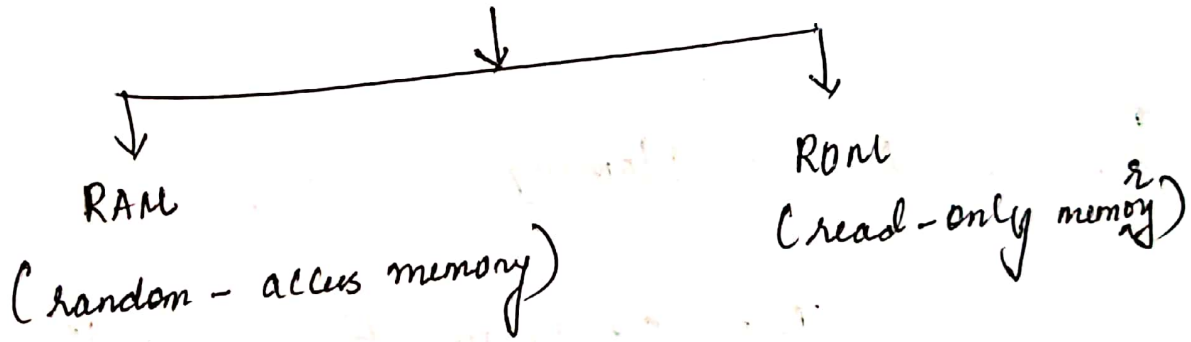
Vinay Reddy Narayana

Memory & Programmable Logic

Memory

A memory unit is a device to which binary information is transferred for storage and from which information is retrieved when needed for processing.

Two types of Memories



Memory write operation:

The process of storing new information into memory is referred to as a memory write operation.

Memory Read operation:

The process of transferring the stored information out of memory is referd to as a memory read operation.

* RAM can perform both write and read operation
* ROM can perform only read operation
  (The ROM is programmed only once)

RANDOM ACCESS MEMORY

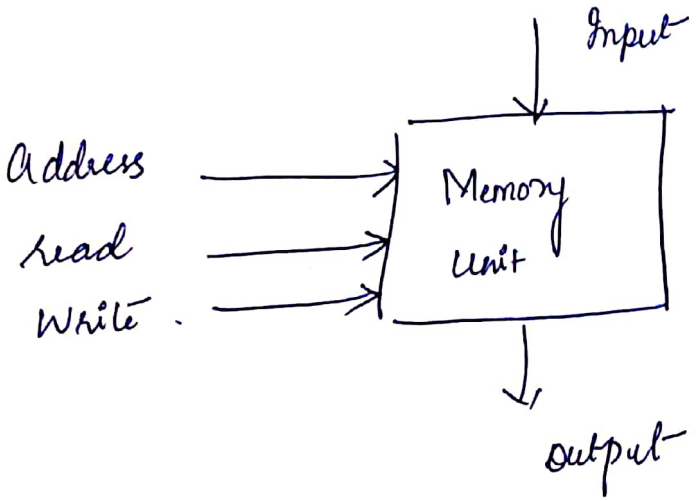
The architecture of memory is such that information can be selectively retrieved from any of its internal locations. The time it takes to transfer information to or from any desired random location is always the same - hence the name random-access memory.

Words & Bytes.

