PATRIoT lab

Lecture 23 – Memory architecture 1

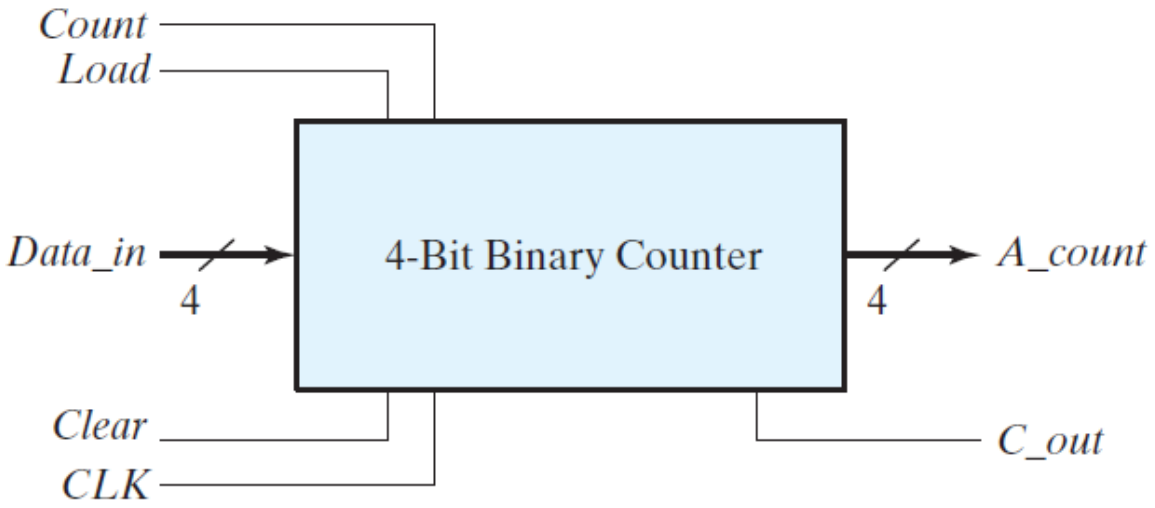
Dr. Aftab M. Hussain,

Assistant Professor, PATRIoT Lab, CVEST

Chapter 7

Synchronous counter – with parallel load

- The four control inputs— *Clear*, *CLK*, *Load*, and *Count* —determine the next state
- The *Clear* input is asynchronous and, when equal to 0, causes the counter to be cleared regardless of the presence of clock pulses or other inputs
- With the *Load* and *Count* inputs both at 0, the outputs do not change, even when clock pulses are applied
- A *Load* input of 1 causes a transfer from inputs I_0 - I_3 into the register during a positive edge of *CLK*
- The *Load* input must be 0 for the *Count* input to control the operation of the counter



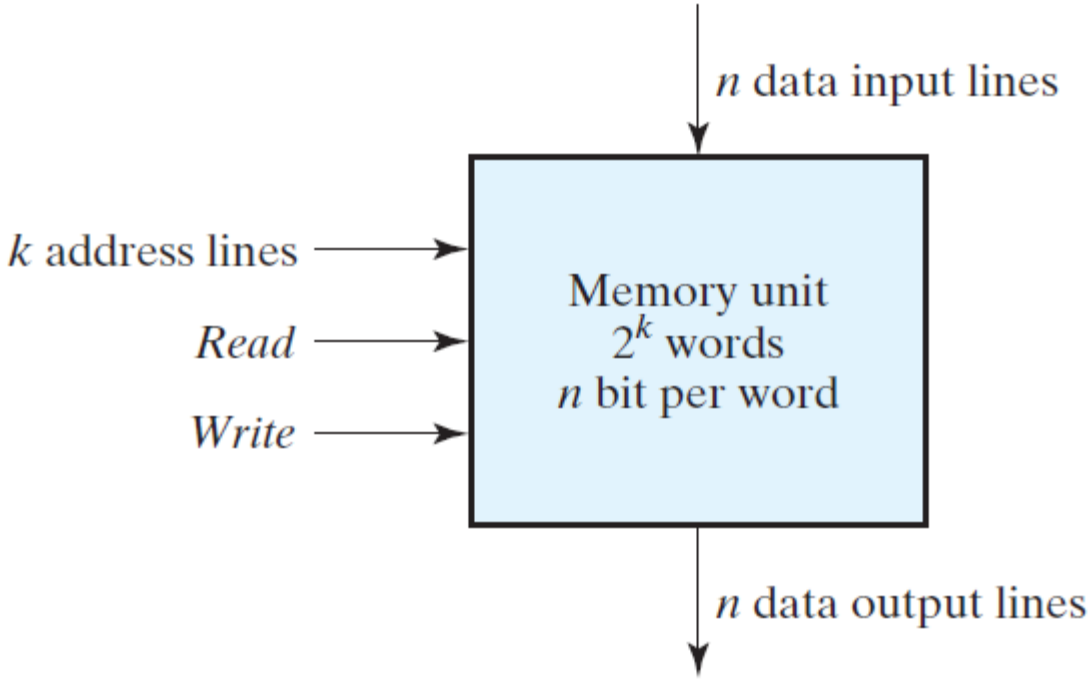
Clear	CLK	Load	Count	Function
0	X	X	X	Clear to 0
1	↑	1	X	Load inputs
1	↑	0	1	Count next binary state
1	↑	0	0	No change

