



## GUÍA DE LABORATORIO 7

**Curso:** Microprocesadores y Sistemas Embebidos

**Semestre:** 2023-II

**Semana:** 11

**Resultado del estudiante:** Herramientas Modernas (K)

### 1. RESULTADOS DE APRENDIZAJE

- Selecciona y utiliza el lenguaje de descripción de hardware (VHDL ó Verilog HDL) en el software Quartus Prime en la solución de problemas de diseño.

### 2. PRODUCTO ACADÉMICO

- Circuito Digital implementado en CPLD MAX II EPM240 mediante programación de hardware (VHDL ó Verilog HDL).

### 3. RECURSOS PARA EMPLEAR

#### 3.1. Materiales:

- Tarjeta CPLD ALTERA U119 MAX II EPM240
- ALTERA mini USB BLASTER
- Adaptador DE 5V para la tarjeta CPLD
- 8 resistencias de 220 ohmios
- 8 resistencias de 1k (ó 10 k)
- 8 LEDs
- 4 pulsadores, 4 switchs (o dip switchs)
- Protoboard y cableado correspondiente.

#### 3.2. Equipos y Software:

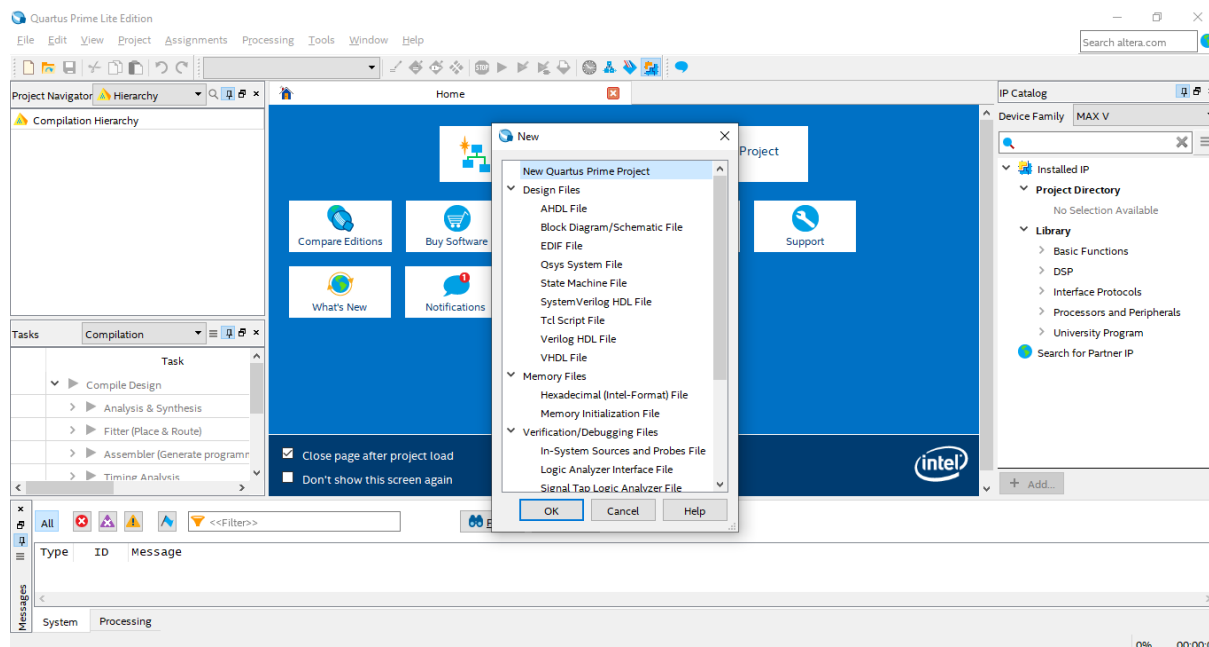
