

Sistemas Digitales

Semana 12: Algoritmos y Diagramas de Flujo

Semestre 2020-2

Profesor: Kalun José Lau Gan

1

Preguntas previas:

- ?
- ??
- ???

2

Agenda:

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

3

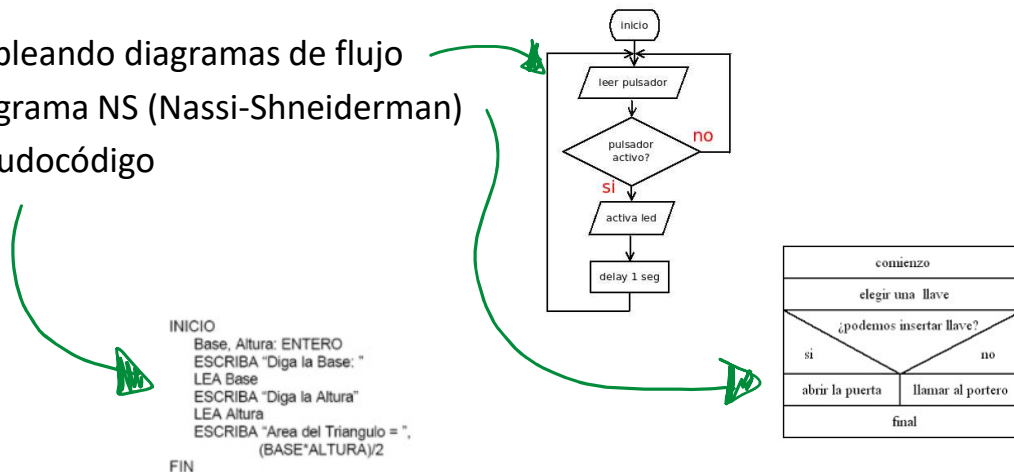
Algoritmos

- ¿Qué es un algoritmo? (5 minutos)
 - 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
 - Receta de cocina
 - Pasos para resolver un problema matemático, ingenieril, etc

4

¿Cómo represento un algoritmo informático?

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



5

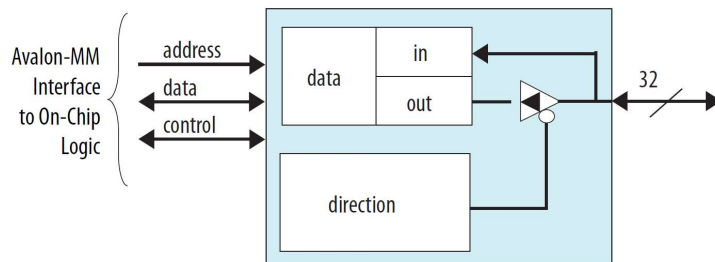
Diagramas de flujo (flowchart)

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

6

Manipulación de puertos en el NIOS II

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



- Revisar librería `altera_avalon_pio_regs.h`

7

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.

8

Librería *altera_avalon_pio_regs.h*

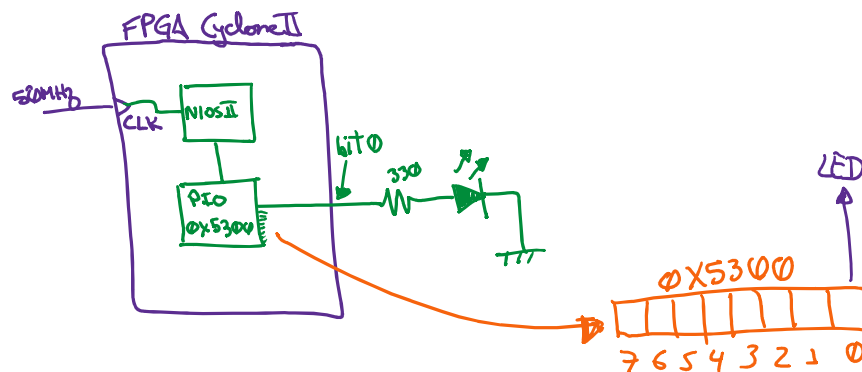
```

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_BITS(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 /* Definitions for direction-register operation with bi-directional PIOs */
42 #define ALTERA_AVALON_PIO_DIRECTION_INPUT  0
43 #define ALTERA_AVALON_PIO_DIRECTION_OUTPUT 1
44
45 #endif /* __ALTERA_AVALON_PIO_REGS_H_ */

```

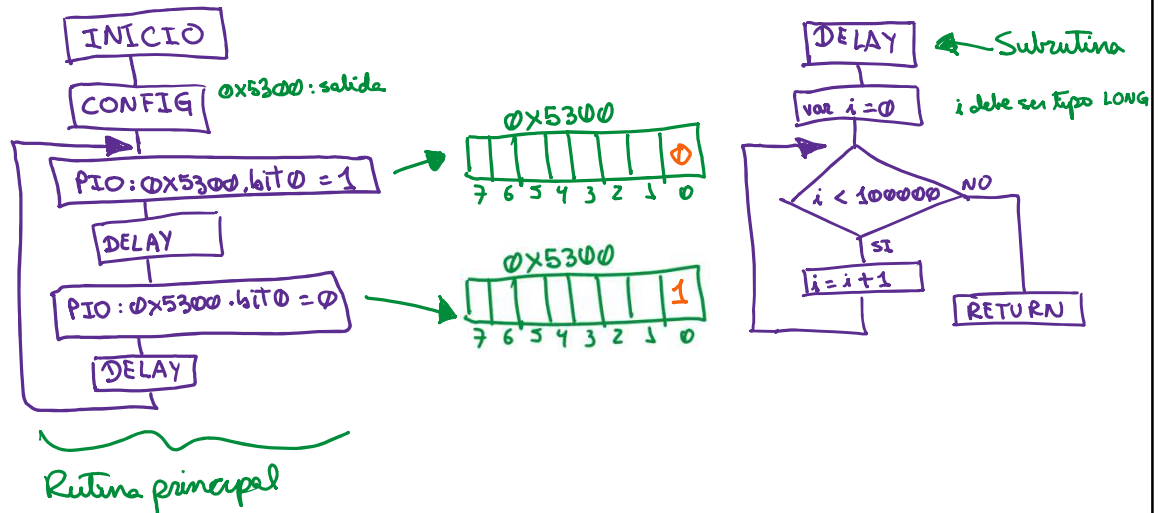
9

Ejemplo 1: Titilar un LED a través del bit0 del registro PIO de 8bits con dirección 0x5300



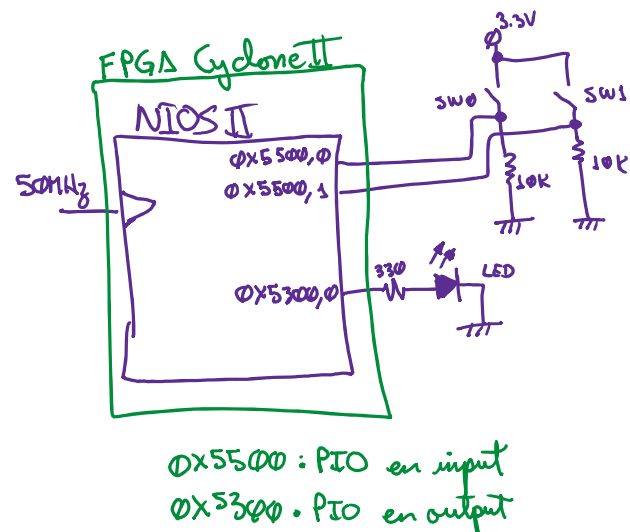
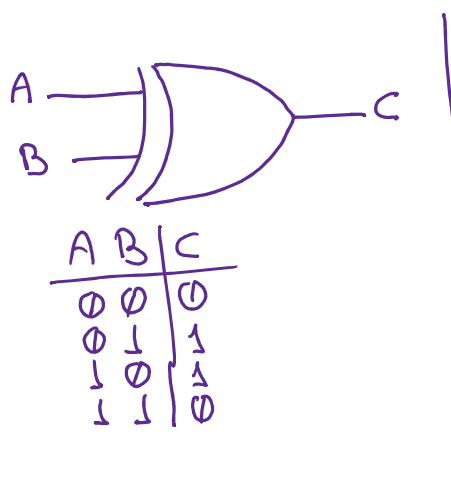
10

