

Manejo de puertos PIO y LCD en el NIOS II

Por: Kalun José Lau Gan

2021

1

Aspectos iniciales:

- Los PIO pueden configurarse de 1 bit a 32 bits de ancho.
- Pueden ser entradas, salidas o bidireccionales.
- Los registros implicados para el uso del PIO son de 32 bits mostrados a continuación:

Offset	Register Name		R/W	Fields				
				(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.				
5	outclear		W	Specifies which output bit to clear.				

Notes

(1) 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.

(2) Writing any value to edgecapture clears all bits to 0.

2

Aspectos iniciales:

- Los PIO se declaran en el QSys
 - De acuerdo a los requerimientos se considerará puertos de entrada o de salida (no es recomendable trabajar con bidireccionales en esta etapa inicial).
 - Tener en cuenta la dirección asignada a cada PIO y revisar que no haya conflicto.
 - Asignar un nombre a la conexión externa del puerto para que aparezcan en el PinPlanner



3

Aspectos iniciales:

- Librería a incluir en el programa:
 - #include "altera_avalon_pio_regs.h"
- Escritura de dato en el PIO:


```
IOWR_ALTERA_AVALON_PIO_DATA( [dirección del PIO], [dato]);
```

 - Ejemplo:


```
IOWR_ALTERA_AVALON_PIO_DATA(0x5100, 0x5A);
```
- Lectura de dato en el PIO:

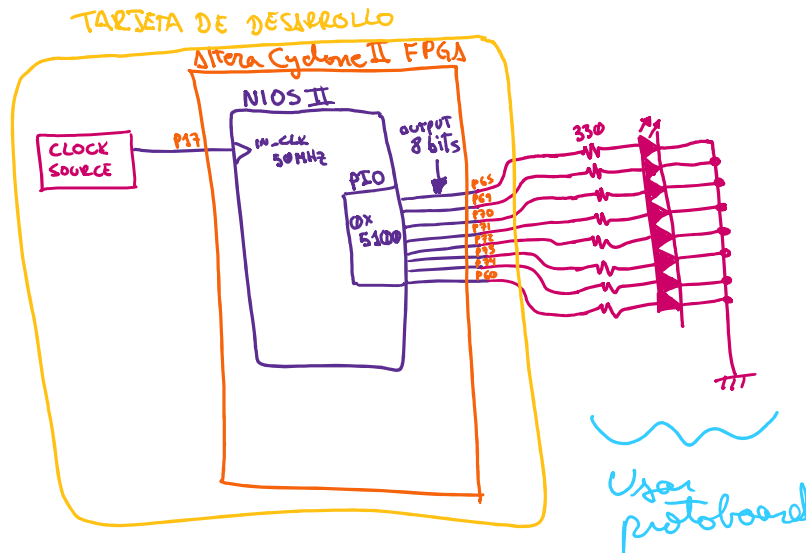

```
IORD_ALTERA_AVALON_PIO_DATA( [dirección del PIO] );
```

 - Ejemplo (se tendrá que declarar una variable donde se alojará lo leído):


```
data_in = IORD_ALTERA_AVALON_PIO_DATA(0x5200);
```

