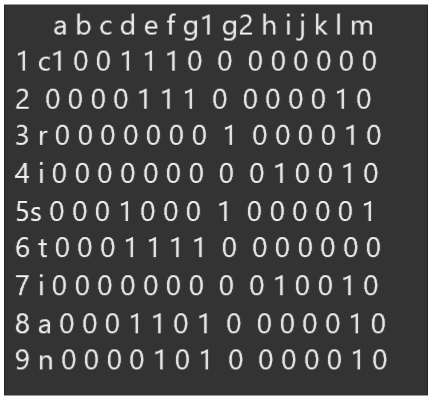
Lab#5
14 segment display with a logic gate decoder

Digital design principles.

Teacher:
Rogelio Hernández Hernández
Work made by:
Christian Aaron Ortega Blanco

Hermosillo, Son. 5 de mayo de 2022.

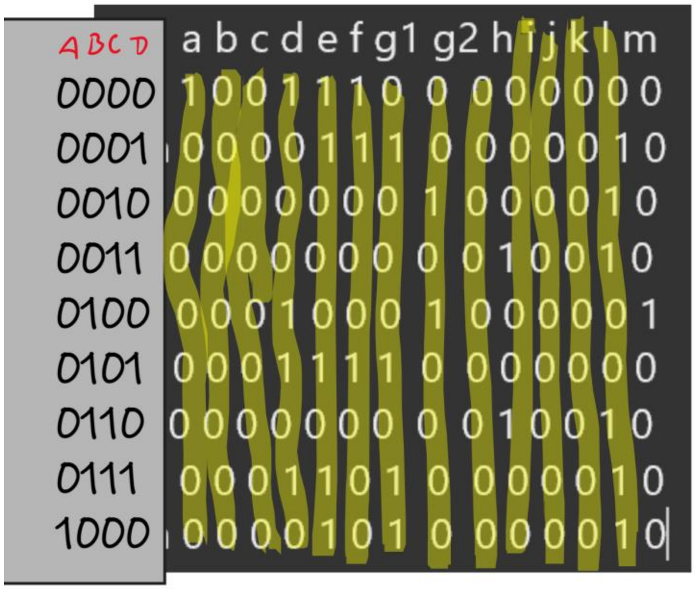
The following table represent the characters of the name “Christian” in a 14 bit format;



		a	b	c	d	e	f	g1	g2	h	i	j	k	l	m
1	c	1	0	0	1	1	1	0	0	0	0	0	0	0	0
2		0	0	0	0	1	1	1	0	0	0	0	0	1	0
3	r	0	0	0	0	0	0	0	1	0	0	0	0	1	0
4	i	0	0	0	0	0	0	0	0	0	1	0	0	1	0
5	s	0	0	0	1	0	0	0	1	0	0	0	0	0	1
6	t	0	0	0	1	1	1	1	0	0	0	0	0	0	0
7	i	0	0	0	0	0	0	0	0	0	1	0	0	1	0
8	a	0	0	0	1	1	0	1	0	0	0	0	0	1	0
9	n	0	0	0	0	1	0	1	0	0	0	0	0	1	0

Table 1 character representation

To make a decoder that uses only 4 bits, where “0000” represents the first character of the word (“C”) and “1000” is the 9th character (“n”), then the following table is created.



ABCD	a	b	c	d	e	f	g1	g2	h	i	j	k	l	m
0000	1	0	0	1	1	1	0	0	0	0	0	0	0	0
0001	0	0	0	0	1	1	1	0	0	0	0	0	1	0
0010	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0011	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0100	0	0	0	1	0	0	0	1	0	0	0	0	0	1
0101	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0110	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0111	0	0	0	1	1	0	1	0	0	0	0	0	1	0
1000	0	0	0	0	1	0	1	0	0	0	0	0	1	0

Table 2 decoder table

If every column of the table is indexed as a function of 9 bits, the circuit form can be easily obtained using Karnaugh maps to reduce the expression, as showed on the next image.

