

Pre-Informe 3

Sistem Verilog

Paula Pérez Muñoz
201603022-0

3.1 Investigación de Conceptos

- a) ¿Qué es un lenguaje de descripción de hardware? ¿Qué lo diferencia de un lenguaje de programación? Un lenguaje de descripción de hardware es un lenguaje que permite el modelamiento de hardware (todo lo descrito en este lenguaje puede ser utilizado para la construcción de hardware real). En algunos casos es posible incluso probar la funcionalidad del modelo. Se diferencia de un lenguaje de programación en que este último sirve para construir y diseñar software, no hardware.
- b) Explique la funcionalidad de los tipos de variable wire, reg y logic, y cuales son sus diferencias. (HINT: Recordar que los registros dependen del CLK)
 - wire: Representan conexiones estructurales entre componentes. No tienen capacidad de almacenamiento.
 - reg: Representan variables con capacidad de almacenar información.
 - logic: Tipo de dato que asignado a una variable puede tomar un valor binario.

La diferencia de reg con el resto de las variables es que esta almacena memoria.

- c) ¿Cuál es la diferencia entre los operadores de asignación assign, <= y =, y en que secciones del modulo se utilizan? La diferencia entre los operadores de asignación es que <= se usa para asignar un valor de manera temporal, mientras que = asigna de forma permanente (se representa como soldar un nodo).
- d) Dada la siguiente expresión:

$$reg[15 : 0]g = 16'hA6B2;$$

Explique su estructura, que significa cada una de sus partes, y cual es el equivalente binario del valor definido.

reg es el tipo de la variable, [15 : 0] indica que la variable puede almacenar hasta 16 bits, *g* es el nombre de la variable y 16'hA6B2 es el valor que se asigna a la variable, el número 16 indica su cantidad de bits, *h* indica su base hexadecimal y A6B2 su valor.

- e) Después de una subida del "clk", el registro "arr" resulta tener el valor binario 0111. Explique por qué el delay y las asignaciones dentro del bloque alwaysff influyen en este resultado. Las asignaciones dentro del bloque alwaysff solo se ejecutarán mientras se cumplan las condiciones descritas dentro de alwaysff, en este caso, el módulo indica que solo se ejecutarán en el canto de subida del clk.

3.2 Análisis de Código

Se puede apreciar en esta tabla que el valor final del *output* es 11011100

subida CLK	x	y	z	m	n	r	s
0	00000000	01110111	000	001	01110111	000	0
1	00000000	01110110	001	010	01110110	000	1
2	00000000	01110100	010	011	01110100	000	1
3	00000000	01110100	011	100	01110100	000	1
4	00010000	01110100	100	101	10000100	000	1
5	00010000	01100100	101	110	01110100	000	1
6	00010000	01000100	110	111	01010100	000	1
7	00010000	01000100	111	000	01010100	000	1
8	00010001	11000100	000	001	11010101	000	1
9	00010011	11000101	001	010	11011000	000	1
10	00010111	11000101	010	011	11011100	000	1

3.3 Diseño de Módulo

```

C:/Modeltech_pe_edu_10.4a/examples/contador5.sv (/contador5) - Default
Ln#
1
2 module contador5(input logic CLK, output reg[4:0] numero, output logic Fib);
3   assign Fib= (numero==5'b00001|| /*1*/
4     numero==5'b00010|| /*2*/
5     numero==5'b00011|| /*3*/
6     numero==5'b00101|| /*5*/
7     numero==5'b01000|| /*8*/
8     numero==5'b01101|| /*13*/
9     numero==5'b10101); /*21*/
10  always @(posedge CLK) begin
11    numero <= (numero + 1);
12  end
13  endmodule
14

```

Figure 1:



Figure 2: