

Aplicaciones con el NIOS II

Por Kalun José Lau Gan

Changelog

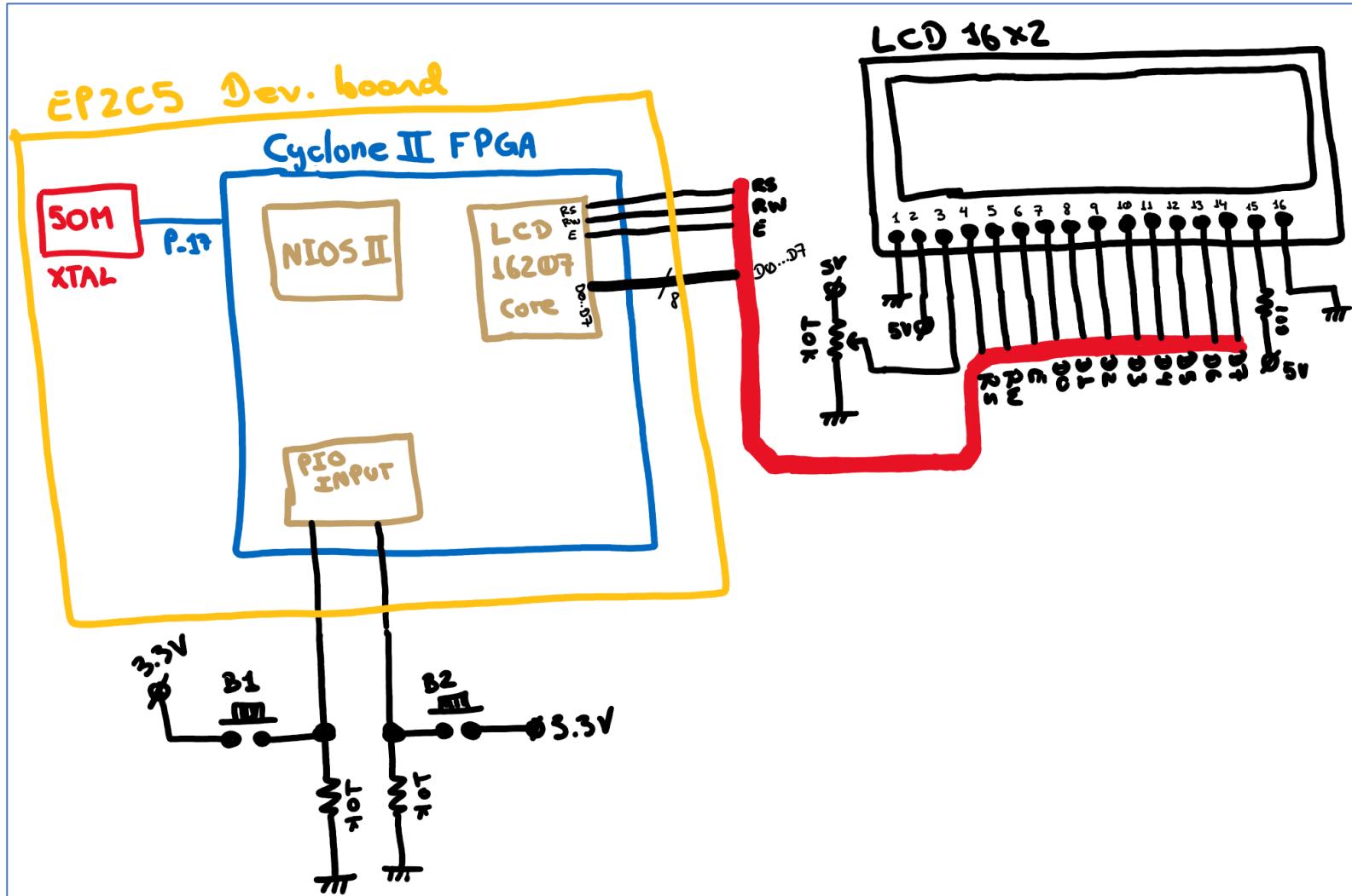
- 17-06-2021 23:36 – Creación del documento
- 19-06-2021 17:42 – Ejercicio 3 completado
- 19-06-2021 22:21 – Ejercicio 1 y 2 completados

Índice de ejemplos

- Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar del estado de dos pulsadores
- Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental
- Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2

Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar el estado de dos pulsadores

- Diagrama de bloques

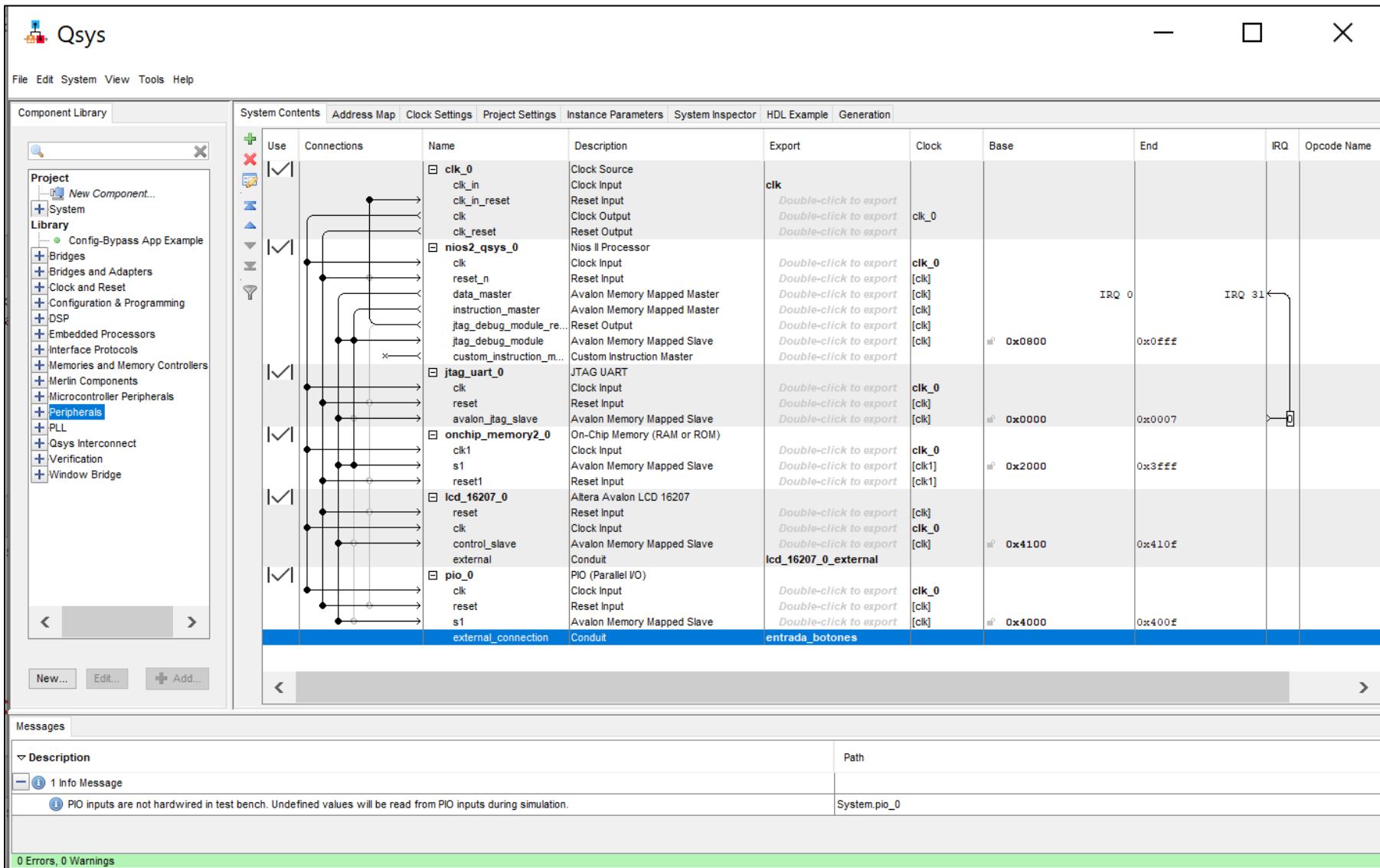


Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar del estado de dos pulsadores

- Aspectos iniciales:
 - Se empleará un PIO entrada de 8 bits para la conexión de los dos pulsadores, dichos pulsadores ocuparán los dos bits menos significativos.
 - En el LCD se visualizará el estado de ambos botones con mensajes “OFF” cuando no estén presionados y “ON” cuando estén presionados.
 - Tener en cuenta que la lógica del programa principal será la de preguntar constantemente (polling) el estado del PIO entrada.

Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar el estado de dos pulsadores

- Esquema en Qsys:
 - El LCD Core se encuentra en la dirección 0x4100
 - El PIO de entrada se encuentra en la dirección 0x4000



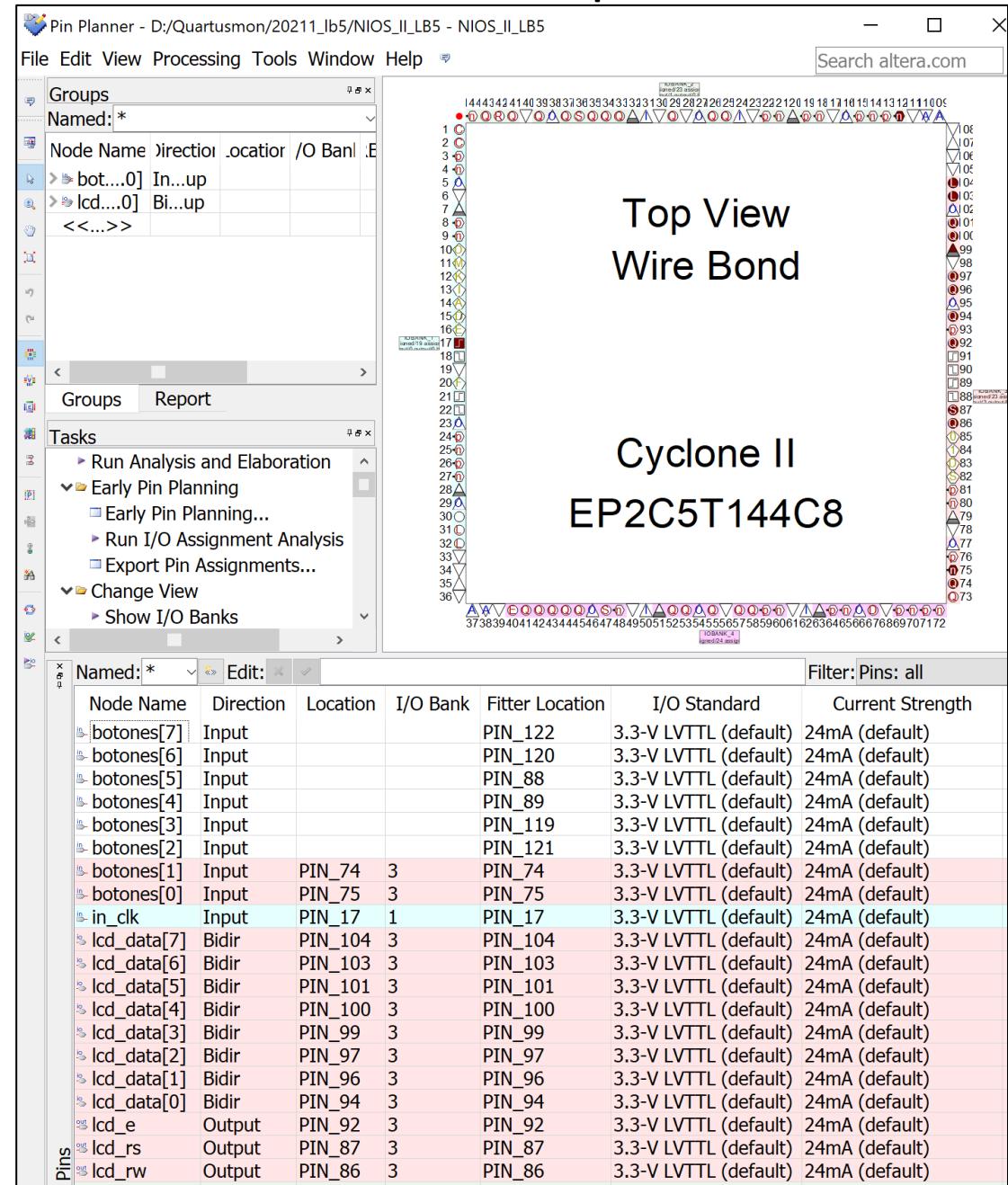
Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar el estado de dos pulsadores

- Instanciación del NIOS II (VHDL estructural):

```
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 NIOS_II_LB5 is
7   port( in_clk: in std_logic;
8         botones: in std_logic_vector(7 downto 0);
9         lcd_rs, lcd_rw, lcd_e: out std_logic;
10        lcd_data: inout std_logic_vector(7 downto 0));
11 end NIOS_II_LB5;
12
13 architecture estructura of NIOS_II_LB5 is
14
15   component nios_lcd is
16     port (
17       clk_clk           : in  std_logic          := 'X';      -- clk
18       entrada_botones_export : in  std_logic_vector(7 downto 0) := (others => 'X'); -- export
19       lcd_16207_0_external_RS : out std_logic;           -- RS
20       lcd_16207_0_external_RW : out std_logic;           -- RW
21       lcd_16207_0_external_data : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
22       lcd_16207_0_external_E  : out std_logic;           -- E
23     );
24   end component nios_lcd;
25
26 begin
27
28   u0 : component nios_lcd
29   port map (
30     clk_clk           => in_clk,    -- clk.clk
31     entrada_botones_export => botones,   -- entrada_botones.export
32     lcd_16207_0_external_RS => lcd_rs,    -- lcd_16207_0_external.RS
33     lcd_16207_0_external_RW => lcd_rw,    -- .RW
34     lcd_16207_0_external_data => lcd_data,  -- .data
35     lcd_16207_0_external_E  => lcd_e      -- .E
36   );
37
38 end estructura;
```

Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar del estado de dos pulsadores

- Pin Planner:
 - Como se mencionó anteriormente, en el PIO de entradas solo se están empleando los dos menos significativos y que se usarán para conectar los dos pulsadores de la aplicación. El resto de señales las dejaremos en blanco y el Pin Planner les auto-asignará pines.
 - Se tendrá que contemplar un mecanismo de enmascaramiento para el PIO entrada para que solamente se lean los dos bits menos significativos.



Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar del estado de dos pulsadores

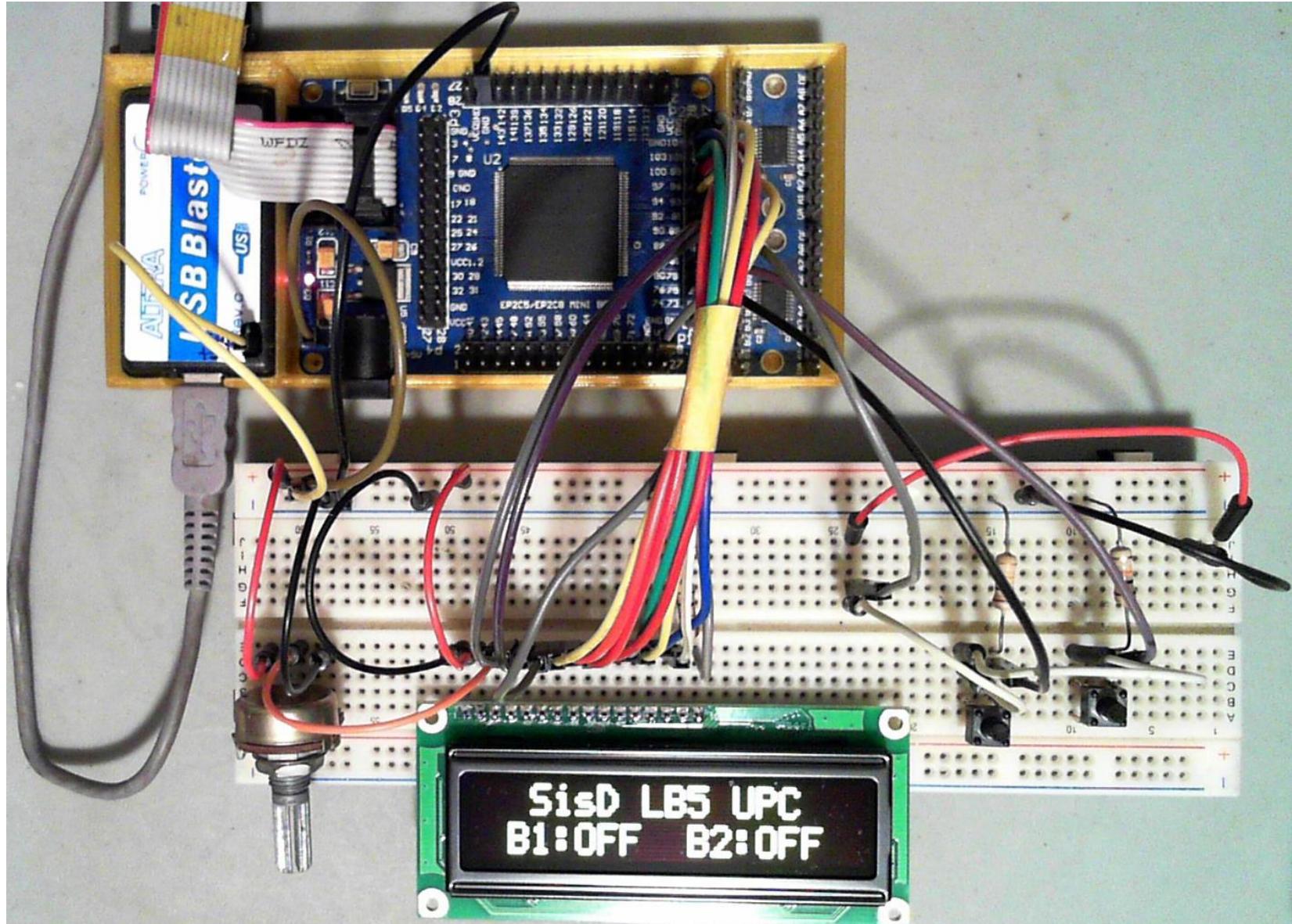
```
1#include "sys/alt_stdio.h"
2#include "system.h"
3#include "unistd.h"
4#include "altera_avalon_lcd_16207_regs.h"
5#include "altera_avalon_pio_regs.h"
6
7#define LCD_WR_COMMAND_REG 0
8#define LCD_RD_STATUS_REG 1
9#define LCD_WR_DATA_REG 2
10#define LCD_RD_DATA_REG 3
11#define LCD_0_BASE 0x4100
12
13unsigned char boton1=0;
14unsigned char boton2=0;
15
16void lcd_init(void){
17    usleep(15000);
18    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
19    usleep(4100);
20    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
21    usleep(100);
22    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
23    usleep(5000);
24    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
25    usleep(100);
26    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
27    usleep(100);
28    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
29    usleep(100);
30    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
31    usleep(100);
32    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
33    usleep(2000);
34    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
35    usleep(2000);
36}
37
38void ESCRIBE_MENSAJE(const char *cadena, unsigned char tam){
39    unsigned char i=0;
40    for(i=0;i<tam;i++){
41        IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
42        usleep(100);
43    }
44}
```

