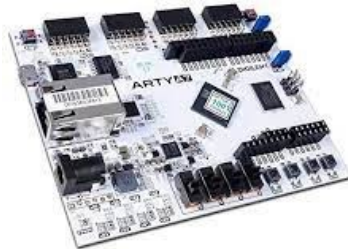


Sistemas Digitales para las Comunicaciones

Ejercicio 11

para la Arty A-35



Profesor: Federico Zacchigna

Alumno: Pablo Daniel Folino

Repositorio de trabajos prácticos: <https://github.com/MSE-SDC/trabajos-practicos-PabloFolino>

2021

Índice

Trabajo práctico ejercicio 11.....	2
Ejecutar el Vivado.....	2
El paso a paso.....	2
Configurar el ILA (Integrated Logic Analyzer).....	11

Trabajo práctico ejercicio 11

Ejecutar el Vivado

Para ejecutar el vivado hay que ir al directorios:

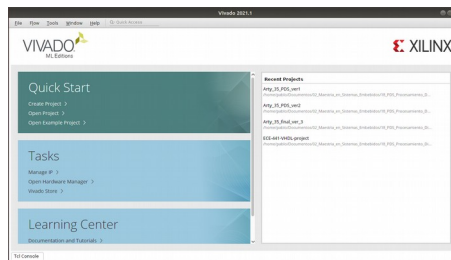
```
cd /tools/Xilinx/Vivado/2021.1/bin y  
escribir sudo ./vivado
```



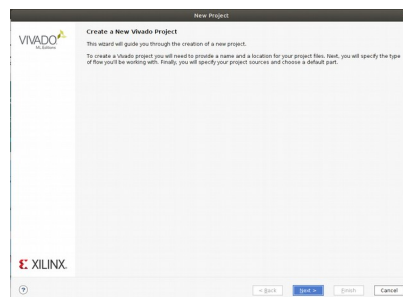
El paso a paso

El profesor trabaja con vivado **2019.2**, y tengo instalado el **2021.1**.

1) Al habrirlo se observa:



2) Se va a: **Create Project**, y hacer click en **Next**.



3) Se Selecciona la carpeta(**Project location**)

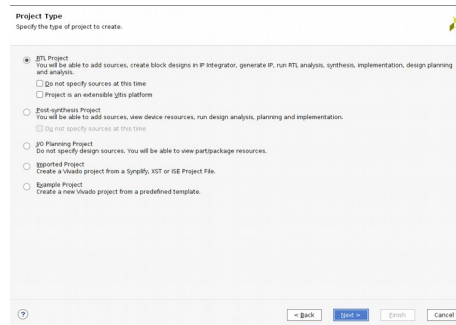
**/home/pablo/Documentos/02_Maestria_en_Sistemas_Embebidos/
21_Sistemas_Digitales_para_las_Comunicaciones/workspace**

y un nombre del proyecto, que en mi caso será:

MSE-SDC_ejercicio11

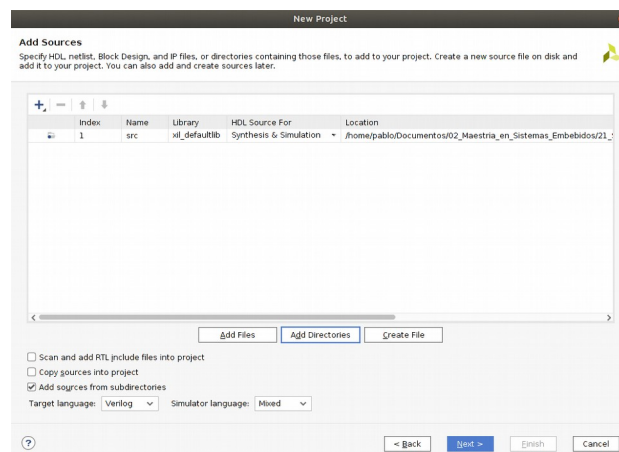
y hacer click en **Next**.

4) Seleccionar proyecto RTL, y hacer click en **Next**.



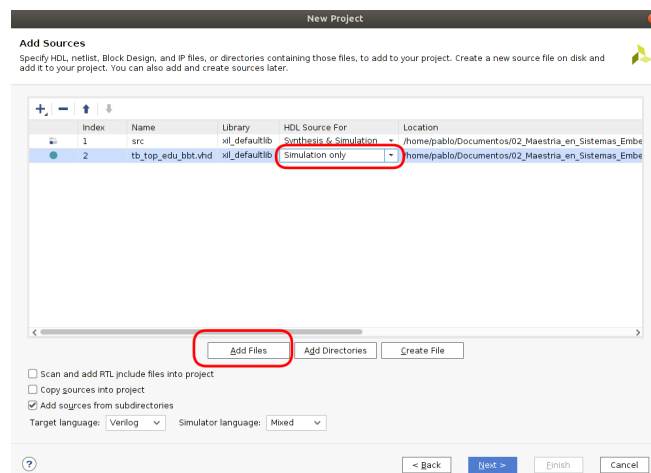
5) Ir al boton "**Add Directories**" y sumar el directorio "**src**", en mi caso se encuentra en la ruta:

/home/pablo/Documentos/02_Maestria_en_Sistemas_Embebidos/21_Sistemas_Digitales_para_las_Comunicaciones/Repositorio/MSE-SDC-6Co2021/modem/src



y hacer click en **Next**.

6) Ir a boton "**Add Files**" y agregamos un solo testbench "**tb_top_edu_bbt**", y marcarlo como "**Simulation Only**".



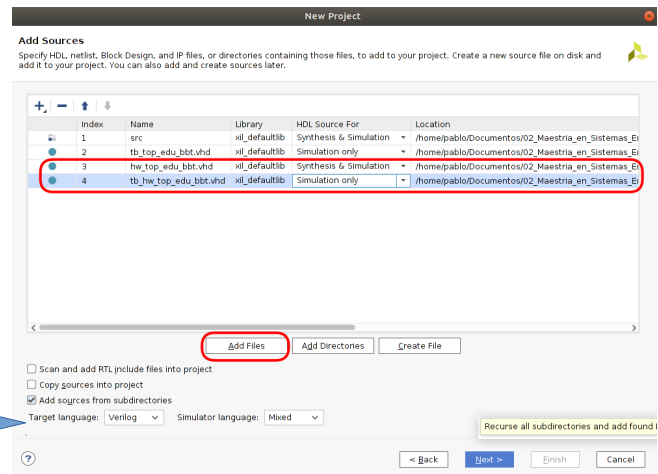
y hacer click en **Next**.

7) Ir a boton **"Add Files"** ir a la carpeta de **"/HW/artyz7-10/edu_bbt"** y agregamos los archivos **"hw_top_edu_bbt.vhd"**, y **"tb_hw_top_edu_bbt.vhd"**, este último lo marcamos como sólo simulación ...

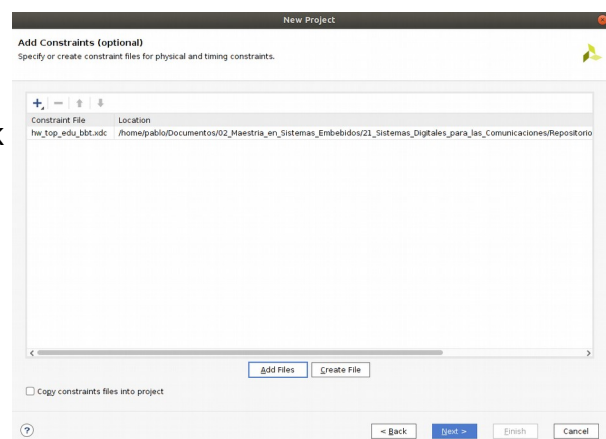
y hacer click en **Next**.

Verificar que el **Target**

sea VHDL

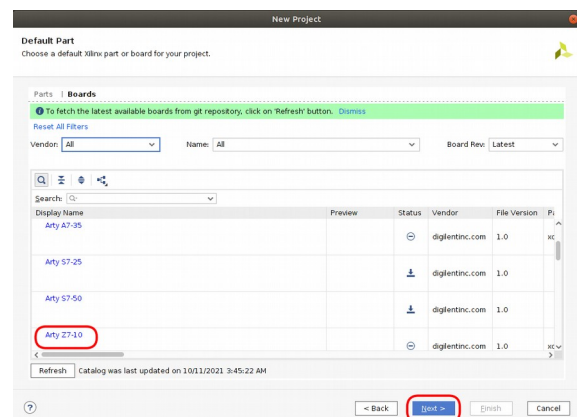


8) Ir al boton **"Add Files"** y agregar el constraints **"hw_top_edu_bbt.xdc"**, y hacer click en **Next**.



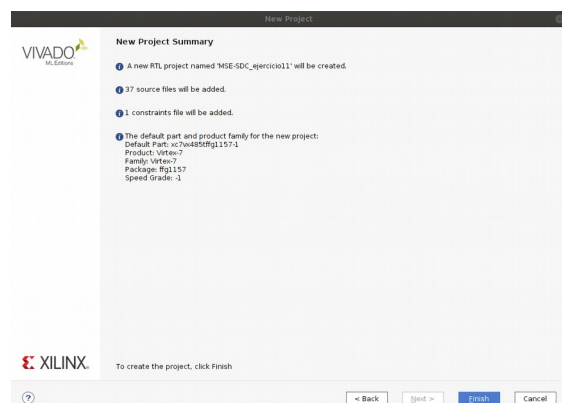
9) Agregamos la placa **"Arty Z7-10"**

y hacer click en **Next**.

