Digital Storage

Memory

Memory

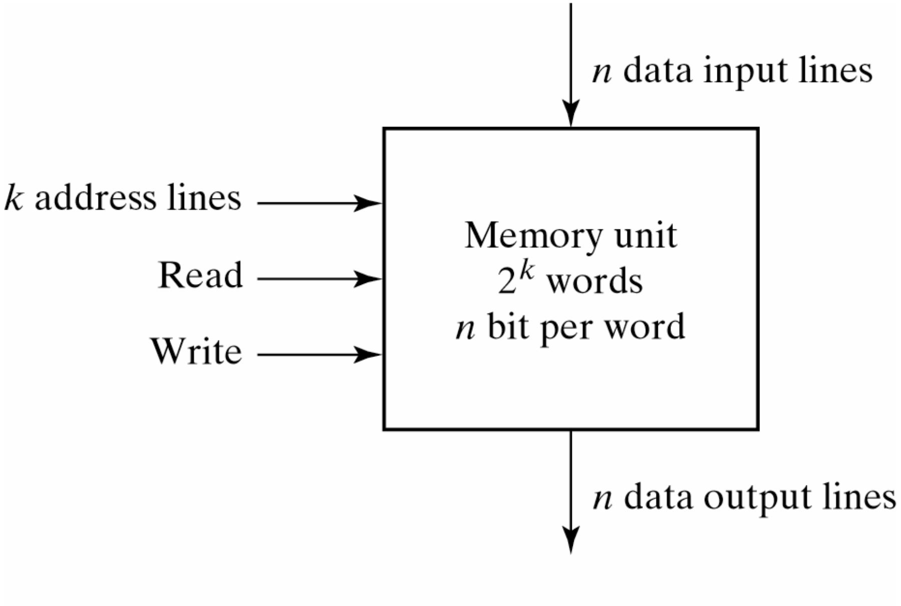
- Memory devices are used to store binary information.
- The two types used in digital systems are random access memory (RAM) devices and read only memory (ROM) devices.
- Memory storage must be reliable. To avoid errors parity check codes are used.

D-RAM

EP-ROM

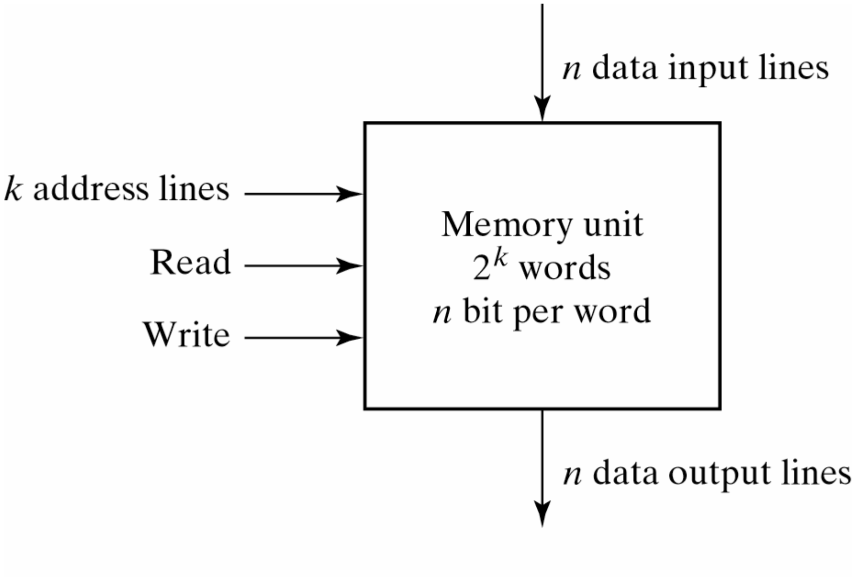
Random Access Memory

- A RAM memory unit consists of a collection of binary storage cells and the circuitry needed to transfer data in and out of the device.
- Information is stored in words (groups of bits), each word usually consists of 8 bits (a byte).

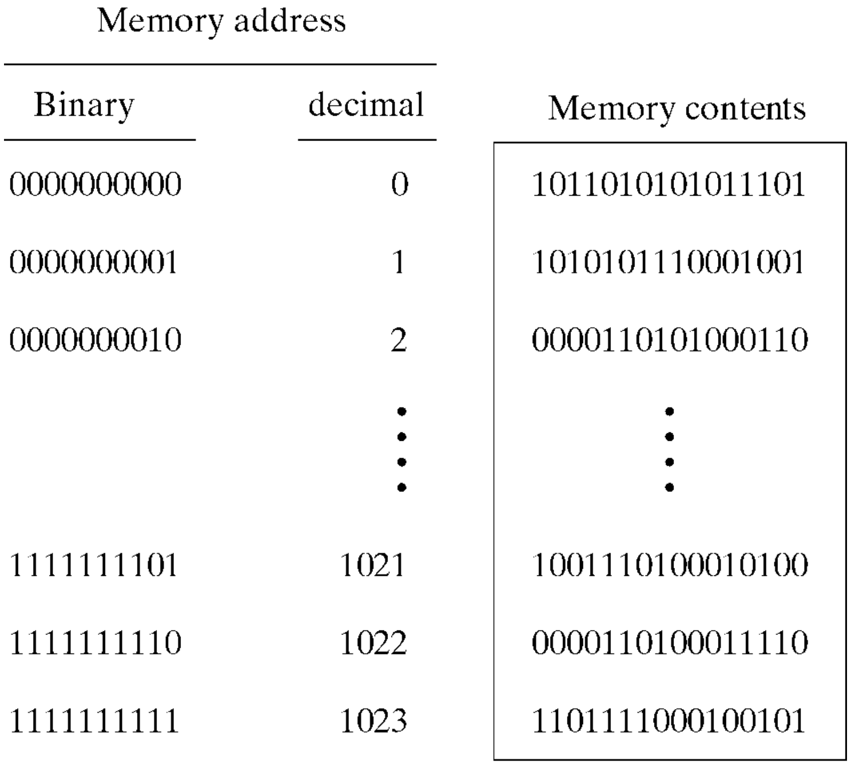


Random Access Memory

- The write input allows a word to be inputted and then stored at a location in memory.
- The read input allows a word to be taken from a location in memory and then outputted.
- Each location in memory is specified by the address input.



Random Access Memory



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

- A memory device is specified by the number of words that may be stored and the number of bits in each word.
- Here the memory device contains 1024 words, with each word containing 16 bits.
- Each location in memory is assigned a 10 bit address.