- Código en Eclipse:

```
46int main(void){
47    alt_putstr("SisD LB5 UPC\n");
48    alt_putstr("Update 11-06-2021 16:10\n");
49    lcd_init();
50    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
51    usleep(2000);
52    ESCRIBE_MENSAJE(" SisD LB5 UPC ", 16);
53    while(1{
54        boton1 = IORD_ALTERA_AVALON_PIO_DATA(0x4000) & 0x01;
55        boton2 = IORD_ALTERA_AVALON_PIO_DATA(0x4000) & 0x02;
56        boton2 = boton2 >> 1;
57        IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC1);
58        usleep(2000);
59        if(boton1 == 1{
60            ESCRIBE_MENSAJE("B1: ON ", 7);
61        }
62        else{
63            ESCRIBE_MENSAJE("B1:OFF ", 7);
64        }
65        IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC9);
66        usleep(2000);
67        if(boton2 == 1{
68            ESCRIBE_MENSAJE("B2: ON ", 7);
69        }
70        else{
71            ESCRIBE_MENSAJE("B2:OFF ", 7);
72        }
73    }
74    return 0;
75}
```

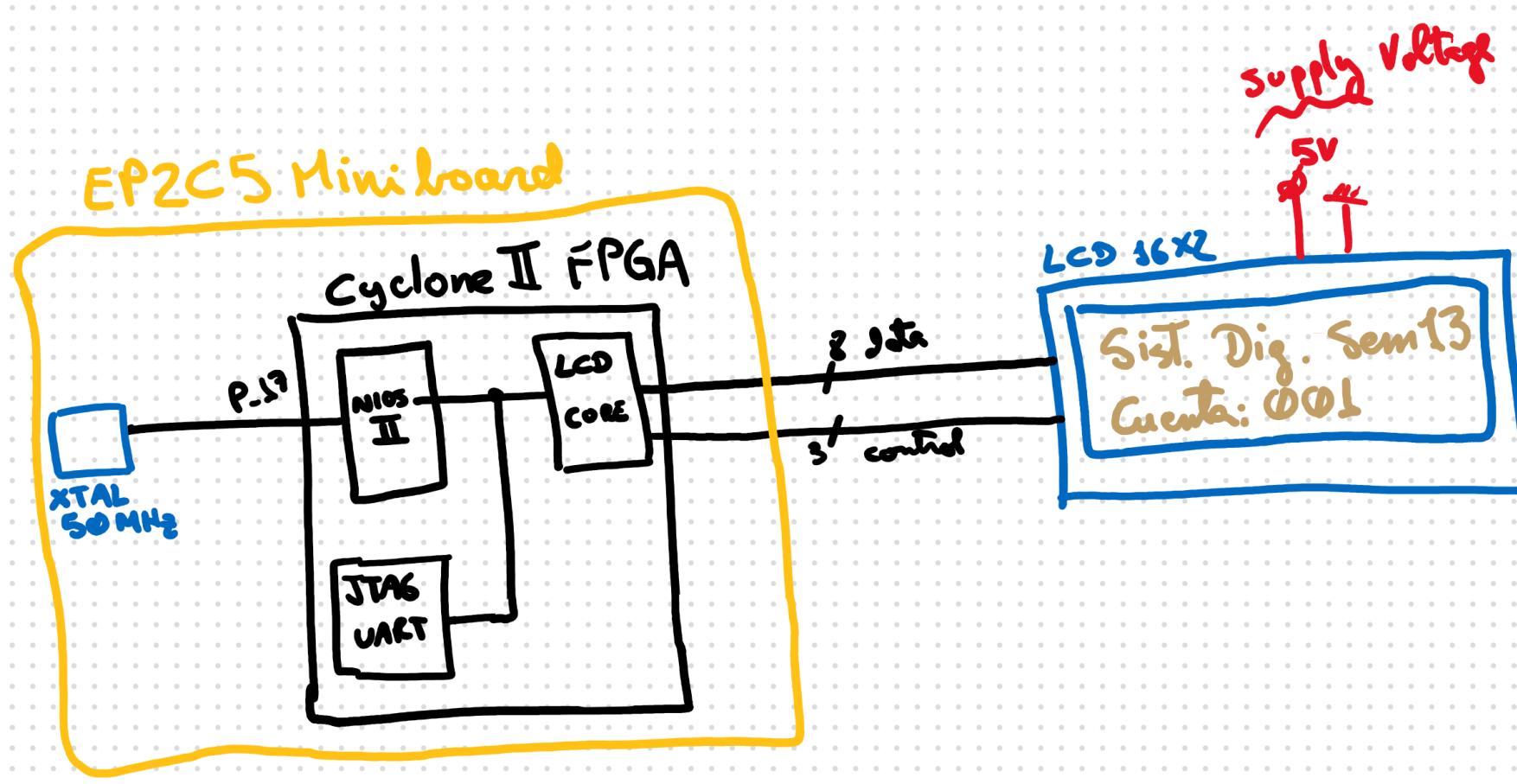
Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar el estado de dos pulsadores

- Circuito implementado:



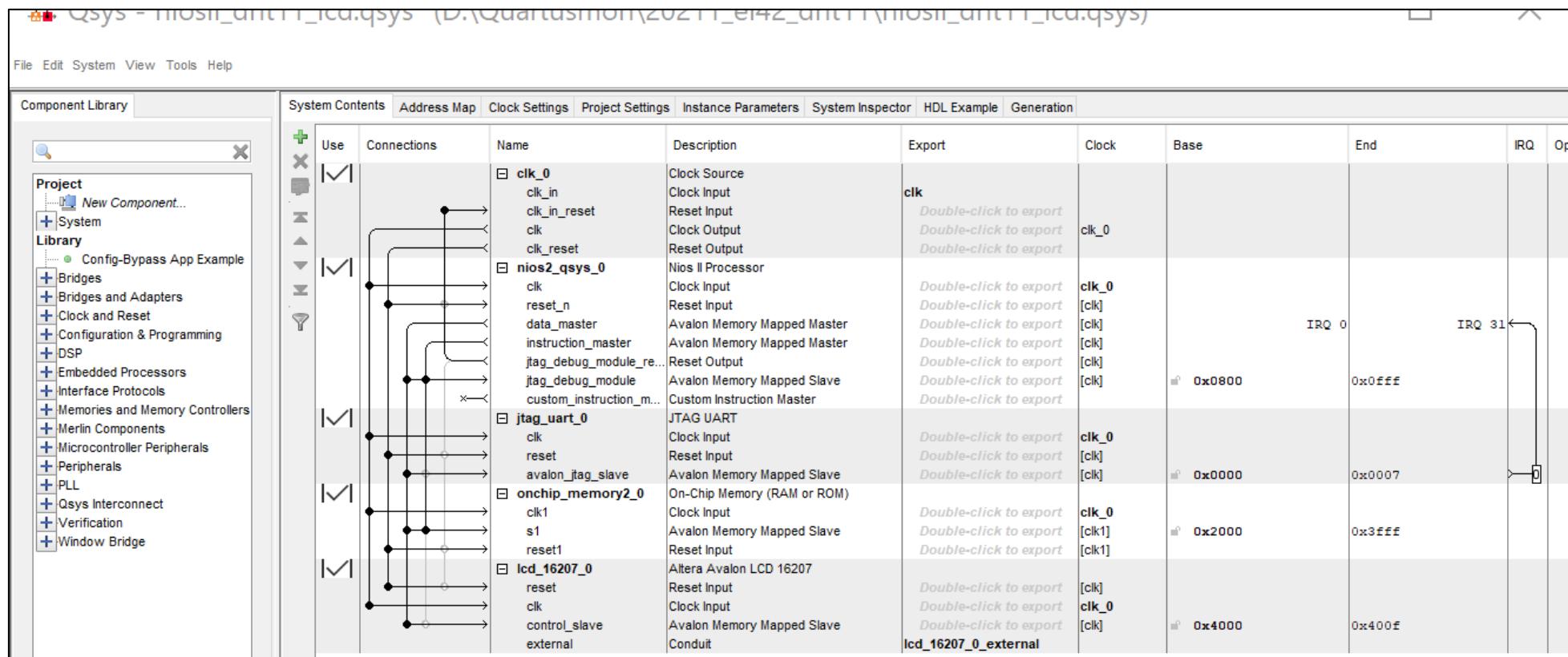
Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Diagrama de bloques



Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Esquema en Qsys:
 - El LCD Core esta mapeado en la dirección 0x4000