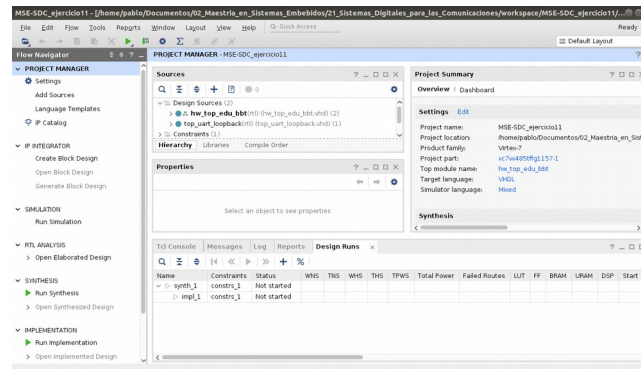


10) El sistema queda:

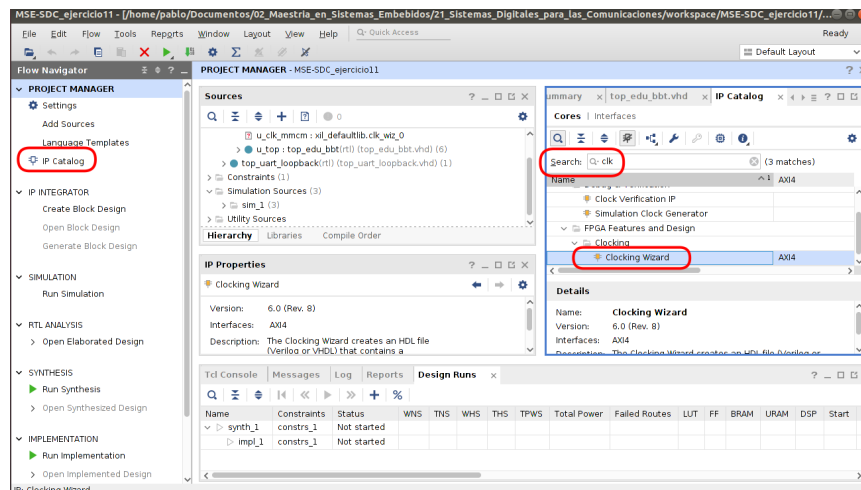
y hacer click en **Next**.



11) Se explica en el video **"Clase5_SDC.mp4"** a las 2 hs y 21 minutos.

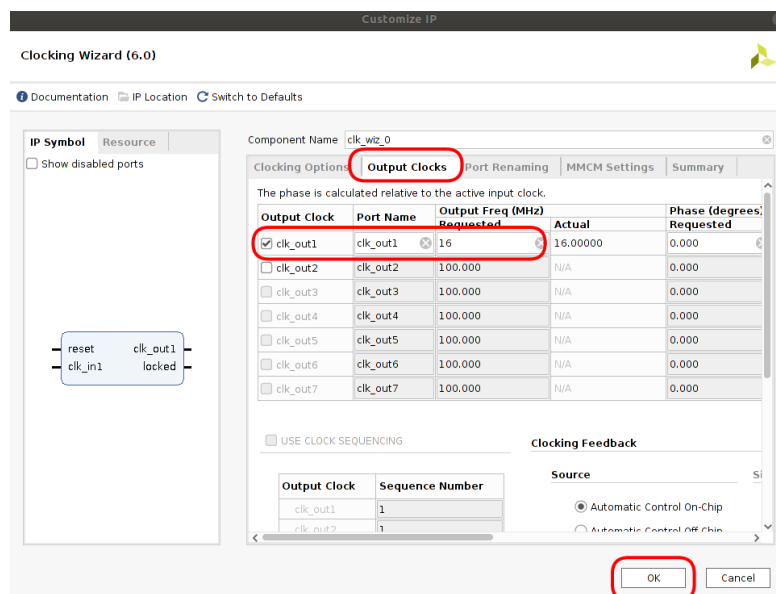


12) Se va a **"IP Catalog"**, en **"Search"** se escribe **clk**, y se elije **"Clocking Wizard"**.



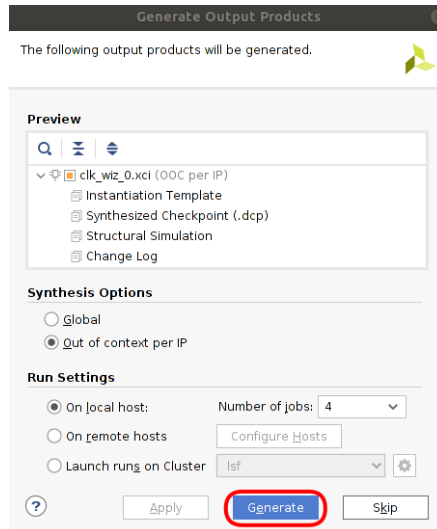
Se hace doble click.

13) Se va ala solapa **"Output Clocks"** y se pone **16MHz** en **clk_out1**, presionando al boton de **OK**.

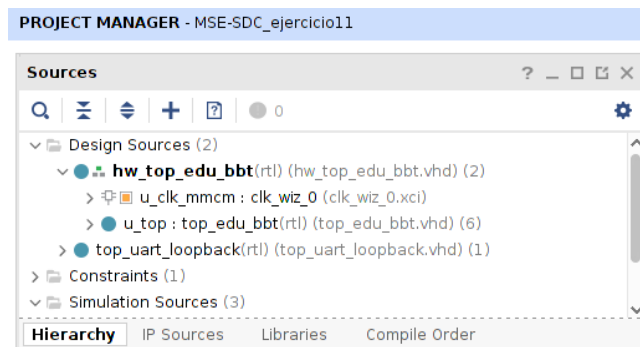


Importante para la Arty Z7: En la solapa **Clock Options** cambiar la frecuencia del clock primario de 100MHz a 125MHz. Si se usa la **Arty A35** dejarlo en 100MHz.