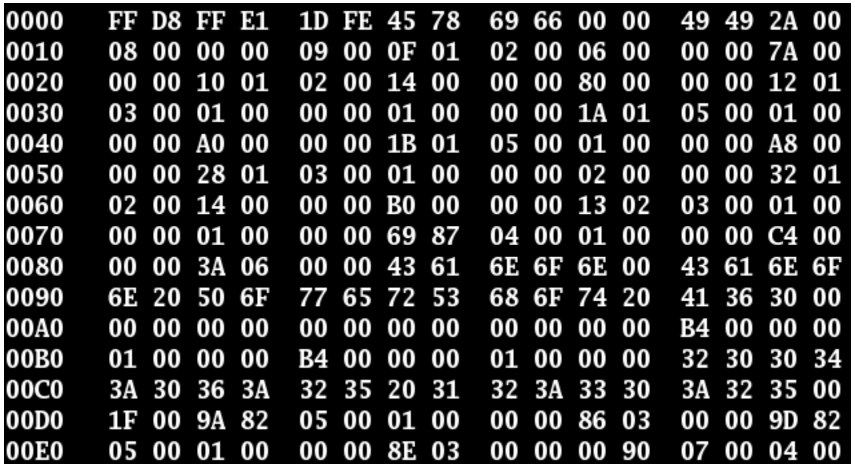
Random Access Memory

- For large memory sizes the convention is to refer to kilo, mega or giga bytes.

  1kB = 2^{10} bytes

  1MB = 2^{20} bytes

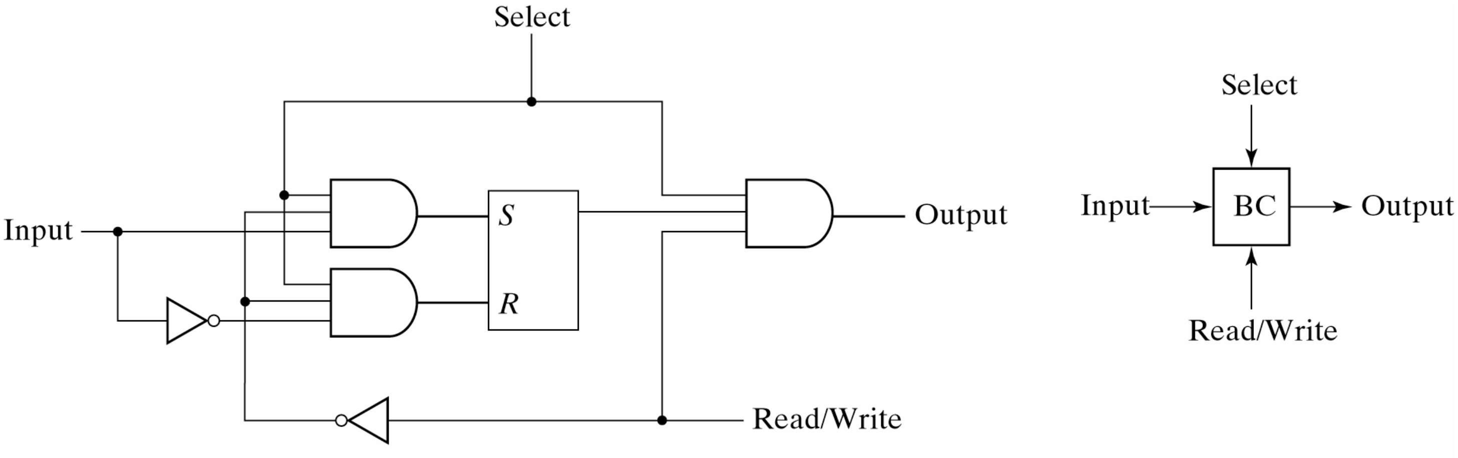
  1GB = 2^{30} bytes
- Note that 1kB = 1024 bytes is sometimes used to refer to 1000 bytes.
- e.g. A 1.92GB memory key contains over 2 billion bytes and so will be marketed as a 2GB memory key.



```
0000    FF D8 FF E1   1D FE 45 78   69 66 00 00   49 49 2A 00
0010    08 00 00 00   09 00 0F 01   02 00 06 00   00 00 7A 00
0020    00 00 10 01   02 00 14 00   00 00 80 00   00 00 12 01
0030    03 00 01 00   00 00 01 00   00 00 1A 01   05 00 01 00
0040    00 00 A0 00   00 00 1B 01   05 00 01 00   00 00 A8 00
0050    00 00 28 01   03 00 01 00   00 00 02 00   00 00 32 01
0060    02 00 14 00   00 00 B0 00   00 00 13 02   03 00 01 00
0070    00 00 01 00   00 00 69 87   04 00 01 00   00 00 C4 00
0080    00 00 3A 06   00 00 43 61   6E 6F 6E 00   43 61 6E 6F
0090    6E 20 50 6F   77 65 72 53   68 6F 74 20   41 36 30 00
00A0    00 00 00 00   00 00 00 00   00 00 00 00   B4 00 00 00
00B0    01 00 00 00   B4 00 00 00   01 00 00 00   32 30 30 34
00C0    3A 30 36 3A   32 35 20 31   32 3A 33 30   3A 32 35 00
00D0    1F 00 9A 82   05 00 01 00   00 00 86 03   00 00 9D 82
00E0    05 00 01 00   00 00 8E 03   00 00 00 90   07 00 04 00
```

Binary Storage Cell

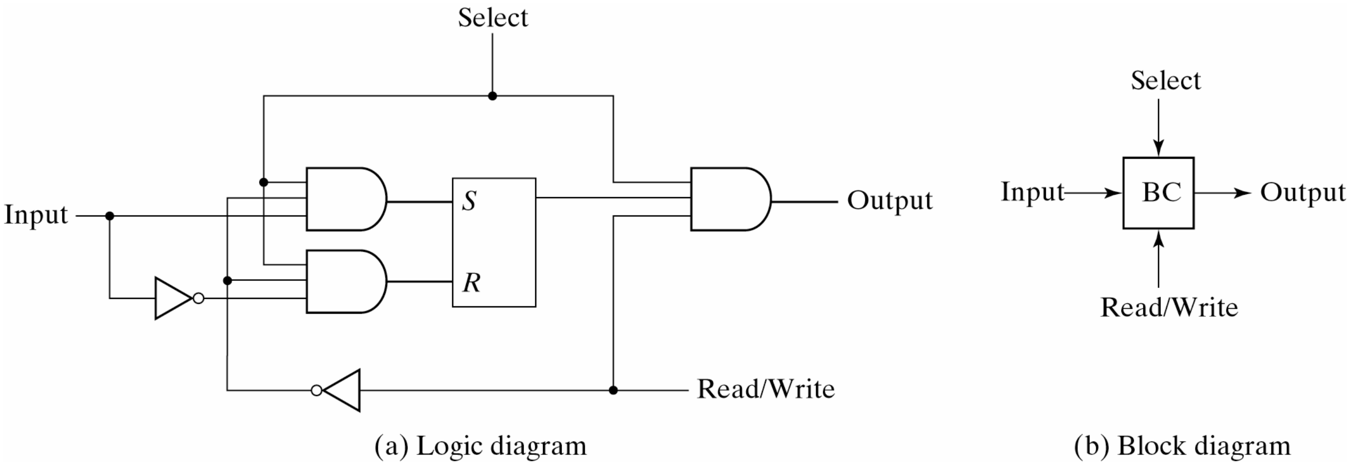
- The internal construction of a RAM consists of binary storage cells and associated decoding circuits for slecting indvidual words.
- The binary storage cell is designed to store a bit of information.
- Each cell has a data input and output and two control inputs Select and Read/Write.



