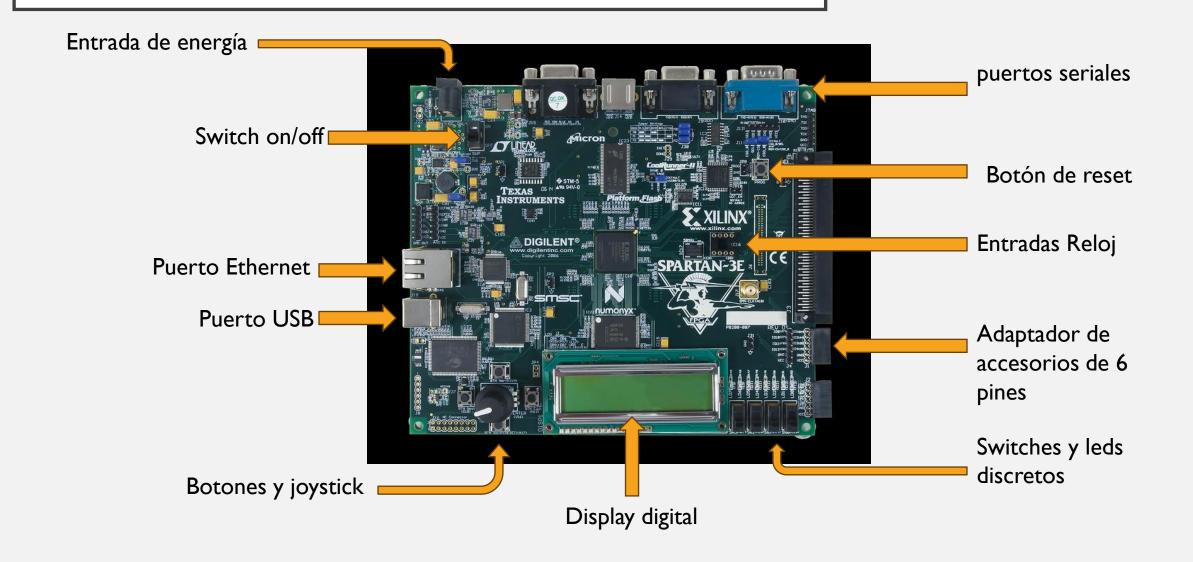


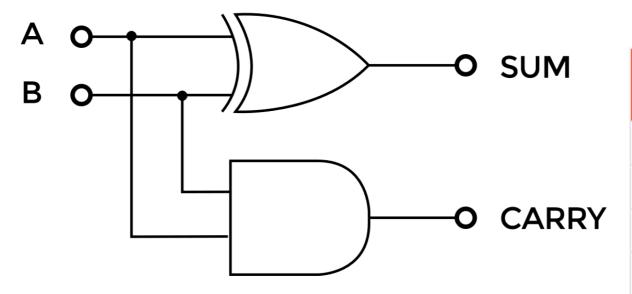
### UN ACERCAMIENTO A LA FPGA





#### PASOS ANTES DE APLICARLO EN XILINX

- Realizar el esquema lógico de medio sumador de bits en papel
- El esquema debe satisfacer lo siguiente:
  - Sume A y B
  - Exista un CARRY



A	В	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

LA TABLA DE VERDAD Y SU ESQUEMA

#### CREAR NUEVO PROYECTO EN XILINX

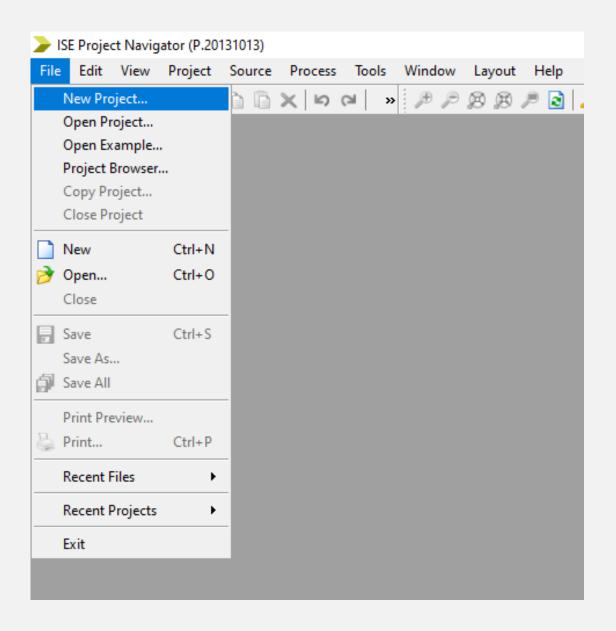
I. Creen una carpeta para guardar los trabajos

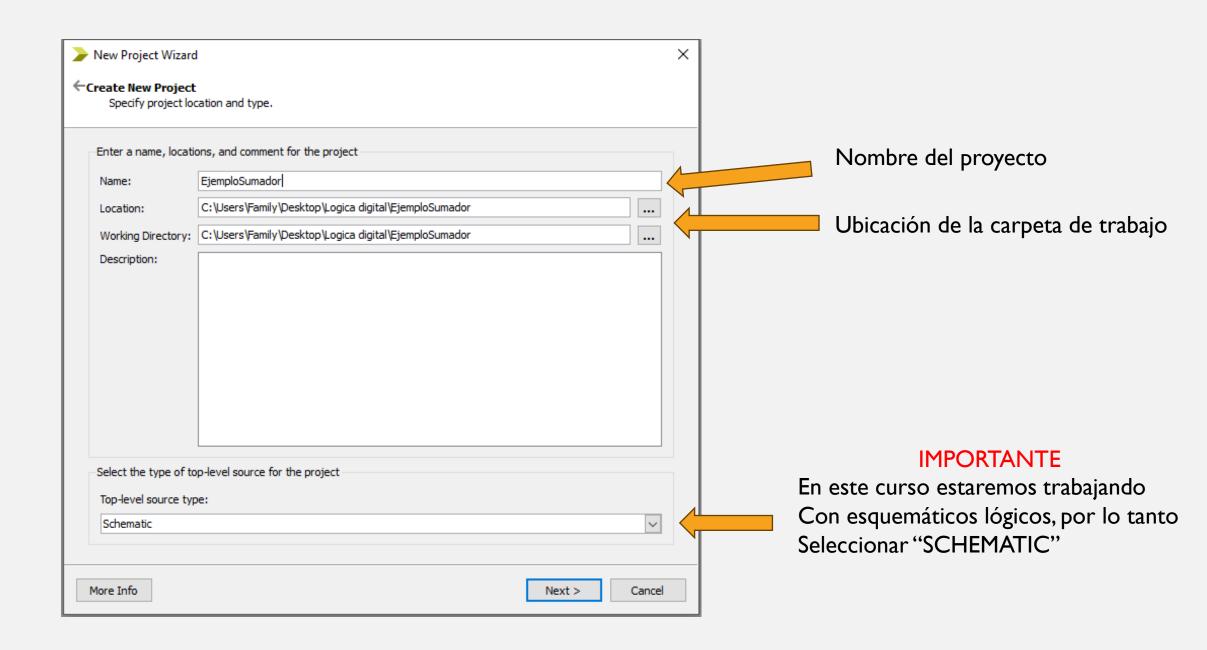


2. En el equipo del laboratorio encontraran esta aplicación

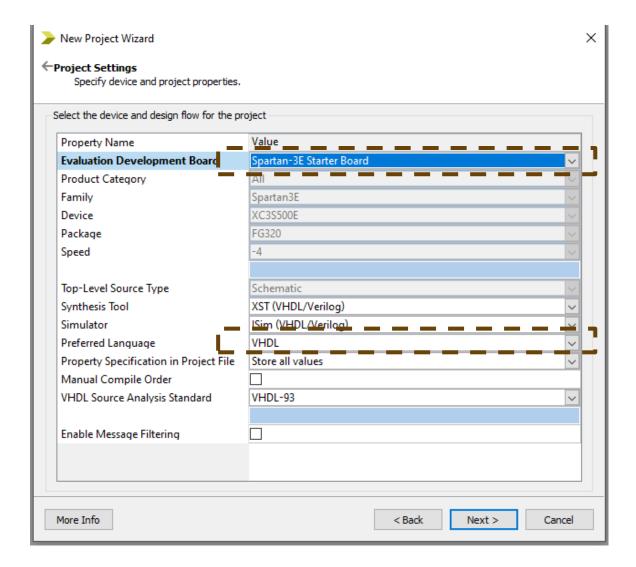


- 3.Despues de abrir el program
- dan click en File -> New Project... para crear un nuevo proyecto





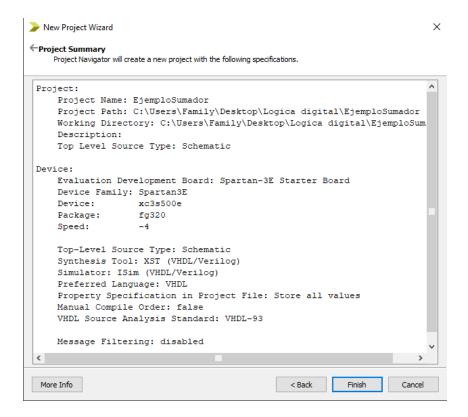
#### CREANDO UN NUEVO PROYECTO



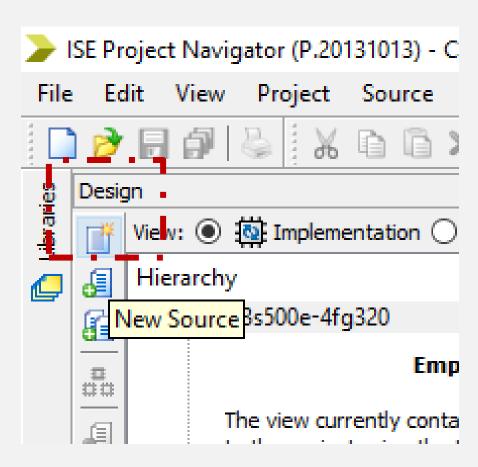
#### **IMPORTANTE**

En este curso estaremos trabajando Con la FPGA SPARTAN 3E STARTED BOARD y El lenguaje VHDL

## GENERARÁ UN RESUMEN DEL PROYECTO

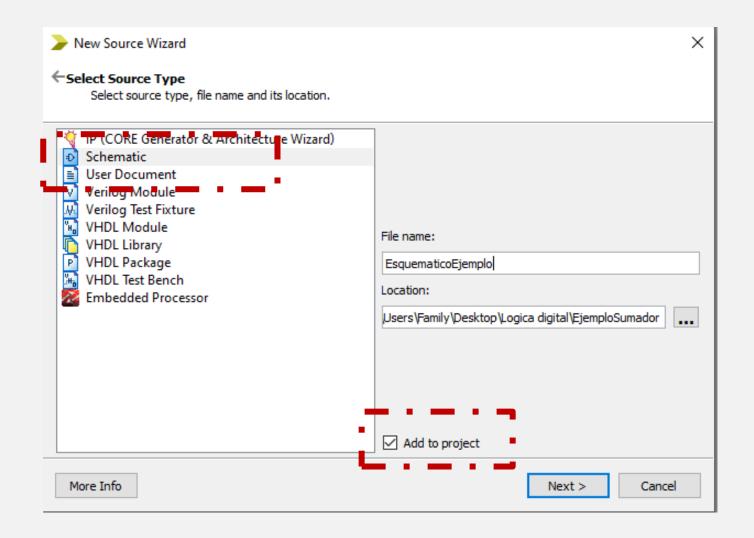


# EMPEZANDO A DISEÑAR EN XILINX



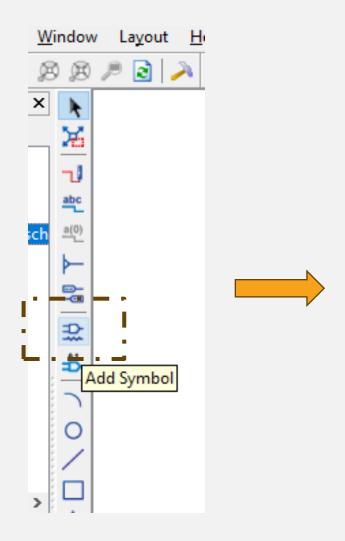
Una vez ya creado el área de trabajo vamos al apartado de "DESING" y creamos el esquema donde dice **New Source** 

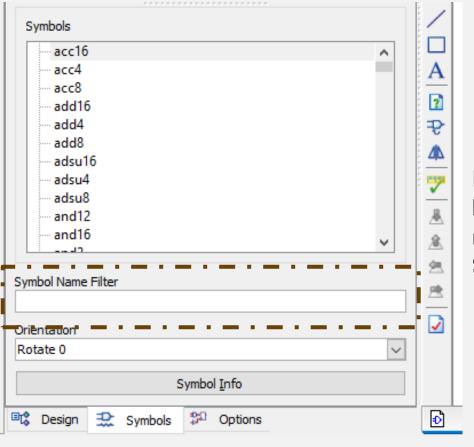
# EMPEZANDO A DISEÑAR EN XILINX



### GENERARÁ UN RESUMEN DE QUE SCHEMATIC ESTA CREADO

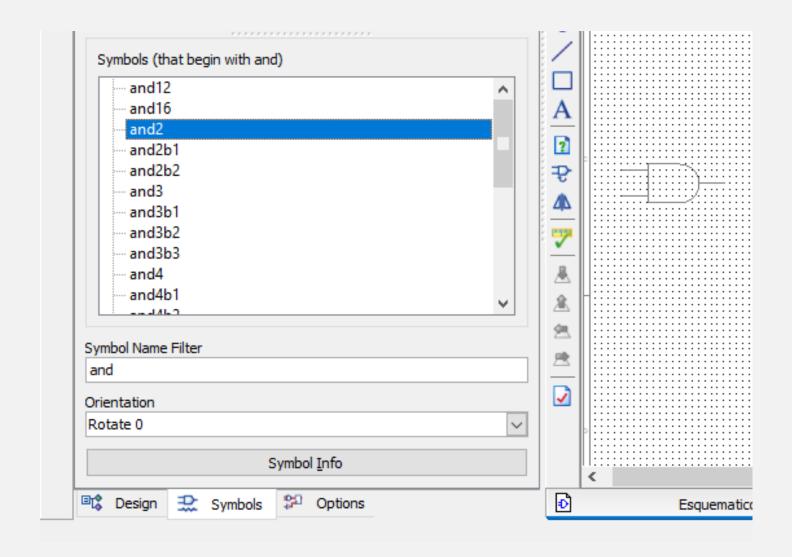
Damos click en **SCHEMATIC**, le definimos un nombre Y verificar que se agregue al proyecto



