

Multiplexor 6:1

Daniel Josué Rodríguez Agraz

OBJETIVOS

- Describir el hardware de un multiplexor 6:1 utilizando la herramienta Questasim y realizar la verificación con distintos casos.
- Desarrollar un testbench para la verificación el circuito.

INTRODUCCIÓN

Un multiplexor (MUX) es un dispositivo que permite dirigir la información digital procedente de diversas fuentes a una única línea para ser transmitida a través de dicha línea a un destino común. El multiplexor básico posee varias líneas de entrada de datos y una única línea de salida. También posee entradas de selección de datos, que permiten conmutar los datos digitales provenientes de cualquier entrada hacia la línea de salida. A los multiplexores también se les conoce como selectores de datos. [1]

La selección de una línea de entrada se controla con un conjunto de líneas de selección. Normalmente, hay 2^n líneas de entrada y n líneas de selección cuyas combinaciones de bits determinan cuál entrada es seleccionada.[2]

Un multiplexor de 6 líneas a 1 conecta una de seis fuentes a un destino común como se muestra en la figura 1. El circuito tiene seis líneas de entrada de datos, una línea de salida y una línea de selección S.

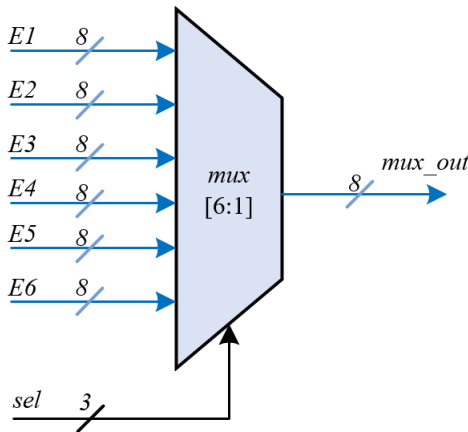


Fig. 1: Multiplexor de 6 líneas a 1.

Como se muestra en la tabla I los valores a la salida del multiplexor están directamente relacionados con el valor binario de la entrada de selección.

Sel[2]	Sel[1]	Sel[0]	Output
0	0	0	E1
0	0	1	E2
0	1	0	E3
0	1	1	E4
1	0	0	E5
1	0	1	E6
1	1	0	0
1	1	1	0

TABLE I: Selección de datos de un multiplexor de 6 entradas y 1 salida.

En este trabajo se realizó la descripción de hardware y la verificación de un multiplexor 6:1 de 8 bits utilizando Questasim en

lenguaje SystemVerilog. El objetivo principal fue diseñar y comprobar el correcto funcionamiento del multiplexor describiendo 14 casos con distintos valores y verificando los resultados a la salida. Tras la verificación, se comprobó que la salida era la esperada ante cambios en las entradas y en el selector, por lo que se pudo comprobar que la descripción era correcta dentro de esos casos.

DESARROLLO

Para el desarrollo de este proyecto, se utilizó el lenguaje de descripción de hardware SystemVerilog en la herramienta Questasim. A continuación, se detallan los pasos seguidos:

A. Descripción del Hardware

Se describe el multiplexor 6:1 en SystemVerilog. Este toma seis entradas de 8 bits y las envía a una única salida de 8 bits, dependiendo de la señal de selección. La señal de selección consta de tres bits, que permiten seleccionar una de las seis entradas. Para lograr esto, se hizo uso de la sentencia 'case' del propio lenguaje, como se muestra en el código a continuación.

Listing 1: Descripción del multiplexor 6:1 en SystemVerilog

```

module MUX6(
  input wire [7:0] E1,E2,E3,E4,E5,E6,
  input wire [2:0] sel,
  output reg [7:0] salida
);

always@(*)
begin
  case(sel)
    3'b000: salida = E1;
    3'b001: salida = E2;
    3'b010: salida = E3;
    3'b011: salida = E4;
    3'b100: salida = E5;
    3'b101: salida = E6;
    default: salida = 8'h00;
  endcase
end
endmodule

```

B. Verificación

En systemverilog, se crea un código de verificación, dónde se relacionan entradas y salidas con entradas y salidas del multiplexor descrito anteriormente y se les asignan valores, para que al realizar la simulación sea posible comparar los resultados a la salida con los resultados esperados. se asignaron 14 casos de simulación; 6 de ellos muestran cada una de las entradas a las salidas; 2 de ellos nos muestran los casos que no se utilizan; y los ultimos 6 son para comprobar que al cambiar la entrada seleccionada, cambia también la salida.

