



Práctica 5

- ASM complejo: gestión de memoria



Enunciado

- Diseñar la ruta de datos y la unidad de control (diagrama ASM y tabla de salidas) de un sistema algorítmico que dados n números ($n > 1$) de 4 bits en binario puro los ordene de mayor a menor según el algoritmo de ordenamiento de la burbuja.

```
i=0;Fin=0;
while i<n do
    j=0;
    while j  $n \leq 2$  do
        if  $Mem(j) \leq Mem(j + 1)$  then
             $aux \leq Mem(j)$ ;
             $Mem(j) \leq Mem(j+1)$ ;
             $Mem(j+1) \leq aux$ ;
        end
         $j = j + 1$ ;
    end
     $i = i + 1$ ;
end
Fin = 1;
```



Especificaciones

- Los números estarán almacenados a partir de la posición 0x00 de una memoria SRAM **síncrona**.
- La memoria será de doble puerto:
 - Se puede realizar de forma independiente dos operaciones de lectura
 - O una operación de escritura
- La definición de puertos es:
 - $din\{1\}$, bus de datos de entrada
 - $addr\{1,2\}$, bus de direcciones
 - $we\{1,2\}$, write enable
 - $dout\{1,2\}$, bus de datos de salida. Los puertos de salida conservan el valor leído hasta que se realice una nueva lectura.



Especificaciones

- El sistema comenzará a funcionar cuando se active la señal inicio y una vez ordenados los 16 números se volverá al estado inicial y se activará la señal de salida fin.
- La memoria RAM tendrá 32 palabras.
- En la ruta de datos se debe usar:
 - Una memoria SRAM de doble puerto
 - La memoria se añadirá al proyecto como un componente descrito en fichero aparte
 - Se encuentra en el fichero del campus mem_2.vhd
 - Contadores ascendentes/descendentes
 - Comparadores, registros y multiplexores



Especificaciones

- Las entradas del sistema son:
 - rst, clk e inicio (1 bit)
 - direccion (5 bits)
- La salidas son
 - fin (1 bit)
 - dato_debug(4 bits)
- Mientras la señal fin sea '1' (es decir, mientras el sistema se encuentre en el estado inicial):
 - dato_debug <= Mem(direccion)
- El resto del tiempo
 - dato_debug <= "0000";



Especificaciones

```
entity practica5 is
  port (  rst: in std_logic;
         clk: in std_logic;
         inicio: in std_logic;
         direccion: in std_logic_vector(4 downto 0);
         dato_debug: out std_logic_vector(3 downto 0)
        );
end practica5;
```



Memoria de doble puerto en VHDL

```
entity rams_2p is
  port (clk : in std_logic;
        we : in std_logic;
        addr1 : in std_logic_vector(4 downto 0);
        addr2 : in std_logic_vector(4 downto 0);
        di : in std_logic_vector(3 downto 0);
        do1 : out std_logic_vector(3 downto 0);
        do2 : out std_logic_vector(3 downto 0)
        );
end rams_2p;
```



Memoria de doble puerto en VHDL

```
architecture circuito of rams_2p is
  type ram_type is array (31 downto 0) of std_logic_vector (3 downto 0);
  signal RAM : ram_type;
begin

  puerto1: process (clk)
  begin
    if rising_edge(clk) then
      if we = '1' then
        RAM(conv_integer(addr1)) <= di;
      end if;
      do1 <= RAM(conv_integer(addr1));
    end if;
  end process puerto1;

  puerto2: process (clk)
  begin
    if rising_edge(clk) then
      do2 <= RAM(conv_integer(addr2));
    end if;
  end process puerto2;
end circuito;
```




Memoria de doble puerto en VHDL

■ CUIDADO:

- Los bloques de memoria de Xilinx son síncronos tanto para lectura como para escritura:
 - Para leer: la dirección del dato buscado tiene que estar en $\text{addr}\{1,2\}$ un ciclo antes de su utilización
 - Para escribir: el dato no estará efectivamente escrito en la memoria hasta que no se produzca un flanco de subida en el reloj (igual que ocurre con los registros)



Corrección de la práctica

- Los estudiantes deben realizar el diagrama ASM e implementarlo en VHDL.
- Para facilitar la corrección el sistema tendrá una entrada de depuración:
 - dato_debug, como se ha explicado en las especificaciones
 - El test-bench una vez haya finalizado el algoritmo recorrerá todas las posiciones de la memoria para comprobar su correcto funcionamiento.



Corrección de la práctica

- La práctica no será correcta si en la consola de simulación aparece ERROR
- Las preguntas del test se deben contestar en el momento en el que el test-bench no de error
- Las preguntas versarán sobre el diseño ASM