

Sistemas Digitales

Semana 12: Algoritmos y Diagramas de Flujo

Profesor: Kalun José Lau Gan

1

Preguntas previas:

- Me sale error de que no reconoce la librería “altera_avalon_lcd_16207_reg.h” en el Eclipse
 - Ha reportado eso debido a que el sopinfo generado por el Qsys no contempla el uso del LCD, se debe de revisar en Qsys si es que se ha llamado al dicho modulo ó revisar que el sopinfo escogido en el Eclipse sea el CORRECTO (revisar fecha y hora de generación del archivo sopinfo).
 - Cabe resaltar que si no se agregó el módulo LCD en el Qsys y se lo agregó en un tiempo después, deberás de realizar todo el procedimiento de implementación de la plataforma NIOS II en el Quartus.
- Al momento de estar implementando salió chispas y humo. ¿Qué puede haber pasado?
 - Cortocircuito, debes de implementar el circuito sin conexión energética.

2

Preguntas previas:

- Luego de haber realizado todo el procedimiento hasta la aplicación del Eclipse, tuve que modificar la asignación de pines en el Pin Planner, que es lo que debo de hacer?
 - En este caso luego de modificar la asignación en el PinPlanner debes de volver a compilar el proyecto en Quartus, revisar si se actualizó la nueva asignación y grabar la plataforma en el FPGA. Luego volver a Eclipse y grabar la aplicación de la plataforma.
- Me quedé sin espacio de memoria en el NIOS II. ¿Qué debo de hacer?
 1. Regresar a Qsys y ampliar la memoria, luego grabar el qsys, luego Generar la plataforma.
 2. Regresar al Quartus y volver a compilar el proyecto, luego grabar la plataforma en el FPGA
 3. Retornar a Eclipse y actualizar (generar) el BSP, volver a compilar el código y grabar la aplicación de la plataforma.

3

Preguntas previas:

- La librería proporcionada para el LCD así como los ejemplos vistos pueden ser usados para diferentes tamaños de LCD?
 - Si, teniendo en cuenta que el controlador integrado en el LCD sea el HD44780: displays del tipo alfanuméricos desde 1x8 hasta 4x40. Una identificación rápida es los pines de interface (14 pines sin backlight, 16 pines con backlight)



4

Preguntas previas:

- ¿Se pueden usar displays para visualizar imágenes con el NIOSII?
 - Si, hay bastante diversidad de displays gráficos, solo hay que tener en cuenta el controlador que esta montado en él:
 - Samsung KS0108B
 - Toshiba T6963C
 - Epson SED1330
 - SSD1306 (OLED)
 - SH1106 (OLED)
 - Los hay de diferentes tamaños y resoluciones:
 - 128x64
 - 256x128
 - 256x64
- Tipos de display:
 - Transflective LCD (no backlight)
 - LCD
 - VFD
 - OLED



5

Preguntas previas:

- Tengo una fuente de 5.33V. ¿Puedo emplearlo para alimentar el LCD?
 - Si, siempre que no supere la barrera de los 5.5VDC.
- Hice un programa para display de cátodo común pero tengo el de ánodo común implementado.
 - Le aplicas el complemento antes de enviar al puerto de salida

```

unsigned char cadena[] = {0x5E, 0x3A, 0x66, 0x05};    //Para cátodo común
unsigned char invertido = 0;
void main(void){
    unsigned char x = 0;
    for(x=0;x<4;x++){
        invertido = !cadena[x];
        IOWR_ALTERA_AVALON_PIO_DATA(0x5300, invertido);
        usleep(500000);
    }
}

```

```
IOWR_ALTERA_AVALON_PIO_DATA(0x5300, !(cadena[x]));
```

6

Agenda:

- Algoritmos
- Diagramas de flujo
- Manipulación de I/O en el NIOS II (PIO)
- Ejemplos

