



Ejercicio 1. (2,5 puntos)

a. Explique la relación entre coste/bit, capacidad y velocidad en las memorias semiconductoras.

Ver teoría

b. Responda a la pregunta: ¿qué diferencia hay entre las memorias de acceso secuencial y las de acceso aleatorio? Cite algún ejemplo de memoria semiconductor para cada uno de estos dos casos.

Ver teoría

c. Explique el concepto de memoria volátil y de memoria con lectura destructiva.

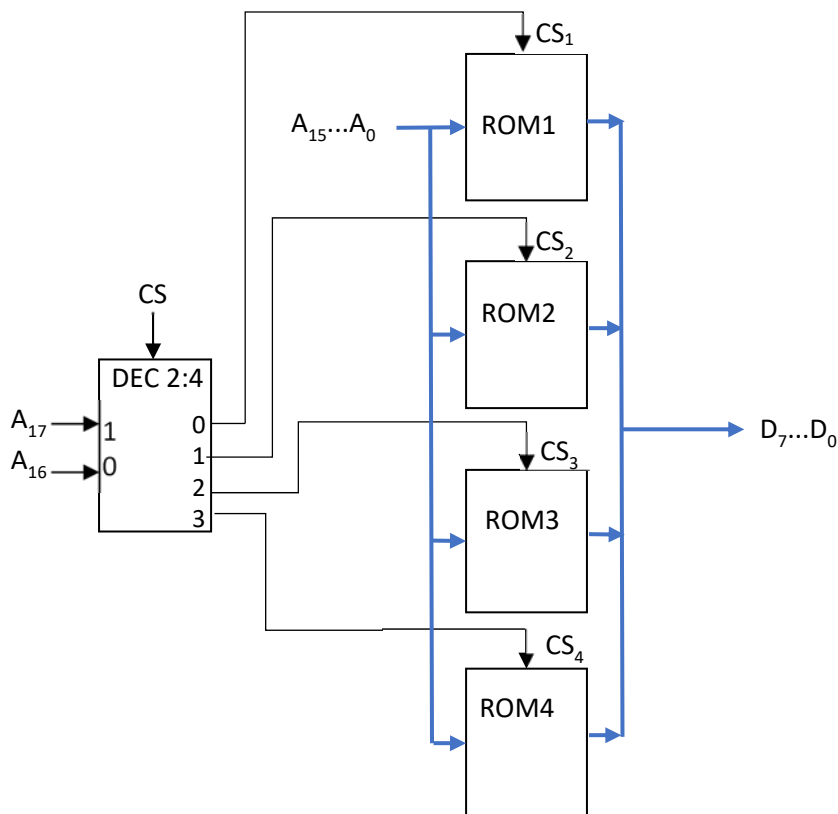
Ver teoría

d. Considere que dispone de un número suficiente de memorias ROM con 16 líneas de dirección, anchura de palabra de 8 bits y señal de selección de chip CS (activa en alta) que posibilita colocar las salidas en alta impedancia. Dispone también de los elementos combinatoriales que necesite. Se pide: obtener una memoria de 256KBytes.

Una memoria de 16 líneas de dirección y anchura de palabra 8 bits tiene 2^{16} direcciones y por tanto es una memoria de capacidad $2^{16} \times 8 = 2^6 \cdot 2^{10} \times 8 = 64K \times 8 = 64K\text{Bytes}$.

Para obtener la memoria de 256KBytes se usarán 4 memorias de 64KB.

El bus de datos de 8 bits debe ser único y por tanto las líneas de salida de las memorias se conectan entre sí. Gracias a las señales CS se puede usar un decodificador para seleccionar solo una de las memorias en cada lectura. Esto se hará a través de las 2 líneas adicionales que posee la memoria de 256KB ($256 = 2^{18}$), líneas A_{17} y A_{16} .



Ejercicio 2. (3,5 puntos)

Considere los siguientes módulos Verilog y conteste a las cuestiones que se le plantean.

```

module circuitoA (input xA, input [2:0] Y, output reg [3:0] out);

always @* begin
    if(xA==0)
        out = 4'bZ;
    else
        case (Y)
            3'h0: out=4'b0000;
            3'h1: out=4'b0001;
            3'h2: out =4'b0011;
            3'h3: out =4'b0111;
            3'h4: out =4'b1111;
            3'h5: out =4'b1110;
            3'h6: out =4'b1100;
            3'h7: out =4'b1000;
        endcase
    end
endmodule
    
```

```

module circuitoB(
    input xB,ck,
    output reg [2:0] out
);

always @(posedge ck, posedge xB)
    if (xB==1)
        out=0;
    else
        out=out+1;

endmodule
    
```

```

module circuitoC(
    input ck,
    input R,
    output [3:0] z
);

wire [2:0] w;

circuitoB inst1 (.ck(ck),.xB(R),.out(w));
circuitoA inst2 (.xA(1'b1),.Y(w),.out(z));

endmodule
    
```

