

# LABORATORIO DE ARQUITECTURA DE COMPUTADORES.

## I. T. I. SISTEMAS / GESTIÓN

### GUÍA DEL ALUMNO

#### *Práctica 3: Memorias*

##### **Objetivos**

- Comprender qué es y cómo funciona un banco de registros
- Comprender cómo se realiza una expansión de memoria

##### **Método de trabajo**

Lee cuidadosamente las indicaciones de la guía, realizando los pasos que se indican. La guía tiene como propósito que **TÚ** te asegures de que comprendes lo que va sucediendo mientras realizas lo que te indica que hagas.

Para todas las sesiones, necesitaremos textos de referencia del lenguaje para realizar programas o para comprender lo que ocurre durante las simulaciones. Por lo tanto se recomienda traer libros, apuntes y cualquier bibliografía de VHDL que sirva para hacer consultas. En el apartado “bibliografía” se proporcionan algunas referencias.

También traer un **disquete** o una **memoria USB** para llevar copia de los ficheros que se generen y se deseen conservar ya que, al tratarse de una máquina compartida, no hay garantías de que más tarde encontremos los trabajos que dejamos hoy. Pero dado que V-System no trabaja bien con los disquetes siempre habrá **que copiar de/al disco duro y trabajar en él.**

##### **Bibliografía**

- “VHDL: Lenguaje para síntesis y modelado de circuitos”. F.Pardo y J.A.Boluda. Ed. Ra-Ma,1999.
- Lluís Terés y otros “VHDL: Lenguaje estándar de diseño electrónico” Mc Graw Hill, 1998
- Buscar en la WEB: Google encuentra referencias y pueden encontrarse libros en pdf

##### **Introducción**

El **banco de registros** es un elemento fundamental en la arquitectura von Neumann. Permite trabajar a gran velocidad, dado que se encuentra en la misma CPU, no necesitando adaptar su velocidad a la de los elementos externos a la misma.

Un banco de registros está formado por un número  $n$  de registros que son capaces de ser seleccionados mediante una señal de control para ser leídos o escritos. Asimismo, el banco de registros puede tener varios puertos de salida y varios de entrada, permitiendo de esta forma la lectura o la escritura simultánea de varios registros.

La figura siguiente muestra un banco de registros que cuenta con un puerto de salida y un puerto de entrada.

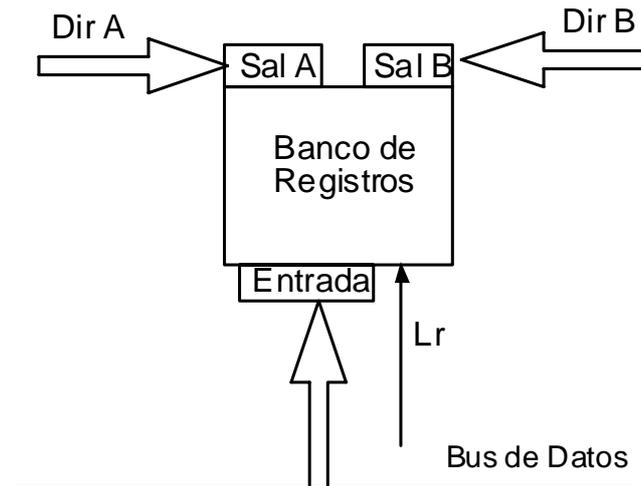


Figura 1: banco de registros

La lectura de los registros requieren las señales de control (DIRA y DIRB) y la escritura de un único registro requiere seleccionar el registro a escribir mediante DIRA y activar la señal de carga LR (por flanco de bajada)

### Actividad 1.

1. Modelar con ARRAYS el banco de registros de la figura 1, para un total de 8 registros de 16 bits a partir de las modificaciones que creas necesarias al fichero de la memoria RAM (ram7489.vhd)
2. Realizar el test bench correspondiente para comprobar su correcto funcionamiento (carga de un valor en el registro 6, carga de un valor en el registro 2, volcado del registro 6 y volcado del registro 2)

## Segunda parte

La **memoria principal** es otro de los elementos fundamentales de un computador, dado que el modelo de ejecución es de programa almacenado. La memoria es un elemento más lento que los registros del procesador, motivo por el que los accesos a memoria afectan al rendimiento del computador.

Para mejorar el rendimiento de la memoria se puede mejorar la tecnología, aumentar el tamaño de palabra o hacer que se acceda a varios módulos de memoria a la vez, es decir, realizando un entrelazado de memoria.

Como elemento intermedio entre los registros de la CPU y la memoria los computadores actuales incorporan una memoria caché con una velocidad cercana a la de los registros y que permiten que se aumente el rendimiento del sistema de memoria.

La siguiente figura muestra el sistema de memoria de un computador elemental que consta de un registro de direcciones y un buffer para volcar el contenido de la memoria al bus de datos. Las señales involucradas son LR (por flanco de bajada) para almacenar una dirección en el registro de direcciones. TMEM (por nivel) para volcar el contenido de la memoria al bus de datos, y las señales activas por nivel MEM (inicio de ciclo de memoria), RD (lectura) y WR (escritura).