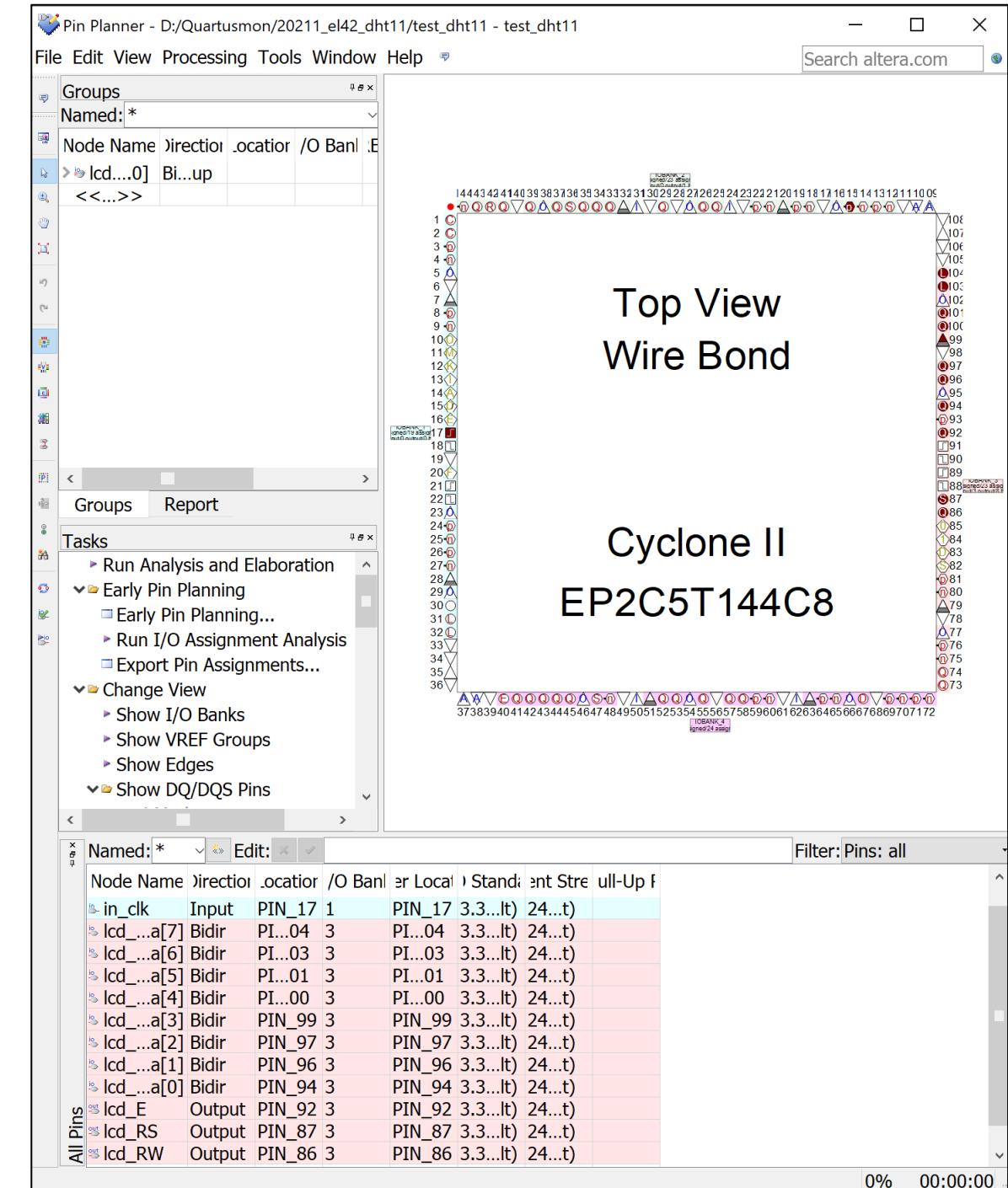
Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Instanciación en Qsys:

```
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 test_cuenta is
7 port(
8     in_clk      : in  std_logic;                                -- clk
9     lcd_RS     : out std_logic;                                -- RS
10    lcd_RW     : out std_logic;                                -- RW
11    lcd_data   : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
12    lcd_E      : out std_logic;                                -- E
13 );
14 end test_dht11;
15
16 architecture estructura of test_cuenta is
17 component niosii_cuenta_lcd is
18 port (
19     clk_clk          : in  std_logic := 'X';                -- clk
20     lcd_16207_0_external_RS : out std_logic;                -- RS
21     lcd_16207_0_external_RW : out std_logic;                -- RW
22     lcd_16207_0_external_data : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
23     lcd_16207_0_external_E  : out std_logic;                -- E
24 );
25 end component niosii_cuenta_lcd;
26
27 begin
28 u0 : component niosii_cuenta_lcd
29 port map (
30     clk_clk          => in_clk,                                -- clk.clk
31     lcd_16207_0_external_RS => lcd_RS,                      -- lcd_16207_0_external.RS
32     lcd_16207_0_external_RW => lcd_RW,                      -- .RW
33     lcd_16207_0_external_data => lcd_data,                  -- .data
34     lcd_16207_0_external_E  => lcd_E);                     -- .E
35 );
36 end estructura;
```

Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Asignación de pines en Pin Planner:



Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Software de la aplicación en Eclipse: Librería “lcd_lib” para el control del LCD 16x2

lcd_lib.h

```
8 #ifndef LCD_LIB_H_
9 #define LCD_LIB_H_
10
11 #include "sys/alt_stdio.h"
12 #include "system.h"
13 #include "unistd.h"
14 #include "string.h"
15 #include "altera_avalon_lcd_16207_regs.h"
16
17 #define LCD_WR_COMMAND_REG 0
18 #define LCD_RD_STATUS_REG 1
19 #define LCD_WR_DATA_REG 2
20 #define LCD_RD_DATA_REG 3
21 #define LCD_0_BASE 0x4000
22
23 void LCD_init(void);
24 void LCD_ESCRIBE_MENSAJE(const char *cadena);
25 void LCD_CHAR_SEND(unsigned char info);
26 void LCD_CLEAR(void);
27 void LCD_LINE1(void);
28 void LCD_LINE2(void);
29
30#endif /* LCD_LIB_H_ */
```

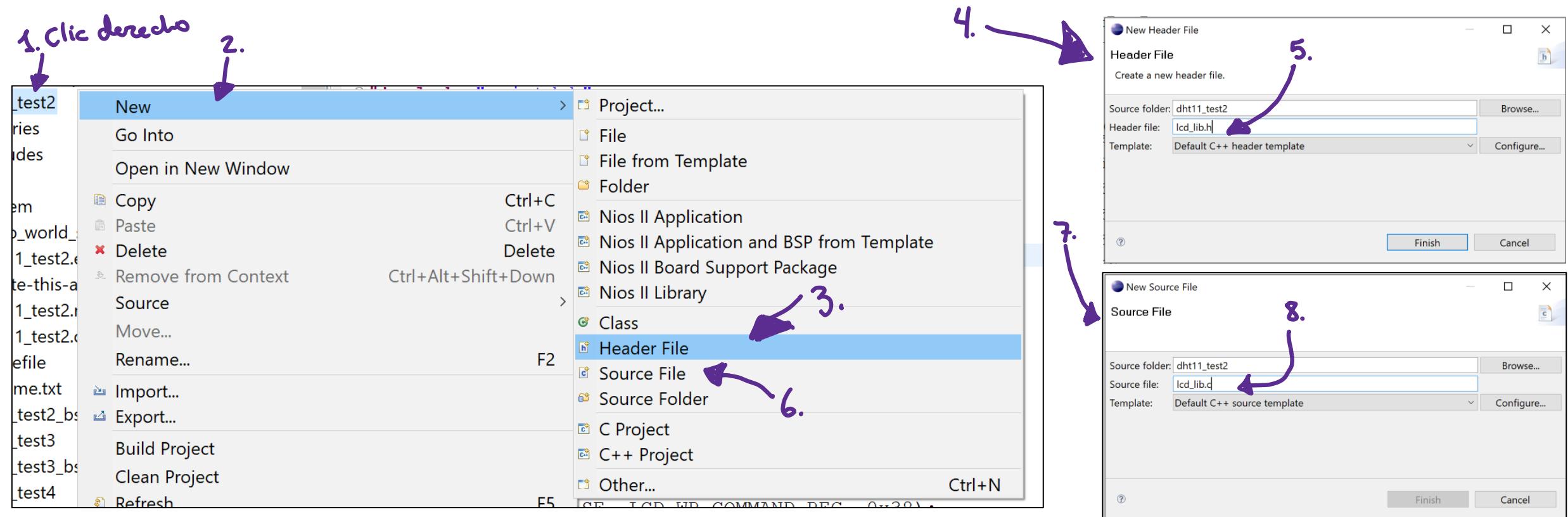
lcd_lib.c

```
8 #include "lcd_lib.h"
9
10 void LCD_init(void){
11     usleep(15000);
12     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
13     usleep(4100);
14     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
15     usleep(100);
16     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
17     usleep(5000);
18     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
19     usleep(100);
20     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
21     usleep(100);
22     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
23     usleep(100);
24     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
25     usleep(100);
26     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
27     usleep(2000);
28     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
29     usleep(2000);
30 }
31
32 void LCD_ESCRIBE_MENSAJE(const char *cadena) {
33     unsigned char tam;
34     tam = strlen(cadena);
35     unsigned char i=0;
36     for(i=0;i<tam;i++) {
37         IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
38         usleep(100);
39     }
40 }
```

```
42 void LCD_CHAR_SEND(unsigned char info){
43     IOWR(LCD_0_BASE, LCD_WR_DATA_REG, info);
44     usleep(100);
45 }
46
47 void LCD_CLEAR(void){
48     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
49     usleep(2000);
50 }
51
52 void LCD_LINE1(void){
53     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
54     usleep(2000);
55 }
56
57 void LCD_LINE2(void){
58     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC0);
59     usleep(2000);
60 }
```

Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Inclusión de librerías en Eclipse:
 - Clic derecho al nombre del proyecto, seleccionar “New” y luego “Header File” y crear el archivo header con nombre “lcd_lib.h”
 - De igual manera volver hacer lo anterior para crear el archivo “lcd_lib.c”



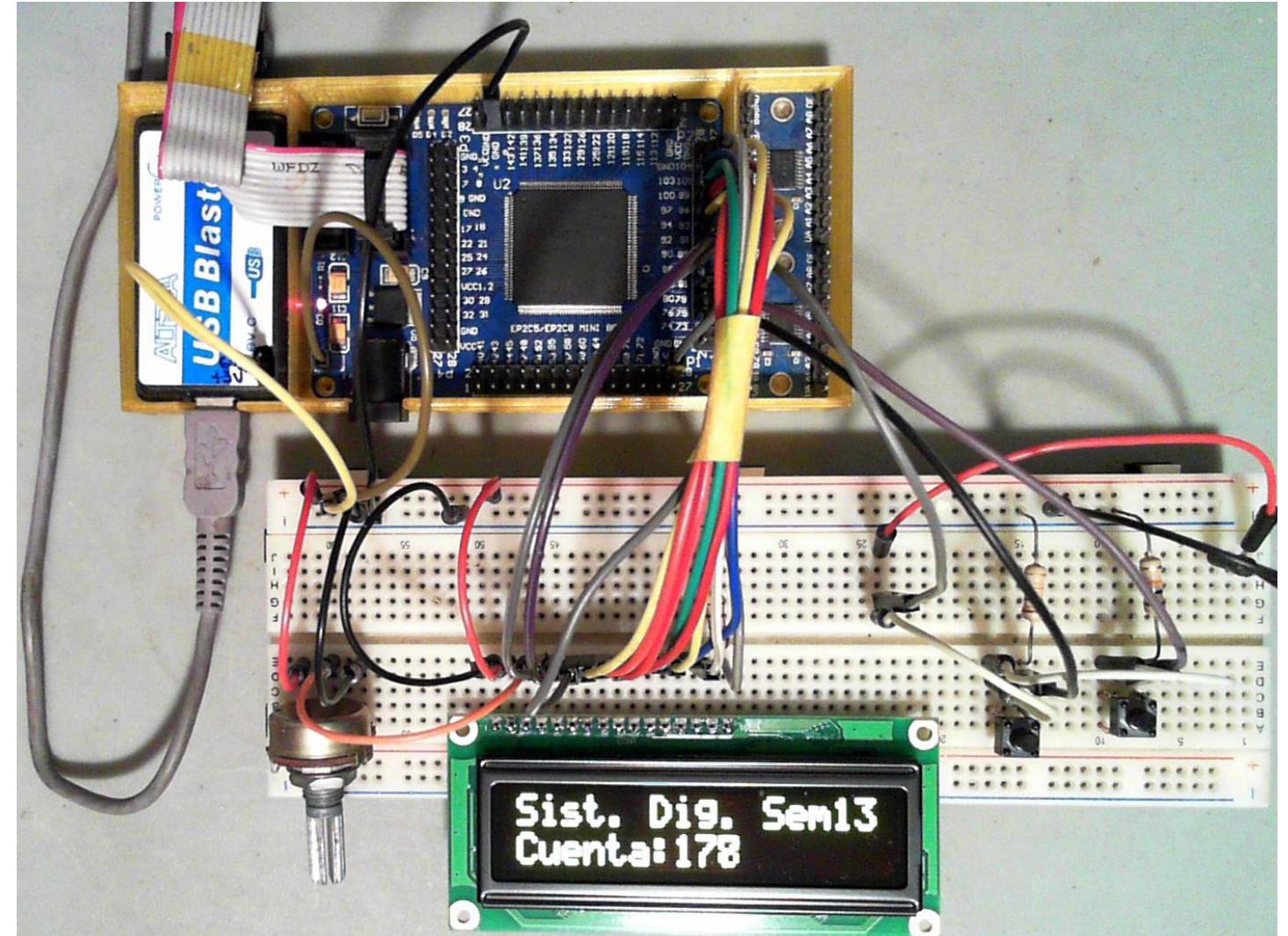
Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Software de la aplicación en Eclipse:
Código principal

```
1 #include "sys/alt_stdio.h"
2 #include "system.h"
3 #include "unistd.h"
4 #include "string.h"
5 #include "altera_avalon_lcd_16207_regs.h"
6 #include "altera_avalon_pio_regs.h"
7 #include "lcd_lib.h"
8
9 unsigned char cuenta=0;
10 unsigned char centena=0;
11 unsigned char decena=0;
12 unsigned char unidad=0;
13
14 void convierte(unsigned char numero){
15     centena = (numero % 1000) / 100;
16     decena = (numero % 100) / 10;
17     unidad = numero % 10;
18 }
19
20 int main(){
21     LCD_init();
22     LCD_LINE1();
23     LCD_ESCRIBE_MENSAJE("Sist. Dig. Sem13");
24     while(1){
25         LCD_LINE2();
26         LCD_ESCRIBE_MENSAJE("Cuenta:");
27         convierte(cuenta);
28         LCD_CHAR_SEND(centena+0x30);
29         LCD_CHAR_SEND(decena+0x30);
30         LCD_CHAR_SEND(unidad+0x30);
31         cuenta++;
32         usleep(100000);
33     }
34 }
```

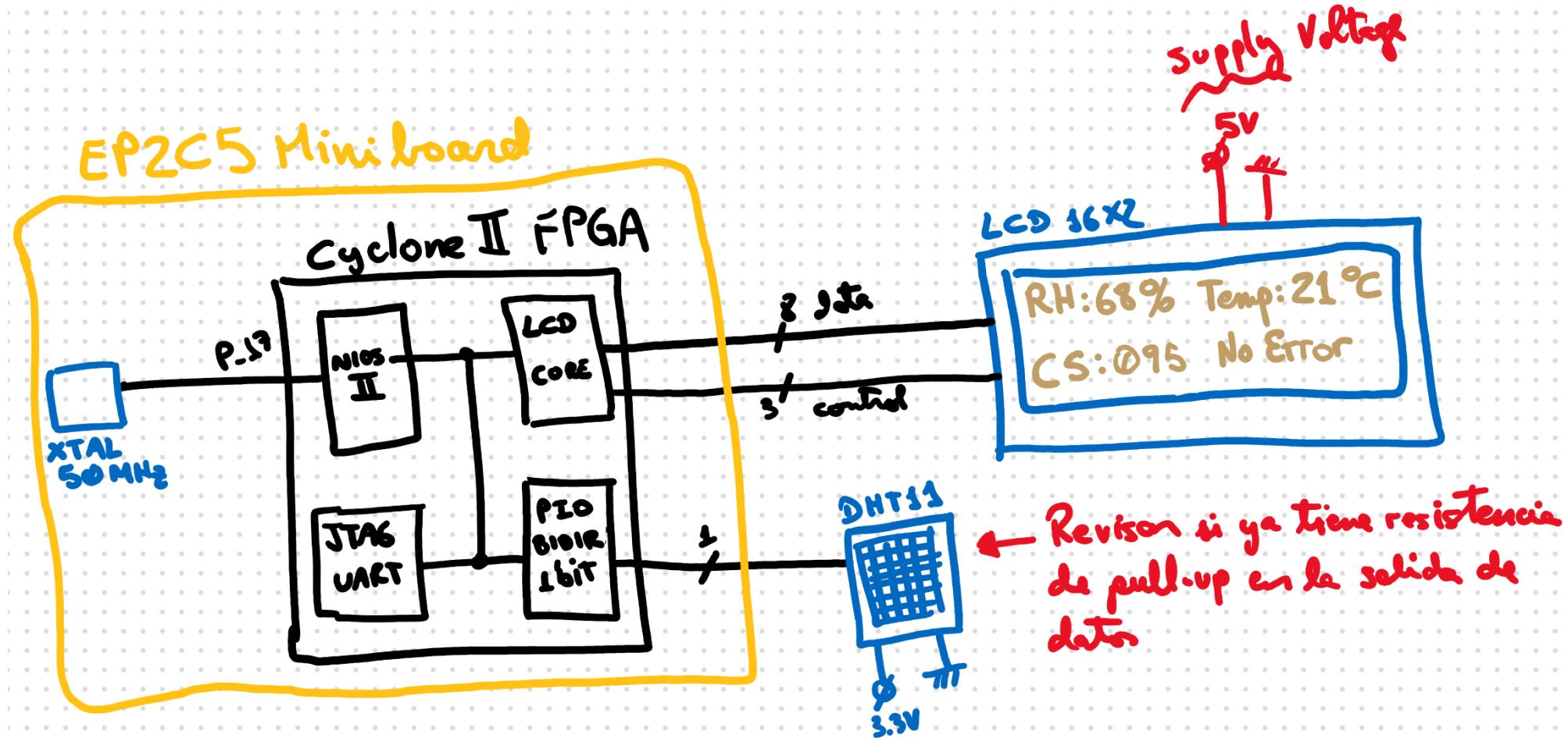
Ejemplo 2: Interface del NIOS II un LCD 16x2 para visualizar un contador autoincremental

