





TEMA 04

LÓGICA COMBINACIONAL PROGRAMABLE

TEMA 4: LÓGICA COMBINACIONAL PROGRAMABLE

- Contexto
- Conocimiento Previo Necesario
- Objetivos del Tema
- Guía de Estudio
- Contenido del Tema
- 4.1. Procesamiento Digital de la Información
- 4.2. Memorias PROM, EPROM, EEPROM y FLASH
- 4.3. Transistores de Puerta Flotante (FAMOS) y Mecanismos de Borrado
 - 4.3.1. Borrado de EPROMs
 - 4.3.2. Borrado de las EEPROMs
 - 4.3.3. Borrado de las Memorias FLASH
- 4.4. Organización Interna y Ejemplos de EEPROM y FLASH
 - 4.4.1. EPROMs
 - 4.4.2. EEPROM
 - 4.4.3. FLASH
- 4.5. PALs y PLAs
- 4.6. Configuraciones de Salida
- 4.7. Nomenclatura y Ejemplo de Circuitos PAL
- 4.8. Problemas
- Preparación de la Evaluación
- Referencias Bibliográficas

- 
- 
- Objetivo 1:** *Conocer la estructura interna de las memorias de sólo lectura (ROM, PROM, EPROM, EEPROM y FLASH). Saber cómo pueden usarse para distintos tipos de aplicaciones.*
- Objetivo 2:** *Conocer las arquitecturas PAL y PLA y saberlas usar en el diseño en lógica combinacional.*

7.1 Procesamiento Digital de la Información

a_3	a_2	a_1	a_0	f
0	0	0	0	0
0	0	0	1	$\overline{x_1} \overline{x_0}$
0	0	1	0	$\overline{x_1} x_0$
0	0	1	1	$\overline{x_1} x_1$
0	1	0	0	$x_1 \overline{x_0}$
0	1	0	1	$\overline{x_0}$
0	1	1	0	$x_1 \oplus x_0$
0	1	1	1	$\overline{x_1} + \overline{x_0}$
1	0	0	0	$x_0 x_1$
1	0	0	1	$x_1 \oplus x_0$
1	0	1	0	x_0
1	0	1	1	$x_0 + \overline{x_1}$
1	1	0	0	x_1
1	1	0	1	$x_1 + \overline{x_0}$
1	1	1	0	$x_1 + x_0$
1	1	1	1	1

Figura 7.1. Tabla que representa las 16 funciones lógicas de dos variables, en función del valor de los coeficientes (a_0, \dots, a_3) de su representación por minterms.

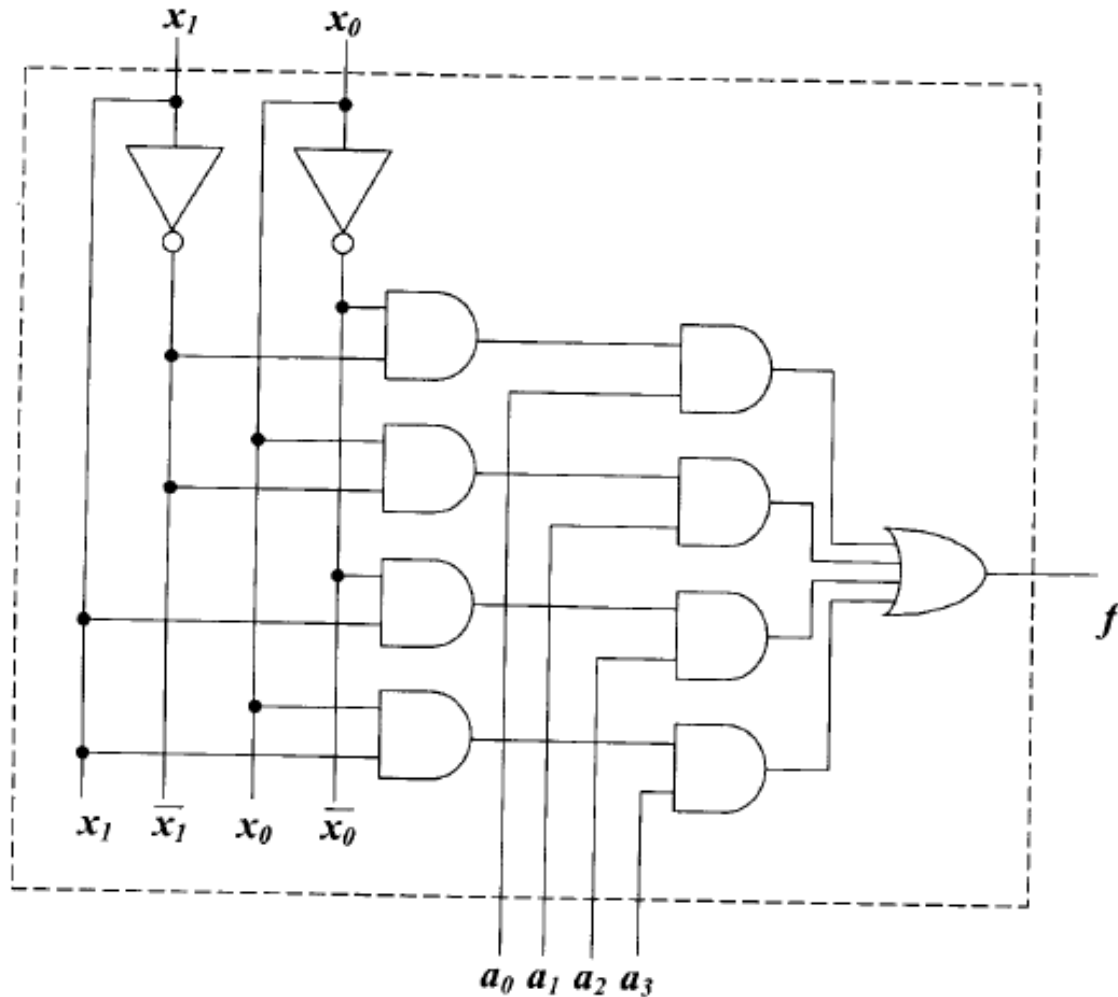


Figura 7.2. Función lógica universal de dos variables. Cambiando el valor de los cuatro términos de control (a_0, \dots, a_3) recorreremos las 16 funciones posibles.

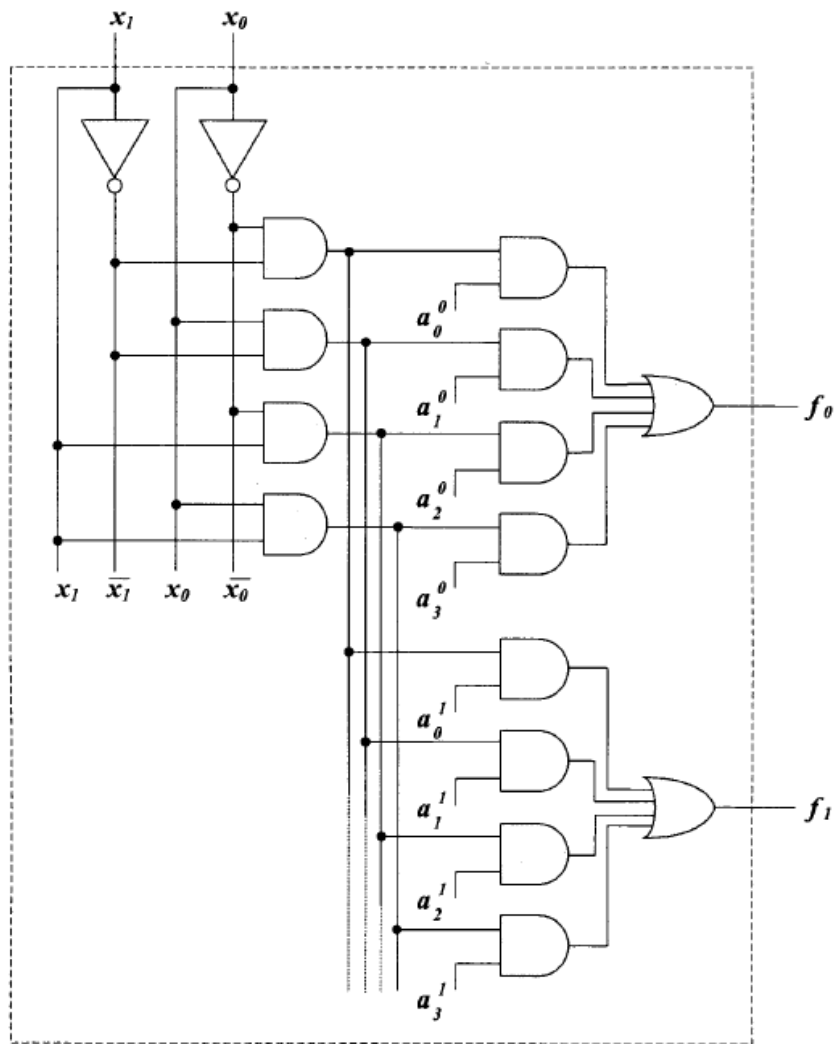


Figura 7.3. Uso de los mismos términos mínimos para obtener a la vez dos funciones universales de 2 variables (f_0, f_1).

- PLD = Dispositivo lógico programable
- Compuestos por dos matrices de líneas y columnas conectadas a un grupo de puestas AND por un lado y a otro grupo de puertas OR por otro.

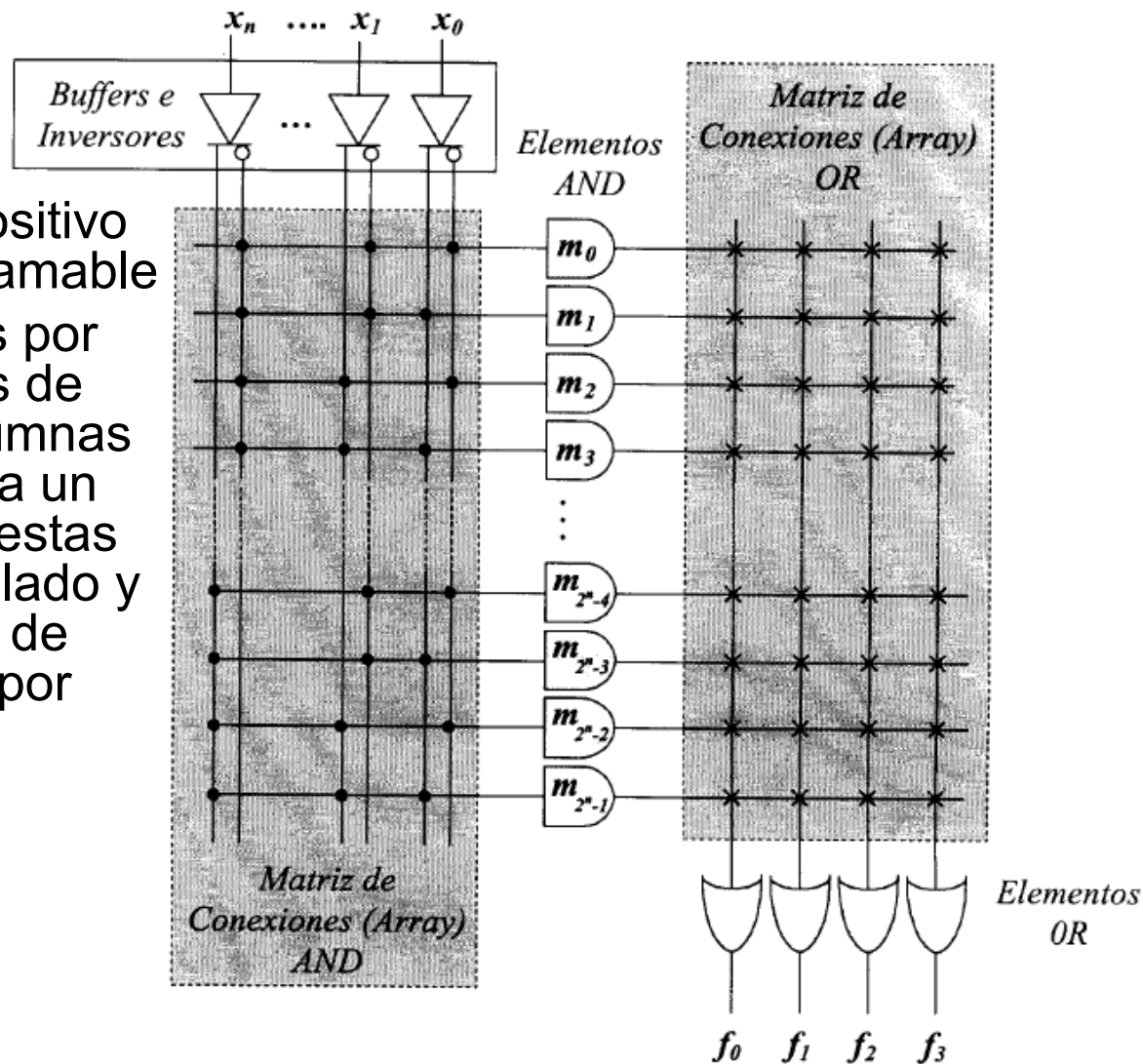
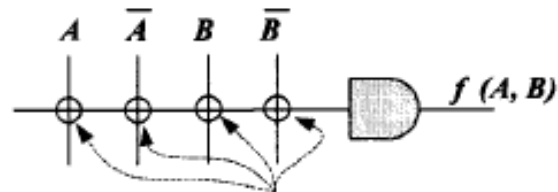
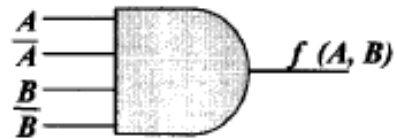


Figura 7.4. Estructura general de la lógica programable (PLDs) con las dos matrices (AND y OR) de conexiones programables y las dos ristas de elementos AND y OR.