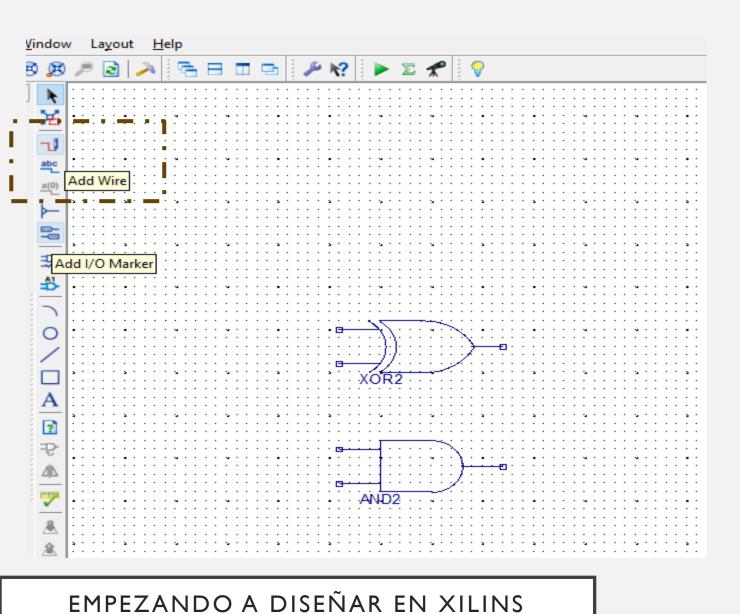


La mejor forma de buscar el símbolo que necesitas es por **Symbol name filter** 

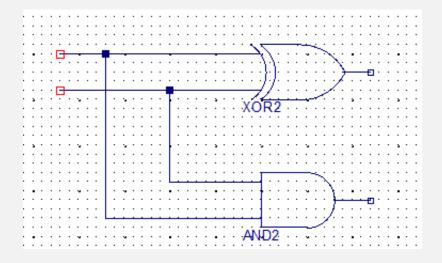
Para empezar a diseñar buscamos los símbolos en add symbol



El numero al lado del símbolo significa la cantidad de entradas que tendrá la compuerta y la letra después del numero significa la cantidad de entradas negadas

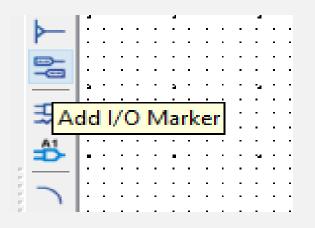


Para conectarlos damos click en **Add Wire** 

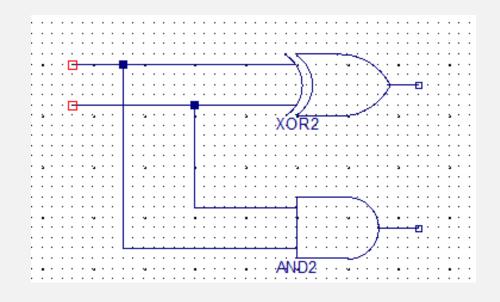


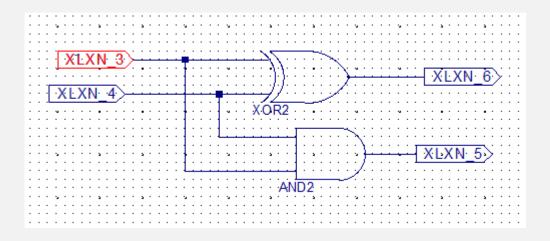
#### NOTA:

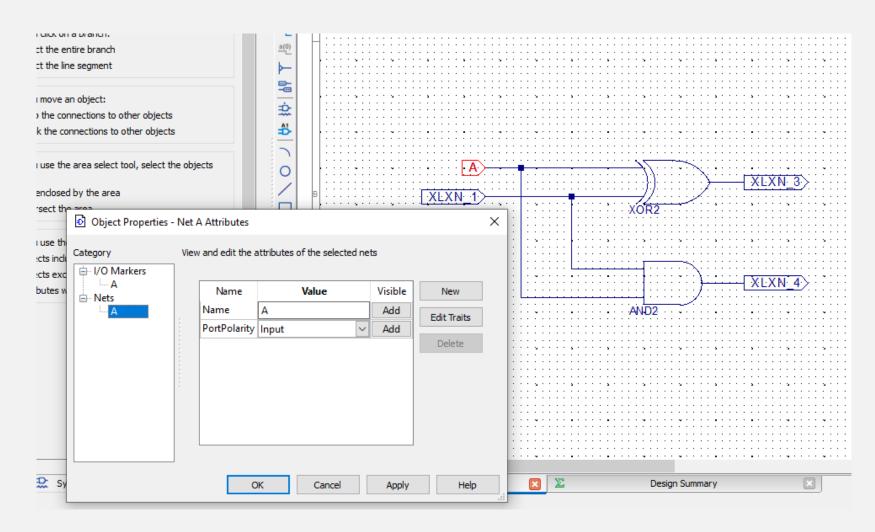
Analizar y estar seguro donde será la conexión De lo contrario, al querer quitar una conexión Puede eliminar cualquier conexión conectada A cada elemento



