

# Pre-Informe 3

Felipe Vicencio  
20071601-9

14 de Agosto 2019

## Pregunta 3.1

- a) Los lenguajes de descripción de hardware son lenguajes similares a los de programación, que permiten modelar sistemas electrónicos y electrónico-digitales. Permiten describir formalmente diseño y funcionamiento de un circuito, y ayudan a simularlo y a analizarlo de manera automática. La principal diferencia que tienen con los lenguajes de programación es que incluyen explícitamente el concepto del tiempo.
- b) **wire** es un tipo de dato de “red”. Se usan para variables que representan conexiones físicas. Usualmente estas variables se usan en asignaciones assign fuera de los bloques always, y se les suele asignar valores continuamente.  
**reg** se usa para guardar registros (valores). Usualmente van al lado izquierdo de asignaciones dentro de bloques always, y generalmente se les asigna un valor al ocurrir un evento (como una subida o bajada de clk, etc.).  
**logic** se puede utilizar como registro o como tipo de dato de “red”. Generalmente se usa como el tipo de dato de las variables de entrada y salida.
- c) **assign** asigna continuamente la expresión del lado derecho del signo =, al **wire** del lado izquierdo. Siempre se usan fuera de un always.  
 $\leftarrow$  es una asignación no bloqueante, es decir, todas las asignaciones de este tipo se ejecutarán en paralelo.  
= en cambio, es una asignación bloqueante, por lo que las asignaciones de este tipo se ejecutarán en orden. Ambas asignaciones ( $\leftarrow$  y =) van dentro de bloques always.
- d) reg [15:0]g Significa que se está declarando un registro de 16 bits, indexado con los dígitos menos significativos a la derecha, y este registro se identificará como “g”.  
Como este registro tiene “= h'A6B2” a la derecha, tendrá un valor inicial igual al número hexadecimal A6B2<sub>16</sub>, es decir, su valor inicial será 1010011010110010<sub>2</sub>.
- e) Antes de la primera subida de clk, se declara  $first = 0$ ,  $second = 1$  y  $arr = 0101$ .  
Luego de que sube clk,  $first$  cambia su valor a 1, pero  $second$  mantiene su valor, por lo que la siguiente línea se puede leer como  $arr[1] = \neg arr[1]$ , por lo que el bit en la posición 1 de  $arr$ , cambia a su negado, y por eso  $arr$  pasa de 0101 a 0111.  
Esto ocurre porque ninguna señal que cambie dentro de un always puede salir del bloque hasta que heste llegue a **end**. Si esto no fuera así  $second$  cambiaría a penas se declara  $first = 1$ , y la línea quedaría como  $arr[1] = \neg arr[0]$ , lo cual no cambiaría el valor de  $arr$ .

## Pregunta 3.2

Usando  $r = 011$ :

Estados iniciales:

$y_0 = 01110111$

$x_0 = 00000000$   
 $z_0 = 000$

$n_0 = 01110111$   
 $m_0 = 001$

**Subidas de CLK: 1**

$\neg y[m] = 0,$   
 $\neg y[z] = 0$   
 $y_1 = 01010111$   
 $x_1 = 00000000$   
 $z_1 = 110$

$n_1 = 01010111$   
 $m_1 = 111$

**Subidas de CLK: 2**

$\neg y[m] = 1,$   
 $\neg y[z] = 0$   
 $y_2 = 01010111$   
 $x_2 = 00000000$   
 $z_2 = 111$

$n_2 = 01010111$   
 $m_2 = 000$

**Subidas de CLK: 3**

$\neg y[m] = 0,$   
 $\neg y[z] = 1$   
 $y_3 = 01010111$   
 $x_3 = 00000001$   
 $z_3 = 000$

$n_3 = 01011000$   
 $m_3 = 001$

**Subidas de CLK: 4**

$\neg y[m] = 0,$   
 $\neg y[z] = 0$

$y_4 = 01010110$   
 $x_4 = 00000001$   
 $z_4 = 001$

$n_4 = 01010111$   
 $m_4 = 010$

**Subidas de CLK: 5**

$\neg y[m] = 0,$   
 $\neg y[z] = 0$   
 $y_5 = 01010100$   
 $x_5 = 00000001$   
 $z_5 = 010$

$n_5 = 01010101$   
 $m_5 = 011$

**Subidas de CLK: 6**

$\neg y[m] = 1,$   
 $\neg y[z] = 0$   
 $y_6 = 01010100$   
 $x_6 = 00000001$   
 $z_6 = 011$

$n_6 = 01010101$   
 $m_6 = 100$

**Subidas de CLK: 7**

$\neg y[m] = 0,$   
 $\neg y[z] = 1$   
 $y_7 = 01010100$   
 $x_7 = 00010001$   
 $z_7 = 100$

$n_7 = 01100101$   
 $m_7 = 101$

**Subidas de CLK: 8**

$\neg y[m] = 1,$

$\neg y[z] = 0$   
 $y_8 = 01010100$   
 $x_8 = 00010001$   
 $z_8 = 101$

$n_8 = 01100101$   
 $m_8 = 110$

### Subidas de CLK: 9

$\neg y[m] = 0,$   
 $\neg y[z] = 1$   
 $y_9 = 01010100$   
 $x_9 = 01010001$   
 $z_9 = 110$

$n_9 = 10100101$   
 $m_9 = 111$

### Subidas de CLK: 10

$\neg y[m] = 1,$   
 $\neg y[z] = 0$   
 $y_{10} = 01010100$   
 $x_{10} = 01010001$   
 $z_{10} = 111$

$n_{10} = 10100101$   
 $m_{10} = 000$

## Pregunta 3.3

```

module Counter (input logic clk,reset,
                output logic fib,
                output logic 5'b0 count

);

reg [4:0] a, [4:0]b;
reg fib1;
assign fib = fib1 | (count == 0);

always_ff @(posedge clk) begin
    if (reset) begin
        count <= 0;
        a <= 5'b0;
        b <= 5'b1;
    end
    else count <= count + 5'b1;
    if (fib1) {a, b} <= {b, count}; end

wire [4:0] next;
assign next = (b + a);

always_comb
    fib1 = (count == next) ? 1:0;

endmodule

```

Figure 1: Modulo del contador

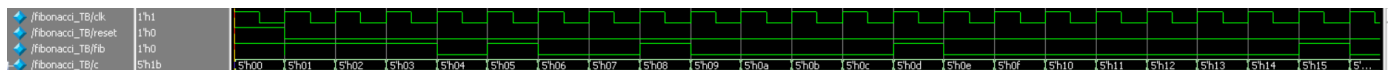


Figure 2: fib marca 1 en: 0, 1, 2, 3, 5, 8, 13 y 21