

Aplicaciones con el NIOS II

Por Kalun José Lau Gan

1

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.
- 23-06-2021 01:18 – Mejor redacción y ampliación de detalle en los ejemplos 2 y 3.
- 03-07-2021 00:49 – Ejercicio 4
- 17-06-2022 23:56 – Ejercicio 5
- 18-06-2022 12:07 – Modificación al ejercicio 5: Interrupciones hacia las entradas de: la base de tiempo de 1Hz y los botones de ajuste para la hora actual.

2

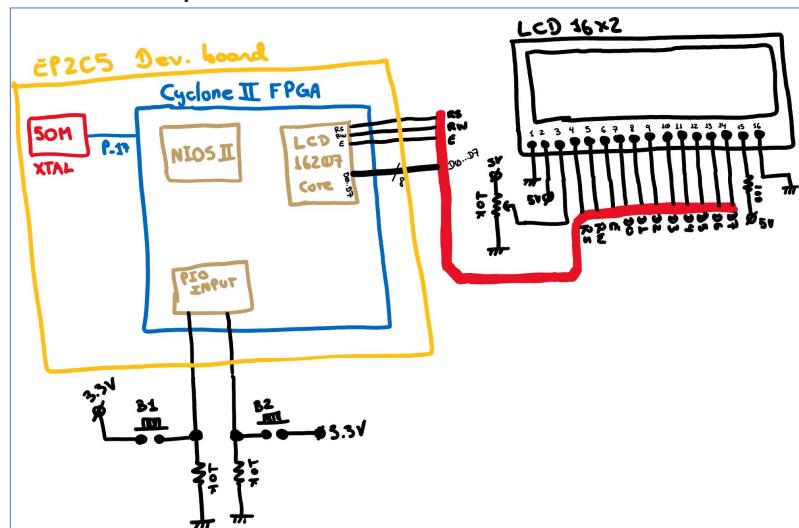
Índice de ejemplos

- Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar del estado de dos pulsadores
- Ejemplo 2: Interface del NIOS II con un LCD 16x2 para visualizar un contador autoincremental, creación de librería lcd_lib
- Ejemplo 3: Interface del NIOS II con un sensor DHT11 y un LCD 16x2, creación de librerías dht11_lib y lcd_lib
- Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 y un LCD16x2
- Ejemplo 5: Interface del NIOS II con módulo DIV_FREQ (VHDL) y un LCD16x2 para función de reloj en formato de 24h y visualización hh:mm:ss

3

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

- Diagrama de bloques



4

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

- Aspectos iniciales:

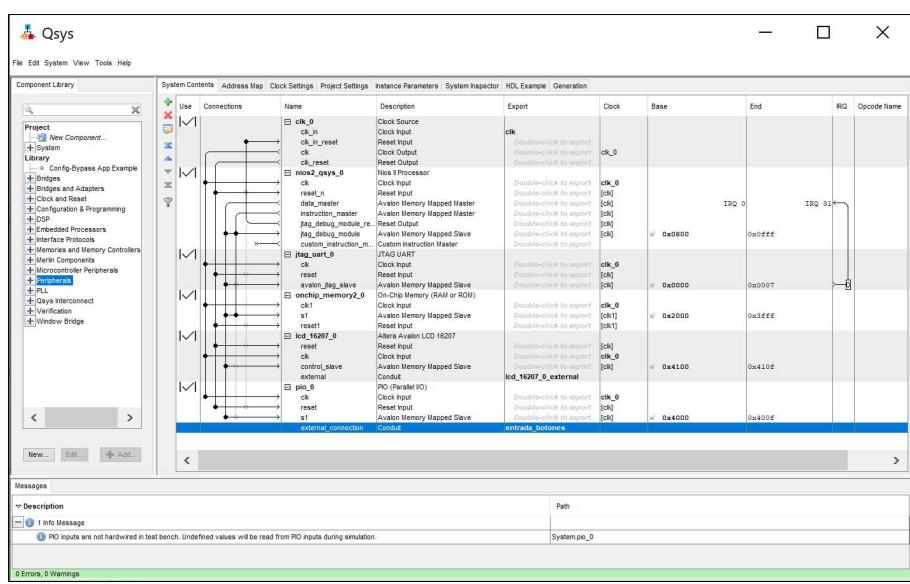
- Se empleará un PIO entrada de 8 bits para la conexión de los dos pulsadores, dichos pulsadores ocuparán los dos bits menos significativos.
- En el LCD se visualizará el estado de ambos botones con mensajes “OFF” cuando no estén presionados y “ON” cuando estén presionados.
- Tener en cuenta que la lógica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

5

Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar del 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



6

Ejemplo 1: Interface del NIOS II con un LCD 16x2 para visualizar del 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;

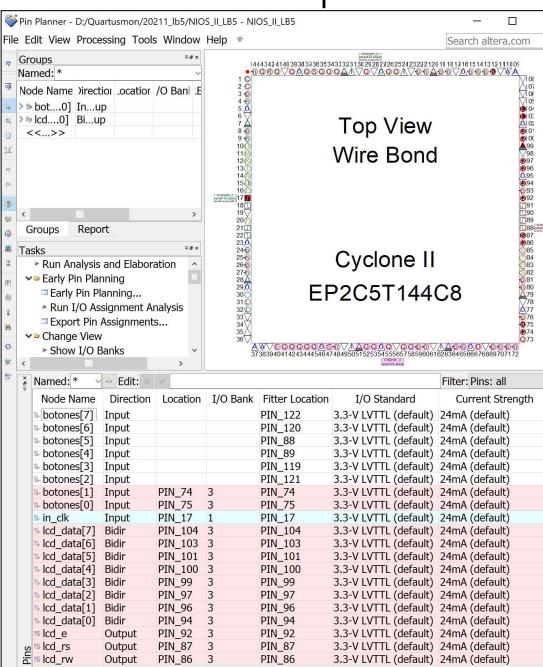
```

7

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.



8

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

- Código en Eclipse:

```

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
13 unsigned char boton1=0;
14 unsigned char boton2=0;
15
16 void 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
38 void 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 }
```

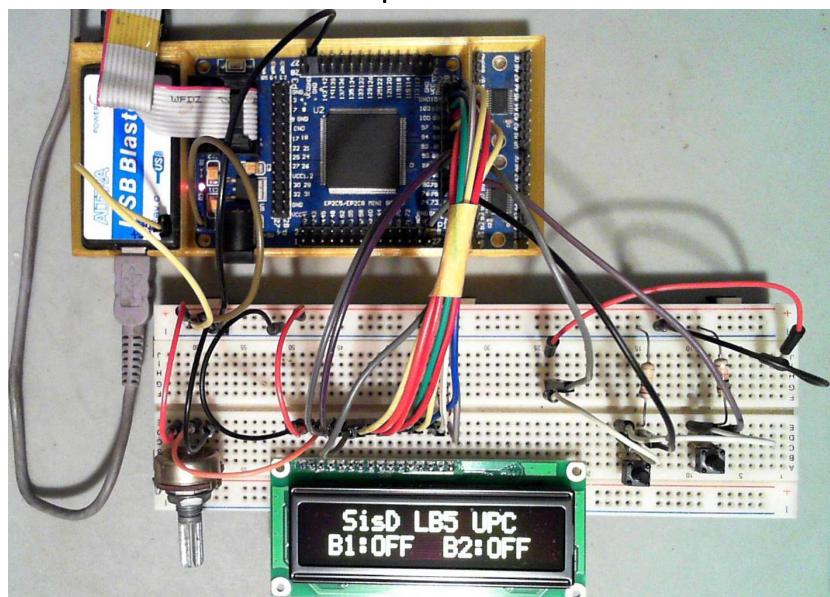
```

