

# Procedimiento para implementar un procesador NIOS II en un FPGA Cyclone II

Por: Kalun José Lau Gan

2021

1

## Revisión (changelog) del documento:

- 15-mayo-2020: Redacción inicial del documento
- 08-mayo-2021: Se añadió información de procedimiento cuando se instancia el NIOS II como componente para otro diseño
- 22-mayo-2021: Se añadió mas detalles en la parte de conexión hacia el NIOS II en el Eclipse

2

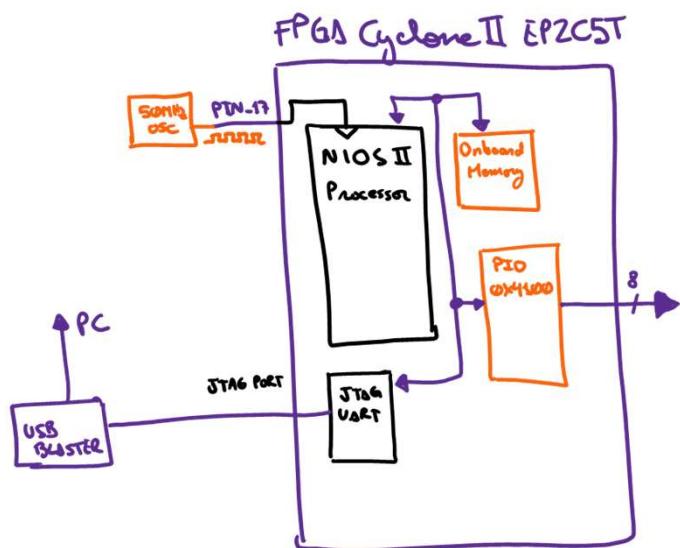
## Agenda:

- Procedimiento para implementar el NIOS II
- Ejemplo de aplicación usando Eclipse y lenguaje C
- NIOS II llamado como componente en VHDL

3

## Aspectos iniciales:

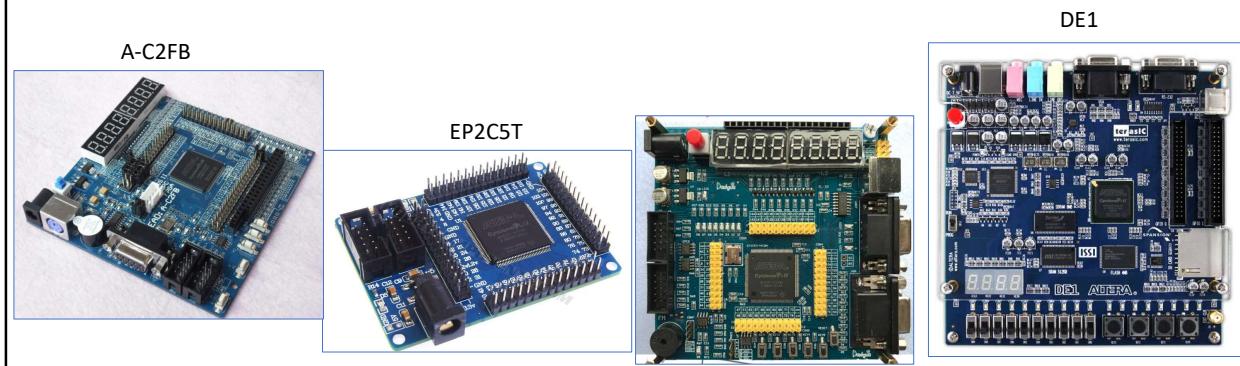
- Tener en cuenta que la implementación del NIOS II en el presente documento será en una FPGA Cyclone II EP2C5T.
- Se empleará el Quartus II versión 13.0 SP1 que es el último que soporta los FPGA Cyclone II.
- Se utilizará recursos del mismo FPGA para implementar una memoria para el NIOS II.
- Como ejemplo se implementará el NIOS II con un puerto PIO con dirección 0x4100, de 8 bits y como salida.
- Se empleará el lenguaje C para la aplicación que correrá el NIOS II



4

## Aspectos iniciales

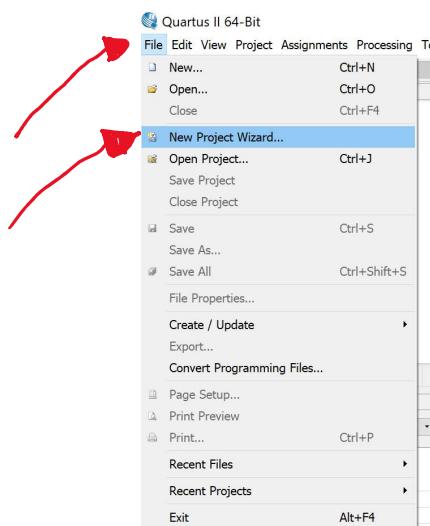
- Existen diversas tarjetas de desarrollo empleando el FPGA Cyclone II EP2C5T.
- Cada tarjeta tendrá (o carecerá de) periféricos externos conectados al FPGA de manera particular por lo que se tendrá que consultar la hoja técnica de cada una de dichas tarjetas para ver sus conexiones.



5

## Procedimiento para implementar el NIOS II

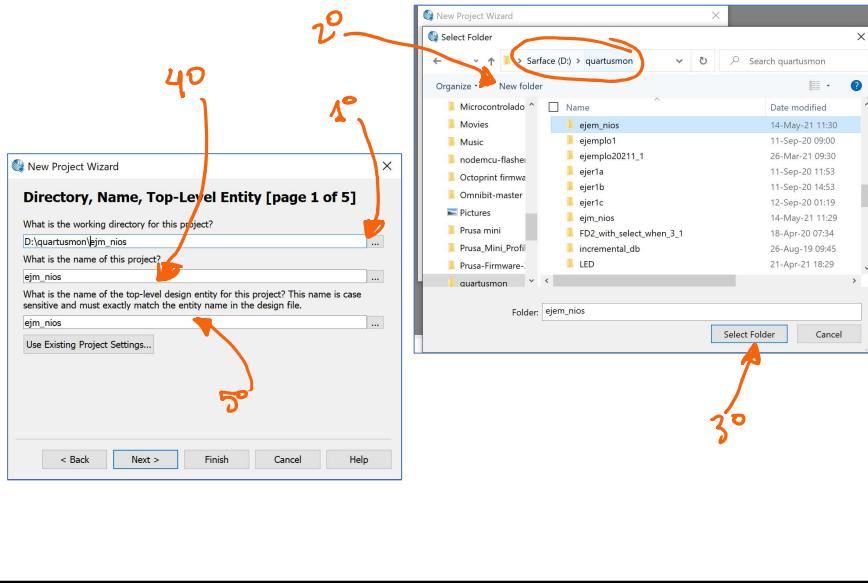
- Usar el Proyect Wizard para crear un proyecto en el Quartus II



6

## Procedimiento para implementar el NIOS II

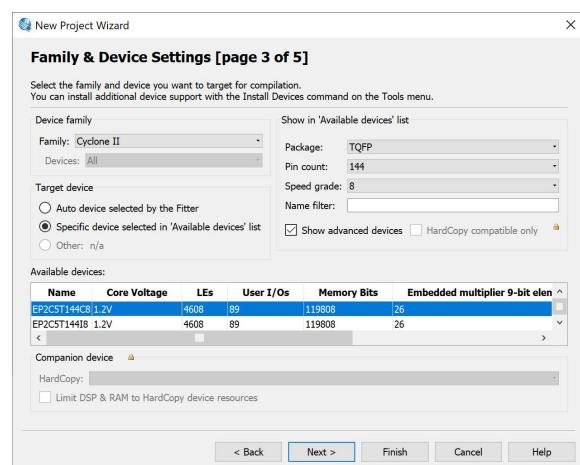
- De preferencia crear una carpeta de manera manual para alojar todo el proyecto.
- Tener en cuenta que la ruta de la carpeta del proyecto sea fácil de encontrar y no debe de haber espacios tanto en la ruta como el nombre del proyecto.
- Otra consideración a tener en cuenta es que el nombre de la carpeta coincide con el nombre del proyecto y también del top-level.
- Tener en cuenta que se estará empleando un mismo nombre para el proyecto, top-level y plataforma NIOS II (en QSys). Esto para evitar instanciar el procesador y facilitar la implementación del ejemplo.



