

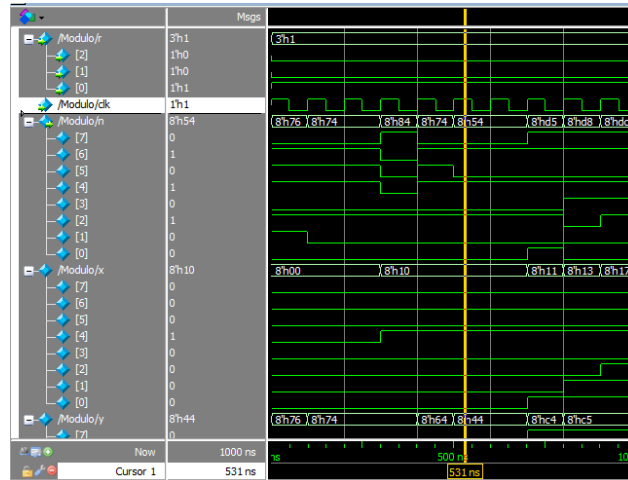
# Pre-Informe 3

Esteban Manríquez  
201673020-6

## 1 Pregunta 1

- a) Un lenguaje de programación de hardware (o HDL) es un lenguaje de programación que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos. Si bien posee similitudes con otros lenguajes de programación (C, Java) al ser ambos descripciones textuales con expresiones, declaraciones y estructuras de control, la diferencia importante entre ambos es que los HDL incluye explícitamente la noción de tiempo, además de que sólo especifica funciones lógicas.
- b) Wire: Es un net data type, lo que significa que es utilizado para representar conexiones físicas, no guarda ningún valor.  
Reg: Es un variable data type, por lo que guarda un valor que se le es asignado. Generalmente es utilizado para modelar registros de hardware.  
Logic: Introducida en SystemVerilog, se utiliza como un nuevo tipo de variable que sirve para asignar valores de data type (ambos anteriores).  
Se diferencian en que wire no representa ningún valor (es decir, no lo guarda), y es utilizado para representar conexiones físicas. Por otro lado, reg se utiliza para guardar variables, donde deben suceder mediante un "trigger", por ejemplo la subida de un CLK. Finalmente, logic es utilizado para englobar ambas variables anteriores mediante un 'logic' después del tipo de variable (wire o reg en este caso).
- Las 3 son variables de asignación, la diferencia es que assign es utilizado para 'variables que cambian' con el tiempo, es decir que son monitoreadas cada cierto tiempo. Por otra parte, '<=' es más utilizado para lógica secuencial, mientras que el '=' es utilizado en lógica combinatorial. Los '<=' normalmente son utilizados en flip-flops, mientras que el assign es utilizado dentro de las secuencias de '@always'.
- reg: Nos indica que es un registro  
'[15:0]': nos indica la cantidad de bits( desde el bit 15 hasta el 0).  
g: nos dice el nombre de la variable.  
h: nos indica la base del número, en este caso es un hexadecimal.  
A6B2: Nos indica el número en cuestión.  
Finalmente, podemos decir que es un registro de 16 bits, asignado a la variable g, donde el valor numérico (en binario) es '1010011010110010'.
- Esto es debido a que las asignaciones utilizadas '<=' nos ayudan a evitar la 'race-around condition', es decir, evitar que el tiempo del clk sea mayor que el de las variables alternan entre sí (dando un caso general para los flip-flop sería cuando j y k valen 1)

## 2 Pregunta 2



Aquí vemos como avanzan los valores con el tiempo. Luego de que se asignan los valores, z empieza a actuar como contador y en cada loop, modificará los valores de y, para luego modificar los de x en referencia al resto. El S 'setea' el valor de z la primera vez para que comience a contar.

## 3 Pregunta 3

```
module fib(output logic [5:0]out);
    int cont = 0; //valor bool
    int cur = 1;
    int prev = 0;
    always#1clk = ~clk;
    always \@(posedge clk) begin
        cont = 0
        cur <= cur + prev;
        prev <= cur;
        if (out == cur)
            cont = 1;
        out += 1;
    end
endmodule
```

Aquí se puede ver como se tienen dos variables que se irán sumando siguiendo la regla de fibonnaci, tendremos un valor de 1 o 0 que nos sirve como bool y nuestro número de 5 bits que con cada loop irá sumando 1 (cuando el clk pase a positivo).