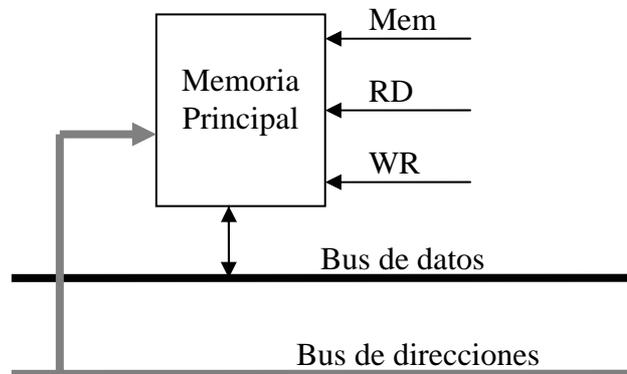


Figura 2. Sistema de memoria

El código siguiente presenta la codificación de la memoria RAM 7489, una memoria de 16 direcciones y un bus de 4 bits, que se muestra en la figura 3.

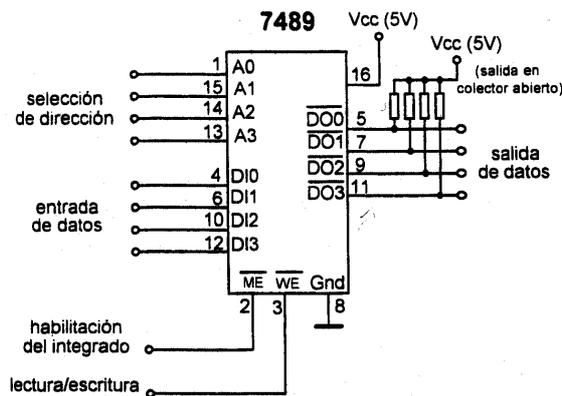


Figura 3. Esquema de la RAM 7489

**Fuente: ram7489.vhd**

```

----- Programa ram7489.vhd -----
USE WORK.arq_pack.ALL;
--se incluye el paquete ram_pack (procedimiento de conversión de lógico a entero,
tipos y señal resuelta)

ENTITY ram_7489 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:=300 ns; retardo_Z:TIME:=300 ns);
  PORT(DATA_IN:IN BIT_VECTOR(3 DOWNTO 0):=X"0";
        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_7489;

ARCHITECTURE comportamiento OF ram_7489 IS
BEGIN
    PROCESS
    -----
    --declaracion de tipos y variables

    TYPE memoria IS ARRAY(15 DOWNT0 0) OF BIT_VECTOR(3 DOWNT0 0);
    VARIABLE palabra:memoria;
    VARIABLE comienzo:BOOLEAN:=true;
    VARIABLE dir_logica:BIT_VECTOR (3 DOWNT0 0);
    VARIABLE dir_entero:INTEGER;
    -----
    --comienza el proceso
    --se despierta con cambios en ADDRESS, DATA_IN, ME y WE

    BEGIN
    WAIT ON ADDRESS, DATA_IN, ME, WE;
    -----
    --inicialmente todas las posiciones de la memoria estan a '0'

    IF comienzo THEN
        FOR i IN palabra'RANGE LOOP
            palabra(i):=('0','0','0','0');
        END LOOP;
        comienzo:=false;
    END IF;
    -----
    --chip de memoria activo

    IF (ME='0') THEN      --la memoria está activada
    -----
    --se lee la dirección
    --(no hay retardo en direccionar las celdas de memoria)

        dir_logica:=ADDRESS;
        logico_entero(dir_logica, dir_entero);
        --se convierte el valor lógico de la dirección a entero
        --se usa el procedimiento declarado en el paquete ram_pack
    -----

        IF (WE='1') THEN      --comienza el ciclo de lectura
            FOR i IN 0 TO 3 LOOP
                IF palabra(dir_entero)(i)='0'THEN
                    DATA_OUT(i)<=TRANSPORT'0'AFTER retardo_salida;
                ELSE DATA_OUT(i)<=TRANSPORT'1'AFTER retardo_salida;
                END IF;
            END LOOP;
            -----

            ELSE      --comienza el ciclo de escritura
                palabra(dir_entero):=DATA_IN;

```

```
--(no hay retardo en escribir en las celdas)
-----
--la salida permanece en estado de alta impedancia

FOR i IN 0 TO 3 LOOP
    DATA_OUT(i)<=TRANSPORT 'Z' AFTER retardo_Z;
    --propagación de WE al estado Z (300 ns)
END LOOP;
END IF;
-----
--chip de memoria inactivo

ELSE      --la memoria no está activada

    FOR i IN 0 TO 3 LOOP
        DATA_OUT(i)<=TRANSPORT 'Z' AFTER retardo_Z;
        --propagación de ME al estado Z (300 ns)
    END LOOP;
END IF;
-----
END PROCESS;
END comportamiento;
-----
```

## Actividad 2.

1. Modificar el paquete en el que se define el modelo de la RAM 7489 para que almacene 16 bits en lugar de 4 (para simplificar modificar solamente la anchura de la palabra, y no hacerlo de manera estructural)
2. A partir del modelo de la RAM 7489 y de los elementos que creas necesarios modelar un sistema de memoria de 128x16 que se corresponda con la figura 2, en el que el Bus de Datos es de entrada salida en el módulo
3. Realizar el test bench correspondiente a la escritura en la dirección 5 del valor E000h. Poner el Bus de Datos en Alta impedancia. Escribir en la dirección 46 el valor 9000h. Poner el Bus de Datos en Alta impedancia. Leer la dirección 5. Leer la dirección 46.