



# Memorias ROM

Arquitectura de Computadoras - Práctico 4

# Funciones lógicas arbitrarias

Las memorias ROM almacenan arreglos de palabras => tablas de verdad. Para una función  $f(x_m, x_{m-1}, \dots, x_0) = (y_n, y_{n-1}, \dots, y_0)$ :

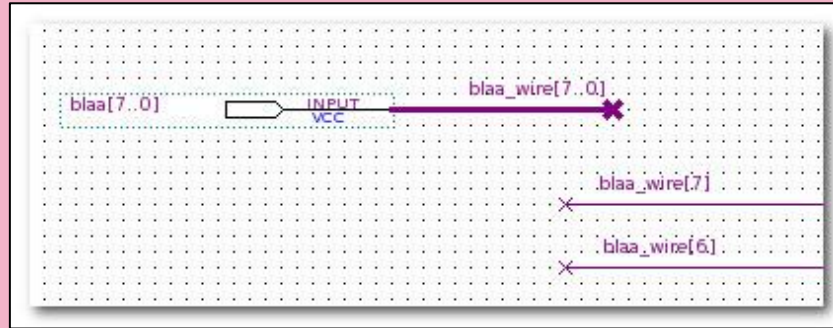
Entradas								Salidas				
Am-1	Am-2	Am-3	....	....	A2	A1	A0	Dn-1	Dn-2	...	D1	D0
0	0	0	....	....	0	0	0	0	0	....	0	0
0	0	0	....	....	0	0	1	1	0	....	1	0
0	0	0	....	....	0	1	0	0	0	....	0	0
....	....	....	....	....	....	....	....	....	....	....	....	....
....	....	....	....	....	....	....	....	....	....	....	....	....
1	1	1	....	....	1	0	1	1	1	....	1	1
1	1	1	....	....	1	1	0	0	1	....	0	1
1	1	1	....	....	1	1	1	1	1	....	1	1

# Capacidad y organización

Tamaño (capacidad): Cantidad de bits que puede almacenar la ROM (preferentemente expresado en bytes)

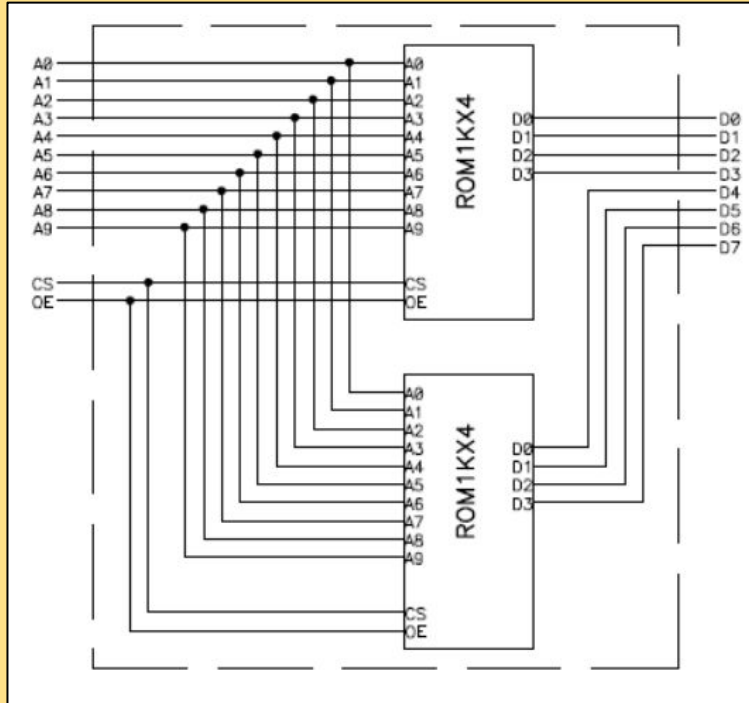
Organización: Cuántas palabras de cuántos bits forman la ROM (Ej: 128K palabras de 24 bits)

Notación:  
buses



# Arreglos de memorias

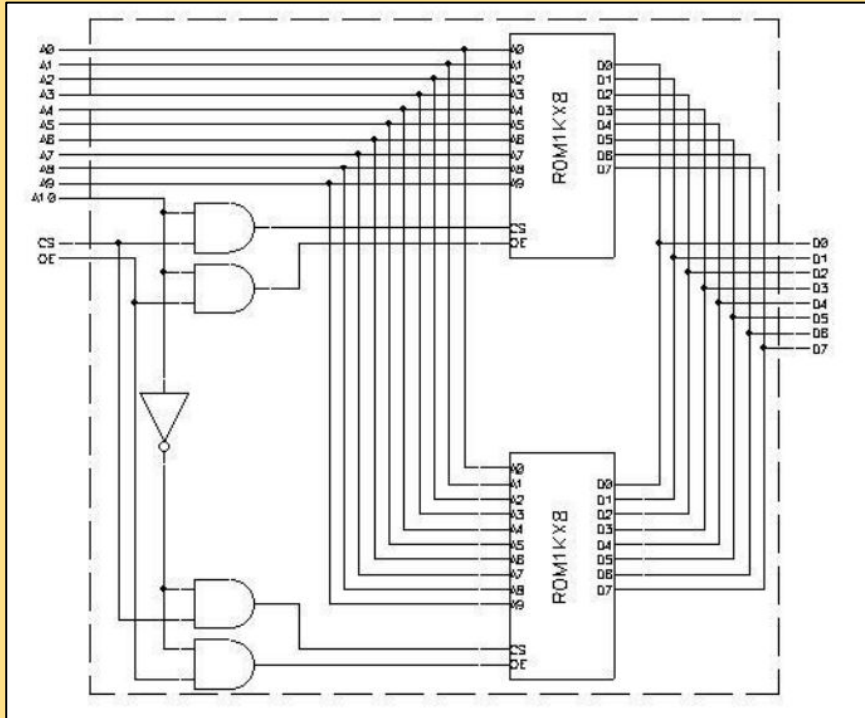
- Ejemplo 1: memoria de mayor tamaño de palabra



- No olvidar CS y OE

# Arreglos de memorias

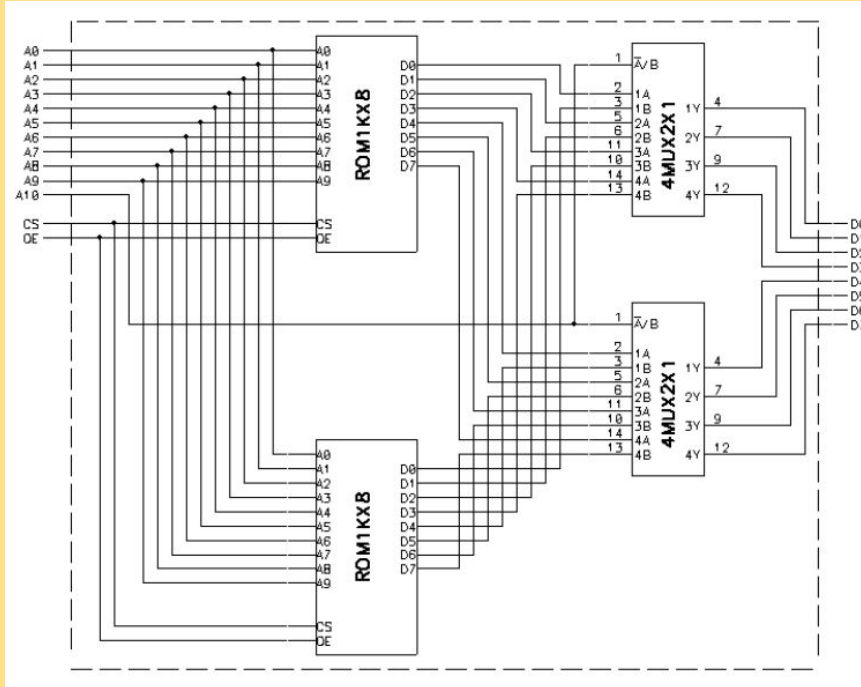
- Ejemplo 2.a): memoria de mayor cantidad de palabras



- No olvidar CS y OE
- Verificar lógica de OR cableado

# Arreglos de memorias

- Ejemplo 2.b): memoria de mayor cantidad de palabras



- No olvidar CS y OE
- ¿Qué problema tiene la conexión de OE en este ejemplo? ¿Cómo se corrige?

# Carga inicial de ROMs

En muchos ejercicios se solicita la elaboración de un programa que cargue el contenido inicial de una ROM. Es decir: que indique para cada entrada de la función (dirección), cuál es su correspondiente salida (palabra de memoria).

```
#define TAM_ROM XXXX // De qué tamaño debe ser la ROM?
type ROM[TAM_ROM] // De qué tipo debe estar definida la ROM?
void cargaRom() {
    for (int i = 0; i < TAM_ROM; i++){
        // cargar palabra i-ésima de la ROM
    }
}
```

# Arreglos de memorias

- Ejercicio 3): memoria de mayor cantidad de palabras a partir de una con mayor tamaño de palabras.
- Construir una ROM de 16K x 16 a partir de otra ROM de 8K x 32 (no olvidar CS ni OE).



# Capacidad y organización

Se desea utilizar una ROM para implementar un sumador simple (sin acarreo) de 8 bits.

Indicar tamaño y organización de la ROM necesaria.

# Carga inicial de ROMs

## Ejercicio

Se desea usar una ROM para implementar una función shift (desplazamiento) aritmético hacia la derecha, que recibe dos entradas: un byte a ser desplazado y tres bits que indican el número de bits a desplazar. En el shift aritmético, se insertan 1's del lado izquierdo si el número es negativo y 0's en caso contrario.

Se pide: Escribir un programa que genere el contenido de la ROM