- Quartus Prime 20.1
- ModelSim 20.1



## 4. PRIMERA PARTE: CONFIGURACIÓN DE SOFTWARE

### *Paso 1:*

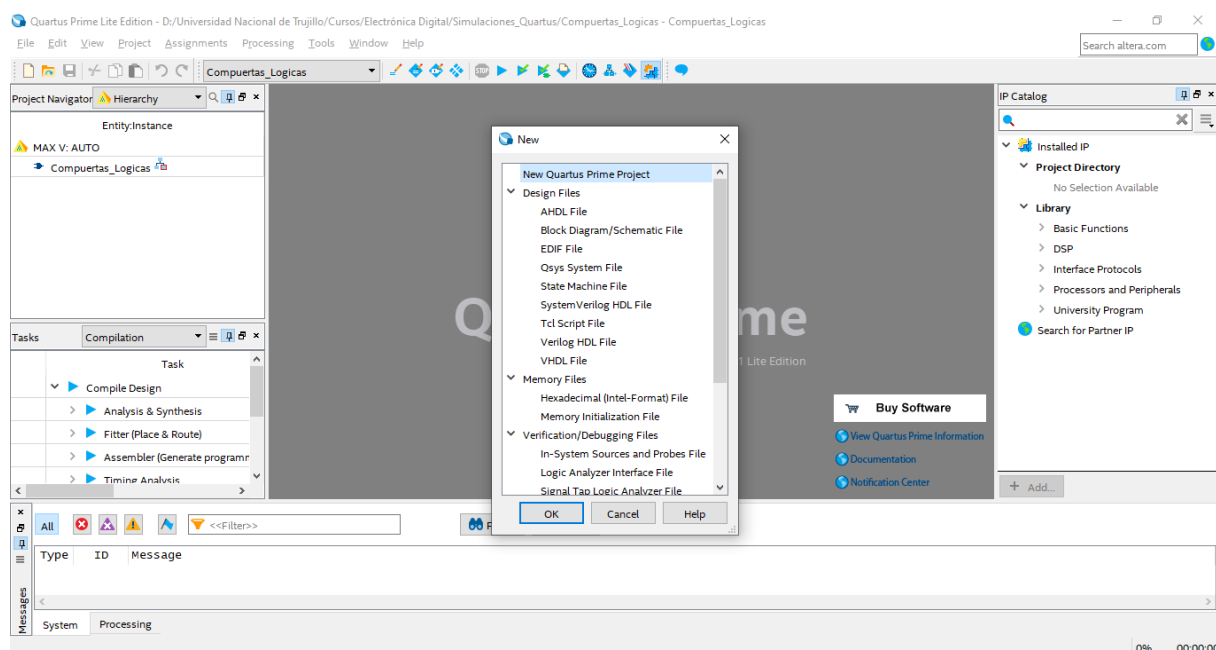
Abrir el software Quartus Prime Lite Edition y seleccionar la opción **New**.



En la ventana que se muestra, seleccionar la opción **New Quartus Prime Project** para crear un nuevo proyecto.

### *Paso 2:*

Luego de haber creado el proyecto, hacer click nuevamente en **New** y seleccionar **VHDL File**.





**Paso 3:**

En el espacio creado (\*.vhd), escribir el siguiente código:

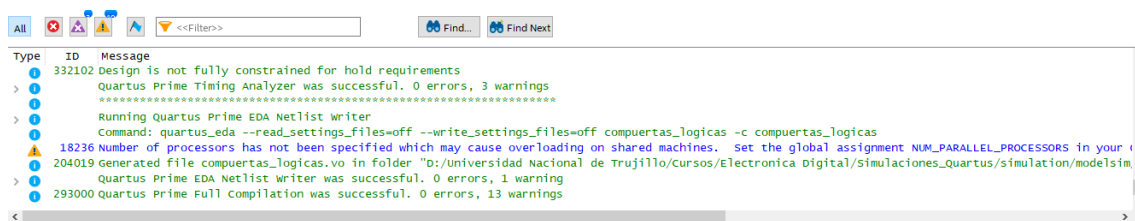
```
— Codigo para un Inversor
entity compuertas_logicas is
    port(A:in bit; X:out bit);
end entity;

architecture inversor of compuertas_logicas is
    begin
        X <= NOT A;
end architecture inversor;
```

Tener en cuenta que en lugar de la palabra *compuertas\_logicas* se deberá colocar el nombre que se le asignó al proyecto.

