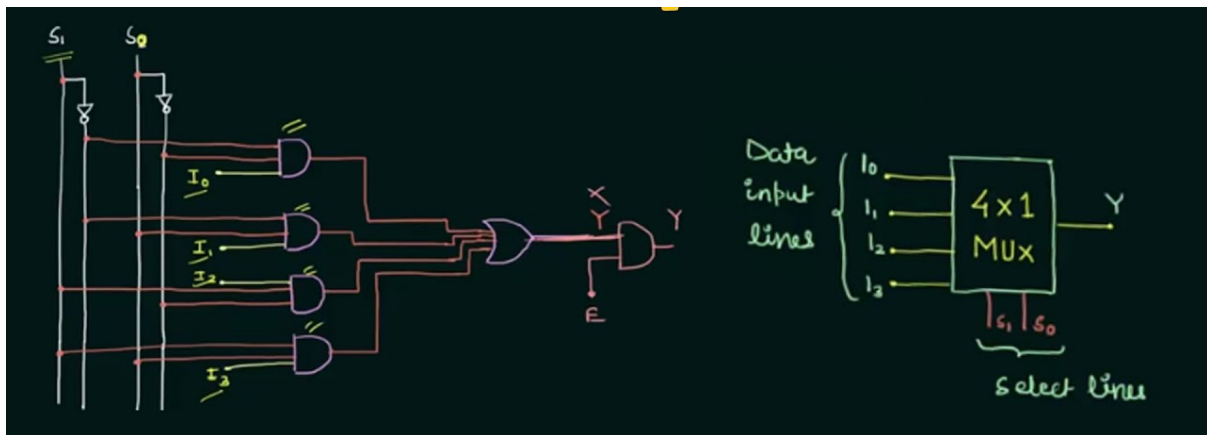


**Instituto Tecnológico y De Estudios  
Superiores de Monterrey  
Laboratorio - Arquitectura de Computadoras  
TE-2031.1**



**Lab 05 - 4 to 1 Multiplexer**

**Entrega: 16 / 10 / 2022**

**Equipo 07:**

**Héctor Javier Pequeño Chairez**

**A01246364**

**Gabriela Jazmín Álvarez Espinoza**

**A00825719**

## Introducción

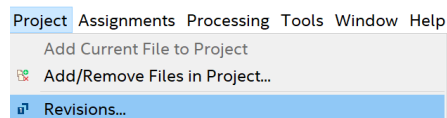
## Desarrollo

En la siguiente práctica iniciamos abriendo el proyecto que creamos en la práctica pasada, en este caso nuestro proyecto es el siguiente:



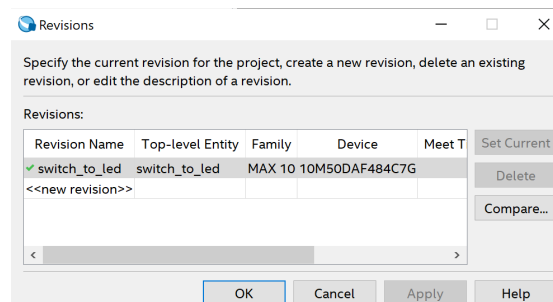
Nombre y ubicación de Nuestro Proyecto.

Después ejecutamos el uso de la herramienta “revisión”, por lo que necesitamos acceder a: Proyecto -> Revisión.



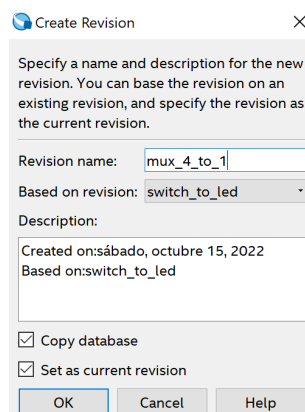
Ubicación de la opción en pantalla.

Esto nos despliega la siguiente ventana emergente:



Ventana con la información de las revisiones.

Hacemos click en “new revision” y agregamos una nueva revisión con el nombre mux\_4\_to\_1.



Nueva revisión creada con el nombre mux\_4\_to\_1.

Una vez creada la revisión, asignamos un archivo tipo VHDL en el cual introducimos el siguiente código.

