

## Boletín 1: Lenguaje de descripción Verilog

Última modificación: 20. Feb. 2017

### Problema 1

Describa la función  $f = a'b + cd'$  en Verilog utilizando las tres descripciones posibles: funcional, estructural y procedimental.

### Problema 2

Repita el ejercicio anterior para  $f = abc + a'b'c' + a'bc'$ .

### Problema 3

Describa en Verilog un decodificador de 3:8 con salidas activas en alto. Utilice descripción funcional y procedimental.

### Problema 4

¿Qué modificaciones habría que realizar en el código Verilog del problema 3 para incorporar señal de habilitación activa en bajo? ¿Y para que las salidas sean también activas en bajo?

### Problema 5

Se desea utilizar la descripción procedimental para implementar un codificador de prioridad de 4:2 que responda a la siguiente tabla de verdad.

$I_0$	$I_1$	$I_2$	$I_3$	$Q_1$	$Q_0$	$E_0$
0	1	1	1	0	0	0
X	0	1	1	0	1	0
X	X	0	1	1	0	0
X	X	X	0	1	1	0
1	1	1	1	0	0	1

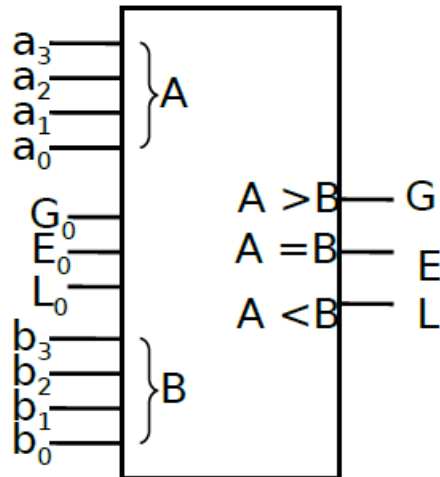
### Problema 6

Describa un convertidor BCD a 7-segmentos en Verilog.

**Boletín 1: Lenguaje de descripción Verilog**

**Problema 7**

Describe un comparador de magnitudes de 4 bits, como el que se muestra a continuación, utilizando una descripción procedimental.



**Problema 8**

Utilizando tanto las descripciones funcional como procedimental, escriba el código Verilog de un MUX 4:1

**Problema 9**

Describe en Verilog un sumador/restador de n bits.

**Problema 10**

Una ALU de 4 bits implementa las funciones que se recogen en la siguiente tabla. Escriba el código Verilog de dicha ALU.

$S_2$	$S_1$	$S_0$	Función ALU	
			$C_{in} = 0$	$C_{in} = 1$
0	0	0	$F = A$	$F = A + 1$
0	0	1	$F = A + B$	$F = A + B + 1$
0	1	0	$F = A + B'$	$F = A + B' + 1$
0	1	1	$F = A - 1$	$F = A$
1	0	0	$F = A \text{ AND } B$	
1	0	1	$F = A \text{ OR } B$	
1	1	0	$F = \text{NOT } A$	
1	1	1	$F = A \text{ XOR } B$	

**Problema 11**

Utilizando el módulo descrito en el problema 10, construya una ALU de 12 bits en Verilog utilizando la descripción estructural.

**Problema 12**

Describa en Verilog un registro de 8 bits con la siguiente tabla de operación.

CL LD	Operación	Tipo
0 x	$REG \leftarrow 0$	asíncrona
1 1	$REG \leftarrow X$	síncrona
1 0	$REG \leftarrow REG$	síncrona

**Problema 13**

Describa en Verilog un registro de desplazamiento con la siguiente tabla de operación.

CL SHR	Operación	Tipo
0 x	$REG \leftarrow 0$	asíncrona
1 1	$REG \leftarrow SHR(REG, X_R)$	síncrona
1 0	$REG \leftarrow REG$	síncrona

**Problema 14**

Describa en Verilog un registro universal de 8 bits con las siguientes operaciones

CL LD SHR SHL	Operación	Tipo
0 x x x	$REG \leftarrow 0$	asínc.
1 1 x x	$REG \leftarrow X$	sínc.
1 0 1 x	$REG \leftarrow SHR(REG, X_R)$	sínc.
1 0 0 1	$REG \leftarrow SHL(REG, X_L)$	sínc.
1 0 0 0	$REG \leftarrow REG$	sínc.