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
- 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
- Most computer memories use words that are multiples of 8 bits in length
- The capacity of a memory unit is usually stated as the total number of bytes that the unit can store

Random Access Memory (RAM)

- Communication between memory and its environment is achieved through data input and output lines, address selection lines, and control lines that specify the direction of transfer
- The *n* data input lines provide the information to be stored in memory, and the *n* data output lines supply the information coming out of memory
- The *k* address lines specify the particular word chosen among the many available
- The two control inputs specify the direction of transfer desired: The *Write* input causes binary data to be transferred into the memory, and the *Read* input causes binary data to be transferred out of memory
- The address lines select one particular word



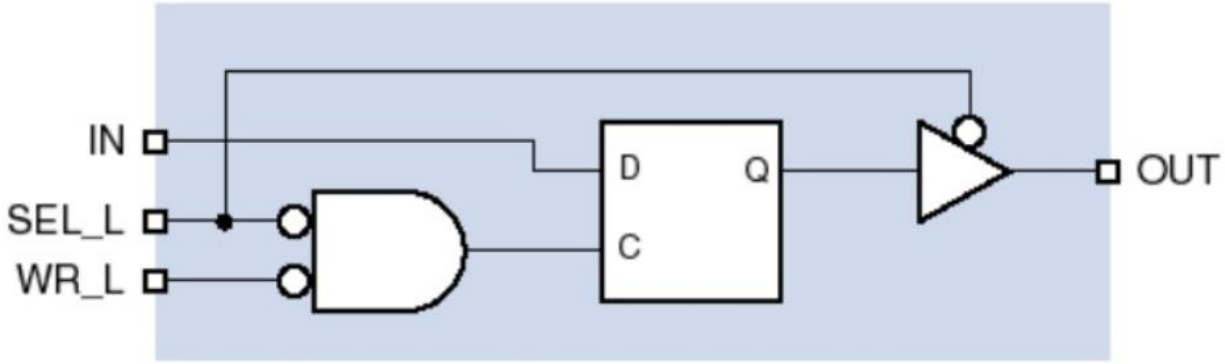
Random Access Memory (RAM)

- Consider, for example, a memory unit with a capacity of 1K words of 16 bits each
- Since 1K = 1,024 = 2^{10} and 16 bits constitute two bytes, we can say that the memory can accommodate 2,048 = 2Kb
- The words are recognized by their decimal address from 0 to 1,023
- The equivalent binary address consists of 10 bits
- A word in memory is selected by its binary address
- When a word is read or written, the memory operates on all 16 bits as a single unit

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

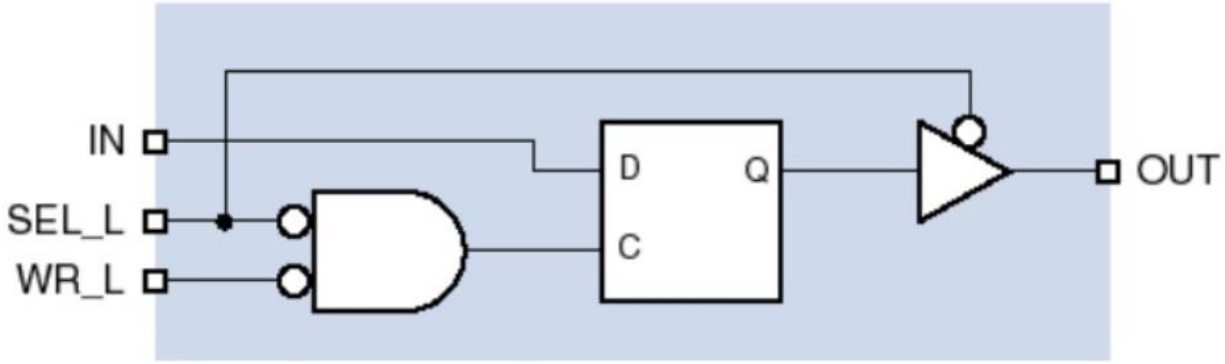
Random Access Memory (RAM) – internal structure