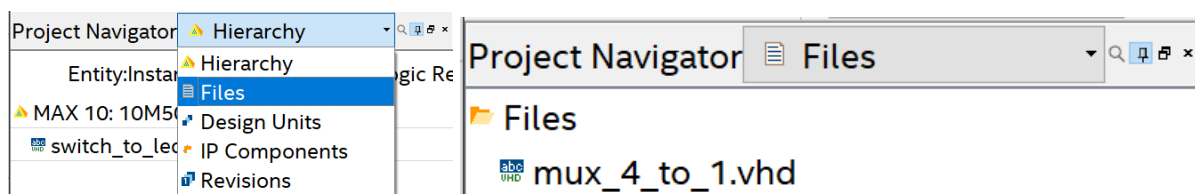
```

1 library ieee;
2 use ieee.std_logic_1164.all;
3
4 entity mux_4_to_1 is
5     port( sw : in std_logic_vector(9 downto 0);
6           ledr : out std_logic_vector(9 downto 0));
7 end entity;
8
9 architecture Behavior of mux_4_to_1 is
10     component mux
11     port ( x0,x1,x2,x3 : in std_logic;
12           sel          : in std_logic_vector(1 downto 0);
13           y            : out std_logic);
14     end component;
15 begin
16     U : mux port map(sw(0), sw(1), sw(2), sw(3),
17                     sw(5 downto 4),
18                     ledr(0));
19 end architecture;
20
21 library ieee;
22 use ieee.std_logic_1164.all;
23
24 entity mux is
25     port ( x0,x1,x2,x3 : in std_logic;
26           sel          : in std_logic_vector(1 downto 0);
27           y            : out std_logic);
28 end entity;
29
30 architecture with_WHEN of mux is
31 begin
32     y <= x0 WHEN sel = "00" ELSE
33          x1 WHEN sel = "01" ELSE
34          x2 WHEN sel = "10" ELSE
35          x3;
36 END ARCHITECTURE;
  
```

Código asignado en Quartus, correspondiente a un mux 4 a 1.

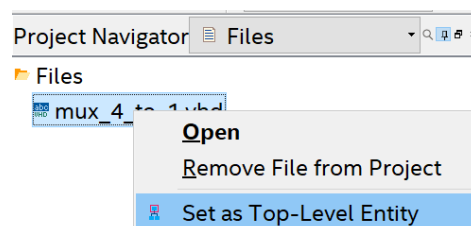
### ***Revisión Control***

Verificamos que los archivos que sean necesarios para esta práctica se encuentren de manera correcta. Por lo que se debió remover el archivo de switch\_to\_led.vhd



Archivo correspondiente al mux 4 a 1.

Agregamos nuestro mux como “Top-level Entity”.



Ejecución “Set as top-level entity”.

Verificamos que nuestros nombres coinciden para evitar errores de compilación.

**General**

You can change the top-level entity for the design; however, it is recommended that you create a new revision for each entity in order to maintain settings information.

Top-level entity:

Recently selected top-level entities:

Coincidencias de nombre “mux\_4\_to\_1”.

### Compilación

Flow Status Successful - Sat Oct 1

- Quartus Prime Timing Analyzer was successful. 0 errors, 5 warnings
- 293000 Quartus Prime Full Compilation was successful. 0 errors, 29 warnings

Compilación ejecutada de manera correcta.

### Descargar diseño en el dispositivo

Ahora cargamos nuestro diseño en el FPGA, por lo que abrimos el programmer y pulsamos start para iniciar la subida del archivo.

Programmer - C:\intelFPGA\_lite\18.1\Quartus Architecture\Lab03NewProjectWizard - mux\_4\_to\_1 - [mux\_4\_to\_1.cdf]

File Edit View Processing Tools Window Help

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress: 00% (Successful)

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
output_files/mux_4_to_1.sof	10M50DAF484	002705E9	002705E9	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

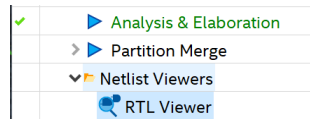
Start Stop Auto Download Delete Add File Change File Save File Add Device Up Down

TDI TDO 10M50DAF484

Carga a la placa realizada de manera correcta.

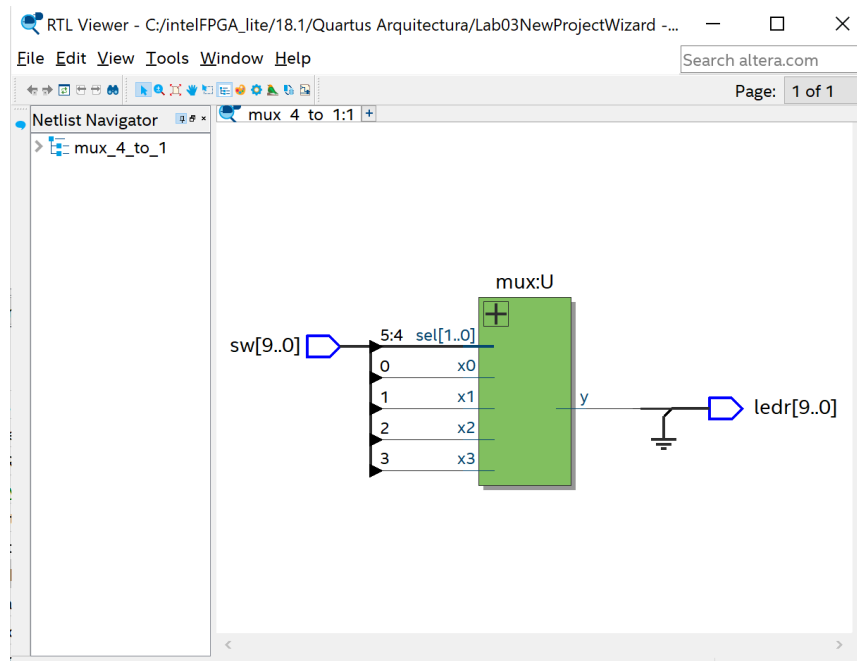
### RTL Viewer

Para poder ver el diseño que se genera por la síntesis del código que generamos, debemos utilizar la herramienta Register Transfer Level Viewer, para esto debemos ingresar en la sección de Analysis & Synthesis y seleccionamos Netlist Viewer.