- En función de donde esté situada la matriz programable se clasifican en:

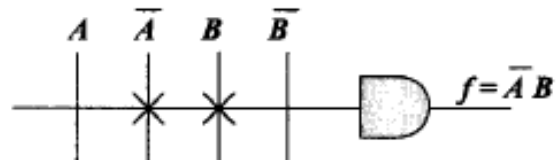
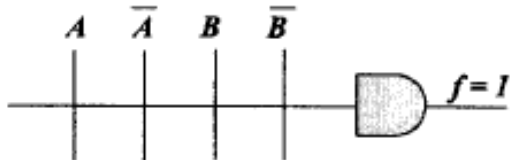
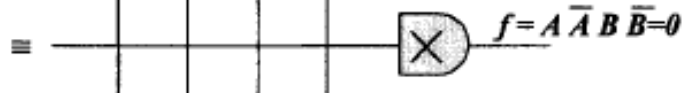
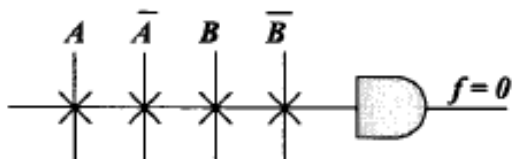
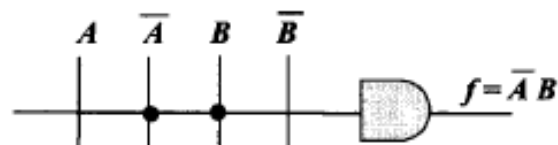
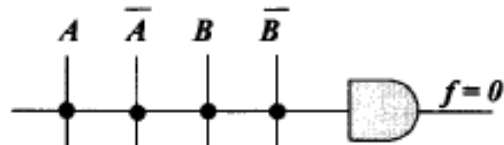
<i>Arquitectura</i>	<i>Matriz AND</i>	<i>Matriz OR</i>
PROM	<i>fija</i>	<i>programable</i>
PAL	<i>programable</i>	<i>fija</i>
PLA	<i>programable</i>	<i>programable</i>

Puerta AND con entrada múltiple



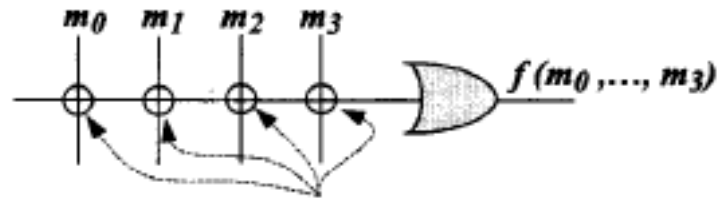
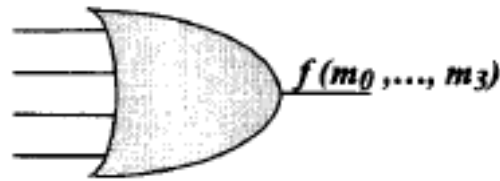
Se especifica el estado funcional de la conexión

Ejemplos



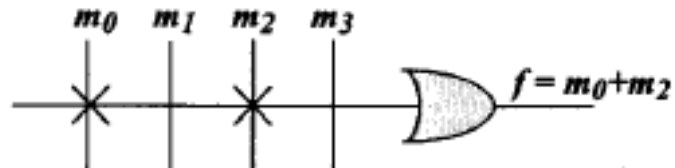
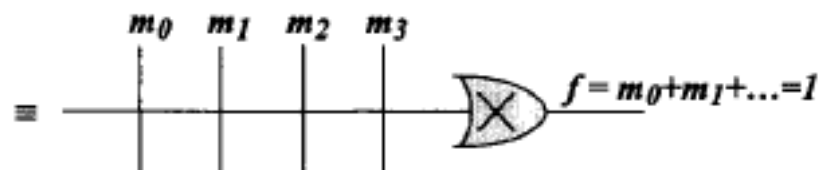
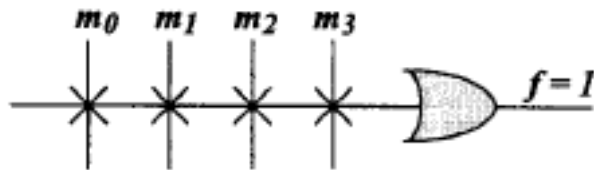
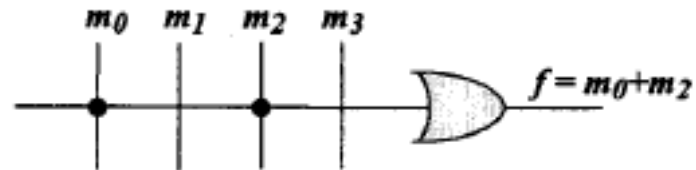
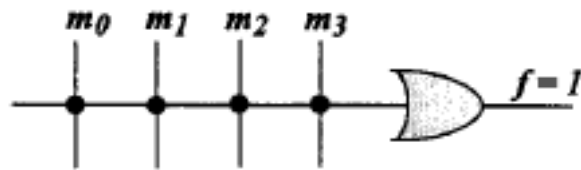
Conexión Permanente	Conexión Programable Intacta	Conexión Programable Fundida

Puerta OR con entrada múltiple (términos producto, m_0, m_1, m_2, m_3)



Se especifica el estado funcional de la conexión

Ejemplos



Conexión Permanente	Conexión Programable Intacta	Conexión Programable Fundida

Ejercicio: Representar con el criterio de representación de PLDs resumido en las figuras 7.6 y 7.7 el circuito de la figura 7.2 con $a_0 = a_3 = 1$ y $a_1 = a_2 = 0$

Solución:

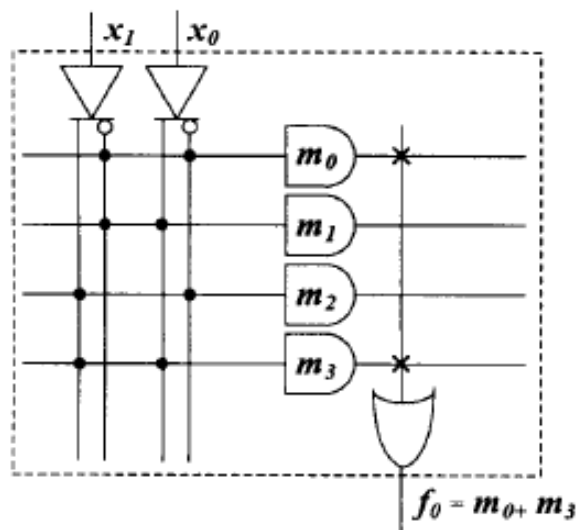


Figura 7.8.

Ejercicio: Repetir el ejercicio anterior para el circuito de la figura 7.3 con $a_0^0 = a_1^0 = 1$, $a_2^0 = a_2^1 = 0$, $a_0^1 = a_3^1 = 1$, $a_1^1 = a_2^1 = 0$.

Solución

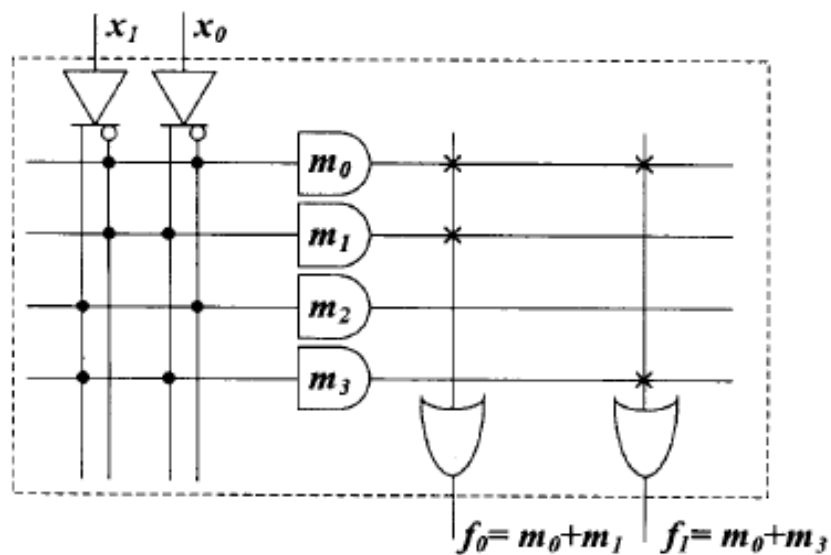


Figura 7.9.

7.2 Memorias PROM, EPROM, EEPROM Y FLASH

Dependiendo del mecanismo usado para borrar tenemos dos nuevos tipos de memorias no volátiles: EPROM y EEPROM (ó E²PROM).

Las EPROM (Erasable PROM) se borran usando luz ultravioleta y se programan eléctricamente y las EEPROM (Electrically Erasable PROM) se borran y se graban eléctricamente, por lo que pueden realizarse ambos procesos sin necesidad de extraer el circuito de memoria del resto de los circuitos que constituían la aplicación.

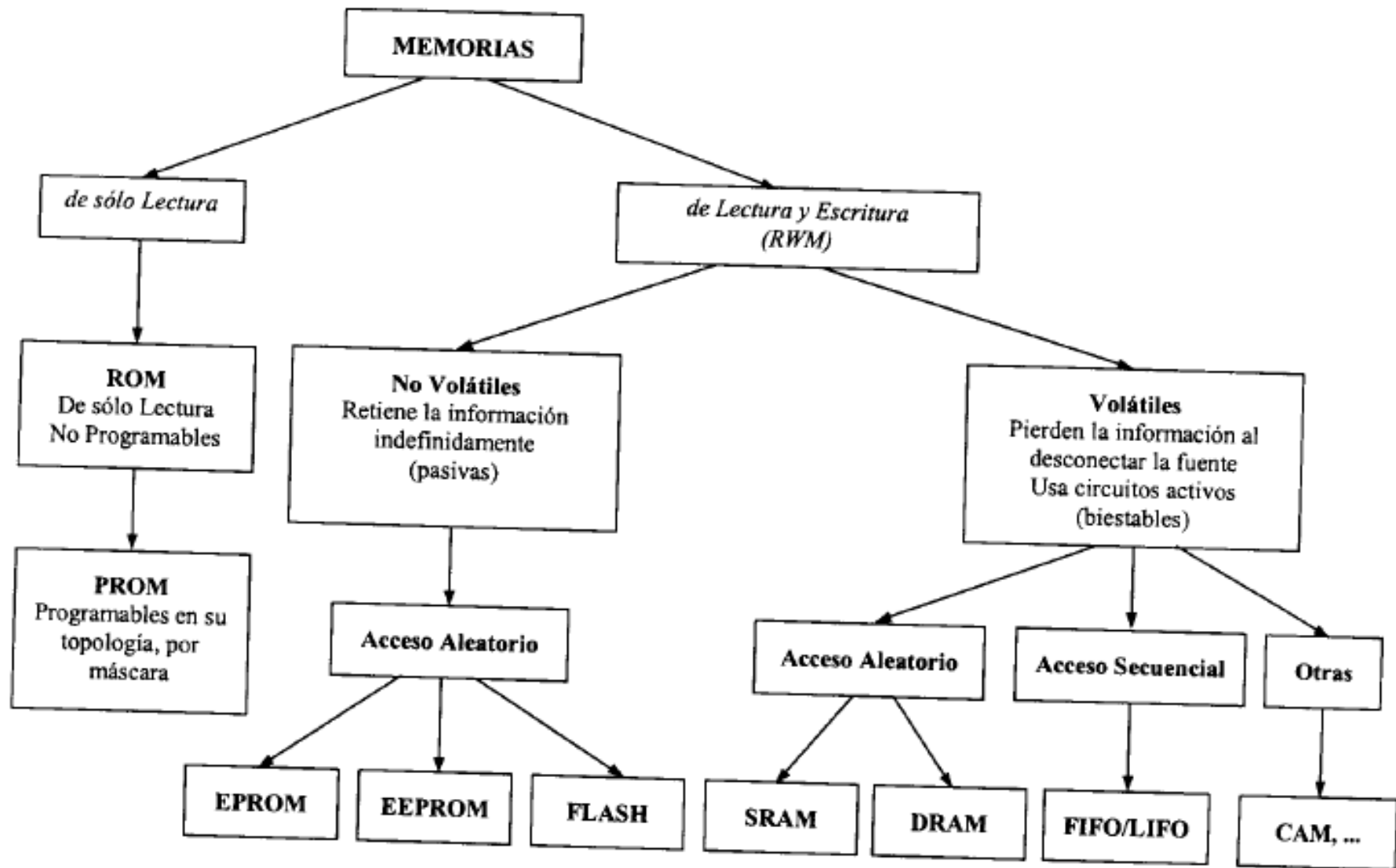


Figura 7.11. Clasificación de los circuitos de memoria.

■ PROM

■ Implementación:

- Simplemente traspasar la **función canónica** al dispositivo PROM

■ Inconvenientes

- 1. Número elevado de entradas \Rightarrow muchas células AND
- 2. Ante simplificaciones extensas, muchas células AND sin utilizar
- 3. Sencillo de implementar, pero alto consumo de recursos

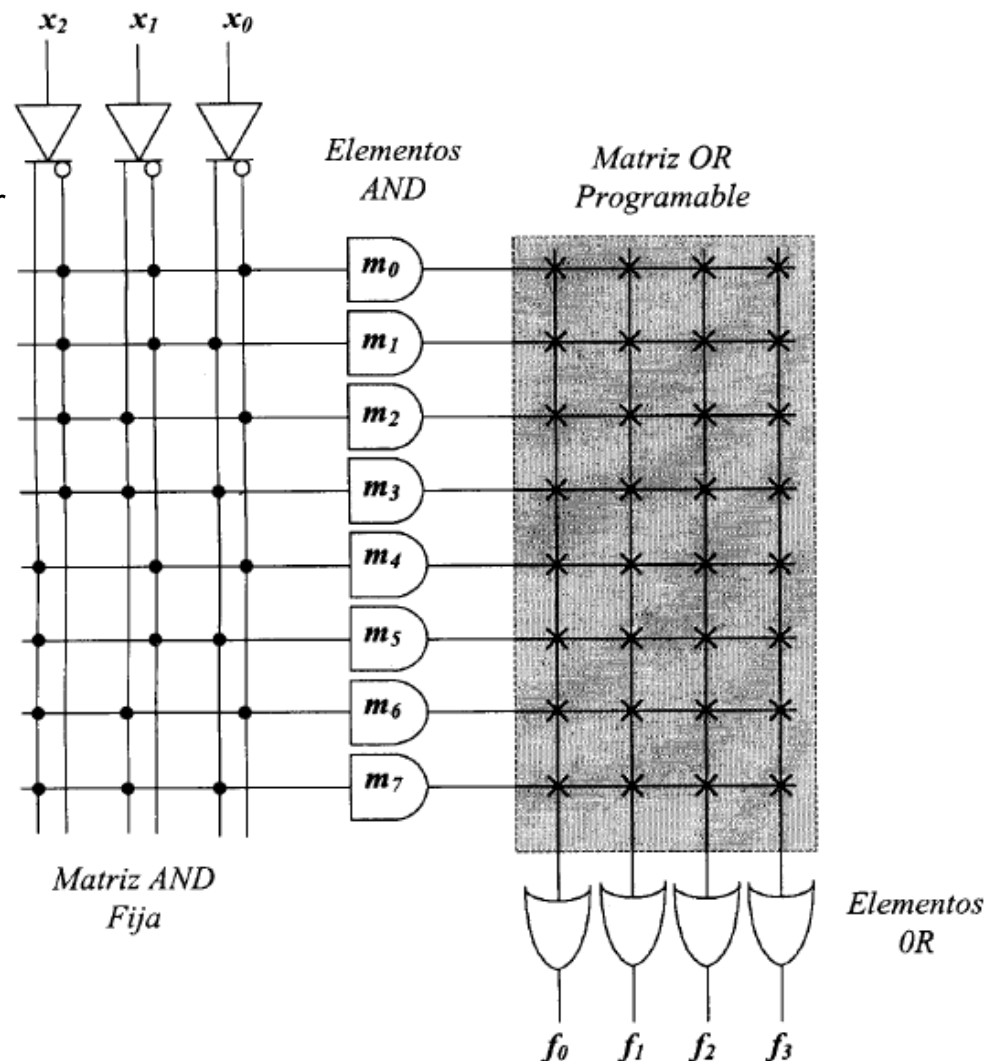


Figura 7.10. Esquema de una memoria PROM de tres variables de entrada (x_0 , x_1 , x_2) y cuatro funciones programables de salida (f_0 , f_1 , f_2 , f_3). Cambiando la naturaleza tecnológica de las cruces de filas y columnas de la parte programable, este mismo esquema es válido para EPROM y EEPROM.

7.3 Transistores de puerta flotante (FLAMOS) y mecanismo de borrado

- Borrado Ultravioleta (varios minutos)
- Borrado eléctrico

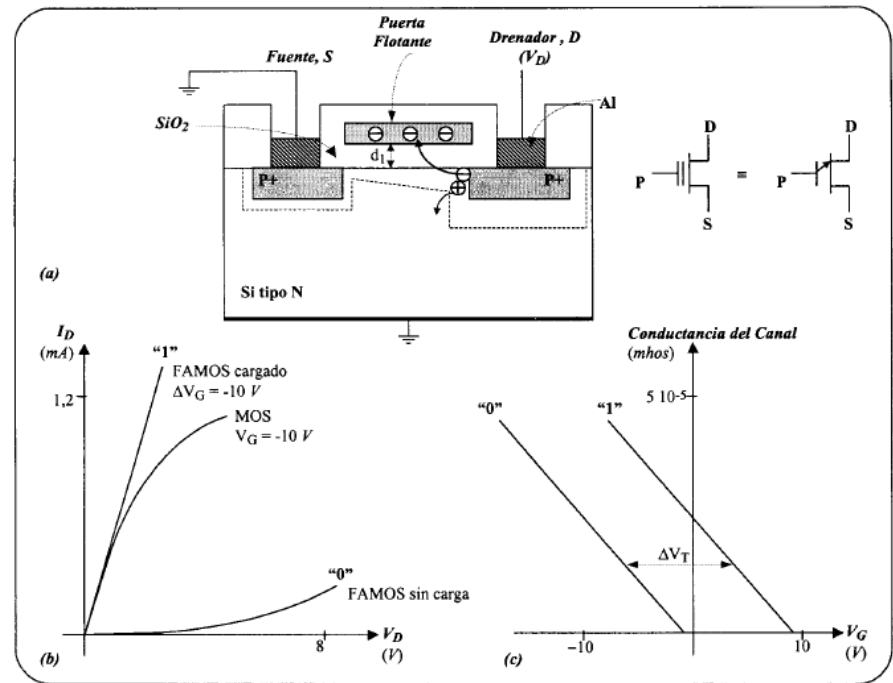


Figura 7.12. Estructura FAMOS y SAMOS. (a) Concepto de puerta flotante en FAMOS, de acuerdo con la propuesta inicial de Kahng y Sze. Modificación de Lizuka. Valores de g_D (b) y ΔV_T (c).

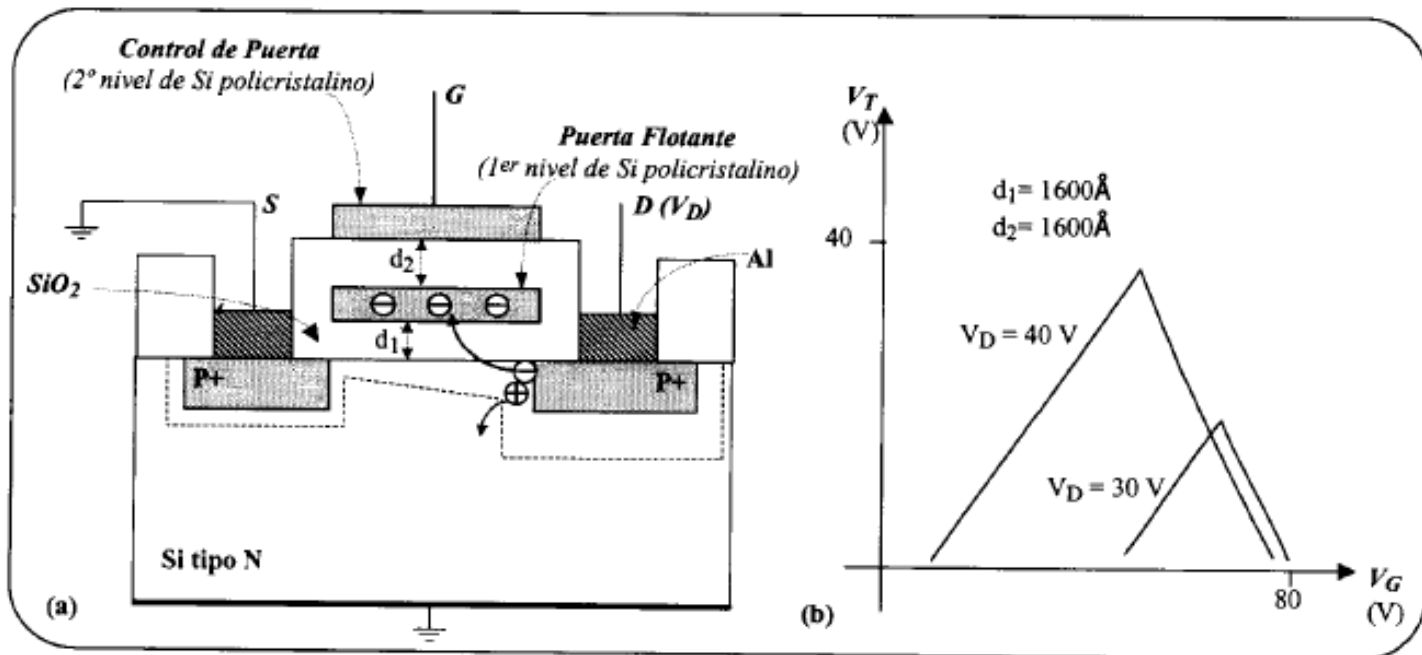


Figura 7.13. (a) Estructura SAMOS con doble puerta para permitir el borrado eléctrico. (b) Variación del umbral (forma cualitativa) en función del potencial en la puerta de control para distintos valores de la tensión de drenador.

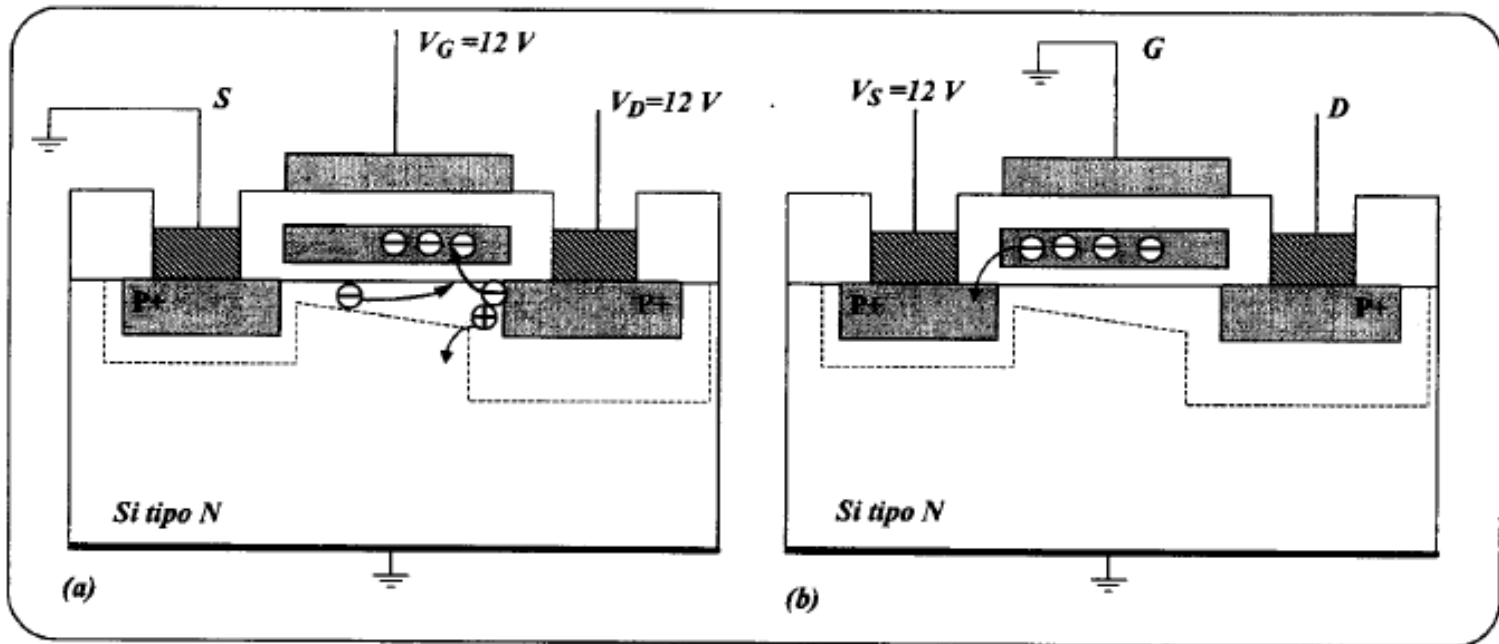


Figura 7.15. Memoria FLASH. (a) Programación. (b) Borrado

	EPROM	EEPROM	FLASH
♦ <i>Densidad</i>	Alta (1 transistor/celda)	Media (más de 1t/c)	Alta (1t/c)
♦ <i>Tamaño</i>	$\geq 16\text{Mbit}$	$\geq 1\text{Mbit}$	$\geq 16\text{Mbit}$
♦ <i>Tiempos de Acceso</i>	Bajo (60ns)	Medio (120 ns)	Bajo (50 ns)
♦ <i>Tiempo de Borrado</i>	Medio (minutos)	—	Muy bajo (<4 s)
♦ <i>Tiempo de Programación</i>	Muy bajo (<5 μs)	Alto ($\approx 4\text{ s}$ /circuito)	Muy bajo (<5 μs)
♦ <i>Reprogramable dentro del Sistema</i>	No	Si	Si
♦ <i>Flexibilidad</i> (N° de ciclos de lectura/escritura sin degradación)	100	10^5	$10^3\text{-}10^5$

Figura.7.16. Tabla de características de los distintos tipo de memorias no volátiles con posibilidad de lectura/escritura (adaptada a partir de la de J.M. Rabaey, 1996).