- Circuito implementado:



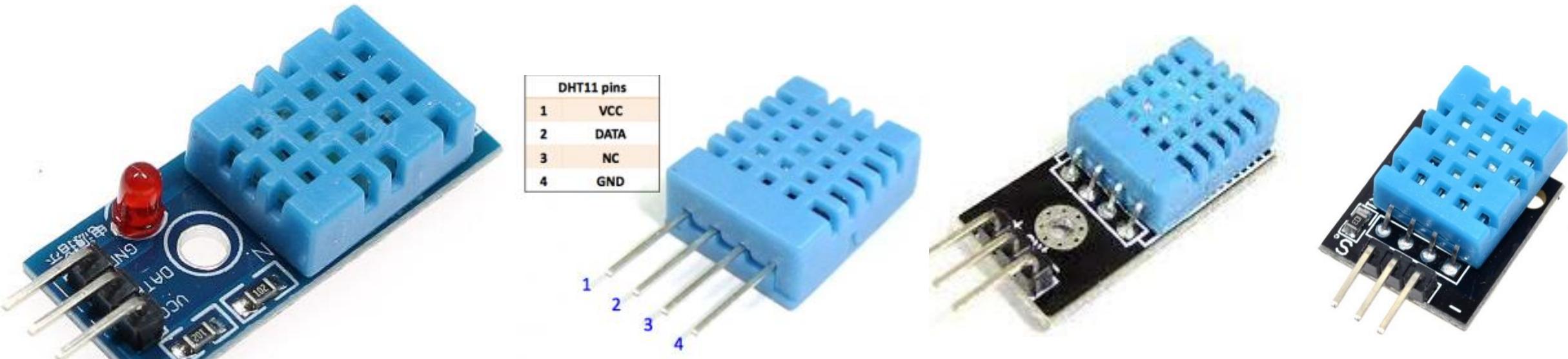
Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2

- Diagrama de bloques



Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2

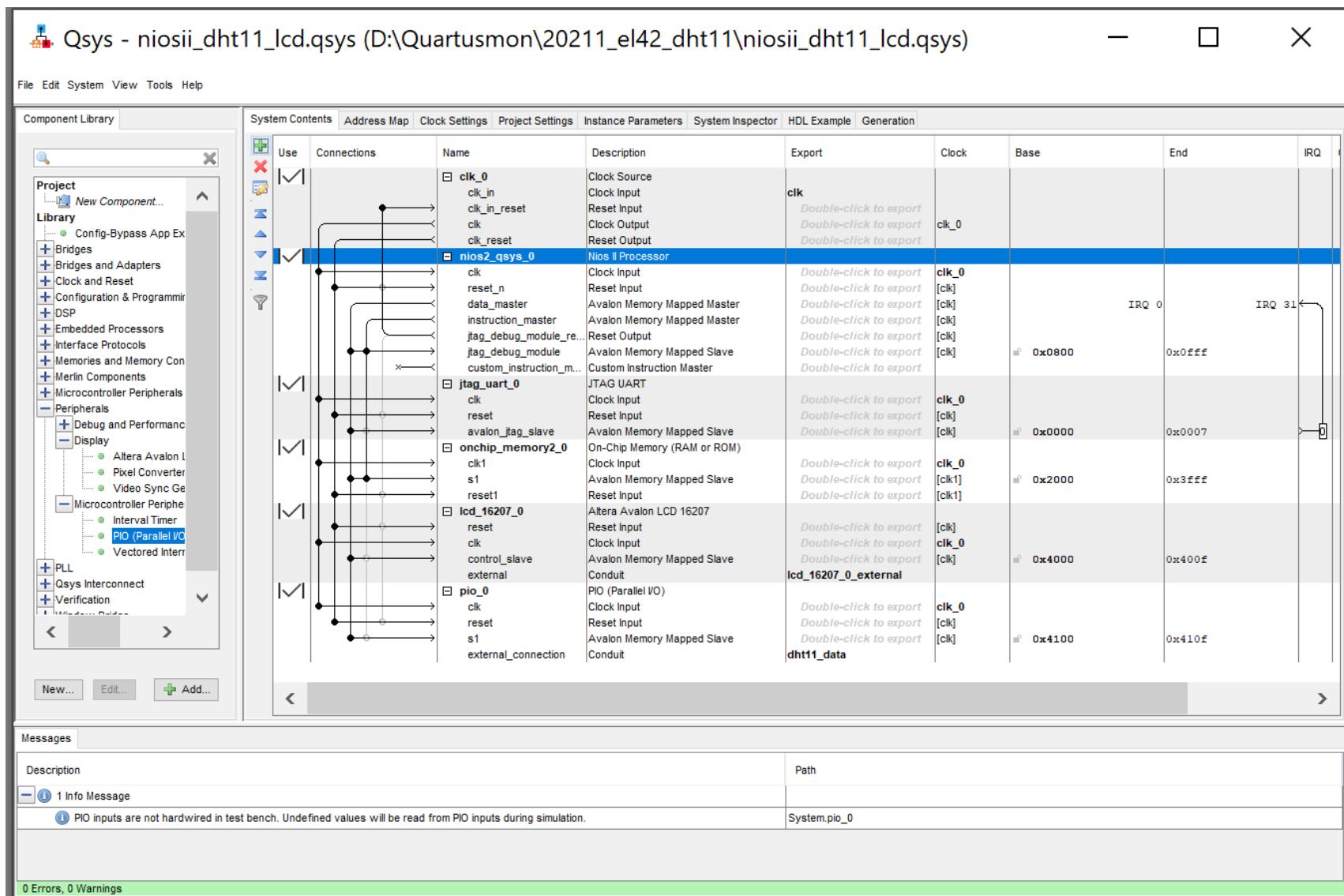
- Aspectos iniciales:
 - Al revisar la hoja técnica del DHT11 podemos ver que el DHT11 tiene un rango de voltaje de operación de 3V a 5.5V por lo que la conexión hacia el FPGA Cyclone II será de manera directa.
 - Dependiendo del modelo de DHT11 puede que tenga integrado la resistencia de pull-up, sobre todo lo que tienen el sensor montado en una PCB:
 - Hoja técnica: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Datasheet-Translated-Version-1143054.pdf>



Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Esquema en Qsys:

- Se está empleando el modelo NIOSII/e
- La memoria se ha especificado de 8KB (8192 bytes) de tamaño y ubicada en la dirección 0x2000 – 0x3FFF
- El LCD Core se encuentra en la dirección 0x4000
- El PIO que se emplea para la comunicación con el DHT11 es del tipo bidireccional y de un 1bit de tamaño, ubicado en la dirección 0x4100



Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

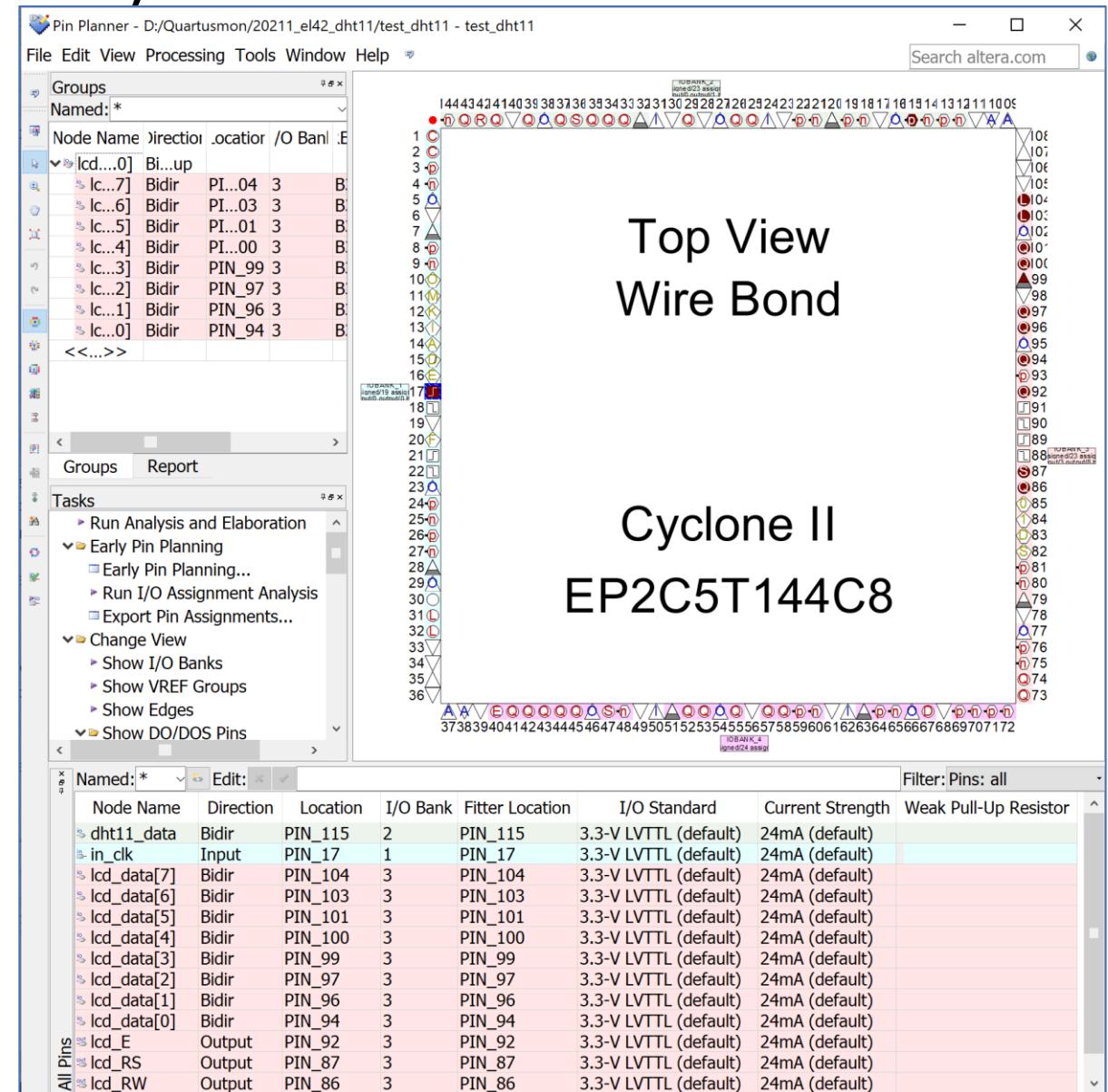
- Instanciación del NIOS II (VHDL estructural)
- Tener en cuenta los puertos de la entidad que se van a conectar al NIOSII deben de respetarse sus tipos y tamaños.

```
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 test_dht11 is
7 port(
8     in_clk      : in  std_logic          := 'X';           -- clk
9     lcd_RS     : out std_logic;          -- RS
10    lcd_RW    : out std_logic;          -- RW
11    lcd_data   : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
12    lcd_E      : out std_logic;          -- E
13    dht11_data : inout std_logic        := 'X'            -- export
14 );
15 end test_dht11;
16
17 architecture estructura of test_dht11 is
18 begin
19     component niosii_dht11_lcd is
20         port (
21             clk_clk      : in  std_logic          := 'X';           -- clk
22             lcd_16207_0_external_RS : out std_logic;          -- RS
23             lcd_16207_0_external_RW : out std_logic;          -- RW
24             lcd_16207_0_external_data : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
25             lcd_16207_0_external_E  : out std_logic;          -- E
26             dht11_data_export       : inout std_logic        := 'X'            -- export
27         );
28     end component niosii_dht11_lcd;
29
30 begin
31     u0 : component niosii_dht11_lcd
32         port map (
33             clk_clk      => in_clk,           -- clk.clk
34             lcd_16207_0_external_RS => lcd_RS,          -- lcd_16207_0_external.RS
35             lcd_16207_0_external_RW => lcd_RW,          -- .RW
36             lcd_16207_0_external_data => lcd_data,        -- .data
37             lcd_16207_0_external_E  => lcd_E,           -- .E
38             dht11_data_export       => dht11_data        -- dht11_data.export
39         );
40 end estructura;
```

Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

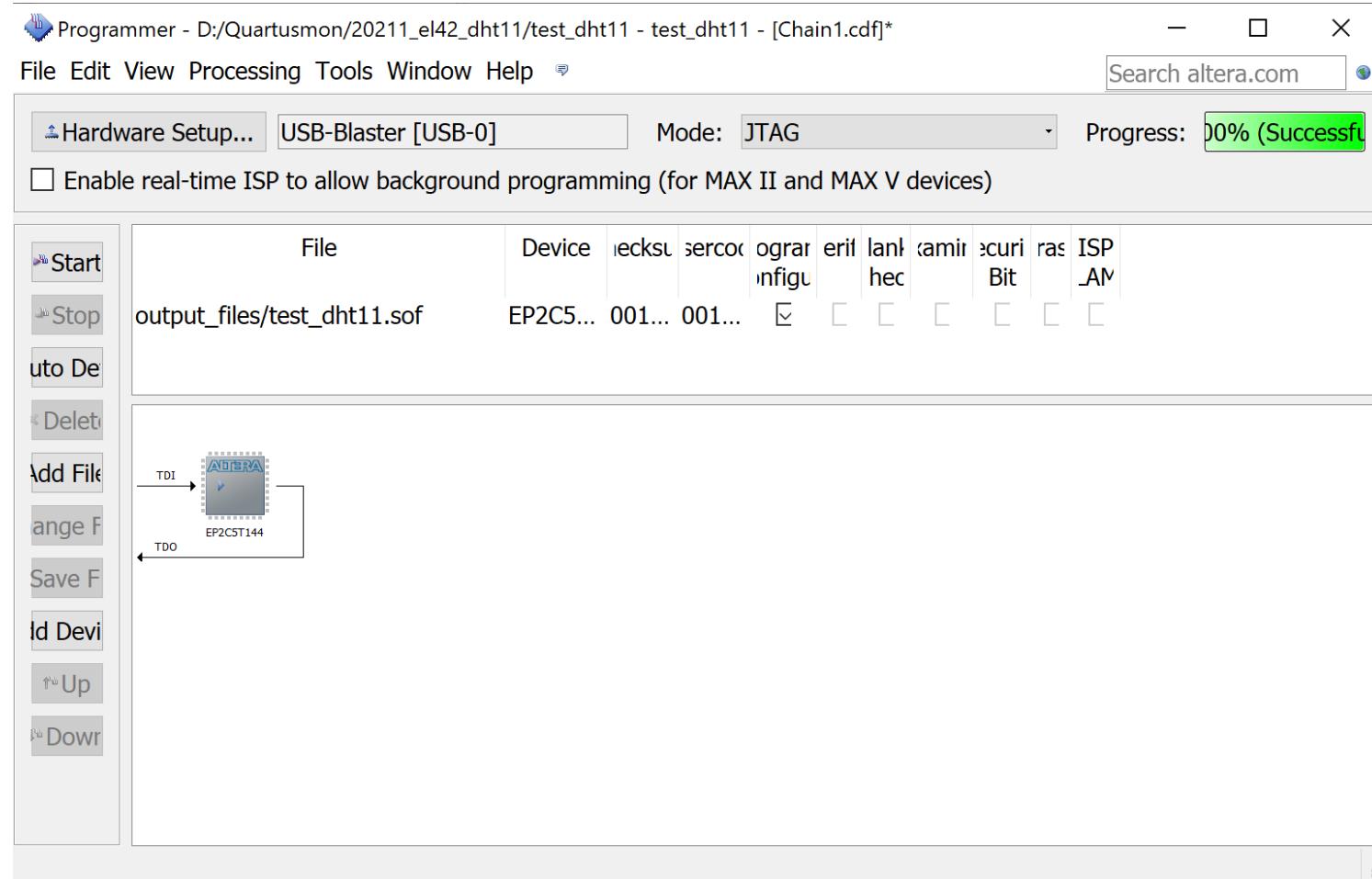
- Asignación de pines en el Pin Planner:

- Al momento de asignar los pines del FPGA con las señales de la entidad tener en consideración el uso de pines I/O y no usar los pines dedicados de reloj u otras funciones.
- Los pines asignados que se muestran fueron validados para este ejemplo y no presentan inconvenientes.



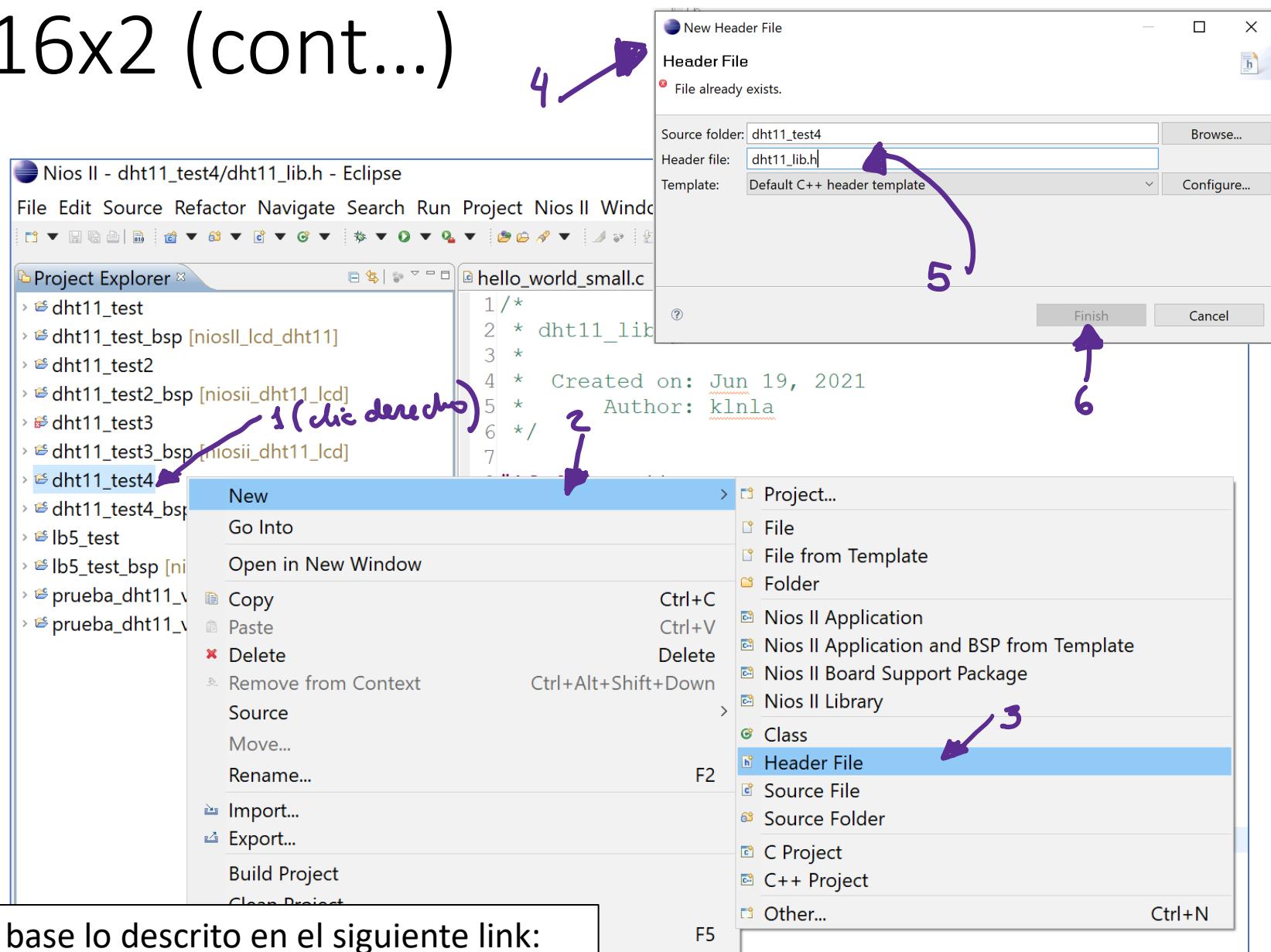
Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Una vez asignados los pines en el Pin Planner se procede a grabar el FPGA con la configuración de hardware desarrollada.
- Luego de grabar satisfactoriamente el FPGA se procederá a hacer la aplicación (software) en el entorno Eclipse.



Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Eclipse: Desarrollo de la librería para DHT11
 - En C para crear una librería se debe de contemplar dos archivos: el “.h” y el “.c”
 - Seguir los siguientes pasos para crear los archivos fuente de la librería para el DHT11. Se deben de crear “dht11lib.h” y “dht11_lib.c”



Nota: La librería fue creada teniendo como base lo descrito en el siguiente link:
<https://www.electronicwings.com/pic/dht11-sensor-interfacing-with-pic18f4550>

Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Eclipse: Desarrollo de la librería para DHT11
 - Una vez creados los archivos .h y .c proceder a colocar las funciones para el DHT11

Contenido de dht11_lib.h

```
8 #ifndef DHT11_LIB_H_
9 #define DHT11_LIB_H_
10
11 #include "sys/alt_stdio.h"
12 #include "system.h"
13 #include "unistd.h"
14 #include "string.h"
15 #include "altera_avalon_pio_regs.h"
16
17 char DHT11_ReadData();
18 void DHT11_Start();
19 void DHT11_CheckResponse();
20
21#endif /* DHT11_LIB_H_ */
```

Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Eclipse: Desarrollo de la librería para DHT11
 - Una vez creados los archivos .h y .c proceder a colocar las funciones para el DHT11

Contenido de dht11_lib.c

```
8 #include "dht11_lib.h"
9
10 char DHT11_ReadData()
11 {
12     char i,data = 0;
13     for(i=0;i<8;i++)
14     {
15         while(!(IORD_ALTERA_AVALON_PIO_DATA(0x4100) & 1));
16         usleep(30);
17         if(IORD_ALTERA_AVALON_PIO_DATA(0x4100) & 1)
18             data = ((data<<1) | 1);
19         else
20             data = (data<<1);
21         while(IORD_ALTERA_AVALON_PIO_DATA(0x4100) & 1);
22     }
23     return data;
24 }
25
26 void DHT11_Start()
27 {
28     IOWR_ALTERA_AVALON_PIO_DIRECTION(0x4100, 1);
29     IOWR_ALTERA_AVALON_PIO_DATA(0x4100, 0);
30     usleep(18000);
31     IOWR_ALTERA_AVALON_PIO_DATA(0x4100, 1);
32     usleep(20);
33     IOWR_ALTERA_AVALON_PIO_DIRECTION(0x4100, 0);
34 }
35
36 void DHT11_CheckResponse()
37 {
38     while(IORD_ALTERA_AVALON_PIO_DATA(0x4100) & 1);
39     while(!(IORD_ALTERA_AVALON_PIO_DATA(0x4100) & 1));
40     while(IORD_ALTERA_AVALON_PIO_DATA(0x4100) & 1);
41 }
```

Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Eclipse: Librería para LCD

lcd_lib.h

```
8 #ifndef LCD_LIB_H
9 #define LCD_LIB_H_
10
11 #include "sys/alt_stdio.h"
12 #include "system.h"
13 #include "unistd.h"
14 #include "string.h"
15 #include "altera_avalon_lcd_16207_regs.h"
16
17 #define LCD_WR_COMMAND_REG 0
18 #define LCD_RD_STATUS_REG 1
19 #define LCD_WR_DATA_REG 2
20 #define LCD_RD_DATA_REG 3
21 #define LCD_0_BASE 0x4000
22
23 void LCD_INIT(void);
24 void LCD_ESCRIBE_MENSAJE(const char *cadena);
25 void LCD_ENVIACHAR(unsigned char caracter);
26 void LCD_CLEAR(void);
27 void LCD_LINE1(void);
28 void LCD_LINE2(void);
29
30#endif /* LCD_LIB_H */
```

lcd_lib.c

```
10 void LCD_INIT(void) {
11     usleep(15000);
12     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
13     usleep(4100);
14     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
15     usleep(100);
16     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
17     usleep(5000);
18     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
19     usleep(100);
20     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
21     usleep(100);
22     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
23     usleep(100);
24     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
25     usleep(100);
26     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
27     usleep(2000);
28     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
29     usleep(2000);
30
31
32 void LCD_ESCRIBE_MENSAJE(const char *cadena) {
33     unsigned char tam;
34     tam = strlen(cadena);
35     unsigned char i=0;
36     for(i=0;i<tam;i++) {
37         IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
38         usleep(100);
39     }
40 }
```

```
42 void LCD_ENVIACHAR(unsigned char caracter) {
43     IOWR(LCD_0_BASE, LCD_WR_DATA_REG, caracter);
44     usleep(100);
45 }
46
47 void LCD_CLEAR(void) {
48     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
49     usleep(2000);
50 }
51
52 void LCD_LINE1(void) {
53     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
54     usleep(2000);
55 }
56
57 void LCD_LINE2(void) {
58     IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC0);
59     usleep(2000);
60 }
```

Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Eclipse: Código fuente de la aplicación

```
1#include "sys/alt_stdio.h"
2#include "system.h"
3#include "unistd.h"
4#include "string.h"
5#include "altera_avalon_lcd_16207_regs.h"
6#include "altera_avalon_pio_regs.h"
7#include "dht11_lib.h"
8#include "lcd_lib.h"
9
10unsigned char centena=0;
11unsigned char decena=0;
12unsigned char unidad=0;
13
14void convierte(unsigned char numero){
15    centena = (numero % 1000) / 100;
16    decena = (numero % 100) / 10;
17    unidad = numero % 10;
18}
```

```
20int main(){
21    alt_putstr("Probando 1 2 3\n");
22    usleep(1000000);
23    alt_putstr("Inicializando aplicacion...\n");
24    usleep(1000000);
25    alt_putstr("Todo listo!\n");
26    LCD_INIT();
27    char RH_Decimal,RH_Integral,T_Decimal,T_Integral;
28    char Checksum;
29    //char value[10];
30    while(1){
31        DHT11_Start();
32        DHT11_CheckResponse();
33        RH_Integral = DHT11_ReadData(); //Recepcion de los 8bytes
34        RH_Decimal = DHT11_ReadData(); //Del DHT11
35        T_Integral = DHT11_ReadData();
36        T_Decimal = DHT11_ReadData();
37        Checksum = DHT11_ReadData();
38        convierte(RH_Integral); //Individualizacion de digitos de RH
39        LCD_LINE1();
40        LCD_ESCRIBE_MENSAJE("RH:");
41        LCD_ENVIACHAR(decena+0x30);
42        LCD_ENVIACHAR(unidad+0x30);
43        LCD_ESCRIBE_MENSAJE("% ");
44        convierte(T_Integral); //Individualizacion de digitos de Temperatura
45        LCD_ESCRIBE_MENSAJE("Temp:");
46        LCD_ENVIACHAR(decena+0x30);
47        LCD_ENVIACHAR(unidad+0x30);
48        LCD_ENVIACHAR(0xDF);
49        LCD_ESCRIBE_MENSAJE("C");
50        LCD_LINE2();
51        LCD_ESCRIBE_MENSAJE("CS:");
52        convierte(Checksum); //Individualizacion de digitos de Checksum
53        LCD_ENVIACHAR(centena+0x30);
54        LCD_ENVIACHAR(decena+0x30);
55        LCD_ENVIACHAR(unidad+0x30);
56        LCD_ENVIACHAR(0x20);
57        //Verificacion de Checksum con visualizacion en el LCD
58        if(Checksum != (RH_Integral + RH_Decimal + T_Integral + T_Decimal)){
59            LCD_CLEAR();
60            LCD_ESCRIBE_MENSAJE("Error!!!");
61        }
62        else{
63            LCD_ESCRIBE_MENSAJE("No Error");
64        }
65        usleep(1000000);
66    }
67}
```

Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2 (cont...)

- Circuito implementado:

