

# Pre-Informe 1

## *Arquitectura y organización de computadores*

Felipe Otero  
20160400-6

### 1 Investigación de conceptos

- a) Un lenguaje de descripción de hardware se utiliza para definir la estructura, diseño y operación de circuitos electrónicos. Hacen posible una descripción formal de un circuito electrónico, posibilitando así el análisis y simulación del mismo. Soportando operaciones paralelas, esperas al canto del reloj, tiempos de propagación.

Los lenguajes de programación comunmente son ordenes secuenciales (no paralelas) de maquinas abstractas, los lenguajes de descripción de hardware te permiten especificar maquinas concretas.

- b) La variable Wire nos sirve para diseñar sistemas digitales combinatoriales (aquellos que no pueden almacenar memoria) mientras que la variable Reg nos permite diseñar sistemas digitales secuenciales, y la variable logic nos sirve para representar variables booleanas.

- c) '=' es una asignacion que bloquea otras asignaciones hasta que se completa la asignacion en esa linea (se utiliza para definir un sistema logico combinatorial), mientras que 'j=' es una asignación que permite asignaciones paralelas, es decir que ocurren en un mismo paso de tiempo, por esto mismo se utilizan para crear sistemas lógicos secuenciales.

'assign' es una forma de asignación continua, por lo tanto sirve para definir variables de tipo wire en la formacion de logica combinatorial.

- d) Se realiza una asignación de bloqueo de una variable llamada 'g' a un tipo reg.

- [15:0] significa un 'arreglo' de 16 bits.

'h' expresa que se guardará un numero en hexadecimal.

El equivalente binario seria

$$reg[15 : 0]g = 16'b1010011010110010$$

- e) Dentro del bloque always notamos dos tipos de asignaciones; una asignacion bloqueo y una asignacion no bloqueo. En este caso, cuando la asignacion bloqueo comienza ejecutarse, ningun otro tipo de asignación se ejecuta en el mismo paso de tiempo, por ende, la asignacion no bloqueo de 'arr[first]' comienza una vez termina la asignación 'first = 1', es por esto que la asignacion no bloqueo se ejecuta de la siguiente forma:

$$arr[1] = -(arr[1])$$

## 2 Análisis de código

x	y	z	m	n	s
0000 0000	01110111	0	0	119	
		0	1		
		0	2		
0000 0000	0111 0110	1	2	118	
0000 0000	0111 0100	2	3	116	
0000 0000	0111 0100	3	4	116	
0001 0000	0111 0100	4	5	132	
0001 0000	0110 0100	5	6	116	
0001 0000	0110 0100	6	7	84	
0001 0000	0100 0100	7	0	84	
0001 0001	1100 0100	0	1	213	
0001 0011	1100 0101	1	2	216	
0001 0111	1100 0101	2	3	220	

## 3 Diseño de módulo

Debido a la naturaleza del problema, un circuito con logica combinacional es suficiente para diseñar lo pedido. Para esto se confecciona un mapa de karnaught con los minterms en los numerosde fibonacci;

$$f(A, B, C, D, C) = \Sigma(1, 2, 3, 5, 8, 13, 21)$$

Entregando como resultado:

$$y = AB'C'E + A'B'C'D + A'CD'E + B'CD'E + A'BC'D'E'$$

Codigo:

```

module Fibonacci (input logic clk,
                  output logic y);
    reg [4:0] c = 5'b0;
    assign y = ((~(c[4]))&(~(c[3]))&(~(c[2]))&(c[0]) |
               ~(c[4]))&(~(c[3]))&(~(c[2]))&(c[1]) |
               ~(c[4]))&(c[2])&(~(c[1]))&(c[0]) |
               ~(c[3]))&(c[2])&(~(c[1]))&(c[0]) |
               ~(c[4]))&(c[3])&(~(c[2]))&(~(c[1]))&(~(c[0]))
    );
    always_ff@(posedge clk)
    begin
        c <= c +1;
    end
endmodule

```