(a) Logic diagram (b) Block diagram

Binary Storage Cell

- The select input enables the cell for reading or writing.
- A zero on the select line forces the RS latch to a no-change state and the cell output to zero.
- A one on the select line allows possible input or output data from the cell.

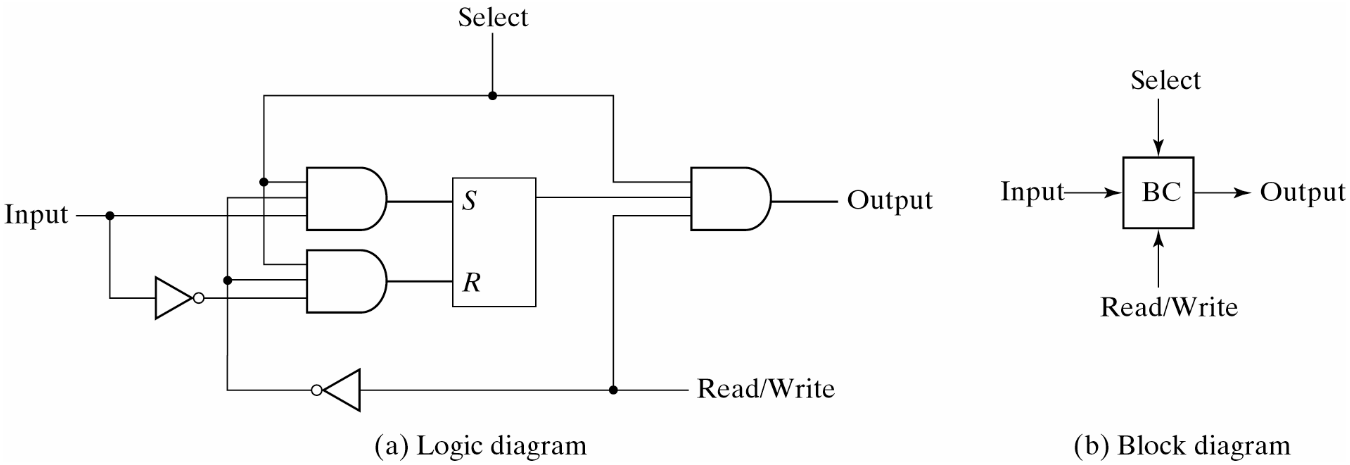


(a) Logic diagram

(b) Block diagram

Binary Storage Cell

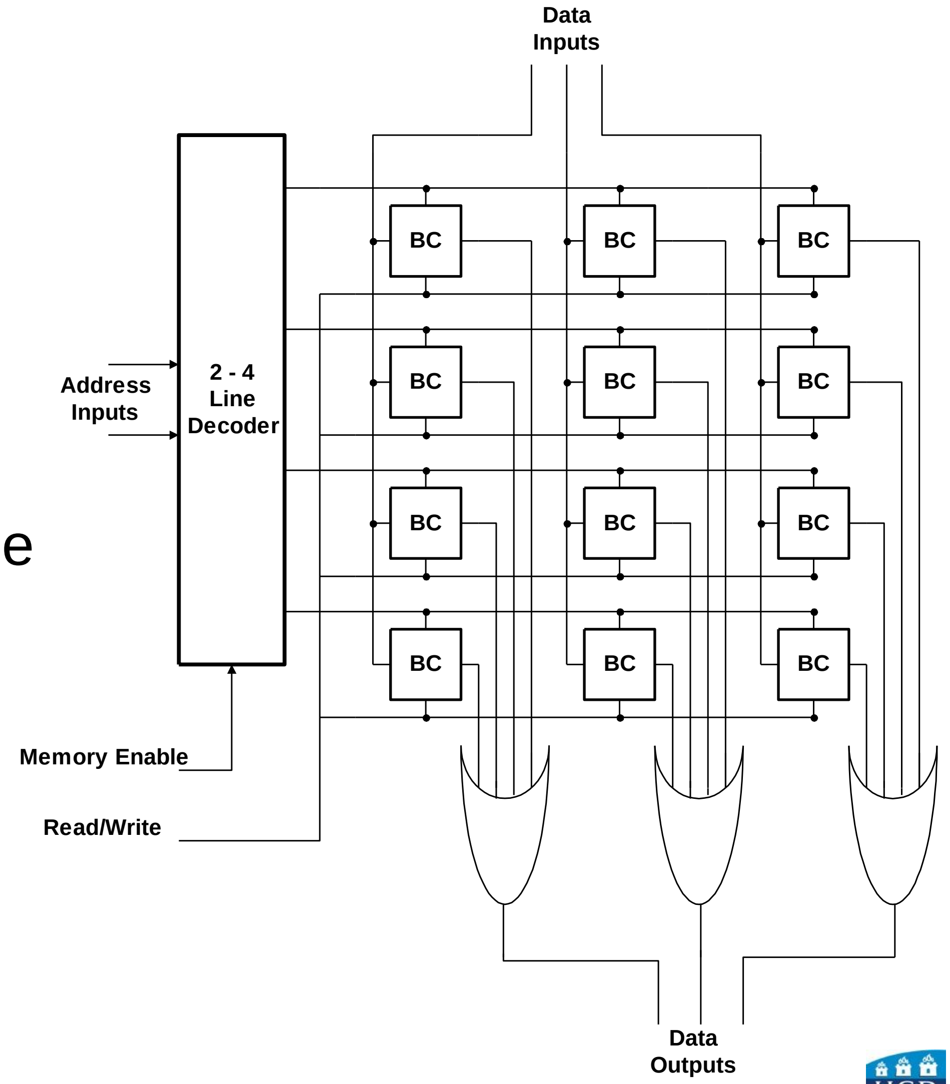
- Logic 0 in the Read/Write input provides the write operation by forming a path from the input terminal to the flip-flop.
- Logic 1 in the Read/Write input provides the read operation by forming a path from the flip-flop to the output.



(a) Logic diagram (b) Block diagram

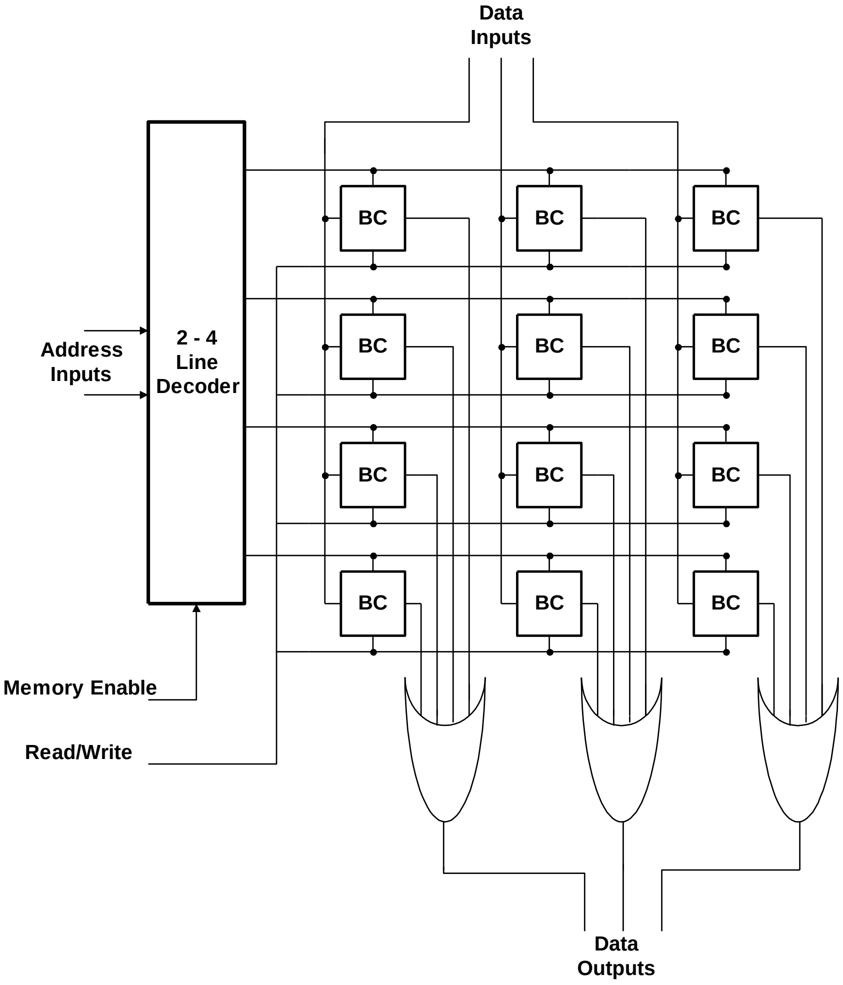
Random Access Memory

- 12 binary storage cells can be used to construct a RAM circuit consisting of 4 words of 3 bits each.
- 2 address lines are used to access the 4 words.
- A **2-4 line decoder** takes the address inputs and selects one of the 4 words.
- When the **memory enable** input is low none of the words are selected.



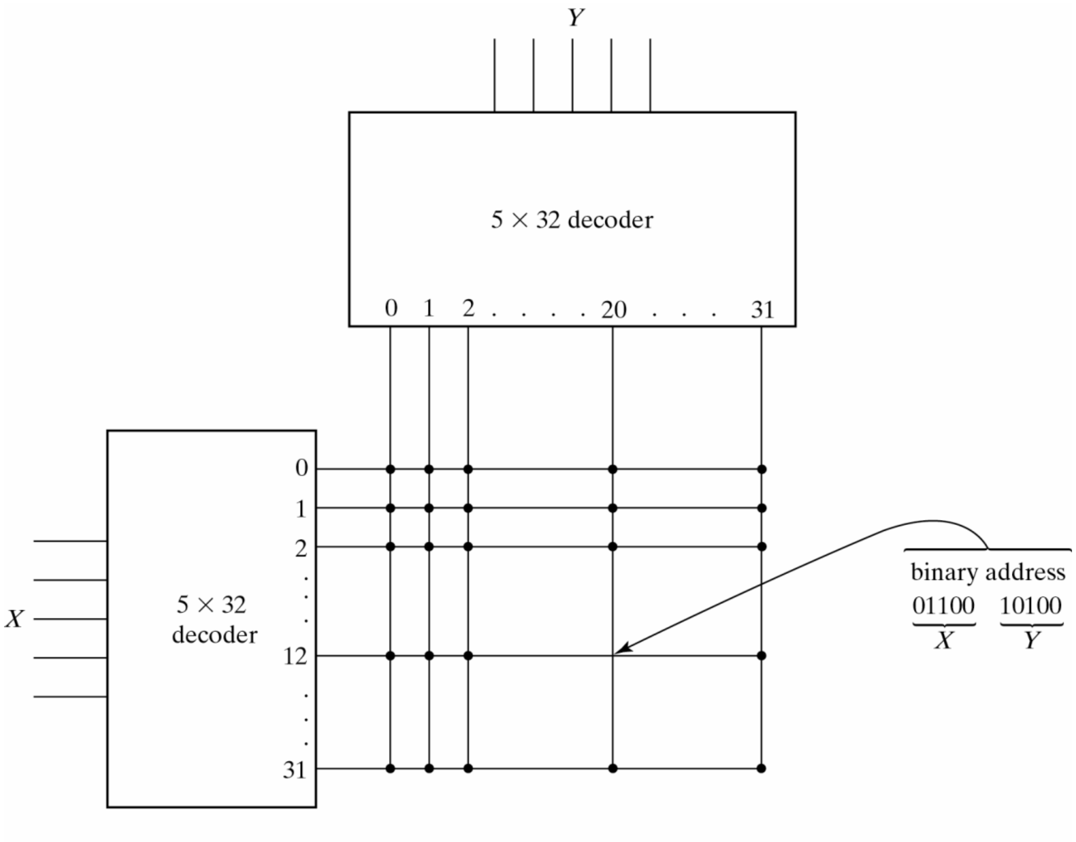
Random Access Memory

- Once a word has been selected the ***Read/Write*** input determines the operation.
- During a **Read** the outputs of the selected binary cells go through OR gates to the output terminals.
- During a **Write** the selected binary cells are loaded by the data inputs.



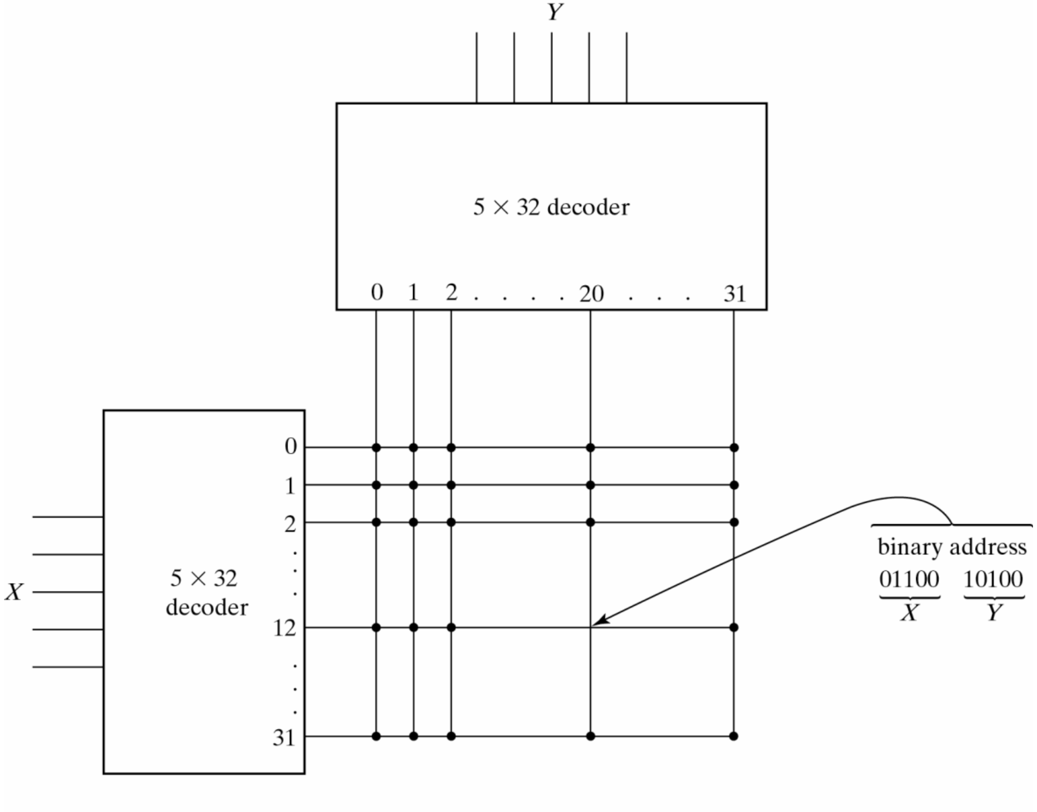
Random Access Memory

- In modern RAM circuits there are a large number of memory addresses.
- To cope with a large number of a addresses two-dimensional selection schemes are often used.



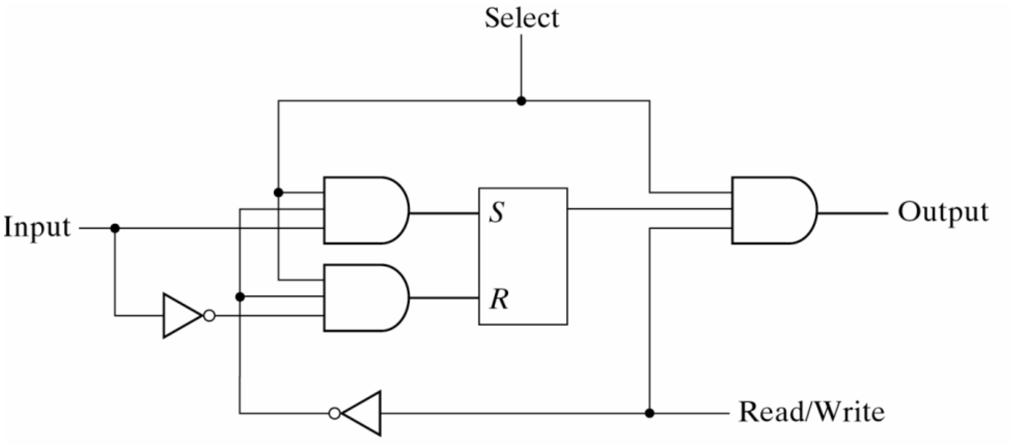
Random Access Memory

- In this 1K word memory a pair of 5x32 decoders are used instead of one 10x1024 decoder.
- A 10x1024 decoder requires 1024 AND gates with 10 inputs each.
- Two 5x32 decoders require 64 AND gates with five inputs each.



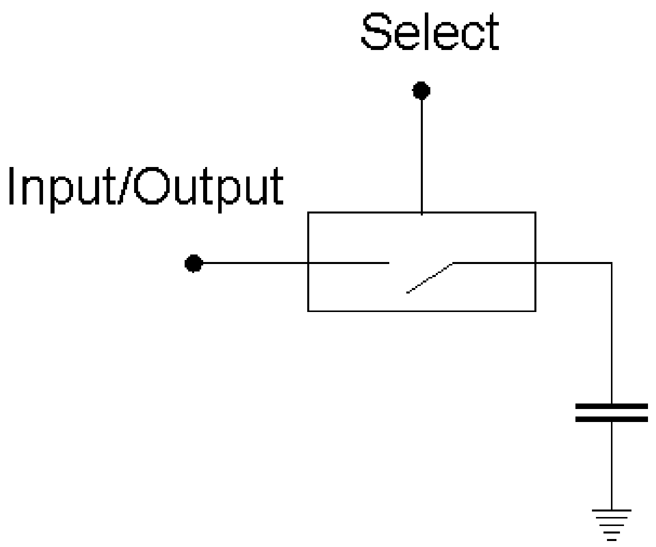
Static RAM

- At a fundamental level RAM circuits are constructed using transistors.
- The binary cell model we have been examining typically is constructed using six transistors.
- This memory cell is known as a SRAM (Static RAM) cell.