7

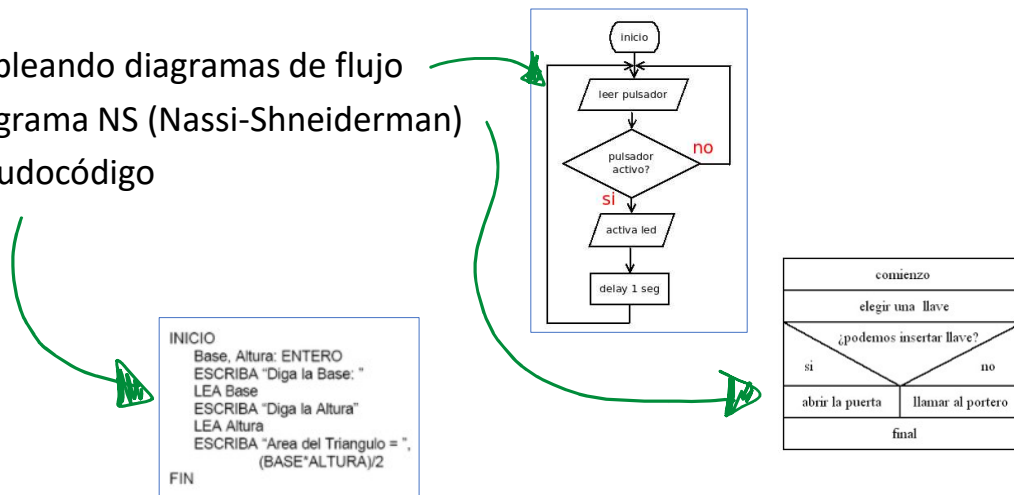
Algoritmos

- ¿Qué es un algoritmo?
 - Secuencia ordenada de instrucciones (o pasos) para resolver un determinado problema o el uso correcto de un equipo.
- Ejemplos:
 - Programa en computadora
 - Manual de instrucciones (de operación, de uso)
 - Hoja de ruta
 - Procedimiento para dar con la solución de un problema (troubleshooting)
 - Receta de cocina
 - Pasos para resolver un problema matemático, ingenieril, etc

8

¿Cómo represento un algoritmo informático?

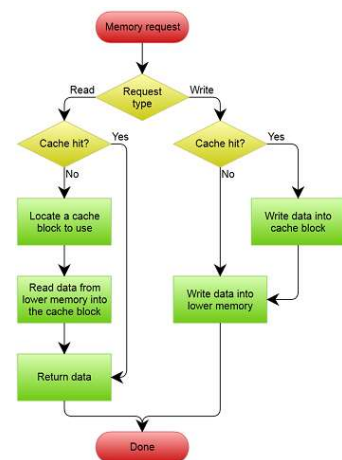
- Empleando diagramas de flujo
- Diagrama NS (Nassi-Shneiderman)
- Pseudocódigo



9

Diagramas de flujo (flowchart)

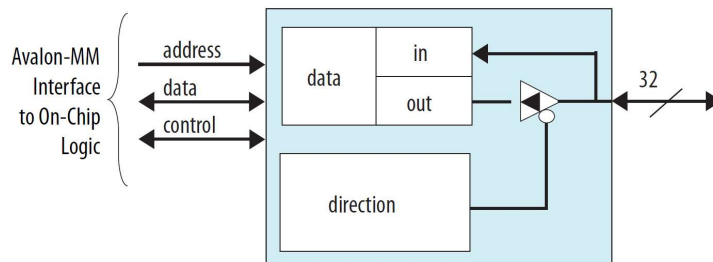
- ¿Quién inventó el diagrama de flujo?
 - Frank Gilbreth (1921)
- Características:
 - Emplea formas geométricas para la representación de un algoritmo
 - Usualmente en formato vertical (de arriba hacia abajo)
 - Empleado en research papers (artículos científicos).



10

Manipulación de puertos en el NIOS II

- Referencia: Manual de IPs de NIOS II, sección "PIO".



- Revisar librería `altera_avalon_pio_regs.h`

11

Registros internos del PIO

Table 273. Register Map for the PIO Core

Offset	Register Name		R/W	(n-1)	...	2	1	0
0	data	read access	R	Data value currently on PIO inputs				
		write access	W	New value to drive on PIO outputs				
1	direction (1)		R/W	Individual direction control for each I/O port. A value of 0 sets the direction to input; 1 sets the direction to output.				
2	interruptmask (1)		R/W	IRQ enable/disable for each input port. Setting a bit to 1 enables interrupts for the corresponding port.				
3	edgecapture (1) , (2)		R/W	Edge detection for each input port.				
4	outset		W	Specifies which bit of the output port to set. Outset value is not stored into a physical register in the IP core. Hence it's value is not reserve for future use.				
5	outclear		W	Specifies which output bit to clear. Outclear value is not stored into a physical register in the IP core. Hence it's value is not reserve for future use.				