46 int 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 }
```

9

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

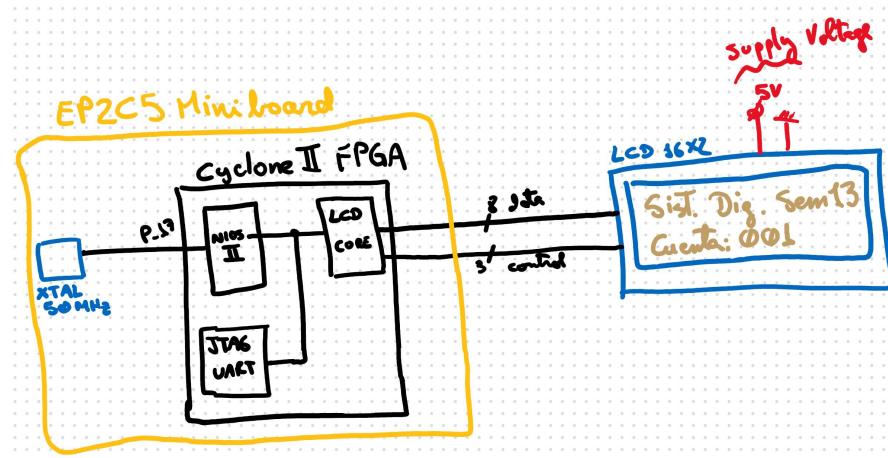
- Circuito implementado:



10

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

- Diagrama de bloques

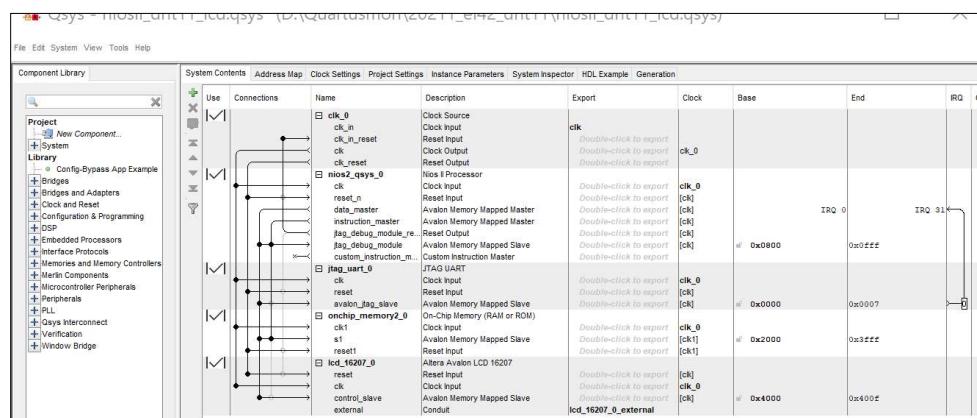


11

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

- Esquema en Qsys:

- El LCD Core esta mapeado en la dirección 0x4000



12

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

- Instanciación en Quartus II:

```

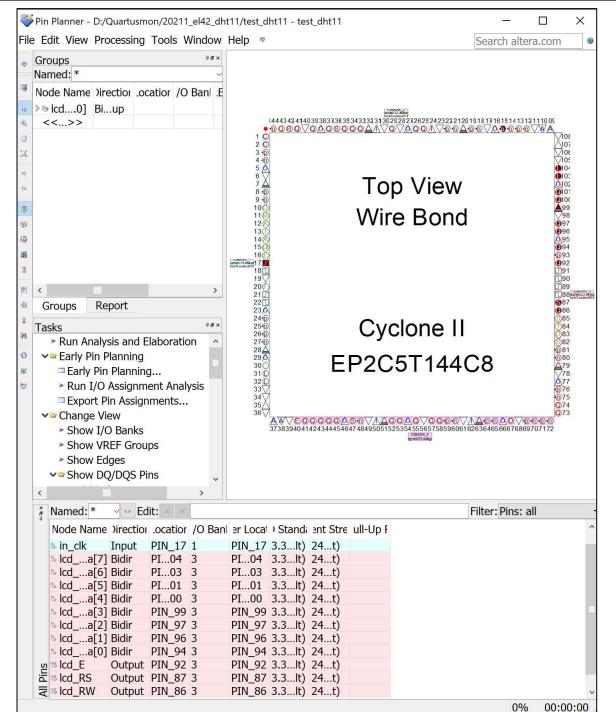
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity test_cuenta is
7   port(
8     in_clk      : in  std_logic;                                -- clk
9     lcd_RS     : out std_logic;                                -- RS
10    lcd_RW     : 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;                                -- 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;

```

13

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

- Asignación de pines en Pin Planner:



14

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

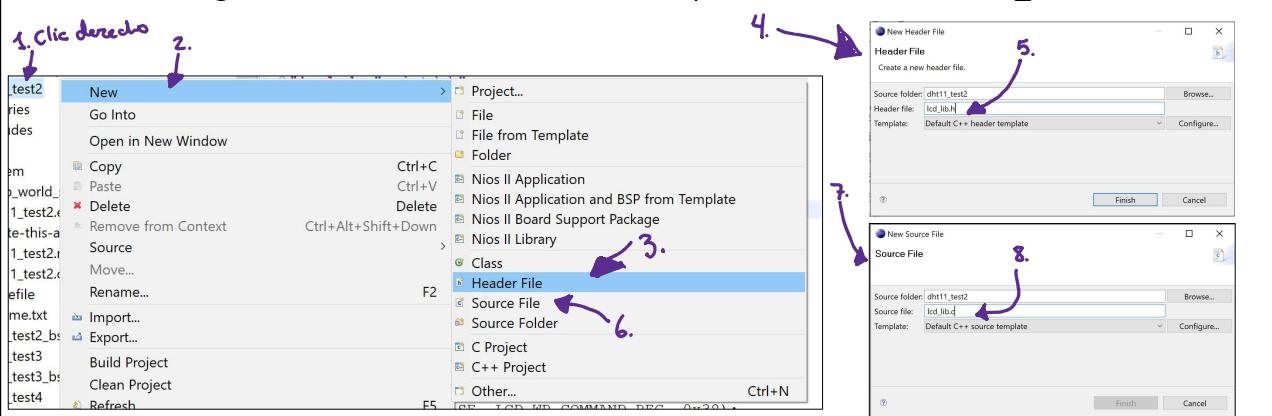
- Software de la aplicación en Eclipse: Librería “lcd_lib” para el control del LCD 16x2

| | |
|---|--|
| <pre>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 23void LCD_init(void); 24void LCD_ESCRIBE_MENSAJE(const char *cadena); 25void LCD_CHAR_SEND(unsigned char info); 26void LCD_CLEAR(void); 27void LCD_LINE1(void); 28void LCD_LINE2(void); 29 30#endif /* LCD_LIB_H_ */</pre> | <pre>lcd_lib.c 1 #include "lcd_lib.h" 2 3 void LCD_init(void) { 4 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38); 5 usleep(15000); 6 usleep(4100); 7 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38); 8 usleep(100); 9 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38); 10 usleep(5000); 11 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38); 12 usleep(100); 13 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08); 14 usleep(100); 15 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C); 16 usleep(100); 17 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06); 18 usleep(100); 19 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02); 20 usleep(2000); 21 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01); 22 usleep(2000); 23} 24 25void LCD_CLEAR(void) { 26 IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC0); 27 usleep(2000); 28} 29 30void LCD_ESCRIBE_MENSAJE(const char *cadena) { 31 unsigned char tam; 32 tam = strlen(cadena); 33 unsigned char i=0; 34 for(i=0;i<tam;i++) { 35 IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]); 36 usleep(100); 37 } 38} 39}</pre> |
|---|--|

15

Ejemplo 2: Interface del NIOS II con 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”



16

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

- Software de la aplicación en Eclipse: Código principal
- Se está empleando una función convierte que permitirá obtener los dígitos individuales (centena, decena y unidad) de la variable "cuenta" a fin de que se pueda visualizar en el LCD.
- Tener en cuenta que para visualizar cada dígito se deberá de sumar 0x30 para que pueda imprimirse el dígito en el LCD según su ROM de caracteres (muy similar a la table ASCII)