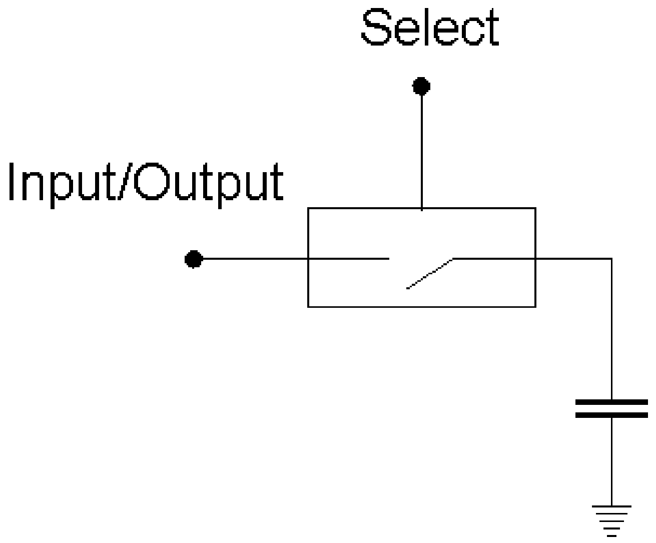
Dynamic RAM

- Higher density memories use more compact memory cell designs.
- The DRAM (Dynamic RAM) memory cell contains a MOS transistor and a capacitor.
- Here we model the cell as a binary switch and a capacitor.



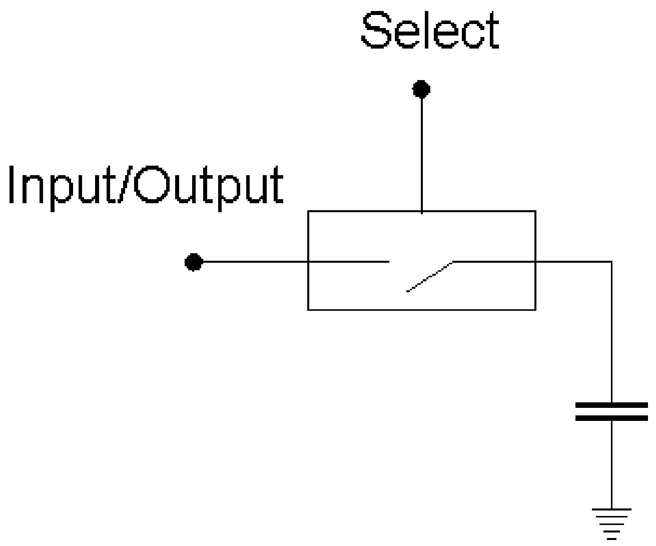
Dynamic RAM

- To write to the cell the Select input is set high closing the switch.
- The logic signal is applied to the input and either charges or discharges the capacitor.
  - To store a one current flows in charging the capacitor.
  - To store a zero current flows out discharging the capacitor.



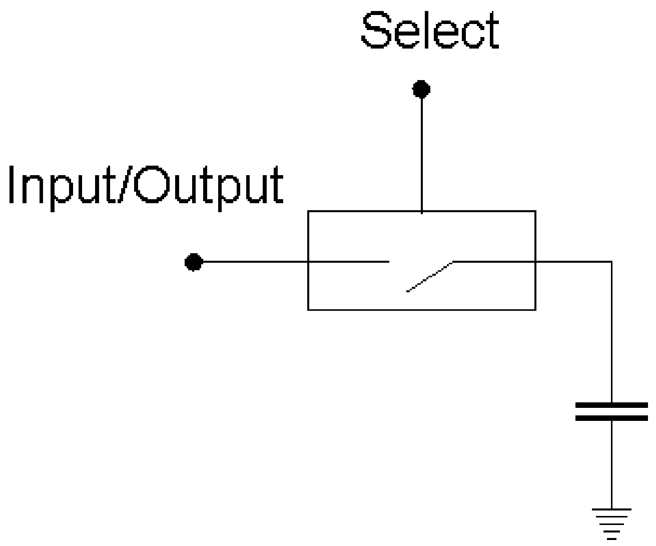
Dynamic RAM

- When the Select input is set low the switch is opened and the capacitor is disconnected from the rest of the circuit.
- The binary value is stored as a charge (or lack of charge) in the capacitor.



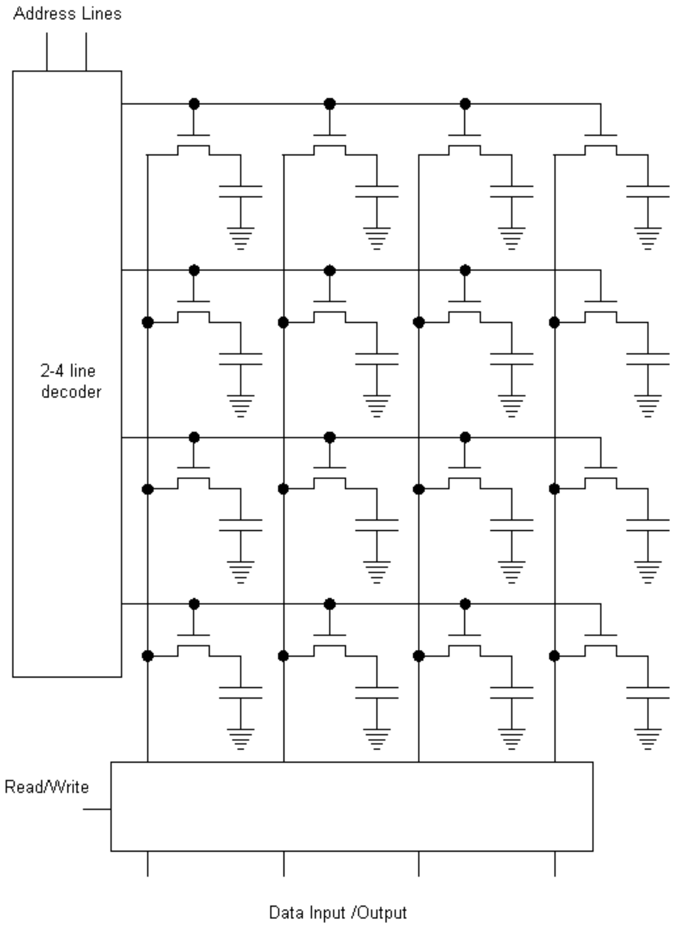
Dynamic RAM

- To read from the cell the Select input is set high closing the switch.
- The capacitor is now connected to the output.
- If the capacitor discharges and current flows out a one is read.
- If the capacitor doesn‘t discharge and now current flows out a zero is read.



Dynamic RAM

- On the right we have a 4x4 DRAM implementation.
- It will typically occupy a quarter of the space of the equivalent SRAM.
- In practice the MOS transistors do not act as perfect switches and charge will leak away over time.
- As a result the charge needs to be dynamically topped up before the capacitors discharge.