7

## Procedimiento para implementar el NIOS II

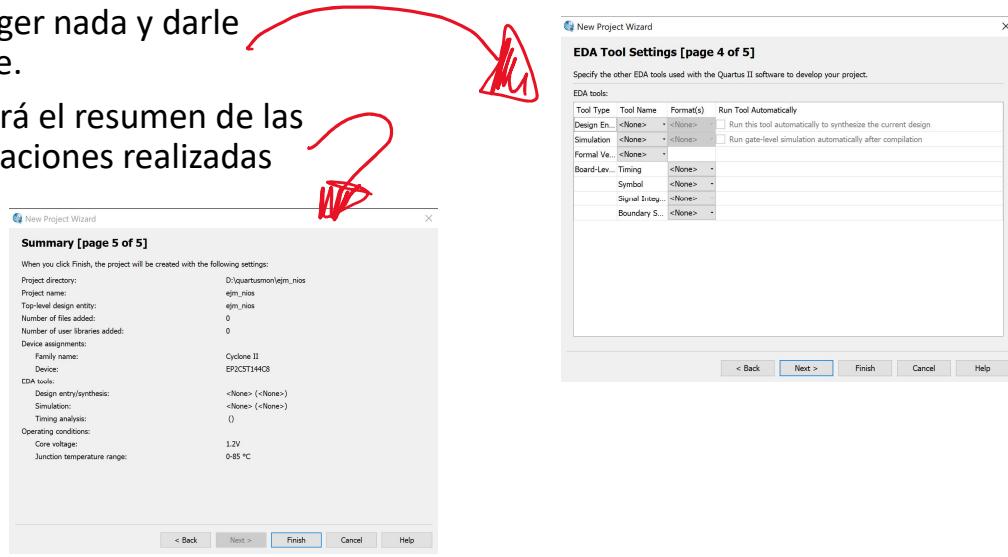
- Escoger el dispositivo FPGA correcto, en este ejemplo usaremos el Cyclone II EP2C5T144C8N



8

## Procedimiento para implementar el NIOS II

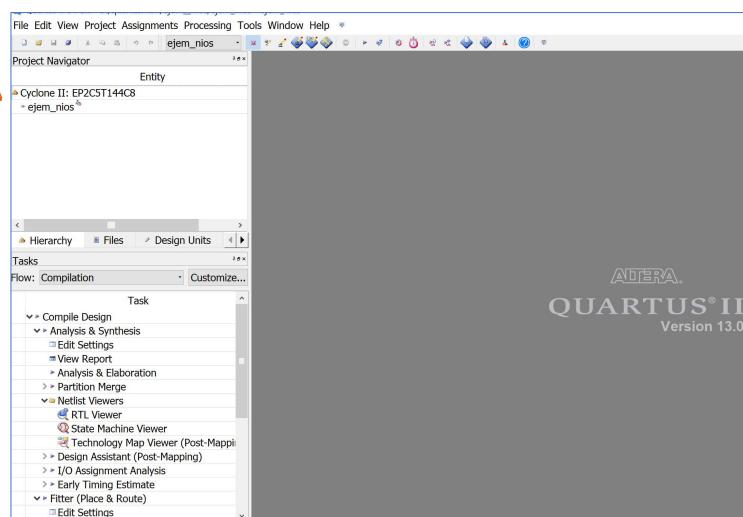
- No escoger nada y darle siguiente.
- Aparecerá el resumen de las configuraciones realizadas



9

## Procedimiento para implementar el NIOS II

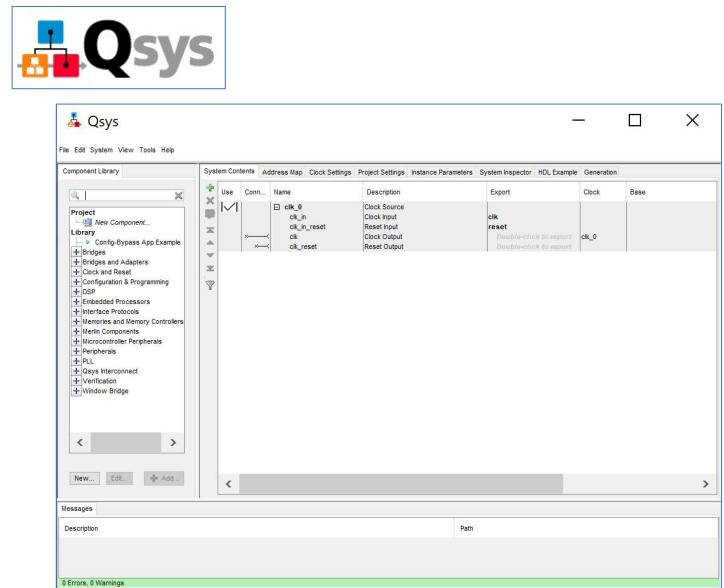
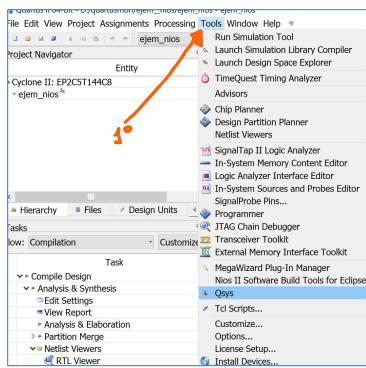
- Al retornar a la ventana principal del Quartus II se deberá verificar en la ventana "Project Navigator" si se escogió el dispositivo correcto y el nombre del top-level



10

## Procedimiento para implementar el NIOS II

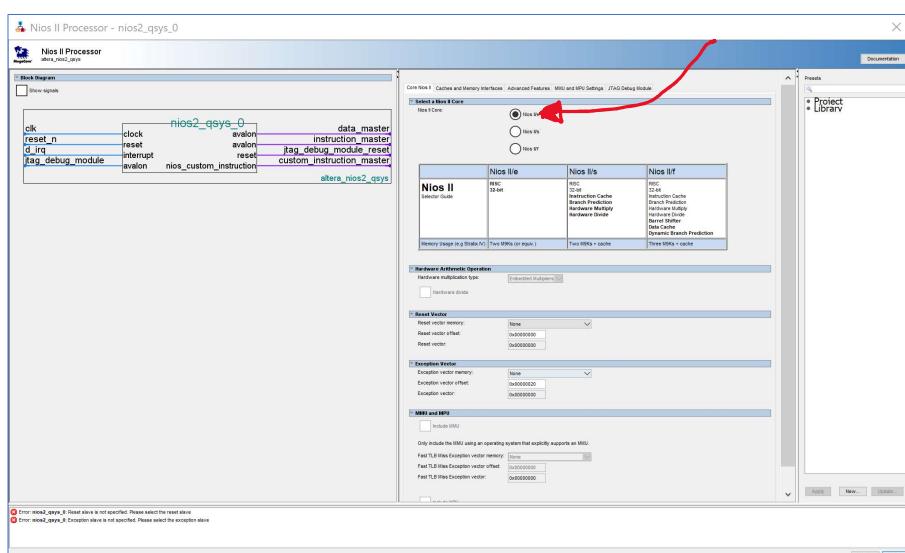
- Abrimos el Qsys para construir la plataforma de procesador NIOS II



11

## Procedimiento para implementar el NIOS II

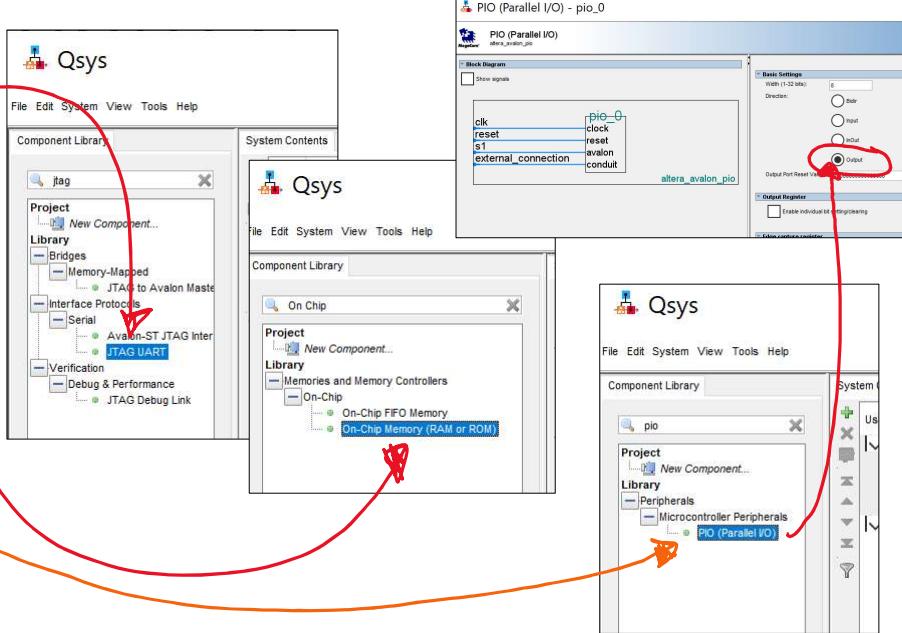
- Agregar el procesador NIOS II desde la ventana de la izquierda.
- En las opciones de la ventana escoger el NIOS II/e
- El Reset Vector y Exception Vector se configurarán después de instalar una memoria en QSys.