**Problema 15**

Describa un contador 4 bits con la siguiente tabla de operación.

CL	Operación	Tipo
1	$CONT \leftarrow 0$	asínc.
0	$CONT \leftarrow CONT + 1 \mid_{\text{mod } 16}$	sínc.

## Boletín 1: Lenguaje de descripción Verilog

### Problema 16

Describa un contador de 4 bits con la siguiente tabla de operación y salida de Cy.

CL EN	Operación	Tipo
1 x	$\text{CONT} \leftarrow 0$	sínc.
0 1	$\text{CONT} \leftarrow \text{CONT} + 1 \mid_{\text{mod } 16}$	sínc.
0 0	$\text{CONT} \leftarrow \text{CONT}$	sínc.

### Problema 17

Escriba el código Verilog de un contador módulo-16 con la siguiente tabla de operación.

LD EN	Operación	Tipo
1 x	$\text{CONT} \leftarrow X$	sínc.
0 1	$\text{CONT} \leftarrow \text{CONT} + 1 \mid_{\text{mod } 16}$	sínc.
0 0	$\text{CONT} \leftarrow \text{CONT}$	sínc.

### Problema 18

Escriba el código Verilog de un contador módulo-10.

EN	Operación	Tipo
1	$\text{CONT} \leftarrow \text{CONT} + 1 \mid_{\text{mod } 10}$	sínc.
0	$\text{CONT} \leftarrow \text{CONT}$	sínc.

### Problema 19

Describa un contador descendente de n bits con salida de borrow.

CL EN	Operación	Tipo
1 x	$\text{CONT} \leftarrow 0$	sínc.
0 1	$\text{CONT} \leftarrow \text{CONT} - 1 \mid_{\text{mod } 16}$	sínc.
0 0	$\text{CONT} \leftarrow \text{CONT}$	sínc.

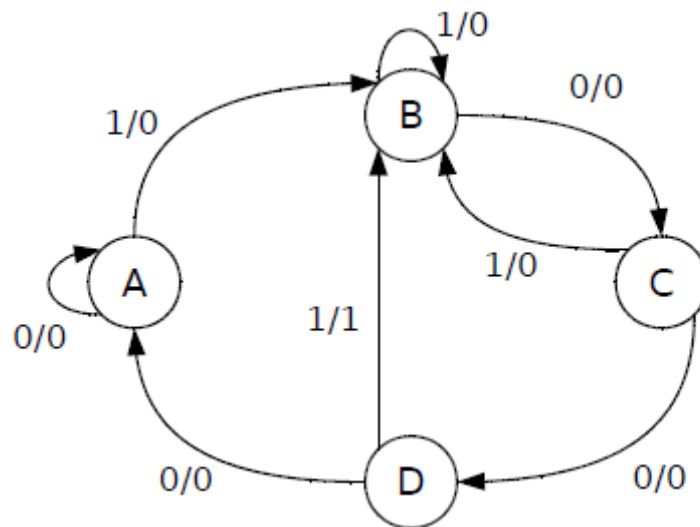
**Problema 20**

Describe un contador reversible de 4 bits con la siguiente tabla de operación.

CL EN UD	Operación	Tipo
1 x x	$CONT \leftarrow 0$	asínc.
0 0 x	$CONT \leftarrow CONT$	sínc.
0 1 0	$CONT \leftarrow CONT+1 _{\text{mod } 16}$	sínc.
0 1 1	$CONT \leftarrow CONT-1 _{\text{mod } 16}$	sínc.