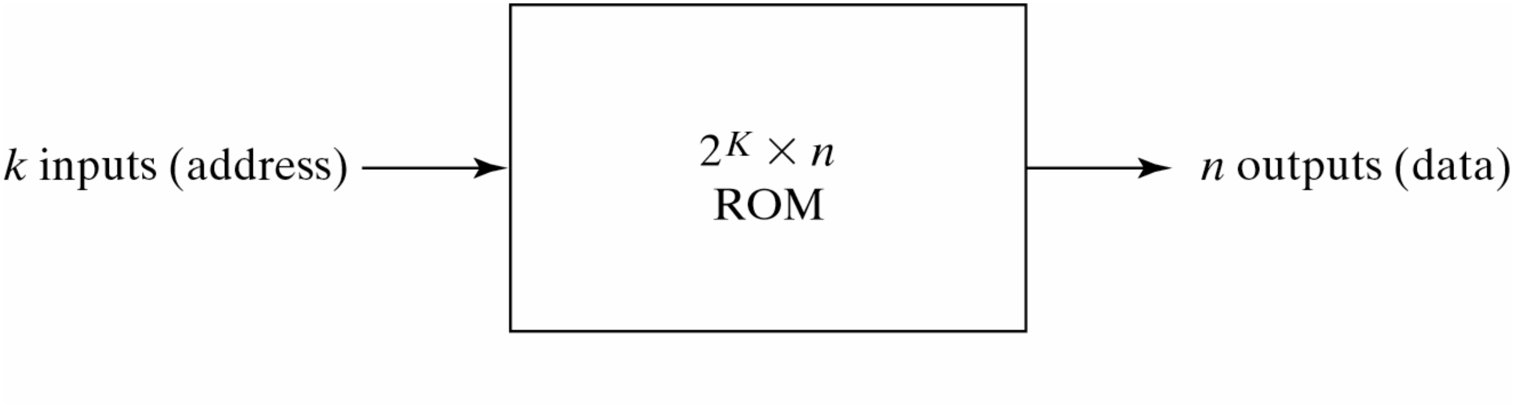


Volatile and Nonvolatile Memory

- Memory units that lose stored information when power is turned off are said to be **volatile**.
- CMOS integrated circuit RAMs, both static and dynamic, are of this category, since the binary cells need external power to maintain the stored information.
- In contrast, a **nonvolatile** memory, such as magnetic disk, retains its stored information after the removal of power. The data stored on magnetic components are represented by the direction of magnetization, which is retained after power off.
- Read-Only-Memory (ROM) is another nonvolatile memory.

Read Only Memory

- A ROM device is designed to store binary information permanently.
- The basic ROM will have k input lines to access one of 2^k words (n-bit).
- The binary information is read from n output lines.
- There is no data input line because there is no write function.

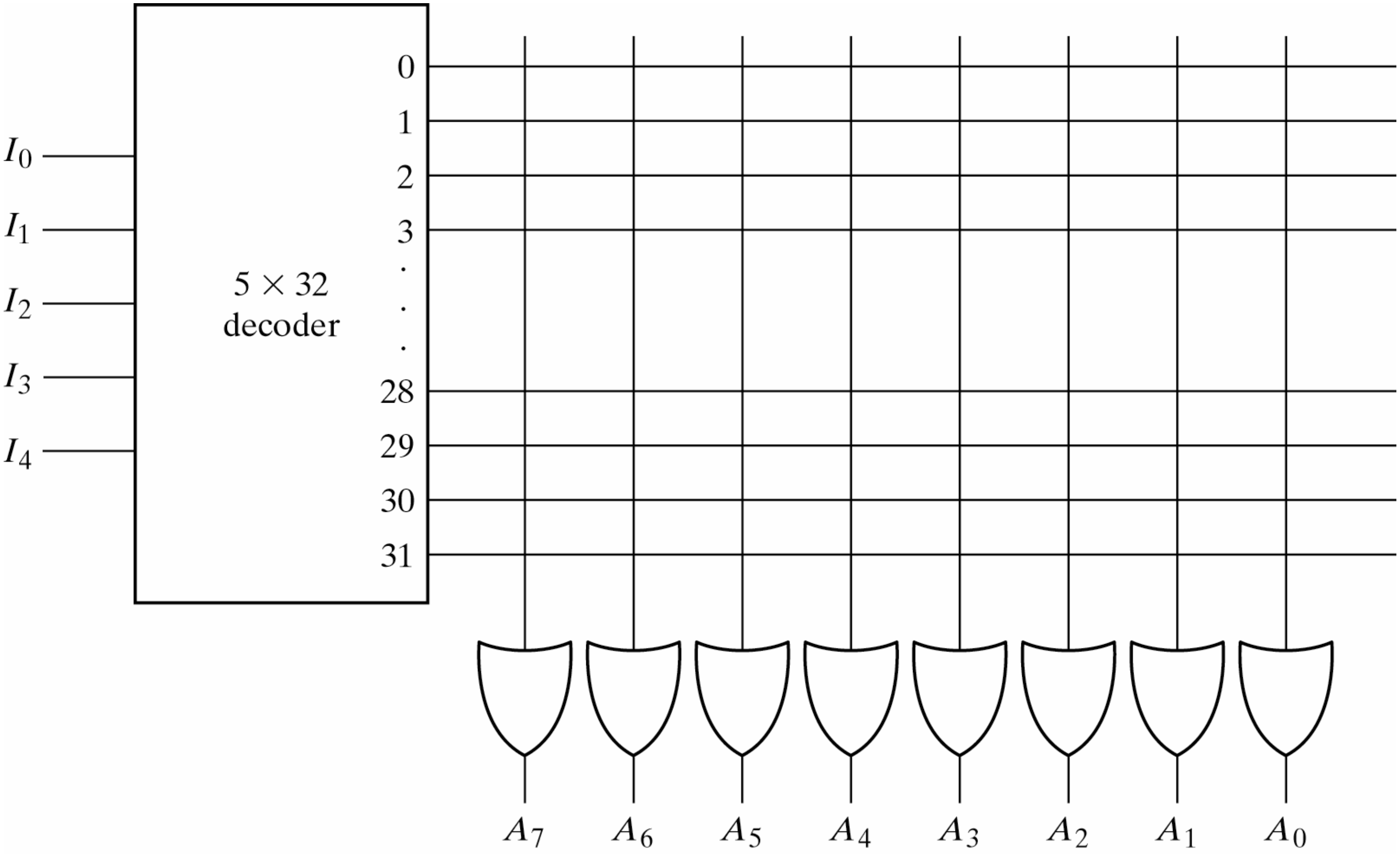


Read Only Memory

To illustrate we examine a 32 x 8 ROM.