```

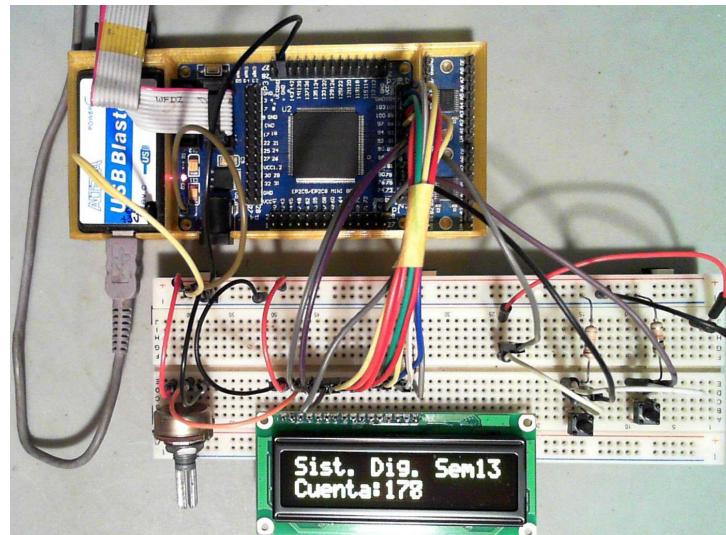
1#include "sys/alt_stdio.h"
2#include "system.h"
3#include "unistd.h"
4#include "string.h"
5#include "altera_avalon_lcd_16207_regs.h"
6
7#include "lcd_lib.h"
8
9unsigned char cuenta=0;
10unsigned char centena=0;
11unsigned char decena=0;
12unsigned char unidad=0;
13
14void convierte(unsigned char numero){
15    centena = (numero % 1000) / 100;
16    decena = (numero % 100) / 10;
17    unidad = numero % 10;
18}
19
20int 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}

```

17

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

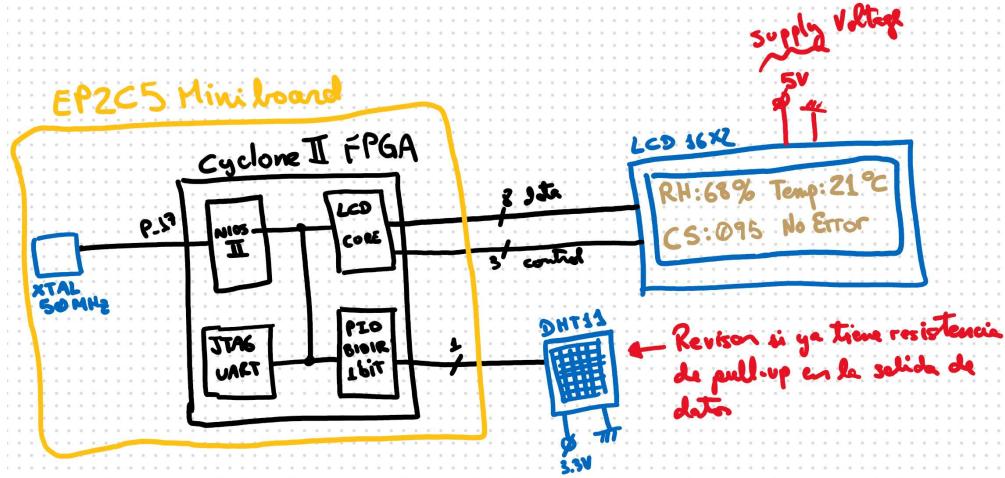
- Circuito implementado:



18

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

- Diagrama de bloques

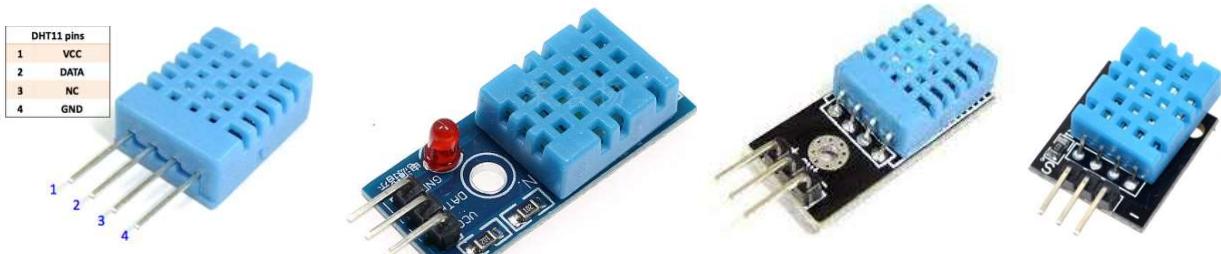


19

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

- Aspectos iniciales:

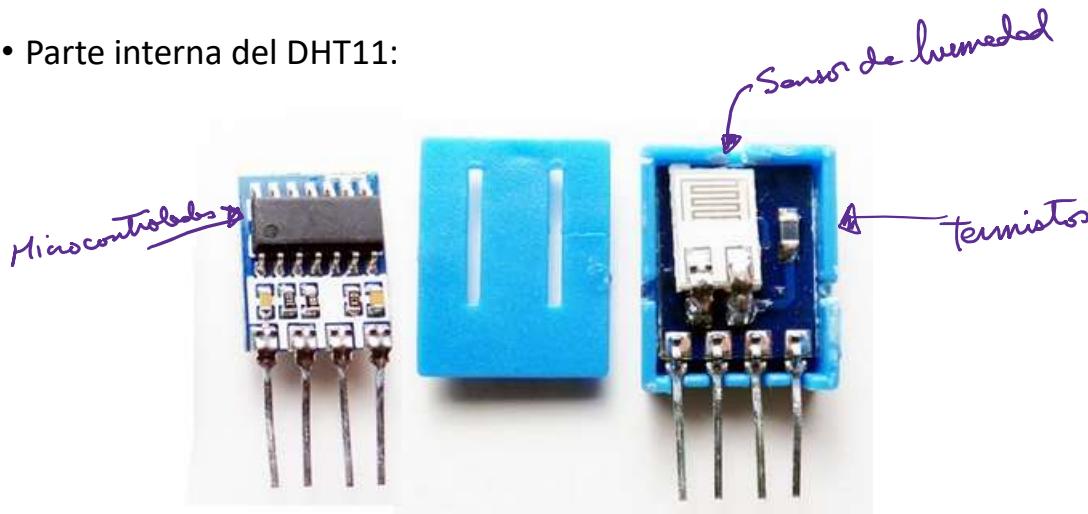
- Al revisar la hoja técnica del DHT11 podemos ver que el DHT11 tiene un rango de voltaje de operación de 3V a 5.5V por lo que la conexión hacia el FPGA Cyclone II será de manera directa.
- Dependiendo del modelo de DHT11 puede que tenga integrado la resistencia de pull-up, sobre todo lo que tienen el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



20

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

- Parte interna del DHT11:

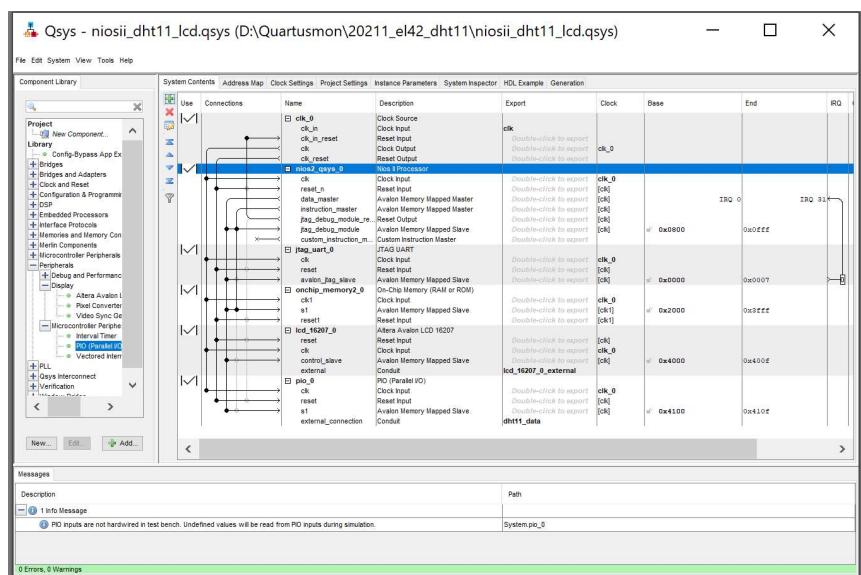


21

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

• Esquema en Qsys:

- Se esta 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



22

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.