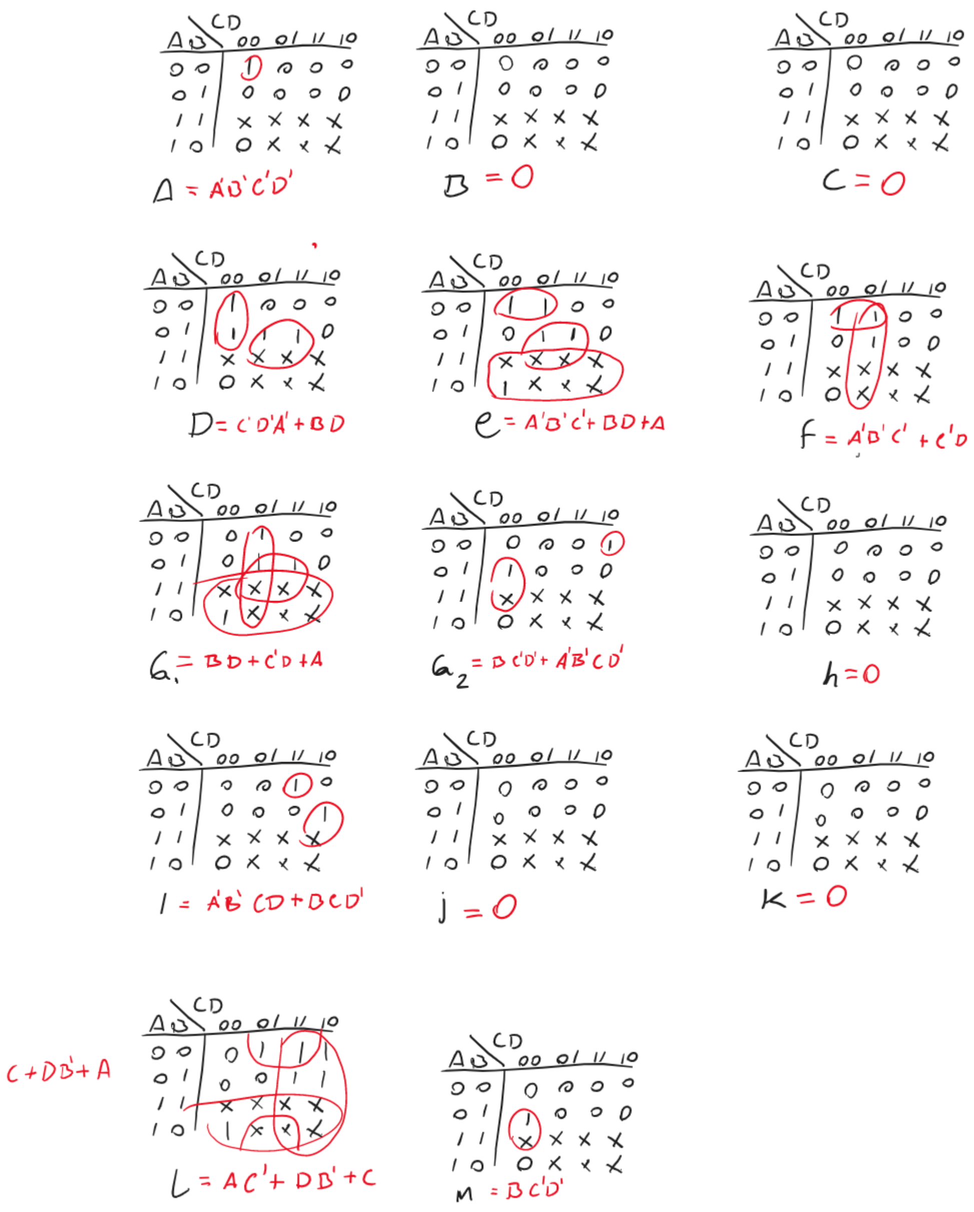


Figure 1 Karnaugh map´s

Now it’s possible to beginning the scrip using the obtained expressions. As it is need it to be structural programing form, every function has is own module definition and connected as if it were a physical circuit.

