

**Ej. 1** — (2.0 puntos) Diseñar la ruta de datos y la unidad de control (diagrama ASM y tabla de salidas) de un sistema algorítmico que dado dos entradas,  $n$  de 4 bits y  $x$ , y una salida,  $y$  de 16 bits, implemente la siguiente relación de recurrencia:

$$a_0 = x \quad (1)$$

$$a_{i+1} = a_i \cdot f_i \quad (2)$$

de forma que la salida  $y = a_{n-1}$ , siendo  $f_i$  una serie de valores almacenados a partir de la dirección 0x00 de una memoria RAM<sup>(1)</sup> síncrona de 32x4 bits y un solo puerto de lectura y escritura. En el caso de que alguna de las entradas  $n$  o  $x$  sea igual a cero el sistema pondrá la salida error a 1, la salida  $y$  a 0, y finalizará la ejecución del algoritmo. El sistema comienza a funcionar cuando la señal  $ini$  se pone a 1. En ese instante se almacenan los valores de  $n$  y  $x$ . Cuando se completa el cálculo el sistema volverá al estado inicial en el que la señal  $fin$  es igual a 1.

```

a ← x;
error ← 0;
if n = 0 or x = 0 then
    error ← 1;
    fin;
end
for i=0, i < n, i++ do
    a ← a · mem(i);
end
y ← a;

entity asm is
    port (clk      : in  std_logic;
          rst_n    : in  std_logic;
          ini      : in  std_logic;

          n        : in  std_logic_vector(n-1 downto 0);
          x        : in  std_logic_vector(n-1 downto 0);
          y        : out std_logic_vector(n-1 downto 0);
          fin      : out std_logic;
          error     : out std_logic
    );
end asm;

entity ram is
    port (clk      : in  std_logic;
          dina     : in  std_logic_vector(n-1 downto 0);
          addra    : in  std_logic_vector(4 downto 0)
          wea      : in  std_logic;
          ena      : in  std_logic;
          douta    : out std_logic_vector(n-1 downto 0);
    );
end ram;

```

En la ruta de datos se puede usar una memoria RAM<sup>(1)</sup> síncrona de un solo puerto, un contador ascendente/descendente<sup>(2)</sup>, registros, una única ALU que realiza sumas y multiplicaciones, y los elementos combinacionales adicionales que se consideren necesarios. Se valorará el uso del hardware mínimo. <sup>(1)</sup>: Suponer que los puertos de salida conservan el valor leído hasta que se realice una nueva lectura. <sup>(2)</sup>: La definición de puertos del contador ascendente/descendente:  $din$ , bus dato de entrada;  $load$ , señal de carga;  $ce$ , habilita la cuenta;  $ud$ , cuenta arriba ( $ud=1$ ) o cuenta abajo ( $ud=0$ );  $dout$ , bus de salida.



**Ej. 2 — (1.5 puntos)** Diseñar, utilizando puertas lógicas de dos entradas:

1. Una red iterativa 1D que dados dos vectores de entrada, A y B, de n bits genere una salida, z, de forma que  $z=1$  si  $A=B$  y  $z=0$  en caso contrario.
2. Una red en árbol que implemente el diseño anterior. Ayuda: la celda básica del primer nivel será distinta a la celda básica de los demás niveles.
3. Suponiendo que todas las puertas de dos entradas tienen un retardo t, calcular el retardo del camino crítico en los dos diseños.



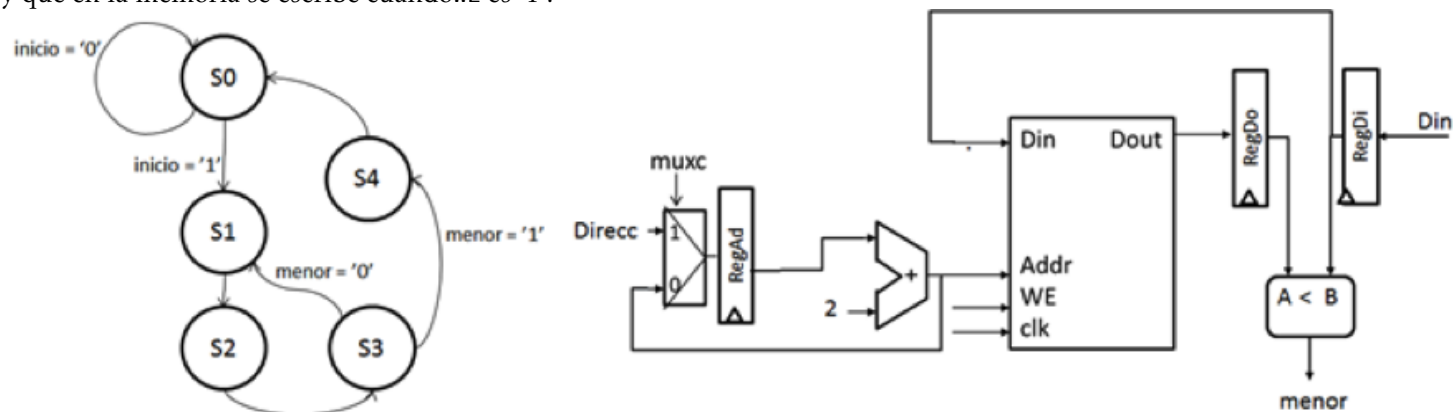
TEC. COMPUT. // TEC. Y ORGA. COMPUT.

