

Sistemas Digitales

Semana 13 – Desarrollo de aplicaciones con el NIOS II
Semestre 2020-2
Profesor Kalun José Lau Gan

1

Preguntas previas:

2

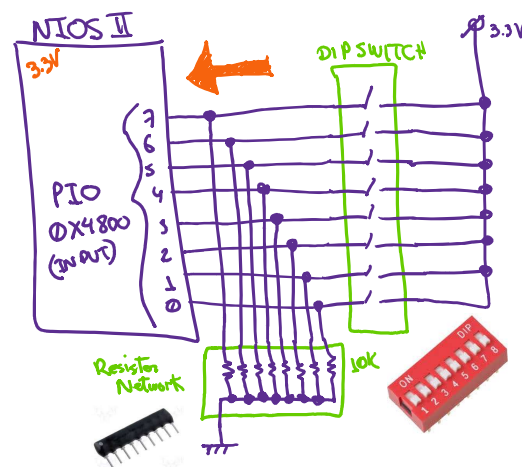
Agenda:

- Lectura de un PIO en el NIOS II
- Ejemplos de aplicaciones relacionadas con lectura de puertos de entrada en el NIOS II
- Interface del NIOS II con un display LCD alfanumérico 44780 20x2

3

Lectura de un PIO en el NIOS II

- Ejemplo: Conectar un DIP switch en un PIO 0x4800



4

Lectura de un PIO en el NIOS II

- Ejemplo: Lectura del PIO en 0x4800 en código C del NIOSII

```
#include "altera_avalon_pio_regs.h"

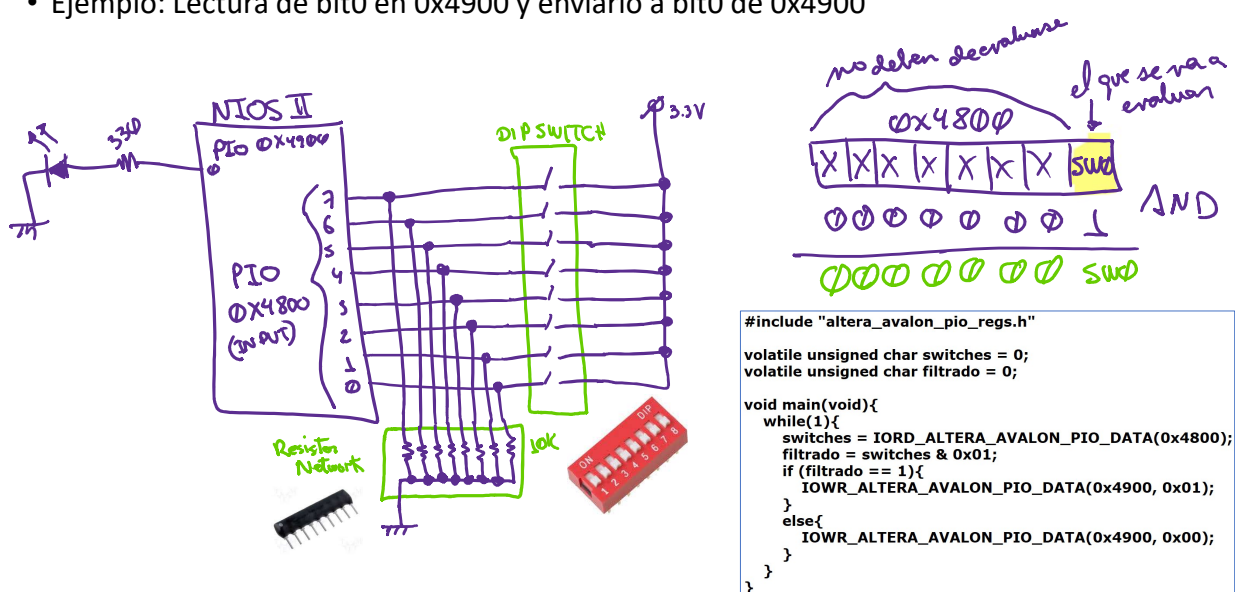
volatile unsigned char switches = 0;

void main(void){
    while(1){
        switches = IORD_ALTERA_AVALON_PIO_DATA(0x4800);
    }
}
```