Note :

- This register may not exist, depending on the hardware configuration. If a register is not present, reading the register returns an undefined value, and writing the register has no effect.
- If the option **Enable bit-clearing for edge capture register** is turned off, writing any value to the edgecapture register clears all bits in the register. Otherwise, writing a 1 to a particular bit in the register clears only that bit.

12

Librería *altera_avalon_pio_regs.h*

```

11 #ifndef __ALTERA_AVALON_PIO_REGS_H_
12 #define __ALTERA_AVALON_PIO_REGS_H_
13
14 #include <io.h>
15
16 #define IOADDR_ALTERA_AVALON_PIO_DATA(base)      __IO_CALC_ADDRESS_NATIVE(base, 0)
17 #define IORD_ALTERA_AVALON_PIO_DATA(base)        IORD(base, 0)
18 #define IOWR_ALTERA_AVALON_PIO_DATA(base, data)  IOWR(base, 0, data)
19
20 #define IOADDR_ALTERA_AVALON_PIO_DIRECTION(base) __IO_CALC_ADDRESS_NATIVE(base, 1)
21 #define IORD_ALTERA_AVALON_PIO_DIRECTION(base)   IORD(base, 1)
22 #define IOWR_ALTERA_AVALON_PIO_DIRECTION(base, data) IOWR(base, 1, data)
23
24 #define IOADDR_ALTERA_AVALON_PIO_IRQ_MASK(base) __IO_CALC_ADDRESS_NATIVE(base, 2)
25 #define IORD_ALTERA_AVALON_PIO_IRQ_MASK(base)    IORD(base, 2)
26 #define IOWR_ALTERA_AVALON_PIO_IRQ_MASK(base, data) IOWR(base, 2, data)
27
28 #define IOADDR_ALTERA_AVALON_PIO_EDGE_CAP(base) __IO_CALC_ADDRESS_NATIVE(base, 3)
29 #define IORD_ALTERA_AVALON_PIO_EDGE_CAP(base)    IORD(base, 3)
30 #define IOWR_ALTERA_AVALON_PIO_EDGE_CAP(base, data) IOWR(base, 3, data)
31
32 #define IOADDR_ALTERA_AVALON_PIO_SET_BIT(base)    __IO_CALC_ADDRESS_NATIVE(base, 4)
33 #define IORD_ALTERA_AVALON_PIO_SET_BITS(base)     IORD(base, 4)
34 #define IOWR_ALTERA_AVALON_PIO_SET_BITS(base, data) IOWR(base, 4, data)
35
36 #define IOADDR_ALTERA_AVALON_PIO_CLEAR_BITS(base) __IO_CALC_ADDRESS_NATIVE(base, 5)
37 #define IORD_ALTERA_AVALON_PIO_CLEAR_BITS(base)   IORD(base, 5)
38 #define IOWR_ALTERA_AVALON_PIO_CLEAR_BITS(base, data) IOWR(base, 5, data)
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63 /* Definitions for direction-register operation with bi-directional PIOs */
64 #define ALTERA_AVALON_PIO_DIRECTION_INPUT 0
65 #define ALTERA_AVALON_PIO_DIRECTION_OUTPUT 1
66
67 #endif /* __ALTERA_AVALON_PIO_REGS_H_ */

```

13

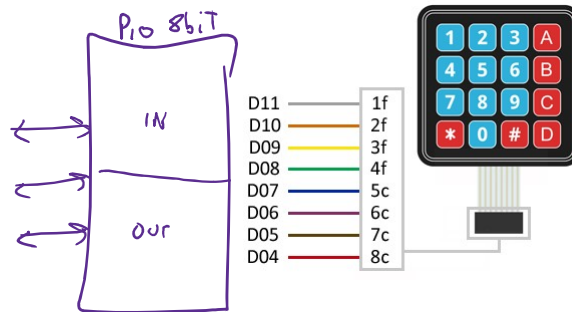
Registros de control para PIO

- Para establecer si un puerto PIO bidireccional de 8 bits (establecido en Qsys) sea salida:
 - Si tengo un PIO 0x6000 bidir y quiero que todos sus bits sean salidas:
 - IOWR_ALTERA_AVALON_PIO_DIRECTION(0x6000, 0xFF);
 - Si tengo un PIO 0x6000 bidir y quiero que todos sus bits sea entradas:
 - IOWR_ALTERA_AVALON_PIO_DIRECTION(0x6000, 0x00);

