

Complido mendoza A01029238

1) What is Verilog?

un sistema de programación de hardware ^{lenguaje} ✓

2) How many values can take an output port?

2, unos y ceros (2), falta X y Z

3) What is the purpose of a Testbench in Verilog?

Probar y ver que es lo que pasa con las señales que se mandan, tanto de entrada como de salida. ✓

4) How do you instantiate a module in Verilog?

necesitar por "module" y el nombre del módulo, a toda poner "1";
dentro poner los inputs y outputs, ejemplo:

```
module boton (
    input wire btn,
    output reg Resp
);
```

~~no están instanciada~~ ←

y para mandarlo a llamar es el nombre del módulo "boton" y su nombre específico, igual con "1";, para dentro le asignar las variables de tu módulo principal al del módulo "boton"

```
boton btn1 (
    .btn(btn),
    .Resp(Resp)
);
```

 ✓

5) What is the purpose of a wire in Verilog? ✓

Como su nombre lo dice, funciona como un "cable", que puede ser "1" o "0", no se guarda el valor y funciona para establecerlo a una variable, ya depende para que quieras usarlo, por ejemplo para unir módulos, o sea que puedes definir el output de un módulo como un wire para mandar la señal a la entrada (input) de otro módulo.

6) difference between blocking and non-blocking assignments

blocking ya está definido, no se puede cambiar después y
non-blocking es más susceptible a los cambios

2/3 blocking = comb
non-blocking = secuencial

7) define an always block

un bloque always es una función que funciona siempre y cuando los parámetros establecidos para su función se cumplan, la función va a correr preferiblemente con un clock o un reset, aunque también le puedes poner otros parámetros.

always @(posedge clk) ✓

begin

... ..

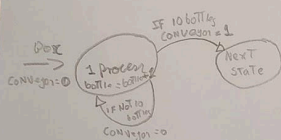
end

... es el nombre del módulo "Botom d"
"()": para objeto de asignar las variables de Tu
del módulo "Botom"

1) I want to create a module that count between 0 to 100,
What length of bits need in the internal register used to count?
 $2^7 = 128$, necesito una longitud de 7 bits para poder mostrar
hasta el número 100

Se calcula con la función de Ceillog

9) Define a FSM diagram for a bottle packer that performs
the following steps: ~~~~



¿Se puede regresar?

10) Define a structure of finite state machine module
primero define en parámetros los estados

Parameter state0 = 0,
state1 = 1,
state2 = 2

la estructura principal de una FSM es con uno o varios
always, en este caso utilizo 3 always.

montado a mano es el nombre del módulo
co, igual con "()", para dentro se asignan las variables de la
Principal al del módulo "Boton"
n Bus (✓
(Bus,
p (Res)

donde el primer always sirve para reset

1) always @(posedge clk or posedge rst)

if (rst)

state <= state 0

los 3 always funcionan con los flancos positivos del reloj

el segundo always me sirve para definir el current state y
cual sería el next state

2) always @(posedge clk) (pseudocódigo solo para entender la idea)

case:

state 0 → next-state <= state 1

state 1 → next-state <= state 2

state 2 → next-state <= state 0

endcase

el tercer always sirve para definir que se va a hacer en ese
estado. como en el ejemplo de las botellas, si está en el primer
estado, se va a quedar en ese estado hasta que se llene la
caja con 10 botellas, ahí se pasa al next-state

* Cronometro parcial
Sumador parcial.