

Pre-Informe 3

César Paulangelo
201773088-9

Pregunta 1

- (a) Un lenguaje de descripción de hardware es un lenguaje diseñado para el diseño de la estructura y operación de varios circuitos electrónicos, siendo estos mayoritariamente circuitos digitales. Además, permiten la descripción formal de los circuitos, y posibilitan su análisis automático y simulación. Una de sus principales diferencias con los lenguajes de programación, es su enfoque. Mientras el primero se enfoca en hardware, el otro se enfoca en la creación de algoritmos y software.
- (b) El tipo de dato REG se usa para almacenar información en verilog como variables. El tipo de dato WIRE es una forma de representar a conexiones entre componentes en circuitos eléctricos. el ultimo tipo de dato, el tipo LOGIC, no esta presente en verilog, se encuentra en system verilog. Tiene como función declarar variables, puertos de input, y puertos de output, y además de todas las redes punto a punto.
- (c) El operador continuo *assign* es una asignación que sucede de manera continua y de manera paralela a cualquier otra tarea computacional. El operador \leq se usa para asignaciones que ocurren solo una vez, pero pueden suceder en cualquier momento. Por último, el operador $=$ realiza asignaciones en orden secuencial, además, el código que se ubica después de la asignación no se ejecutara hasta que la asignación hay tenido lugar.
- (d) La expresión consta de varias partes, la primera es la definición del tipo de dato a utilizar(en este caso, un registro), luego se define el tamaño en bits. donde el numero mas a la izquierda equivale al bit mas significativo (N-1). Luego viene el nombre de la variable, el operador de asignación, para finalmente terminar con el valor a asignar a g. Este valor corresponde a numero hexadecimal de 16 bits equivalente en binario a 1010 0110 1011 0010 o 42674 en decimal
- (e) Como los valores solo cambian al subir el clock ocurre un delay en las asignaciones. Es por eso, que en el primer clk, ocurre la asignación $arr[1] = arr[1]$, para luego en el segundo clk el WIRE second tenga su valor actualizado.

Pregunta 2

Clock#	clk	r	n	x	y	z	m	s
1	0	000	1110111	00000000	01110111	000	001	0
	1		01110110		000	010	1	
	0		01110100		001	010		
2	1	01110100	01110100	010	011			
	0							
3	1	01110100	01110100	011	100			
	0							
4	1	10000100	00010000	01110100	100	101		
	0							
5	1	01110100	01110100	01100100	101	110		

Pregunta 3

```
Module jkff(input reset ,
  input clk ,
  input j ,
  input k ,
  output reg q ,
  output qnot);
assign qnot=~q;
always @(posedge clk)
  if (reset) q<=1'b0; else
    case ({j , k})
      2'b00: q<=q;
      2'b01: q<=1'b0;
      2'b10: q<=1'b1;
      2'b11: q<=~q;
    endcase
endmodule

module contador(output logic o);
logic reset;
logic clk;
reg [4:0] fibo;
reg [4:0] fiboant;

logic q0;
logic q1;
logic q2;
logic q3;
logic q4;
logic j;
logic notq0;
logic notq1;
logic notq2;
logic notq3;
logic notq4;
reg [4:0] n;
assign n[0] = q0;
assign n[1] = q1;
assign n[2] = q2;
assign n[3] = q3;
assign n[4] = q4;
// Contador de jk flip-flop de 5 bits
jkff ff0 (.reset(reset),
  .clk(clk),
  .j(j),
  .k(j),
  .q(q0),
  .qnot(notq0));
jkff ff1 (.reset(reset),
  .clk(notq0),
  .j(j),
  .k(j),
  .q(q1),
```

```

.qnot(notq1));
jkff ff2 (. reset (reset ),
.clk(notq1),
.j(j),
.k(j),
.q(q2),
.qnot(notq2));
jkff ff3 (. reset (reset ),
.clk(notq2),
.j(j),
.k(j),
.q(q3),
.qnot(notq3));
jkff ff4 (. reset (reset ),
.clk(notq3),
.j(j),
.k(j),
.q(q4),
.qnot(notq4));

initial
begin
reset <= 1; #10; // tiempo de espera antes de comenzar
j <= 1;
fibo <= 5'd1;
fiboant <= 5'd1;
reset <= 0; #0;
o <= 1'b0;
end

always
begin
clk <= 1; #5;
clk <= 0; #5;
end

always @(negedge clk )
begin
if (fibo == n) begin
o <= 1;
fibo <= fibo + fiboant;
fiboant <= fibo;
end else begin
o <= 0;
end
end

endmodule

```

