



IE0323

Sistemas Digitales I

Laboratorio #5

Diagramas Esquemáticos

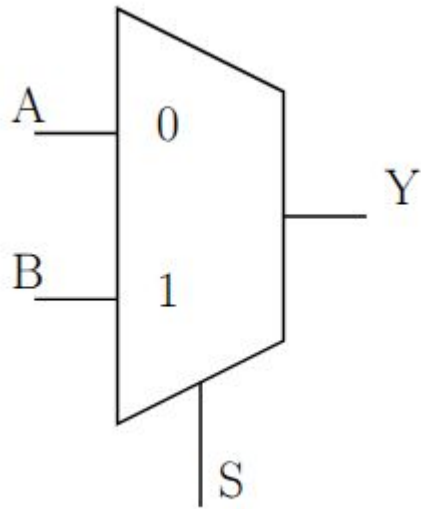


Objetivos

- Comprender el funcionamiento básico de los multiplexores y demultiplexores.
- Implementar multiplexores de mayor capacidad utilizando configuraciones en árbol.
Diseñar circuitos digitales combinacionales utilizando Verilog conductual.
- Programar y probar el comportamiento de los circuitos diseñados en hardware mediante una FPGA.

Multiplexores

Selecciona cuál entrada de datos se propaga a la salida



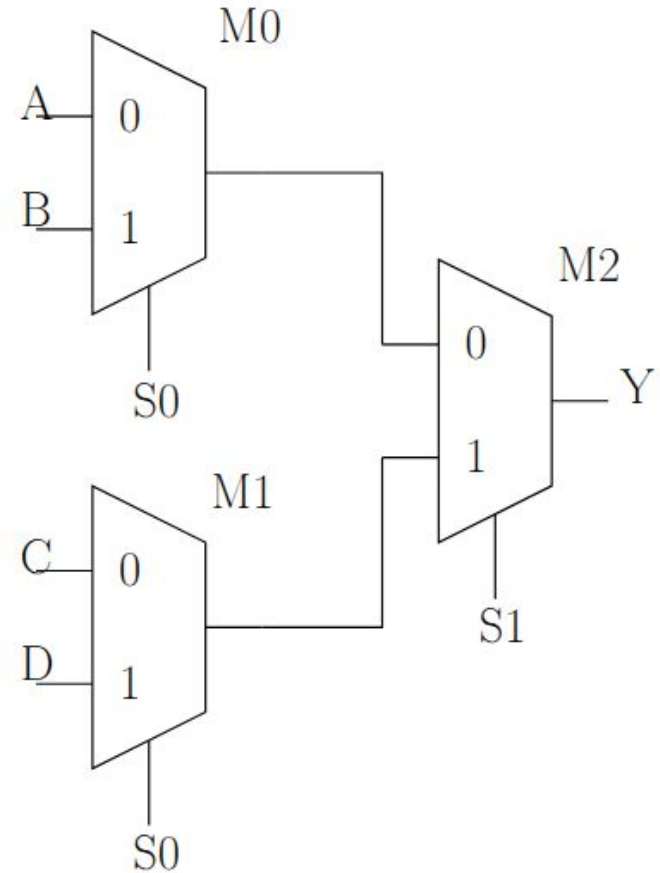
S	A	B	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

S	A	B	Y
0	0	X	0
0	1	X	1
1	X	0	0
1	X	0	0

`assign Y = (S==0)?A:B;`

Árboles de Multiplexores

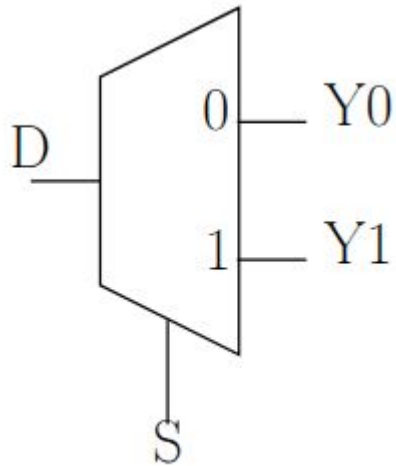
S1	S0	A	B	C	D	Y
0	0	0	X	X	X	0
0	0	1	X	X	X	1
0	1	X	0	X	X	0
0	1	X	1	X	X	1
1	0	X	X	0	X	0
1	0	X	X	1	X	1
1	1	X	X	X	0	0
1	1	X	X	X	1	1