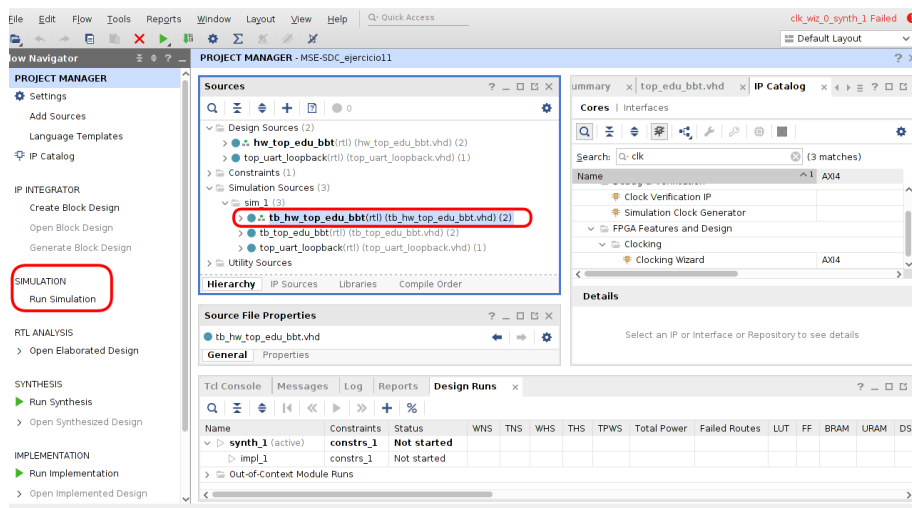
14) Aparece el siguiente cartel, y presionar el boton de **"Generate"**.



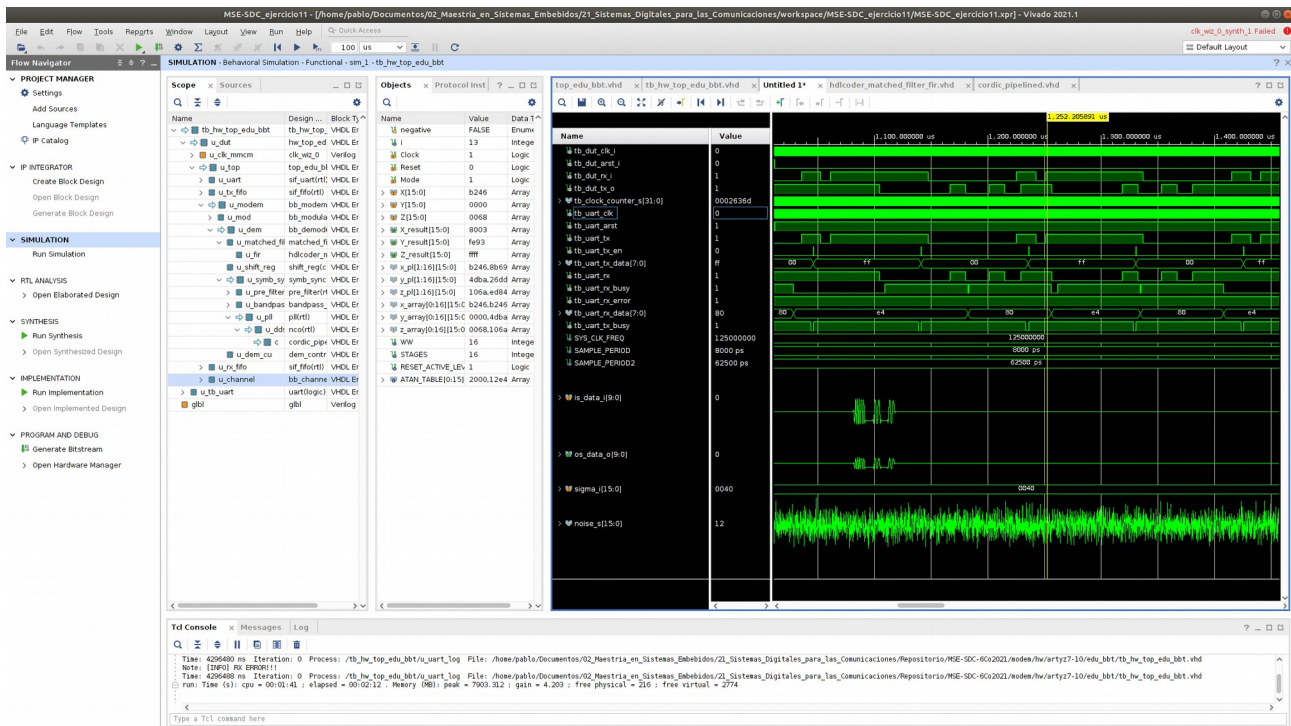
15) Verificar que en la pantalla de Source quede así:



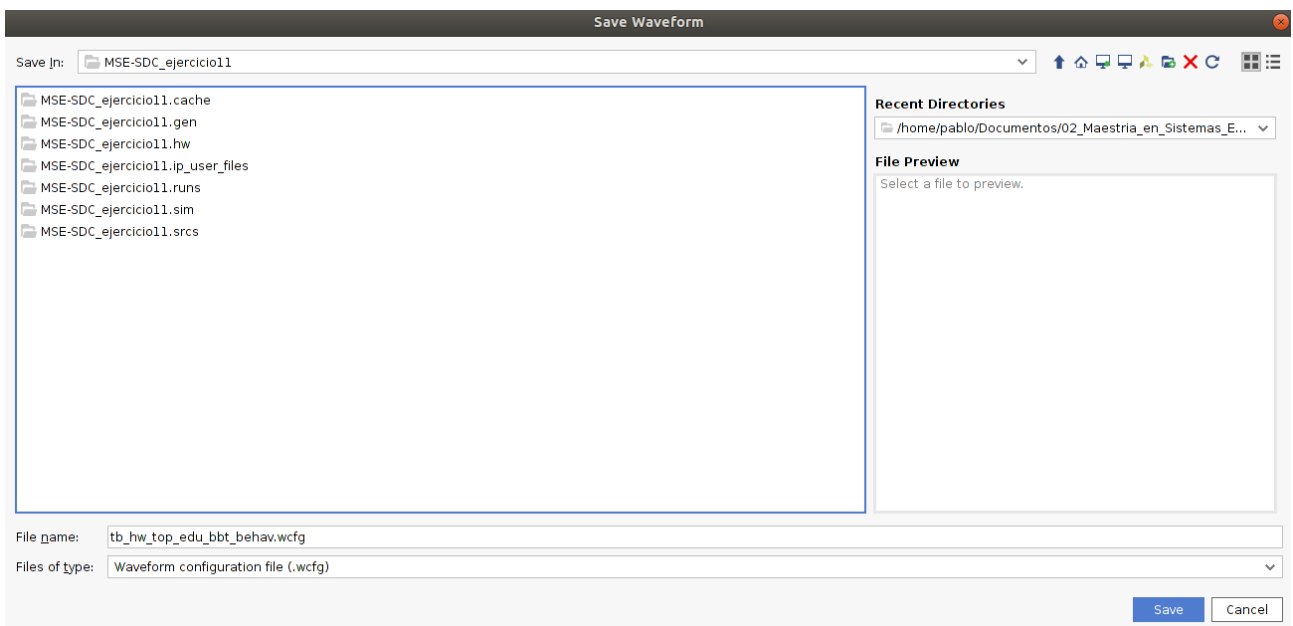
16) Para simular se puede ir a **"Simulation Source"** y posicionarse en el archivo **"tb_hw_top_edu_bbt.vhd"**, ir al menu a la izquierda **"SIMULATION"**, y luego presionar **"Run Simulation"**.



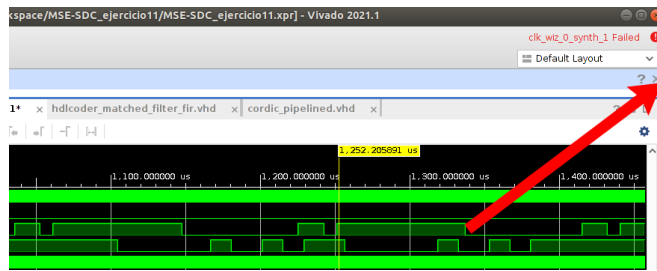
17) Agrego por ejemplo las señales del canal, y simulo de nuevo. Importante: para ver la señales analógicas, posicionaese en la señal con el boton derecho del mouse seleccioner el menú **"Waveform Style"** y luego seleccionar **"Analog"**. IR nuevamente al menú y seleccionar **"Radix"** y seleccionar **"Signed Decimal"**.



18) Si se presiona **"CTRL+S"** se puede guardar las señales



19) Cerramos la simulación con la "x" de la esquina superior derecha.



20) Generar el Bitstream

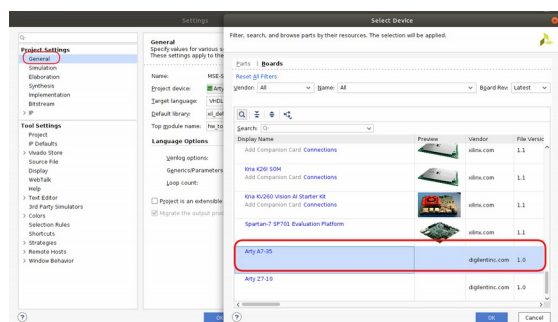
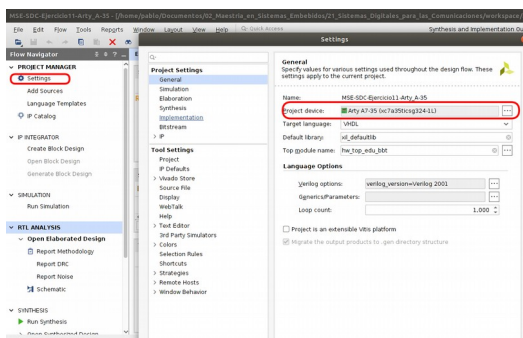
a. Seleccionar el directorio de trabajo:

**/home/pablo/Documents/02_Maestria_en_Sistemas_Embebidos/
21_Sistemas_Digitales_para_las_Comunicaciones/workspace**

21) Como tengo otra placa la ARTY A-35, copio el proyecto.

