# EXAMEN DE SEPTIEMBRE DE 2006 I. T. Informática de Gestión

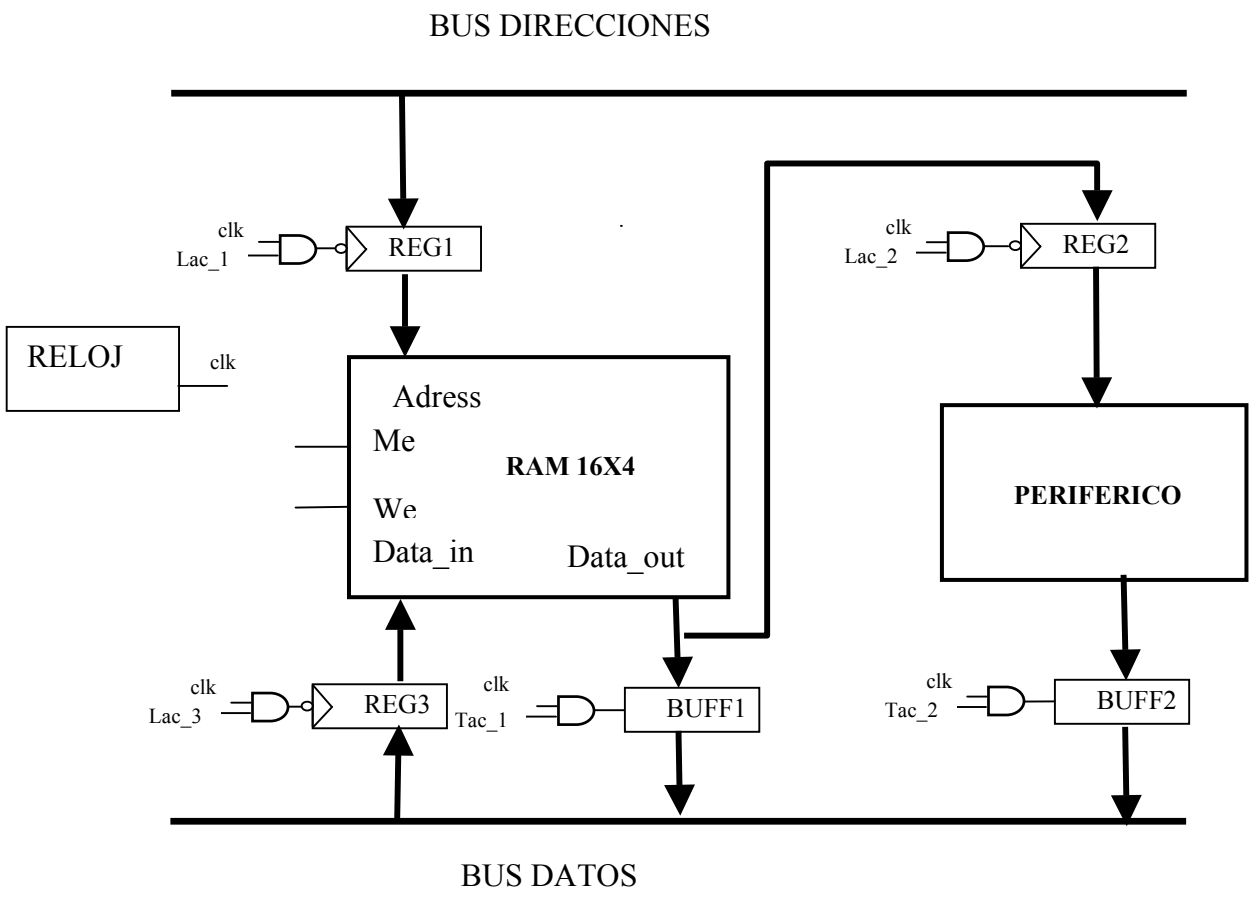


**Figura 1.**

Dado el circuito de la figura 1 del que se incluyen los modelos VHDL de todos los elementos.

ME activa a nivel alto
WE = 1 se realiza escritura en memoria; WE = 0 se realiza lectura de la memoria

La señal de carga de los registros es activa por flanco de bajada y el control de los buffers es activo a nivel alto.

