G O K A R A J U R A N G A R A J U

Institute of Engineering and Technology

(Autonomous Institute under JNTU Hyderabad)

II B Tech I Semester – 2021-22 – GR20 Regulation

DIGITAL LOGIC DESIGN

Answers to Problem Sheet 5

Unit V (Memory and Programmable Logic & Hardware Description Language)

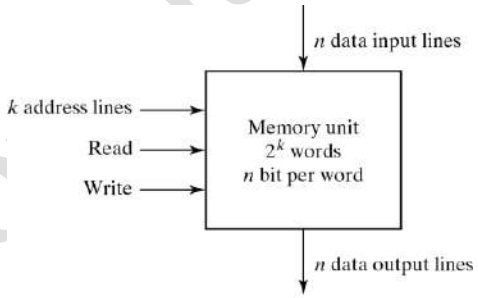
Syllabus: **Memory and Programmable Logic**: Introduction, Random Access Memory, Memory Decoding, Error Detection and correction, Read Only Memory, Programmable logic Array, Programmable Array Logic, Sequential Programmable Devices.
Hardware Description Language: Hardware Description Language, Definition, Structural Definition of HDL, HDL Models for Combinational circuits, HDL for Models for Sequential circuits.

---

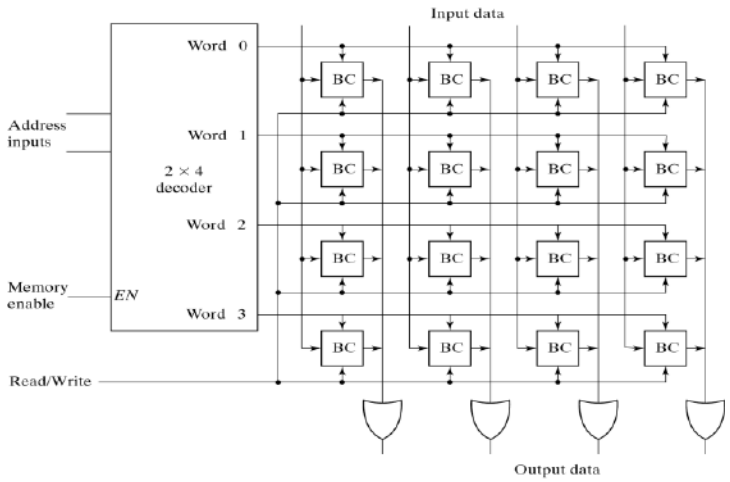
1. Draw the block diagram of a RAM unit, and explain the process of memory decoding in RAM.

Answer: A RAM memory unit is a collection of storage cells together with associated circuits needed to transfer information in and out of the device.

Block Diagram of RAM:



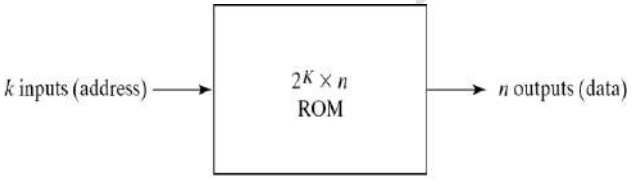
Memory Decoding in RAM: In addition to the storage components in a memory unit, there is a need for decoding circuits to select the memory word specified by the input address. The internal construction of a random access memory of m words and n bits per word consists of mXn binary storage cells and associated decoding circuits for selecting individual words. The binary storage cell is the basic building block of the memory unit.



2. Draw the block diagram of ROM.

Answer: A ROM is essentially a memory device in which permanent binary information is stored. The binary information must be specified by the designer and is then embedded in the unit to form the required interconnection pattern. Once the pattern is established, it stays within the unit even when power is turned off and on again.