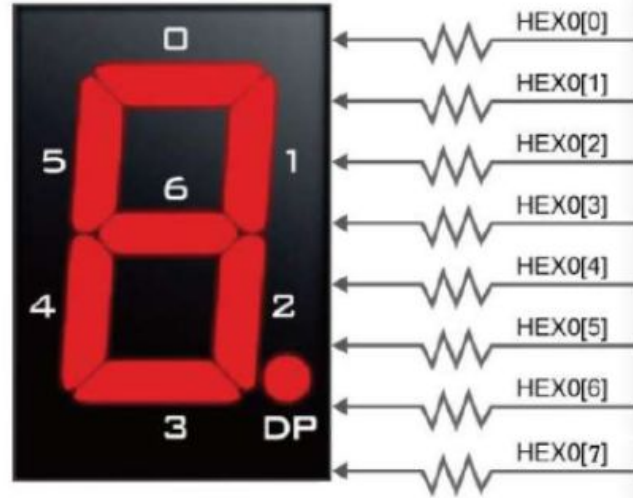
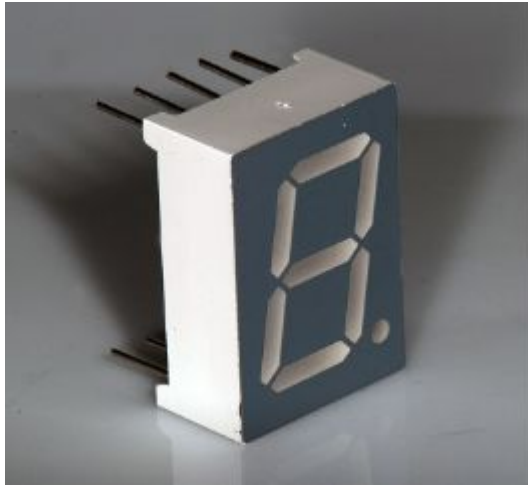
Demultiplexor

Selecciona **POR CUAL SALIDA** se propaga el dato de la entrada



S	D	Y0	Y1
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

Display de 7 segmentos



Señales activas en bajo == Poner en 0 para encender



Pines del display de 7 segmentos

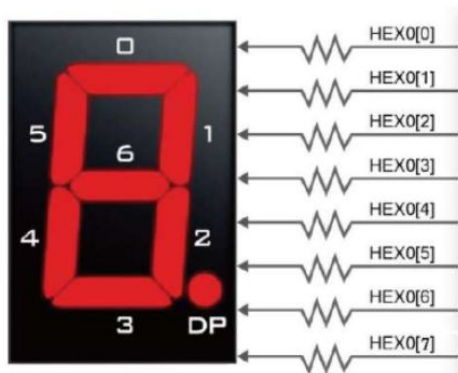
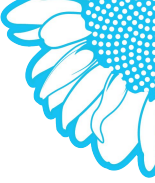


Table 3-6 Pin Assignment of 7-segment Displays

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTTL
HEX01	PIN_E15	Seven Segment Digit 0[1]	3.3-V LVTTTL
HEX02	PIN_C15	Seven Segment Digit 0[2]	3.3-V LVTTTL
HEX03	PIN_C16	Seven Segment Digit 0[3]	3.3-V LVTTTL
HEX04	PIN_E16	Seven Segment Digit 0[4]	3.3-V LVTTTL
HEX05	PIN_D17	Seven Segment Digit 0[5]	3.3-V LVTTTL
HEX06	PIN_C17	Seven Segment Digit 0[6]	3.3-V LVTTTL
HEX07	PIN_D15	Seven Segment Digit 0[7], DP	3.3-V LVTTTL



Ejemplo Introductorio – Mux 2:1

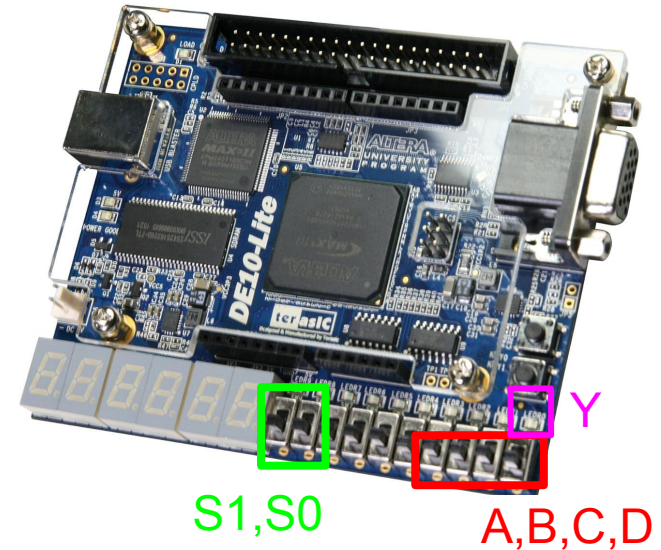
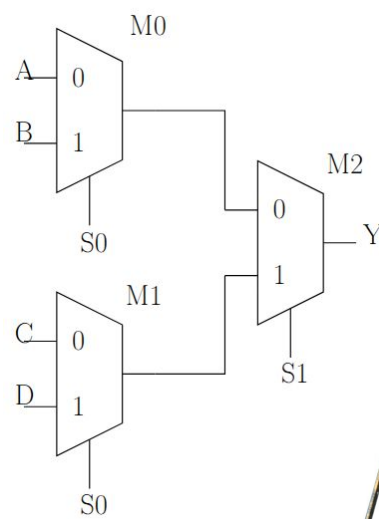
```
mux_2x1.v

1  //Declaracion del modulo mux2x1
2  module mux_2x1 (A, B, S, Y);
3      //Declaracion de puertos
4      input A, B, S;
5      output reg Y;
6
7      //Cada vez que alguna entreada cambie
8      always @(*) begin
9          //Si S=0 la salida es A
10         if (S == 0) begin
11             Y = A;
12         end
13         //Caso contraro es B
14         else begin
15             Y = B;
16         end
17     //Termina always
18     end
19 //Termina modulo
20 endmodule
```


Trabajo en clase 1/3

Utilizando un árbol de multiplexores, implemente en Verilog un módulo nuevo que funcione como un multiplexor 4x1. Este módulo deberá contener múltiples instancias del módulo mux_2x1.

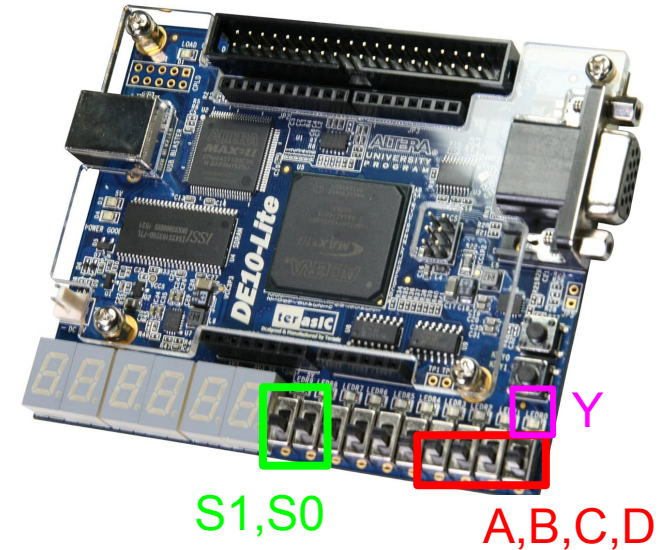
Asigne las entradas del multiplexor a los interruptores SW3-SW0 de la tarjeta de desarrollo y la salida al LED LEDR0. Utilice los interruptores SW8-SW9 para las señales de selección del multiplexor.



Trabajo en clase 2/3

Implemente un multiplexor
4x1 utilizando **únicamente**
Verilog Conductual.

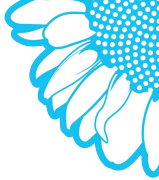
No instanciar módulos adicionales



S1,S0

A,B,C,D





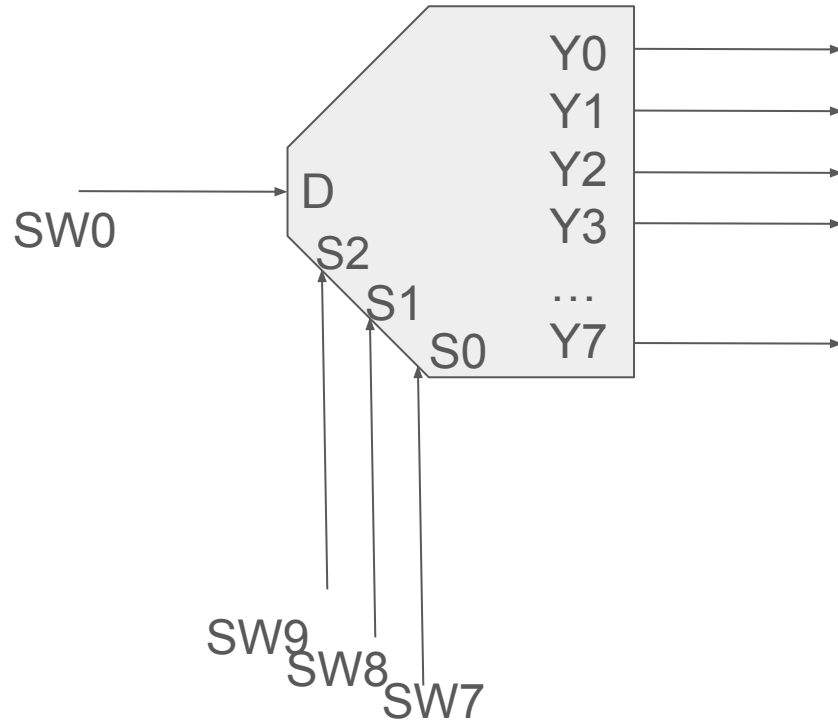
Trabajo en clase 3/3

Implemente un circuito que controle un segmento particular del display de siete segmentos HEX0 (utilice los 7 segmentos y el punto decimal). Para esto debe utilizar un demultiplexor que utilice las señales de selección para elegir a cuál segmento del display se le enviará el dato que indique el interruptor SW0. El segmento seleccionado deberá reflejar el estado actual del interruptor SW0: si está cerrado, el segmento seleccionado deberá iluminarse y si está abierto, el segmento deberá permanecer apagado. Los segmentos no seleccionados deberán permanecer apagados. Utilice los interruptores SW9, SW8 y SW7 para seleccionar el segmento a controlar, utilice la menor cantidad posible de interruptores.

Por ejemplo, si el valor de los switches $\{SW9, SW8, SW7\} = \{1, 1, 0\}$, se debe controlar el segmento 6, es decir el segmento del centro.



Trabajo en clase 3/3



Hacia los 7
segmentos
(+punto
decimal)