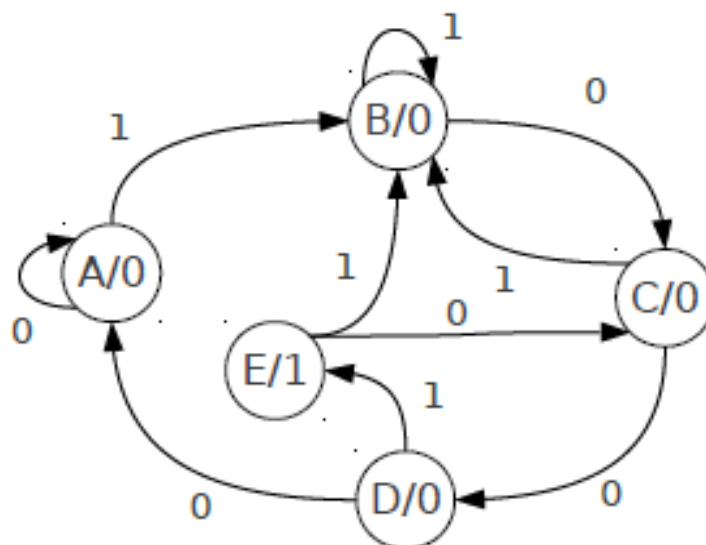
**Problema 21**

Describe la siguiente FSM usando Verilog.



**Problema 22**

Describe la siguiente FSM usando Verilog



## Boletín 1: Lenguaje de descripción Verilog

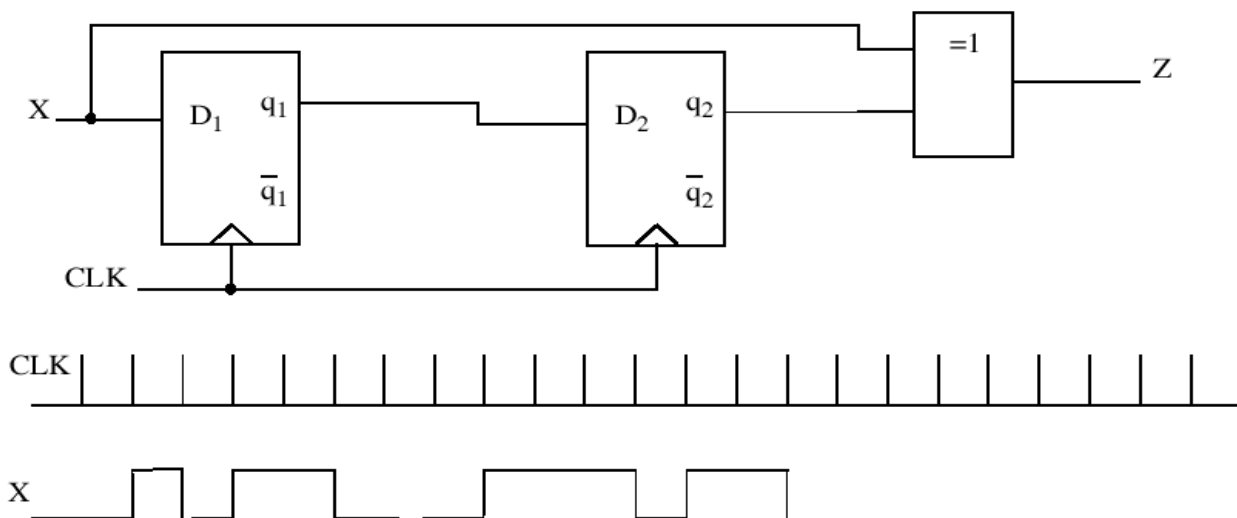
### Problema 23

Describe la siguiente FSM usando Verilog

X \ S	0	1
E <sub>0</sub>	E <sub>0</sub> ,00	E <sub>1</sub> ,00
E <sub>1</sub>	E <sub>2</sub> ,00	E <sub>1</sub> ,01
E <sub>2</sub>	E <sub>2</sub> ,10	E <sub>3</sub> ,10
E <sub>3</sub>	E <sub>0</sub> ,10	E <sub>3</sub> ,11

