Digital Logic Systems – Registers, Counters and Memory Units- IV

Assist. Prof. Özge ÖZTİMUR KARADAĞ

ALKÜ

Memory Unit

- Memory Unit: a device to which binary information is transferred for storage and from which information is retrieved when needed for processing. A memory unit is a collection of cells capable of storing a large quantity of binary information.
- Two types of memory operations:
    - Read: The process of transferring the stored information out of memory is referred to as a memory read operation.
    - Write: The process of storing new information into memory is referred to as a memory *write* operation.

- Two types of memories in digital systems:
    - RAM (random Access memory) performs both read and write
    - ROM (read only memory) performs only read

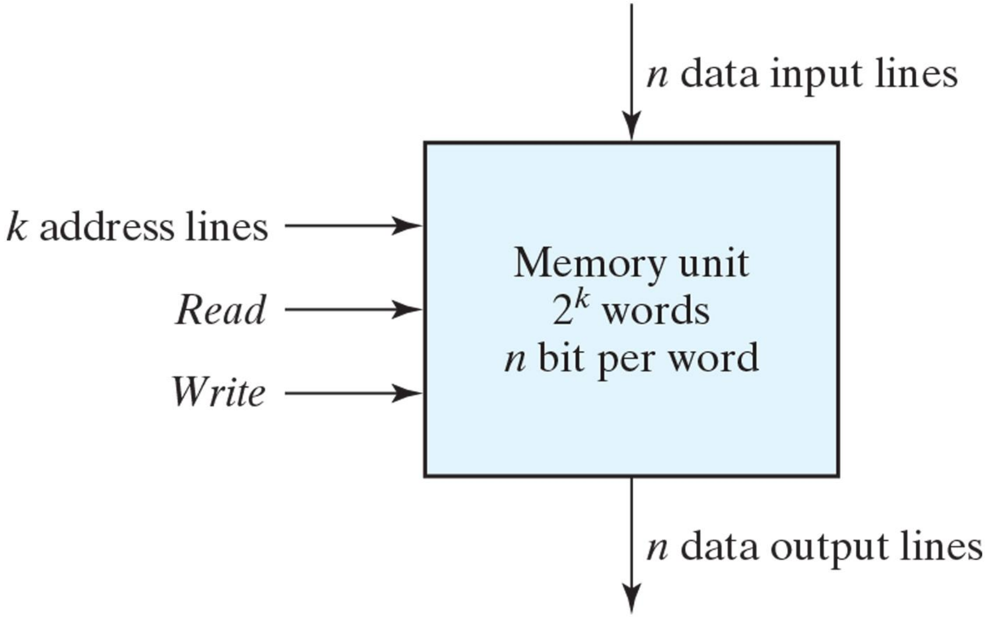
Random Access Memory (RAM)

- A memory unit is a collection of storage cells together with associated circuits needed to transfer information into and out of a device.
- The architecture of memory is such that information can be selectively retrieved from any of its internal locations. The time it takes to transfer information to or from any desired random location is always the same-hence the name *random Access memory,* abbreviated RAM.
- In contrast, the time required to retrieve information that is stored on magnetic tape depends on the location of the data.

RAM

- A memory unit stores binary information in groups of bits called *words.*
- A word in memory is an entity of bits that move in and out of storage as a unit.
- A memory word is a group of 1's and 0's and may represent a number, an instruction, one or more alphanumeric characters or any other binary-coded information.
- A group of 8 bits is called a *byte.* Most computer memories use words that are multiples of 8 bits in length. 16 bit word contains two bytes and a 32-bit word is made up of four bytes.
- The capacity of a memory unit is usually stated as the total number of bytes that the unit can store.
- Communication between memory and its environment is achieved through:
  - data input and output lines,
  - address selection lines,
  - control lines that specify the direction of transfer

- Block diagram of memory unit:



RAM

- Memory capacity and addressing
  - 1024 words → 10 bits for adressing
  - K (kilo) → 2^{10}
  - M (Mega) → 2^{20}
  - G (Giga) → 2^{30}