Una vez conectados definimos la entrada y salida Con **Add I/O Marker** 



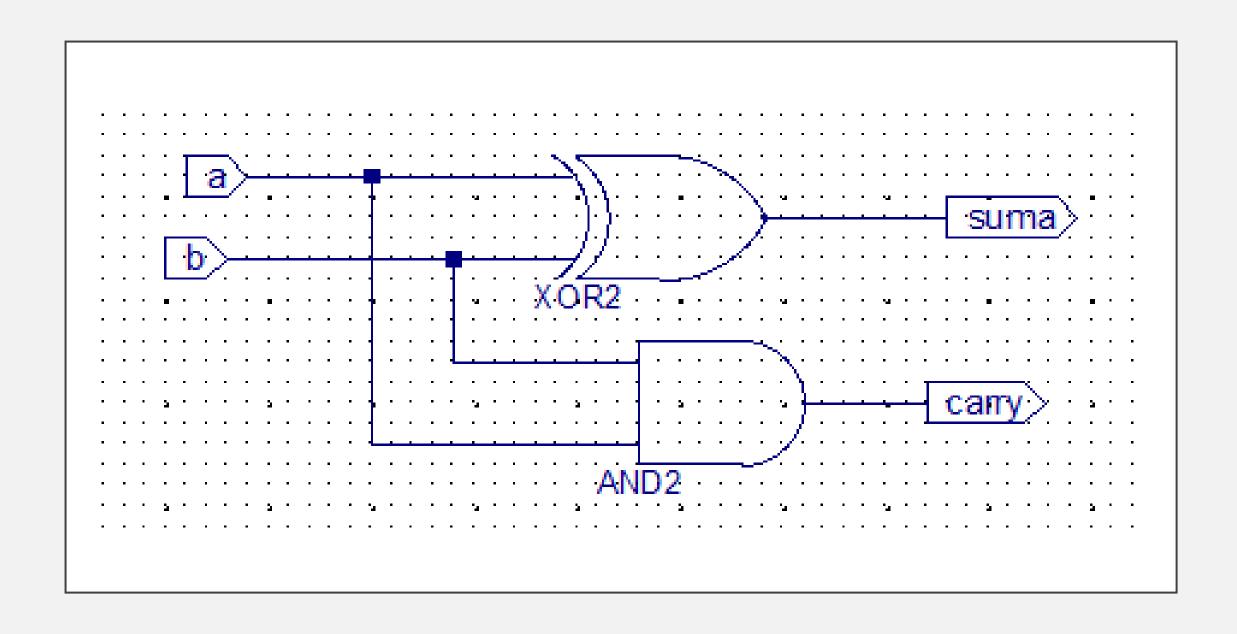




Posterior a ello damos doble click a una entrada/salida para definirle un nombre y direccion

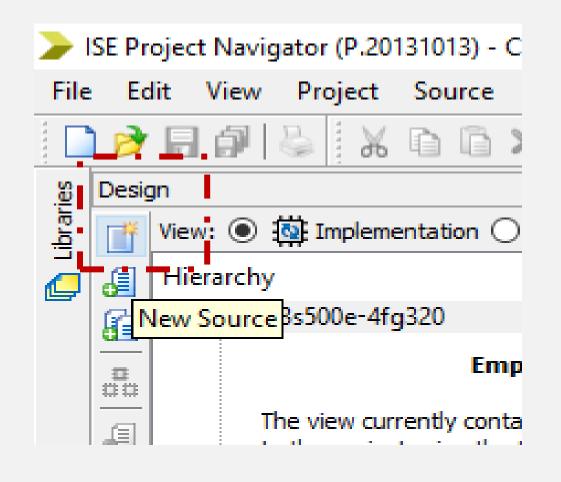
#### NOTA:

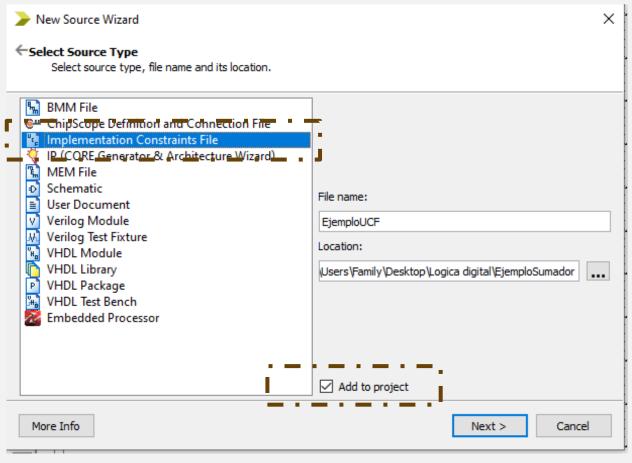
Es recomendable dejar los nombres en minúscula, sin espacios y sin ningún carácter diferente a las letras ya que estos nombres serán usados para programar

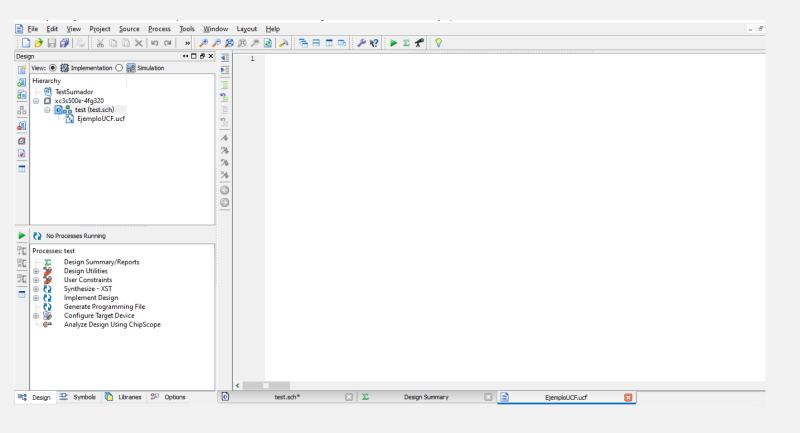


#### CREEMOS UN ARCHIVO UCF

Después de crear el esquemático, es necesario asociarlo con la FPGA. Para ello es necesario crear un archivo **UCF** 