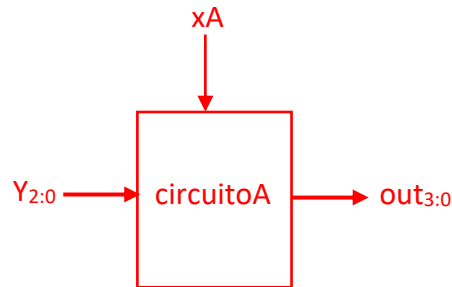
- a. ¿Qué tipo de descripción (estructural, funcional o procedimental) se ha usado para cada uno de los circuitos?

circuitoA y circuitoB → procedimental

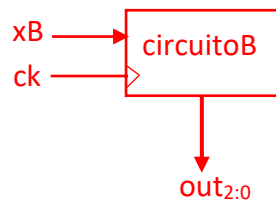
circuitoC → estructural

- b. Observe la función que realizan circuitoA y circuitoB y diga de qué circuitos se trata y dibuje sus diagramas de bloque. Diga también si la señal xB en circuitoB es síncrona o asíncrona.

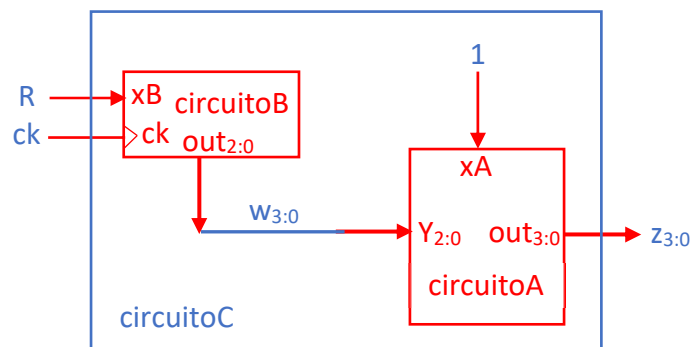
circuitoA → memoria ROM de 8 palabras de 4 bits (capacidad $2^3 \times 8$), donde **xA** es la señal de selección de chip activa en alto, **Y** es el bus de direcciones (3 bits) y **out** es el bus de datos (4 bits).



circuitoB → contador binario ascendente de 3 bits (módulo 8), donde **xB** es la señal de reset asíncrona activa en alto, **ck** es la señal de reloj (flanco de subida) y **out** es el bus de salida (3 bits).



- c. Realice un dibujo de circuitoC donde se observen claramente los módulos que lo componen, como están conectados entre sí, cuáles de las señales son internas y cuáles son entradas y salidas de circuitoC.