4

Ejemplo: PIO de salida de 8 bits en dirección 0x5100 conectado a ocho LEDs



5

Ejemplo: PIO de salida de 8 bits en dirección 0x5100

- El siguiente programa emite a través del PIO 0x5100 tres datos de manera secuencial y cíclica con un retardo definido por la subfunción delay_s
- Los datos son códigos en ASCII de la sigla UPC
- El comando alt_putstr se usa para enviar mensajes a la consola del NIOS II en el Eclipse

```
#include "sys/alt_stdio.h"
#include "system.h"
#include "altera_avalon_pio_regs.h"

void delay_s(int tiempo){
    volatile int interno = 0;
    while (interno < tiempo*10000){
        interno++;
    }
}

int main()
{
    alt_putstr("Hello from Nios II!\n");

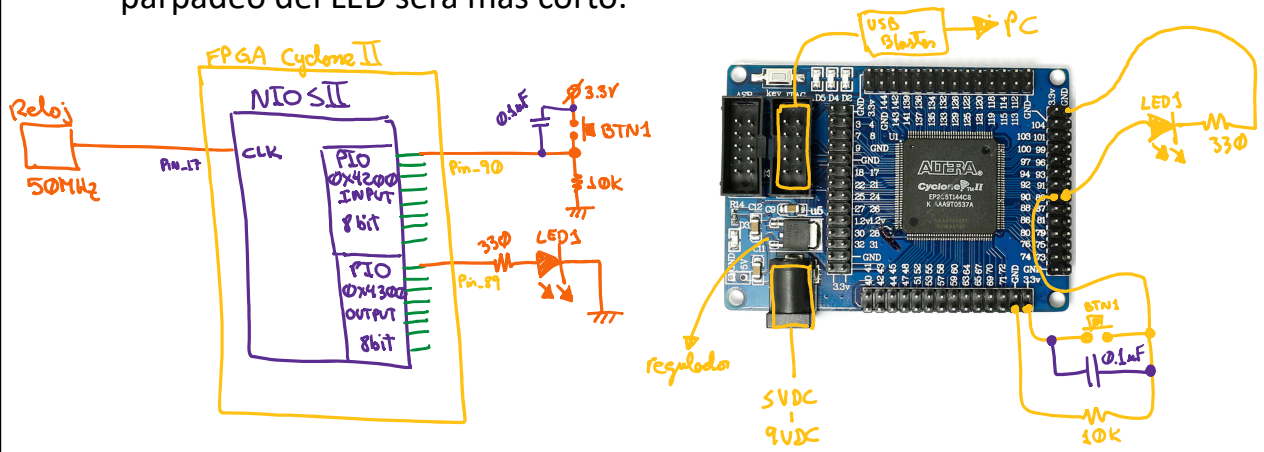
    /* Event loop never exits. */
    while (1){
        IOWR_ALTERA_AVALON_PIO_DATA(0x5100, 0x55); //Emite letra U
        delay_s(30);
        IOWR_ALTERA_AVALON_PIO_DATA(0x5100, 0x50); //Emite letra P
        delay_s(30);
        IOWR_ALTERA_AVALON_PIO_DATA(0x5100, 0x43); //Emite letra C
        delay_s(30);
    }

    return 0;
}
```

6

Ejemplo: Lectura de PIO entradas

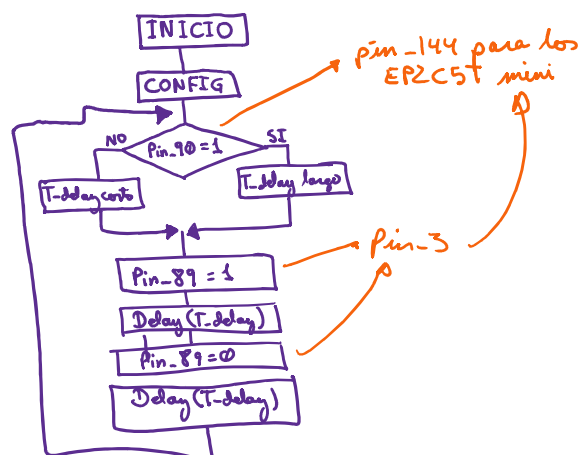
- En el siguiente circuito, cuando se presione el botón el tiempo de parpadeo del LED será mas corto.



7

Ejemplo: Lectura de PIO entradas

- Algoritmo en diagrama de flujo



8

Ejemplo: Lectura de PIO entradas

- Código en C para NIOS II

```
#include "sys/alt_stdio.h"
#include "altera_avalon_pio_regs.h"
#include "system.h"

volatile unsigned char dato_in = 0;
volatile long tiempo = 0;

void main(void)
{
    alt_putstr("Hello from Nios II!\n");
    while(1){
        dato_in = (IORD_ALTERA_AVALON_PIO_DATA(0x4200)) & 0x01;
        if(dato_in == 1){
            tiempo = 200000;
        }
        else{
            tiempo = 100000;
        }
        IOWR_ALTERA_AVALON_PIO_DATA(0x4300, 0x01);
        usleep(tiempo);
        IOWR_ALTERA_AVALON_PIO_DATA(0x4300, 0x00);
        usleep(tiempo);
    }
    return 0;
}
```

Nota: Se esta haciendo un enmascaramiento al momento de leer el PIO entrada para que solo se tome en cuenta el bit0 del puerto.

