

Memorias RAM Modeladas en Lógica Programable.

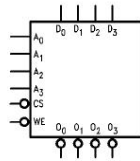
Juan Carlos Herrera Lozada, CIDETEC IPN, 2008. jlozada@ipn.mx

Introducción al caso de diseño

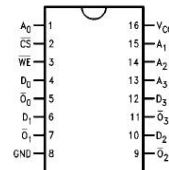
Partiremos del análisis de una memoria RAM clásica, por ejemplo el dispositivo 74F189, cuya hoja de especificaciones indica lo siguiente.

Order Number	Package Number	Package Description
74F189PC	N16E	16-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

Logic Symbols



Connection Diagram



Consta de 16 localidades, de cuatro bits cada una, para una capacidad total de 64 bits. Con cuatro líneas de direcciones (A_3, A_2, A_1, A_0) es posible acceder a las 16 localidades.

El funcionamiento implica el uso del pin CS (*Chip Select*) con lógica negativa para habilitar la memoria y permitir la escritura o la lectura de datos. Cuando CS tenga asignado un valor lógico alto, las salidas de la RAM (O_3, O_2, O_1, O_0) presentan tercer estado debido a que el dispositivo no está habilitado (*Inhibit*), tal y como se aprecia en la tabla correspondiente, en la figura siguiente.

Unit Loading/Fan Out

Pin Names	Description	U.L.	Input I_{IH}/I_{IL}
		HIGH/LOW	Output I_{OH}/I_{OL}
A_0-A_3	Address Inputs	1.0/1.0	20 μ A/-0.6 mA
\overline{CS}	Chip Select Input (Active LOW)	1.0/1.0	20 μ A/-1.2 mA
\overline{WE}	Write Enable Input (Active LOW)	1.0/1.0	20 μ A/-0.6 mA
D_0-D_3	Data Inputs	1.0/1.0	20 μ A/-0.6 mA
$\overline{O}_0-\overline{O}_3$	Inverted Data Outputs	150/40 (33.3)	-3.0 mA/24 mA (20 mA)

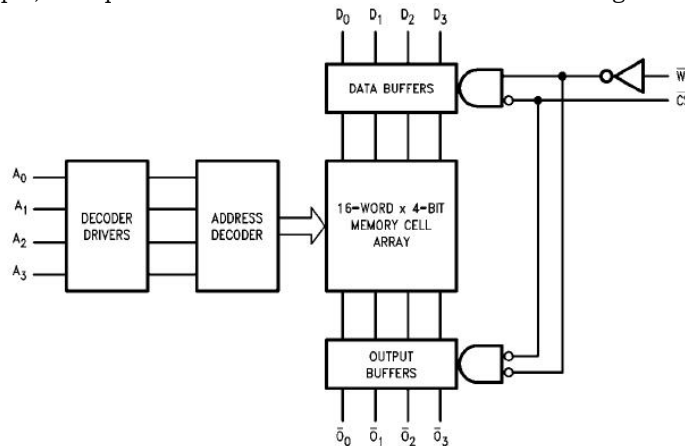
Function Table

Inputs		Operation	Condition of Outputs
\overline{CS}	\overline{WE}		
L	L	Write	High Impedance
L	H	Read	Complement of Stored Data
H	X	Inhibit	High Impedance

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

El pin WE (*Write Enable*) se activa con lógica negativa para permitir escribir un dato a la memoria a la vez que las salidas toman un tercer estado. Cuando WE tiene lógica positiva, es posible leer un dato a través de los cuatro pines de salida del dispositivo.

De un modo simple, la arquitectura de la memoria RAM consta de los siguientes bloques.



Observa que el pin CS puede inhibir en cualquier momento el funcionamiento de la RAM, es decir, deshabilita el buffers de entrada y de salida al mismo tiempo. Cuando CS tenga un "0" lógico es cuando la memoria trabaja de manera normal, por lo que con WE en "0" lógico se activa sólo el buffer de entrada y se desactiva el de salida (*tercer estado*).

