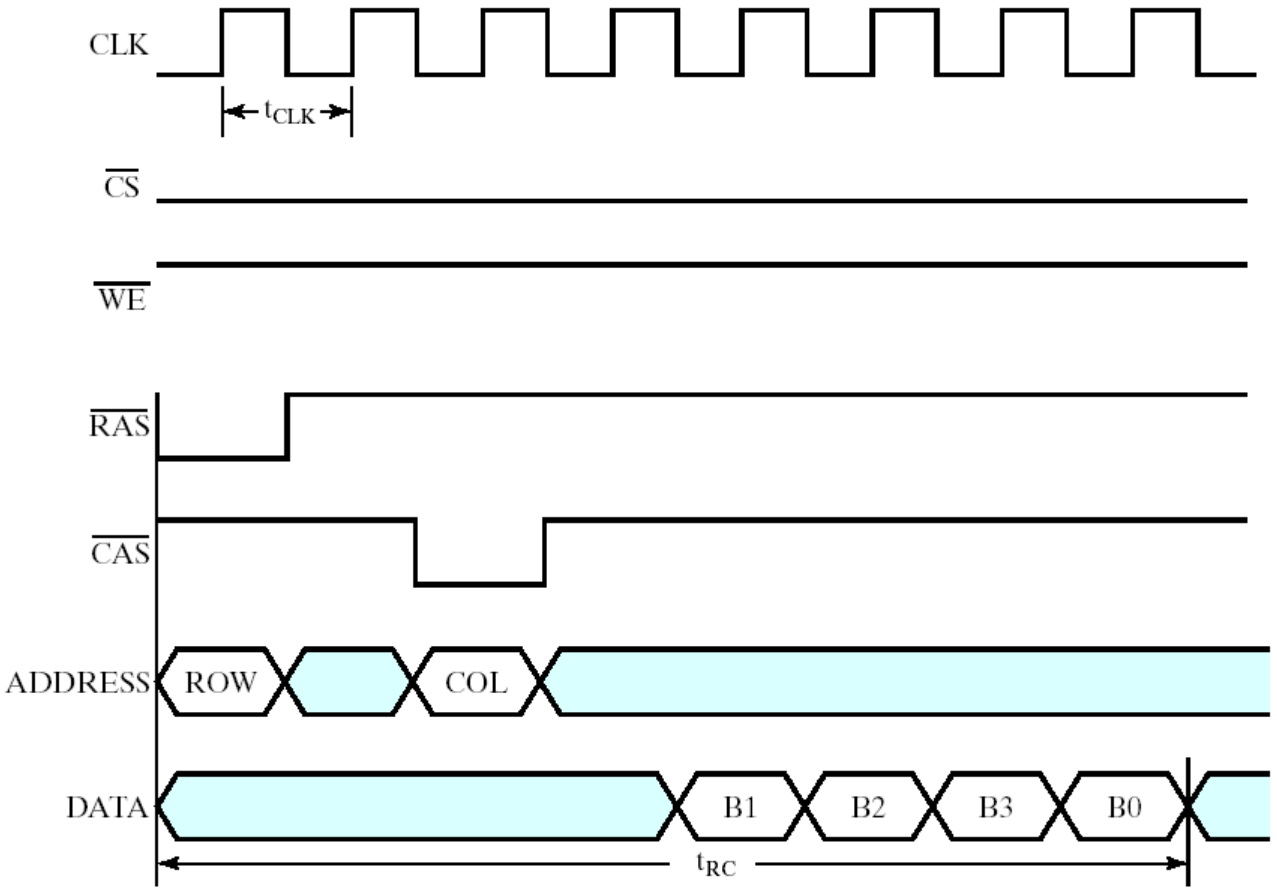
$\Rightarrow$ 13 row addr bits & 11 column addr bits

(16,384 ÷ 8 = 2,048)



Timing Waveform

- Timing of a burst read cycle: burst length = 4



Comparison of SDRAM and DRAM

- Comparing the byte rate for reading bytes from SDRAM to that of the basic DRAM:
  - Assumptions:
    - The read cycle time t_{RC} for the basic DRAM = 60 ns
    - The clock period t_{CLK} for the SDRAM = 7.5 ns
  - The byte rates:
    - For the basic DRAM: 1 byte per 60 ns $\Rightarrow$ 16.67 MB/sec
    - For the SDRAM w/ burst length = 4: 8 clock cycles to read 4 bytes

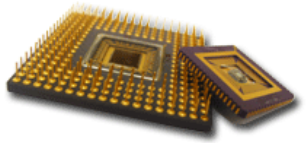
      $\Rightarrow$ read cycle time = $7.5 \times 8 = 60$ ns

      $\Rightarrow$ 4 bytes per 60 ns = 16.67MB/sec x4 = 66.67 MB/sec
    - For the SDRAM w/ burst length = 8:

      $\Rightarrow$ read cycle time = $60 + (8 - 4) \times 7.5 = 90$ ns $\Rightarrow$ 88.89 MB/sec
    - For the SDRAM w/ burst length = 2048 (the entire row):

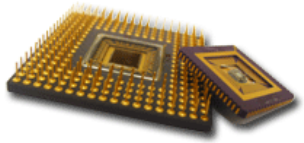
      $\Rightarrow$ read cycle time = $60 + (2048 - 4) \times 7.5 = 15{,}390$ ns

      $\Rightarrow$ 133.07 MB/sec $\rightarrow$ one byte per 7.5 ns



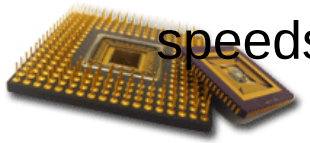
Double Data Rate Synchronous DRAM

- Transfers data on both edges of the clock
- Provide a transfer rate of 2 data words per clock cycle
- Example: Same as for synchronous DRAM
  - Read cycle time = 60 ns
  - Memory Bandwidth: $(2 \times 32)/(60 \times 10^{-9}) = 1066$ Mbytes/sec



RAMBUS DRAM (RDRAM)

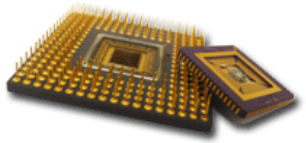
- Uses a packet-based bus for interaction between the RDRAM ICs and the memory bus to the processor
- The bus consists of:
  - A 3-bit row address bus
  - A 5-bit column address bus
  - A 16 or 18-bit (for error correction) data bus
- The bus is synchronous and transfers on both edges of the clock
- Packets are 4-clock cycles long giving 8 transfers per packet representing:
  - A 12-bit row address packet
  - A 20-bit column address packet
  - A 128 or 144-bit data packet
- Multiple memory banks are used to permit concurrent memory accesses with different row addresses
- The electronic design is sophisticated permitting very fast clock speeds



Arrays of DRAM ICs

- Design principles of DRAM arrays:
  - similar to that of SRAM arrays
  - differences: different requirements for the control and addressing of DRAM arrays
- DRAM controller:
  - a complex sync seq ckt w/ the external CPU clock
  - Functions performed:
    1. controlling separation of the addr into a row addr and a column addr and providing these addrs at the required times
    2. providing the RAS and CAS signals at the required times for read, write, and refresh ops
    3. performing refresh ops at the necessary intervals
    4. providing status signals to the rest of the system

       (e.g.: memory is busy performing refresh)



Summary

- Memory Definition and Operation
- SRAM Cell and ICs
- DRAM Cell and ICs
- DRAM Types