Memory and programmable logic: RAM

LECTURE 4

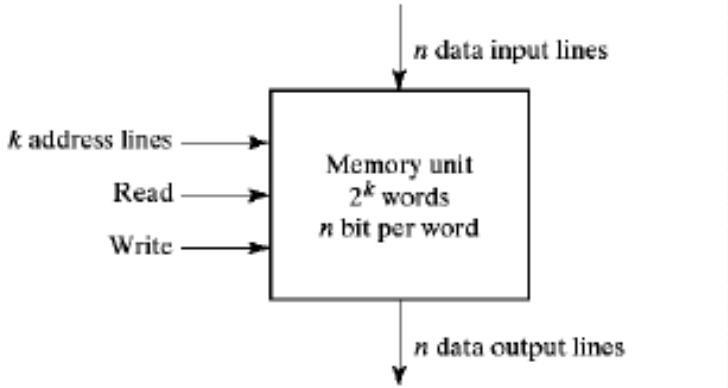
Dronacharya Group of Institutions

Outline

- **Introduction**
- Random-Access Memory
- Memory Decoding
- Error Detection and Correction
- Read-Only Memory
- Programmable Devices
- Sequential Programmable Devices

Mass Memory Elements

- Memory is a collection of binary cells together with associated circuits needed to transfer information to or from any desired location.



- Two primary categories of memory:
  - Random access memory (RAM)
  - Read only memory (ROM)

Programmable Logic Device

- The binary information within the device can be specified in some fashion and then embedded within the hardware.
- Most of them are programmed by breaking the fuses of unnecessary connections.
- Four kinds of PLD are introduced:
- Read-only memory (ROM)
- Programmable logic array (PLA)
- Programmable array logic (PAL)
- Field-programmable gate array (FPGA)

Outline

- Introduction
- **Random Access memory**
- Memory Decoding
- Error Detection and Correction
- Read-Only Memory
- Combinational Programmable Devices
- Sequential Programmable Devices

Random Access Memory

- A **word is the basic unit that** moves in and out of memory
- The length of a word is often multiples of a byte (=8 bits)
- Memory units are specified by its **number of words** and the **number of bits in** each word
- Ex: 1024(words) x 16(bits)
- Each word is assigned a particular **address, starting** from 0 up to 2k . 1
- (k = number of address lines)

Memory address		
Binary	decimal	Memory contest
0000000000	0	1011010101011101
0000000001	1	1010101110001001
0000000010	2	0000110101000110
⋮	⋮	⋮
1111111101	1021	1001110100010100
1111111110	1022	0000110100011110
1111111111	1023	1101111000100101

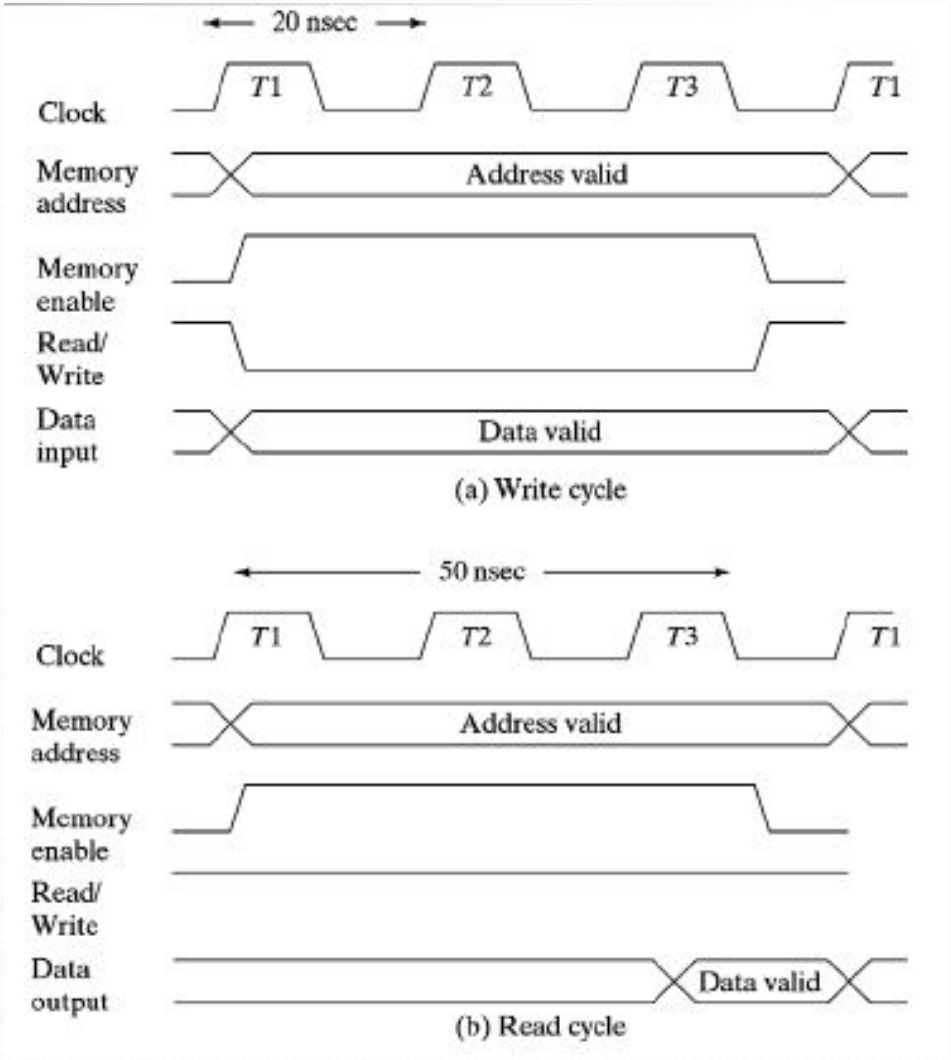
Content of a 1024 × 16 Memory

Write and Read Operations

- Write to RAM
- Apply the binary address of the desired word to the **address lines.**
- Apply the data bits that must be stored in memory to the **data input lines.**
- Activate the **write control.**
- Read from RAM.
- Apply the binary address of the desired word to the **address lines.**
- Activate the **read control.**

Timing Waveforms

- CPU clock = 50 MHz cycle time = 20 ns
- Memory access time = 50 ns.
- The time required to complete a read or write operation.
- The control signals must stay active for at least 50 ns 3 CPU cycles are required.



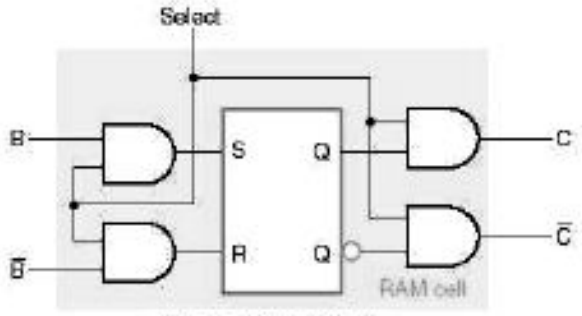
Types of Memories

- Access mode:
- Random access: any locations can be accessed in any order
- Sequential access: accessed only when the requested word has been reached (ex: hard disk)
- Operating mode:
- Static RAM (SRAM)
- Dynamic RAM (DRAM)
- Volatile mode:
- Volatile memory: lose stored information when power is turned off (ex: RAM)
- Non-volatile memory: retain its storage after removal of power (ex: flash, ROM, hard-disk, .)

SRAM vs. DRAM

Static RAM:

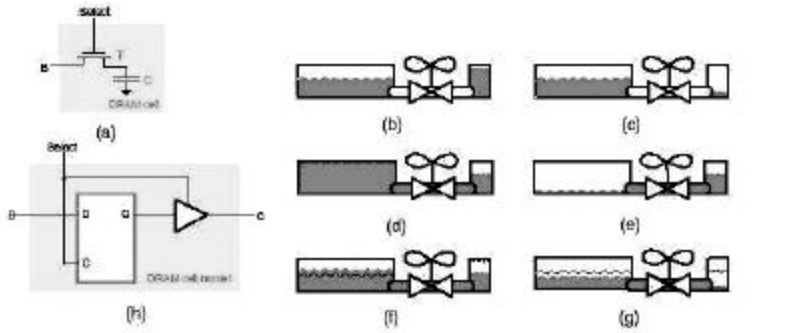
1. Use internal latch to store the binary information.
2. Stored information remains valid as long as power is on.
3. Shorter read and write cycles
4. Larger cell area and power consumption.