Para eso se debe ir al menú "File", luego a "Project" y "Save as", escribir en **Project Name: "MSE-SDC-Ejercicio11-Arty_A-35"**, y en **Project Location: "/home/pablo/Documents/02_Maestria_en_Sistemas_Embebidos/21_Sistemas_Digitales_para_las_Comunicaciones/workspace"**

22) Voy a "Settings" y cambio la placa:



22) Mirando el Constraints original lo adapto para la nueva placa, a continuación se muestran las líneas utilizadas:

```
## Clock signal

set_property -dict { PACKAGE_PIN E3      IOSTANDARD LVCMOS33 } [get_ports { clk_i }]; #IO_L12P_T1_MRCC_35
Sch=gclk[100]


## LEDs

set_property -dict { PACKAGE_PIN H5      IOSTANDARD LVCMOS33 } [get_ports { led_o[0] }]; #IO_L24N_T3_35 Sch=led[4]
set_property -dict { PACKAGE_PIN J5      IOSTANDARD LVCMOS33 } [get_ports { led_o[1] }]; #IO_25_35 Sch=led[5]
set_property -dict { PACKAGE_PIN T9      IOSTANDARD LVCMOS33 } [get_ports { led_o[2] }]; #IO_L24P_T3_A01_D17_14
Sch=led[6]

set_property -dict { PACKAGE_PIN T10     IOSTANDARD LVCMOS33 } [get_ports { led_o[3] }]; #IO_L24N_T3_A00_D16_14
Sch=led[7]


## Buttons

set_property -dict { PACKAGE_PIN D9      IOSTANDARD LVCMOS33 } [get_ports { arst_i }]; #IO_L6N_T0_VREF_16 Sch=btn[0]


## USB-UART Interface

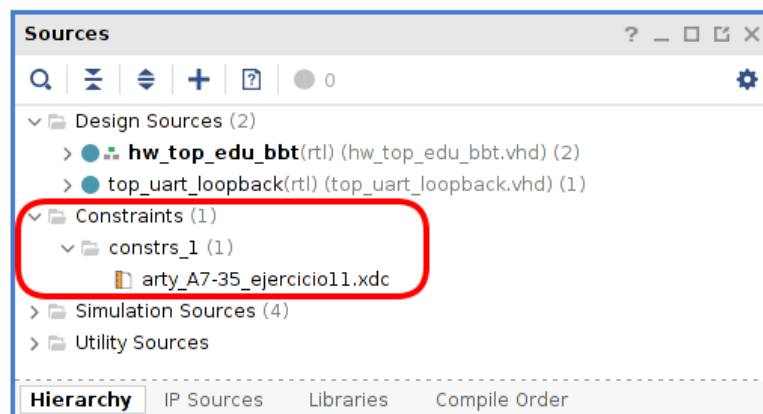
set_property -dict { PACKAGE_PIN D10     IOSTANDARD LVCMOS33 } [get_ports { tx_o }]; #IO_L19N_T3_VREF_16
Sch=uart_rxd_out

set_property -dict { PACKAGE_PIN A9      IOSTANDARD LVCMOS33 } [get_ports { rx_i }]; #IO_L14N_T2_SRCC_16
Sch=uart_txd_in
```

se lo guarda en un archivo **"arty_A7-35_ejercicio11.xdc"**.

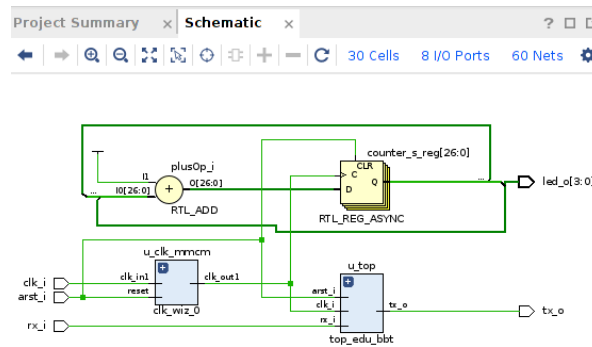
23) A continuación descargar el viejo Constraints **"hw_top_edu_bbt.xdc"**, y cargar el nuevo archivo Constraints **"arty_A7-35_ejercicio11.xdc"**.

El **"Source"** debe quedar:



24) Verificar todo, eso implica:

- Correr la simulación
- Correr el análisis RTL, que debe dar:

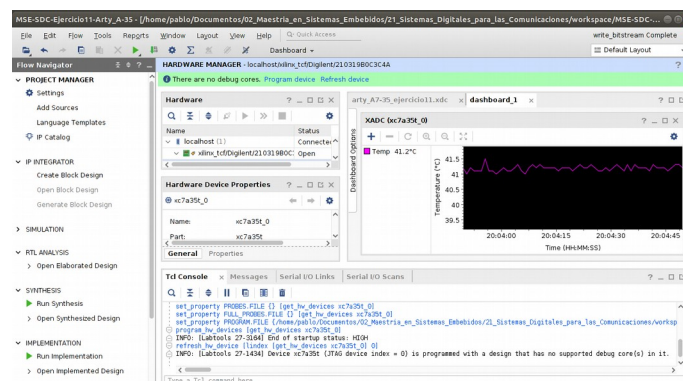


- Sintetizar, y verificar que no haya errores.
- Correr la implementación, y verificar los recursos.
- Generar el Bitstream.
- Programar la placa Artty A-35.