7.4 Organización interna y ejemplos de EEPROM y FLASH

Empecemos con la estructura interna. Todos los circuitos EPROM, EEPROM y FLASH incluyen, junto a la matriz de celdas, otras facilidades adicionales para controlar el proceso de lectura del estado lógico de esas celdas y su reprogramación fuera o dentro del sistema digital en el que operan. Así, son necesarios:

- Circuitos para direccionar las celdas en las que se desea actuar.
- Circuitos para controlar la selección de “chip” y la ejecución del proceso que se desea realizar en la celda seleccionada en ese “chip”.
- Un conjunto de amplificadores digitales (“buffers”) y registros para adaptar las señales de entrada y salida en sus conexiones al bus.
- Otros circuitos especiales en el caso de las EEPROM FLASH.

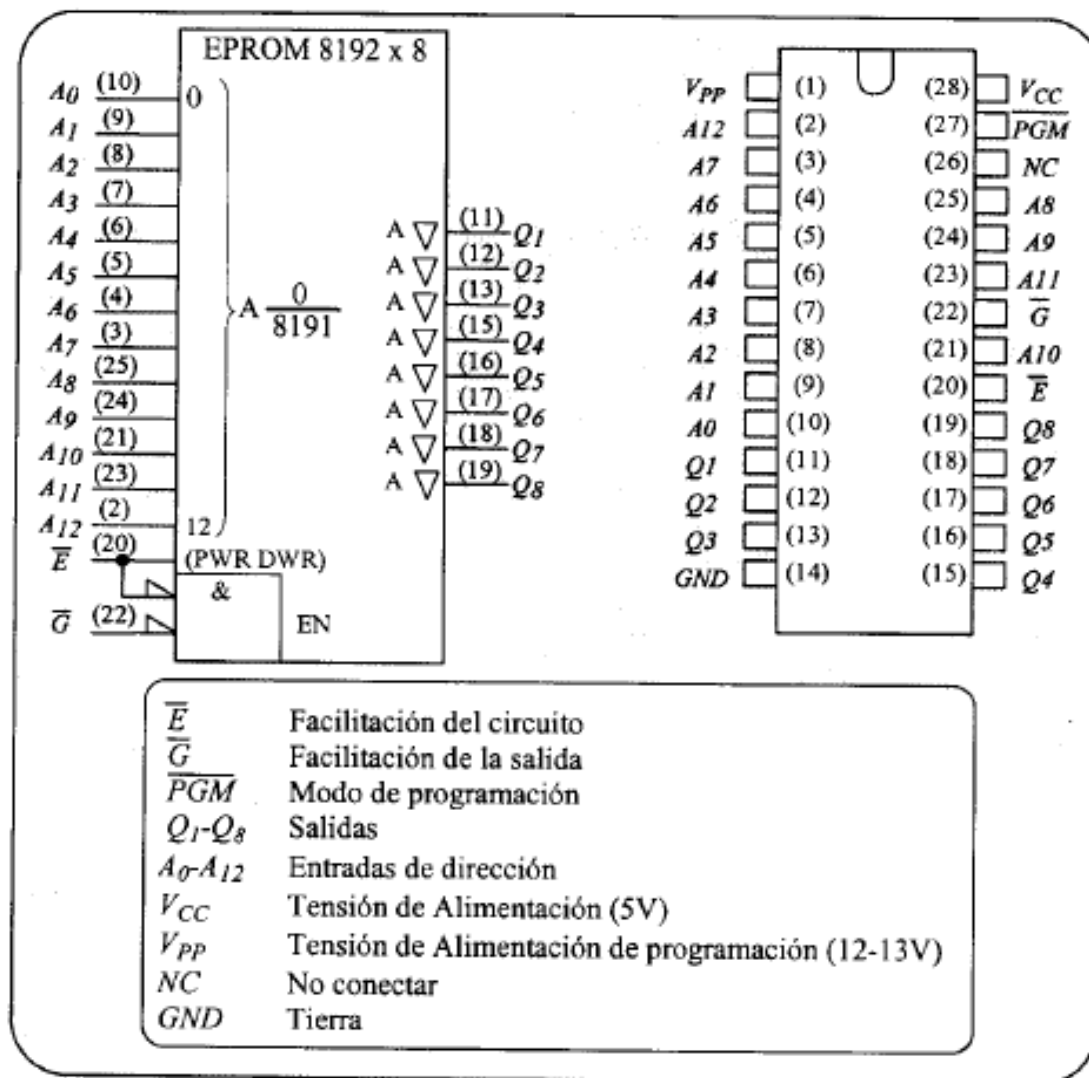


Figura 7.17. Símbolo lógico y significado de los terminales de la memoria EPROM, TMS27C64.

<i>FUNCIÓN</i>	<i>MODOS</i>					
	<i>Lectura</i>	<i>Salida Inhibida</i>	<i>Standby</i>	<i>Programación</i>	<i>Verificación</i>	<i>Inhibición del Programa</i>
\overline{E}	V _{IL}	V _{IL}	V _{IH}	V _{IL}	V _{IL}	V _{IH}
\overline{G}	V _{IL}	V _{IH}	x	V _{IH}	V _{IL}	x
<i>PGM</i>	V _{IH}	V _{IH}	x	V _{IL}	V _{IH}	x
<i>VPP</i>	V _{CC}	V _{CC}	V _{CC}	V _{PP}	V _{PP}	V _{PP}
<i>VCC</i>	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}	V _{CC}
<i>A₉</i>	x	x	x	x	x	x
<i>A₀</i>	x	x	x	x	x	x
<i>Q₁ - Q₈</i>	D _{out}	alta Z	alta Z	D _{in}	D _{out}	alta Z

Figura 7.18. Tabla de modos de programación de la Memoria EPROM TMS27C64.

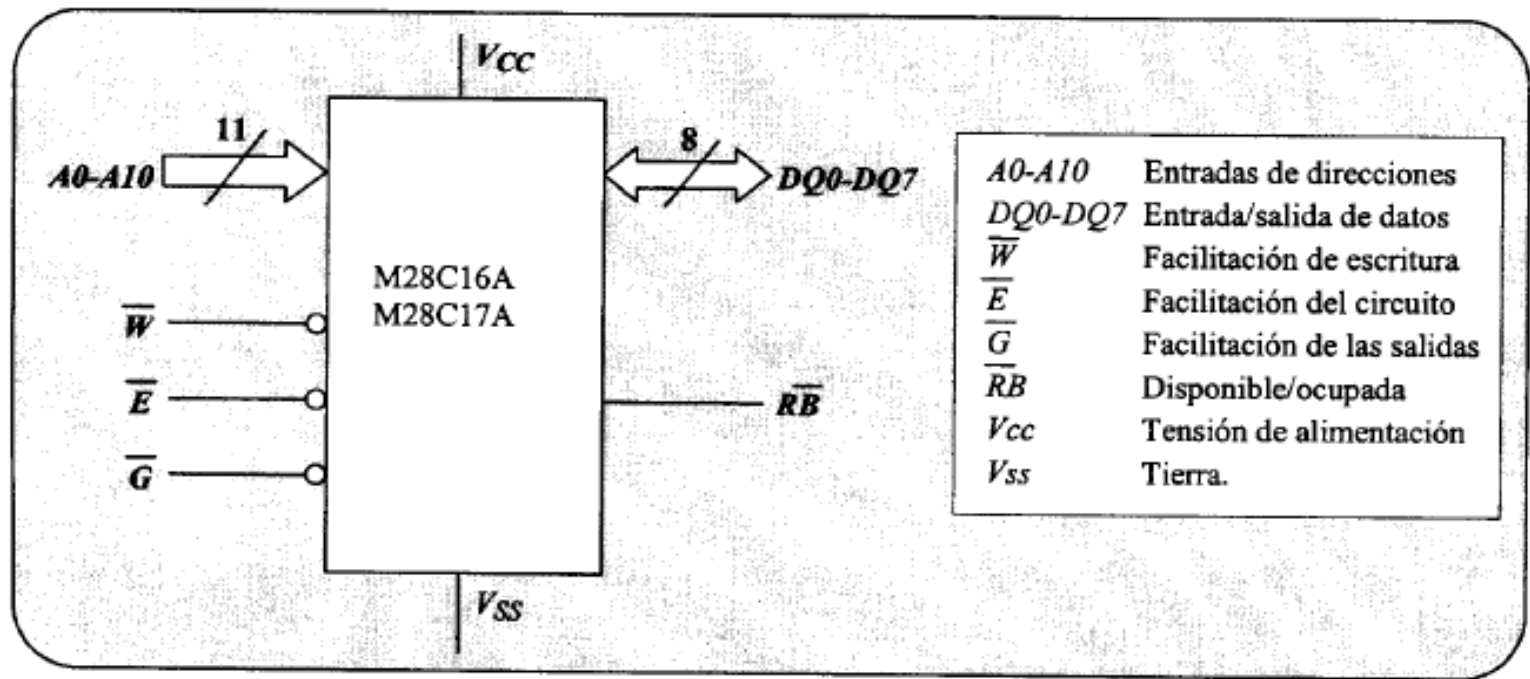


Figura 7.19. (a) Diagrama lógico de la EEPROM M28C16A de ST Microelectronics. (b) descripción del significado de sus terminales

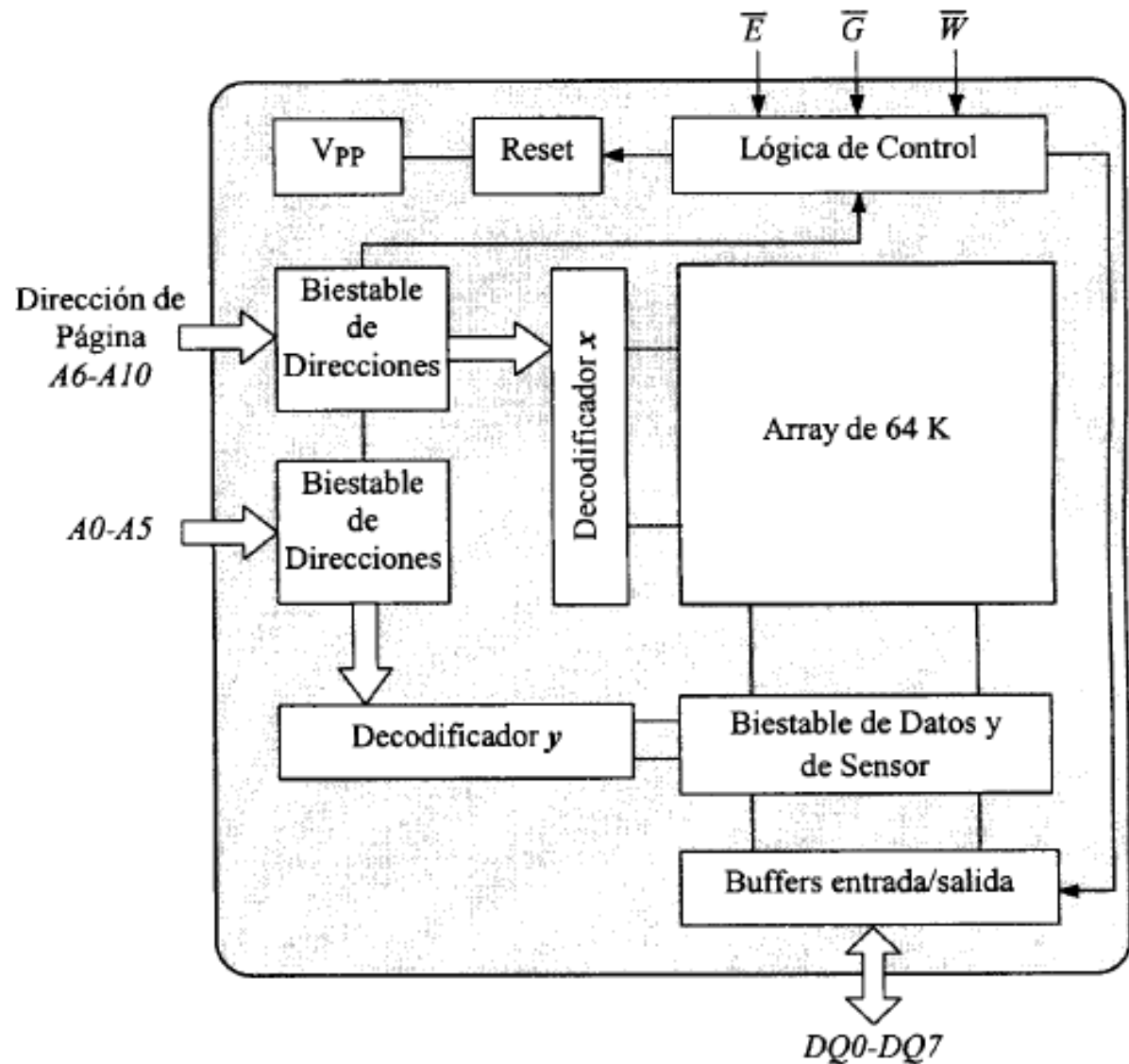
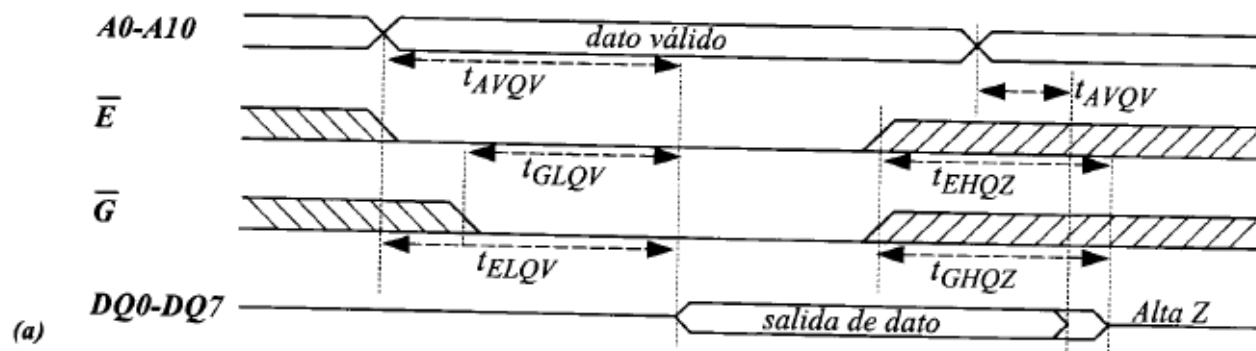