Module definition

```
module
decoder(A,B,C,D,s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13
);
  input A,B,C,D;
  output s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13;
  m_A u1(A,B,C,D,s0);
  m_B u2(A,B,C,D,s1);
  m_C u3(A,B,C,D,s2);
  m_D u4(A,B,C,D,s3);
  m_E u5(A,B,C,D,s4);
  m_F u6(A,B,C,D,s5);
  m_G_1 u7(A,B,C,D,s6);
  m_G_2 u8(A,B,C,D,s7);
  m_H u9(A,B,C,D,s8);
  m_I u10(A,B,C,D,s9);
  m_J u11(A,B,C,D,s10);
  m_K u12(A,B,C,D,s11);
  m_L u13(A,B,C,D,s12);
  m_M u14(A,B,C,D,s13);
endmodule
```

Function Module example

```
module m_A(A,B,C,D,s);
  input A,B,C,D;
  output s;
  wire n1,n2,n3,n4,n5,n6,n7;
  Not_gate u1(A,n1);
  Not_gate u2(B,n2);
  Not_gate u3(C,n3);
  Not_gate u4(D,n4);

  And_gate u5(n1,n2,n3,n5);
  And_gate_2 u6(n5,n4,s);

endmodule
```

Bench test

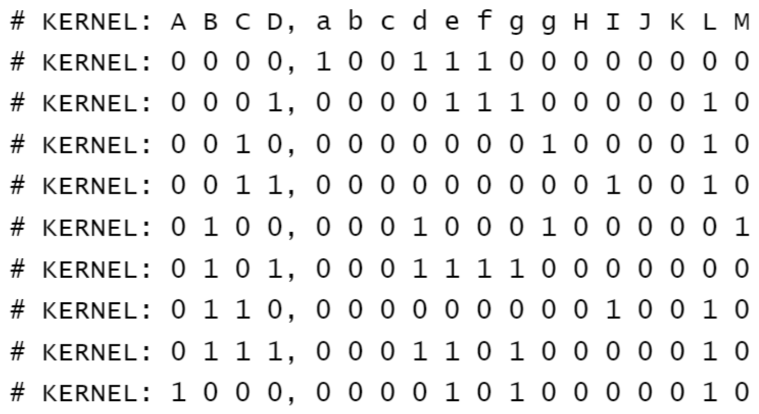
```
`timescale 1ns/1ps
`include "Not_gate.sv"
`include "And_gate.sv"
`include "Or_gate.sv"
`include "And_gate_2.sv"
`include "Or_gate_2.sv"
`include "m_A.sv"
`include "m_B.sv"
`include "m_C.sv"
`include "m_D.sv"
`include "m_E.sv"
`include "m_F.sv"
`include "m_G_1.sv"
`include "m_G_2.sv"
`include "m_H.sv"
`include "m_I.sv"
`include "m_J.sv"
`include "m_K.sv"
`include "m_L.sv"
`include "m_M.sv"
module decoder_TB;
  reg A,B,C,D;
  wire s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13;
  int i;

  decoder
UUT(A,B,C,D,s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13);

  initial
    begin
      $dumpfile("decoder.vcd");
      $dumpvars(1,decoder_TB);
      $display("A B C D, a b c d e f g g H I J K L M");
      $monitor ("%0b %0b %0b %0b, %0b %0b %0b %0b %0b
%0b   %0b   %0b   %0b   %0b   %0b   %0b   %0b   %0b
",A,B,C,D,s0,s1,s2,s3,s4,s5,s6,s7,s8,s9,s10,s11,s12,s13);
      for(i=0;i<9;i++)begin
        {A,B,C,D}=i;
        #1;
      end
      $finish;
    end
  endmodule
```