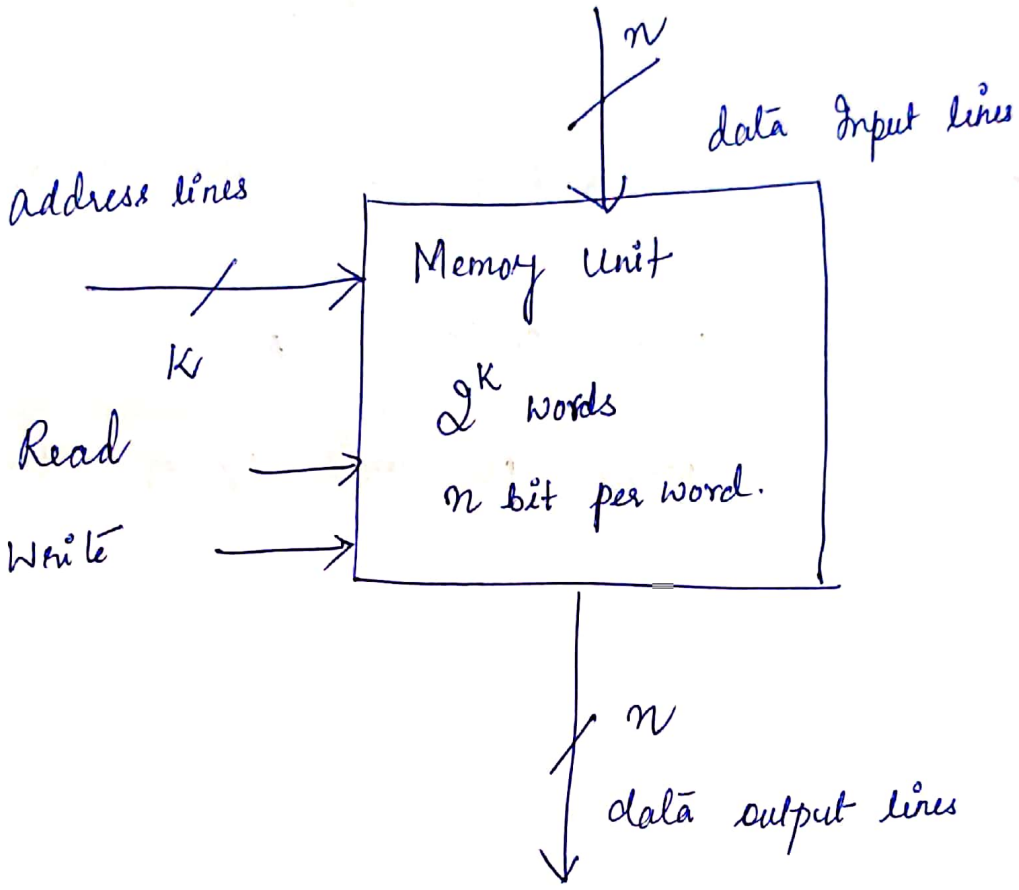
* A memory unit stores binary information in group of bits called words.
* A group of 8 bits is called a byte.
* A 16 bit word contains two bytes.
* A 32 bit word is made up of four bytes.

The capacity of a memory unit is usually stated as the total number of bytes that the unit can store.

Memory unit.



Block Diagram of a Memory Unit :



* Each word in the memory is assigned an identification number, called an address starting from 0 up to $2^K - 1$.

Eg: 1

* If the memory unit has 10 bit address line, how many locations can it have.?

$\Longrightarrow$ 2^{10} locations = 1024 locations.

* If a memory unit has 10 bit address line, with each address location capable of storing 1 byte (8 bits), what is the capacity of the memory

$\Longrightarrow$ The capacity of the memory $= 2^{10}$ bytes

$= 1024$ bytes

$= 1K$ bytes

$= 1K$ bytes of memory.

$1K \approx 1024$.