Figura 7.20. Diagrama de bloques de la estructura interna de la EEPROM M28C16A

MODO	\overline{E}	\overline{G}	\overline{W}	<i>DQ0-DQ7</i>
Lectura	V _{IL}	V _{IL}	V _{IH}	<i>Salida de Datos</i>
Escritura	V _{IL}	V _{IH}	V _{IL}	<i>Entrada de Datos</i>
Standby/Inhibición de Escritura	V _{IH}	x	x	<i>Alta Z</i>
Inhibición de Escritura	x	x	V _{IH}	<i>Salida de Datos ó Alta Z</i>
Inhibición de Escritura	x	V _{IL}	x	<i>Salida de Datos ó Alta Z</i>
Inhibición de Lectura	x	V _{IH}	x	<i>Alta Z</i>

Figura 7.21. Tabla de valores de las variables de control de la EEPROM M28C16A durante los distintos modos de operación (lectura, escritura).



Símbolo	Nomenclatura Alternativa	Parámetro	Condición de la Prueba	M28C16A/ M28C17A				Unidad	
				-15		-20			
				Min	Max	Min	Max		
t_{AVQV}	t_{ACC}	Dirección válida para Dato válido	$E=V_{IL}, G=V_{IL}$		150	200	ns		
t_{ELQV}	t_{CE}	Circuito facilitado en baja para salida válida	$G=V_{IL}$		150	200	ns		
t_{GLQV}	t_{OE}	Salida facilitada en baja para salida válida	$E=V_{IL}$		70	80	ns		
t_{EHQZ}	t_{DF}	Circuito facilitado en alta para salida en alta Impedancia	$G=V_{IL}$		0	50	0	60	ns
t_{GHQZ}	t_{DF}	Salida facilitada en alta para salida en alta Impedancia	$E=V_{IL}$		0	50	0	60	ns
t_{AXQX}	t_{OH}	Transición de la dirección para transición de la salida	$E=V_{IL}, G=V_{IL}$		0	0	0	ns	

(b)

Figura 7.22. (a) Forma de onda de los valores estacionarios y las transiciones de las distintas señales de control de la EEPROM durante una operación de lectura. (b) Valores correspondientes de los distintos intervalos temporales


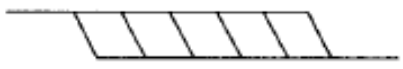

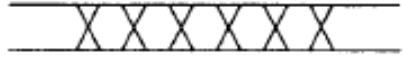
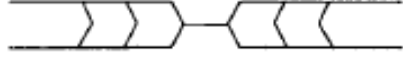
<i>Forma de Onda</i>	<i>Significado en las Entradas</i>	<i>Significado en las Salidas</i>
	Deben ser estacionarias	Valores estacionarios (salida válida)
	Pueden cambiar de H a L	Estará cambiando de H a L
	Pueden cambiar de L a H	Estará cambiando de L a H
	Situación irrelevante. Se permite cualquier cambio	Cambiando. Estado no conocido
	No se aplica	Estado de alta impedancia (línea central)

Figura 7.23. Significado de las distintas formas de onda que se usan para representar la evolución temporal de las líneas de datos y direcciones y las señales de control durante los procesos de lectura y grabación de EEPROMs.

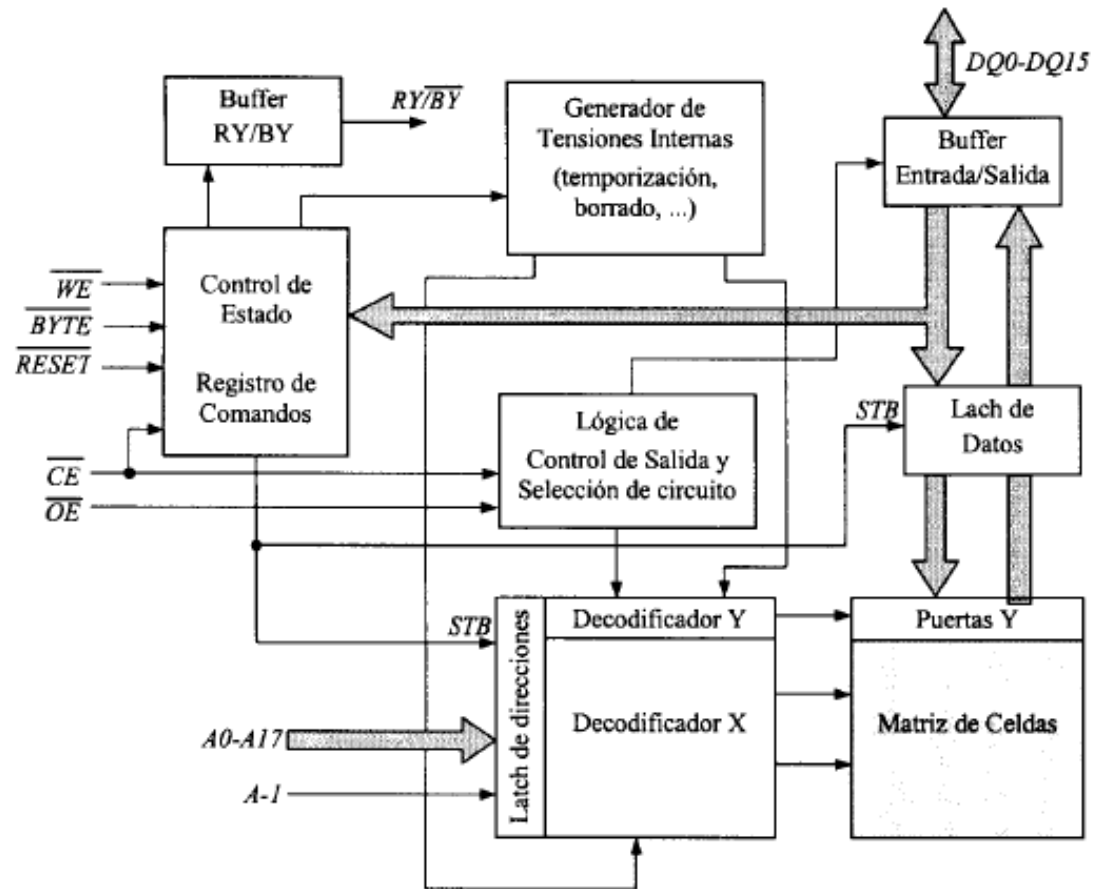


Figura 7.24. Esquema simplificado de la estructura interna de la memoria FLASH de AMD, Am29F400T.

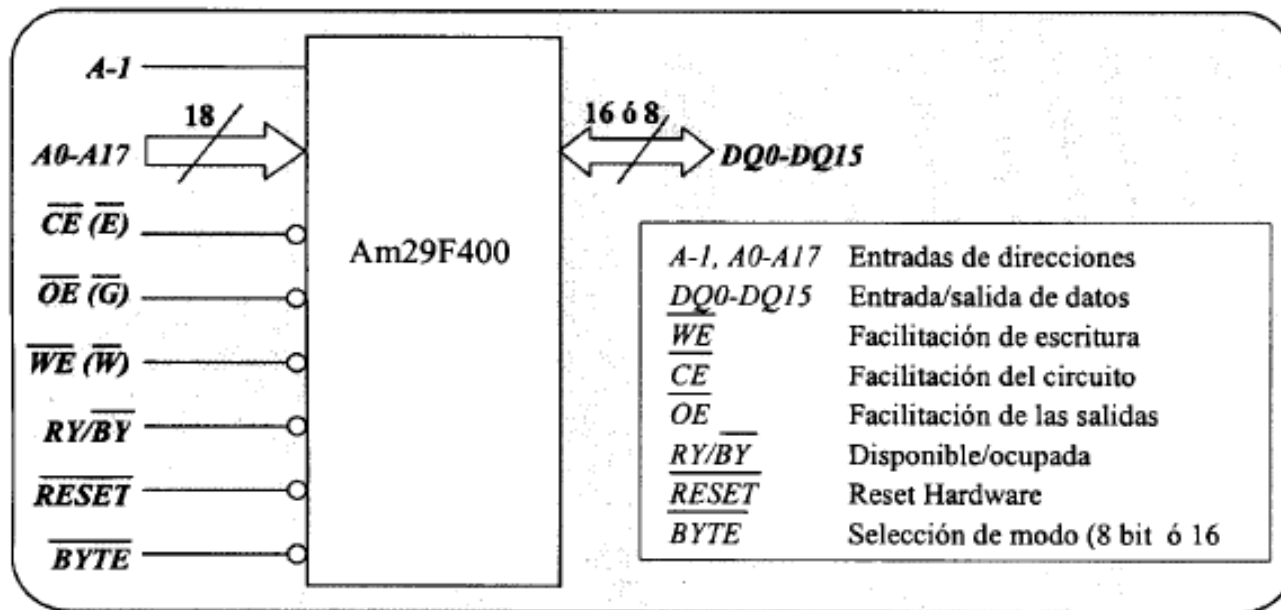


Figura 7.25. Símbolo lógico de la memoria FLASH de AMD, Am29F400T y significado de la función que realizan sus terminales.

7.5 PALs y PLAs

- PAL
- Programable la matriz **AND** \Rightarrow uso para muchas entradas y pocos términos minterm

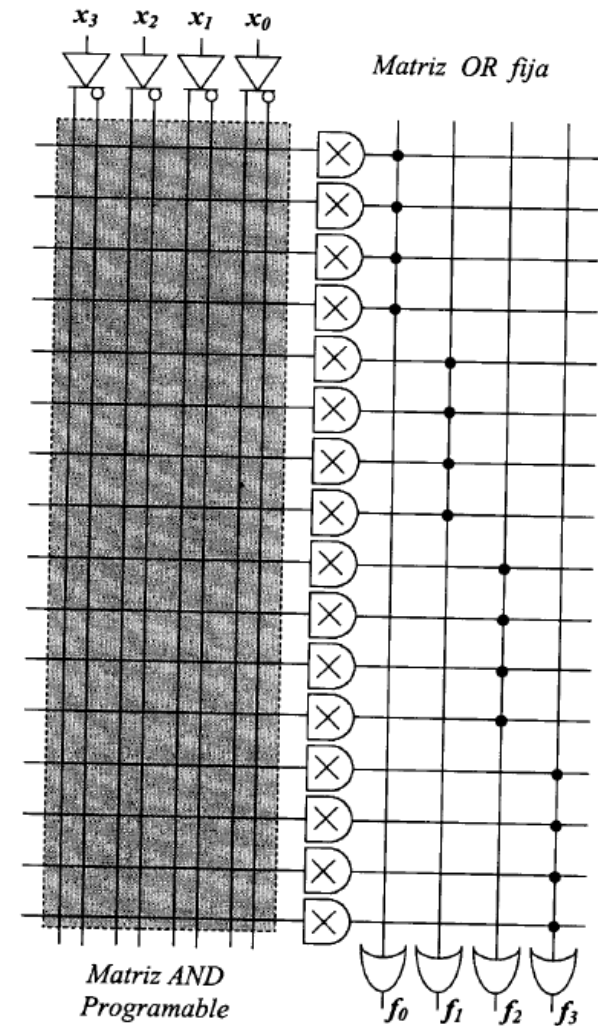


Figura 7.26. Arquitectura de los circuitos programables PAL.

