RISC (Reduced Instruction Set Computer)

- Reduced Instruction Set Computing (RISC), is a microprocessor CPU design philosophy that favors a smaller and simpler set of instructions that all take about the same amount of time to execute.

Instruction Pipeline

- Instruction pipelining is a method for increasing the throughput of a digital circuit, particularly a CPU, and implements a form of instruction level parallelism.
- The idea is to divide the logic into stages, and to work on different data within each stage.
- If you have two loads of laundry to do, you can either wash the first load and then dry the first load, before moving onto the next, or, you can wash the first load, and when you put it in to dry, you can put the next load into wash. If each step takes 20 minutes, then you will finish in 60 minutes instead of 80.
- A pipelined digital circuit works the same way. Data enters the first stage, and takes some time to process. When the data finishes the first stage, the clock ticks, and the intermediate results are latched into registers at the head of the next stage, while the next set of data enters the beginning of the first stage.

Harvard Architecture

- The term **Harvard architecture** originally referred to computer architectures that used physically separate storage and signal pathways for their instructions and data (in contrast to the *von Neumann architecture*).
- In a computer with a von Neumann architecture, the CPU can be either reading an instruction or reading/writing data from/to the memory. Both cannot occur at the same time since the instructions and data use the same signal pathways and memory.
- In a computer with Harvard architecture, the CPU can read both an instruction and data from memory at the same time. A computer with Harvard architecture can be faster because it is able to fetch the next instruction at the same time it completes the current instruction. Speed is gained at the expense of more complex electrical circuitry.

Types of Memories

- **Volatile** –Memory units lose the stored information when power is turned off.
  - **Random access Memory (RAM)**–each word occupy one particular location. the access time is always the same.
    - **Static RAM** –The stored information remains valid as long as power is applied to the unit. Static **RAM** is easier to use and has shorter read and write cycles.
    - **Dynamic RAM**–The capacitors must be periodically recharged by refreshing the dynamic memory. Dynamic **RAM** offers reduced power consumption and larger storage capacity.
- **Nonvolatile**–A nonvolatile memory retains its stored information after removal of power.
  - Magnetic disk
  - ROM (Read Only Memory)

Random-Access Memory

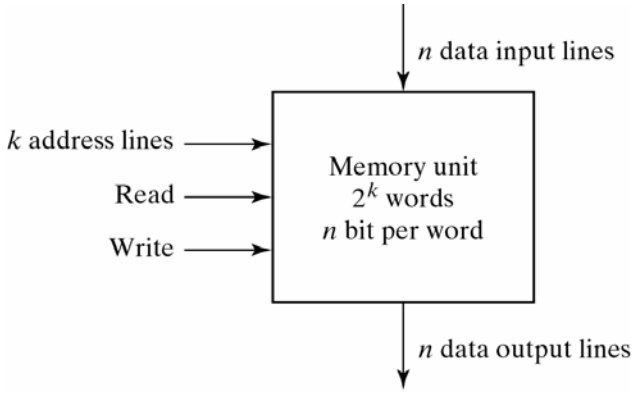


Fig. 7-2 Block Diagram of a Memory Unit

- The address lines select one particular word.
- A decoder inside the memory accepts this address and opens the paths needed to select the word specified

Content of a 1024x16 Memory

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

Fig. 7-3 Content of a 1024 × 16 Memory

- The 1K * 16 memory has 10 bits in the address and 16 bits in each word

Write and Read Operations

- Write operation
  1. Transfer the binary address of the desired word to the address lines.
  2. Transfer the data bits that must be stored in memory to the data input lines.
  3. Activate the write input
- Read operation
  1. Transfer the binary address of the desired word to the address lines.
  2. Activate the read input

Memory Decoding

- The equivalent logic of a binary cell that stores one bit of information
- The binary cell stores one bit in its internal flip-flop
- It has three inputs and one output. The read/write input determines the cell operation when it is selected.