```

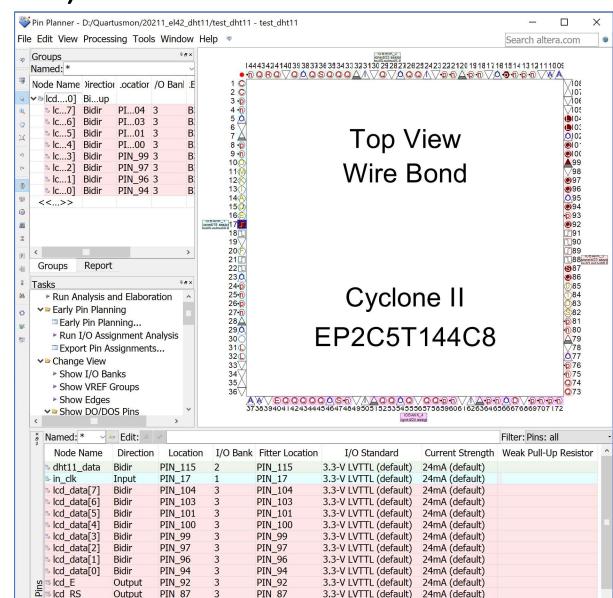
1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity test_dht11 is
7   port(
8     in_clk      : in  std_logic          := 'X';           -- clk
9     lcd_RS     : out std_logic;          -- RS
10    lcd_RW     : out std_logic;          -- 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   component niosii_dht11_lcd is
19     port (
20       clk_clk      : in  std_logic          := 'X';           -- clk
21       lcd_16207_0_external_RS  : out std_logic;          -- RS
22       lcd_16207_0_external_RW  : out std_logic;          -- RW
23       lcd_16207_0_external_data : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
24       lcd_16207_0_external_E   : out std_logic;          -- E
25       dht11_data_export       : inout std_logic        := 'X'           -- export
26     );
27   end component niosii_dht11_lcd;
28
29 begin
30   u0 : component niosii_dht11_lcd
31     port map (
32       clk_clk      => in_clk,           -- clk.clk
33       lcd_16207_0_external_RS  => lcd_RS,           -- lcd_16207_0_external.RS
34       lcd_16207_0_external_RW  => lcd_RW,           -- .RW
35       lcd_16207_0_external_data => lcd_data,         -- .data
36       lcd_16207_0_external_E   => lcd_E,            -- .E
37       dht11_data_export       => dht11_data        -- dht11_data.export
38     );
39 end estructura;

```

23

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

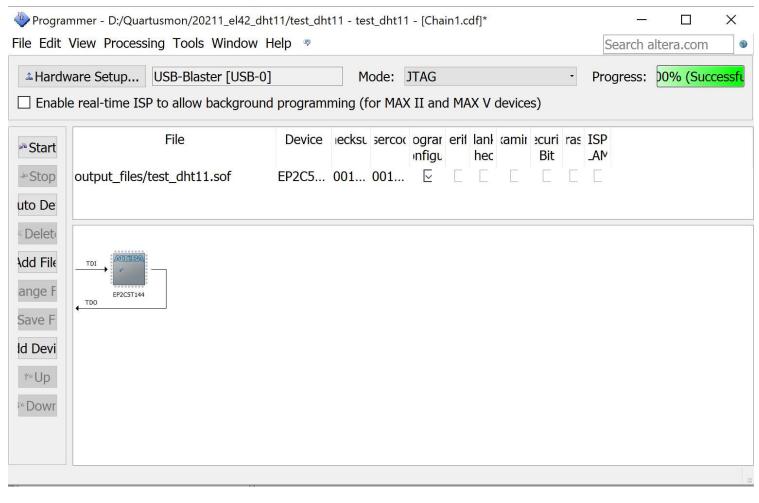
- Asignación de pines en el Pin Planner:
 - Al momento de asignar los pines del FPGA con las señales de la entidad tener en consideración el uso de pines I/O y no usar los pines dedicados de reloj u otras funciones.
 - Los pines asignados que se muestran fueron validados para este ejemplo y no presentan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



24

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

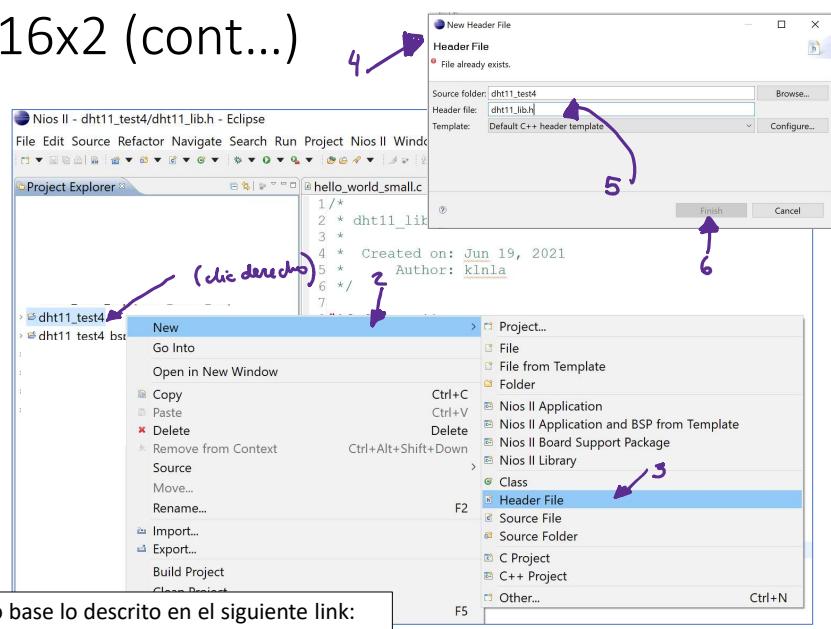
- Una vez asignados los pines en el Pin Planner se procede a grabar el FPGA con la configuración de hardware desarrollada.
- Luego de grabar satisfactoriamente el FPGA se procederá a hacer la aplicación (software) en el entorno Eclipse.



25

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 "dht11_lib.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>

26

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
17char DHT11_ReadData();
18void DHT11_Start();
19void DHT11_CheckResponse();
20
21#endif /* DHT11_LIB_H_ */

```

27

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
10char 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
26void 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
36void 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}

```

28

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_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 }
```

29

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

- Eclipse: Código fuente de la aplicación
 - Empleando librerías dht11_lib y lcd_lib
 - Tener en cuenta que el DHT11 actualiza sus datos cada un segundo

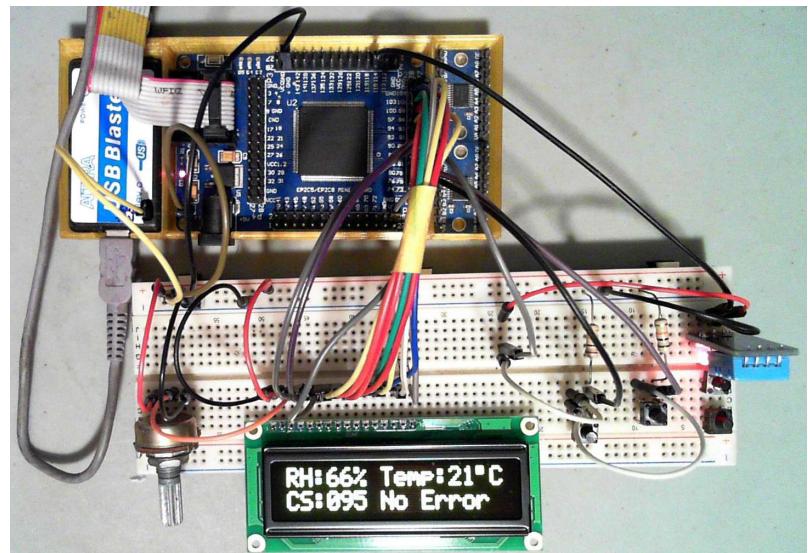
```
1 #include "sys/alt_stdio.h"
2 #include "system.h"
3 #include "unistd.h"
4 #include "string.h"
5 #include "altera_avalon_lcd_16207_regs.h"
6 #include "altera_avalon_pio_regs.h"
7 #include "dht11.lib.h"
8 #include "lcd_lib.h"
9
10 unsigned char centena=0;
11 unsigned char decena=0;
12 unsigned char unidad=0;
13
14 void convierte(unsigned char numero){
15     centena = (numero % 1000) / 100;
16     decena = (numero % 100) / 10;
17     unidad = numero % 10;
18 }
```

```
20 int main(){
21     alt_putstr("Probando 1 2 3\n");
22     usleep(1000000);
23     alt_putstr("Iniciando 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(unidad+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(0x0F);
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); //Envio de valor de Checksum
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 }
```

30

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

- Circuito implementado:



31

Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 (ultrasonido para medición de distancia) y un LCD 16x2

- El sensor HC-SR04 es un módulo que permite medir la distancia entre éste y un objeto entre un rango de 3cm y 300cm.
- Trabaja con alimentación de 5V por lo que se necesitará de un conversor de niveles lógicos 3.3V-5V, se recomienda emplear el TXS0108
- Los transductores trabajan en 40KHz

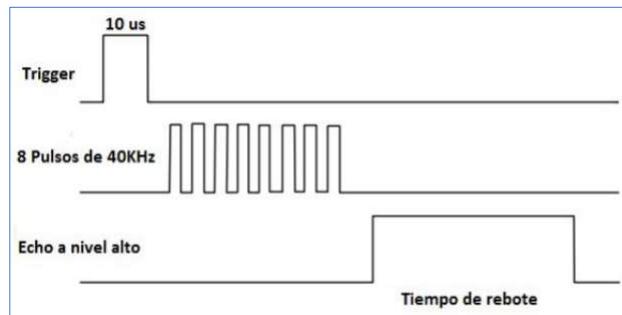


32

Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 (ultrasonido para medición de distancia) y un LCD 16x2

- Posee dos puertos:

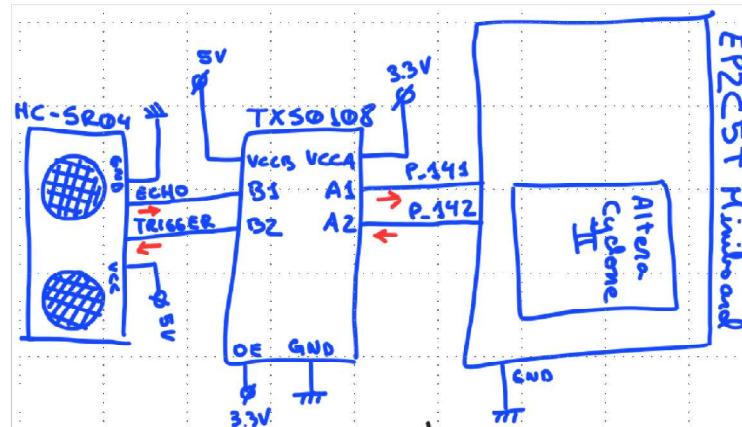
- Trigger: El microcontrolador host le envía al puerto “trigger” del HC-SR04 un pulso activo en alto de 10us para que este último envíe ocho pulsos de sonido de 40KHz.
- Echo: Luego del envío de los ocho pulsos de sonido de 40KHz el HC-SR04 enviará un pulso al microcontrolador host con determinado ancho, este ancho representará el tiempo del eco y mediante un cálculo matemático se obtendrá la distancia entre el módulo y el objeto.



33

Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 (ultrasonido para medición de distancia) y un LCD 16x2

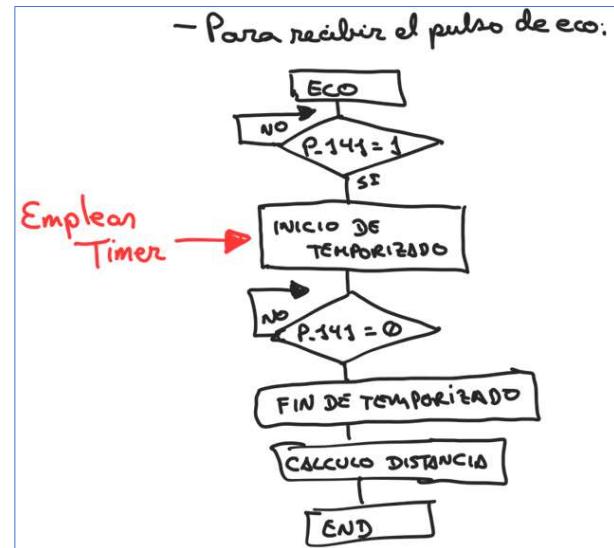
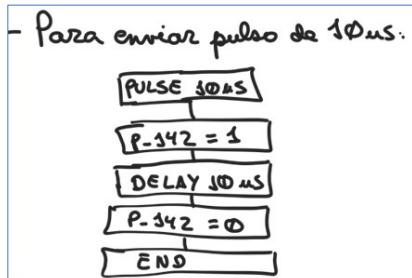
- Interface del sensor HC-SR04 con el FPGA Cyclone II empleando el conversor de niveles lógicos TXS0108



34

Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 (ultrasonido para medición de distancia) y un LCD 16x2

- Algoritmo para obtener la distancia



35

Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 (ultrasonido para medición de distancia) y un LCD 16x2

- Cálculos matemáticos para obtener la distancia (según hoja técnica del HC-SR04)
- Velocidad del sonido: 340m/s, 29.412us por centímetro

Duración del ancho de pulso:

$$T_{dur} = \text{cuenta del timer} \times \text{base de Tiempo}$$

Fórmula para el cálculo de la distancia:

$$\text{distancia} = T_{dur} \times \frac{340 \text{ (m/s)}}{2}$$

36

Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 (ultrasonido para medición de distancia) y un LCD 16x2

- Según lo anterior podemos usar la siguiente fórmula:

$$D_{cm} = \left(\frac{\left(\frac{Pulso_{\mu s}}{2} \right)}{29} \right)$$

- Si obtenemos 100μs en el ida y vuelta del sonido, empleando la fórmula anterior obtendremos 1.7 centímetros.

37

Ejemplo 4: Interface del NIOS II con un sensor HC-SR04 (ultrasonido para medición de distancia) y un LCD 16x2

- Código propuesto:

```
/*propuesta inicial de funciones para el HC-SR04, no se han realizado pruebas aun*/
void pulse_out_10us(void){
    IOWR_ALTERA_AVALON_PIO_DATA([address], 1);
    usleep(10);
    IOWR_ALTERA_AVALON_PIO_DATA([address], 0);
}

/*la función siguiente no es muy preciso debido al tiempo de ejecución de
las instrucciones, para mejorar dicha precision se deberá de emplear un
temporizador independiente*/

unsigned int measure(void){
    pulse_out_10us();
    unsigned int cuentas=0;
    float distance_cm=0;
    while(IORD_ALTERA_AVALON_PIO_DATA([address]) == 0);
    while(IORD_ALTERA_AVALON_PIO_DATA([address]) == 1){
        cuentas++;
        usleep(1);
    }
    distance_cm = ((cuentas /2) / 29);
    return distance_cm;
}
```

38

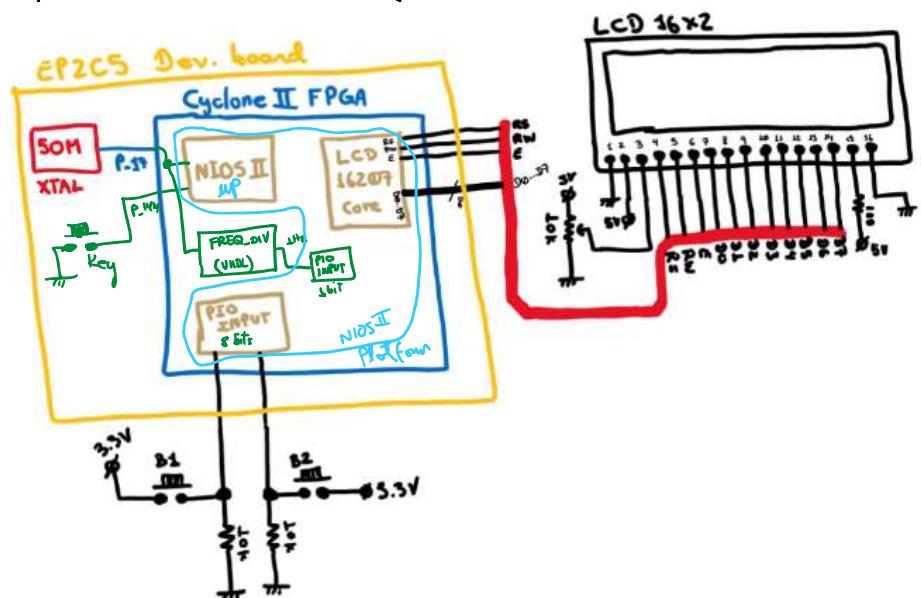
Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Para la base de tiempo se ha implementado un divisor de frecuencia en VHDL el cual recibe los 50MHz provenientes del oscilador (PIN_17) y arroja una señal a 1Hz.
- La señal de 1Hz será ingresado a la plataforma NIOSII a través de un PIO entrada de un bit.
- Esta utilizando la modalidad “polling” para preguntar si el PIO entrada de un bit hubo cambio y hacer el incremento de los segundos. Queda pendiente habilitar interrupciones para mejorar el desempeño de la aplicación.
- Se estará visualizando la hora en la segunda línea del LCD 16x2
- ~~Esta pendiente el ajuste para la hora actual por lo que se esta colocando una hora inicial de manera manual en la declaración de las variables de hora y minuto.~~

39

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ DIV 50MHz-1Hz

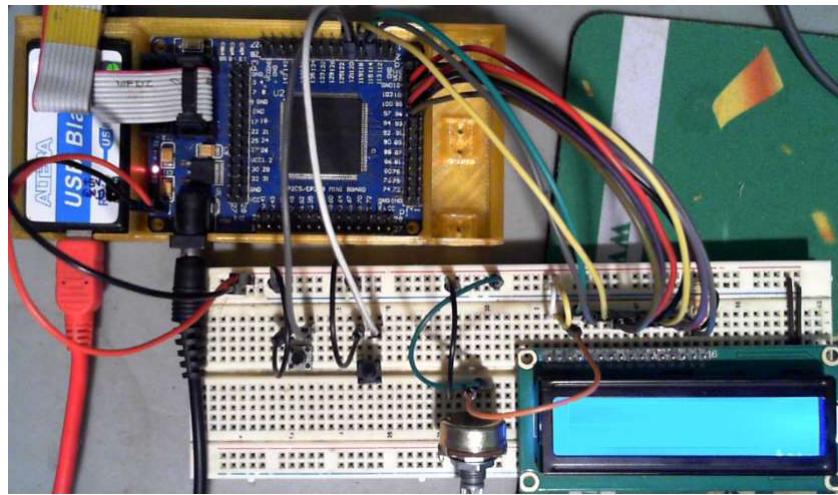
- Circuito propuesto:



40

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Circuito propuesto:

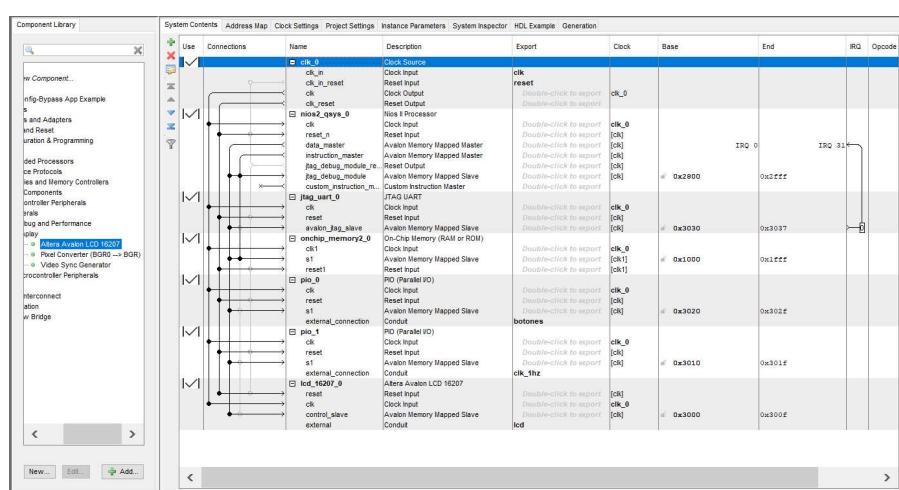


41

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Qsys:

- PIO_1 es entrada de un bit donde se ingresará la señal proveniente del FREQ_DIV
- PIO_0 es entrada de ocho bits donde están conectados los dos botones de ajuste para la hora



42

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Código en VHDL del FREQ_DIV:

```

1 library IEEE;
2 use IEEE.std_logic_arith.all;
3 use IEEE.std_logic_1164.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity freq_div is
7 port( in_clk: in std_logic;
8       out_clk: out std_logic);
9 end freq_div;
10
11 architecture descripcion of freq_div is
12
13 signal cuenta: std_logic_vector(24 downto 0);
14 signal interno: std_logic;
15
16 begin
17 process(in_clk)
18 begin
19 if rising_edge(in_clk) then
20 if cuenta = 25000000 then
21   cuenta <= (others => '0');
22   interno <= not interno;
23 else
24   cuenta <= cuenta + 1;
25 end if;
26 end process;
27 out_clk <= interno;
28 end descripcion;

```

43

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- VHDL estructural:

```

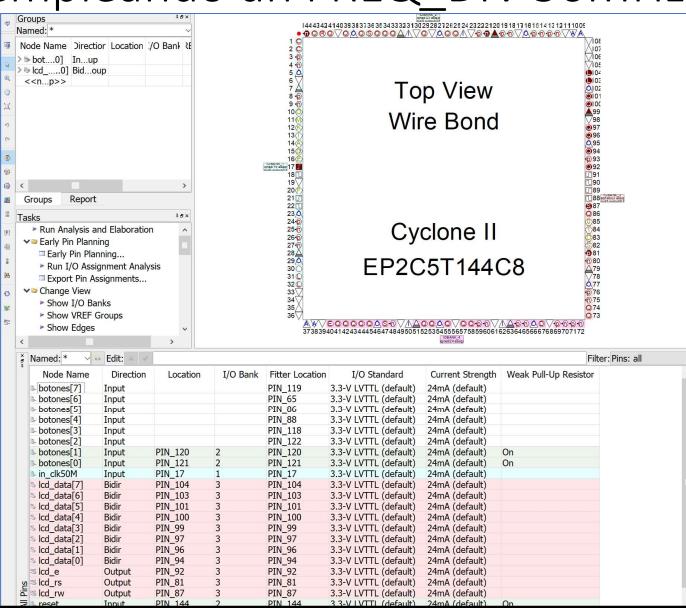
1 library IEEE;
2 use IEEE.std_logic_arith.all;
3 use IEEE.std_logic_1164.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity nios_reloj is
7 port( in_clk50M, reset:      in std_logic;
8       botones:        in std_logic_vector(7 downto 0);
9       lcd_rs, lcd_rw, lcd_e: out std_logic;
10      lcd_data:       inout std_logic_vector(7 downto 0));
11 end nios_reloj;
12
13 architecture estructura of nios_reloj is
14
15 signal clk_1hz:  std_logic;
16
17 component freq_div is
18 port( in_clk: in std_logic;
19       out_clk: out std_logic);
20 end component freq_div;
21
22 component nios_clock is
23 port(
24   clk_clk      : in  std_logic           := 'X';
25   reset_reset_n : in  std_logic           := 'X';
26   clk_1hz_export : in  std_logic           := 'X';
27   botones_export : in  std_logic_vector(7 downto 0) := (others => 'X');
28   lcd_RS       : out std_logic;
29   lcd_RW       : out std_logic;
30   lcd_data     : inout std_logic_vector(7 downto 0) := (others => 'X');
31   lcd_E        : out std_logic
32 );
33 end component nios_clock;
34
35 begin
36   u0 : component nios_clock
37   port map (
38     clk_clk      => in_clk50M,
39     reset_reset_n => reset, --
40     clk_1hz_export => clk_1hz, --
41     botones_export => botones, --
42     lcd_RS       => lcd_rs,
43     lcd_RW       => lcd_rw,
44     lcd_data     => lcd_data,
45     lcd_E        => lcd_e
46   );
47
48   u1 : component freq_div
49   port map (
50     in_clk      => in_clk50M,
51     out_clk    => clk_1hz
52   );
53
54 end estructura;

```

44

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

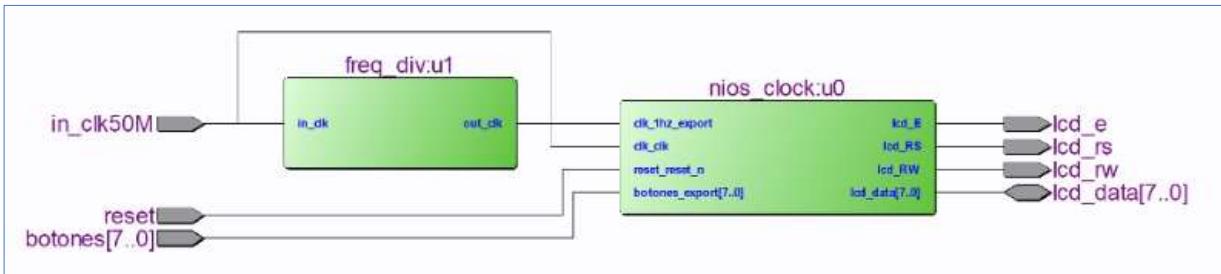
- PinPlanner:



45

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

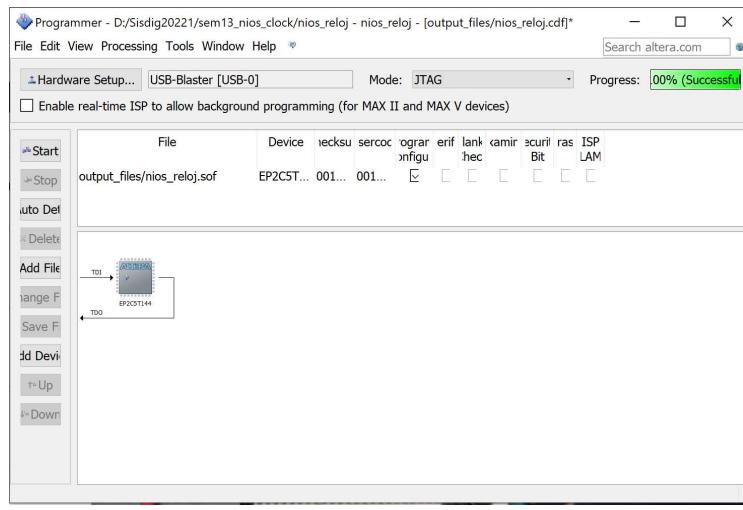
- Vista RTL del sistema NIOSII + FREQ_DIV:



46

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Programación del FPGA:



47

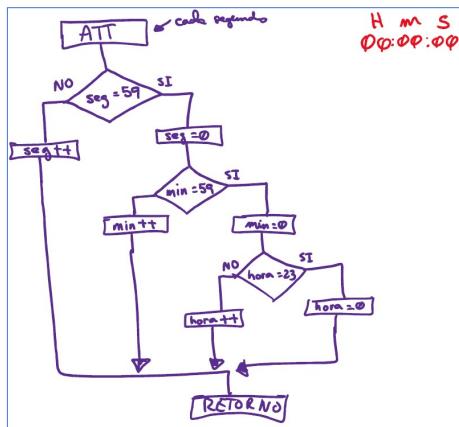
Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Sistema de cuenta de un reloj:
- Sistema de 24 horas:
 - 00:00:00 – hh:mm:ss
 - 23:59:59 – hh:mm:ss
- Sistema de 12 horas:
 - 01:00:00 – hh:mm:ss
 - 12:59:59 – hh:mm:ss

48

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Algoritmo en flowchart y pseudocódigo para el sistema de cuenta de un reloj de formato 24h:



```

if(CLK_1HZ == 1){
    if(segundos == 59){
        segundos = 0;
        if(minutos == 59){
            minutos = 0;
            if(horas == 23){
                horas = 0;
            } else{
                horas++;
            }
        } else{
            minutos++;
        }
    } else{
        segundos++;
    }
}
    
```