- 32 words, 8 bits each
- 5 address lines and a 5 x 32 decoder
- 8 OR gates, 32 inputs each

Read Only Memory

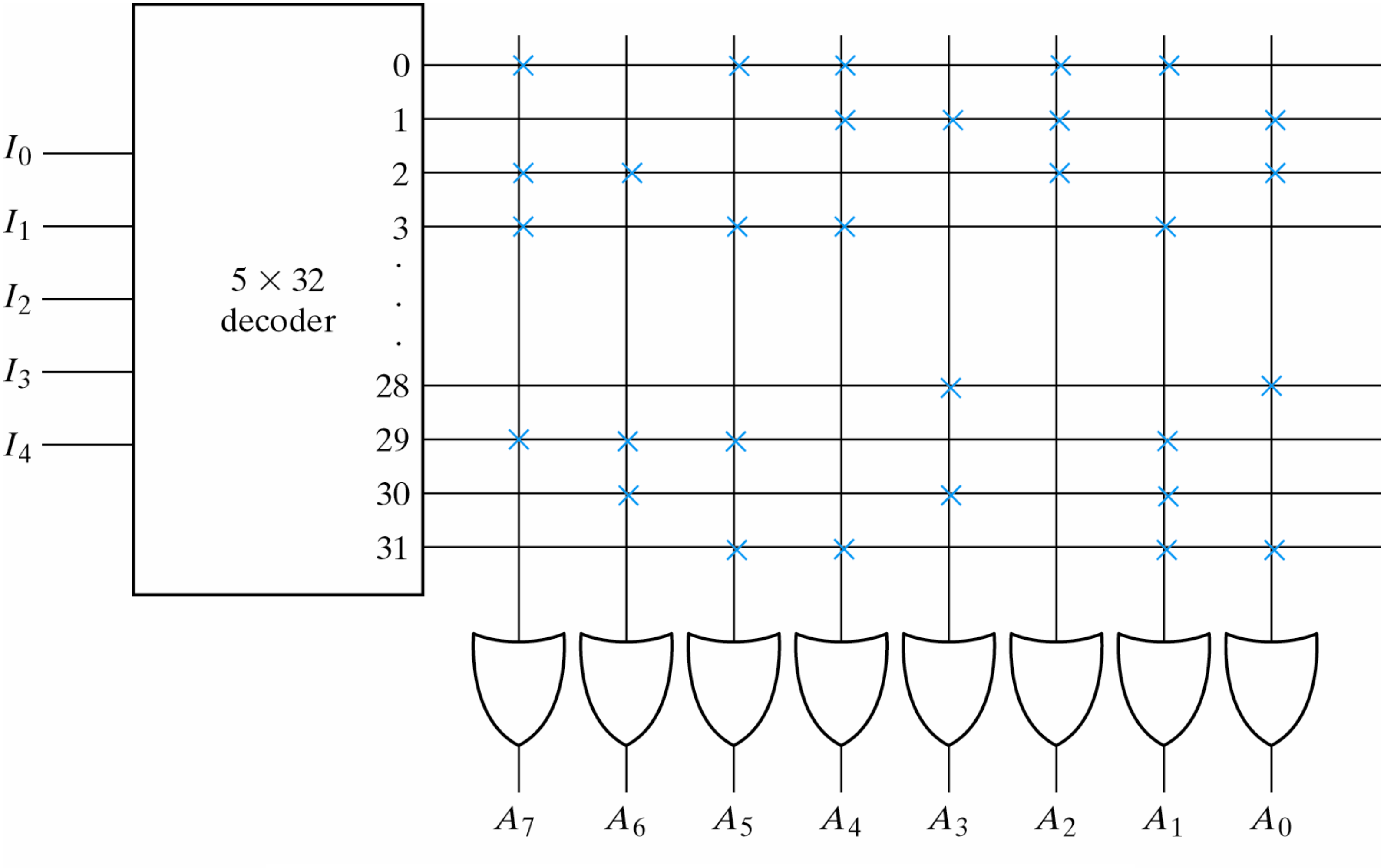


Read Only Memory

Programming the ROM

- The intersections between the lines are known as cross points.
- The cross points are separated by programmable fuses.
- To program the ROM fuses are alternatively blown (creating an open circuit) or left unchanged (short circuit).
- A connected cross point is represented by an ‘x’

Read Only Memory



Read Only Memory

- The first word of the ROM is 10110110 and is stored at address 00000.
- The fourth word 10110010 is stored at 00011
  - When 00011 is loaded into the decoder all of the 32 outputs are zero except for output 3.
  - The input lines into the OR gates will all be set low except for the fourth lines.
  - From the OR gate truth table we know that the output of the ROM will be $\{A_7,A_6,A_5,A_4,A_3,A_2,A_1\}=10110010$

Types of ROMs

- The required paths in a ROM may be programmed in several different ways.
- **Mask Programming**, where cross points are programmed during manufacture of the semiconductor.
- **Programmable ROM (PROM)**: initial unit contains all the fuses intact, giving all 1’s in the bits of the stored words. The fuses can be blown by the application of a high-voltage pulse to the device through a special pin. A blown fuse defies a binary 0 state and an intact fuse gives a binary 1 state. This procedure allows the user to program the PROM to achieve the desired functionality.

Types of ROMs

- The hardware procedure for mask programming ROMs or PROMs is irreversible, and once programmed, the fixed pattern is permanent and cannot be altered.
- **EPROM** (erasable programmable ROM) circuits can be returned to an initial state through the application of ultra-violet light.
- **EEPROM** (electrically erasable programmable ROM) can be erased using an electrical signal.

Flash Memory

- Flash memory is another type of nonvolatile memory.
- It is similar to EEPROM, but have additional built-in circuitry to selectively program and erase the device in-circuit, without the need for a special programmer.
- Flash memory has widespread applications in many areas, such as mobile phones, digital camera, MP3 player, USB data storage, etc.
- One more recent application for flash memory (called solid-state drive, SSD) is as a replacement for hard disks, e.g. in MacBook Air.

Exercises

1) A memory unit is built to accommodate a word count of 8K with 16 bits per word. How many address lines and input/output data lines are required?
2) How many bytes are stored in this memory unit?
3) A 16K 4-bit memory uses a 2D memory address scheme. What is the size of each decoder and how many AND gates are required in total?

Solutions

1) 8K words = 8 1024 = 8192 words = 2^{13} words. This means we need 13 address lines to encode 8192 possible memory locations. Since each word is 16 bits wide, we need 16 data lines (i.e. a data bus 16 lines wide).
2) 8,192 words @ 16 bits each = 131,072 bits = 16,384 bytes.
3) We require 16K different memory locations = 2^{14} memory locations = $2^7 \cdot 2^7$ memory locations = Two 7 128 decoders. This will require 256 AND gates.