Static RAM Cell

Dynamic RAM:

1. Use a capacitor to store the binary information.
2. Need periodically refreshing to hold the stored info.
3. Longer read and write cycles
4. Smaller cell area and power consumption.

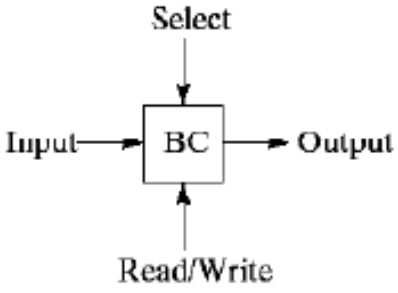


Dynamic RAM Cell, Hydraulic Analogy of Cell Operation, and Cell Model

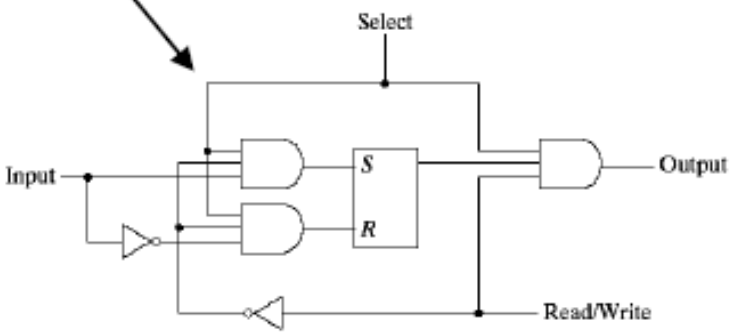
Outline

- Introduction
- Random-Access Memory
- **Memory Decoding**
- Error Detection and Correction
- Read-Only Memory
- Combinational Programmable Devices
- Sequential Programmable Devices

Memory Construction



A SRAM Cell



(a) Logic diagram

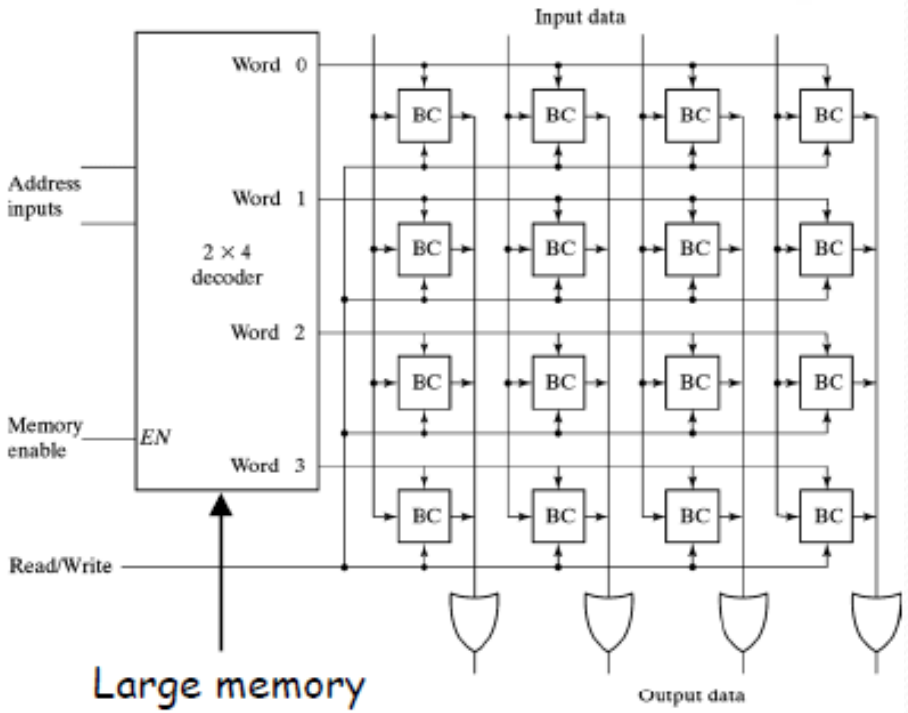


Diagram of a 4 × 4 RAM

Large memory will require large decoder

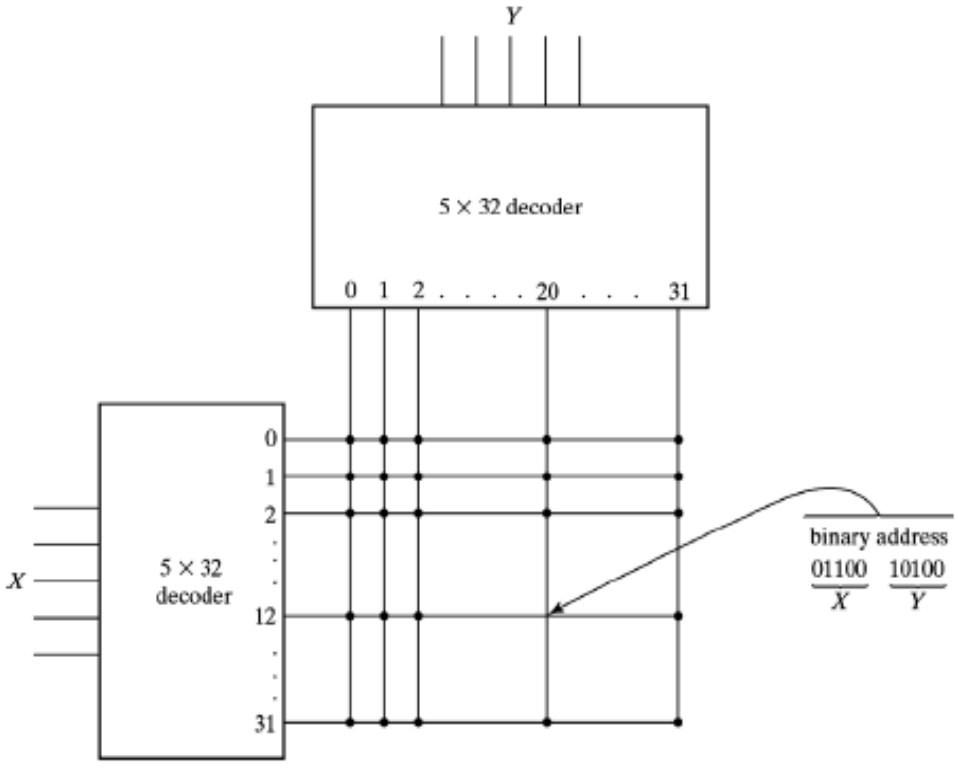
Coincident Decoding

- Address decoders are often divided into two parts.
- A two-dimensional scheme.
- The total number of gates in decoders can be reduced.
- Can arrange the memory cells to a square shape.
- EX: 10-bit address

404 = 0110010100

X = 01100 (first five)

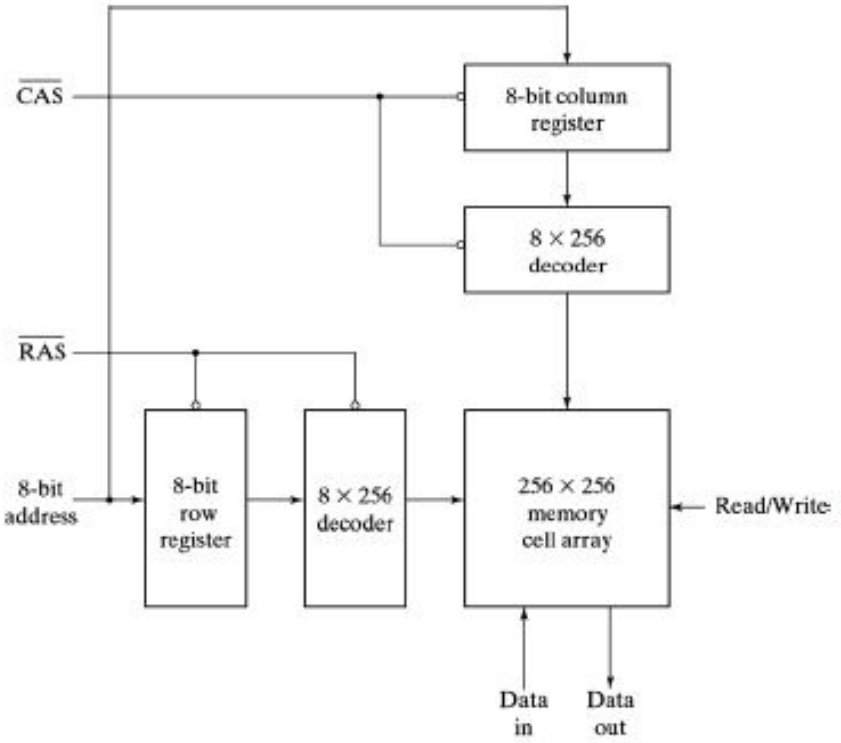
Y = 10100 (last five)