49

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Desarrollo en Eclipse, se está empleando 'polling' para monitorear la entrada de reloj de 1Hz

```

#include "sys/alt_stdio.h"
#include "system.h"
#include "string.h"
#include "unistd.h" //usleep
#include "altera_avalon_pio_regs.h" //PIO
#include "altera_avalon_lcd_1e207_regs.h" //LCD
//declaración de variables globales
unsigned char horas = 21;
unsigned char minutos = 56;
unsigned char segundos = 0;
unsigned char decena = 0;
unsigned char unidad = 0;
unsigned char indicador = 0;

void convierte(unsigned char numero) {
    decena = numero / 10;
    unidad = numero % 10;
}

void lcd_init(void) {
    usleep(15000);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x3B);
    usleep(4100);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x3B);
    usleep(1100);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x3B);
    usleep(5000);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x3B);
    usleep(1000);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0B);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0B);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
    usleep(2000);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0B);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0B);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
    usleep(1000);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[0]);
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[1]);
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[2]);
    usleep(100);
}

void ESCRIBE_MENSAJE(const char *cadena) {
    unsigned char i=0;
    unsigned char tam;
    tam = strlen(cadena);
    for(i=0;i<tam;i++){
        IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
        usleep(100);
    }
}

void ENVIA_CHAR(unsigned char caracter) {
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, caracter);
    usleep(100);
}
    
```

50

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Pruebas iniciales: mensajes enviados hacia la consola de NIOSII

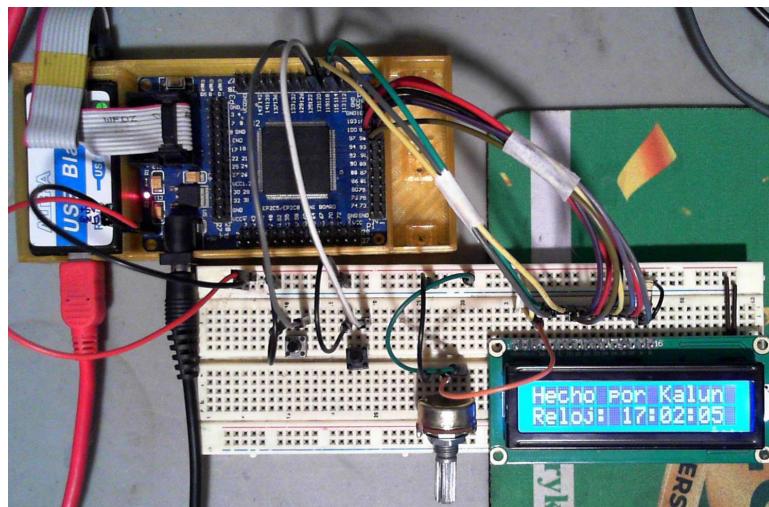


The screenshot shows a terminal window titled "Nios II Console". The title bar includes tabs for "Problems", "Tasks", "Console", "Properties", and "Nios II Console". The main area of the window displays the following text:
nios relojito - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 name: jtaguart_0
Hello from Nios II!
Hola UPCino, soy el NIOS II
Hecho por Kalun Lau
Prueba Semana 13
Aplicación de reloj

51

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

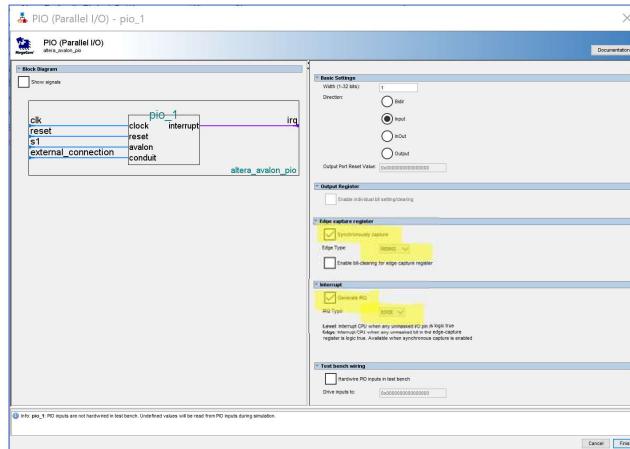
- Pruebas iniciales: Funcionamiento en el circuito en físico



52

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Añadiendo la función de interrupción para el PIO entrada de 1Hz
 - Se debe de modificar la configuración del PIO entrada de 1Hz en el Qsys para que tenga la función de interrumpir al CPU:



53

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Añadiendo la función de interrupción para el PIO entrada de 1Hz
 - En el diagrama de conexiones del Qsys se deberá asignar un IRQ al PIO entrada de un bit donde se esta recibiendo la señal de 1Hz proveniente del FREQ_DIV, en este ejemplo se asignó IRQ1



54

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Añadiendo la función de interrupción para el PIO entrada de 1Hz

- Funciones de:

1. Configuración o inicialización de la interrupción (debe de ser llamado en el main)
2. Rutina de la interrupción (lo que debe de hacer ante un evento de interrupción)

```
77//funcion para inicializar la interrupcion del PIO de 1Hz
78void init_pio_1hz(void){
79    void* edge_capture_ptr = (void*) &edge_capture;
80    IOWR_ALTERA_AVALON_PIO_IRQ_MASK(0x3010, 0x01);
81    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(0x3010, 0x00);
82    alt_irq_register( 1,
83                      edge_capture_ptr,
84                      handle_lhz_interrupts );
85}
```

Referencia:

<https://www.intel.com/content/www/us/en/docs/programmable/683282/current/an-isr-to-service-a-button-pio-interrupt.html>

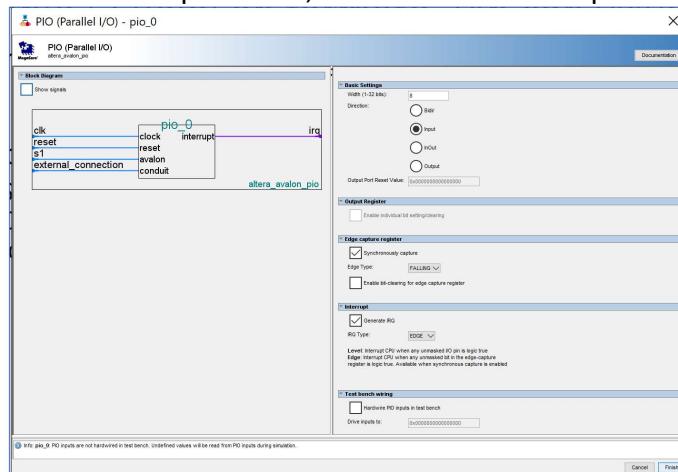
```
25//funcion para la interrupcion del PIO donde se esta ingresando
26//la señal de 1Hz proveniente del FREQ DIV
27void handle_lhz_interrupts(void* context, alt_u32 id){
28    volatile int* edge_capture_ptr = (volatile int*) context;
29    *edge_capture_ptr = IORD_ALTERA_AVALON_PIO_EDGE_CAP(0x3010);
30    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(0x3010, 0x00);
31    if(segundos == 59){
32        segundos = 0;
33        if(minutos == 59){
34            minutos = 0;
35            if(horas == 23){
36                horas = 0;
37            }
38            else{
39                horas = horas + 1;
40            }
41        }
42        else{
43            minutos = minutos + 1;
44        }
45    }
46    else{
47        segundos = segundos + 1;
48    }
49}
```

55

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Añadiendo la función de interrupción para el PIO entrada de los botones

- Se debe de modificar la configuración del PIO entrada de los botones en el Qsys para que tenga la función de interrumpir al CPU, teniendo en cuenta que los botones son activos en bajo:

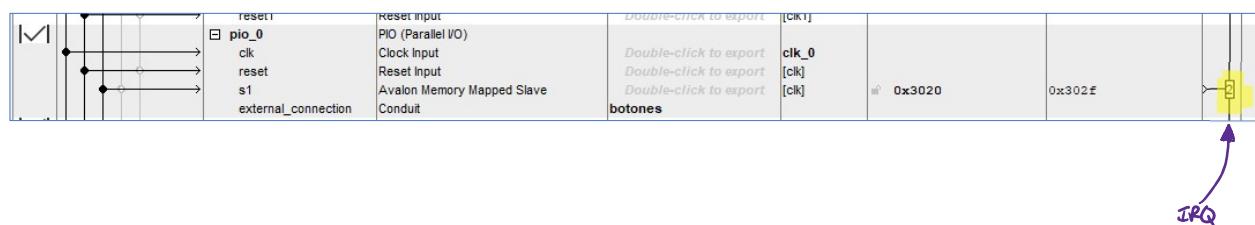


56

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Añadiendo la función de interrupción para el PIO entrada de 1Hz