14

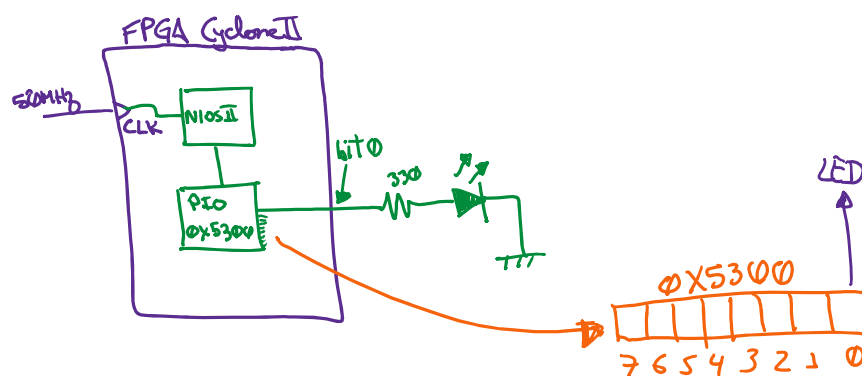
Caso: teclado matricial

- Hay que hacer un proceso de escaneo de teclas



15

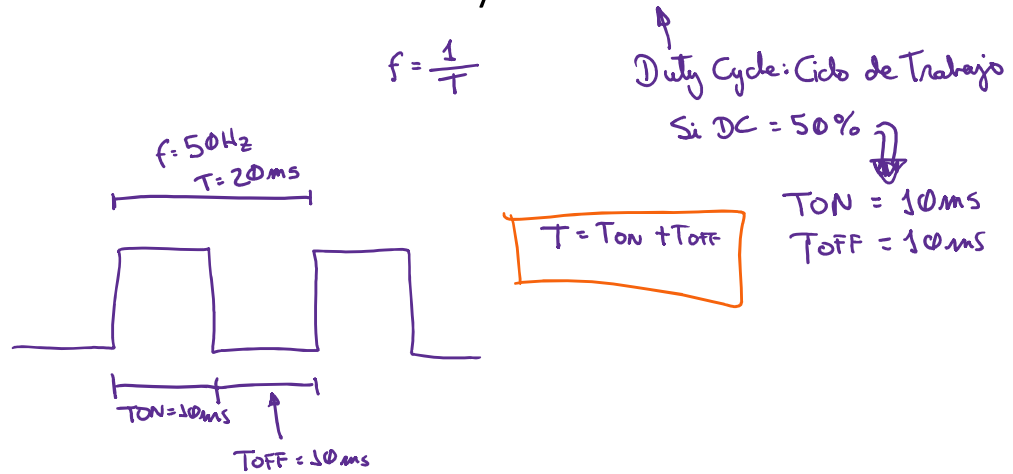
Ejemplo 1: Generar una señal cuadrada de 50Hz con 50% DC a través del bit0 del PIO 0x5300



$$T = \frac{1}{f}$$

16

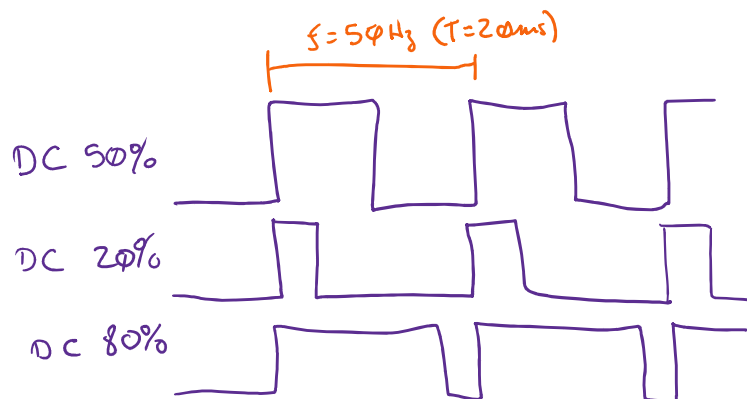
¿Onda cuadrada de 50Hz y 50% DC?



17

¿PWM (Pulse-width Modulation)?

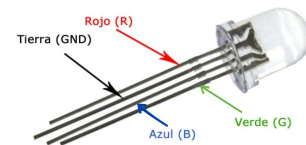
- Modulación por ancho de pulso
- Frecuencia el fija, ancho de pulso variable (duty cycle)



18

Usos del PWM

- Control de velocidad de motores
- Posición angular en servomecanismos
- Intensidad de la luz de fondo en pantallas
- Control de contraste en pantallas
- Control de color en LEDs RGB
- Controlar la intensidad luminosa de un LED
- Control de temperatura para incubadoras, peceras, impresoras 3D, deshumidificadores



19

Ejemplo de PWM en Eclipse

- Programa para emitir un PWM de 50Hz y 50% DC

```
//Establezco base de tiempo de 1us trabajando con usleep()
//DC50%
void main(void){
    while(1){
        IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);
        usleep(10000);
        IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);
        usleep(10000);
    }
}
```

20

Ejemplo de PWM en Eclipse

- Programa para emitir un PWM de 50Hz y 25% DC

```
//DC25%  
void main(void){  
    while(1){  
        IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);  
        usleep(5000);  
        IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);  
        usleep(15000);  
    }  
}
```

21

Ejemplo de PWM en Eclipse

- Programa para emitir un PWM de 50Hz y 75% DC

```
//DC75%  
void main(void){  
    while(1){  
        IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);  
        usleep(15000);  
        IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);  
        usleep(5000);  
    }  
}
```

22

Ejemplo de PWM en Eclipse

- Programa para emitir un PWM de 50Hz y según una entrada establecer el DC

```

unsigned int periodo = 20000;
unsigned int ton = 0;

void main(void){
    while(1){
        if((IORD_ALTERA_AVALON_PIO_DATA(0x5400 & 0x01) == 0x01){
            ton = 5000
            unsigned int x = 0;
            for(x = 0; x<ton; x++){
                IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);
                usleep(1);
            }
            for(x = 0; x<(periodo-ton); x++){
                IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);
                usleep(1);
            }
        }
        else{
            ton = 15000
            unsigned int x = 0;
            for(x = 0; x<ton; x++){
                IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);
                usleep(1);
            }
            for(x = 0; x<(periodo-ton); x++){
                IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);
                usleep(1);
            }
        }
    }
}

```

23

Ejemplo de PWM en Eclipse

- Programa para emitir un PWM de 50Hz y según dos entradas establecer el DC

```

unsigned int periodo = 20000;
unsigned int ton = 0;
unsigned char entrada = 0;

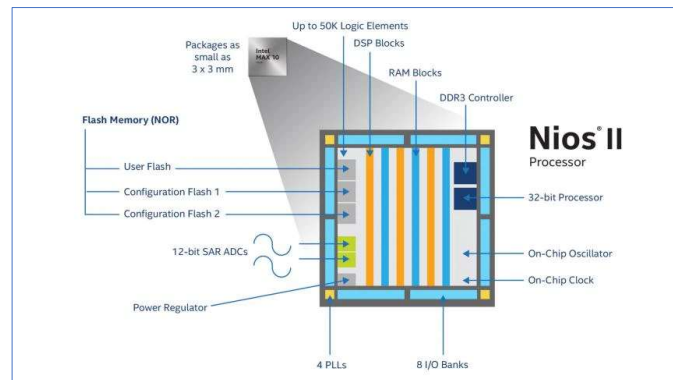
void main(void){
    while(1){
        entrada = IORD_ALTERA_AVALON_PIO_DATA(0x5400 & 0x03);
        switch(entrada){
            case 0:
                ton = 5000
                unsigned int x = 0;
                for(x = 0; x<ton; x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);
                    usleep(1);
                }
                for(x = 0; x<(periodo-ton); x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);
                    usleep(1);
                }
                break;
            case 1:
                ton = 10000
                unsigned int x = 0;
                for(x = 0; x<ton; x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);
                    usleep(1);
                }
                for(x = 0; x<(periodo-ton); x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);
                    usleep(1);
                }
                break;
            case 2:
                ton = 15000
                unsigned int x = 0;
                for(x = 0; x<ton; x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);
                    usleep(1);
                }
                for(x = 0; x<(periodo-ton); x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);
                    usleep(1);
                }
                break;
            case 3:
                ton = 1000
                unsigned int x = 0;
                for(x = 0; x<ton; x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x01);
                    usleep(1);
                }
                for(x = 0; x<(periodo-ton); x++){
                    IOWR_ALTERA_AVALON_PIO_DATA(0x5300, 0x00);
                    usleep(1);
                }
                break;
        }
    }
}