**1**.- Realizar el modelo VHDL estructural del circuito de la figura 1.
**(2,5 puntos)**

**2**.- Calcular la frecuencia máxima de reloj para realizar la transferencia de un dato que se encuentra en la dirección de memoria indicada por el bus de direcciones al periférico.
**(2,5 puntos)**

**3**.- Realizar el programa test-bench que permita probar la transferencia de memoria al periférico propuesta en el apartado anterior, cargando en el

periférico el dato contenido en la dirección de memoria 7 y luego el de la dirección 8.
Dar los valores adecuados a las señales de entrada necesarias.
**(2,5 puntos)**

**4**.- Realizar el esquema y el modelo VHDL estructural de una memoria RAM de 32 palabras de 4 bits usando como componentes la RAM de 16 palabras de 4 bits de la figura 1 y las puertas lógicas necesarias.
**(2,5 puntos)**

**MODELOS VHDL DE LOS COMPONENTES:**

```
ENTITY reloj IS
        GENERIC(periodo:TIME:=?? ns);
        PORT(reloj:OUT BIT:='0');
END reloj;

ARCHITECTURE comportamiento OF reloj IS
...
END comportamiento;
```

```
ENTITY and2 IS
        GENERIC (retardo: TIME := 2 ns);
        PORT (e1, e2: IN BIT; sal: OUT BIT);
END and2;

ARCHITECTURE comportamiento OF and2 IS
...
END comportamiento;
```

```
USE WORK.arq_pack.ALL;
ENTITY registro4 IS
        GENERIC(retardo_carga:TIME:=10 ns);
        PORT(entrada:IN vector_bus; control:IN BIT:='0';
        salida:OUT vector_bus:=('0','0','0','0'));
END registro4;

ARCHITECTURE comportamiento OF registro4 IS
...
END comportamiento;
```

```
USE WORK.arq_pack.ALL;
ENTITY buffer4 IS
        GENERIC(retardo_transferencia:TIME:=15 ns;retardo_Z:TIME:=10 ns);
        PORT(entrada:IN vector_bus; control:IN BIT:='0';
        salida:OUT vector_bus);
END buffer4;

ARCHITECTURE comportamiento OF buffer4 IS
...
END comportamiento;
```

```
USE WORK.arq_pack.ALL;
ENTITY periferico4 IS
        PORT(entrada:IN vector_bus; salida:OUT vector_bus:=('0','0','0','0'));
END periferico4;

ARCHITECTURE comportamiento OF periferico4 IS
...
END comportamiento;
```

```
USE WORK.arq_pack.ALL;
--se incluye el paquete ram_pack (procedimiento de conversión de lógico a entero,
tipos y señal resuelta)

ENTITY ram_exa IS
        --retardo_salida: retardo de propagación de las celdas de memoria a la salida
        --retardo_Z: retardo de propagación del estado Z a la salida
        GENERIC(retardo_salida:TIME:=30 ns; retardo_Z:TIME:=15 ns);
        PORT(DATA_IN:IN vector_bus:=('Z','Z','Z','Z');
        ADDRESS:IN BIT_VECTOR(3 DOWNTO 0):=X"0";
        ME,WE:IN BIT:='0';
        DATA_OUT:OUT vector_bus:=('Z','Z','Z','Z'));
END ram_exa;

ARCHITECTURE comportamiento OF ram_exa IS
...
END comportamiento;
```

```
PACKAGE arq_pack IS
        TYPE tri_estado IS ('0', '1', 'Z');
        TYPE vector_bus IS ARRAY (3 DOWNTO 0) OF tri_estado;
        TYPE array_vector_bus IS ARRAY (INTEGER RANGE <>) OF vector_bus;
        FUNCTION resolucion (entrada: array_vector_bus) RETURN vector_bus;
        --la señal del bus será la resuelta
        SUBTYPE bus_resuelto IS resolucion vector_bus;
        PROCEDURE logico_entero(VARIABLE vector:IN BIT_VECTOR;
                                VARIABLE entero:OUT INTEGER);
        PROCEDURE entero_logico(VARIABLE entero:IN INTEGER;
                                VARIABLE vector:OUT BIT_VECTOR);
END arq_pack;
PACKAGE BODY arq_pack IS
        .....
END arq_pack;
```