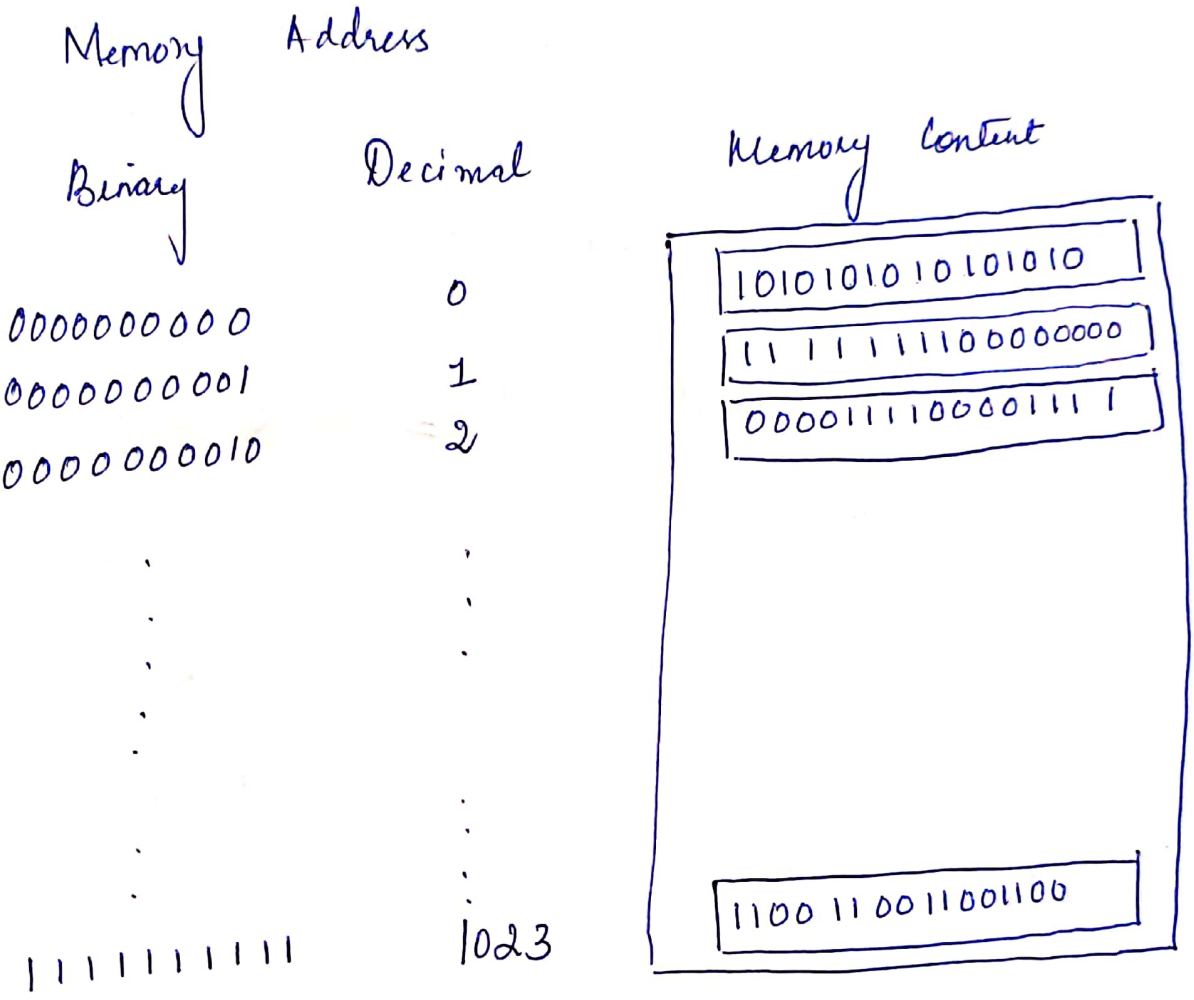
Eg: 2

A 64K x 10 memory will have how many address lines and how many bits in each address location ?

* $64K = 2^{16}$ $\longrightarrow$ Hence 16 bit address line.
* Each address line will have 10 bits.

An example of 1K x 16 memory is shown below —

* It has 10 bits in the address and 16 bits in each word.



Contents of a 1024 x 16 memory

The number of bits in the address is determined from the relationship $2^k \geq m$, where m is the total number of words and k is the number of address bits needed to satisfy the relationship.

Eg

Consider $k = 10$ and $m = 1024$

$2^{10} \geq 1024$ word $\left\{ \text{word} \leq \frac{1024}{1024} = 1 \right.$

$1024 \geq 1024$ word

1 word = 16 bits

↳ Each address location will have 16 bits

Write & Read operations

The two operations that RAM can perform are the write and read operations.

The steps that must be taken for the purpose of transferring a new word to be stored into memory → Write operation.

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

2. Apply the data bits that must be stored in memory to the data input lines.

3. 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.

The steps for transferring a stored word out of memory → Read operation

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

2. Activate the read input.

Control Inputs to Memory Chip.

Memory Enable	Read/write	Memory Operation.
0	X	None.
1	0	Write operation
1	1	Read operation.

Timing waveforms:

(a) Memory write cycle
(b) Memory read cycle

* Assume a CPU operates with a clock frequency of 50 MHz, giving a period of 20 ns for one clock cycle.