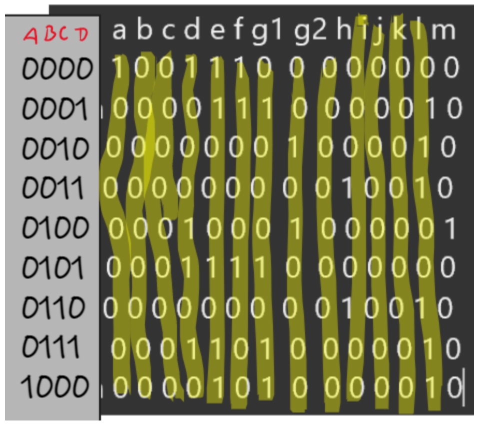
Results

At the end the design does what is meant to be, with a 4-bit array we can select every of the 9 characters. that conform the word “Christian”, this behavior is characteristic of a decoder circuit.



```
# KERNEL: A B C D, a b c d e f g g H I J K L M
# KERNEL: 0 0 0 0, 1 0 0 1 1 1 0 0 0 0 0 0 0 0
# KERNEL: 0 0 0 1, 0 0 0 0 1 1 1 0 0 0 0 0 1 0
# KERNEL: 0 0 1 0, 0 0 0 0 0 0 0 1 0 0 0 0 1 0
# KERNEL: 0 0 1 1, 0 0 0 0 0 0 0 0 1 0 0 1 0
# KERNEL: 0 1 0 0, 0 0 0 1 0 0 0 1 0 0 0 0 0 1
# KERNEL: 0 1 0 1, 0 0 0 1 1 1 1 0 0 0 0 0 0 0
# KERNEL: 0 1 1 0, 0 0 0 0 0 0 0 0 0 1 0 0 1 0
# KERNEL: 0 1 1 1, 0 0 0 1 1 0 1 0 0 0 0 0 1 0
# KERNEL: 1 0 0 0, 0 0 0 0 1 0 1 0 0 0 0 0 1 0
```

Figure 2 Resulted decoder



ABCD	a	b	c	d	e	f	g1	g2	h	i	j	k	l	m
0000	1	0	0	1	1	1	0	0	0	0	0	0	0	0
0001	0	0	0	0	1	1	1	0	0	0	0	0	1	0
0010	0	0	0	0	0	0	0	1	0	0	0	0	1	0
0011	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0100	0	0	0	1	0	0	0	1	0	0	0	0	0	1
0101	0	0	0	1	1	1	1	0	0	0	0	0	0	0
0110	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0111	0	0	0	1	1	0	1	0	0	0	0	0	1	0
1000	0	0	0	0	1	0	1	0	0	0	0	0	1	0

Figure 3 characters to conform the word "Christian"

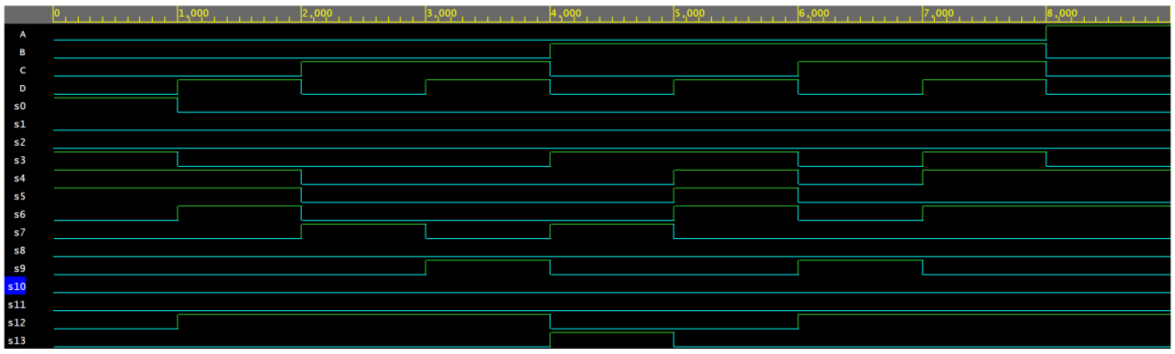


Figure 4 Input-Output signal graph