- En el diagrama de conexiones del Qsys se deberá asignar un IRQ al PIO entrada de ocho bits donde se están conectados los dos botones de ajuste para la hora actual, en este ejemplo se asignó IRQ2



57

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Añadiendo la función de interrupción para el PIO entrada de botones

- Funciones de:

1. Configuración o inicialización de la interrupción (debe de ser llamado en el main)
2. Rutina de la interrupción (lo que debe de hacer ante un evento de interrupción)

```

87//funcion para inicializar la interrupcion del PIO de los botones
88void init_pio_setclk(void){
89    void* edge_capture_ptr = (void*) &edge_capture;
90    IOWR_ALTERA_AVALON_PIO_IRQ_MASK(0x3020, 0x03);
91    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(0x3020, 0x00);
92    alt_irq_register( 2,
93                      edge_capture_ptr,
94                      handle_setclk_interrupts );
95}

```

51//funcion para la interrupcion del PIO donde se esta ingresando
52//los dos botones activos en bajo para configurar las horas y los minutos

```

53void handle_setclk_interrupts(void* context, alt_u32 id){
54    volatile int* edge_capture_ptr = (volatile int*) context;
55    *edge_capture_ptr = IORD_ALTERA_AVALON_PIO_EDGE_CAP(0x3020);
56    IOWR_ALTERA_AVALON_PIO_EDGE_CAP(0x3020, 0x00);
57    if((IORD_ALTERA_AVALON_PIO_DATA(0x3020) & 0x01) == 0){
58        if(horas == 23){
59            horas = 0;
60        }
61        else{
62            horas = horas + 1;
63        }
64        usleep(80000);
65    }
66    if(((IORD_ALTERA_AVALON_PIO_DATA(0x3020) & 0x02) >> 1) == 0){
67        if(minutos == 59){
68            minutos = 0;
69        }
70        else{
71            minutos = minutos + 1;
72        }
73        usleep(80000);
74    }
75}

```

Referencia:

<https://www.intel.com/content/www/us/en/docs/programmable/683282/current/an-isr-to-service-a-button-pio-interrupt.html>

58

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Código completo contemplando interrupciones

```

1#include "sys/alt_stdio.h"
2#include "sys/alt_irq.h"
3#include "alt_types.h"
4#include "system.h"
5#include "string.h"                                //strlen
6#include "unistd.h"                               //usleep
7#include "altera_avalon_pio_regs.h"               //PIO
8#include "altera_avalon_lcd_1620t_regs.h"          //LCD
9
10//sección de configuración del LCD
11#define LCD_WR_COMMAND_REG 0
12#define LCD_RD_DATA_REG 1
13#define LCD_WR_DATA_REG 2
14#define LCD_RD_DATA_REG 3
15#define LCD_O_BASE 0x3000
16
17//declaración de variables globales
18unsigned char horas = 10;
19unsigned char minutos = 10;
20unsigned char segundos = 20; //hora inicial 10:10:20
21unsigned char decena = 0;
22unsigned char unidad = 0;
23unsigned char indicador = 0;
24volatile int edge_capture;
25
26//función para la interrupción del PIO donde se está ingresando
27//la señal de 1Hz proveniente del FREQ DIV
28//en esta función se encuentra el sistema de cuenta del reloj
29void handle_lhz_interrupts(void* context, alt_u32 id){
30    volatile int* edge_capture_ptr = (volatile int*) context;
31    *edge_capture_ptr = IORD ALTERA AVALON PIO_EDGE_CAP(0x3010);
32    IOWR ALTERA AVALON PIO_EDGE_CAP(0x3010, 0x00);
33    if(segundos == 59){
34        segundos = 0;
35        if(minutos == 59){
36            minutos = 0;
37            if(horas == 23){
38                horas = 0;
39            }
40        }
41        horas = horas + 1;
42    }
43    else{
44        minutos = minutos + 1;
45    }
46}
47
48else{
49    segundos = segundos + 1;
50}
51}
52
53//función para la interrupción del PIO donde se está ingresando
54//los dos botones activos en bajo para configurar las horas y los minutos
55void handle_setclk_interrupts(void* context, alt_u32 id){
56    volatile int* edge_capture_ptr = (volatile int*) context;
57    *edge_capture_ptr = IORD ALTERA AVALON PIO_EDGE_CAP(0x3020);
58    IOWR ALTERA AVALON PIO_EDGE_CAP(0x3020, 0x00);
59    if((IORD ALTERA AVALON PIO_DATA(0x3020) & 0x01) == 0){
60        if(horas == 23){
61            horas = 0;
62        }
63    }
64    else{
65        horas = horas + 1;
66    }
67}
68
69if(((IORD ALTERA AVALON PIO_DATA(0x3020) & 0x02) >> 1) == 0){
70    minutos = 0;
71}
72
73else{
74    minutos = minutos + 1;
75}
76
77}
78
79//función para inicializar la interrupción del PIO de 1Hz
80void init_pio_lhz(void){
81    void* edge_capture_ptr = (void*) &edge_capture;
82    IOWR ALTERA AVALON PIO_IRQ_MASK(0x3010, 0x01);
83    IOWR ALTERA AVALON PIO_EDGE_CAP(0x3010, 0x00);
84    alt_irq_register(-1, edge_capture_ptr,
85                      handle_lhz_interrupts );
86}
87
88
89//función para inicializar la interrupción del PIO de los botones
90void init_pio_setclk(void){
91    void* edge_capture_ptr = (void*) &edge_capture;
92    IOWR ALTERA AVALON PIO_IRQ_MASK(0x3020, 0x03);
93    IOWR ALTERA AVALON PIO_EDGE_CAP(0x3020, 0x00);
94    alt_irq_register(-2, edge_capture_ptr,
95                      handle_setclk_interrupts );
96}
97
98
99//función para obtener los dígitos decena y unidad de una variable
100void convierte(unsigned char numero){
101    decena = numero / 10;
102    unidad = numero % 10;
103}

```

59

Ejemplo 5: Reloj en formato 24h con visualización hh:mm:ss empleando un FREQ_DIV 50MHz-1Hz

- Código completo contemplando interrupciones

```

105//rutina de inicialización del LCD
106void LCD_init(void){
107    usleep(15000);
108    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x38);
109    usleep(4100);
110    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x38);
111    usleep(100);
112    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x38);
113    usleep(5000);
114    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x38);
115    usleep(100);
116    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x08);
117    usleep(100);
118    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x0C);
119    usleep(100);
120    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x06);
121    usleep(100);
122    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x02);
123    usleep(2000);
124    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x01);
125    usleep(2000);
126}
127
128//función para escribir una cadena de caracteres en el LCD
129void LCD_escribenensaje(const char *cadena){
130    unsigned char i=0;
131    unsigned char tam;
132    tam = strlen(cadena);
133    for(i=0;i<tam;i++){
134        IOWR(LCD_O_BASE, LCD_WR_DATA_REG, cadena[i]);
135        usleep(100);
136    }
137}
138
139//función para enviar un carácter al LCD
140void LCD_enviarchar(unsigned char carácter){
141    IOWR(LCD_O_BASE, LCD_WR_DATA_REG, carácter);
142    usleep(100);
143}
144
145//función para ir a la primera linea del LCD
146void LCD_primerolinea(void){
147    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0x02);
148    usleep(2000);
149}
150
151//función para ir a la segunda linea del LCD
152void LCD_segundolinea(void){
153    IOWR(LCD_O_BASE, LCD_WR_COMMAND_REG, 0xC0);
154    usleep(100);
155}
156
157//rutina principal, encargado de las configuraciones iniciales y
158//de la visualización de la hora en el LCD
159int main()
160{
161    LCD_init();
162    init_pio_lhz();
163    init_pio_setclk();
164    alt_putstr("Hello MPCino, soy el NIOS II\n");
165    alt_putstr("Hello MPCino, soy el NIOS II\n");
166    alt_putstr("Hecho por Kalun Lau\n");
167    alt_putstr("Prueba Semana 13\n");
168    alt_putstr("Aplicación de reloj v10.0 release\n");
169    LCD_primerolinea();
170    LCD.escribenensaje("Hecho por Kalun");
171    usleep(2000);
172    sleep(2000);
173    usleep(2000);
174    while (1){
175        LCD.segundolinea();
176        LCD.escribenensaje("Reloj: ");
177        convierte(horas);
178        LCD.enviarchar(decena+48);
179        LCD.enviarchar(unidad+48);
180        LCD.enviarchar('.');
181        convierte(minutos);
182        LCD.enviarchar(decena+48);
183        LCD.enviarchar(unidad+48);
184        LCD.enviarchar('.');
185        convierte(segundos);
186        LCD.enviarchar(decena+48);
187        LCD.enviarchar(unidad+48);
188    }
189}

```

60