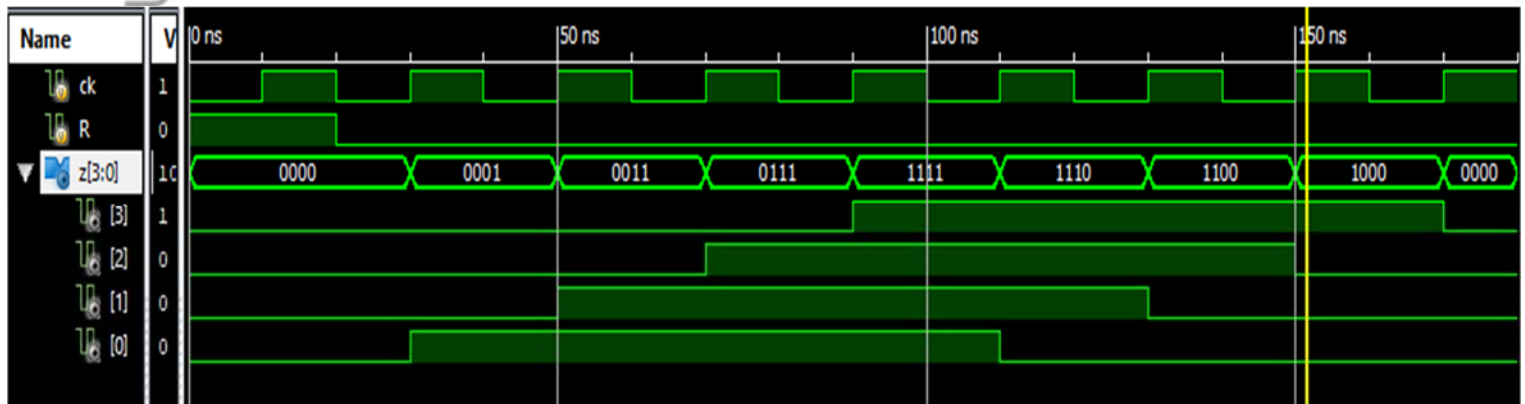
d. Considere el testbench que se suministra a continuación y dibuje las ondas de entrada que proporciona. Dibuje también las salidas con que respondería el módulo circuitoC.

```

module circuitoC_tb;
    reg ck;
    reg R;
    wire [3:0] z;

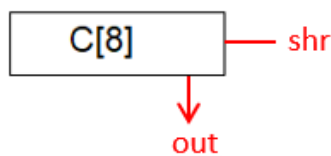
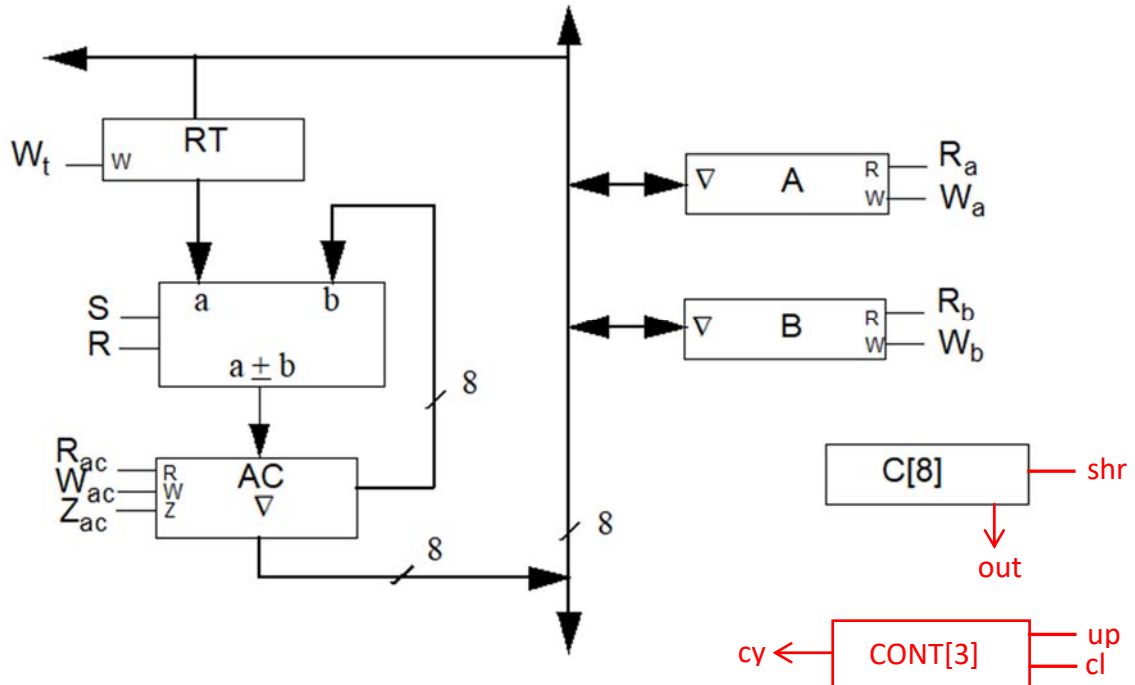
    circuitoC uut ( .ck(ck), .R(R), .z(z));

    initial begin
        ck= 0;
        R=1;
        #20 R=0;
        #160;
        $finish;
    end
    always begin
        #10 ck=~ck;
    end
endmodule
    
```



Ejercicio 3. (4 puntos)

Complete la unidad de datos de la figura para realizar la operación $B \leftarrow N \times A$, donde N viene dado por el número de unos contenidos en el registro C. Debe definir a nivel RT el registro C tras completarlo adecuadamente. Puede añadir otros elementos (debe describirlos). Obtenga la carta ASM de la unidad de control.



shr	C ←	out =
0	C	C ₀
1	SHR(C,0)	C ₀

cl up	CONT ←	cy
0 0	CONT	=1 si
0 1	CONT + 1	[CONT]=7
1 -	0	

El registro C será un registro de desplazamiento. Desplazaremos sus bits y cada vez que haya un 1 sumaremos A. De esta forma sumaremos A tantas veces como unos haya en C. Necesitamos también el contador CONT para recorrer exactamente los 8 bits del registro C.

Al principio, comenzamos poniendo a 0 el AC y el CONT, a la vez que cargamos A en RT. Después comenzamos el bucle donde comprobamos el valor del bit de C que toque, incrementamos el contador, desplazamos el dato de C para la próxima iteración y en caso de que el bit haya sido 1 hacemos $AC \leftarrow AC + RT$ (sumamos A al AC). Repetimos esto mientras $Cy = 0$.

Finalmente, cargamos el resultado en B haciendo $B \leftarrow AC$.

La carta ASM queda como sigue:

