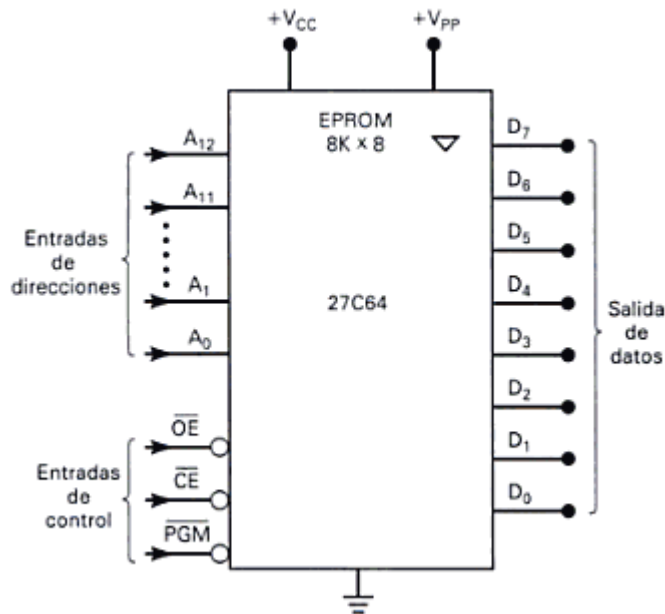


Eprom 2764 fuente: tocci

Las EPROMs están disponibles en una amplia gama de capacidades y tiempos de acceso; los dispositivos con capacidad de $512K \times 8$ y tiempo de acceso de 20 ns son muy comunes. El 27C64 es un ejemplo de una EPROM pequeña. El 27C64 es una EPROM CMOS $8K \times 8$ que opera de una fuente de alimentación individual de +5 V durante operación normal. En la figura 11-12(a) se muestra el símbolo lógico para el 27C64. Note que presenta 13 entradas de dirección, puesto que $2^{13} = 8192$ y 8 salidas de datos. Tiene cuatro entradas de control. \overline{CE} es la entrada de habilitación del chip que se usa para colocar el dispositivo en el modo de espera, donde su consumo de potencia se reduce. \overline{OE} es la habilitación de salida y se usa para controlar los búferes triestados de salida de datos del dispositivo, de modo que éste se pueda conectar a un bus de datos del microprocesador sin contención de bus. V_{pp} es el voltaje especial de programación que se requiere durante el proceso de programación. \overline{PGM} es la entrada de habilitación de programa que se activa para almacenar datos en la dirección seleccionada.



El 27C64 tiene varios modos de operación que controlan los pines *CE*, *OE*, *V_{pp}* y *PGM*, como se muestra en la figura 11-12(c). El modo de programa se usa para escribir nuevos datos en las celdas EPROM. Esto se hace con mucha frecuencia en una EPROM "limpia", cuando se ha borrado previamente con luz UV, de manera que las celdas contienen puros unos. El proceso de programación escribe una palabra de ocho bits en la ubicación de dirección una sola vez como sigue: (1) la dirección se aplica a los pines de dirección; (2) los datos deseados se colocan en los pines de datos, los cuales funcionan como entradas durante el proceso de programación; (3) un voltaje de programación mayor de 12.75 V se aplica a *V_{pp}*; (4) *CE* se mantiene en BAJO; (5) *PGM* se pulsa a BAJO durante 100 μ s y los datos se leen de regreso. Si los datos no se almacenaran con éxito, se aplicaría otro pulso a *PGM*. Esto se repite en la misma dirección, hasta que los datos se almacenen de manera correcta. Dicho proceso se repite para todas las ubicaciones de memoria. Si se hace de forma manual, puede tomar varias horas; sin embargo, por lo general se hace de manera automática con un programador EPROM comercial, de manera muy similar a los programadores PROM que se describieron antes. Una EPROM limpia se puede programar en menos de un minuto una vez que los datos deseados hayan sido ingresados, transferidos o descargados en el programador EPROM. El 27C512 es una EPROM común de 64K \times 8 que opera de forma muy similar al 27C64, pero que ofrece mayor capacidad de almacenamiento.

Tabla 1 - Comandos de la memoria EPROM-Flash													
Comandos	Ciclos	Primer ciclo de escritura		Segundo ciclo de escritura		Tercer ciclo de escritura		Cuarto ciclo de escritura		Quinto ciclo de escritura		Sexto ciclo de escritura	
		Dir.	Datos	Dir.	Datos	Dir.	Datos	Dir.	Datos	Dir.	Datos	Dir.	Datos
Lectura/Reset	1	XXX	F0										
Lectura/Reset	4	5555	AA	2AAA	55	5555	F0	RA	RD				
Autoselección	4	5555	AA	2AAA	55	5555	90						
Byte de Programa	4	5555	AA	2AAA	55	5555	A0	PA	PD				
Borrar Memoria	6	5555	AA	2AAA	55	5555	80	5555	AA	2AAA	55	5555	10
Borrar sector	6	5555	AA	2AAA	55	5555	80	5555	AA	2AAA	55	SA	30

Para todos los comandos de direc. los bits de direcciones A15-a18 son X (no importa su valor), excepto Direc. de Programa (PA) y Direc. de Sector (SA)

Todos los datos y direcciones están en notación hexadecimal.

RA = Dirección de la posición de memoria que se va a leer.

PA = Dirección de la posición de memoria que se va a programar (en el flanco negativo de WE).

SA = Dirección del sector que se va a borrar.

RD = Contenido de la posición de memoria RA.

PD = Contenido de la posición de memoria PA (en el flanco negativo de WE).

Como crear un archivo para grabar en la EPROM fuente: internet

Se puede hacer con cualquier editor de texto plano como Notepad pero por la estructura del archivo es más eficiente crearlo con Excel o similar e importarlo al editor de texto plano (ASCII).

Todos los números del archivo deben estar en hexadecimal (base 16).

La primera columna contiene el segundo bit más significativo de la dirección de memoria (16^1), las demás columnas contienen los datos en cada dirección ordenados ascendentemente según dirección.

Por ejemplo, en el archivo de texto de la figura, el dato **FA** de la primera fila, segunda columna quedará en la dirección 0000000h, **01** en la dirección 0000001h, **75** en la dirección 0000010h, **AC** en 0000012h

```

$0000,FA 01 04 A3 2D 3F 00 00 56 79 7F 57 1B BB FA 0A
$0010,75 1B AC FA 0A FA 01 2D 3F 00 11 56 79 7F 04 A3
$0020,. . .
.
.
.
$00F0,. . .
$0100,. . .
$0110,. . .
.
.
.

```

Figura 1

Configuración de pines de EPROM 2764

1	Vpp	Vcc	28
2	A12	PGM'	27
3	A7	NC	26
4	A6	A8	25
5	A5	A9	24
6	A4	A11	23
7	A3	OE'	22
8	A2	A10	21
9	A1	CE'	20
10	A0	O7	19
11	Oo	O6	18
12	O1	O5	17
13	O2	O4	16
14	Vss	O3	15

Figura 2

Los pines A0 a A12 son las entradas de direccionamiento de la EPROM, siendo A0 el bit menos significativo. A su vez, los pines O0 a O7 son las salidas, con O0 el bit menos significativo. En el modo de "lectura" de la EPROM, que es el que nos interesa, los terminales PGM' y Vpp deben ir a nivel "1", mientras que los terminales *enable*, CE' y OE' deben ir a "0".

Si seguimos el ejemplo de la Figura 1, tendríamos algo como lo siguiente:

Entradas													Salidas							
A ₁₂	A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	O ₇	O ₆	O ₅	O ₄	O ₃	O ₂	O ₁	O ₀
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
...													...							
0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	1	0	0
...													...							