CME2206 LAB 2
Memory Unit

Preliminary Work

1- Please read "Preliminary Reading – Memory Unit" at the end of this document.
2- Please examine "Quartus II Tutorial - How to Add Mega Functions to Your Design" and try the steps.

Experiment

1- You are expected to design a 16x4 data memory and 16x9 code memory. Data memory will be used to get operands and store results of operations. Code memory will be operated to store embedded instructions of programs. Firstly you should decide which memory types (RAM or ROM) you will use for these memories and then add relevant megafunctions (lpm_rom, lpm_ram_dq) or relevant installed plugins (RAM: 1-Port or ROM: 1-Port) to your design using Quartus II based upon your decision.
2- Fill the memory files as below

Data Memory

Address		Data value
0	0000	0110
1	0001	0000
2	0010	0000
3	0011	0000
4	0100	0000
5	0101	0111
6	0110	0000
7	0111	0000
8	1000	0000
9	1001	0000
10	1010	0011
11	1011	0000
12	1100	0000
13	1101	1011
14	1110	0000
15	1111	0000

Code Memory

Address		Instruction
0	0000	000000000
1	0001	000000000
2	0010	000000000
3	0011	000000000
4	0100	000000000
5	0101	000100100
6	0110	000000000
7	0111	000000000
8	1000	000100000
9	1001	000000000
10	1010	000000000
11	1011	000000000
12	1100	100001111
13	1101	000000000
14	1110	000001010
15	1111	000000000

3- Verify your design using waveform for following states
   a. Write value 12_{10} to the address 7_{10} of data memory
   b. Read value from address 13_{10} of data memory
   c. Read value from address 12_{10} of code memory

Preliminary Reading – Memory Unit

Memory Unit

A **memory unit** is a collection of storage cells together with associated circuits needed to transfer information in and out of storage. The memory 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 code, one or more alphanumeric characters, or any other binary-coded information.
A group of eight bits is called a byte. Most computer memories use words whose number of bit is a multiple of 8. Thus a 16-bit word contains two bytes, and a 32-bit word is made up of four bytes. The capacity of memories in commercial computers is usually stated as the total number of bytes that can be stored.

The internal structure of a memory unit is specified by the **number of words** it contains and the **number of bits** in each word. Special input lines called **address lines** select one particular word. Each word in memory is assigned an identification number, called an **address**, staring from 0 and continuing with 1, 2, 3, up to $\mathbf{2^k - 1}$ where **k** is the number of address lines. The selection of a specific word inside the memory is done by applying the k-bit binary address to the address lines.

A **decoder** inside the memory accepts this address and opens the paths needed to select the bits of the specified word.

Computer memories may range from 1024 words, requiring an address of 10 bits, to 2^{32} words, requiring 32 address bits. It is customary to refer to the number of words (or bytes) in a memory with one of the letters:

- **K(Kilo)** is equal to 2^{10}
- **M(Mega)** is equal to 2^{20}
- **G(Giga)** is equal to 2^{30}

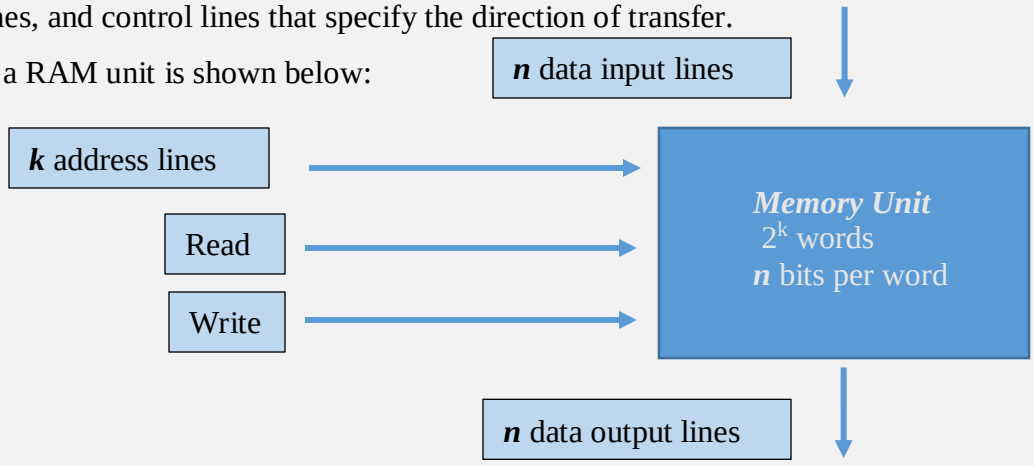
Two major types of memories are used in computer systems: **Random Access Memory (RAM)** and **Read Only Memory (ROM)**.

Random-Access Memory

In random-access memory (RAM) the memory cells can be accessed for information transfer from any desired random location. That is, the process of locating a word in memory is the same and requires an equal amount of time no matter where the cells are located physically in memory: thus the name "random access."

Communication between a memory and its environment is achieved through data input and output lines, address selection lines, and control lines that specify the direction of transfer.

A block diagram of a RAM unit is shown below:

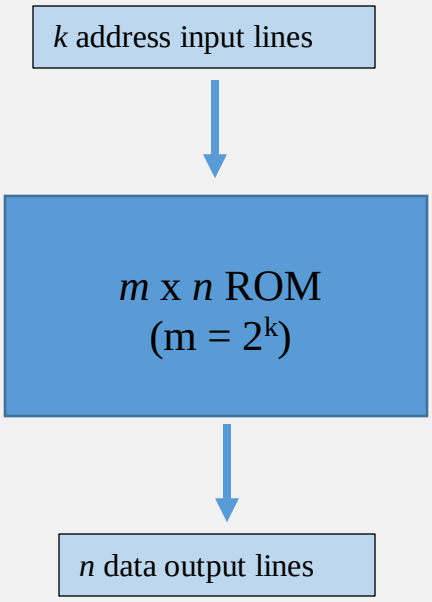


Read-Only Memory

As the name implies, a **read-only memory (ROM)** is a memory unit that performs the read operation only; it does not have a write capability. This implies that the binary information stored in a ROM is made permanent during the hardware production of the unit and cannot be altered by writing different words into it.

Whereas a RAM is a general-purpose device whose contents can be altered during the computational process, a ROM is restricted to reading words that are **permanently** stored within the unit. The binary information to be stored, specified by the designer, is then embedded in the unit to form the required interconnection pattern. ROMs come with special internal electronic fuses that can be **programmed** for a specific configuration. Once the pattern is established, it stays within the unit even when power is turned off and on again.

An **m x n** ROM is an array of binary cells organized into **m** words of **n** bits each. As shown in the block diagram below, a ROM has **k** address input lines to select one of $\mathbf{2^k = m}$ words of memory, and n input lines, one for each bit of the word. An integrated circuit ROM may also have one or more enable inputs for expanding a number of packages into a ROM with larger capacity.



Resource: *Mano M.M., Computer System Architecture, 3rd ed.*