- PLA
- Programable las dos células \Rightarrow mayor coste, mayor versatilidad \Rightarrow facilidades de diseño
- Notación:
 - $PLA\ 3 \times 4 \times 2 \Rightarrow$ (3 entradas, 4 AND y 2 salidas)

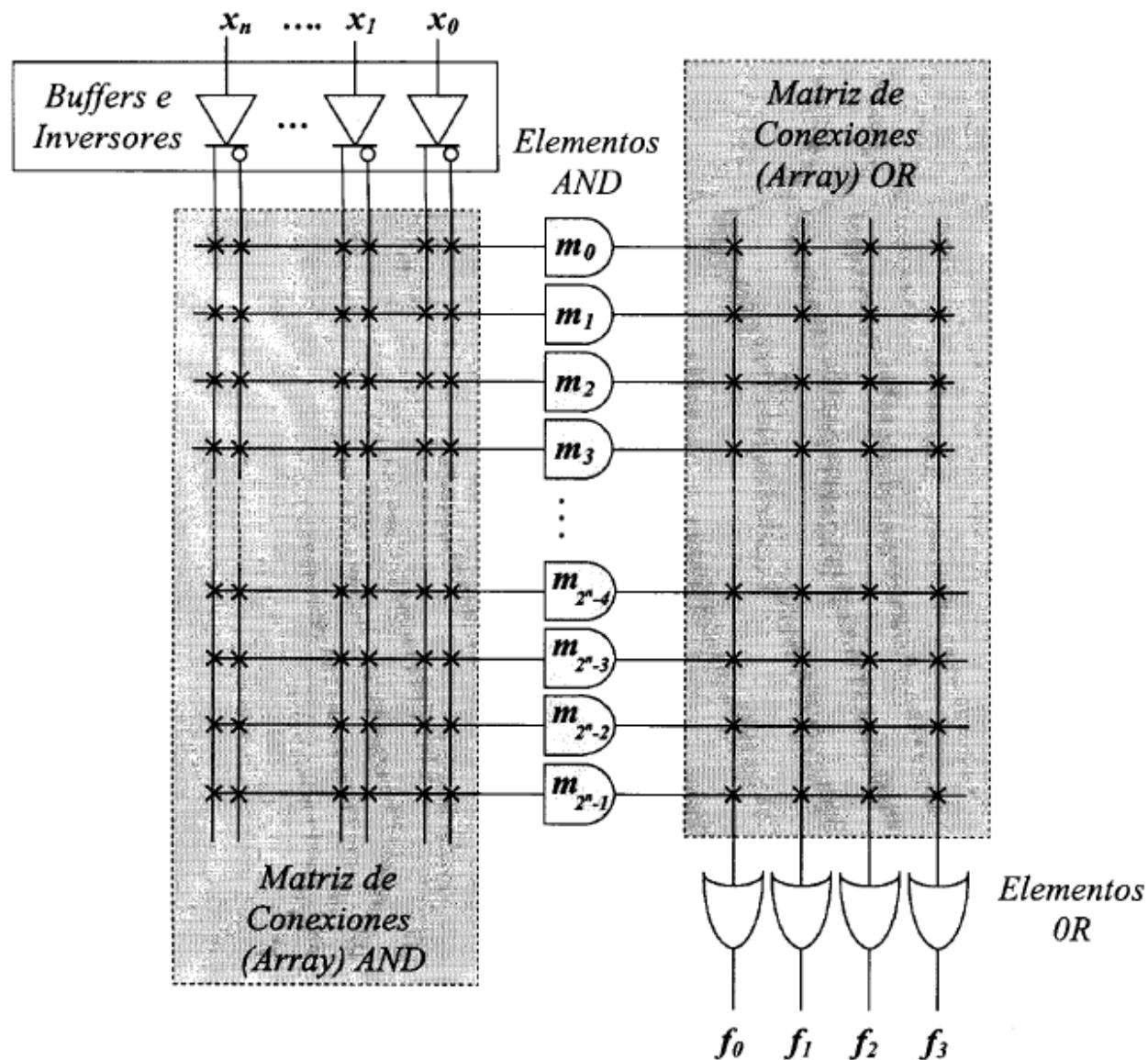


Figura 7.27. Arquitectura de los circuitos PLA.

Ejercicio: Supongamos que disponemos de una PLA 3*4*2. Es decir, una PLA con 3 líneas de entrada (x_0, x_1, x_2), 4 productos o funciones AND y 2 líneas de salida (f_0, f_1) y queremos programarla para que implemente las funciones:

$$f_0(x_0, x_1, x_2) = \sum m(3, 5, 6, 7)$$

$$f_1(x_0, x_1, x_2) = \sum m(0, 2, 4, 6)$$

Solución

Para implementar estas funciones lo primero que tenemos que hacer es simplificar las expresiones y minimizarlas. Así,

$$f_0(x_0, x_1, x_2) = \sum m(3, 5, 6, 7) = \bar{x}_2 x_1 x_0 + x_2 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0 = x_1 x_0 + x_2 x_0 + x_2 x_1$$

$$f_1(x_0, x_1, x_2) = \sum m(0, 2, 4, 6) = \bar{x}_2 \bar{x}_1 \bar{x}_0 + \bar{x}_2 x_1 \bar{x}_0 + x_2 \bar{x}_1 \bar{x}_0 + x_2 x_1 \bar{x}_0 = \bar{x}_0$$

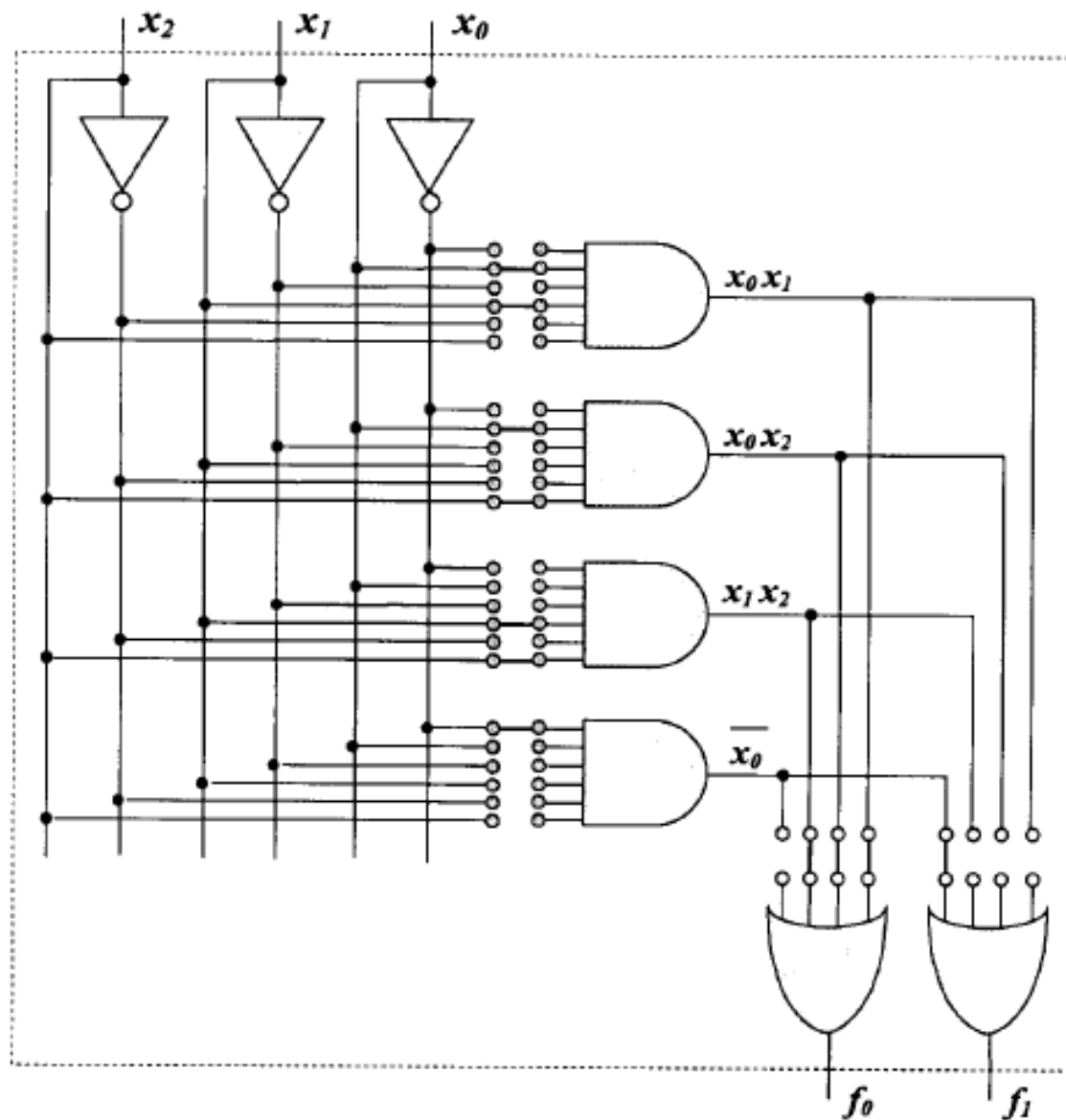
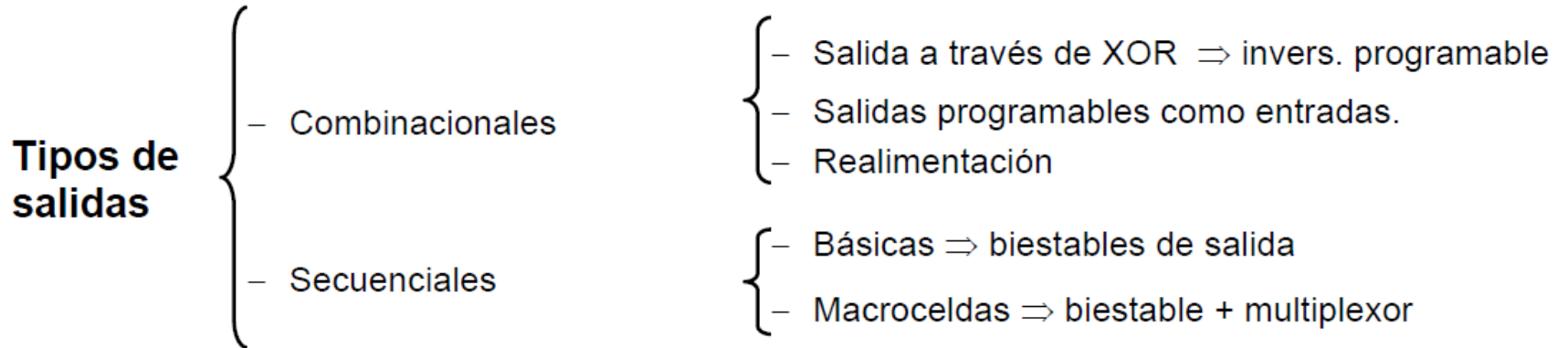


Figura 7.28. Síntesis con una PAL 3*4*2 de las funciones $f_0(x_0, x_1, x_2) = \sum m(3, 5, 6, 7)$, $f_1(x_0, x_1, x_2) = \sum m(0, 2, 4, 6)$.