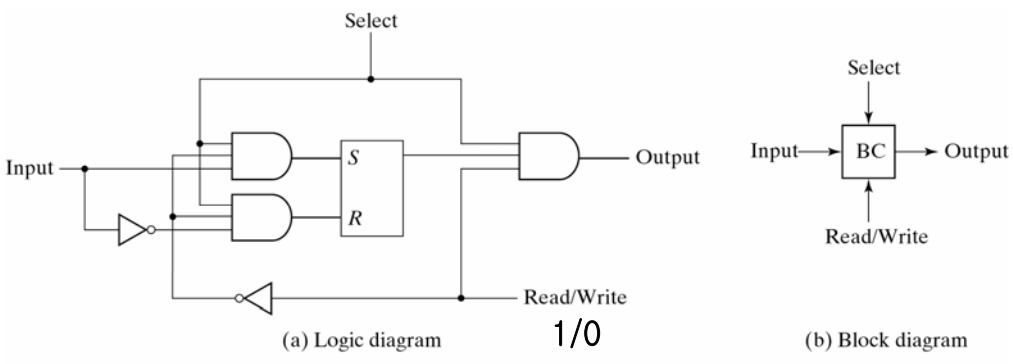
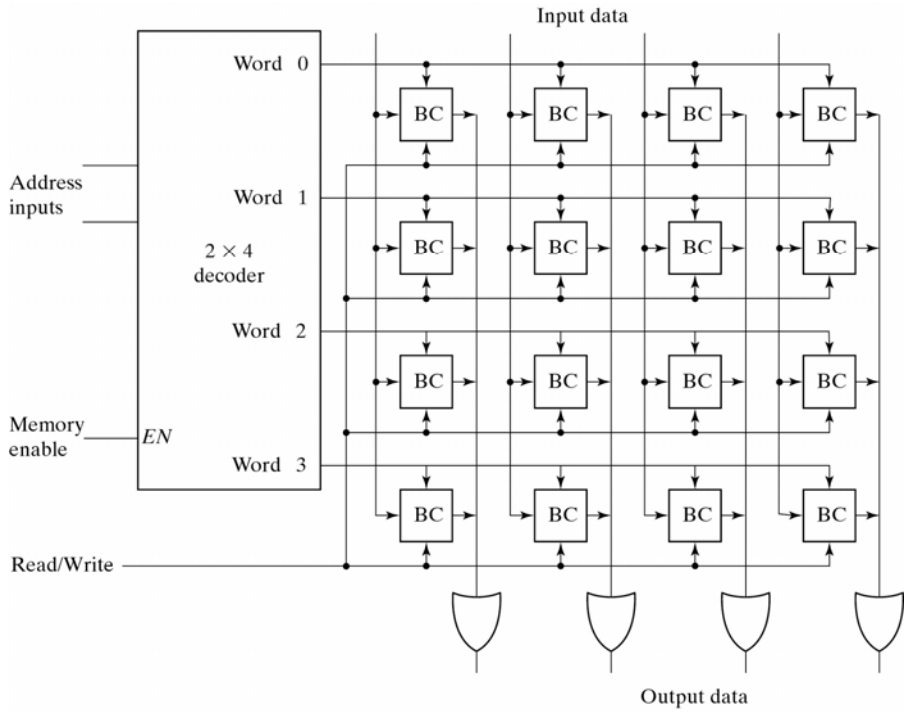


Fig. 7-5 Memory Cell

Internal Construction



4 words of 4 bits
=16 binary cells

Fig. 7-6 Diagram of a 4 × 4 RAM

Read-Only Memory

ROM = AND gates connected as a decoder + a number of OR gates

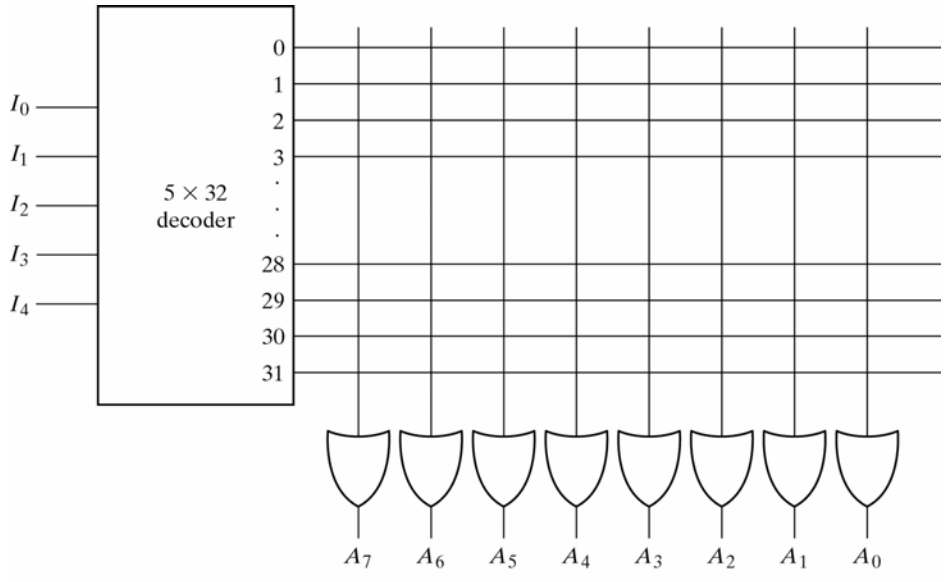


Fig. 7-10 Internal Logic of a 32 × 8 ROM

ROM Truth Table (Partial)

Table 7-3
ROM Truth Table (Partial)

Inputs					Outputs							
I4	I3	I2	I1	I0	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	0	1	0	1	1	0	1	1	0
0	0	0	0	1	0	0	0	1	1	1	0	1
0	0	0	1	0	1	1	0	0	0	1	0	1
0	0	0	1	1	1	0	1	1	0	0	1	0
		⋮					⋮					
1	1	1	0	0	0	0	0	0	1	0	0	1
1	1	1	0	1	1	1	1	0	0	0	1	0
1	1	1	1	0	0	1	0	0	1	0	1	0
1	1	1	1	1	0	0	1	1	0	0	1	1

Combinational Circuit Implementation

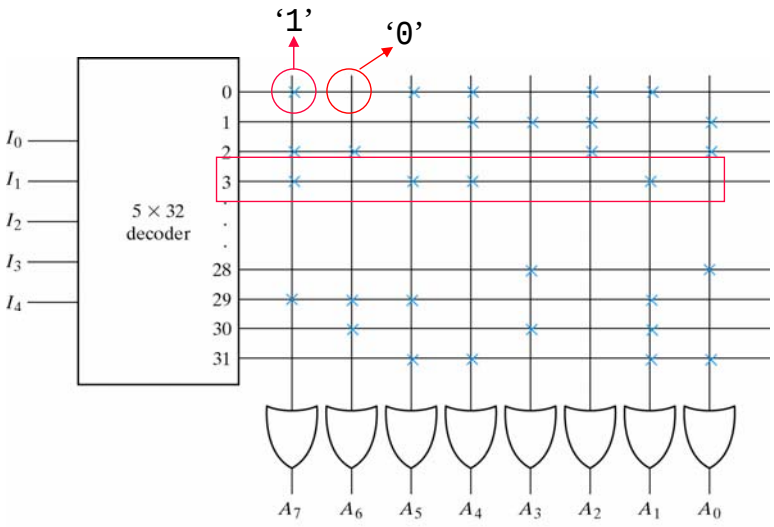


Fig. 7-11 Programming the ROM According to Table 7-3

Types of ROMs

- PROM: Programmable Read-Only Memory
  - Irreversible and permanent
- EPROM(Erasable PROM)
  - can be restructured to the initial value (UV light->discharge)
- EEPROM(Electrically erasable PROM)
  - can be erased with electrical signals instead of ultra violet light.

EEPROM vs. Flash Memory

- EEPROM
  - 1바이트 write 할 때마다 수ms 이상이 소요됨
  - 실시간으로 사용되는 변수를 저장하는 메모리나 스택 메모리로 사용 불가능
  - 오랫동안 저장하고, 주로 읽어 사용하는 기능, 전원을 꺼도 유지되는 중요한 설정 값 저장용도로 사용
- Flash Memory
  - 페이지 모드 writing 기능만을 가지는 EEPROM
  - 바이트 단위로 read 가능, but block (page, sector)단위로만 write 가능
  - Block 크기: 64, 128, 256 바이트 ~ 128KB
  - EEPROM보다 메모리 셀의 구조가 간단하여 훨씬 대용량의 메모리 소자를 만드는데 적합
  - 1개의 block 전체를 write하는데 수ms 정도 걸리므로, 대용량 데이터를 write하는 경우에 EEPROM보다 훨씬 빠르다