

Pre-Informe 3

INF-245

Josefina Onetto
201773036-6

Pregunta 1

- a) Lenguaje de programación que define la estructura, diseño y operación de circuitos electrónicos. Así, estos lenguajes hacen posible la descripción formal del circuito electrónico, posibilitando su análisis automático y su simulación. Su diferencia con los lenguajes de programación comúnmente usados es que el HDL incluye explícitamente la noción de tiempo.
- b)
- **Wire:** Esto es usado para conectar diversos elementos, representan cables reales. Pueden ser leídos y asignados, pero hay que darles valores continuamente, porque no tienen memoria.
 - **Reg:** Representan elementos de almacenamiento de datos. Mantiene su valor hasta que se le asigna uno nuevo.
 - **Logic:** Similar a reg, pero con un nombre diferente. Las diferencias entre logic y reg son que logic no permite múltiples drivers, ya que siempre toma el último valor asignado, mientras que reg o wire dará un valor x si tiene 2 asignaciones de valores diferentes.
- c)
- **Assign:** Modela lógica combinacional para denotar relación entre varias variables. (fuera de las estructuras always@)
 - **=** Se conoce como asignación con bloqueo. Esto se usa dentro de estructuras concurrentes para asignar valores, de forma que cada asignación puede afectar al valor de la siguiente.
 - **<=** Se conoce como asignación sin bloqueo. Se usa dentro de estructuras concurrentes para asignar valores de forma que las asignaciones de tipo <= no afectan el valor de otras asignaciones <= (se dice que esto se ejecuta con un cierto delay). En otras palabras, cada asignación se ejecuta como si no se hubieran ejecutado las demás.
- d) `reg [15:0] g = 16'hA6B2`
- **reg[15:0]g:** Esto indica que se tiene una variable de tipo reg y de nombre g que almacena un número de 16 bits.
 - **16'h:** Se refiere a un número hexadecimal de 16 bits.
 - **A6B2:** El número que g va a almacenar.
Su equivalente en binario corresponde a 1010 0110 1011 0010.
- e) Esto pasa porque el valor de first está siendo asignado con bloqueo, por lo que afecta a la siguiente línea. En cambio, el valor de second está en función de first por el assign, por lo tanto existe un delay que provoca que su valor no se actualice antes de que se ejecute la segunda línea.

Pregunta 2:

r	s	x	y	z	m	n
000	0	0000 0000	0111 0111	000	001	0111 0111
				000		
	1	0000 0000	0111 0110	001	010	0111 0110
		0000 0000	0111 0100	010	011	0111 0100
		0000 0000	0111 0100	011	100	0111 0100
		0001 0000	0111 0100	100	101	1000 0100
		0001 0000	0110 0100	101	110	0111 0100
		0001 0000	0100 0100	110	111	0101 0100
		0001 0000	0100 0100	111	000	0101 0100
		0001 0001	1100 0100	000	001	1101 0101
		0001 0011	1100 0101	001	010	1101 1000
		0001 0111	1100 0101	010	011	1101 1100

Note que partimos con $r = 000$, y que cada fila representa una subida del clk , excepto por las primeras dos que son parte de la primera iteración.

Pregunta 3:

```
module up_counter
(
output logic [4:0] c,
output logic fib,
input logic clk, output logic [4:0] first,
output logic [4:0] second
);
initial begin
    c <= 1;
    first <= 5'b00001;
    second <= 5'b00001;
    fib <= 1;
end
always @(posedge clk) begin
    c = c + 1;
    if (first+second == c) begin
        second <= first;
        first <= c;
        fib <= 1;
    end
    else begin
        fib <= 0;
    end
end
endmodule
```