Después de crear el UCF se abrirá una Notepad para ingresar los comandos para asignar las entradas físicas de la FPGA

#### NOTA:

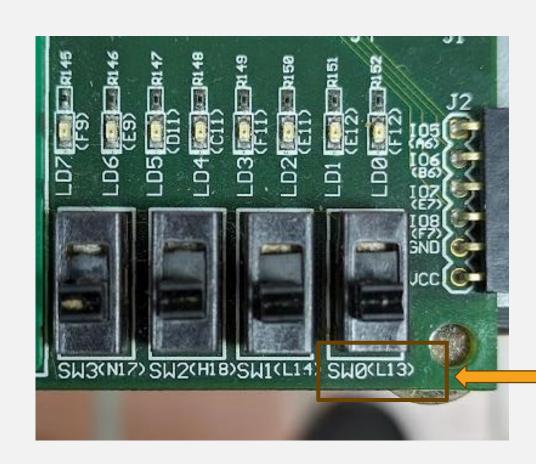
En intu, el profesor les brindará los comandos de asignación de pines,

**ASIGNANDO PINES** 

### DOS FORMAS DE SOLUCION

Solución I CLICK AQUI Solución 2 CLICK AQUI

## SOLUCIÓN I CON SWITCHES



Vamos a asignar estos switches como entradas

#### Recordemos:

Los switches están organizado de derecha a izquierda y se cuentan desde 0 a 3 y cada switch tiene su localización en paréntesis,

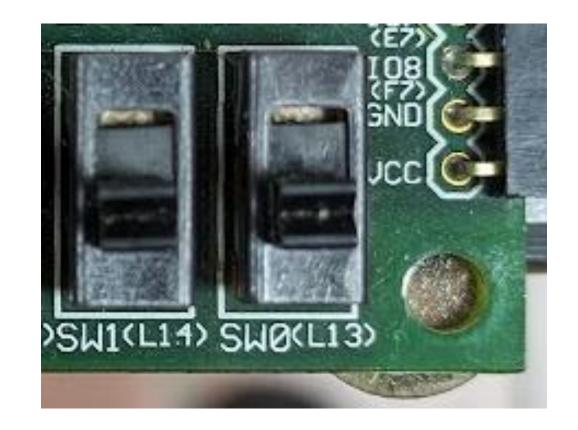
En este caso el switch 0 es el L13

## ASIGNANDO PINES DE ENTRADA

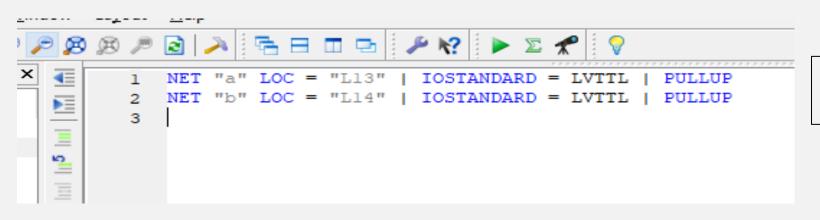
Usaremos estos dos switches para simular las entradas a y b

#### Recordemos:

La FPGA internamente esta interconectada con miles de circuitos, pero la mayoría de ellos están en un estado de "apagado" y la finalidad del UCF es activar el circuito especifico para cumplir con las condiciones dadas el SCHEMATIC



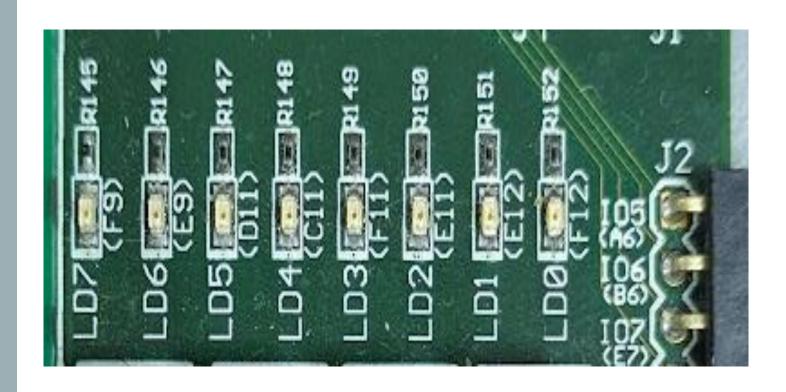
### ASIGNANDO PINES DE ENTRADA



En este caso estamos declarando en la FPGA los switches L13 y L14 como entrada a y b