5

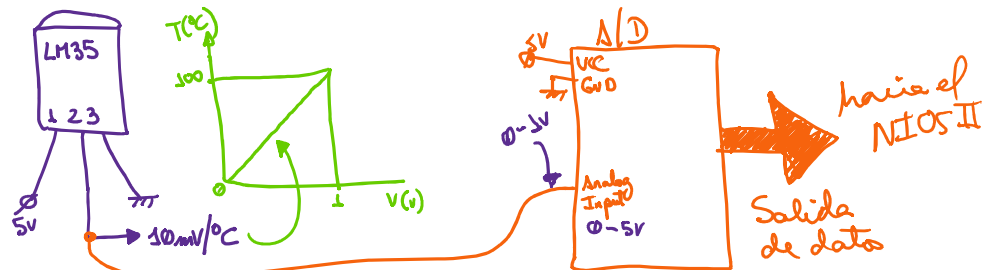
Lectura de un PIO en el NIOS II

- Ejemplo: Lectura de bit0 en 0x4900 y enviarlo a bit0 de 0x4900



6

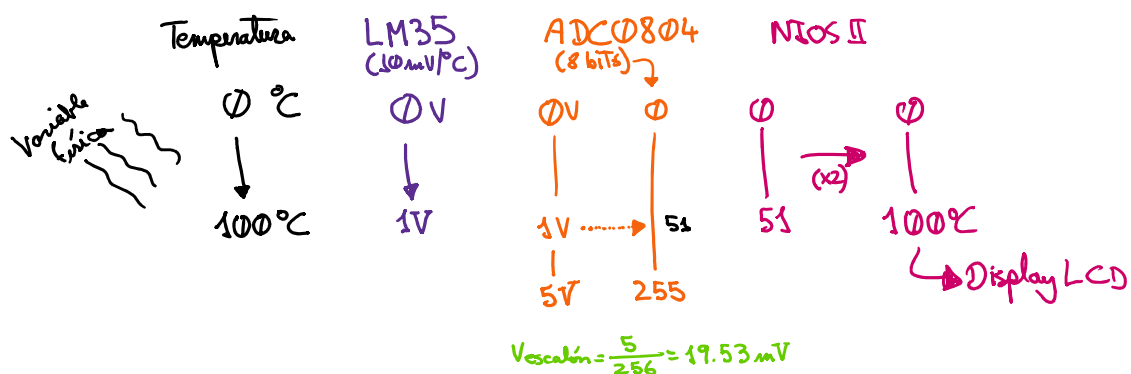
Adquisición de señales analógicas con el NIOS II



- Tenemos que atender esa señal analógica con un conversor A/D para que el NIOS II lo reciba en formato digital.
- Tener en cuenta que la alimentación del LM35 es de 5V (posiblemente será igual para como el A/D)
- La selección del A/D (tiempo de adquisición, resolución, forma de conversión) dependerá de la naturaleza de la señal a adquirir y de la calidad que se desea registrar.

7

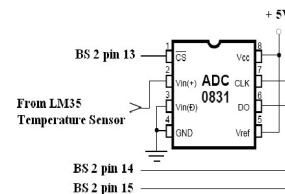
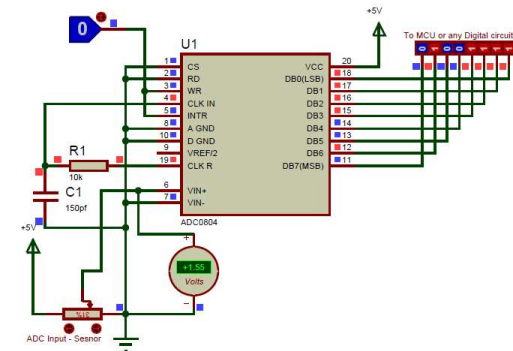
Escalamiento de medidas:



8

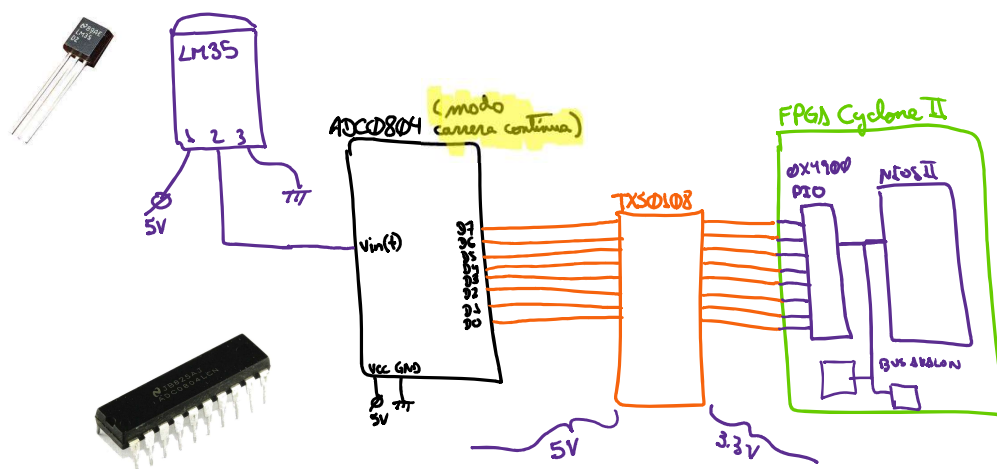
Conversores A/D disponibles localmente:

- ADC0804 (conversor A/D 8bits en paralelo un solo canal) ✓
- ADC0809 (conversor A/D 8bits en paralelo, ocho canales) ✗
- ADC0831 (conversor A/D 8bits con interface serial, un canal) ✓



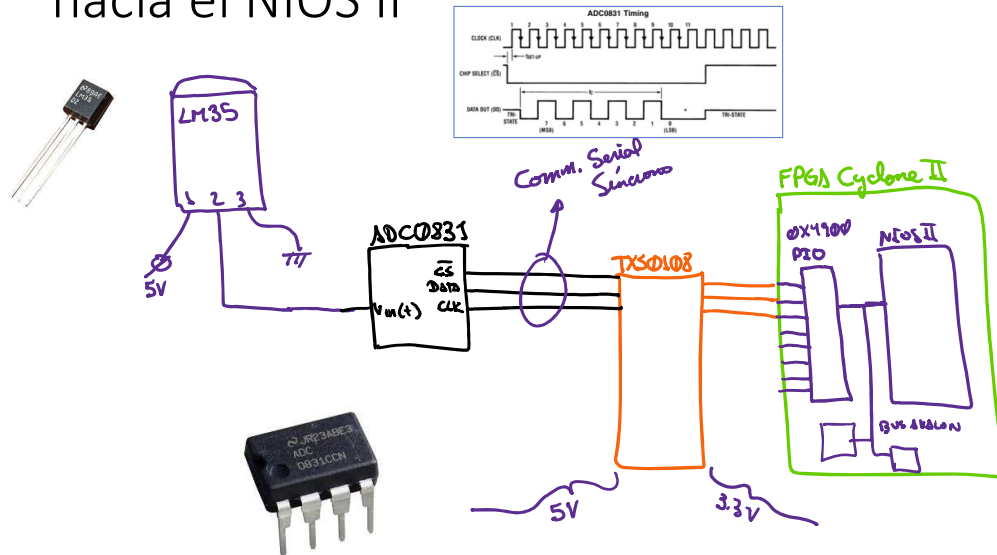
9

Circuito propuesto para conectar un LM35 hacia el NIOS II



10

Circuito propuesto para conectar un LM35 hacia el NIOS II



11

Función para obtener los dígitos de una variable

- Sirve para poder enviar dígito por dígito hacia consola o hacia un display

```
volatile unsigned char dato = 0;

volatile unsigned int millar = 0;
volatile unsigned int centena = 0;
volatile unsigned int decena = 0;
volatile unsigned int unidad = 0;

void convierte(unsigned int numero){
    millar = numero / 1000;
    centena = (numero % 1000) / 100;
    decena = (numero % 100) / 10;
    unidad = numero % 10;
}
```