12

## Procedimiento para implementar el NIOS II

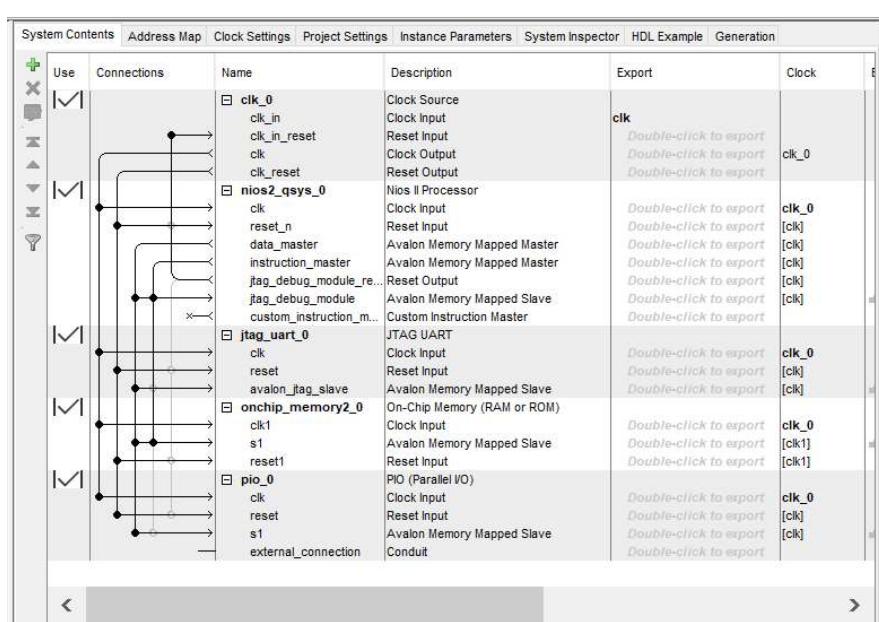
- Agregar el JTAG UART (configuración por defecto)
- Agregar On-Chip Memory (configuración por defecto)
- Agregar PIO, verificar que sea de ancho de 8 bits y colocarlo como salida



13

## Procedimiento para implementar el NIOS II

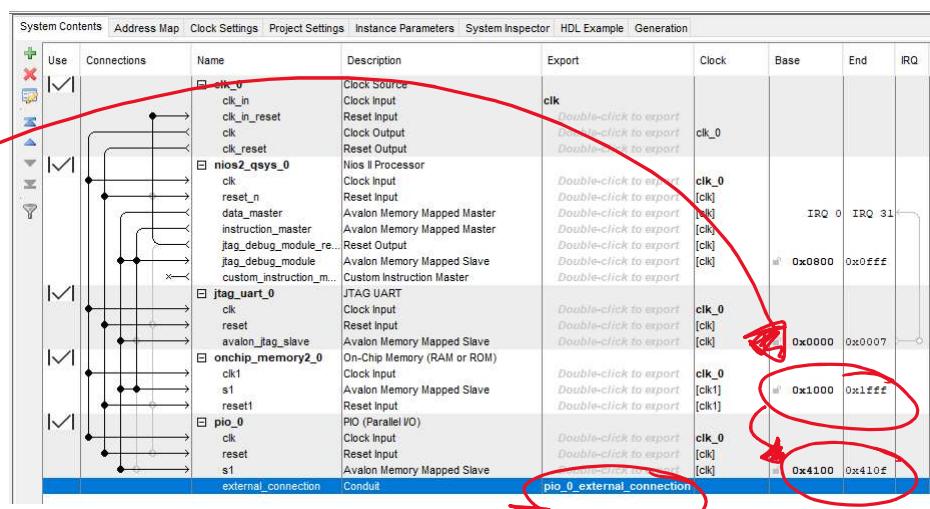
- Hacer las conexiones de los componentes de la plataforma NIOS (reloj principal, reset, buses Avalon: datos e instrucciones)



14

## Procedimiento para implementar el NIOS II

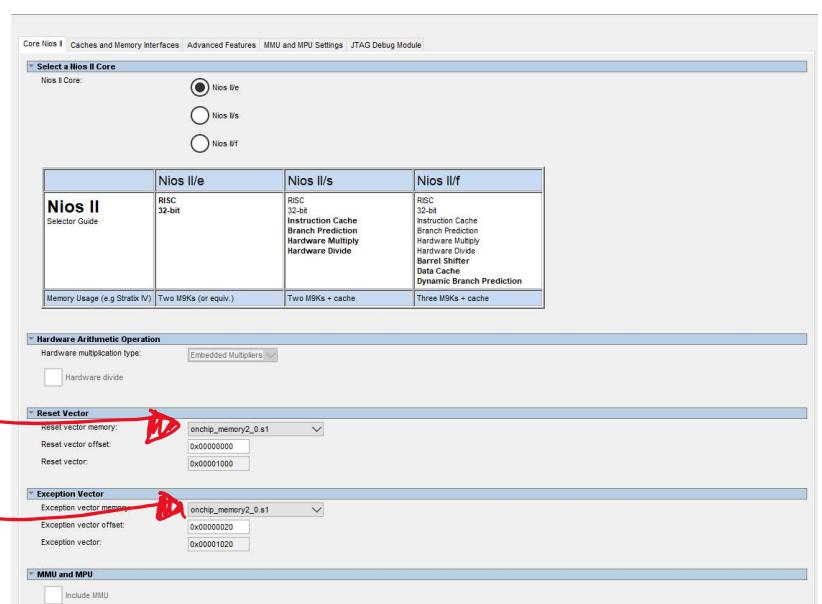
- Hacer el mapeo de direcciones para que no haya conflicto de recursos
- Definir nombre de exportación del puerto de salida PIO



15

## Procedimiento para implementar el NIOS II

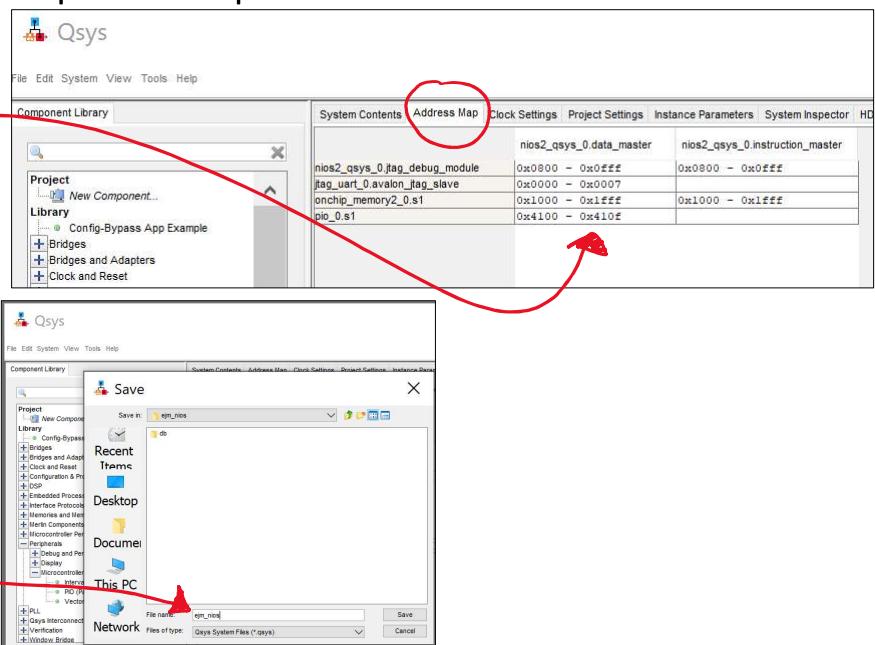
- Seleccionar onchip\_memory tanto en Reset Vector como en Exception Vector



16

## Procedimiento para implementar el NIOS II

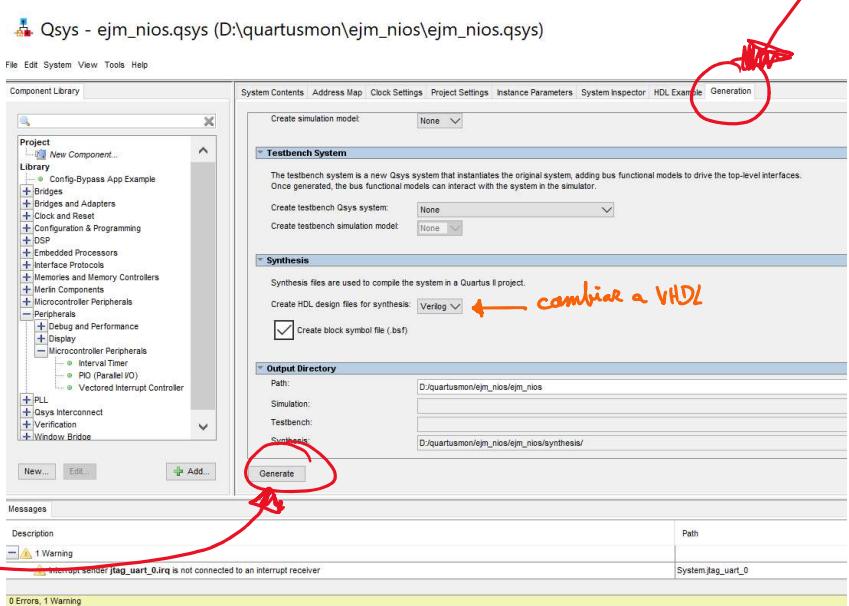
- Revisar mapeo de memoria
- Grabar lo realizado usando el mismo nombre del top-level