- The internal construction of a RAM of *m* words and *n* bits per word consists of *m* * *n* binary storage cells and associated decoding circuits for selecting individual words
- The binary storage cell (SR latch) is the basic building block of a memory unit
- The storage part of the cell is modeled by an *SR* latch with associated gates to form a *D* latch
- *Note that this is not a D flip-flop*



Write and Read operations

- The two operations that RAM can perform are the write and read operations
- The write signal specifies a transfer-in operation and the read signal specifies a transfer-out operation
- On accepting one of these control signals, the internal circuits inside the memory provide the desired operation

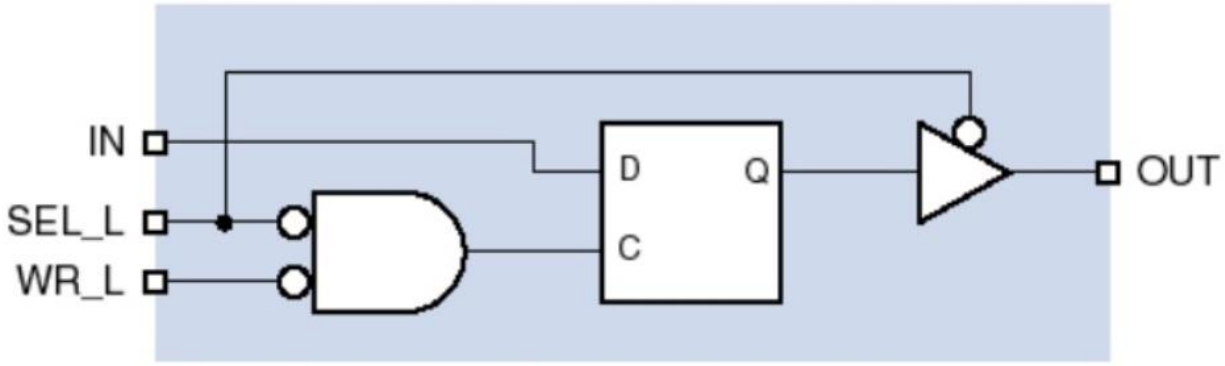


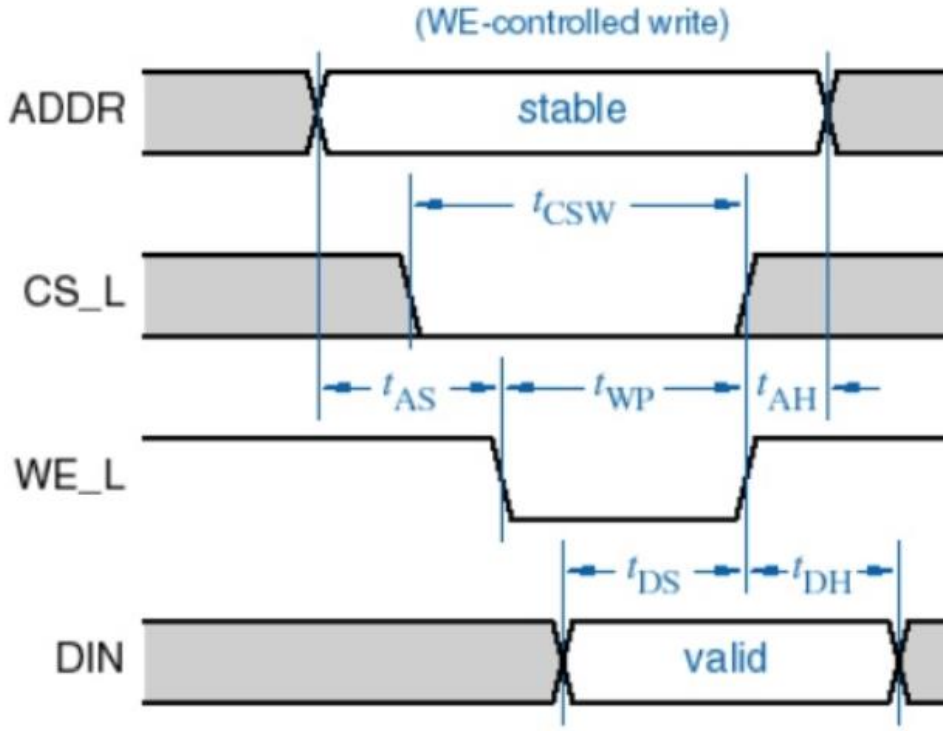
Write and Read operations

- The steps that must be taken for the purpose of transferring a new word to be stored into memory are as follows:

1. Apply the binary address of the desired word to the address lines.
2. Activate the *write* input.
3. Apply the data bits that must be stored in memory to the data input lines.

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



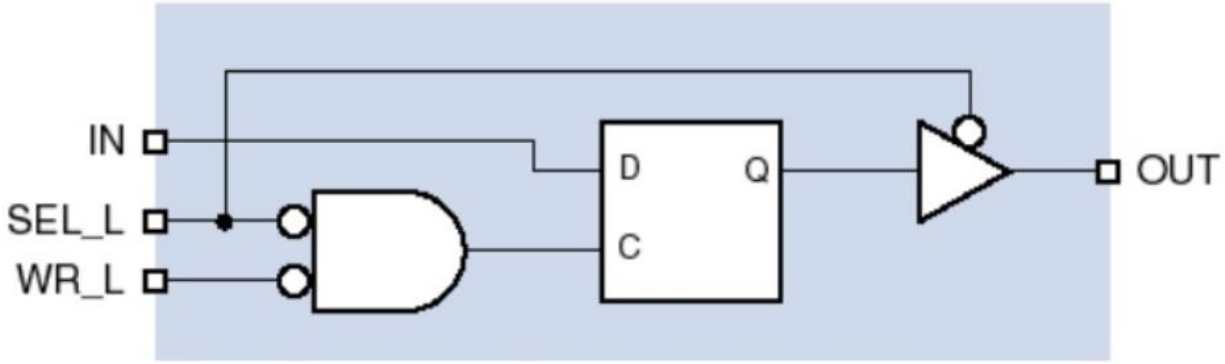


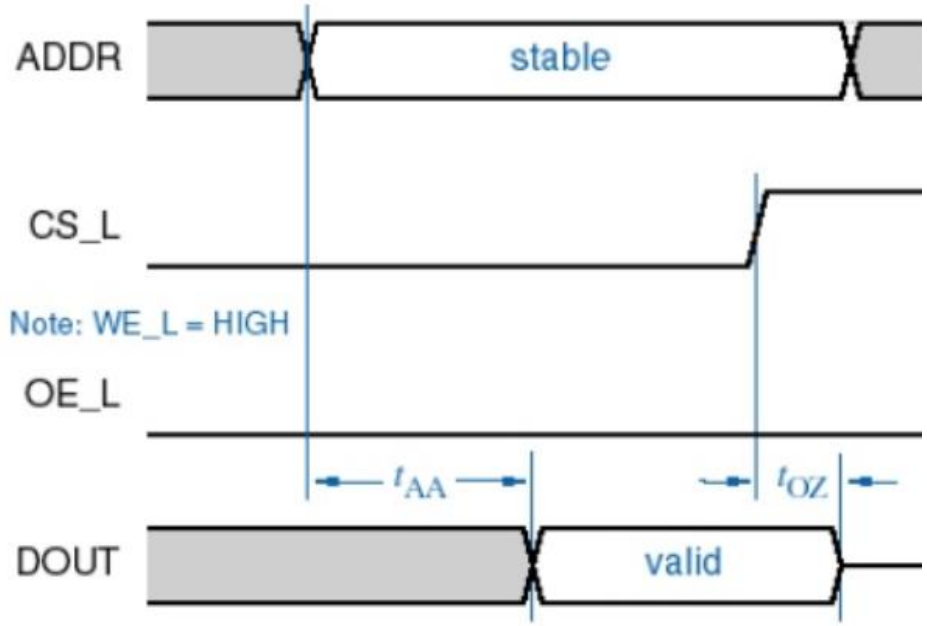
Write and Read operations

- The steps that must be taken for the purpose of transferring a stored word out of memory are as follows:

1. Apply the binary address of the desired word to the address lines

- The data shows up on the output lines after a certain delay from the application of the stable address (selecting the word)





Memory design