$$f = 50\ MHz$$

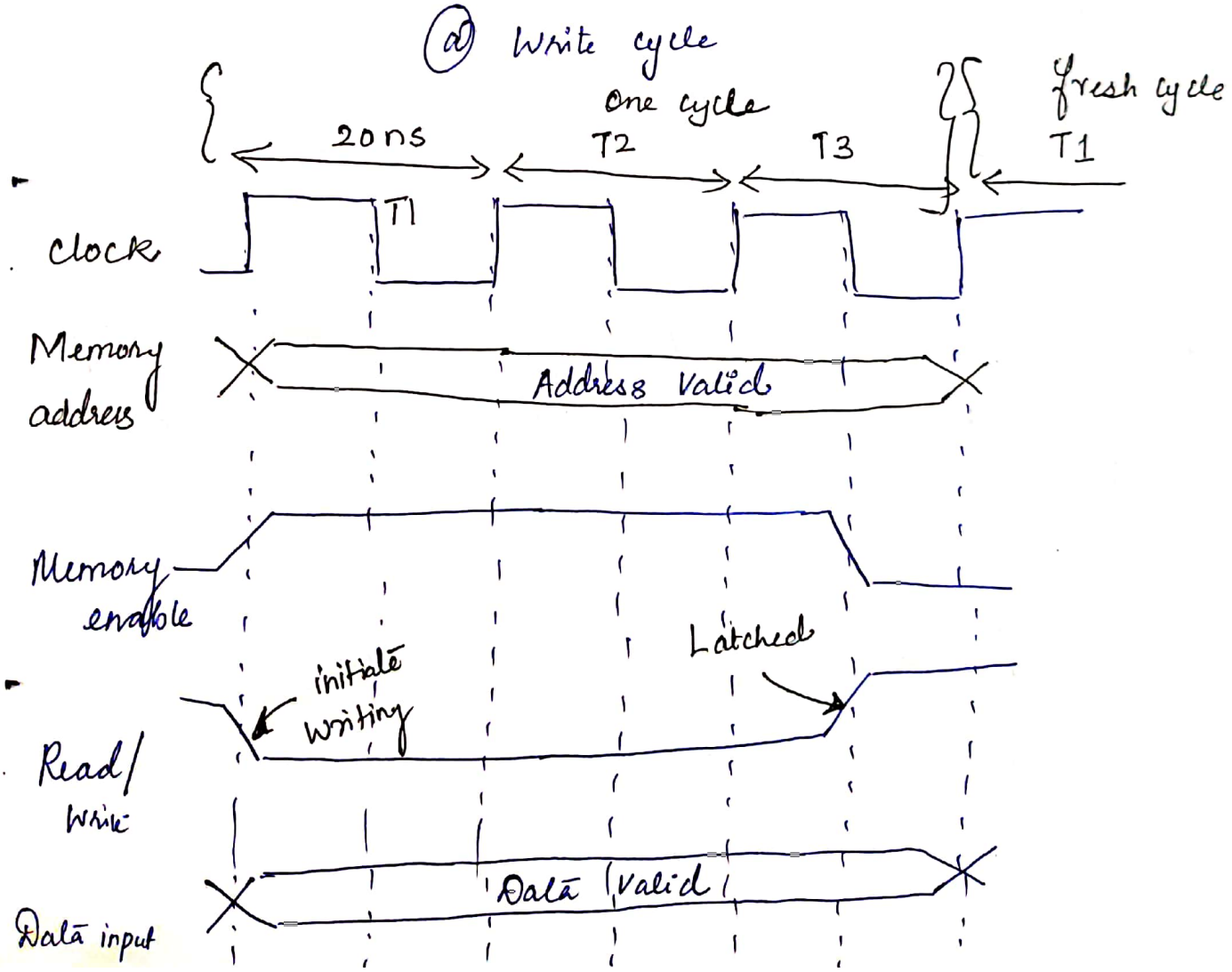
$$T = \frac{1}{50} = 20\ ns$$

Assumption:

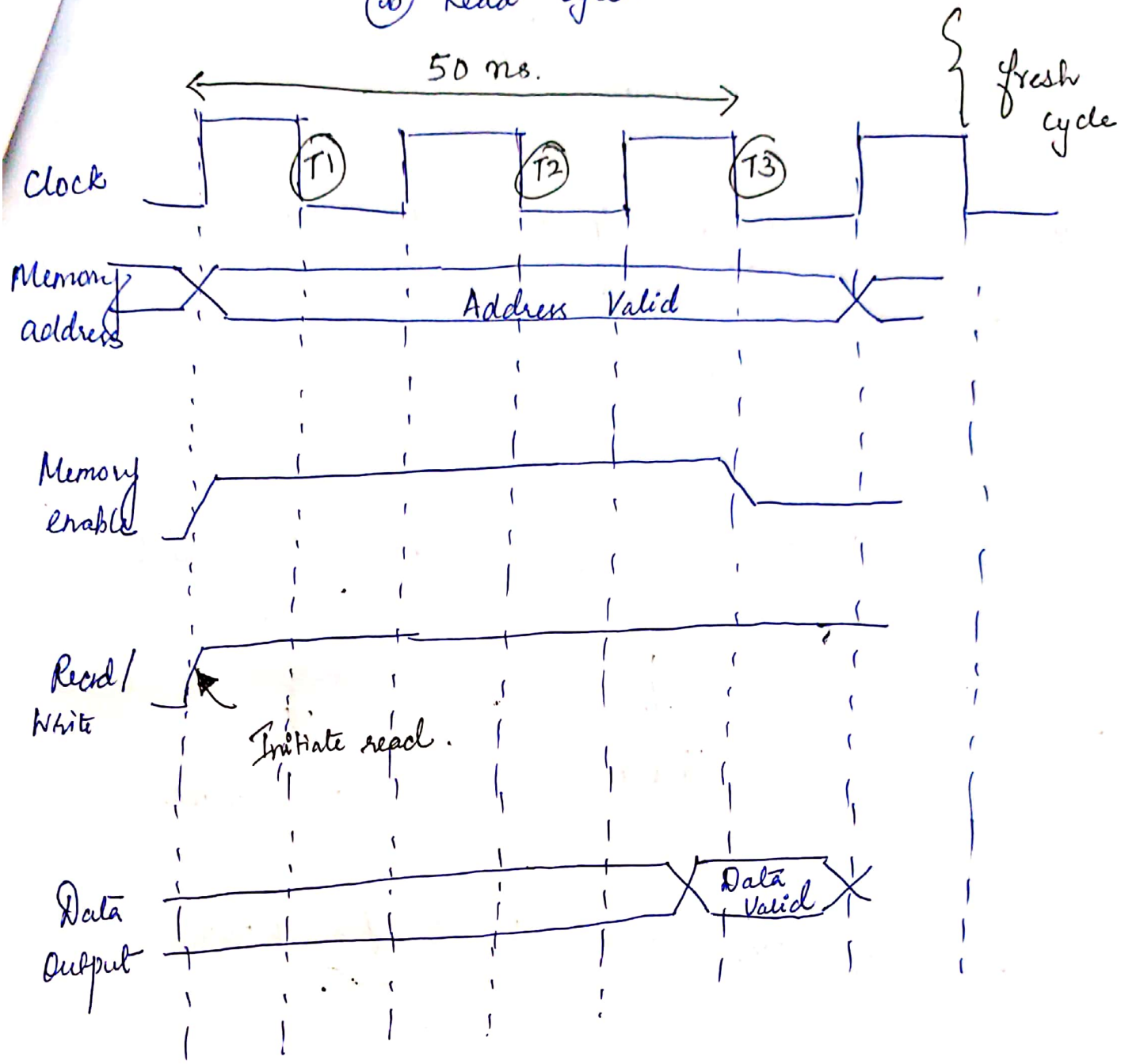
The CPU Communicates with a memory whose access time and cycle time do not exceed 50 ns.

Meaning

The write cycle terminates the storage of the selected word within 50-ns interval and the read cycle provides the output data of the selected word within 50 ns or less.



(b) Read cycle.



Memory Description in HDL

Memory is modeled in the Verilog hardware description language (HDL) by an array of registers.

* It is declared with a keyword 'reg', using a two-dimentional array.

* The first number specifies the number of bits in the word. and the second number specifies the number of words in the memory

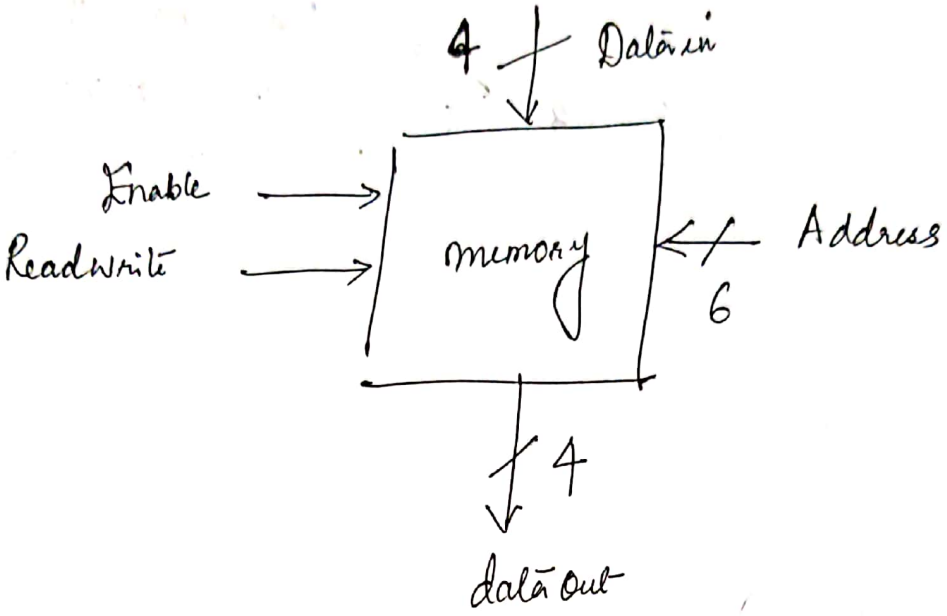
Eg: reg [15:0] memword [0:1023];

[15:0] → 16 bit

[0:1023] → 2^{10} locations.