```

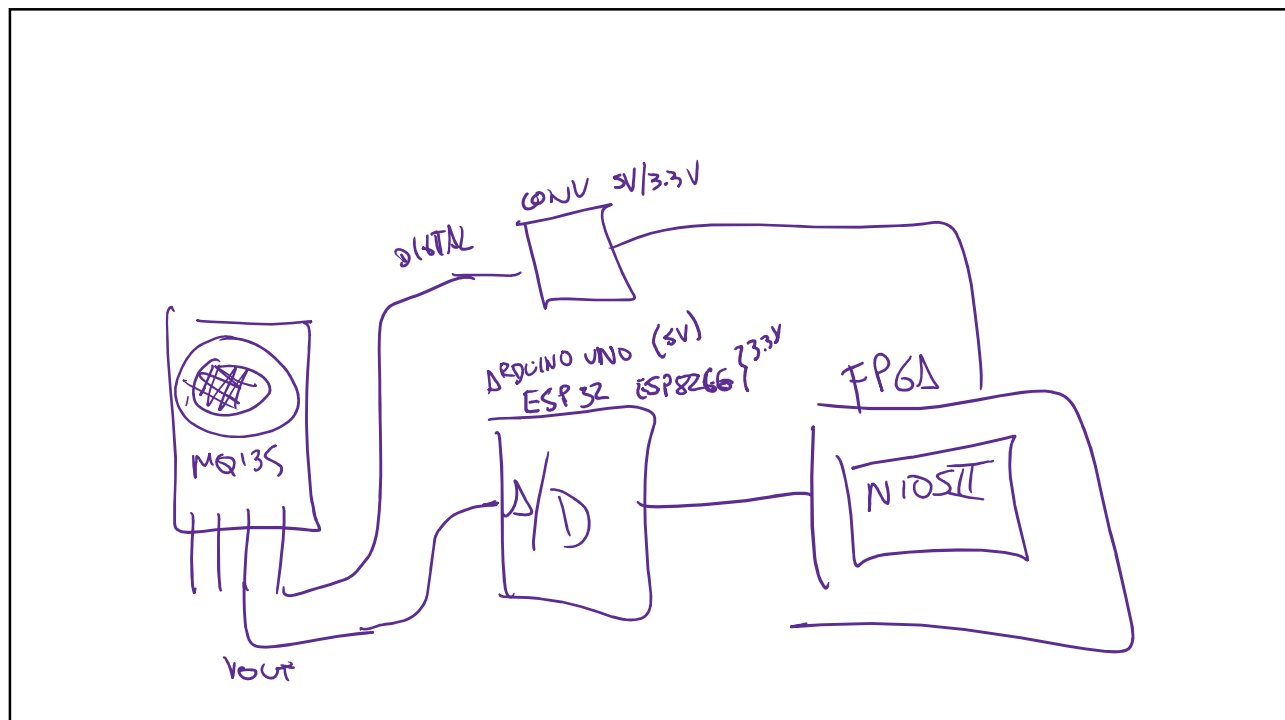
24

¿Y si requiero cambiar de manera continua los valores de DC en una señal PWM?

- Se necesitaría contar con un conversor analógico/digital para recibir un valor numérico dentro de un rango, el FPGA empleado no posee uno.
- El FPGA Intel MAX10 si lo posee:



25



26