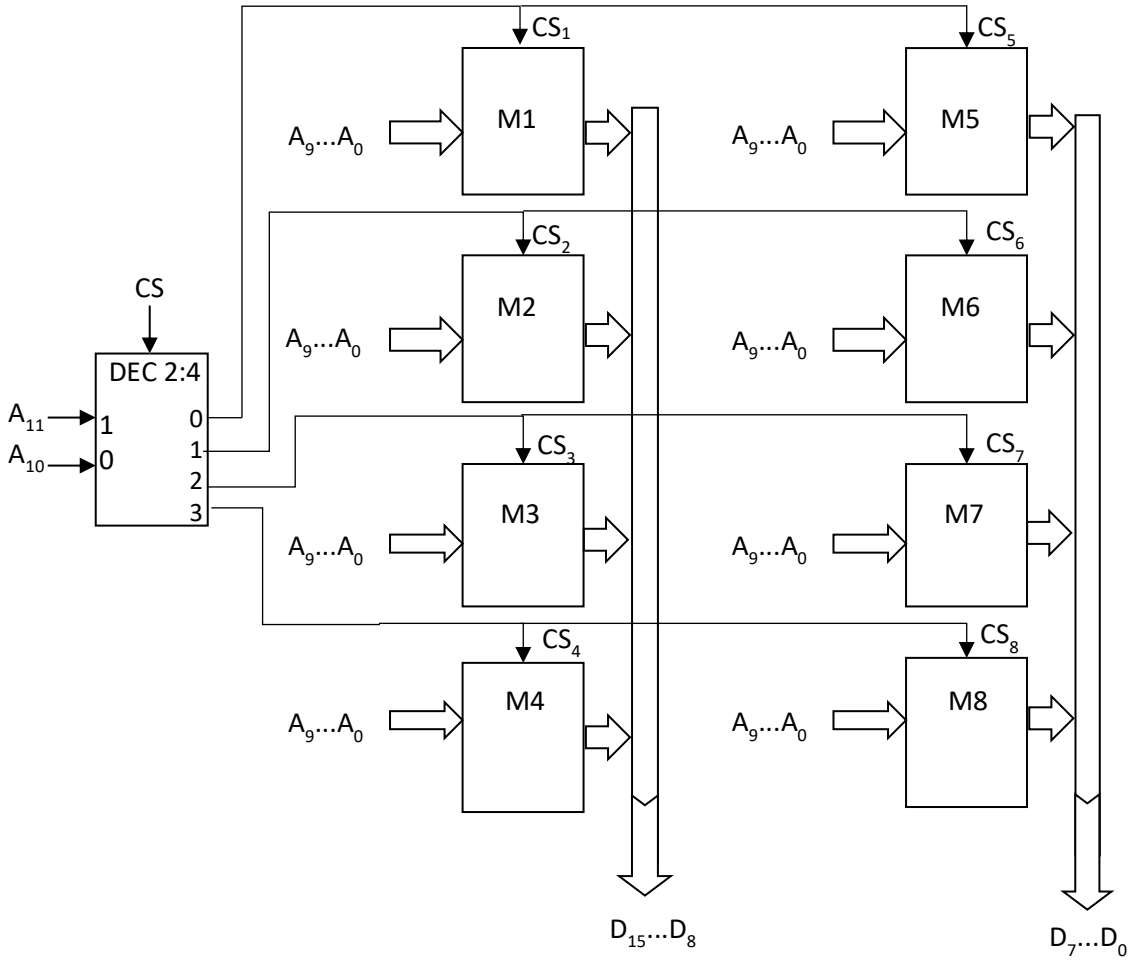
ALUMNO:_____

Memorias (10 puntos - peso 40%)

1. Explique la diferencia entre memoria de acceso aleatorio y memoria de acceso secuencial. **(1 punto)**
2. ¿Qué es una memoria volátil? **(1 punto)**
3. Si consideramos la jerarquía de memorias en un computador ¿Qué relación existe entre velocidad y coste por bit? ¿Y qué relación hay entre coste por bit y capacidad? **(2 puntos)**
4. En la figura se muestra una asociación de memorias (M1, M2, …,M8). Todas las memorias son iguales y contienen palabras de 8 bits. Responda las cuestiones:
    a. ¿Qué capacidad tiene cada una de las memorias?, ¿cuántas palabras?, ¿de qué longitud? (**1.5 puntos)**
    b. Si consideramos solo la asociación de las memorias M1 y M5 ignorando el resto del circuito ¿qué capacidad tiene la memoria resultante? ¿cuántas palabras? ¿de qué longitud? **(1.5 puntos)**
    c. Si consideramos el decodificador junto con las memorias M1, M2, M3 y M4 (es decir, ignorando el resto de las memorias), ¿qué capacidad tiene la memoria resultante? ¿cuántas palabras? ¿de qué longitud? (**1.5 puntos)**
    d. Si consideramos el circuito completo, ¿qué capacidad tiene la memoria resultante? ¿cuántas palabras? ¿de qué longitud? **(1.5 puntos)**



Verilog (10 puntos – peso 60%)

1. Responda las cuestiones que se plantean: **(2 puntos)**
   e. Cite los tres tipos de descripción existentes en Verilog
   f. Explique la diferencia entre una variable tipo wire y una variable tipo reg. ¿Qué tipo ha de utilizarse dentro de un proceso always? ¿y en una descripción con assign? ¿y en una descripción estructural?
   g. Explique la diferencia entre utilizar conexión nombrada y conexión posicional.
   h. ¿Qué significa el signo + en Verilog?

2 El siguiente código describe un contador de 4 bits. Diga qué operaciones controlan las señales *in1* e *in2*. Rellene la lista de sensibilidad del proceso always para que el contador esté disparado por flanco de bajada y tenga puesta a cero asíncrona. **(2 puntos)**

```
module contador (input ck,in1,in2,
                 output [3:0] z);

reg [3:0] q;

always @( lista de sensibilidad)
    if (in1 == 1)
        q <= 0;
    else if (in2==1)
        q <= q + 1;
assign z = q;

endmodule
```

3 A partir del código anterior haga las modificaciones necesarias para que corresponda a un registro de 4 bits con puesta a cero y carga en paralelo <u>síncronas</u> controladas por in1 e in2. **(2 puntos)**

4 Considere el siguiente código, ¿a qué dispositivo corresponde? Puede decirlo con palabras o dar una tabla. **(2 puntos)**

```
module ejemplo (input [7:0] in, input w, input shl, ck,
                output [7:0] out);

reg [7:0] q;

always @(negedge ck)
    if (w)
        q = in;
    else if (shl)
        q = {q[6:0],q[7]};
assign out = q;
endmodule
```

5 Considere el siguiente testbench (corresponde al módulo del apartado anterior) y dibuje las ondas que genera. **(2 puntos)**

```
module ejemplo_tb;

 reg [7:0] in;
 reg w, shl, ck;
 wire [7:0] out;

 ejemplo  uut (.in(in),.w(w),.shl(shl),.ck(ck),.out(out));

 always #10 ck = ~ck;

 initial begin
        ck=0; w=0; shl=0; in = 8'h8a;
        @ (posedge ck) w=1;
        @ (posedge ck) w=0;
        @ (posedge ck) shl=1;
         repeat (3) @(posedge ck);
         shl=0;
        @ (posedge ck);
        @ (posedge ck);
        $finish;
 end
 endmodule
```