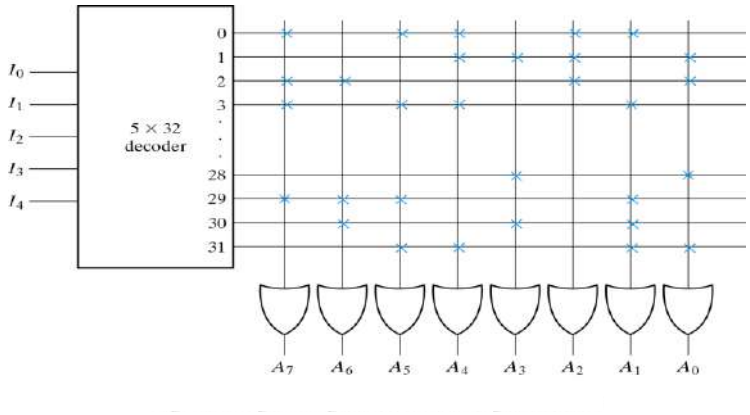
Block Diagram of ROM:



3. Explain the process of Boolean function implementation using ROM

Answer: The internal operation of a ROM can be interpreted in two ways. The first interpretation is that of a memory unit that contains a fixed pattern of stored words. The second interpretation is of a unit that implements a combinational circuit. From this point of view, each output terminal is considered separately as the output of a Boolean function expressed as a sum of minterms.

For example,

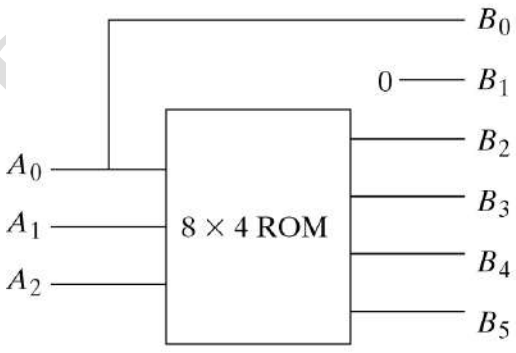


- A7(i4,i3,i2,i1,i0)=(0,2,3,…,29)

4. Design a combinational circuit using a ROM that accepts a 3 bit number and generates an output binary number equal to the square of the input number.

Answer:

Inputs			Outputs						decimal
a2	a1	a0	b5	b4	b3	b2	b1	b0	
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	1
0	1	0	0	0	0	1	0	0	4
0	1	1	0	0	1	0	0	1	9
1	0	0	0	1	0	0	0	0	16
1	0	1	0	1	1	0	0	1	25
1	1	0	1	0	0	1	0	0	36
1	1	1	1	1	0	0	0	1	49



(a) Block diagram

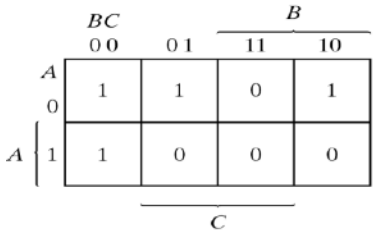
A_2	A_1	A_0	B_5	B_4	B_3	B_2
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	1
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	0	1	1	0
1	1	0	1	0	0	1
1	1	1	1	1	0	0

(b) ROM truth table

5. Define a PLA. Implement the following two boolean functions with a PLA:

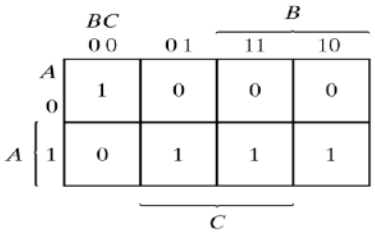
F1(A,B,C)=∑(0,1,2,4) F2(A,B,C)=∑(0,5,6,7)

Answer: The programmable logic array is similar to the PROM in concept except that the PLA does not provide full decoding of the variables and does not generate all the minterms. The decoder is replaced by an array of AND gates that can be programmed to generate any product term of the input variables. The product terms are the connected to the OR gates to provide the sum of products for the required boolean functions.