## ASIGNANDO PINES DE SALIDA

Usaremos, los leds discretos para mostrar la solución



#### ASIGNANDO PINES DE SALIDA

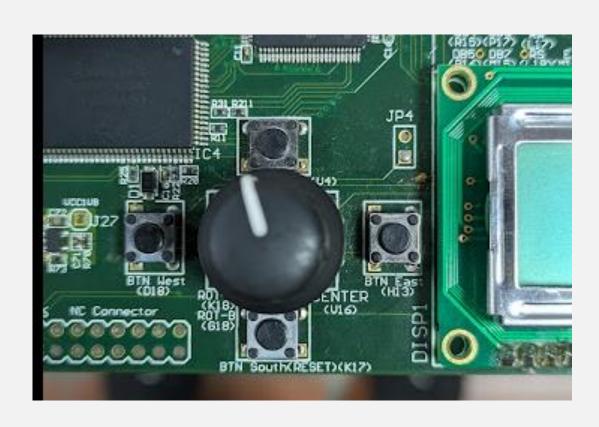
```
1  #pines de entrada (switch)
2  NET "a" LOC = "L13" | IOSTANDARD = LVTTL | PULLUP
3  NET "b" LOC = "L14" | IOSTANDARD = LVTTL | PULLUP
4  # pines de salida (leds discretos)
5  NET "suma" LOC = "F12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
6  NET "carry" LOC = "E12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
7  8
```

En este caso estamos declarando en la FPGA los leds F12y E12 como salida SUMA y CARRY

NET "carry" LOC = "E12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;



## SOLUCIÓN 2 CON BOTONES

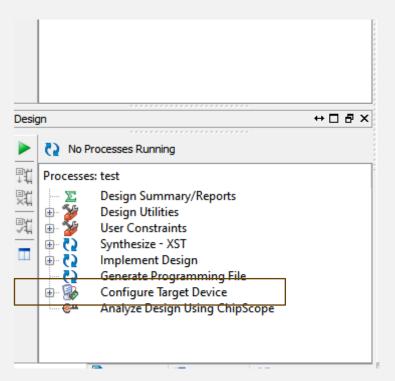


Vamos a asignar dos de estos botones como entradas

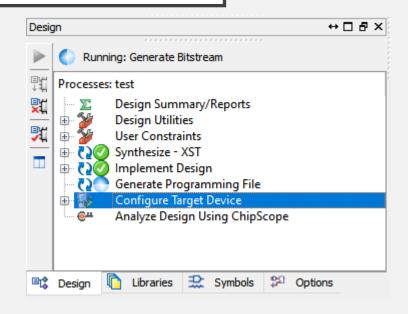
# ASIGNANDO PINES DE ENTRADA/SALIDA

```
NET "a" LOC = "H13" | IOSTANDARD = LVTTL | PULLDOWN;
NET "b" LOC = "K17" | IOSTANDARD = LVTTL | PULLDOWN;
# pines de salida (leds discretos)
NET "suma" LOC = "F12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
NET "carry" LOC = "E12" | IOSTANDARD = LVTTL | SLEW = SLOW | DRIVE = 8;
```

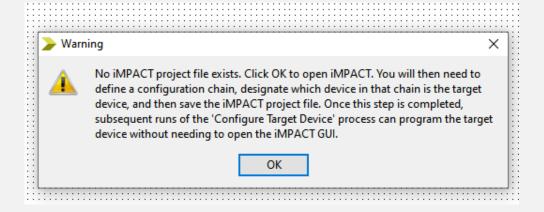
En este caso estamos declarando en la FPGA los pushbottons H13 y K17 como entrada a y b



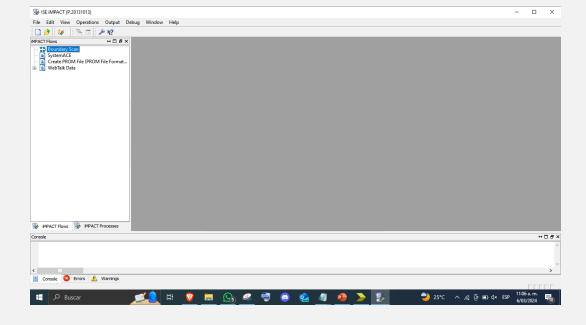
Para ejecutar el Schematic en la FPGA, vamos al apartado de DESING y damos doble click en Configure Target Device

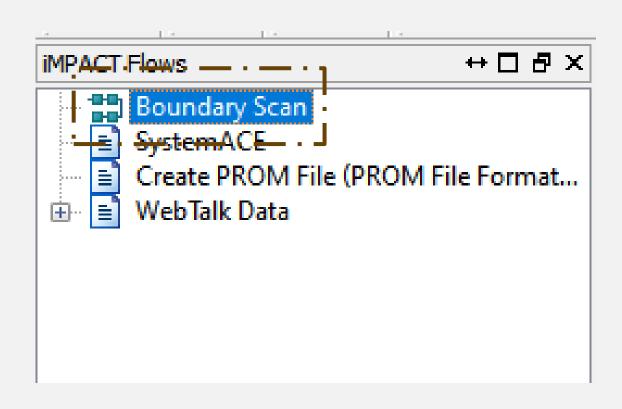


Internamente está validando la informacion y creando el archive binario

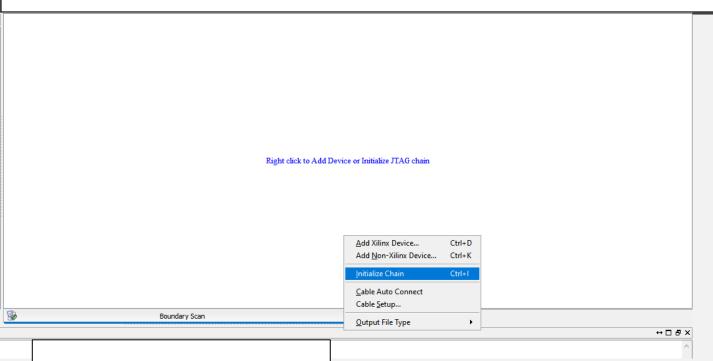


Si les sale este warning, no se preocupen, solo denle OK y se abrirá este programa