17

## Procedimiento para implementar el NIOS II

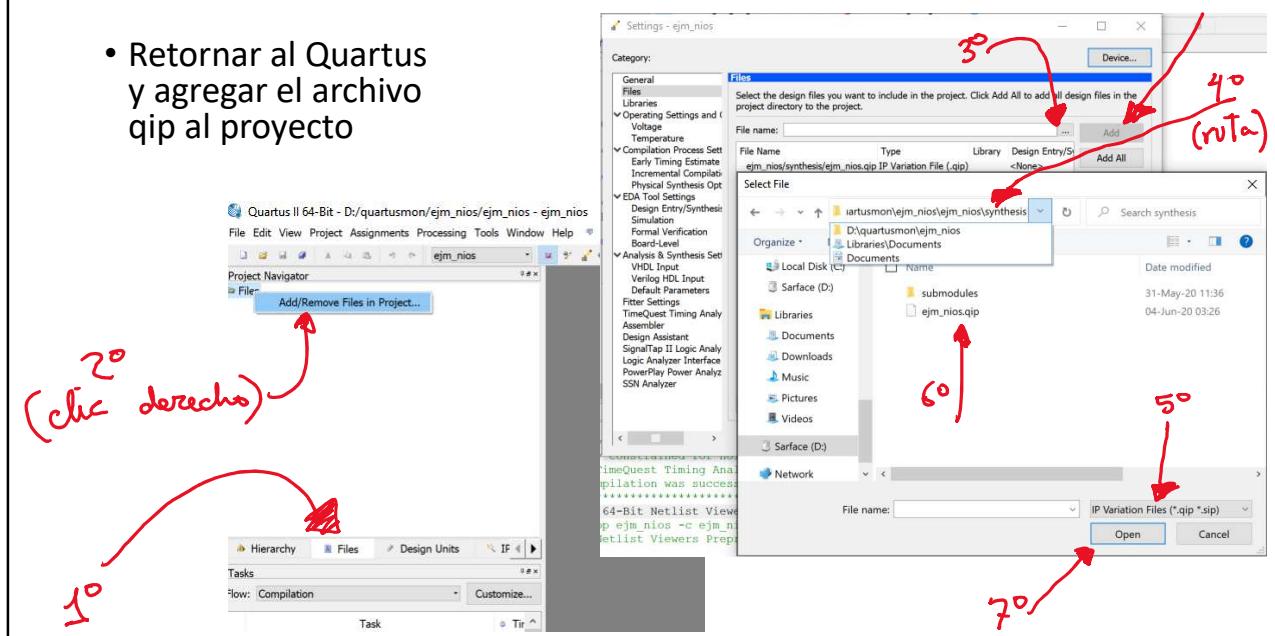
- Dirigirse a la última pestaña (Generation) y pulsar el botón "Generate", esto construirá el sopcinfo necesario para el Eclipse.
- Nótese que no debe de salir error alguno al término de la generación



18

## Procedimiento para implementar el NIOS II

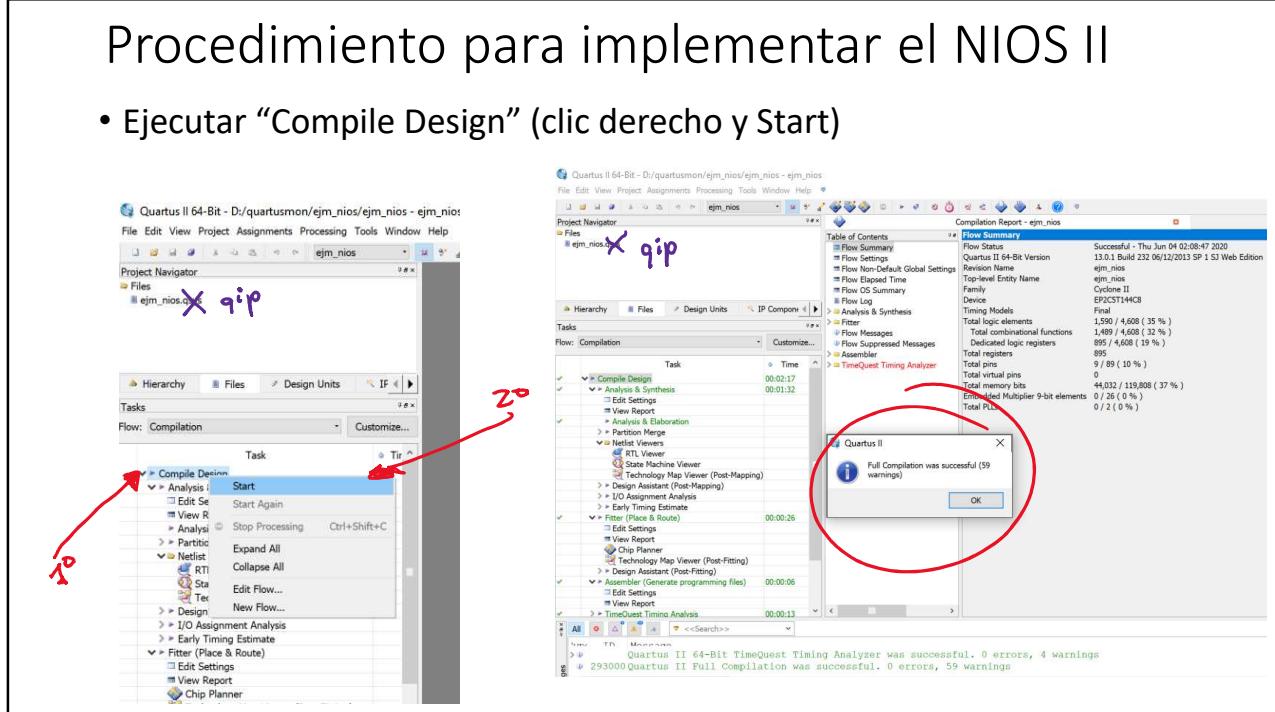
- Retornar al Quartus y agregar el archivo qip al proyecto



19

## Procedimiento para implementar el NIOS II

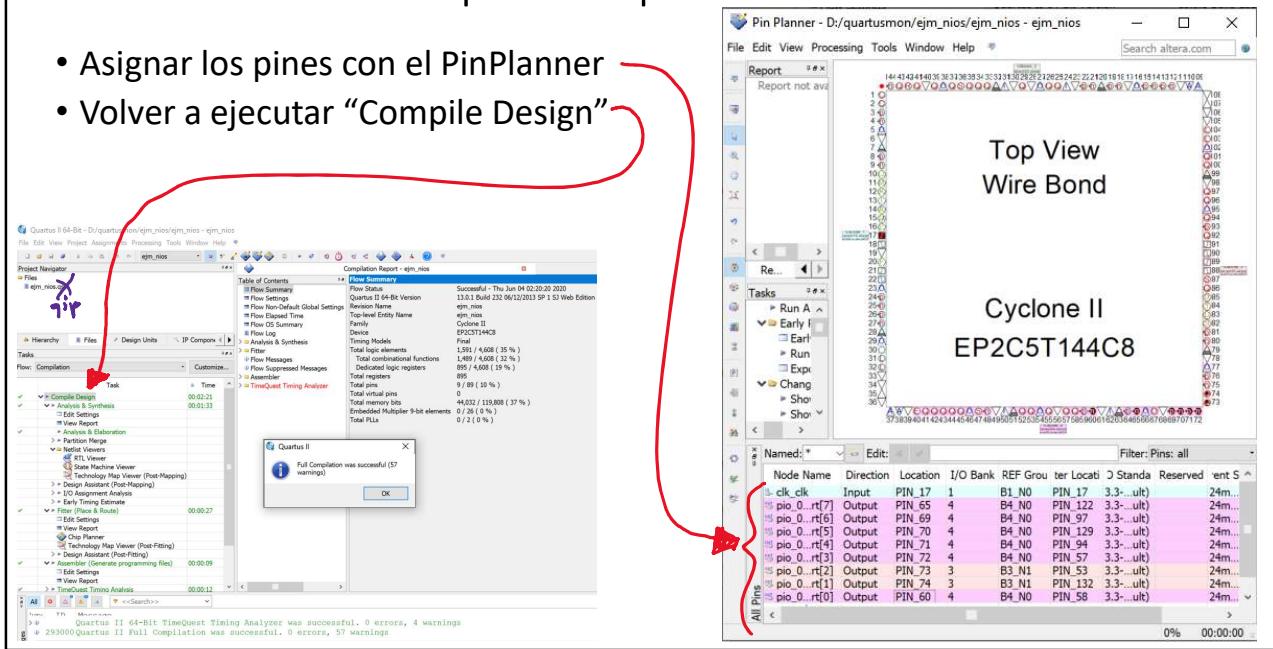
- Ejecutar “Compile Design” (clic derecho y Start)



20

## Procedimiento para implementar el NIOS II

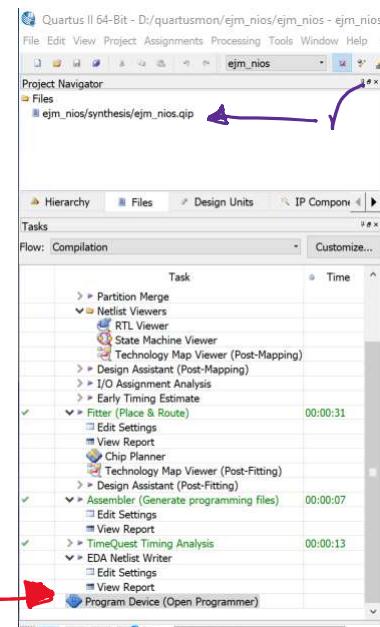
- Asignar los pines con el PinPlanner
- Volver a ejecutar “Compile Design”