$F_1 = A'B' + A'C' + B'C'$

$F_1 = (AB + AC + BC)'$

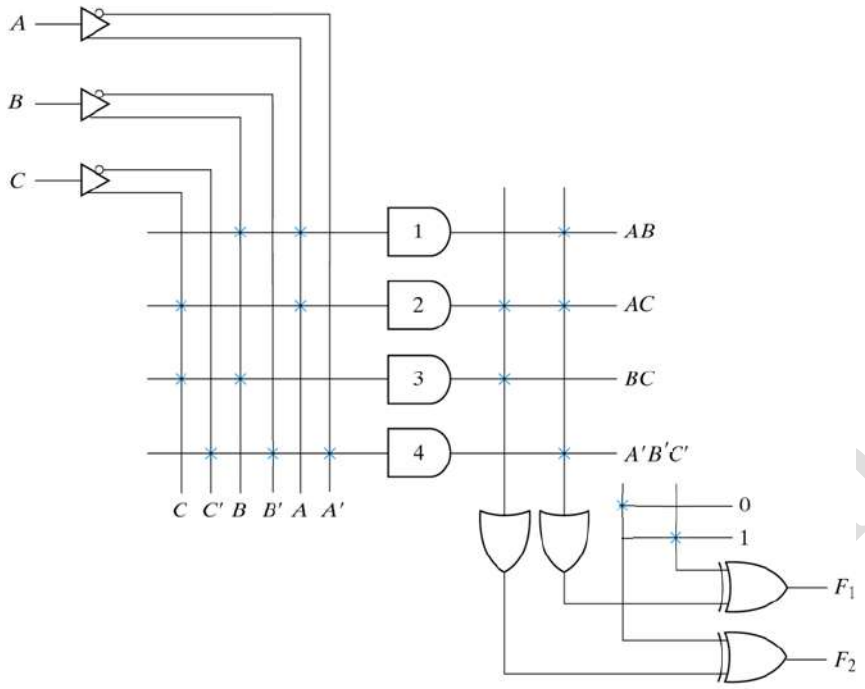


$F_2 = AB + AC + A'B'C'$

$F_2 = (A'C + A'B + AB'C')'$

PLA programming table

	Product term	Inputs A	B	C	Outputs (C) F_1	(T) F_2
AB	1	1	1	–	1	1
AC	2	1	–	1	1	1
BC	3	–	1	1	1	–
$A'B'C'$	4	0	0	0	–	1



6. Define a PAL. Implement the following Boolean functions with PAL.

   w(A,B,C,D) =∑(2,12,13) x(A,B,C,D)= ∑(7,8,9,10,11,12,13,14,15)

   y(A,B,C,D)= ∑(0,2,34,5,6,7,8,10,11,15) z(A,B,C,D)= ∑(1,2,8,12,13)

Answer: The programmable logic is a programmable logic device with a fixed OR array and a programmable AND array. Because only the AND gates are programmable, the PAL is easier to program, but is not as flexible as the PLA.

The simplified form of the given Boolean expressions are:

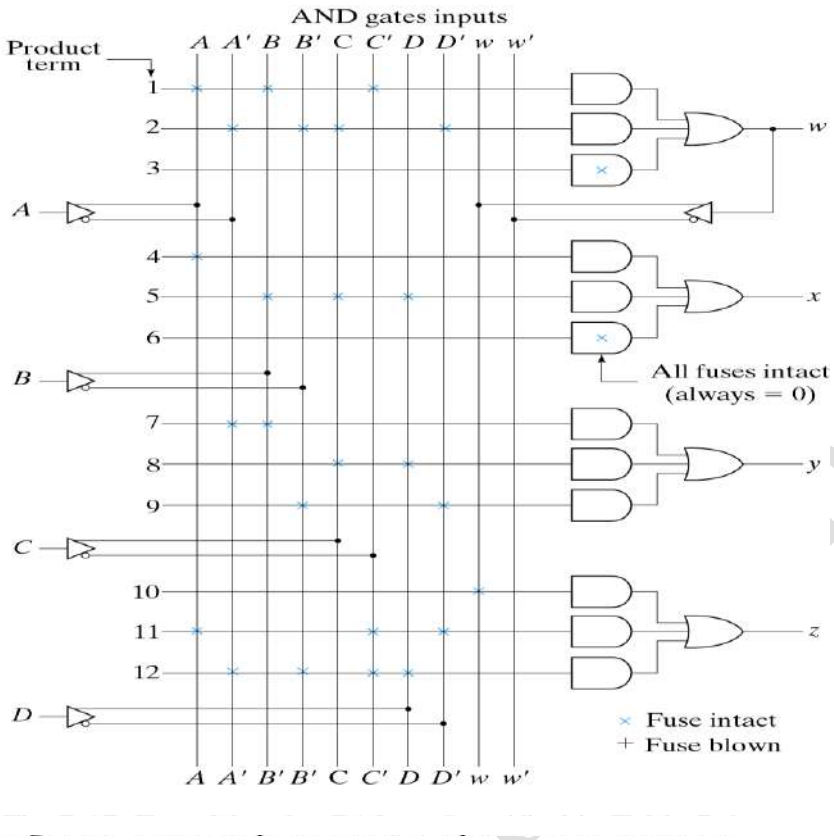
$w=ABC^1+A^1B^1CD^1$

X=A+BCD

$Y=A^1B+CD+B^1D^1$

$Z=ABC^1+A^1B^1CD^1+AC^1D^1+A^1B^1C^1D$

$=W+AC^1D^1+A^1B^1C^1D$



7. Define HDL. Write a HDL routine to implement the following boolean functions:

$x = A+BC+B^1D$ $y=B^1C+BC^1D^1$

Answer: A hardware description language is a language that describes the hardware of digital system in a textual form. It resembles a programming language, but is specifically oriented to describing hardware structures and behavior. As a documentation language, HDL is used to represent and document digital systems in a form that can be read by both human and computers and is suitable as an exchange language between designers.

```
// Circuit specified with Boolean expression
Module circuit_bin(x,y,A,B,C,D);
  input A,B,C,D;
  output x,y;
  assign x=A | (B & C) | (~B & D)
  assign y=(~B & C) | (B & ~C & ~D)
endmodule
```

8. Write a Data flow description In HDL for 2-to-4 Line Decoder.

Answer:

```
//Dataflow Description of a 2-to4 Line Decoder
```

```
module  decoder_df(A,B,E,D);
input       A,B,E;
output      [0:3] D;
assign      D[0]=~(~A & ~B & ~E);
            D[1]=~(~A & B & ~E);
            D[2]=~(A & ~B & ~E);
            D[3]=~(A & B & ~E);
Endmodule
```

9. Write Behavioral Description of Universal Shift Register in HDL.

Answer:

```
// Behavioral Description of Universal Shift Register
Module   shftreg(s1,s0,pin,lfin,rtin,A,clk, Clr);
Input    s1,s0;
Input    lfin,rtin;
Input    clk,Clr;
Input    [3:0] pin;
Output   [3:0] A;
Reg      [3:0] A;
Always   @(posedge clk or negedge Clr)
         If(~clr) A=4'b0000;
         Case({s1,s0})
         2'b00:  A=A;
         2'b00:  A={rtin,A[3:1]};
         2'b00:  A={A[2:0],lfin};
         2'b11:  A=Pin;
         Endcase
endmodule
```

Mode Control

S1	S0	Register Operation
0	0	No change
0	1	Shift right
1	0	Shift left
1	1	Parallel load

10. Write Hierarchial Descrption of 4-bit adder in HDL.

Answer:

```
//Hierarchial description of 4-bit Adder
    Module Halfadder(S,C,x,y);
        Input   x,y;
       Output   S,C;
//Instantiate Primitive gates
          Xor   (S,x,y);
          And   (C,x,y);
    Endmodule
//description of Full Adder
    Module fulladder(S,C,x,y,z);
        Input   x,y,z;
       Ouput    S,C;
         Wire   S1,D1,D2;
    Halfadder   HA1(S1,D1,x,y);
                HA2(S,D2,S1,z);
           Or   g1(C,D2,D1);
    endmodule
```

```
//description of 4 bit adder
Module _4bitadder(S,C4,A,B,C0);
       Input   [3:0] A,B;
       input   C0;
      output   [3:0] S;
      output   C4;
        wire   C1,C2,C3;
    fulladder  FA0(S[0],C1,A[0],B[0],C0);
               FA1(S[1],C2,A[1],B[1],C1);
               FA2(S[2],C3,A[2],B[2],C2);
               FA3(S[3],C4,A[3],B[3],C3);
endmodule
```

11. What is an UDP? Write rules for writing an UDP.

Answer: The logic gates used in HDL descriptions with keywords and, or etc., are defined by the system and are referred to as system primitives. The user can create additional primitives by defining them in a tabular form. These types of circuits are referred to as user defined primitives. The user defined primitives are declared with the keyword primitive.

Rules for writing UDP:

- It is declared with the keyword primitive followed by a name and port list.
- There can be only one output and it must be listed first in the port list and declared with an output keyword.
- There can be any number of inputs. The order in which they are listed in the input declaration must conform to the order in which they are given values in the table that follows.
- The truth table is enclosed within the keywords table and endtable.
- The values of the inputs are listed in order ending with a colon(:). The output is always the last entry in a row followed by a semicolon(;).
- It ends with the keyword endprimitive.