=> 1,024 words with 16 bits per word.

HDL Model for a memory given below —.



Solution

Address = 6 bit

$2^6 = 64$ (locations)

Each location will have 4 bits.

Memory size is 64 words of four bits each.

Verilog Code (Behavioural Style)

```
module memory ( Enable, Readwrite, Data in, Data out,
                Address);

input  Enable, Readwrite;
input  [3:0] Data in;
output [3:0] Data out;
```

```
input [5:0] Address;
reg [3:0] dataout;
reg [3:0] mem [0:63];
```

Describing the memory in verilog.

```
always@(Enable or Readwrite)
  if (Enable)
    if (Readwrite)
      Dataout = mem[Address]; // Read
    else
      mem[Address] = Data_in; // Write
  else
    Dataout = 4'bz; // High Impedance State
endmodule
```

Error Detection & Correction.

* The dynamic physical interaction of the electrical signals affecting the data path of a memory unit may cause occasional errors in storing and retrieving the binary information.

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

* The most common error detection scheme is the parity bit.

Error Detection :

A parity bit is generated and stored along with the data word in the memory. The parity of the word is checked after reading it from memory. The data word is accepted if the parity of the bits read out is correct, else an error is detected.

Error Correction

⇒ When the word is read back from memory, the associated parity bits are also read from memory and compared with a new set of check bits generated from the data that have been read.

If the check bits are correct, no error has occured. If the check bits do not match the stored parity, they generate a unique pattern, called a Syndrome, that can be used to identify the bit that is in error

* One of the most common error-correcting codes used in RAM's is the Hamming code.

⇒ In the Hamming code, k parity bits are added to an n-bit data word to form a new word of n+k bits.

CORRECTION & DETECTION

Data = 11000100.

Vinay Reddy

Bit - positions

1	2	3	4	5	6	7	8	9	10	11	12
P1	P2	1	P4	1	0	0	P8	0	1	0	0

pg-9

The four parity bits are P1, P2, P4, P8

$2^3 \to 8$, $2^2 \to 4$, $2^1 \to 2$, $2^0 \to 1$

0 → 0 0 0 0
1 → 0 0 0 1
2 → 0 0 1 0
3 → 0 0 1 1
4 → 0 1 0 0
5 → 0 1 0 1
6 → 0 1 1 0
7 → 0 1 1 1
8 → 1 0 0 0
9 → 1 0 0 1
10 → 1 0 1 0
11 → 1 0 1 1
12 → 1 1 0 0
13 → 1 1 0 1
14 → 1 1 1 0
15 → 1 1 1 1

To calculate $P_1 = 3 \oplus 5 \oplus 7 \oplus 9 \oplus 11 = 0$

To calculate $P_2 = 3 \oplus 6 \oplus 7 \oplus 10 \oplus 11 = 0$

To calculate $P_4 = 5 \oplus 6 \oplus 7 \oplus 12 = 1$

To calculate $P_8 = 9 \oplus 10 \oplus 11 \oplus 12 = 1$

The 8-bit data word is stored in the memory together with the 4-parity bits as a 12 bit composite word.

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

1.(a). Pg no 32 - 33.

1.(b) Pg no 10, 11 and 12.

2.(a) → Pg no - 49.

2.(b) → Pg no - 43.

2.(c) → Pg no 52.

We have to Calculate the Check bits

C_1 = XOR of bits (1, 3, 5, 7, 9, 11) = 0

C_2 = XOR of bits (2, 3, 6, 7, 10, 11) = 0

C_4 = XOR of bits (4, 5, 6, 7, 12) = 0

C_8 = XOR of bits (8, 9, 10, 11, 12) = 0

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

	C_8	C_4	C_2	C_1	
For no error :	0	0	0	0	Syndrome.
For error with bit 1 :	0	0	0	1	
For error with bit 5 :	0	1	0	1	