7.6 Configuraciones de salida



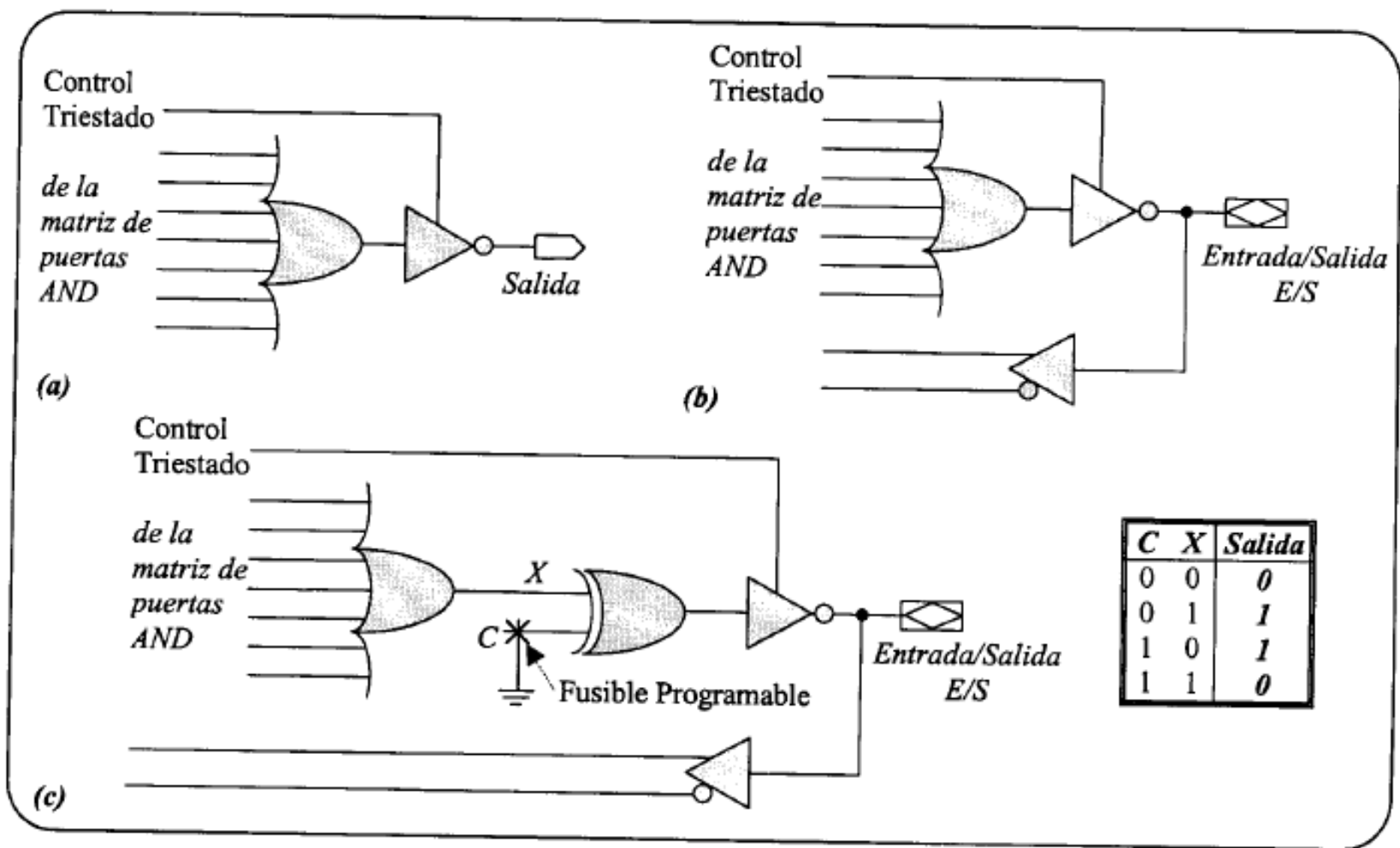


Figura 29. Distintas posibilidades de salida combinacional de un circuito PAL. (a) Salida básica (activa en baja). (b) Salida (también activa en baja) con posibilidad de ser programada como entrada y con realimentación. (c) Lo mismo que (b) pero incluyendo una puerta XOR para permitir programar la polaridad.

figura 7.30. Hay tres aspectos a destacar en este circuito:

1. La salida, $Q(t)$, coincide con el valor de la entrada (su “estado”) en el instante anterior, $Q(t)=D(t-\Delta t)$. Es decir, el biestable D es de hecho un retardo, lo que posibilita toda la lógica secuencial y, por consiguiente, el uso de la PAL como módulo universal de diseño de contadores, registros y autómatas finitos en general.
2. El circuito es síncrono. Es decir, todas las transiciones de datos desde D a Q y \bar{Q} ocurren a las subidas o bajadas de un pulso de reloj que es *común* a todas las salidas de la PAL.
3. La señal de control de los buffer es la misma para todas las líneas de salida.

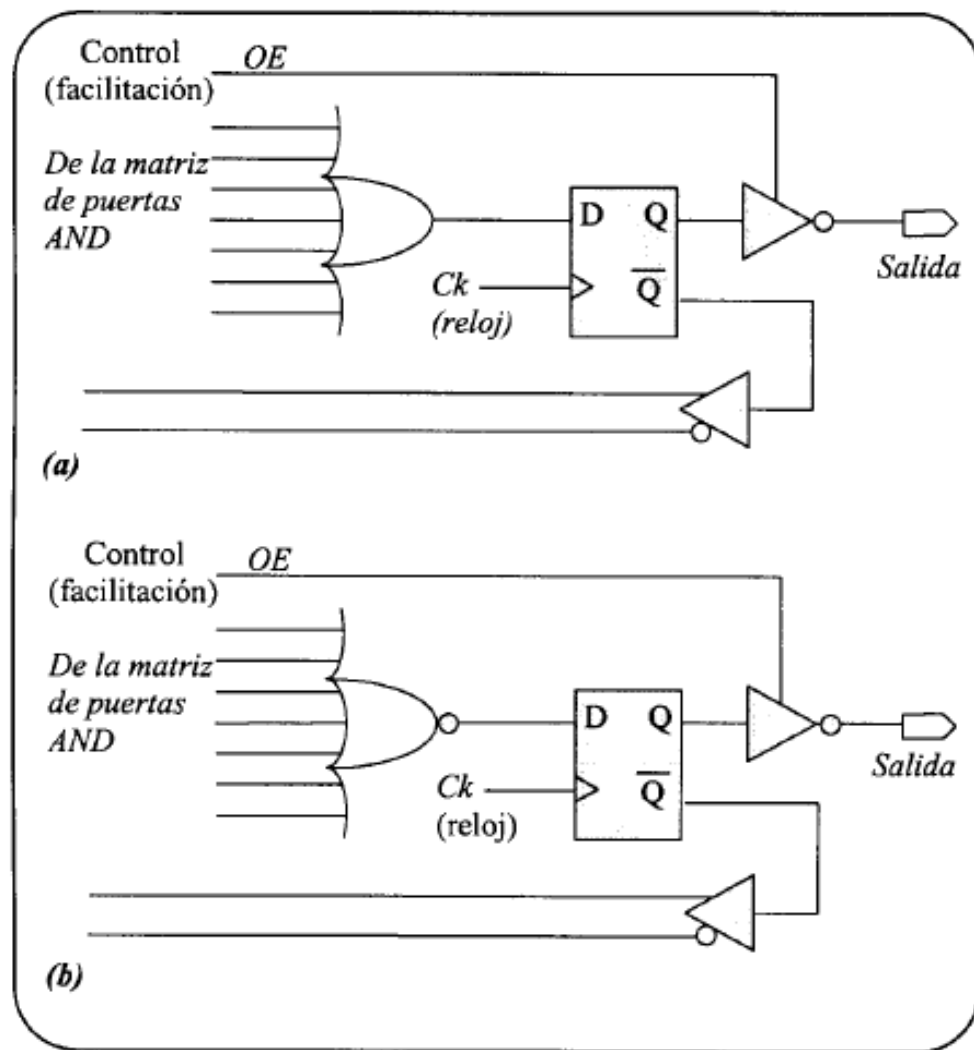


Figura 7.30. Salidas de una PAL a través de un biestable D (registered output). **(a)** Activa en baja. **(b)** Activa en alta.

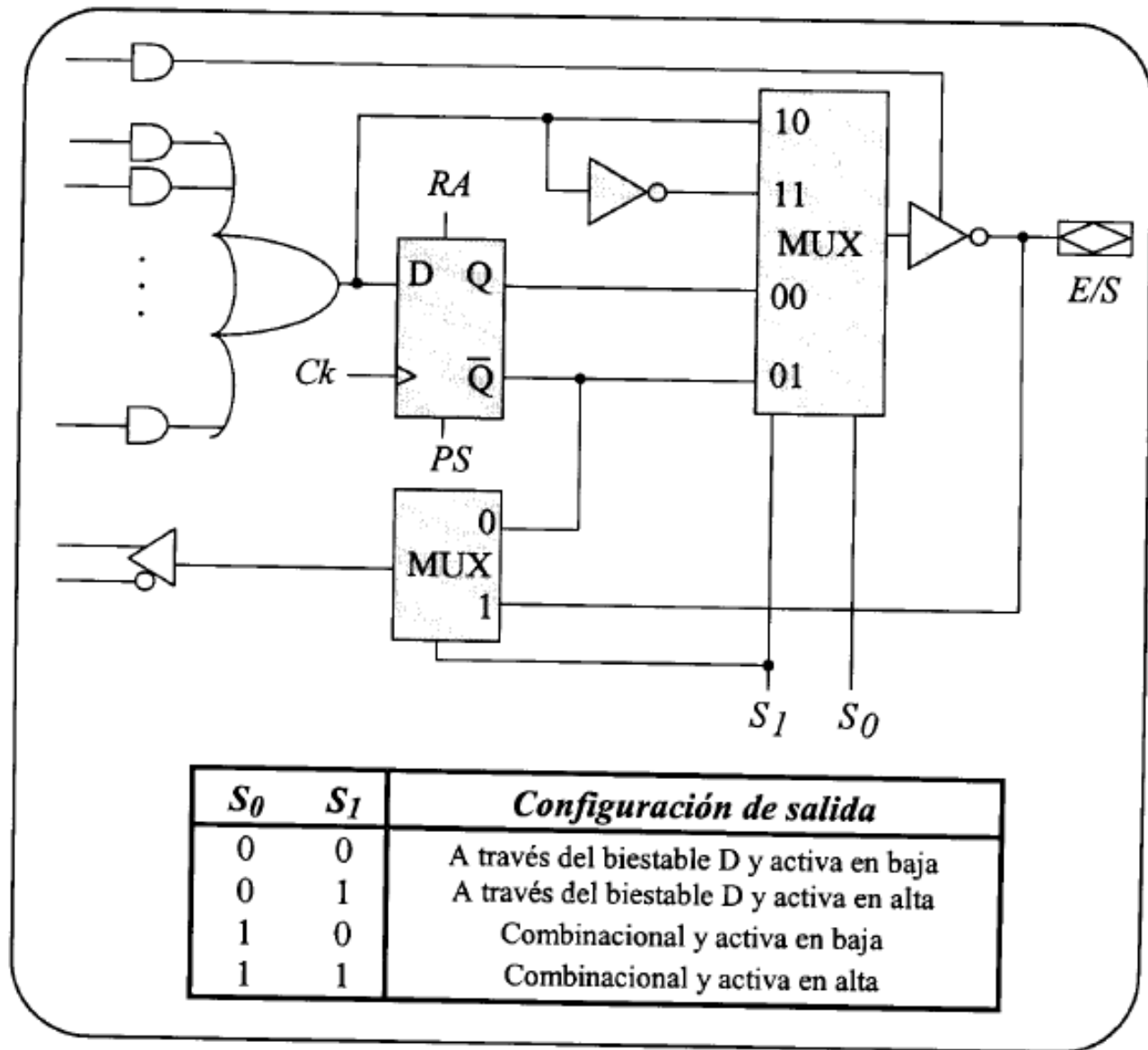


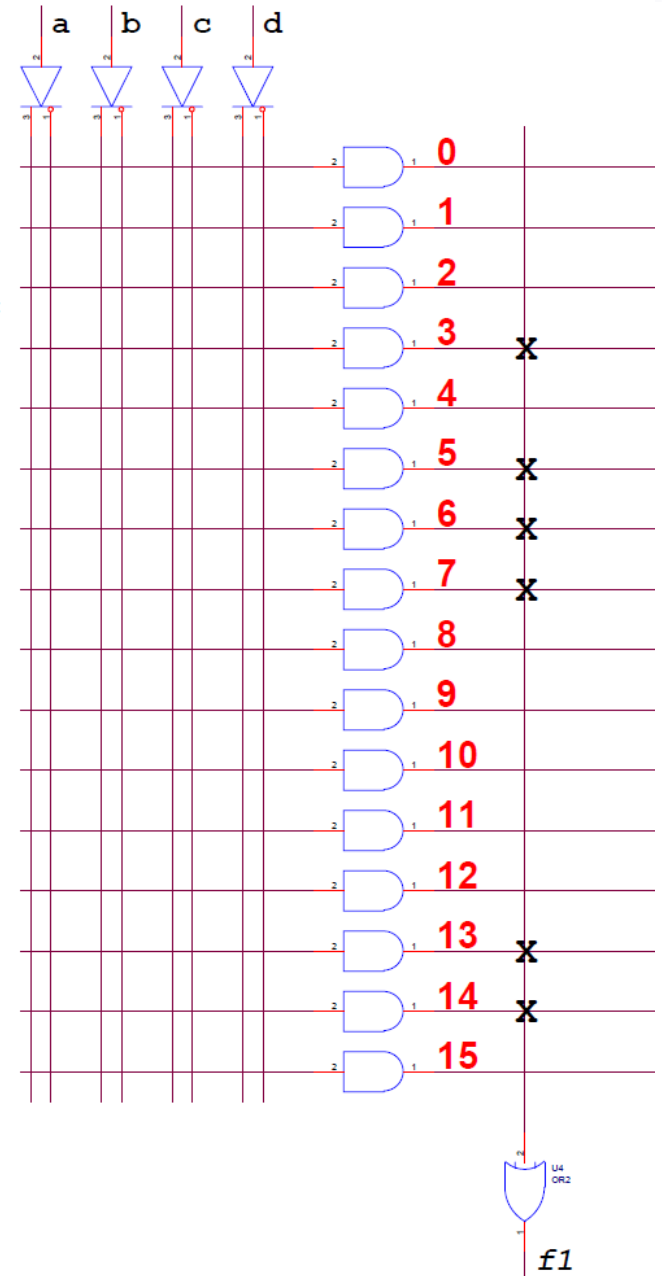
Figura 7.31. Esquema de una macrocelda de salida típica de circuitos PAL, PALCE y GAL. Ck =pulsos de reloj, RA = Reset asíncrono, PS =Preset síncrono..