Ejemplo 1: Titilar un LED a través del bit0 del registro PIO de 8bits con dirección 0x5300



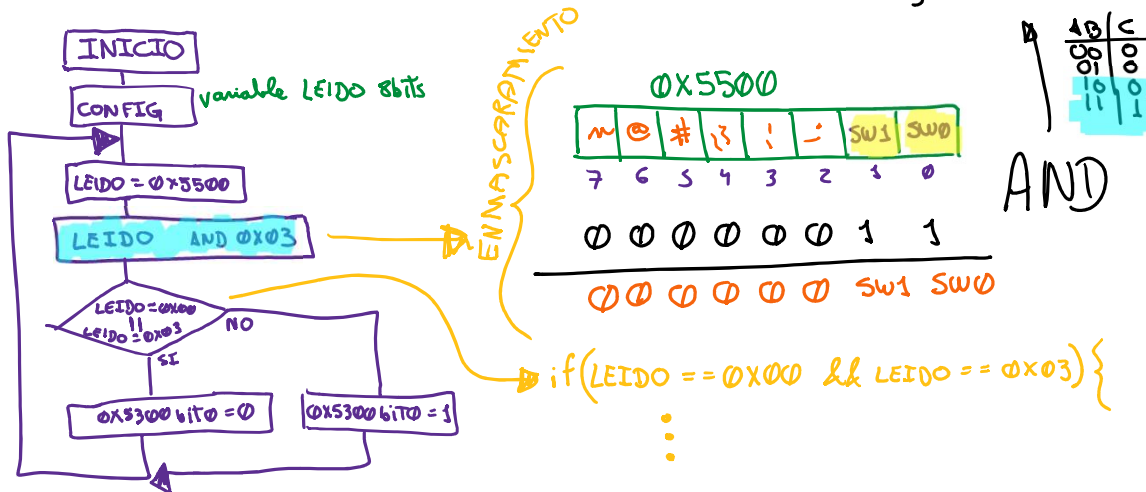
11

Ejemplo 2: Modelar una compuerta XOR en diagrama de flujo



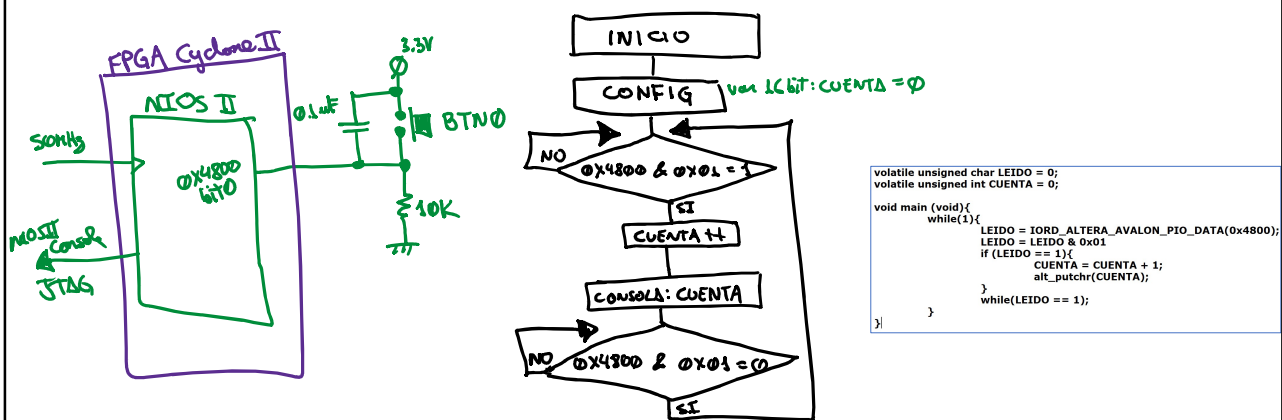
12

Ejemplo 2: Modelar una compuerta XOR en diagrama de flujo



13

Ejemplo 3: Detectar la cantidad de veces que se presiona un botón y visualizarlo como mensaje en la consola del NIOS II



14

Ejemplo 4: Desarrollar un algoritmo para visualizar las siglas UPC en un display de siete segmentos del tipo cátodo común