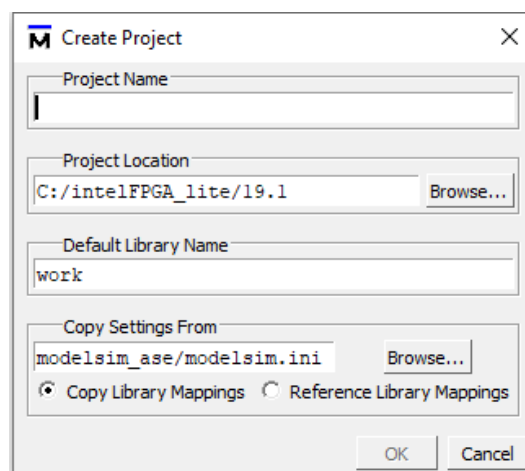
**Paso 4:**

Proceder a compilar el proyecto seleccionando **Start Compilation**, deberá resultar una compilación correcta en la ventana de mensajes, tal como se muestra en la figura:



**Paso 5:**

A continuación, abrir el software ModelSim y crear un nuevo proyecto (**File > New > Project...**), deberá de aparecer la siguiente ventana:



Proceder a asignar nombre y locación al Proyecto (asegurarse que ningún elemento de la ruta posea caracteres poco comunes como tildes). No modificar los 2 últimos espacios.

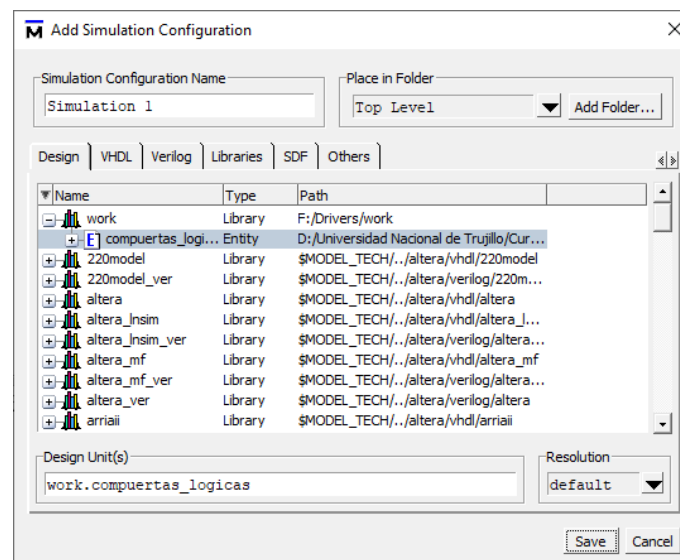


### Paso 6:

En la ventana que aparece, elegir **Add Existing File** y buscar la ruta del archivo \*.vhd donde se encuentra nuestro código VHDL. El archivo deberá aparecer en la ventana **Project**. A continuación, seleccionar el archivo subido, hacerle click derecho y seleccionar **Compile Selected**. En la ventana **Transcript** deberá aparecer un mensaje en verde que señale que la compilación fue exitosa.

### Paso 7:

Luego, hacer click derecho al archivo subido y en **Add to Project** seleccionar **Simulation configuration...**, deberá aparecer la siguiente ventana:

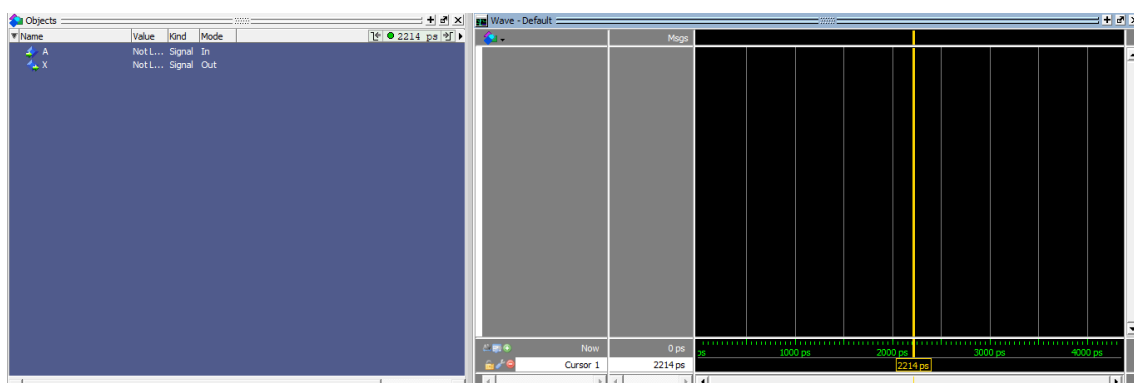


En la pestaña **Design** buscar **work** y ahí el nombre del archivo subido, seleccionarlo y presionar **Save**.

### Paso 8:

En el archivo **Simulation 1** que ha de haber aparecido hacer doble click o click derecho y seleccionar **Execute**.

En caso las ventanas **Objects** y **Wave** no aparezcan, buscarlas y seleccionarlas en el menú **View**. De preferencia, tener la pantalla como se muestra a continuación:





**Paso 9:**

Arrastrar los objetos a la ventana **Wave**. A continuación, asignar algún valor al objeto A, mediante el siguiente comando en la ventana **Transcript**:

**force A 1**

Luego ejecutar 1000 ps de simulación, escribiendo el comando:

**run 1000**

Evaluar el resultado, y proceder a hacer otras simulaciones (para otras compuertas lógicas) siguiendo el mismo proceso.

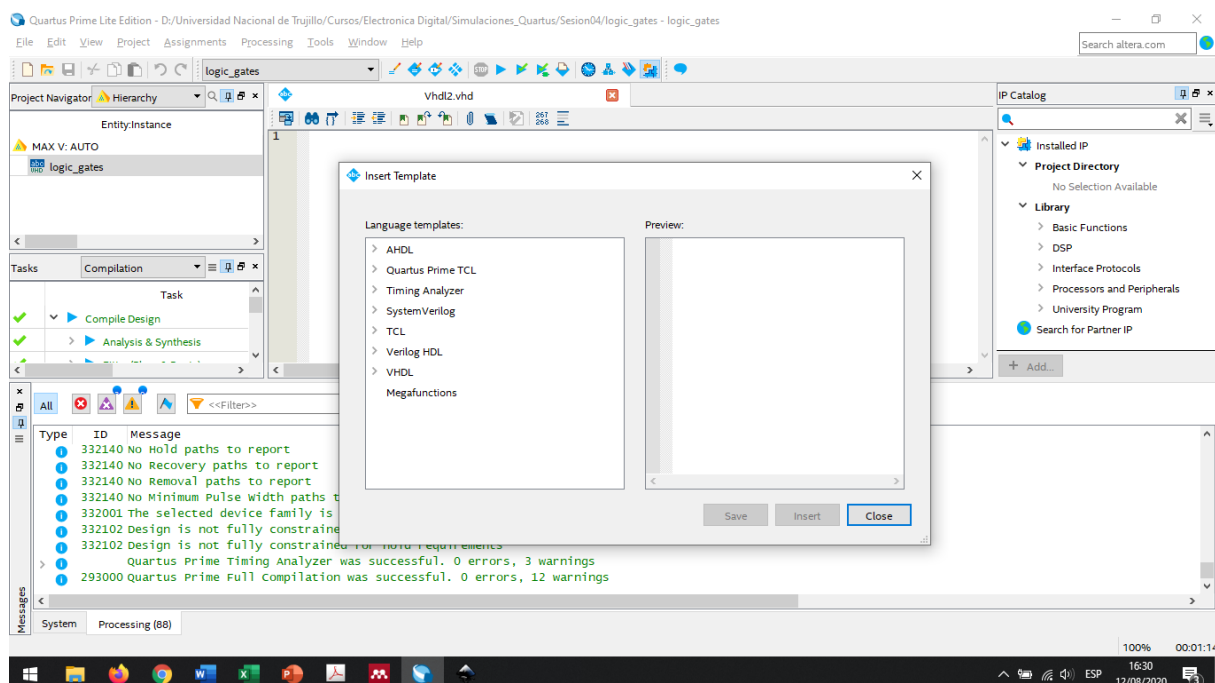
## 5. SEGUNDA PARTE: CIRCUITO DIGITAL EN HDL