Implementar mediante PROMs la función lógica:

$$f_1 = \prod M(0, 3, 4, 5, 6, 7, 11, 13, 14, 15)$$

$$\overline{f_1} = \prod M(1, 2, 8, 9, 10, 12)$$

$$f_1 = \overline{M1 \cdot M2 \cdot M8 \cdot M9 \cdot M10 \cdot M12} = \overline{M1} + \overline{M2} + \overline{M8} + \overline{M9} + \overline{M10} + \overline{M12} = m14 + m13 + m7 + m6 + m5 + m3$$

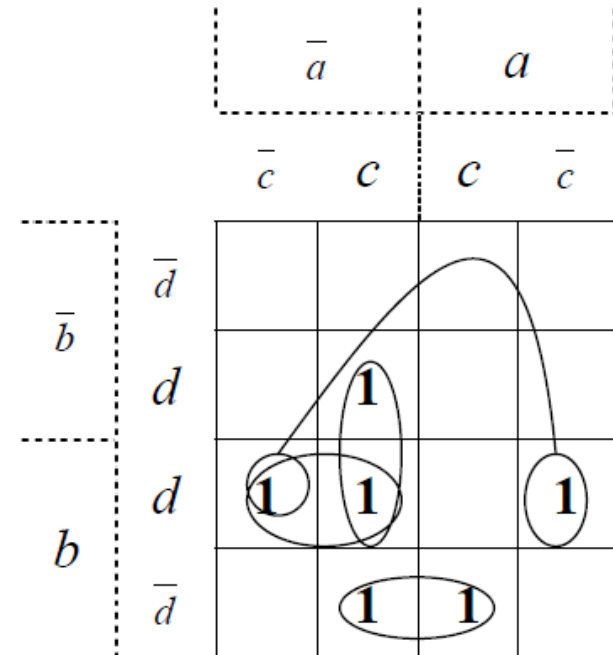
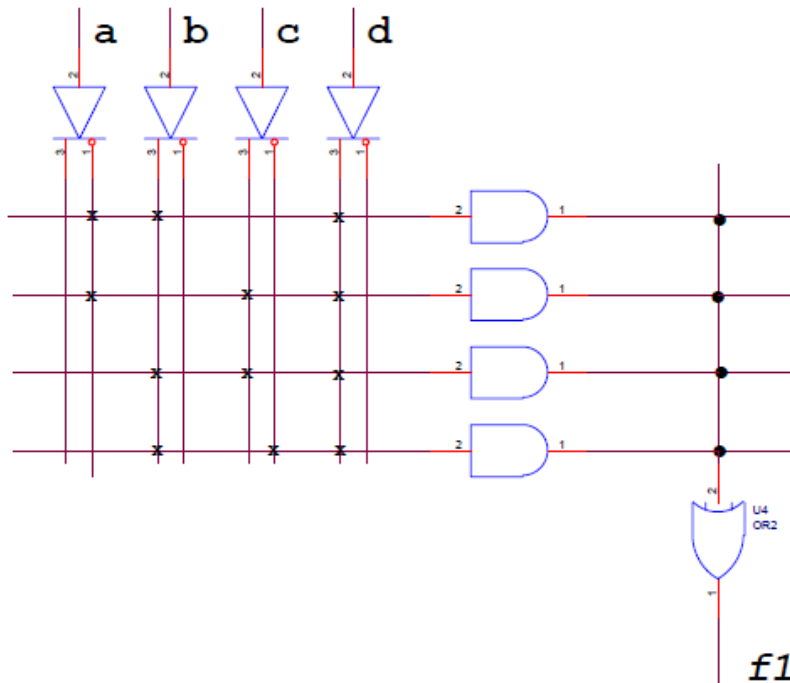


Implementación mediante PALs

Primeramente se ha de simplificar:

$$f_1 = m_3 \cdot m_5 \cdot m_6 \cdot m_7 \cdot m_{13} \cdot m_{14}$$

$$f_1 = \bar{a}bd + \bar{a}cd + bcd + b\bar{c}d$$



Ejemplo con PLAs.

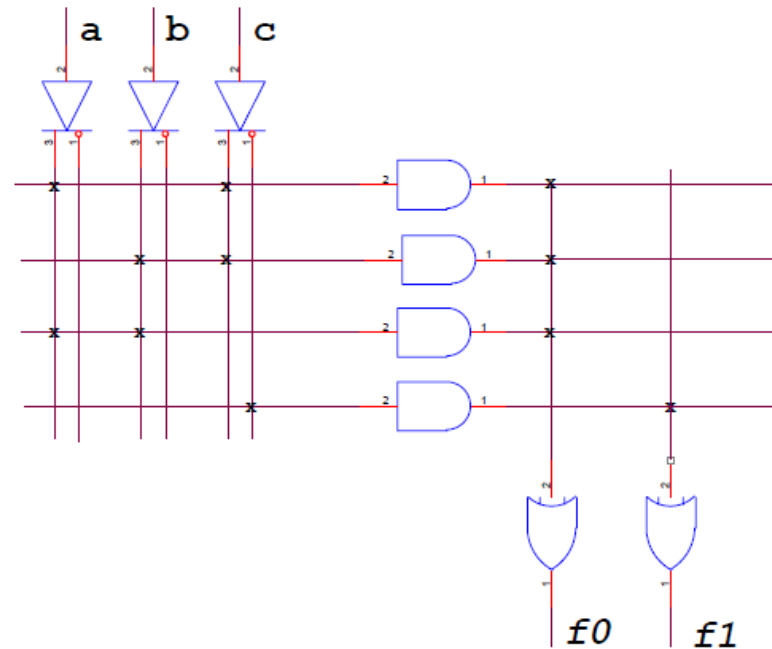
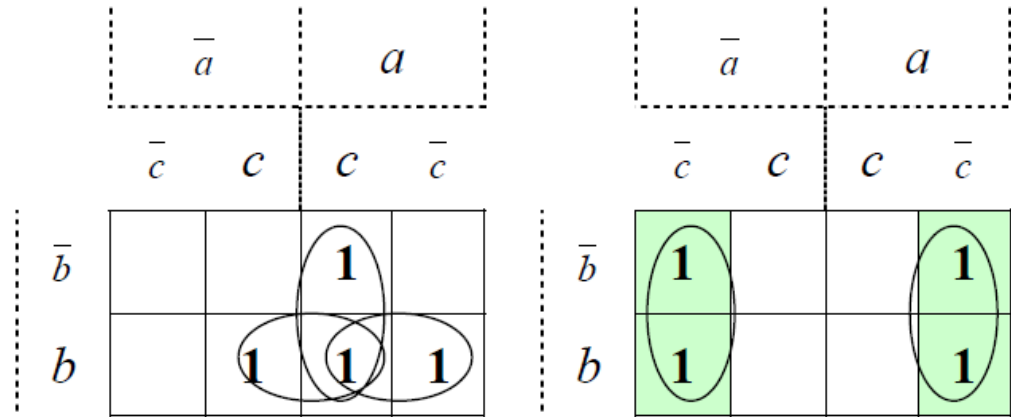
Con una PLA de $3 \times 4 \times 2$ implementar

$$f_0(a,b,c) = \sum m(3,5,6,7)$$

$$f_1(a,b,c) = \sum m(0,2,4,6)$$

$$f_0(a,b,c) = ac + bc + ab$$

$$f_1(a,b,c) = \bar{c}$$



7.7 Nomenclatura y ejemplo de circuito PAL

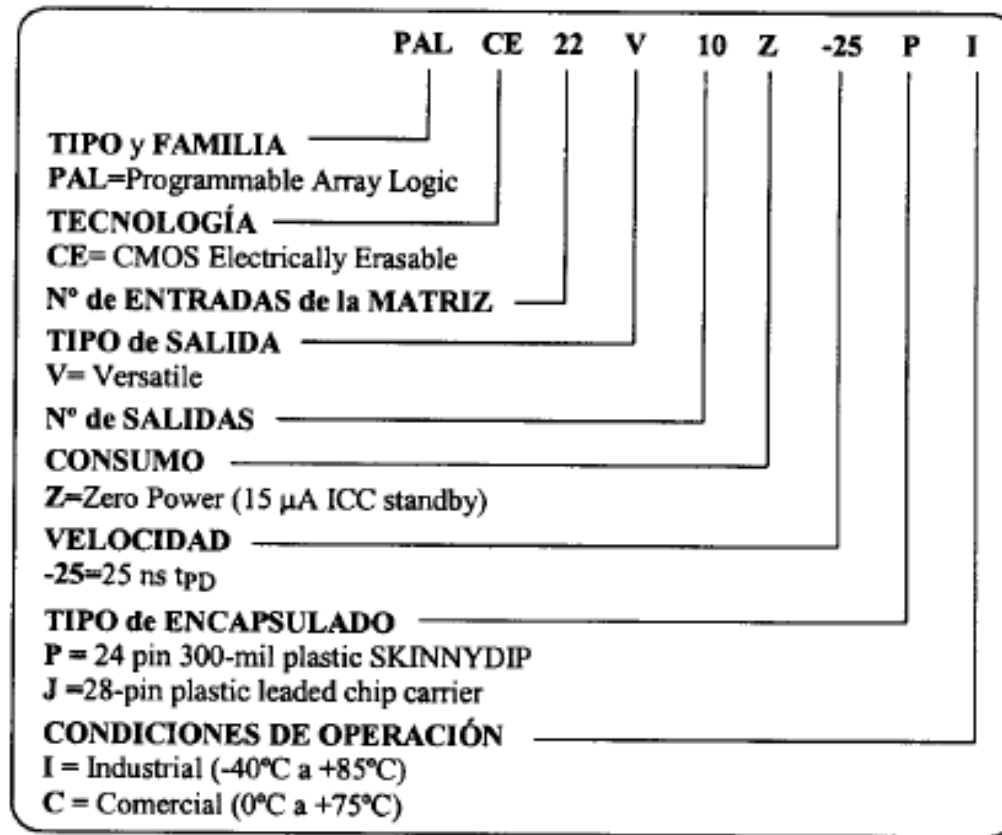


Figura 7.32. Nomenclatura usada para la familia PALCE

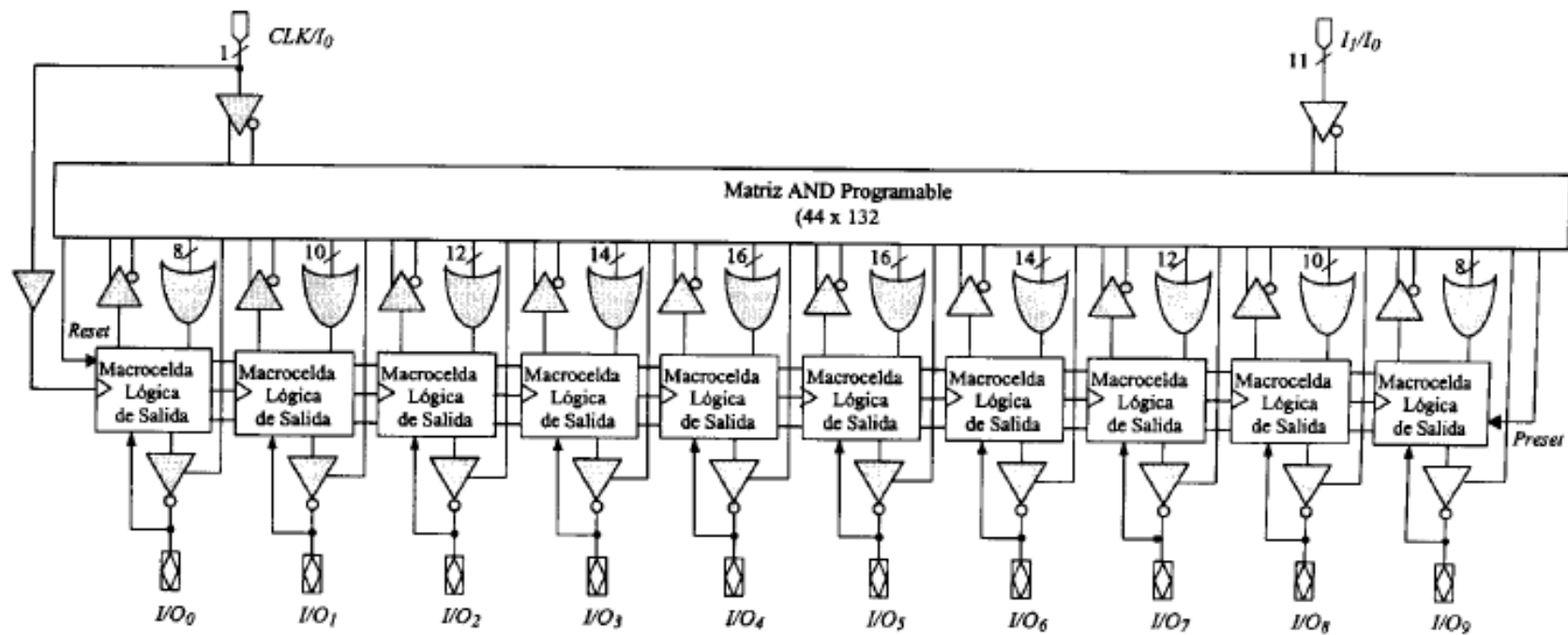


Figura 7.33. Versión simplificada del esquema de la PALCE 22V10Z-25.