9

El LCD alfanumérico

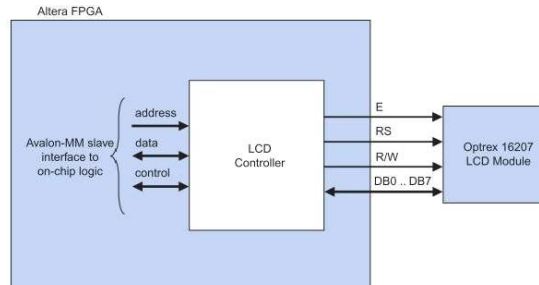


- Se pueden visualizar mensajes tanto en letras como en números.
- Su ROM de caracteres se asemeja a la tabla ASCII de 7bits
- Posee hasta ocho caracteres personalizados (CGRAM)
- Interface de 8 bits de datos + 3 líneas de control
- Hoja técnica:
<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	00	01	02	03	04	05	06	07	08	09	A	B	C	D	E	F
xxxx0001	(2)	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
xxxx0010	(3)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
xxxx0011	(4)	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
xxxx0100	(5)	U	V	W	X	Y	Z	[\]	^	_	`	a	b	c
xxxx0101	(6)	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
xxxx0110	(7)	s	t	u	v	w	x	y	z	{		}	~			
xxxx0111	(8)															
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

10

Módulo LCD del NIOSII: Optrex 16207 LCD Controller Core

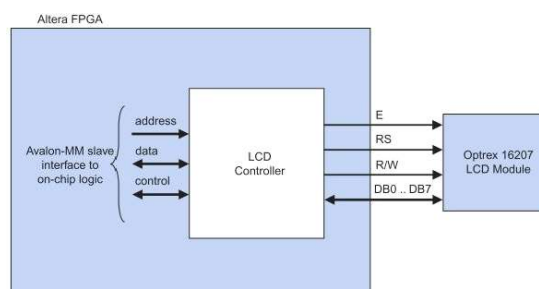


- Provee la interface en el NIOSII para comunicarse con un LCD alfanumérico basado en el controlador Hitachi HD44780.
- Se instancia en el QSys, tener en consideración de no crear conflicto con los demás módulos instanciados.

external_connection	lcd_16207_0	Conduit	lcds			
	reset	Altera Avalon LCD 16207				
	clk	Reset Input	Double-click to export	[clk]		
	control_slave	Clock Input	Double-click to export	clk_0		
	external	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x5200	0x520f
		Conduit	lcd1602			

11

Módulo LCD del NIOSII: Optrex 16207 LCD Controller Core



- Provee la interface en el NIOSII para comunicarse con un LCD alfanumérico basado en el controlador Hitachi HD44780.
- Se instancia en el QSys, tener en consideración de no crear conflicto con los demás módulos instanciados.
 - Establecer nombre a la conexión externa para que se pueda conectar hacia pines físicos del FPGA en el PinPlanner

external_connection	lcd_16207_0	Conduit	lcds			
	reset	Altera Avalon LCD 16207				
	clk	Reset Input	Double-click to export	[clk]		
	control_slave	Clock Input	Double-click to export	clk_0		
	external	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x5200	0x520f
		Conduit	lcd1602			

12

Comandos en el LCD alfanumérico

- 0x01 – Limpia el display
 - 0x02 – Mueve el cursor al inicio de la primera línea
 - 0x0C – Apaga el cursor
 - 0x0E – Cursor visible estático
 - 0x0F – Cursor visible parpadeando
 - 0x10 – Mueve cursor una posición a la izquierda
 - 0x14 – Mueve cursor una posición a la derecha
 - 0xC0 – Mueve cursor al inicio de la segunda línea
 - 0x94 – Mueve el cursor al inicio de la terca línea
 - 0xD4 – Mueve el cursor al inicio de la cuarta línea
- } Para LCDs de cuatro líneas*

15

Visualización de mensajes en el display:

- Visualizar “hola” en la primera linea

```
void main(void){
    //Funcion para inicializar el LCD
    lcd_init();

    //Comando para ubicarnos al inicio de la primera línea
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);

    //Visualización del mensaje "hola"
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'h');
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'o');
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'l');
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'a');
    usleep(100);

    while(1);

    return(0);
}
```

16

Visualización de mensajes en el display:

- Visualizar "hola" (cadena) en la primera línea

```
const unsigned char cadena[] = {"hola"};
int x_var = 0;

void main(void){
    //Funcion para inicializar el LCD
    lcd_init();

    //Comando para ubicarnos al inicio de la primera línea
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);

    //Visualización del mensaje "hola"
    for(x_var=0;x_var<4;x_var++){
        IOWR(LCD_0_BASE, LCD_WR_DATA_REG,cadena[x_var]);
        usleep(100);
    }
    while(1);
    return(0);
}
```

17

Visualización de mensajes en el display:

- Parametrizando en una función la visualización de una cadena

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

tam: Tamaño de la cadena

- Ejemplo

```
void main(void){
    //Funcion para inicializar el LCD
    lcd_init();

    //Comando para ubicarnos al inicio de la primera línea
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);

    //Visualización del mensaje "hola"
    ENVIA_MENSAJE("hola", 4);
    while(1);
    return(0);
}
```

18

Programa ejemplo completo:

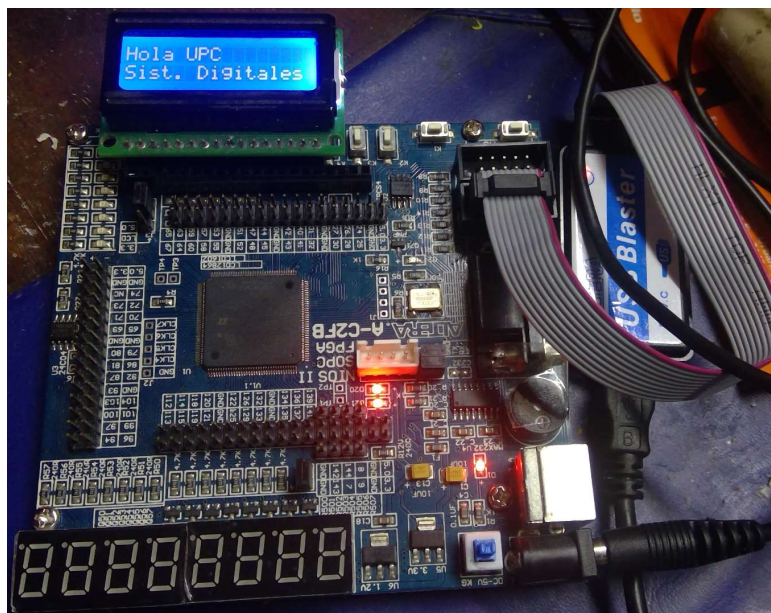
```

1#include "sys/alt_stdio.h"
2#include "system.h"
3#include "altera_avalon_lcd_16207_regs.h"
4
5#define LCD_WR_COMMAND_REG 0
6#define LCD_RD_STATUS_REG 1
7#define LCD_WR_DATA_REG 2
8#define LCD_RD_DATA_REG 3
9#define LCD_0_BASE 0x5200
10
11void lcd_init(void){
12    usleep(15000);
13    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
14    usleep(4100);
15    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
16    usleep(100);
17    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
18    usleep(5000);
19    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
20    usleep(100);
21    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
22    usleep(100);
23    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
24    usleep(100);
25    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
26    usleep(100);
27    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
28    usleep(2000);
29    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
30    usleep(2000);
31}
32
33void ESCRIBE_MENSAJE(const char *cadena,unsigned char tam)
34{
35    unsigned char i = 0;
36    for(i = 0; i<tam; i++)
37    {
38        IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
39        usleep(100);
40    }
41}
42
43void main(void)
44{
45    lcd_init();
46    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02); //Primera linea
47    usleep(2000);
48    ESCRIBE_MENSAJE("Hola UPC", 8);
49    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC0); //Segunda linea
50    usleep(2000);
51    ESCRIBE_MENSAJE("Sist. Digitales", 15);
52    while (1);
53}

```

19

Implementación en la tarjeta A-C2FB



20

Consideraciones finales

- Tener en cuenta que si se desea visualizar el contenido de alguna variable se tendrá que obtener los dígitos individuales (unidad, decena, centena, etc) para que pueda ser enviado uno por uno al display LCD.
- Un ejemplo de rutina de individualización de dígitos:

```
42 void convierte(unsigned int numero){  
43     d_millar = numero / 10000;  
44     millar = (numero %10000) /1000;  
45     centena = (numero % 1000) / 100;  
46     decena = (numero % 100) / 10;  
47     unidad = numero % 10;  
48 }
```