**Paso 1:**

Crear un proyecto en el software Quartus Prime.

**Paso 2:**

Crear un archivo VHDL en el proyecto mencionado. A continuación se insertará una plantilla (template) al archivo mencionado, para este propósito ir al menú **Edit** y seleccionar la opción **Insert Template...**, deberá mostrarse la ventana que se encuentra en la siguiente figura:



**Paso 3:**

Seguir la siguiente ruta:

**VHDL ~ Full Designs ~ Arithmetic ~ Adders**



Seleccionar la opción **Unsigned adder**, de esta manera se creará el código para un sumador sin signo, el cuál debería ser como el que se detalla a continuación:

```
— Quartus Prime VHDL Template
— Unsigned Adder

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity unsigned_adder is

    generic
    (
        DATA_WIDTH : natural := 8
    );

    port
    (
        a : in unsigned ((DATA_WIDTH-1) downto 0);
        b : in unsigned ((DATA_WIDTH-1) downto 0);
        result : out unsigned ((DATA_WIDTH-1) downto 0)
    );

end entity;

architecture rtl of unsigned_adder is
begin

    result <= a + b;

end rtl;
```

Como se mencionó anteriormente, se debe cambiar el nombre de la entidad (**entity**) para que la compilación sea exitosa.

Cabe mencionar que el código importado se trata de un sumador de 8 bits, el código permite variar la cantidad de bits a utilizar.

#### **Paso 4:**

Proceder a realizar la simulación del código anterior en el software ModelSim.

#### **Paso 5:**

A continuación, realizaremos un procedimiento similar. Sin embargo, el código obtenido y simulado corresponderá al lenguaje de descripción Verilog HDL.



Creamos un archivo Verilog HDL (**Verilog HDL File**) en el proyecto mencionado e insertamos la plantilla correspondiente al sumador sin signo, para lo cual seguimos la siguiente ruta:

**Verilog HDL ~ Full Designs ~ Arithmetic ~ Adders**

Agregando de la misma manera la opción **Unsigned Adder**. Deberá mostrarse el siguiente código:

```
// Quartus Prime Verilog Template
// Unsigned Adder

module unsigned_adder
#(parameter WIDTH=16)
(
    input [WIDTH-1:0] dataa ,
    input [WIDTH-1:0] datab ,
    input cin ,
    output [WIDTH:0] result
);

    assign result = dataa + datab + cin;

endmodule
```

**Paso 6:**

Para que la comparación con el código en VHDL sea adecuada, se deberá cambiar la cantidad de bits considerados por el código Verilog HDL, cambiándose de 16 a 8. También se podría omitir el acarreo, ya que el código VHDL no lo considera.

**Paso 7:**

Compile el código y realice su simulación en ModelSim, de esta manera podrá realizar una comparación y llegar a una conclusión.

**Paso 8:**

Repita los anteriores pasos, con el propósito de compilar y simular los siguientes circuitos, tanto en lenguaje VHDL como Verilog HDL:

- **Signed Adder**
- **Unsigned Adder/Subtractor (Addsub)**
- **Signed Adder/Subtractor (Addsub)**