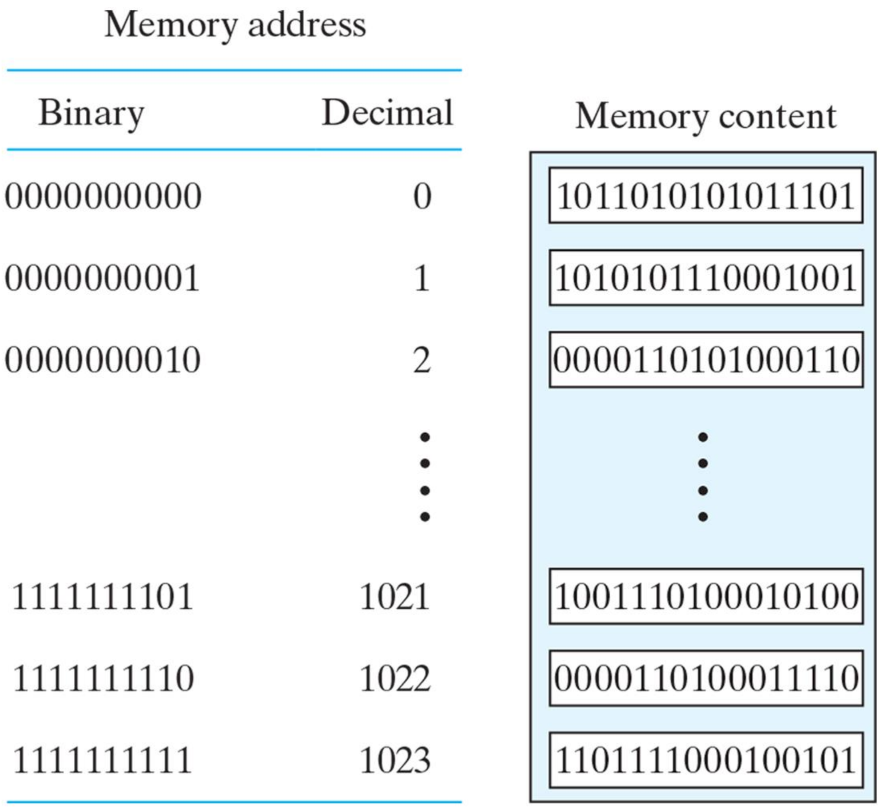
- Ex:
  - 64K = 2^{16}
  - 2M = 2^{21}
  - 4G=2^{32}

RAM

- Ex: What is the capacity of a memory which has 1K words and each word is 16 bits long.
  - 1K=1024 =2^{10}
  - 16 bit = 2 byte
  - The capacity of the memory unit is 2Kbytes.

RAM

- Contents of a 1024x16 memory:



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

RAM

- Ex: A memory of 64Kx10
  - Each word 10bits
  - 16 bits for adressing → 64K = 2^{16}

  - $2^k = m$
    - k: adres bits
    - m: number of words in memory

RAM

- Read and Write Operations
  - Write: A transfer-in operation
    - The steps that must be taken for the purpose of transferring a new word to be stored into memory
      - 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* input.

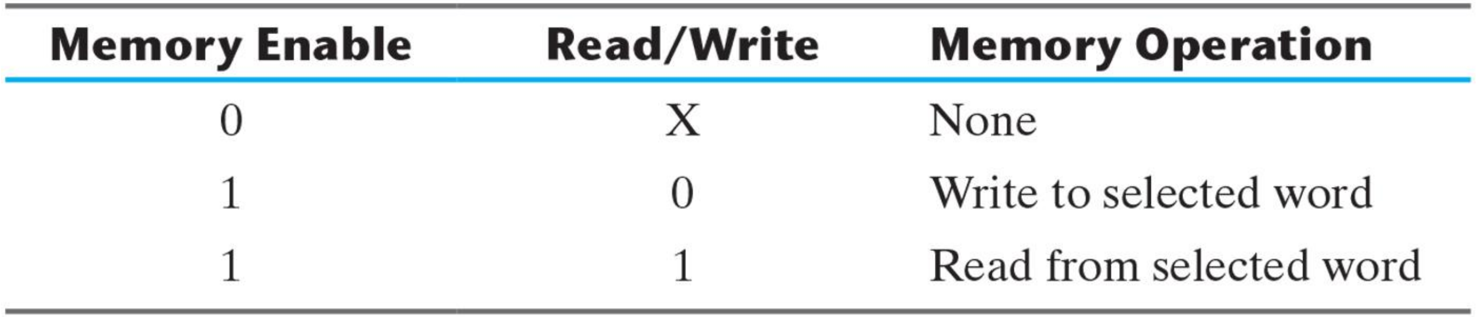
    - The memory unit will then take the bits from the input data lines and store them in the word specified by the address lines.

RAM

- Read and Write Operations
  - READ: trasfer-out operation
    - The steps that must be taken for the purpose of transferring a stored word out of memory:
      - Apply the binary address of the desired word to the address lines.
      - Activate the *read* input.
    - The memory unit will then take the bits from the word that has been selected by the address and apply them to the output data lines.

RAM

- Control inputs to memory chip:



Memory Enable	Read/Write	Memory Operation
0	X	None
1	0	Write to selected word
1	1	Read from selected word

RAM

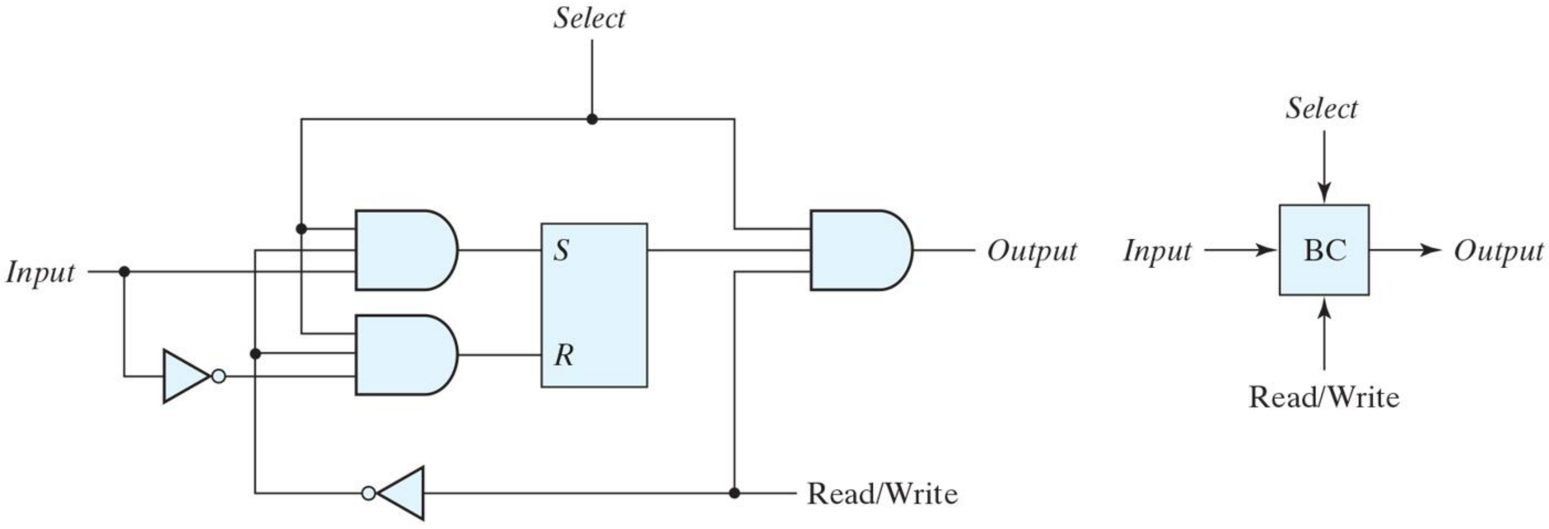
- Access time: Time required to select a word to be read or written.
- In a random-access memory, the access time is always the same regardless of the particular location of the word.
- In a sequential -access memory the time it takes to access a word depends on the position of the word with respect to the position of the read head: therefore. the Access time is variable .
- Integrated circuit RAM units are available in two operating modes:
    - Static (SRAM): Consits of internal flip-flops which stores binary information. The stored information remains valid as long as power is applied to the unit.
    - Dynamic (DRAM): stores the binary information in the form of electric charges on capacitors provided inside the chip by MOS transistors. The stored charge on the capacitors tends to discharge with time and the capacitors must be periodically recharged by *refreshing* the dynamic memory. Refreshing is done by cycling through the words every few milliseconds to restore the decaying charge.
- DRAM offers reduced power consumption and larger storage capacity in a single memory chip.
- SRAM is easier to use and has shorter read and write cycles.

RAM

- Memory units that lose stored information when power is turned off are said to be *volatile*
  - SRAM and DRAM are volatile, since the binary cells need external power to maintain the stored information
- Magnetic disks are nonvolatile since it retains its stored information after the removal of power.