Un algoritmo de diseño para analizar el funcionamiento sería: Se mantiene CS desactivado (nivel bajo) y se coloca una dirección, por ejemplo A3=1, A2=0, A1=1, A0=0, lo que permite acceder a la localidad número 10. Se activa WE con un nivel bajo para escribir un dato. Se coloca el dato D3=1, D2=1, D1=1, D0=0, almacenando 1110 en la localidad 10. Se activa CS con un nivel alto para permitir la operación descrita. Debido a que se está trabajando en el modo de escritura, las salidas mostrarán tercer estado.

Para leer un dato, se deshabilita CS y se escribe la dirección de la localidad que se desea leer. Posteriormente se le asigna a WE un nivel lógico alto y se procede a habilitar CS para validar la operación descrita. En el modo de lectura, las salidas de la memoria O3,O2,O1 y O0 muestran un dato almacenado.

El diagrama a bloques de la arquitectura se puede modularizar para diseñar el dispositivo con PLDs. Dependiendo de la arquitectura PLD a utilizar, así como del HDL, se puede modelar una solución completa dentro de un solo dispositivo. Específicamente, la arquitectura GAL permite buffers de tercer estado sólo en sus macroceldas (recordando que se encuentran sólo en sus salidas), lo que indica que no es posible deshabilitar entradas de datos, sino sólo las salidas del dispositivo. Para un ejercicio de diseño sencillo en una primera aproximación, utilizaremos tres dispositivos GAL: Uno para el buffer tri-state de entrada, otro para el buffer tri-state de salida, y el último para los registros.

Los códigos escritos en ABEL HDL que modelan los buffers con tercer estado de cuatro bits, para la entrada y la salida, son los siguientes:

```
MODULE buf3sti
cs, we, d3, d2, d1, d0 pin;
sb3, sb2, sb1, sb0 pin istype 'com';
dato=[d3, d2, d1, d0];
sal_buf=[sb3, sb2, sb1, sb0];

equations
sal_buf.oe= (!cs & !we);
sal_buf=dato;

END
```

```
MODULE buf3sto
cs, we, d3, d2, d1, d0 pin;
sb3, sb2, sb1, sb0 pin istype 'com';
dato=[d3, d2, d1, d0];
sal_buf=[sb3, sb2, sb1, sb0];

equations
sal_buf.oe= (!cs & we);
sal_buf=dato;

END
```

Write Enable es la única diferencia entre ambos buffers debido a que WE se active en nivel bajo cuando se escribe un dato y se active en un nivel alto cuando se lee el dato.

Para el módulo de registros es posible considerar el siguiente código. Una RAM es asíncrona por naturaleza, es decir, no requiere una señal de reloj que sincronice los cambios, pero sí requiere una señal de habilitación que permita almacenar un dato en los registros asíncronos (latches). El tipo declarado en las salidas (*reg_g*) hace alusión a esperar un cambio derivado de una señal entre compuertas y no de un reloj.

Nótese que el código listado a continuación se trata sólo de un módulo que tiene una sola línea de dirección (*a0*), para controlar dos únicos registros de 4 bits cada uno. Habría que adecuarla para trabajar en conjunto con los buffers anteriores.

```
MODULE regs
a0, d2, d1, d0 pin;
b3r0, b2r0, b1r0, b0r0 node istype 'reg_g,dc';
b3r1, b2r1, b1r1, b0r1 node istype 'reg_g,dc';
sr3, sr2, sr1, sr0 pin istype 'com,dc';
dato=[d3, d2, d1, d0];
direcc=[a0];
reg0=[b3r0, b2r0, b1r0, b0r0];
reg1=[b3r1, b2r1, b1r1, b0r1];
sal_reg=[sr3, sr2, sr1, sr0];

equations
when direcc==0 then {reg0:=dato; reg1:=reg1; sal_reg:=reg0;}
else {reg1:=dato; reg0:=reg0; sal_reg:=reg1;}

END
```