

Sistemas Digitales

Semana 13 Laboratorio

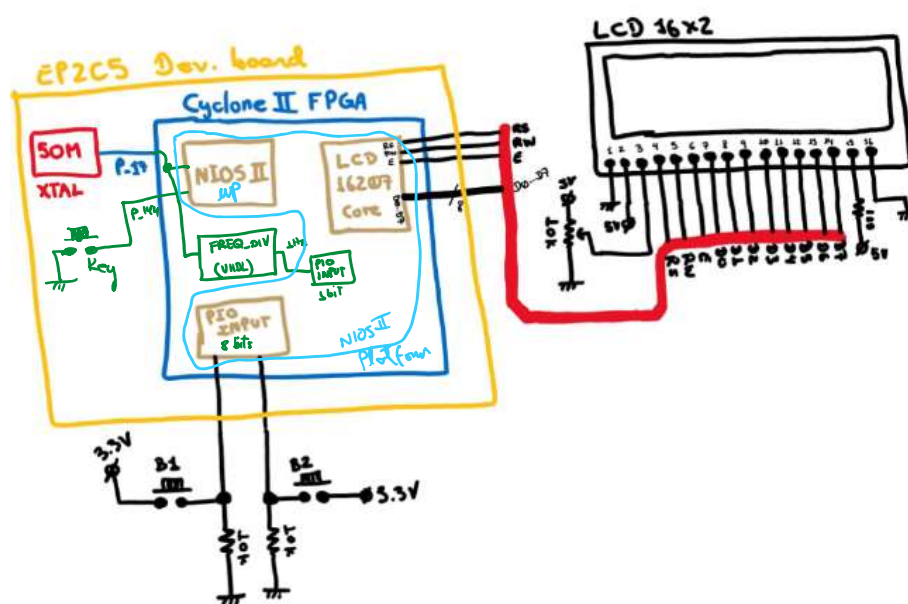
Por Kalun Lau

1

Preguntas previas

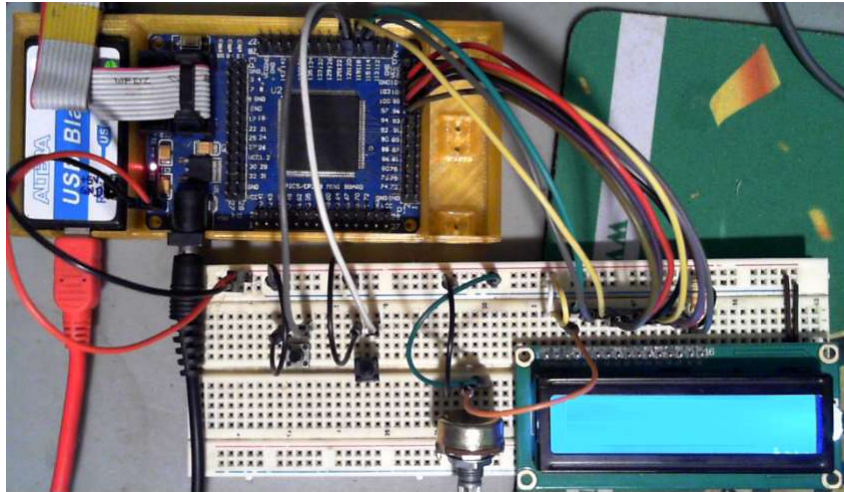
2

4



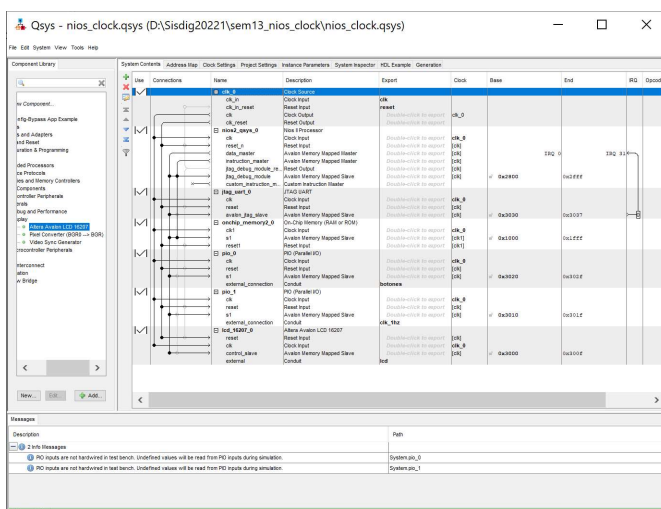
Circuito implementado

- La asignación de pines esta a criterio del desarrollador



5

Configuración de la Plataforma NIOS II en Qsys



- Tener en cuenta los mensajes de la parte información, no debe de haber error o advertencia alguna

6

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 description 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 if;
27     end process;
28     out_clk <= interno;
29 end description;

```

7

Asignación de pines en PinPlanner

- Tener en cuenta el tipo de pin del FPGA antes de asignar.

Top View Wire Bond

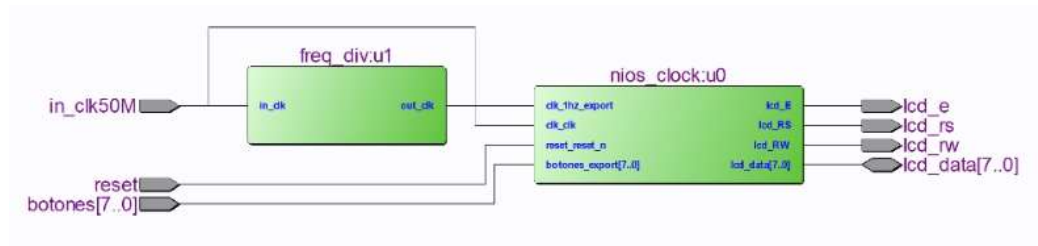
Cyclone II EP2C5T144C8

| Node Name | Direction | Location | I/O Bank | Filter Location | I/O Standard | Current Strength | Weak Pull-Up Resistor |
|-------------|-----------|----------|----------|-----------------|------------------------|------------------|-----------------------|
| botones[7] | Input | PIN_119 | 2 | PIN_119 | 3.3-V LVTTTL (default) | 24mA (default) | |
| botones[6] | Input | PIN_65 | 3 | PIN_65 | 3.3-V LVTTTL (default) | 24mA (default) | |
| botones[5] | Input | PIN_86 | 3 | PIN_86 | 3.3-V LVTTTL (default) | 24mA (default) | |
| botones[4] | Input | PIN_88 | 3 | PIN_88 | 3.3-V LVTTTL (default) | 24mA (default) | |
| botones[3] | Input | PIN_118 | 3 | PIN_118 | 3.3-V LVTTTL (default) | 24mA (default) | |
| botones[2] | Input | PIN_122 | 3 | PIN_122 | 3.3-V LVTTTL (default) | 24mA (default) | |
| botones[1] | Input | PIN_120 | 3 | PIN_120 | 3.3-V LVTTTL (default) | 24mA (default) | |
| botones[0] | Input | PIN_121 | 2 | PIN_121 | 3.3-V LVTTTL (default) | 24mA (default) | On |
| in_clkSOM | Input | PIN_17 | 1 | PIN_17 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[7] | Bidir | PIN_104 | 3 | PIN_104 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[6] | Bidir | PIN_103 | 3 | PIN_103 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[5] | Bidir | PIN_101 | 3 | PIN_101 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[4] | Bidir | PIN_100 | 3 | PIN_100 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[3] | Bidir | PIN_99 | 3 | PIN_99 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[2] | Bidir | PIN_97 | 3 | PIN_97 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[1] | Bidir | PIN_96 | 3 | PIN_96 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_data[0] | Bidir | PIN_94 | 3 | PIN_94 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_e | Output | PIN_92 | 3 | PIN_92 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_rs | Output | PIN_81 | 3 | PIN_81 | 3.3-V LVTTTL (default) | 24mA (default) | |
| lcd_rw | Output | PIN_87 | 3 | PIN_87 | 3.3-V LVTTTL (default) | 24mA (default) | |
| reset | Input | PIN_144 | 2 | PIN_144 | 3.3-V LVTTTL (default) | 24mA (default) | On |

8

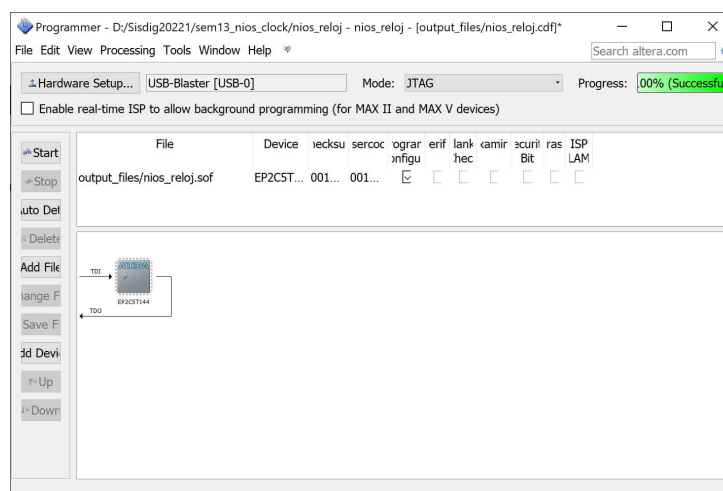
Vista del RTL

- Se muestra los componentes internos del diseño de la aplicación



9

Etapa de grabación de la plataforma NIOS II en la tarjeta de FPGA



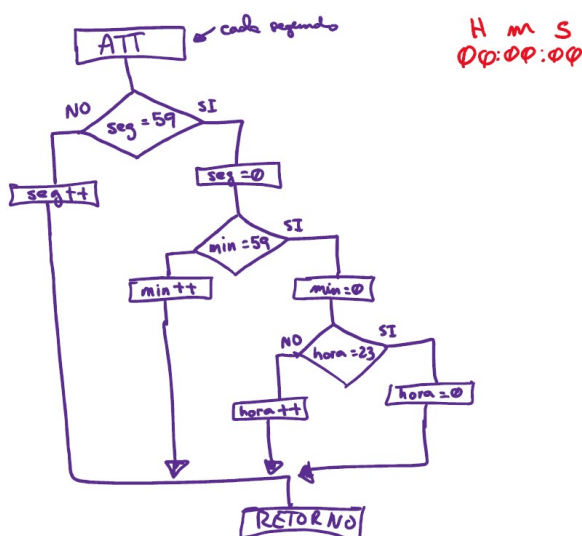
10

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

11

Algoritmo en diagrama de flujo y pseudocódigo del sistema de cuenta de un reloj de 24 horas



```

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++;
    }
}
  
```

12

Código propuesto en Eclipse

• Pendientes:

- Uso de interrupciones para la entrada de reloj de 1Hz y botones de ajuste para la hora actual.
- Proceso de ajuste para la hora actual.

```
#include <sys/alt_stdio.h>
#include <system.h>
#include <string.h>
#include <unistd.h>
#include "altera_avalon_pio_regs.h"
#include "altera_avalon_lcd_16207_regs.h"

#define LCD_WR_COMMAND_REG 0
#define LCD_RD_STATUS_REG 1
#define LCD_WR_DATA_REG 2
#define LCD_RD_DATA_REG 3
#define LCD_0_BASE 0x3000

//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, 0x38);
    usleep(4100);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
    usleep(5000);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
    usleep(100);
    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);
}

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);
}

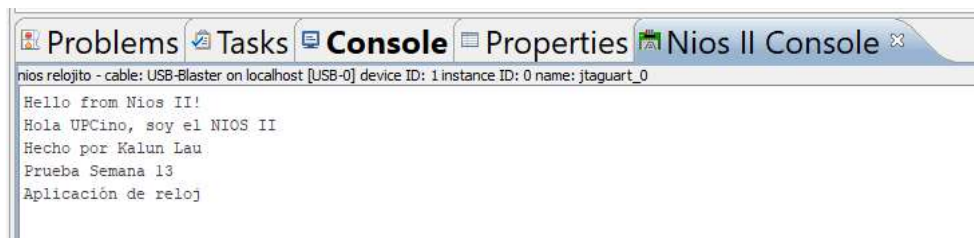
void SEG_LINEA(void){
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xc0);
    usleep(100);
}

int main()
{
    lcd_init();
    alt_putstr("Hello from Nios II!\n");
    alt_putstr("Hola UPCino, soy el NIOS II\n");
    alt_putstr("Hecho por Kalun Lau\n");
    alt_putstr("Prueba Semana 13\n");
    alt_putstr("Aplicación de reloj\n");
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);
    ESCRIBE_MENSAJE("Hecho por Kalun");
    usleep(2000);
    usleep(2000);
    while (1){
        SEG_LINEA();
        ESCRIBE_MENSAJE("Reloj: ");
        convierte(horas);
        ENVIA_CHAR(decena+48);
        ENVIA_CHAR(unidad+48);
        ENVIA_CHAR(':');
        convierte(minutos);
        ENVIA_CHAR(decena+48);
        ENVIA_CHAR(unidad+48);
        ENVIA_CHAR(':');
        convierte(segundos);
        ENVIA_CHAR(decena+48);
        ENVIA_CHAR(unidad+48);

        if(((IORD ALTERA_AVALON_PIO_DATA(0x3010) & 0x01) == 1) && indicador == 1){
            if(segundos == 59){
                segundos = 0;
                if(minutos == 59){
                    minutos = 0;
                    if(horas == 23){
                        horas = 0;
                    }
                    else{
                        horas = horas + 1;
                    }
                }
                else{
                    minutos = minutos + 1;
                }
            }
            else{
                segundos = segundos + 1;
            }
            indicador = 0;
        }
        if((IORD ALTERA_AVALON_PIO_DATA(0x3010) & 0x01) == 0){
            indicador = 1;
        }
    }
}
```

13

Mensajes en la consola del NIOS II



14

Pruebas de implementación