Opción RTL Viewer para verificar el circuito.

En esta ocasión se nos despliega el circuito lógico que fue creado después de realizar la síntesis de nuestro código.



Mux 4 a 1, con un selector de dos bits.

### ***Funcionalidad del circuito***

En este caso, el código lo que realiza es la definición de nuestros switches y leds dentro de nuestro mux, teniendo 10 switches de entrada y 10 leds de salida.

```
library ieee;
use ieee.std_logic_1164.all;

entity mux_4_to_1 is
    port( sw : in std_logic_vector(9 downto 0);
          ledr : out std_logic_vector(9 downto 0))
end entity;
```

Definición de switches y leds.

En el siguiente componente definimos el mux 4 a 1, definiendo 4 variables de entradas denominadas x0, x1, x2, x3, luego contamos con un selector de dos bits siendo de tipo entrada, por último definimos una salida “y”, de un bit.

```
architecture Behavior of mux_4_to_1 is
    component mux
        port ( x0,x1,x2,x3 : in std_logic;
              sel       : in std_logic_vector(1 downto 0);
              y         : out std_logic);
    end component;
```

Entradas y salidas del componente mux.

Ahora hacemos un mapeo de las entradas y salidas definidas en nuestro mux, en donde asignamos uno a uno los switches de nuestro multiplexor, en este caso sw(0) a sw(3) siendo las 4 entradas de donde se obtendrá la lectura hacia “y”. Después se asigna al s(4) y s(5) el selector, con estas dos entradas podremos controlar el valor de esos dos bits. Por último asignamos el led 0 el resultado de “y”, recordando que “y” obtendrá el valor de x0, x1, x2 o x3, según el selector.

```
begin
    U : mux port map (sw(0), sw(1), sw(2), sw(3),
                     sw(5 downto 4),
                     ledr(0));
end architecture;
```

### Problemas y soluciones

Dentro de la práctica, tuvimos pocos problemas debido a la guía en base a la descripción de la práctica, sin embargo al generar el RTL Viewer tuvimos problemas, pues teníamos una versión previa del archivo y no se estaba generando el RTL Viewer. Checamos la carpeta en donde se generó la práctica y eliminamos todos los archivos, volvimos a escribir el código y logramos generar la vista del RTL Viewer en donde podíamos ver de manera gráfica las conexiones que tenemos actualmente mediante la declaración de inputs y outputs.

### Reflexión Individual

**Héctor:** En esta práctica realizamos una repetición de los pasos que ya habíamos ejecutado en el antiguo laboratorio, solo que en este caso recordamos conceptos como “port map”, el cual nos ayuda a conectar diferentes parámetros por “referencia”, además de que realizamos el uso de la herramienta RTL viewer, que nunca había utilizado. Conocía que este tipo de lenguajes de programación de hardware realizaban una conversión a esquemático pero jamás había visto cómo es que se observaba uno. Por último, también es importante tener en cuenta que realizamos el recordatorio de cómo funciona un multiplexor, en este caso de 4 entradas y 1 salida.

**Gabriela:** Gracias a esta práctica y el uso de RTL Viewer me ayudo a poder conectar los conceptos de declaración de variables, ya relacionada a lo que podría decir que es un producto final, que en este caso es el multiplexor con sus inputs y outputs, esto creo yo me puede ayudar a poder a de cierta manera, ver de manera física la creación de sub módulos que nosotros mismos podemos crear.

**Video**

[https://drive.google.com/file/d/1PZ2JFMbwEryC1f\\_q5HN--x8B8DxzJt5u/view?usp=sharing](https://drive.google.com/file/d/1PZ2JFMbwEryC1f_q5HN--x8B8DxzJt5u/view?usp=sharing)

**Bibliografía**

1. Manual DE10-lite :

[https://www.terasic.com.tw/cgi-bin/page/archive\\_download.pl?Language=English&No=1021&FID=a13a2782811152b477e60203d34b1baa](https://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=1021&FID=a13a2782811152b477e60203d34b1baa)

2. Guia Practica lab 05:

[https://docs.google.com/document/d/1kll1m6uc2swWt3tTW7OggFLLa\\_-0BL6pFTGbt36q9eQ/edit?usp=sharing](https://docs.google.com/document/d/1kll1m6uc2swWt3tTW7OggFLLa_-0BL6pFTGbt36q9eQ/edit?usp=sharing)