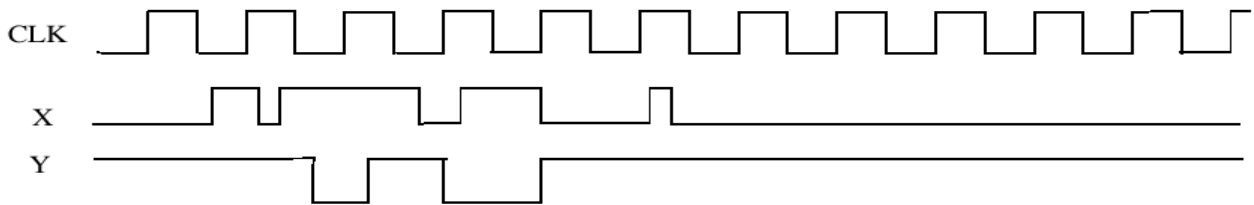
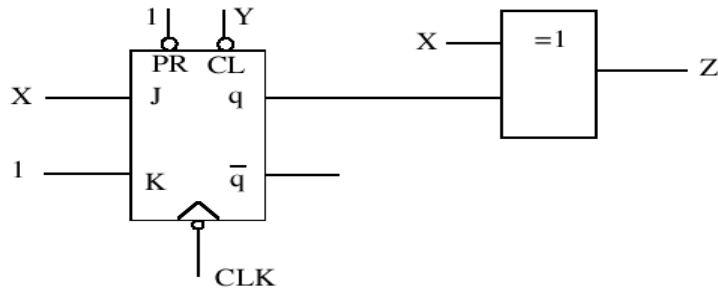
NS, Y,Z

**Problema 24** Describe el siguiente circuito utilizando código Verilog. Emplee descripción procedimental para los biestables y estructural para el conjunto. Obtenga la salida z para la secuencia de entrada que se muestra, para ello cree el fichero de test que permita su simulación.



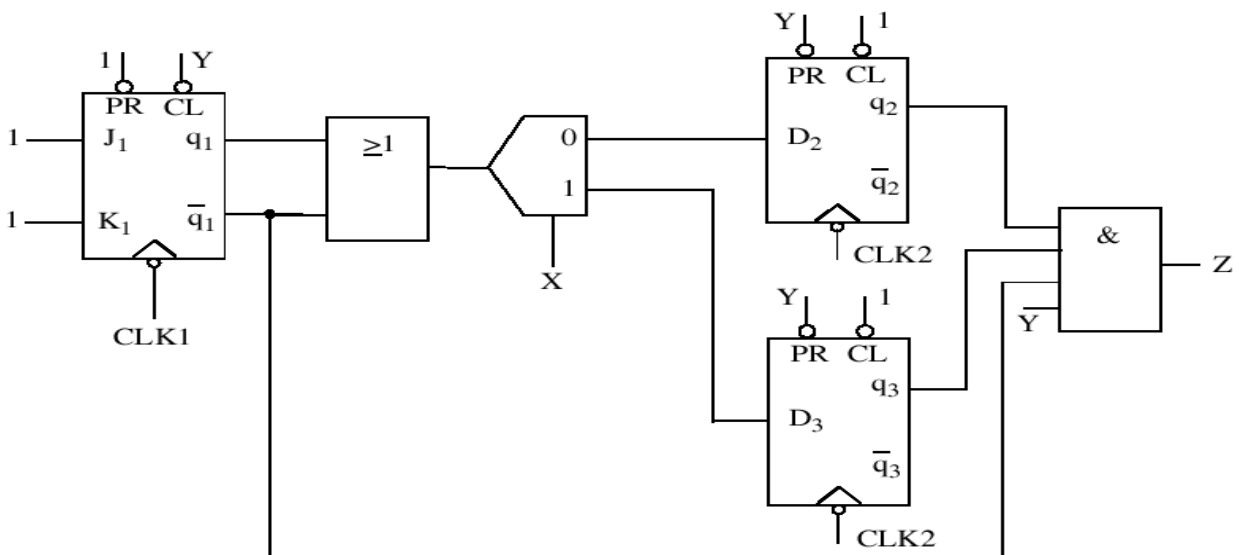
**Problema 25**

Repita el ejercicio anterior para el siguiente esquemático.