2º

GII — GIC

09/02/2015

Ej. 4 — (1.5 puntos) Completar el cronograma correspondiente al siguiente sistema e indicar el contenido final de la memoria. Las entradas al sistema son Inicio, Din, Direcc, clk y rst (=0 en el cronograma); la salida del sistema es Fin. El resto de señales son señales internas según tabla adjunta. Considerar que la memoria es síncrona tanto para lectura como para escritura y que en la memoria se escribe cuando WE es '1'.

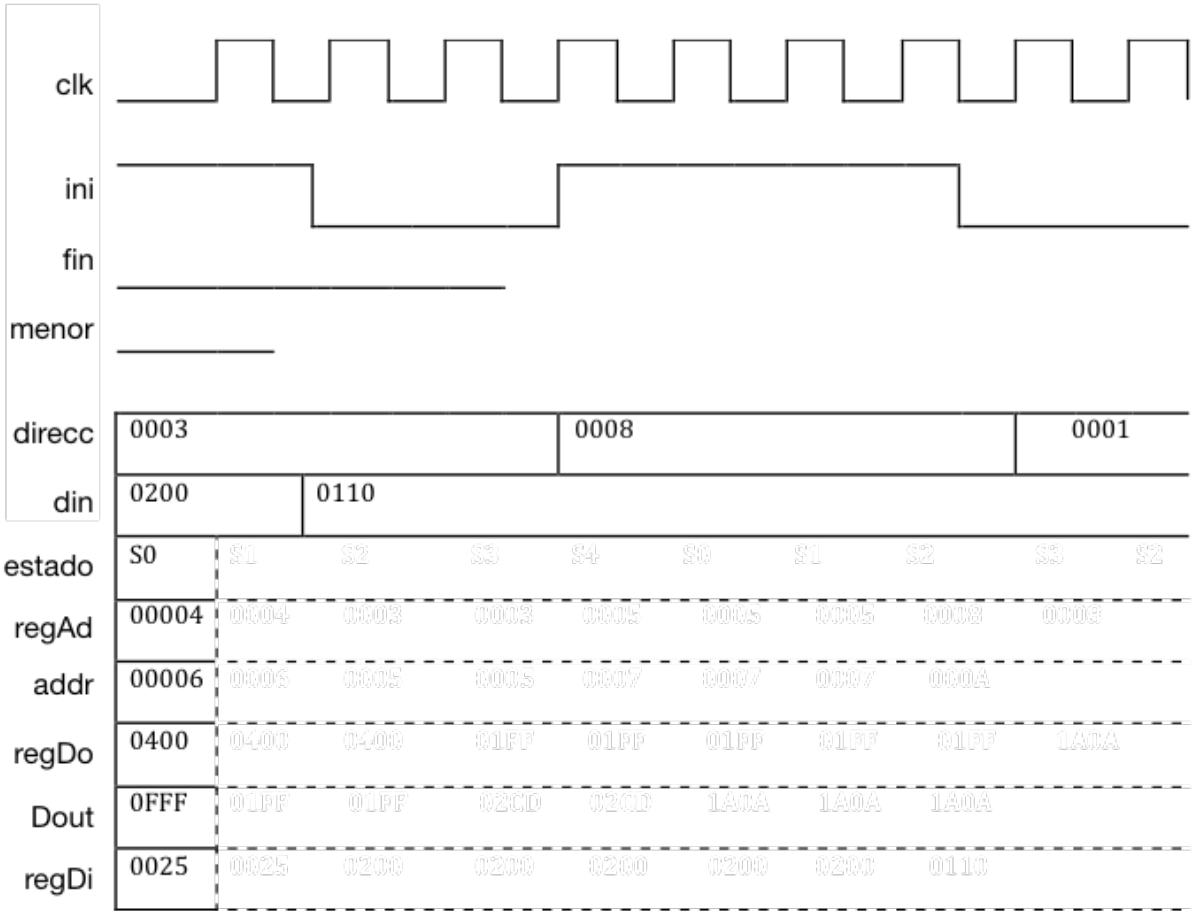


Estado	ad_ld	do_ld	di_ld	muxc	WE	fin
S0	0	0	0	0	0	0
S1	1	0	1	1	0	0
S2	0	1	0	0	0	0
S3	1	0	0	1	0	0
S4	0	0	0	0	1	1

Señales de control

Dirección	Dato
0x04	0x0123
0x05	0x02CD
0x06	0x01FF
0x07	0x1A0A
0x08	0x0C00
0x09	0x1000
0x0A	0x0110

Contenido de la memoria



TEC. COMPUT. // TEC. Y ORGA. COMPUT.

2º

GII — GIC

09/02/2015