Two-Dimensional Decoding Structure for a 1K-Word Memory

Address Multiplexing

- Memory address lines often occupy too much I/O pads
- 64K = 16 lines
- 256M = 28 lines
- Share the address lines of X and Y domains
- Reduce the number of lines to a half
- An extra register is required for both domain to store the address
- Two steps to send address
- RAS=0: send row address
- CAS=0: send column address



Address Multiplexing for a 64K DRAM

Outline

- Introduction
- Random-Access Memory
- Memory Decoding
- Error Detection and Correction
- Read-Only Memory
- Combinational Programmable Devices
- Sequential Programmable Devices

Error Detection & Correction

- Memory arrays are often very huge.
- May cause occasional errors in data access.
- Reliability of memory can be improved by employing error-detecting and correcting codes.
- Error-detecting code: only check for the **existence** of errors
- Most common scheme is the parity bit.
- Error-correcting code: check the existence and **locations of errors.**
- Use multiple parity check bits to generate a **syndrome** that can indicate the erroneous bits.
- Complement the erroneous bits can correct the errors.

Hamming Code

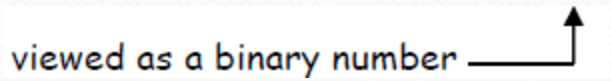
- k parity bits are added to an n-bit data word
- The positions numbered as a **power of 2 are** reserved for the parity bits
- Ex: original data is 11000100 (8-bit)

⇒ Bit position: **1 2 3 4 5 6 7 8 9 10 11 12**

- P1 P2 1 P4 1 0 0 P8 0 1 0 0
- P1 = XOR of bits (3,5,7,9,11) = 0
- P2 = XOR of bits (3,6,7,10,11) =0
- P4 = XOR of bits (5,6,7,12) = 1
- P8 = XOR of bits (9,10,11,12) = 1
- The composite word is 001110010100 (12-bit)

Hamming Code

- When the 12 bits are read from memory, the parity is checked over the **same combination of bits including the parity bit**
- C1 = XOR of bits (1,3,5,7,9,11)
- C2 = XOR of bits (2,3,6,7,10,11)
- C4 = XOR of bits (4,5,6,7,12)
- C8 = XOR of bits (8,9,10,11,12)
- (001110010100) " C = C8C4C2C1 = 0000 : no error
- (101110010100) " C = C8C4C2C1 = 0001 : bit 1 error
- (001100010100) " C = C8C4C2C1 = 0101 : bit 5 error



General Rules of Hamming Code

- The number of parity bits:
- The syndrome C with k bits can represent 2k . 1 error locations
- (0 indicates no error)
- 2k . 1 ≥ n + k " 2k . 1 . k ≥ n
- The members of each parity bit:

Number of Check Bits, k	Range of Data Bits, n
3	2-4
4	5-11
5	12-26
6	27-57
7	58-120

- C1(P1): have a .1. in **bit 1 of their location numbers**

1(0001), 3(0011), 5(0101), 7(0111), 9(1001), .

- C2(P2): have a .1. in **bit 2 of their location numbers**

2(0010), 3(0011), 6(0110), 7(0111), 10(1010), .

- C: with parity bit; P: without parity bit itself

Extension of Hamming Code

- Original Hamming code can detect and correct only a **single error.**
- Multiple errors are not detected.
- Add an extra bit as the parity of total coded word.
- Ex: 001110010100P13 (P13=XOR of bits 1 to 12).
- Still single-error correction but double-error detection.
- Four cases can occur:
- If C=0 and P=0, no error occurred
- If C≠0 and P=1, single error occurred (can be fixed)
- If C≠0 and P=0, double error occurred (cannot be fixed)
- If C=0 and P=1, an error occurred in the P13 bit