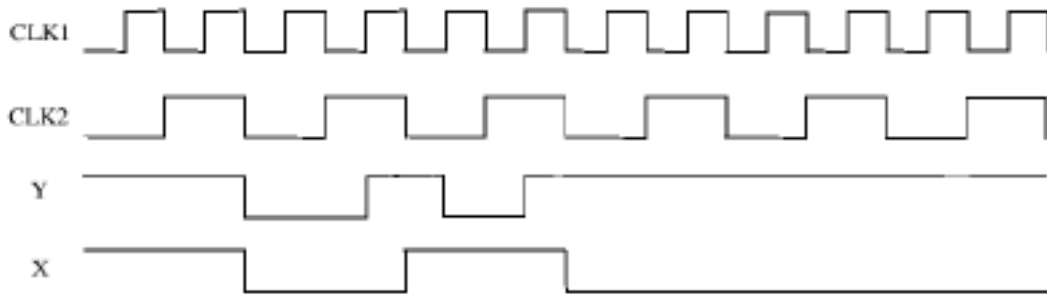


**Problema 26**

Describe en Verilog el siguiente circuito y obtenga el fichero de test necesario para su simulación, utilizando las secuencias de entradas que se muestran en la figura.

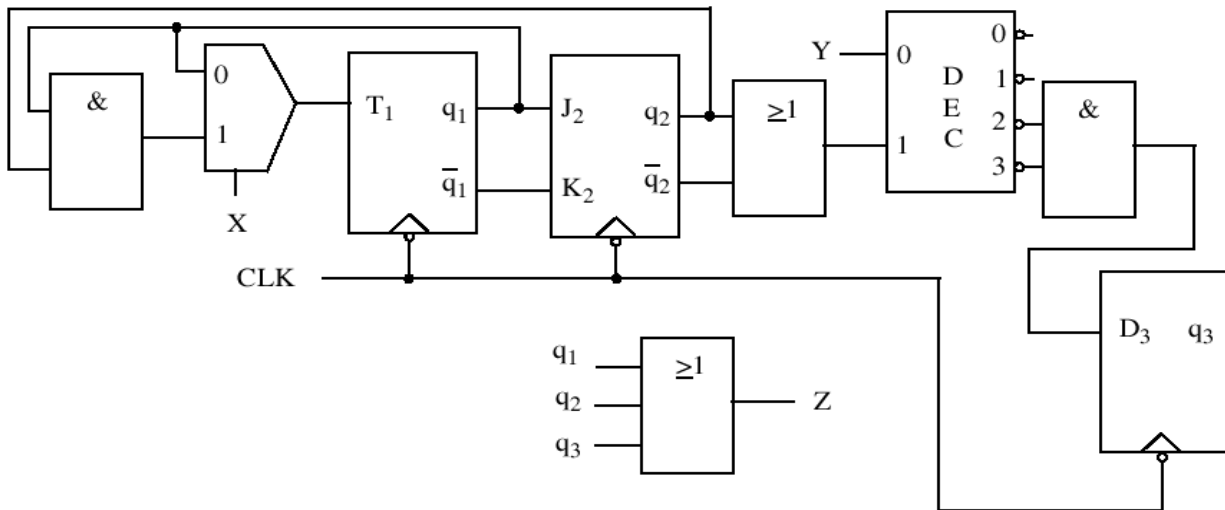


**Boletín 1: Lenguaje de descripción Verilog**

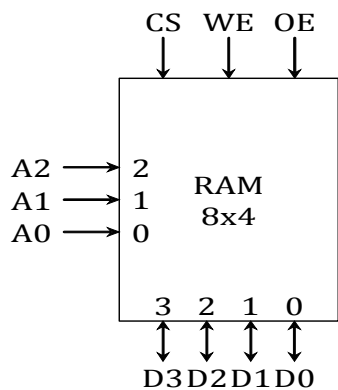


**Problema 27**

Describe el siguiente circuito en Verilog.



**Problema 28** Describe la siguiente RAM en Verilog





**Problema 29**

Indique qué dispositivo se describe a continuación y de su tabla de verdad.

```
module incognita(  
    input CS,  
    input [2:0] A,  
    output reg [3:0] D  
);  
  
always @(CS, A)  
    if (CS)  
        case (A)  
            0: D = 'h3;  
            1: D = 'h8;  
            2: D = 'hA;  
            3: D = 'hB;  
            4: D = 'h7;  
            5: D = 'h5;  
            6: D = 'hC;  
            default: D = 'h4;  
        endcase  
    else  
        D = 'hZ;  
    end  
endmodule
```