21

## Procedimiento para implementar el NIOS II

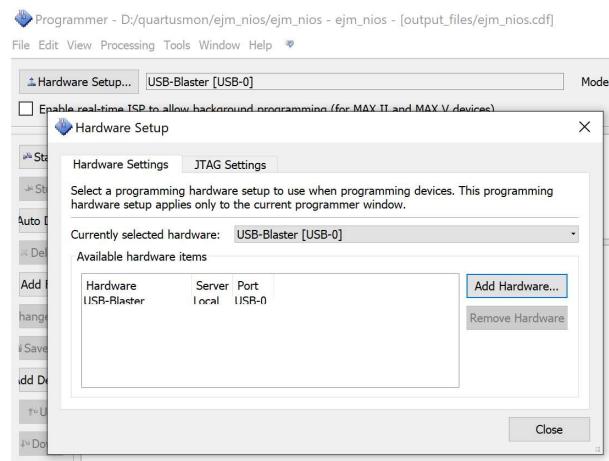
- Conectar el USB-Blaster al computador
- Conectar el USB-Blaster a la tarjeta de FPGA
- Conectar la fuente de energía a la tarjeta FPGA y encender la tarjeta
- Programar el dispositivo
  - Program Device (Open Programmer)



22

## Procedimiento para implementar el NIOS II

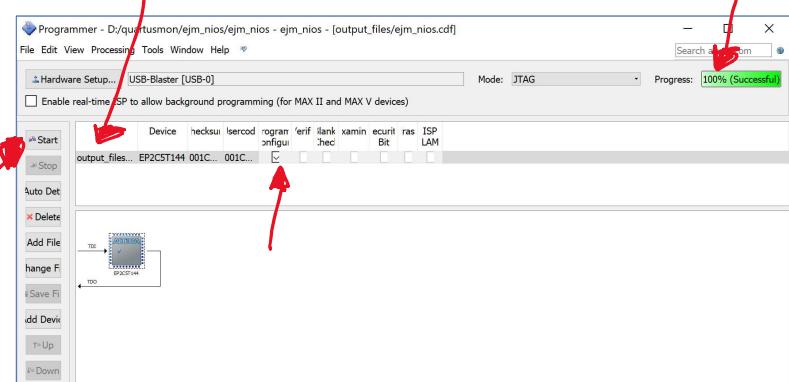
- En la ventana de Programmer, si no aparece el USB Blaster: entrar a Hardware Setup y seleccionar USB-Blaster (previamente se debió de instalar manualmente el driver y el S.O. Windows debe de haber reconocido correctamente el USB-Blaster)



23

## Procedimiento para implementar el NIOS II

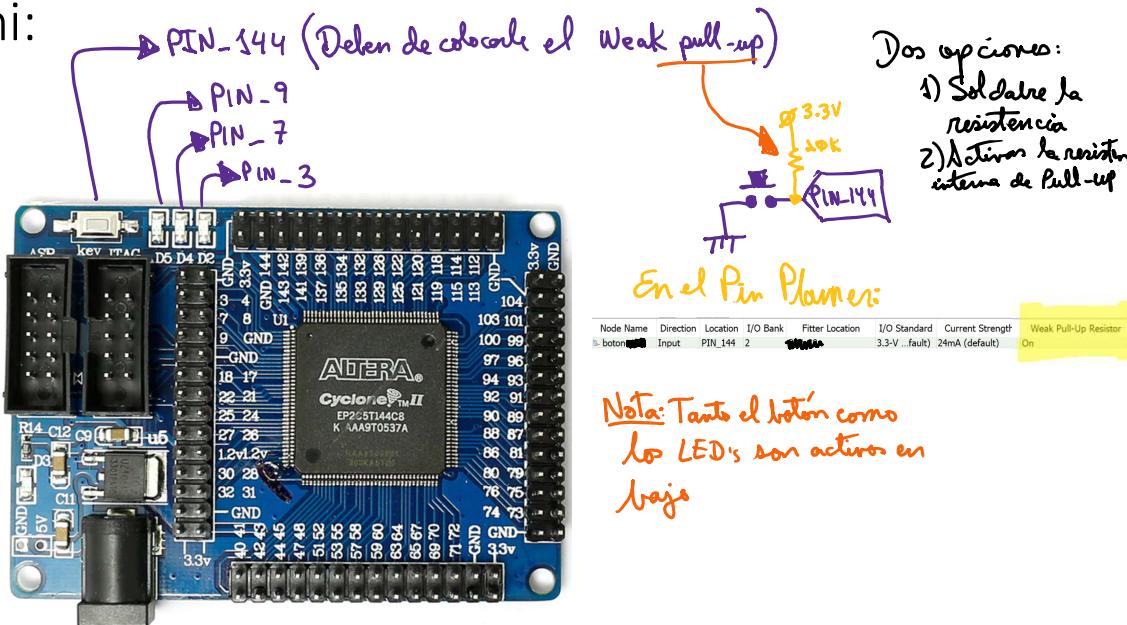
- Verificar que el archivo .sof sea el correcto
- Clic en Start y se iniciará la grabación del FPGA, esperar al 100%



24

## Usando los periféricos integrados del EP2C5T

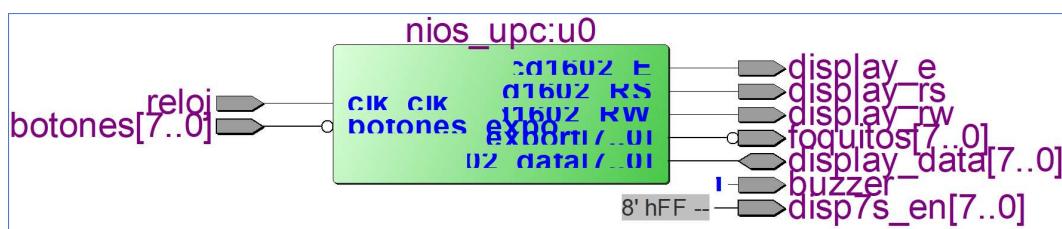
Mini:



25

## Vista RTL del NIOS II implementado:

- Nota: Orientado a la tarjeta A-C2FB:

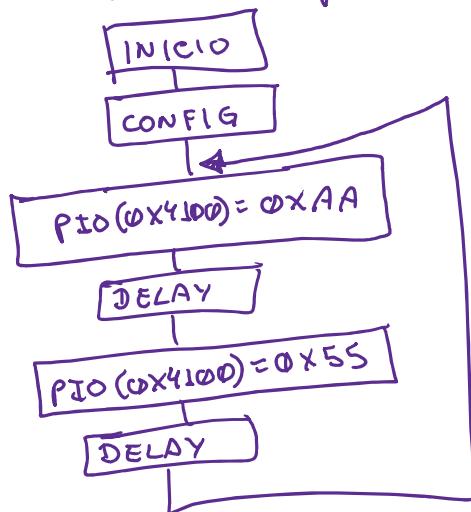


26

## Procedimiento para implementar el NIOS II

- Aplicación ejemplo a desarrollar en C para el procesador NIOS II:
  - Efecto de iluminación con los LEDs de la tarjeta que están conectados al PIO con dirección 0x4100

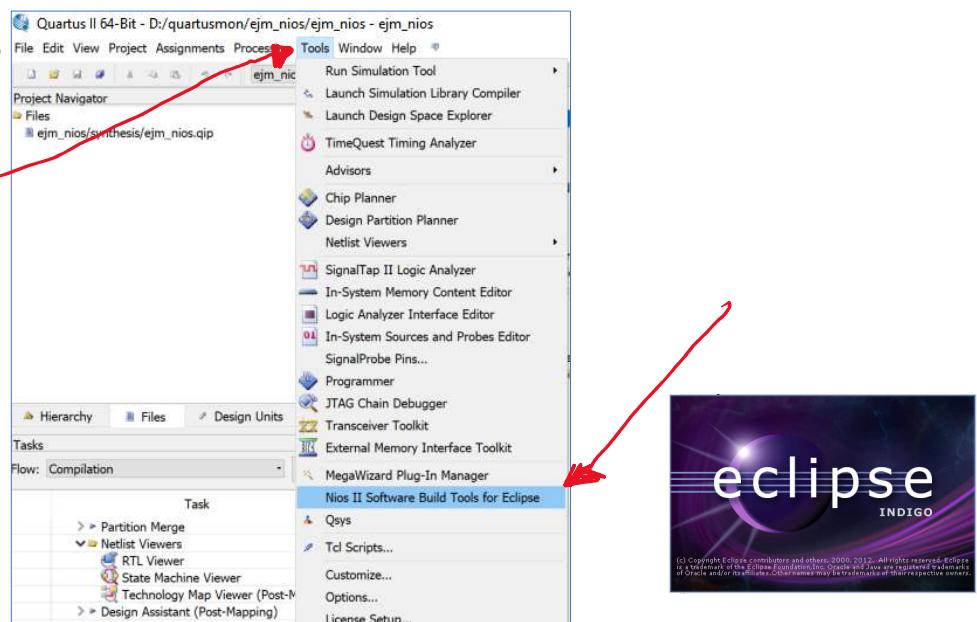
Diagrama de flujo:



27

## Procedimiento para implementar el NIOS II

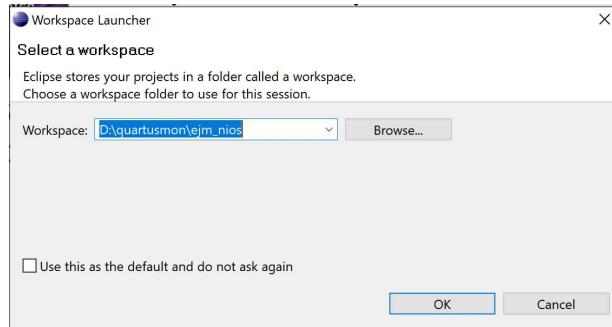
- Abrir el Eclipse



28

## Procedimiento para implementar el NIOS II

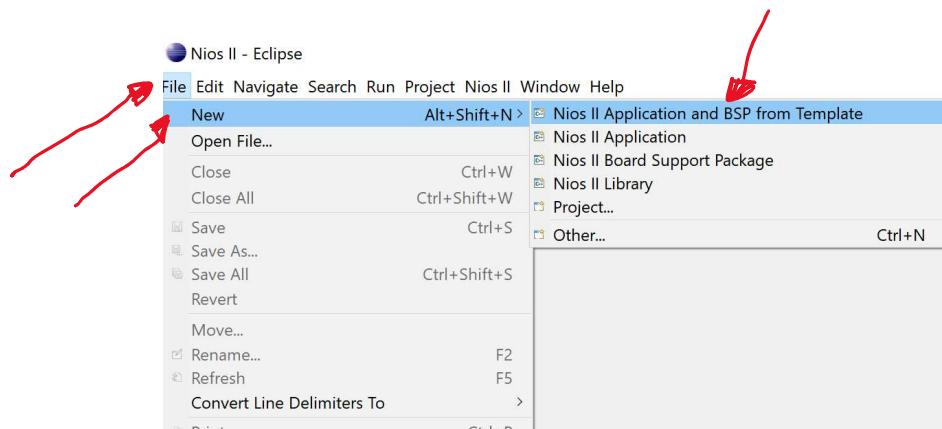
- Usar la misma carpeta donde esta creado el proyecto ejemplo



29

## Procedimiento para implementar el NIOS II

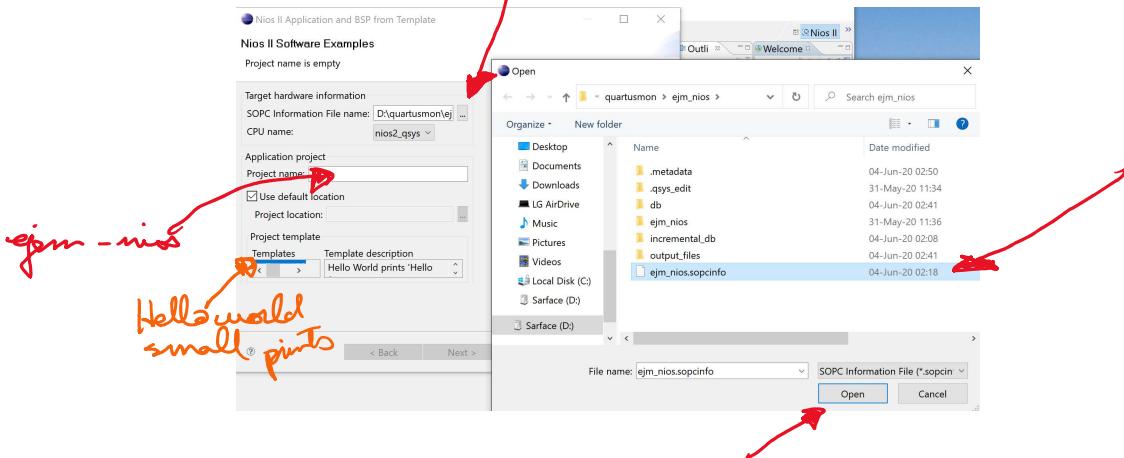
- Para crear el archivo fuente, escoger Nios II Application and BSP from Template



30

## Procedimiento para implementar el NIOS II

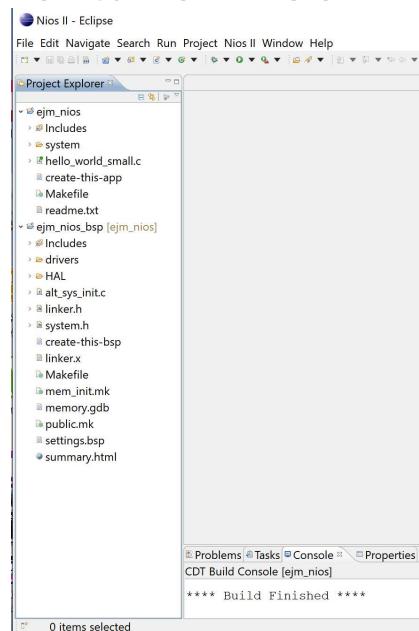
- Seleccionar el sopinfo creado en Qsys:
- El nombre del proyecto en Eclipse deberá ser el mismo
- Escoger el Template Hello World small prints



31

## Procedimiento para implementar el NIOS II

- Se habrá creado un árbol de archivos



32

## Procedimiento para implementar el NIOS II

- Abrir el archivo fuente  
hello\_world\_small.c

The screenshot shows the Eclipse IDE interface with the title "Nios II - parpadea/hello\_world\_small.c - Eclipse". The Project Explorer view on the left shows a project named "ejm\_nios" containing files like "Binaries", "Includes", "obj", and "system", along with the source file "hello\_world\_small.c". The right-hand editor pane displays the code for "hello\_world\_small.c". The code includes function prototypes for "alt\_printf", "alt\_putstr", "alt\_putchar", and "alt\_getchar", followed by the inclusion of the standard input-output header "stdio.h". A purple arrow points from the text "Abrir el archivo fuente" to the "hello\_world\_small.c" file in the Project Explorer.

```

Function
=====
alt_printf
alt_putstr
alt_putchar
alt_getchar
*/
#include <stdio.h>

```

33

## Procedimiento para implementar el NIOS II

- Escribir el código de programa siguiente:

The screenshot shows the code for "hello\_world\_small.c" with several handwritten annotations in red:

- A brace on the first two lines of the code is labeled "función para generar un retraso".
- A brace around the entire "main" function is labeled "función main".
- A brace under the "delay" loop is labeled "repetición infinita".
- An annotation points to the line "#include "altera\_avalon\_pio\_regs.h"" with the text "librería para usar los PIO".
- An annotation points to the line "IOWR\_ALTERA\_AVALON\_PIO\_DATA(0x4100, 0xAA);" with the binary value "10101010" above it and the text "escribe 0XAA en el puerto con dirección 0x4100".
- An annotation points to the line "IOWR\_ALTERA\_AVALON\_PIO\_DATA(0x4100, 0x55);" with the binary value "01010101" above it and the text "escribe 0X55 en el puerto con dirección 0x4100".

```

#include <stdio.h>
#include "system.h"
#include "altera_avalon_pio_regs.h"

void delay(int a){
    volatile int i = 0;
    while (i<a*10000){
        i++;
    }
}

int main(){
    while(1){
        IOWR_ALTERA_AVALON_PIO_DATA(0x4100, 0xAA);
        delay(30);
        IOWR_ALTERA_AVALON_PIO_DATA(0x4100, 0x55);
        delay(30);
    }
    return 0;
}

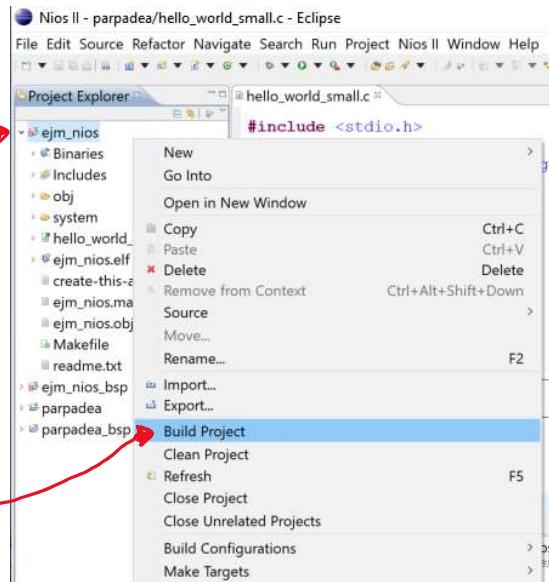
```

34

## Procedimiento para implementar el NIOS II

- Armar el proyecto (Build Project)