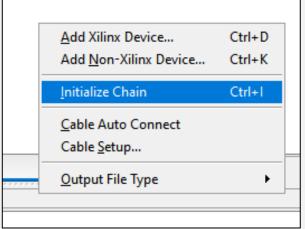




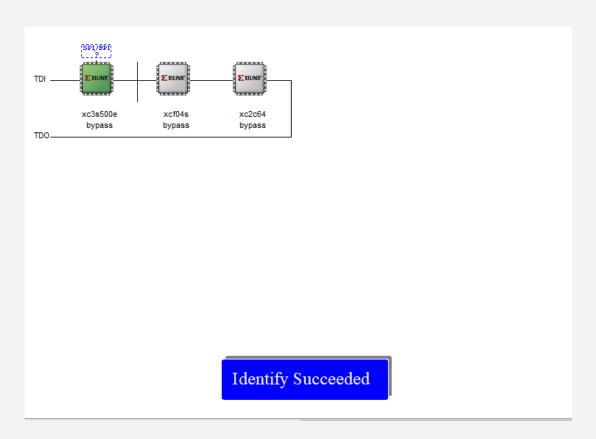
Damos click en **Boundary Scan** 



Despues de darle click se abrirá este recuadro

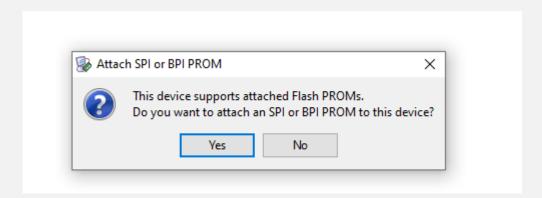


Dentro del cuadro, damos click derecho y le damos click en **Iniciate Chain** 

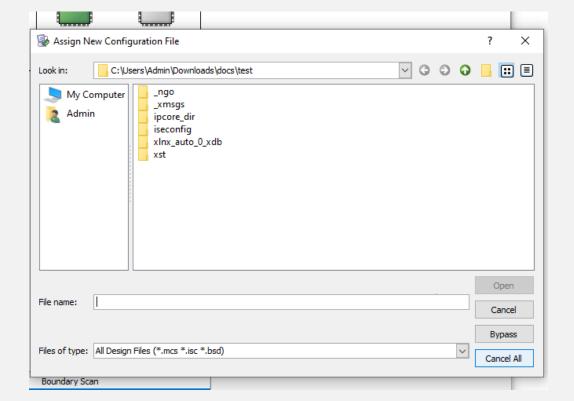


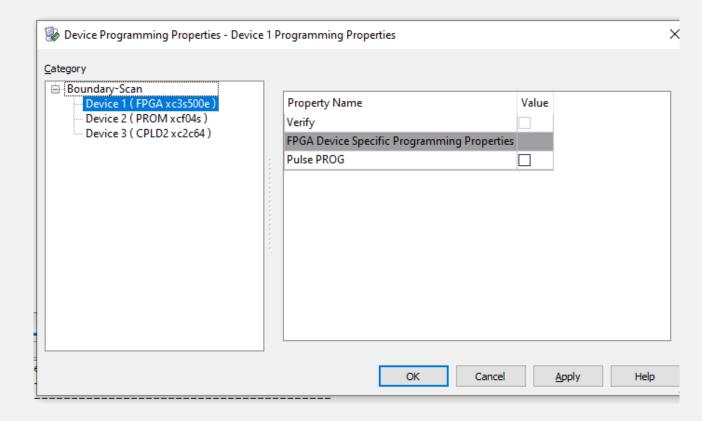
Al darle click se abrirá este cuadro, esto muestra los nucleos de procesamiento,

Le damos doble click al primer nucleo y se abrira un panel de archivos, **Abrimos el archivo.bit** 

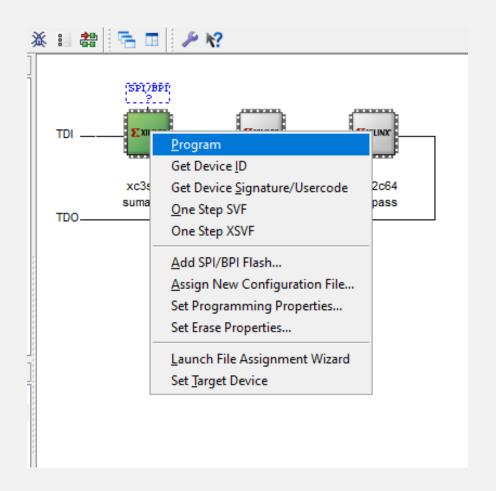


# Le damos click a **NO** y luego en **CANCEL ALL**





Le damos click en **OK** 



Finalmente le damos click derecho y le damos en **PROGRAM** y listo ya se puede ver en la FPGA el funcionamineto

# **TALLER**