Importante: cuando se hace todo esto abrir el "Monitor de Sistema" de Ubuntu, y verificar la **Memoria de Intercambio**, si esta se llena el sistema operativo se tilda.

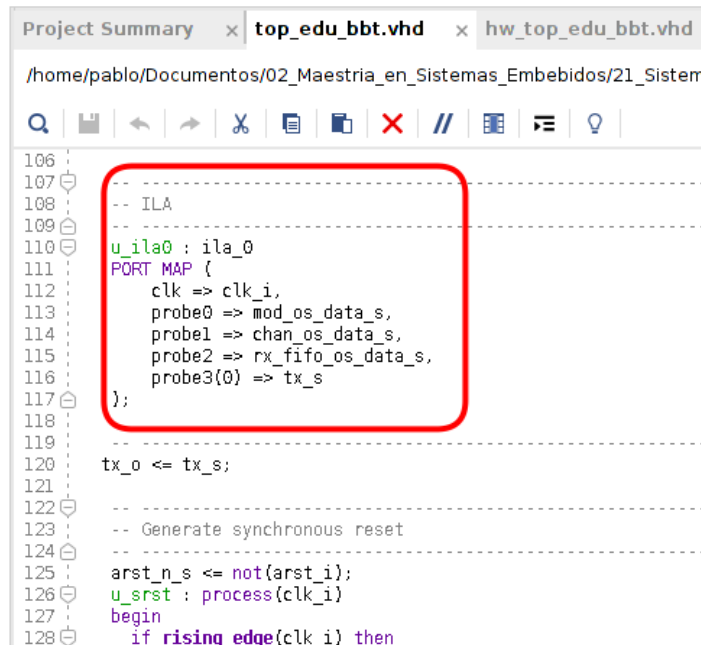
Para programar la placa ir al directorio `"/workspace/MSE-SDC-Ejercicio11-Arty_A-35/MSE-SDC-Ejercicio11-Arty_A-35.runs/impl_1"`, y bucar el archivo `"hw_top_edu_bbt.bit"`.

25) Abrir una terminal como por ejemplo GtKTerm configurar en **115200** baudios y mandar algunos caracteres.



Configurar el ILA (Integrated Logic Analyzer)

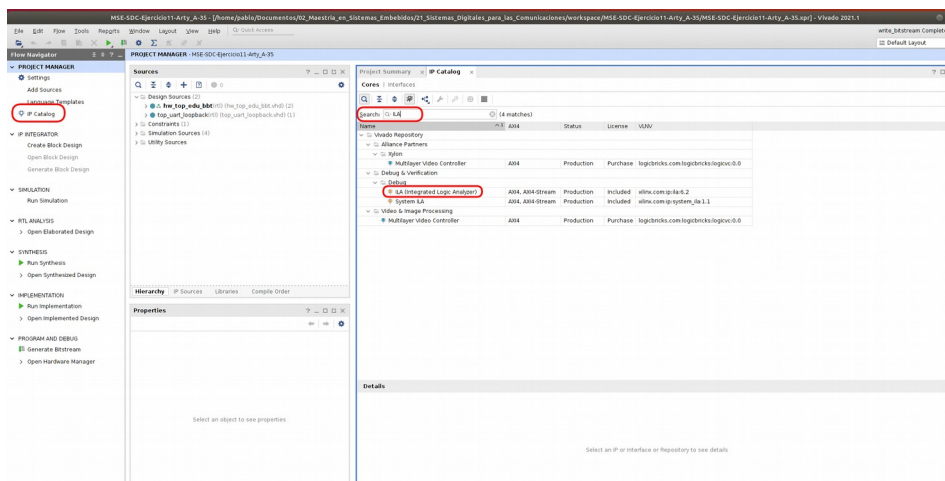
En el archivo “top_edu_bbt.vhd”, ya se tiene un ILA precargado:



```
106
107
108 -- ILA
109
110 u_ila0 : ila_0
111 PORT MAP (
112     clk => clk_i,
113     probe0 => mod_os_data_s,
114     probe1 => chan_os_data_s,
115     probe2 => rx_fifo_os_data_s,
116     probe3(0) => tx_s
117 );
118
119
120 tx_o <= tx_s;
121
122
123 -- Generate synchronous reset
124
125 arst_n_s <= not(arst_i);
126 u_srst : process(clk_i)
127 begin
128     if rising_edge(clk_i) then
```

Faltaría agregar el en el **IP Catalog** el módulo de referencia, respetando la señales declaradas:

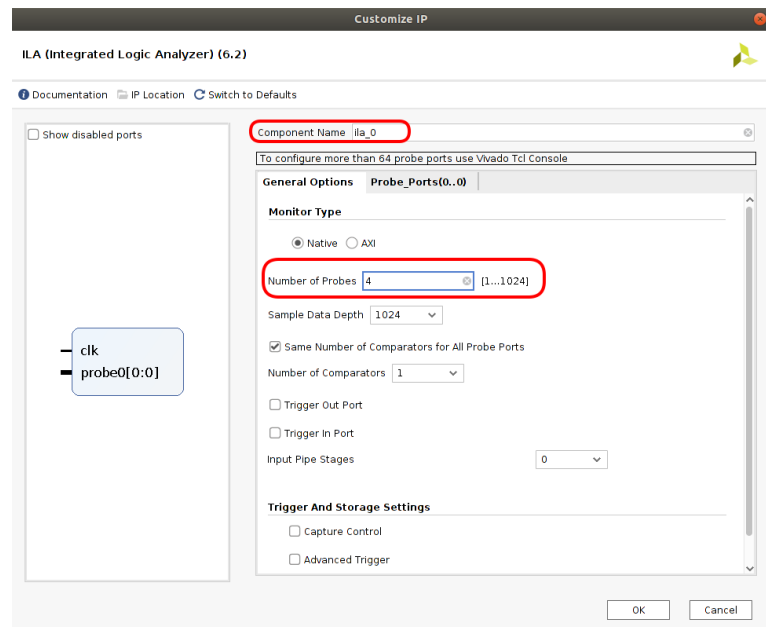
1) Ir al manú izquierdo de la aplicación a la opción “IP Catalog”, poner en el vbuscador la palabra “ILA”, y agregar el módulo:



2) Al hacer doble click sobre se abre la siguiente pantalla:

ILA (Integrated Logic Analyzer)

Verificar que el nombre del módulo sea **ila_0** , y que el número de pruebas sea **4**.

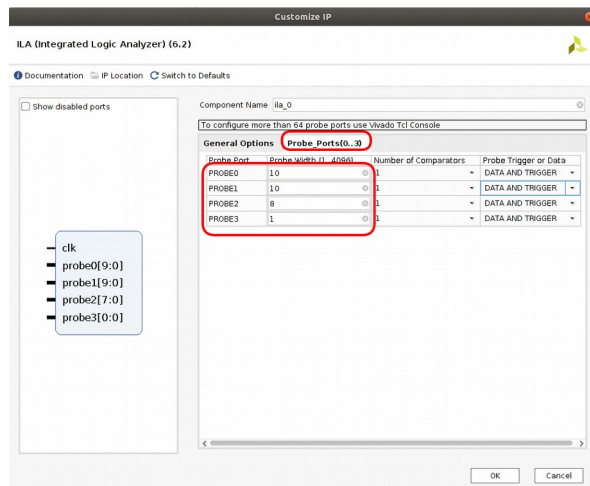


3) Pasar a la siguiente solapa **"Probe_Ports"**, y haer coincidir cada punto de prueba con la longitud de la variable del archivo **"top_edu_bbt.vhd"**.

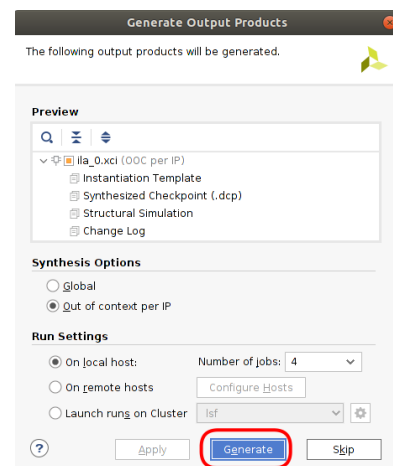
```

93 COMPONENT ila_0
94 PORT (
95     clk : IN STD_LOGIC;
96     probe0 : IN STD_LOGIC_VECTOR(9 DOWNT0 0);
97     probe1 : IN STD_LOGIC_VECTOR(9 DOWNT0 0);
98     probe2 : IN STD_LOGIC_VECTOR(7 DOWNT0 0);
99     probe3 : IN STD_LOGIC_VECTOR(0 DOWNT0 0);
100 );
101 END COMPONENT ;
102
103
104 begin
105
106
107
108 -- ILA
109
110 u_ila0 : ila_0
111 PORT MAP (
112     clk => clk_i,
113     probe0 => mod_os_data_s,
114     probe1 => chan_os_data_s,
115     probe2 => rx_fifo_os_data_s,
116     probe3(0) => tx_s
117 );
118
119
120

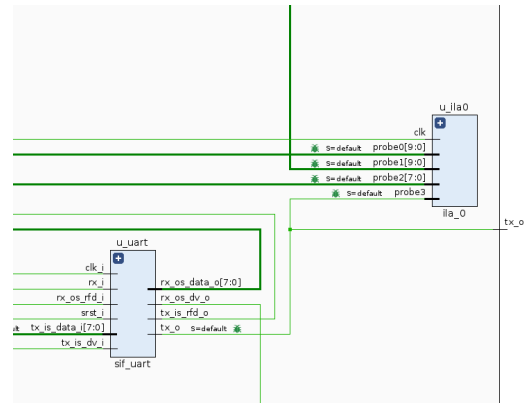
```



4) Al hacer click en el boton **"OK"**, se abre el siguiente cuadro y clickear en el boton **"Generate"**.



5) Si se corre el análisis RTL, verificar que el módulo ILA este conectado:



6) Generar el Bitstream, programar la FPGA y verificar que el ILA anda funcionando.

a. Acordarse de cambiar la frecuencia de refresco, que debe ser menor que la mitad de los 16MHz. En este caso se puso 6MHz.

b. Configurar el disparo, que para este caso se lo configuró con flanco descendente de **tx_s**.