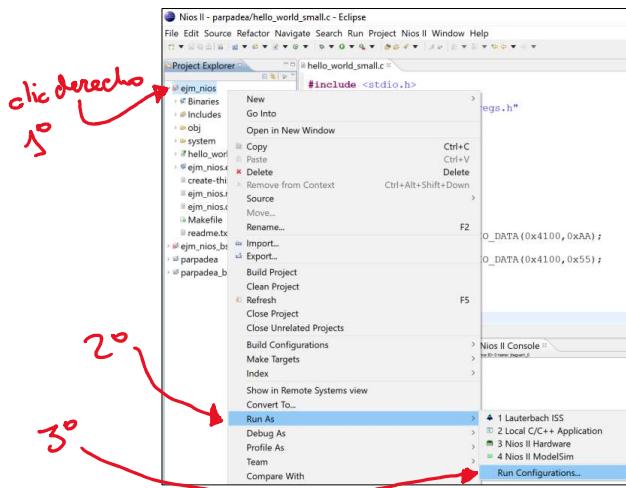
clic derecho



35

## Procedimiento para implementar el NIOS II

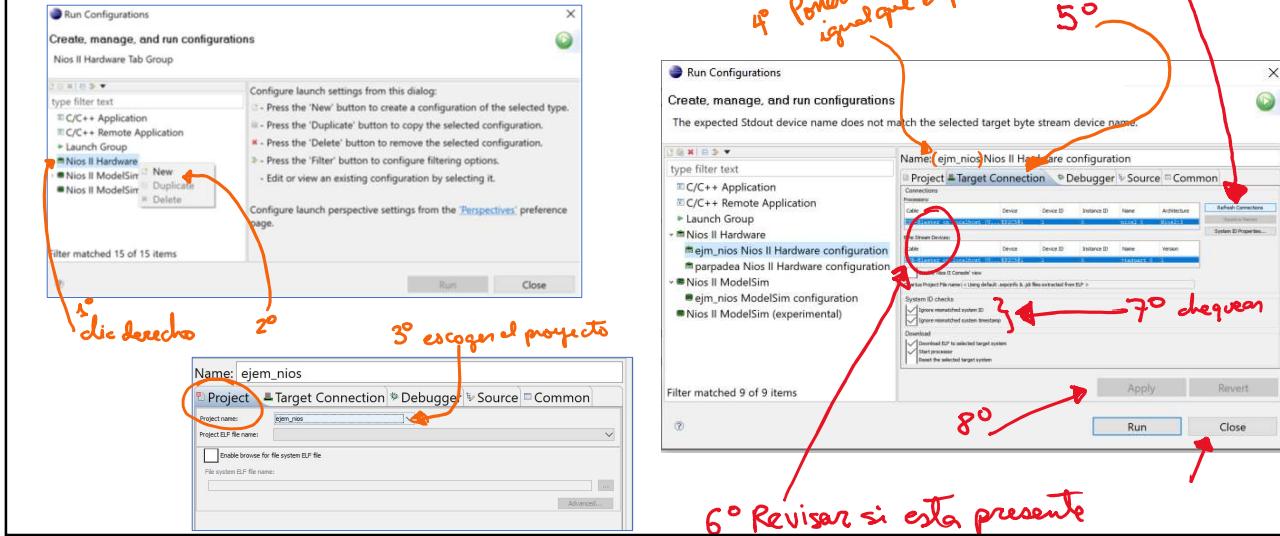
- Configurar conexión del NIOS II previamente grabado con el Eclipse



36

## Procedimiento para implementar el NIOS II

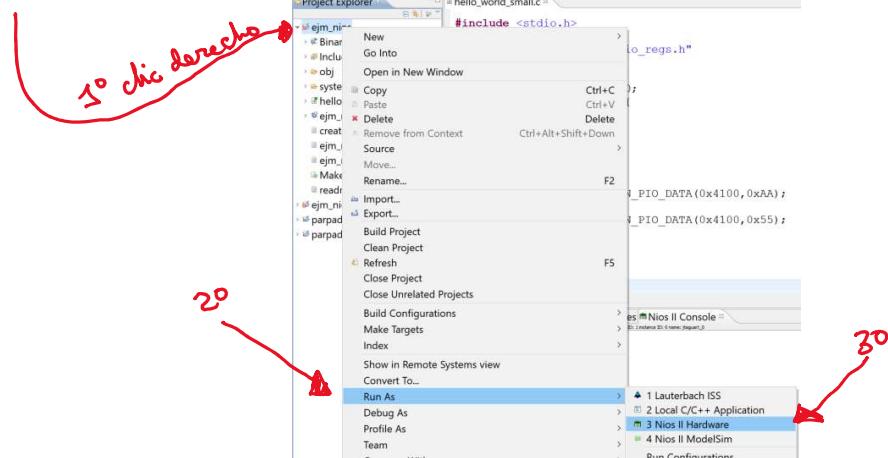
- Configurar conexión del NIOS II previamente grabado con el Eclipse



37

## Procedimiento para implementar el NIOS II

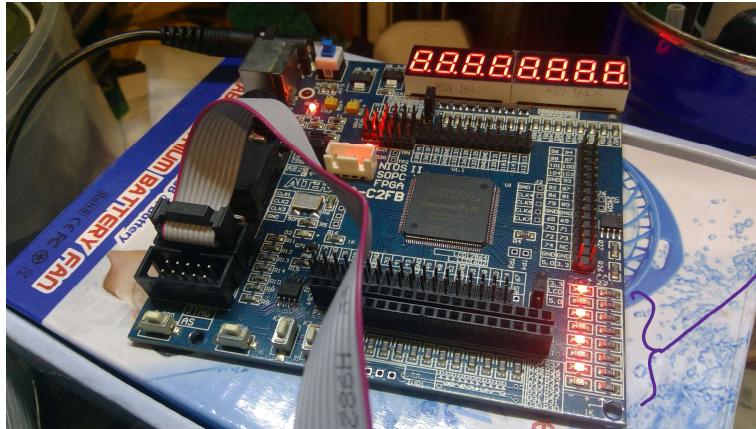
- Correr el código en el procesador NIOS II implementado en el FPGA



38

## Procedimiento para implementar el NIOS II

- Verificar el funcionamiento en la tarjeta A-C2FB:

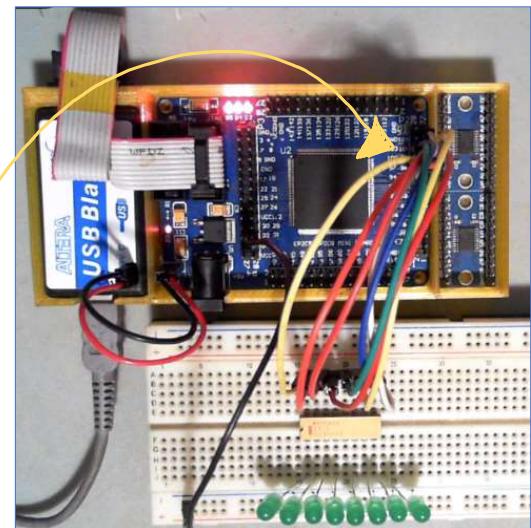
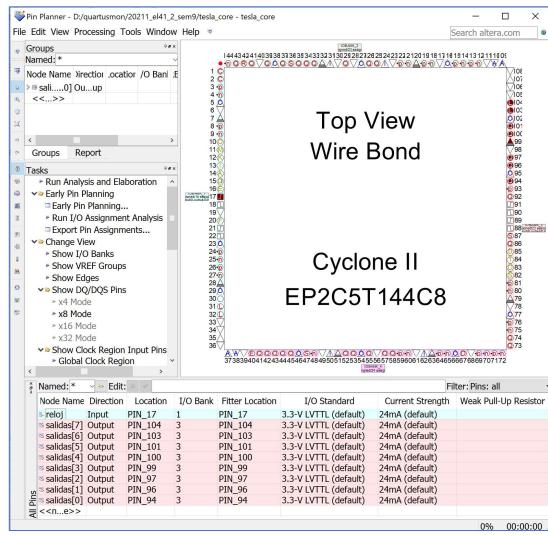


Leds conectados al  
PIO con dirección  
 $0x4100$  del NIOSII

39

## Procedimiento para implementar el NIOS II

- Verificar el funcionamiento en la tarjeta EP2C5T Miniboard:



40

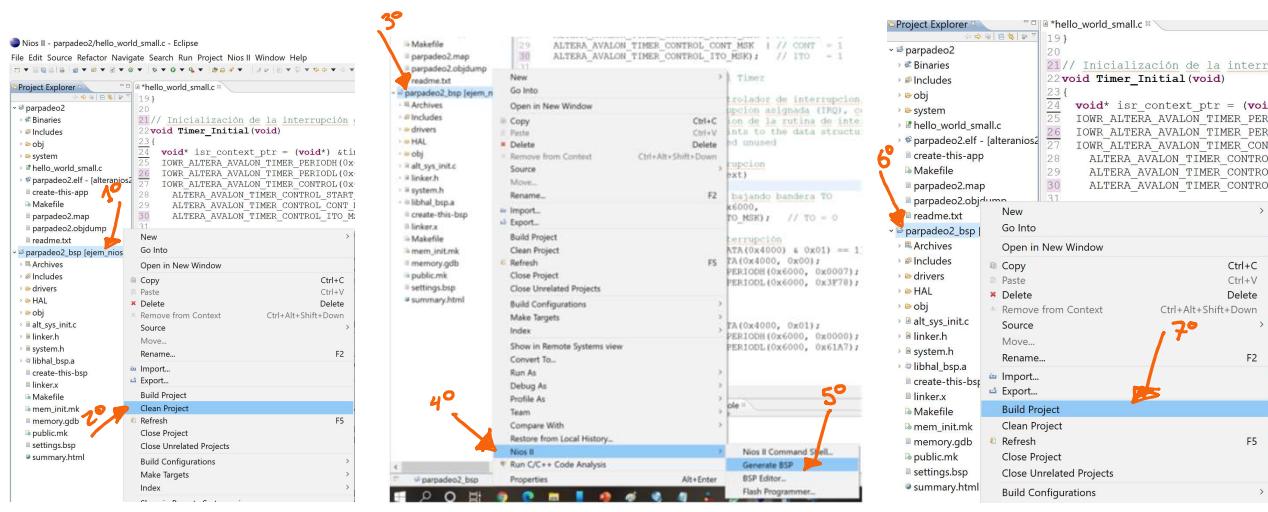
# Consideraciones al modificar la estructura en QSys o en el proyecto en Quartus II

