

Pre-Informe 3

Diego Moraga Araya
201773035-8

Pregunta 1: Investigación de conceptos

- a) Es un lenguaje de programación especializado que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos, y generalmente, de circuitos electrónicos digitales. Así, los lenguajes de descripción de hardware hacen posible una descripción formal de un circuito electrónico y posibilitan su análisis automático y su simulación. La diferencia entre los HDL y otros lenguajes de programación está en que el HDL incluye explícitamente la noción de tiempo.
- b) Los elementos wire son cables simples (o buses de ancho arbitrario). Los elementos reg son similares a los wire, pero se pueden usar para almacenar información ("estado") como registros. No hay absolutamente ninguna diferencia entre el reg y el logic en SystemVerilog, excepto por la forma en que se escriben, son sinónimos de palabras clave. logic está destinado a reemplazar reg porque originalmente se pensaba que reg era la abreviatura de registro. También tenga en cuenta que la lógica es un tipo de datos para una señal, mientras que el cable es un tipo de señal.
- c) En primer lugar **assign** es una asignación continua para cablear fuera de un bloque always y cuando cambia el valor de la derecha cambia, se actualiza el de la izquierda. El operador \leq se denomina non-blocking se realiza en cada canto positivo del clock y se evalúan en paralelo, sin garantizar orden; se utilizan en los bloques always. Finalmente el operador $=$ se denomina blocking y se utiliza dentro de los bloques always para imponer un orden de asignación secuencial.
- d) En primer lugar *reg* representa el tipo de dato que se está declarando, luego $[15 : 0]$ indica el tamaño del bus, en este caso de 16-bit, *g* es el nombre de la variable o dato que se iguala a un número escrito en hexadecimal que representa el valor 1010011010110010 en binario.
- e) En este caso influyen las asignaciones dado que cuando se le reasigna con un blocking al first en el bloque always, de cierta manera por el delay, second no alcanza a actualizarse antes de que se ejecute el non-blocking del always, dado el assign por el cual fue definido second y porque fue definido afuera del bloque always.

Pregunta 2: Análisis de Código

Ruteo de n con $r = 000$:

$n = 01110111$ (estado inicial)

- 1er posedge CLK
 $n = 01100110$
- 2do posedge CLK
 $n = 01110100$
- 3er posedge CLK
 $n = 01110100$

- 4to posedge CLK
 $n = 10000100$
- 5to posedge CLK
 $n = 01110100$
- 6to posedge CLK
 $n = 01010100$
- 7mo posedge CLK
 $n = 01010100$
- 8vo posedge CLK
 $n = 11010101$
- 9no posedge CLK
 $n = 11011000$
- 10mo posedge CLK
 $n = 11011100$

Pregunta 3: Diseño de Módulo

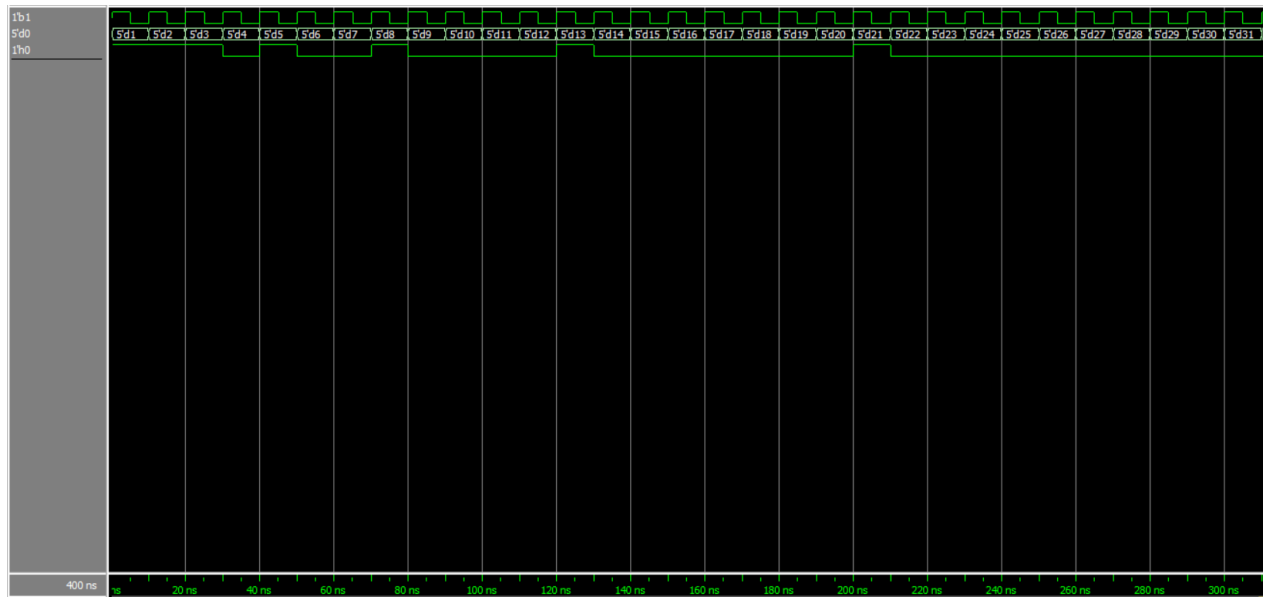


Figure 1: Wave de simulación

```

1 module contador(clk,n);
2   input logic clk;
3   reg [4:0] cont = 5'd0;
4   output reg n = 1'b1;
5   reg [4:0] f1 = 5'd1; //uno
6   reg [4:0] f2 = 5'd2; //dos
7   reg [4:0] f3 = 5'd3; //tres
8   reg [4:0] f4 = 5'd5; //cinco
9   reg [4:0] f5 = 5'd8; //ocho
10  reg [4:0] f6 = 5'd13; //trece
11  reg [4:0] f7 = 5'd21; //veintiuno
12  reg [4:0] f8 = 5'd0; //cero
13  always@(posedge clk)
14  begin
15    cont<=cont+1;
16    if((((cont+1) == f1) | ((cont+1)==f2) | ((cont+1)==f3) | ((cont+1)==f4) | ((cont+1)==f5) | ((cont+1)==f6) | ((cont+1)==f7)) & n)
17      begin
18        continue;
19      end
20
21    else
22      begin
23        if(((cont+1) == f1) | ((cont+1)==f2) | ((cont+1)==f3) | ((cont+1)==f4) | ((cont+1)==f5) | ((cont+1)==f6) | ((cont+1)==f7) | ((cont+1)==f8))
24          begin
25            n = ~n;
26          end
27        else begin
28          n=0;
29        end
30      end
31  end
32 endmodule
33

```

Figure 2: Código simulado