12

Código ejemplo en C para NIOS II

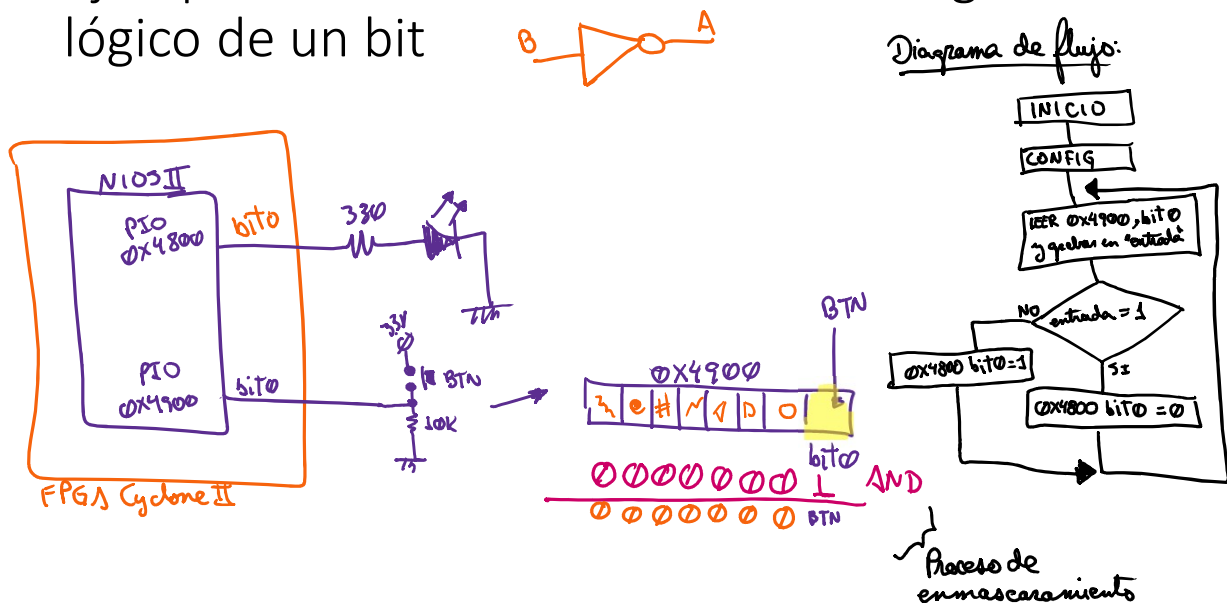
```
volatile unsigned char lectura = 0;
volatile unsigned int centena = 0;
volatile unsigned int decena = 0;
volatile unsigned int unidad = 0;

void convierte(unsigned char numero){
    centena = numero / 100;
    decena = (numero % 100) / 10;
    unidad = numero % 10;
}

void main(void){
    while(1){
        lectura = IORD_ALTERA_AVALON_PIO_DATA(0x4900); //valores entre 0 y 51
        convierte(lectura * 2); //valores entre 0 y 102
        alt_putchar(centena+48);
        alt_putchar(decena+48);
        alt_putchar(unidad+48);
    }
}
```

13

Ejemplo de circuito: Desarrollar un negador lógico de un bit



14

Código en C para NIOS II:

```
#include "altera_avalon_pio_regs.h"

volatile unsigned char switches = 0;
volatile unsigned char filtrado = 0;

void main(void){
    while(1){
        switches = IORD_ALTERA_AVALON_PIO_DATA(0x4900);
        filtrado = switches & 0x01;
        if (filtrado == 1){
            IOWR_ALTERA_AVALON_PIO_DATA(0x4800, 0x00);
        }
        else{
            IOWR_ALTERA_AVALON_PIO_DATA(0x4800, 0x01);
        }
    }
}
```

15

Manejo del LCD en el NIOS II

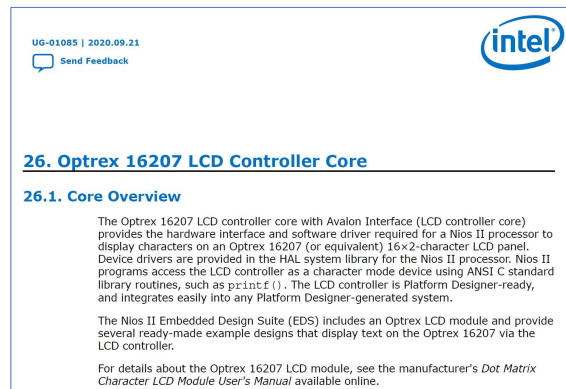


- Manejado por un controlador Hitachi HD44780:
 - ROM de caracteres integrado y similar a ASCII en los primeros 7 bits
 - Posee CGRAM para caracteres personalizados (8 máximo)
- Interface de 8 bits (o 4bits) en paralelo (revisar hoja técnica)
- Diversos tamaños desde 1x8 hasta 4x40

16

Uso de la interface integrada en NIOS II para manejo del LCD

- Revisar documentación de periféricos embebidos para NIOS II (ug_embedded_ip.pdf)



19

Fin de la sesión

20