- Estando en la etapa de desarrollo del programa de usuario del NIOS II en Eclipse requieres modificar la estructura del hardware o parámetros tanto en el QSys como en el Quartus II, será necesario hacer lo siguiente:
  - Si se modificó algo en QSys (estructura o parámetros), se tendrá que volver a generar el sopcinfo, luego compilar el proyecto en el Quartus y grabar el FPGA.
  - Si se modificó algo en el Quartus (tanto en estructura como en asignación de pines) se compilará el proyecto para luego grabarlo en el FPGA.

41

# Consideraciones al modificar la estructura en QSys o en el proyecto en Quartus II

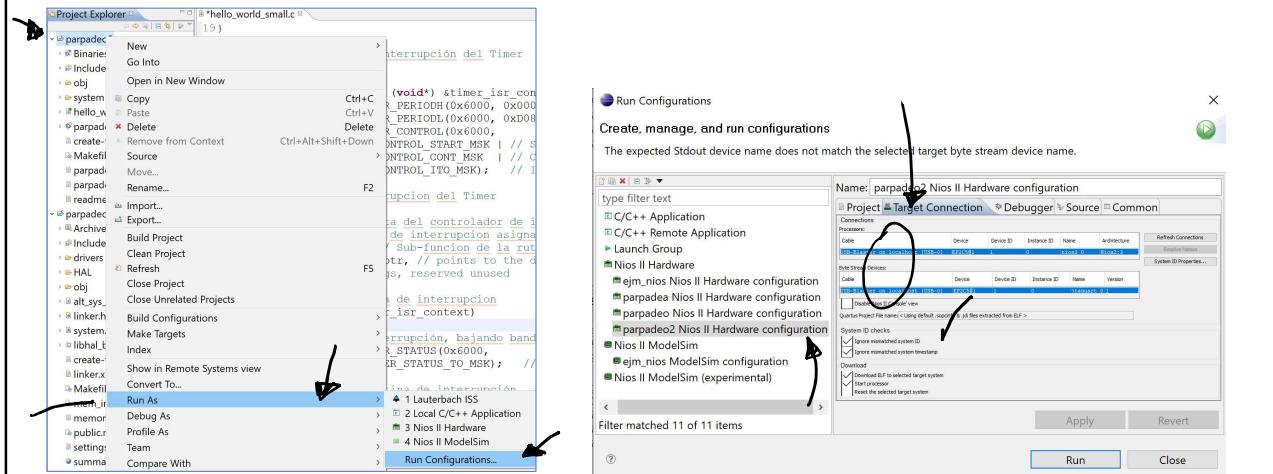
- En Eclipse hacer lo siguiente:



42

# Consideraciones al modificar la estructura en QSys o en el proyecto en Quartus II

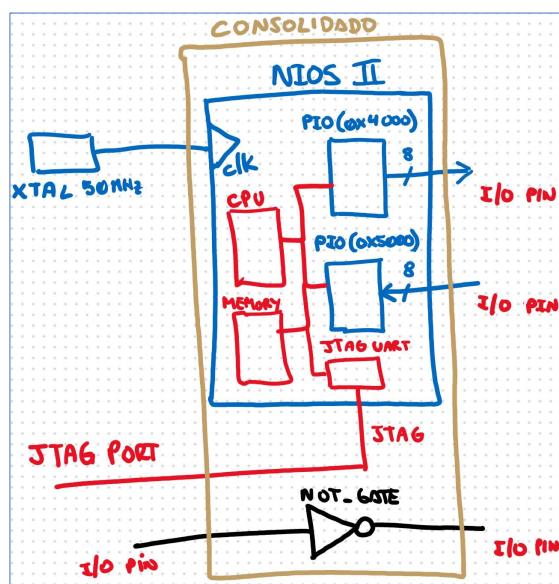
4. Revisar en Run As / Run Configurations si aparece el NIOS II implementado ingresando al menú Target Connection



43

## NIOS II llamado como componente que forma parte de otro sistema

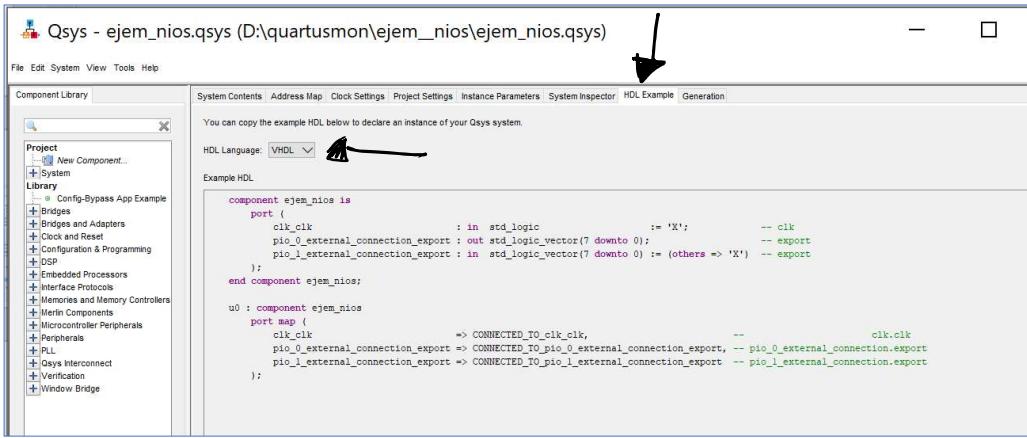
- El procesador NIOS II generado en el Qsys se puede instanciar como componente en VHDL.
- Se sigue el procedimiento de descripción estructural de VHDL para el llamado de componentes y su posterior descripción de las conexiones.
- En este ejemplo construiremos el top-level llamado consolidado el cual albergará dos componentes:
  - El procesador NIOS II
  - Una compuerta NOT\_GATE



44

## NIOS II llamado como componente que forma parte de otro sistema

- Para obtener el detalle del componente NIOS II en VHDL:
  - Entrar al QSys y seleccionar “HDL Example” en el menú horizontal

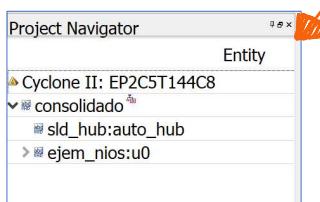


45

## NIOS II llamado como componente que forma parte de otro sistema

- Código estructural en VHDL

- Al compilar el proyecto aparecerá una nueva jerarquía



```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity consolidado is
7 port( clk : in std_logic;
8       in_not: in std_logic;
9       out_not: out std_logic;
10      in_nios: in std_logic_vector(7 downto 0);
11      out_nios:out std_logic_vector(7 downto 0));
12 end consolidado;
13
14 architecture estructural of consolidado is
15
16   component ejem_nios is
17     port(
18       clk_clk : in std_logic := 'X';
19       pio_0_external_connection_export : out std_logic_vector(7 downto 0);
20       pio_1_external_connection_export : in std_logic_vector(7 downto 0) := (others => 'X');
21     );
22   end component ejem_nios;
23
24 begin
25   u0 : component ejem_nios
26     port map (
27       clk_clk => clk,
28       pio_0_external_connection_export => out_nios, -- pio_0_external_connection.export
29       pio_1_external_connection_export => in_nios -- pio_1_external_connection.export
30     );
31   out_not <= not in_not;
32 end estructural;

```

46

## Consideraciones finales

- Será necesario revisar la hoja técnica de la tarjeta de FPGA Cyclone II que se está utilizando para ver que dispositivos externos posee.
- Para el caso de la tarjeta A-C2FB se tendrá que instanciar el NIOS II en un código VHDL para poder declarar señales adicionales y desactivar algunos de esos dispositivos externos conectados al FPGA.
  - El puerto del FPGA conectado al buzzer deberá de declararse como salida y activado en alto para que deje de sonar.
  - Tanto los LEDs, displays como los pulsadores que están en la tarjeta son activos en bajo.
- Si se graba el NIOS II en la memoria de configuración externa EPICS4, solo se grabará la implementación del procesador mas no el código fuente. Para ello se tiene la opción de usar esa misma EPICS4 para almacenar el programa de usuario luego de la configuración del FPGA, pero hasta el momento (22/05/2021) no se ha podido verificar.