Listing 2: Descripción del multiplexor 6:1 en SystemVerilog

```

'timescale 1ns/1ps

module MUX6_TB();

  reg [7:0] E1,E2,E3,E4,E5,E6;
  reg [2:0] sel;

```

```

reg [7:0] salida;

MUX6 dub
(
    .E1(E1),
    .E2(E2),
    .E3(E3),
    .E4(E4),
    .E5(E5),
    .E6(E6),
    .sel(sel),
    .salida(salida)
);

initial
begin
    sel = 0'b000;
    E1 = 8'hE1;
    E2 = 8'hE2;
    E3 = 8'hE3;
    E4 = 8'hE4;
    E5 = 8'hE5;
    E6 = 8'hE6;
    #100;
    E1 = 8'hEA;
    #100;
    sel = 0'b001;
    #100;
    E2 = 8'hEB;
    #100;
    sel = 0'b010;
    #100;
    E3 = 8'hEC;
    #100;
    sel = 0'b011;
    #100;
    E4 = 8'hED;
    #100;
    sel = 0'b100;
    #100;
    E5 = 8'hEE;
    #100;
    sel = 0'b101;
    #100;
    E6 = 8'hEF;
    #100;
    sel = 0'b110;
    #100;
    sel = 0'b111;
    #100;
    $stop;
end
endmodule

```

RESULTADOS

Tras finalizar el código de verificación, se procede a realizar la simulación, donde se puede observar gráficamente cómo la salida cambia en función de los valores asignados a las entradas, los valores que se usarán de prueba y las salidas se muestran en la tabla II.

Como se muestra en la figura 2a, cuando el selector (sel) toma el valor de 3'b000, se selecciona la primera entrada (E1), y es este valor el que se refleja en la salida. Al cambiar el valor de la entrada

Sel (3 bits)	E1	E2	E3	E4	E5	E6	Output
000	0xE1	0xE2	0xE3	0xE4	0xE5	0xE6	0xE1
000	0xEA	0xE2	0xE3	0xE4	0xE5	0xE6	0xEA
001	0xEA	0xE2	0xE3	0xE4	0xE5	0xE6	0xE2
001	0xEA	0xEB	0xE3	0xE4	0xE5	0xE6	0xEB
010	0xEA	0xEB	0xE3	0xE4	0xE5	0xE6	0xE3
010	0xEA	0xEB	0xEC	0xE4	0xE5	0xE6	0xEC
011	0xEA	0xEB	0xEC	0xE4	0xE5	0xE6	0xE4
011	0xEA	0xEB	0xEC	0xED	0xE5	0xE6	0xED
100	0xEA	0xEB	0xEC	0xED	0xE5	0xE6	0xE5
100	0xEA	0xEB	0xEC	0xED	0xEE	0xE6	0xEE
101	0xEA	0xEB	0xEC	0xED	0xEE	0xE6	0xE6
101	0xEA	0xEB	0xEC	0xED	0xEE	0xEF	0xEF
110	0xEA	0xEB	0xEC	0xED	0xEE	0xEF	0x00
111	0xEA	0xEB	0xEC	0xED	0xEE	0xEF	0x00

TABLE II: Selección de datos de un multiplexor de 6 entradas y 1 salida.

(a) Resultados de la simulación (Parte 1).

(b) Resultados de la simulación (Parte 2).

Fig. 2: Resultados obtenidos tras la simulación del MUX 6:1.

E1, el valor de la salida también cambia de manera correspondiente. Lo mismo ocurre con el resto de las entradas; al cambiar el valor del selector a 3'b001, la salida toma el valor de la entrada E2, y al modificar el valor de la entrada seleccionada, la salida se actualiza en consecuencia.

CONCLUSIONES

La verificación de circuitos digitales es fundamental para asegurar el correcto funcionamiento de un circuito lógico. Utilizando las herramientas de Questasim, se logró diseñar y verificar un multiplexor de 6 entradas y 1 salida de 8 bits. Los resultados obtenidos confirman que el diseño del multiplexor es correcto y que su funcionamiento cumple con las expectativas, asegurando que los casos críticos están adecuadamente cubiertos y operan de manera adecuada.

REFERENCES

- [1] T. L. Floyd, *Fundamentos de Sistemas Digitales*, 9th ed.
- [2] M. M. Mano, *Digital design*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2002.