Memory Decoding

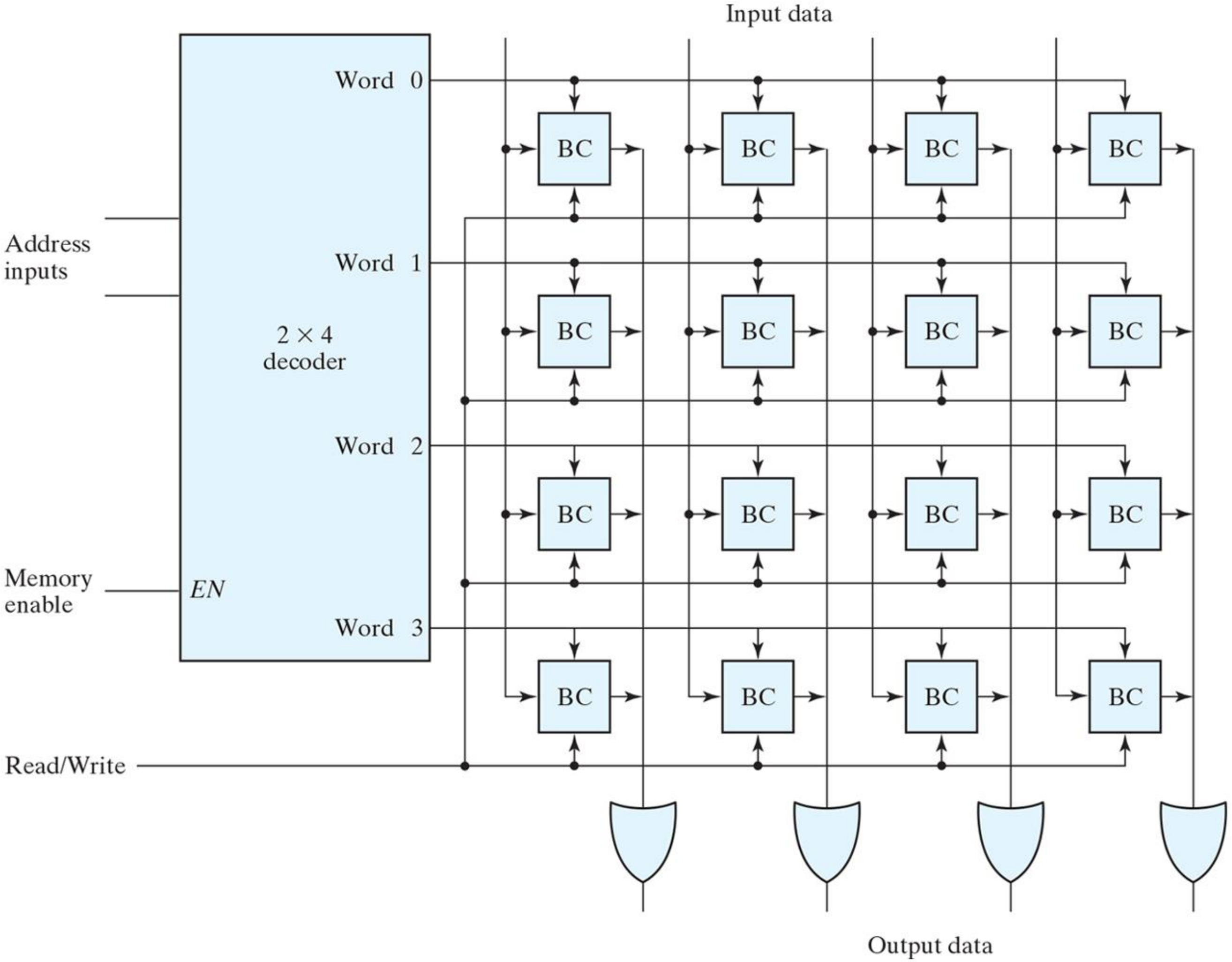
- Internal Construction: The internal construction of a RAM of *m* words and *n* bits per word consists of *m* X n binary storage cells and associated decoding circuits for selecting individual words.
- The binary srorage cell is the basic building block of a memory unit:
  - Read/write=1 → Read
  - Read/write=0→ Write



(a) Logic diagram

(b) Block diagram

Diagram of a 4x4 RAM

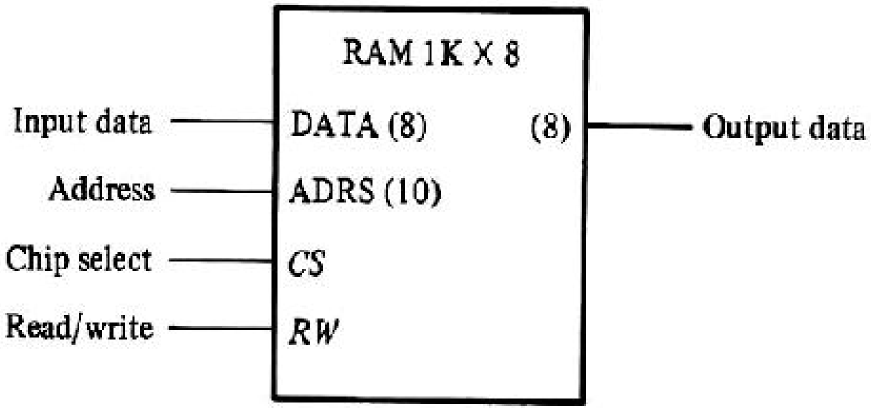


Memory Decoding

- A memory with 2^k words of *n* bits per word requires *k* address lines that go into a *k* X 2^k decoder. Each one of the decoder outputs selects one word of n bits for reading or writing .

Set of RAMs

- Set of RAM integrated circuits: A set of integrated circuits of RAMs can be connected to obtain a larger memory.
- Capacity of the memory:
    - Increasing the number of words → increase in the length of adress
    - Increase in the number of bits in words→ increase in the input/output data lines
    - Block diagram of 1Kx8 RAM

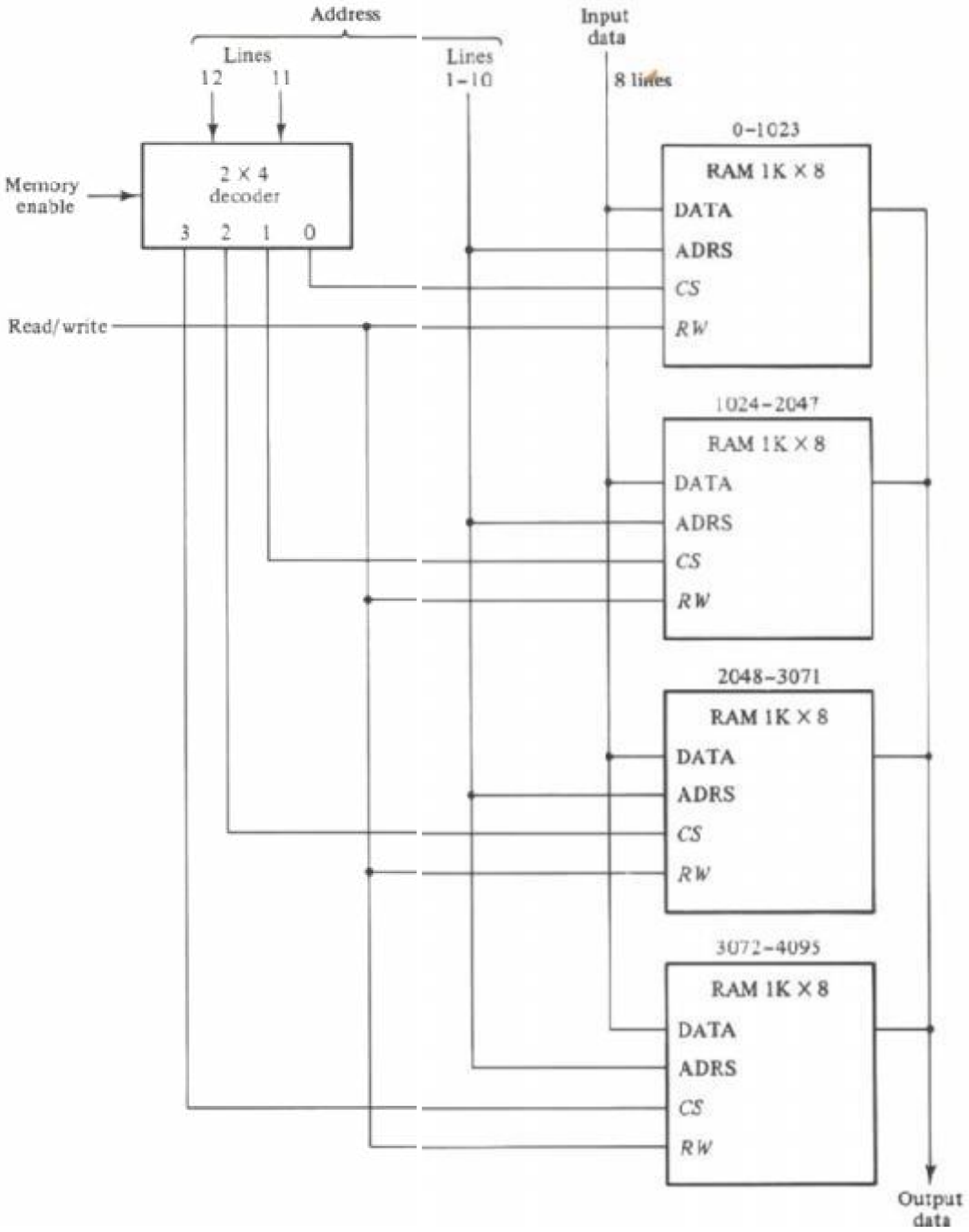


Set of RAMs

- Increase the number of words in the memory using a set of RAMs
  - Addition of each bit to adress → doubles the number of words
  - Ex: 2 RAMs doubles the number of words, and adds one more bit to the address bits.

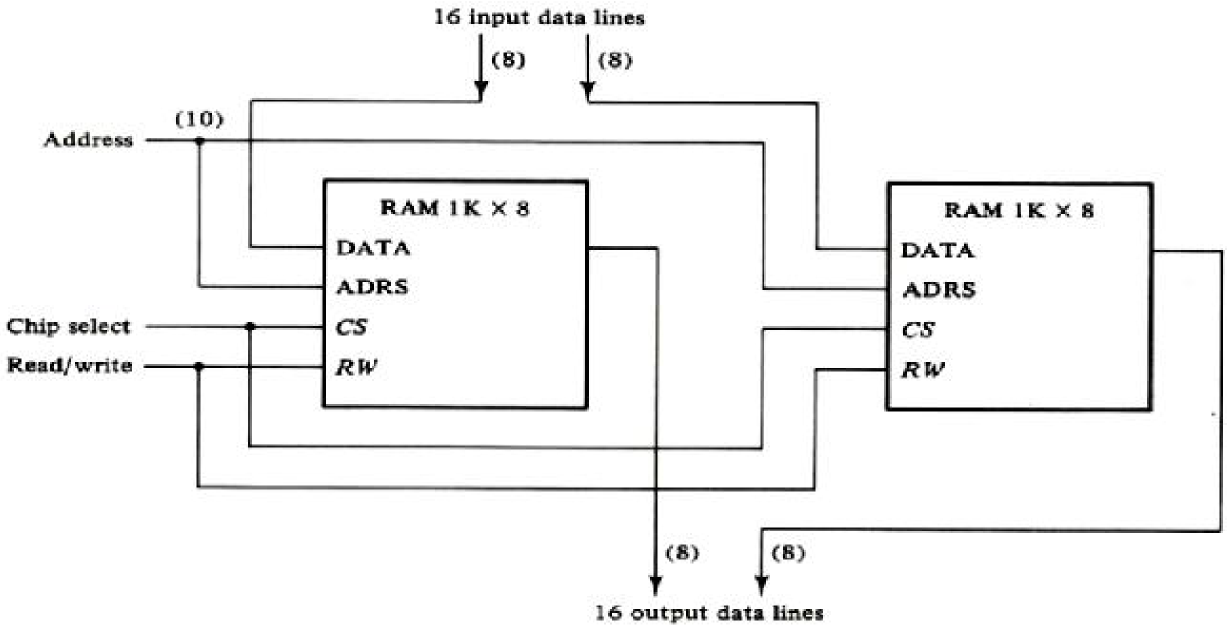
Set of RAMs

- Using 4 1Kx8 RAMs obtain a 4Kx8 RAM:



Set of RAMS

- Two RAMs with same number of words can be combined to obtain a RAM with a longer word.
    - Ex: two RAMs with size 1Kx8 are used to obtain a memory of 1Kx16:



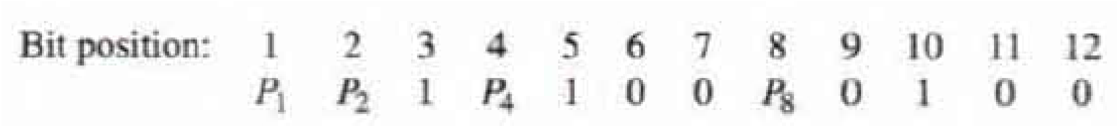
Error Detection and Correction

- Errors can arise while storing retrieving binary data.

