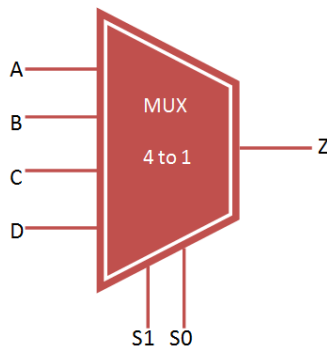


Documentación Ejercicio 2:

El ejercicio 2 consiste en un multiplexor 4-1, lo cual significa que es un dispositivo que toma cuatro entradas diferentes y elige cual dato pasar a la única salida mediante un código de selección. Adicionalmente, el tamaño de las entradas y salida debe ser parametrizable. Es decir que el tamaño de estos se debe poder ajustar a la necesidad. Para entender el funcionamiento de este multiplexor se presenta una tabla de verdad que explica su comportamiento.

Selección		Salida
Sel_1	Sel_0	out
0	0	in1
0	1	in2
1	0	in3
1	1	in4

En esta tabla de verdad se puede apreciar como el dato de la salida a depender del código de selección introducido. Cada una de las cuatro entradas tiene un código correspondiente que hace que su dato pase a la salida. Esto significa que solamente un valor esta disponible a la vez. Esto también se puede representar de forma gráfica.



En la imagen anterior se representa como el modulo tiene cuatro entradas y mediante los valores de S0 y S1 se genera una selección del dato que pasa a la salida Z. Aplicando esto a SystemVerilog se obtiene el siguiente código:

```
module Mux #(parameter WIDTH = 8) (  
    in1,  
    in2,  
    in3,  
    in4,  
    sel,  
    out  
);  
  
input Logic [WIDTH:0] in1;  
input Logic [WIDTH:0] in2;  
input Logic [WIDTH:0] in3;  
input Logic [WIDTH:0] in4;  
input Logic [1:0] sel;  
output Logic [WIDTH:0] out;  
  
always_comb begin  
    case (sel)  
        2'b00: begin  
            out = in1;  
        end  
  
        2'b01: begin  
            out = in2;  
        end  
  
        2'b10: begin  
            out = in3;  
        end  
  
        2'b11: begin  
            out = in4;  
        end  
  
        default: begin  
            out = 1'b0;  
        end  
    endcase  
end  
endmodule
```

En este código se genera un modulo llamado MUX el cual contiene cinco señales de entrada y una de salida. Las señales de entrada son los cuatro datos de entrada y una señal de selección para elegir el dato deseado. Además, la única señal de salida es el dato que se elige transmitir. Lo primero a destacar es que como se mencionó anteriormente, el ancho de los buses de entrada y salida debe ser parametrizable. Para esto se utiliza un “#(parameter WIDTH = 8)” antes de la declaración del módulo. Con esta instrucción se está estableciendo un parámetro llamado “WIDTH” el cual tiene un valor default de 8. Este parámetro luego se usa para declarar el tamaño de las señales de entrada y salida. Este valor al ser instanciado en un testbench puede ser configurado al cualquier valor deseado. Esto se hace para no restringir el ancho de los buses al escribir el módulo. Sin embargo, la parametrización de estos se puede hacer directamente cuando se vayan a usar.

Ahora para lograr el funcionamiento del multiplexor, utiliza la herramienta “case” de systemverilog. La cual permite tomar una señal y asignar que hacer dependiendo

del valor de esta señal. En este caso la señal que toma las decisiones es “sel” la cual puede contener una de las 4 combinaciones posibles para cada entrada. Con esto se toma cada combinación y en caso de que se tenga alguna se le asigna el valor correspondiente de entrada a la salida. Este es una forma bastante sencilla de implementar el comportamiento del MUX 4-1 en systemverilog.