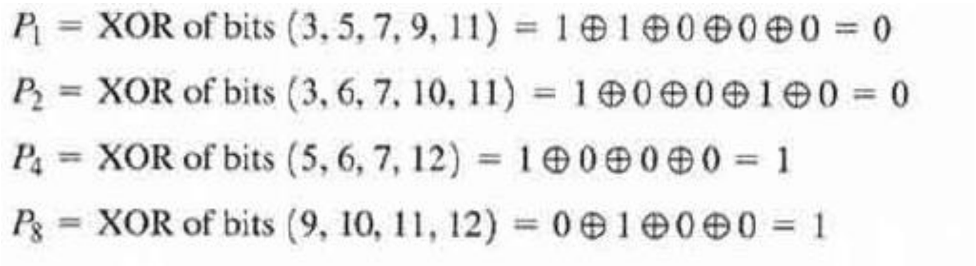
- The reliability of a memory unit may be improved by employing error-detecting and error-correcting codes.

Error Detection and Correction

- Hamming Code
  - *k* parity bits are added to an n-bit data word forming a new word of n+*k* bits.
  - The bit positions are numbered in sequence from 1 to *n+k,*
  - Those positions numbered as a power of 2 are reserved for the parity bits
  - Remaining bits are data bits:
  - ex:11000100 8-bit data word. 4 parity bit are added :



Bit position:	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	1	P_4	1	0	0	P_8	0	1	0	0



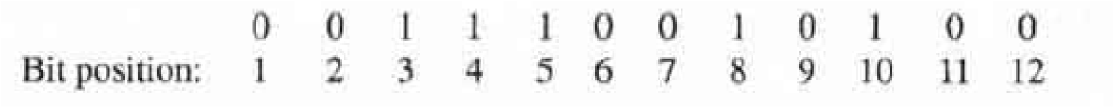
$$P_1 = \text{XOR of bits } (3, 5, 7, 9, 11) = 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 = 0$$

$$P_2 = \text{XOR of bits } (3, 6, 7, 10, 11) = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$P_4 = \text{XOR of bits } (5, 6, 7, 12) = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

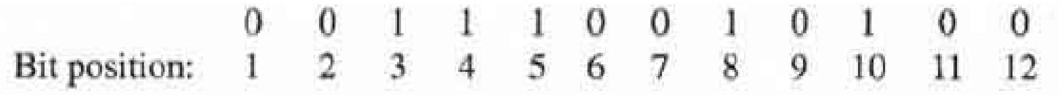
$$P_8 = \text{XOR of bits } (9, 10, 11, 12) = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

  - each parity bit is set so that the total number of 1's in the checked positions including the parity bit, is always even.



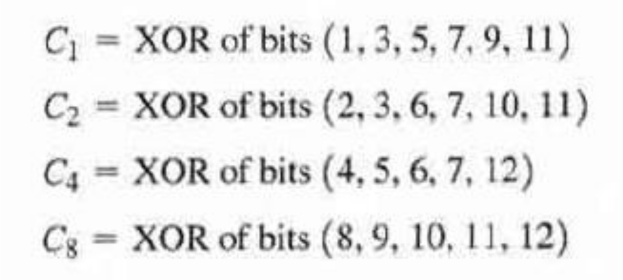
	0	0	1	1	1	0	0	1	0	1	0	0
Bit position:	1	2	3	4	5	6	7	8	9	10	11	12

Error Detection and Correction



	0	0	1	1	1	0	0	1	0	1	0	0
Bit position:	1	2	3	4	5	6	7	8	9	10	11	12

- When the 12 bits are read from memory. they are checked again for errors. The parity is checked over the same combination of bits including the parity bit. The 4 check bits are evaluated as follows:



$$C_1 = \text{XOR of bits } (1, 3, 5, 7, 9, 11)$$

$$C_2 = \text{XOR of bits } (2, 3, 6, 7, 10, 11)$$

$$C_4 = \text{XOR of bits } (4, 5, 6, 7, 12)$$

$$C_8 = \text{XOR of bits } (8, 9, 10, 11, 12)$$

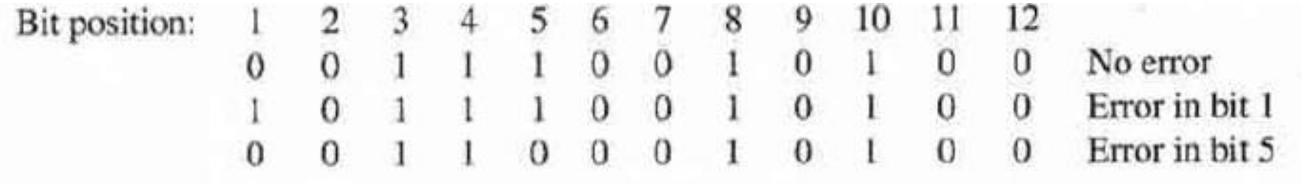
- A 0 check bit designates even parity over the checked bits and a 1 designates odd parity

- Since the bits were stored with even parity, the result $C=C_8C_4C_2C_1=0000$ → no error
- $C \neq 0$ → the 4 -bit binary number formed by the check bits gives the position of the erroneous bit.

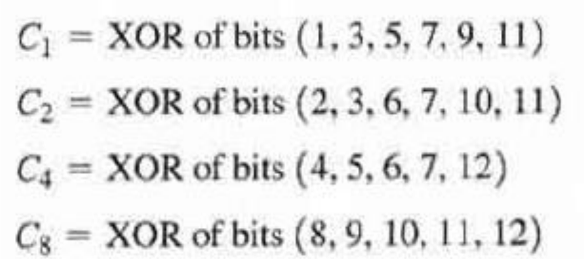
Error Detection and Correction

11000100

- Hamming Code



Bit position:	1	2	3	4	5	6	7	8	9	10	11	12	
	0	0	1	1	1	0	0	1	0	1	0	0	No error
	1	0	1	1	1	0	0	1	0	1	0	0	Error in bit 1
	0	0	1	1	0	0	0	1	0	1	0	0	Error in bit 5

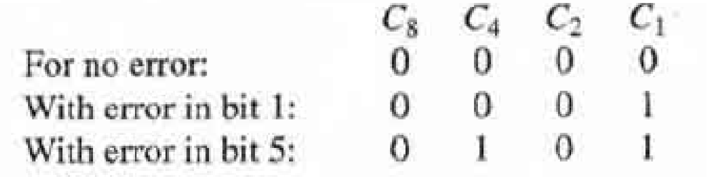


$$C_1 = \text{XOR of bits } (1, 3, 5, 7, 9, 11)$$

$$C_2 = \text{XOR of bits } (2, 3, 6, 7, 10, 11)$$

$$C_4 = \text{XOR of bits } (4, 5, 6, 7, 12)$$

$$C_8 = \text{XOR of bits } (8, 9, 10, 11, 12)$$



	C_8	C_4	C_2	C_1
For no error:	0	0	0	0
With error in bit 1:	0	0	0	1
With error in bit 5:	0	1	0	1

- The error can be corrected by complementing the corresponding bit. Note that an error can occur in the data word or in one of the parity bits.

Error Detection and Correction

- Single-Error Correction, Double-Error Detection:
- The Hamming code can detect and correct only a single error.
- By adding an other parity bit to the coded word, the Hamming code can be used to correct a single error and detect double errors.
- P_{13} is evaluated from the exclusive-OR of the other 12 bits.
- Ex:001110010100P_{13}
  - the check bits are evaluated as is the parity *P* over the entire 13 bits.
  - P=0 -> no error.
  - P=1 –> parity over 13 bits are incorrect.
  - One of the following arise:
    - C=0 and P=0 → no error occurred
    - C≠0 and P=1 → a single error occurred that can be corrected.
    - C≠0 and P=0 →a double error occurred that is detected but that cannot be corrected
    - C=0 and P=1 → error in bit P_{13}
    - This scheme may detect more than two errors, but is not guaranteed to detect all such errors