

ESCUELA POLITÉCNICA NACIONAL

TESIS DE GRADO

SISTEMAS DE DETECCIÓN Y CORRECCIÓN DE
ERRORES EN MEMORIAS DE COMPUTADORAS.

Tesis previa a la obtención del
título de ingeniero en electrónica
y telecomunicaciones.

Roberto Corral Davalos, -

Diciembre

1982

CERTIFICO QUE EL PRESENTE TRABAJO
HA SIDO ELABORADO EN SU TOTALIDAD
POR EL Sr. ROBERTO CORRAL DAVALOS

H. Jacobsen

DIRECTOR DE TESIS

☞ nombre en imprenta !!

INTRODUCCION

INTRODUCCION

A medida que los sistemas de computación avanzan en sus más sofisticadas aplicaciones, su confiabilidad se vuelve de gran importancia.

La utilización de los sistemas de detección y corrección de errores hacen posible reducir al mínimo la probabilidad de que un error pueda producir graves consecuencias en un determinado sistema, aumentando de esta manera la confiabilidad del mismo.

El presente es un trabajo teórico-práctico en el cual se construye un sistema de 16 kbytes de memoria dinámica con detección y corrección de errores el cual se acopla al microcomputador MEK 6800 D2.

El sistema detecta y corrige errores simples (1 bit errado) y detecta los dobles utilizando para ello el código hamming y cinco bits adicionales de memoria en los que se almacena información generada por circuitos de paridad y que está relacionada con la palabra de datos.

En un ciclo de escritura, los 8 bits de datos, además de grabarse en su memoria correspondiente, ingresan, en una determinada combinación dada por el código hamming, a circuitos de paridad; obteniéndose información que se graba en los cinco bits adicionales de memoria.

Luego, en un ciclo de lectura, la combinación de bits hamming y bits de datos forman una palabra apuntadora que indica al sistema si es que hubo o no error simple, y la posición de este, para que conociendo ya cual es el bit errado, lo corrija por simple inversión.

El error más generalizado que afecta a los sistemas de memoria es producido por radiaciones alfa cuya fuente puede ser el material que conforman los integrados de memoria.

Estos errores al ser de caracter aleatorio se presentan en relaciones muy pequeñas, por ejemplo, para un sistema de 16 kbytes de memoria. la relación es de un error en 5 millones de horas de operación.

En el transcurso de este trabajo, se cubren tópicos como: Causas de errores en circuitos de memoria, en la sección 1-2, se dan algunos fundamentos matemáticos y definiciones que hacen posible desarrollar los códigos de detección de errores. También se cubre la explicación de varios códigos y de varios circuitos de detección y corrección de errores. Sin embargo de tener esta generalidad, el código y el circuito utilizados en el presente trabajo se ilustran con mayor detalle. Esto es: En la sección 1-3 se explica el funcionamiento del código hamming y en la sección 2-4 (CIRCUITO D) se analiza el funcionamiento del circuito construido.

En los capítulos 3 y 4 se dan los detalles para el diseño y construcción del sistema de memoria.

CONTENIDO

Introducción	1
--------------------	---

CAPITULO I

INTRODUCCION A LOS METODOS DE DETECCION Y CORRECCION DE ERRORES.

1-1	Causas de errores en circuitos de memoria	1-1
1-2	Códigos de detección de errores	1-9
	Códigos de chequeo de paridad	1-11
	Definición de peso Hamming	1-15
	Definición de distancia Hamming	1-15
	Definición de distancia mínima	1-16
1-3	Código de paridad Hamming para detección y corrección de errores	1-18
	Código Hamming	1-20
1-4	Otros Códigos	1-31
	Códigos cíclicos	1-31

CAPITULO II

CIRCUITOS DE DETECCION Y CORRECCION DE ERRORES.

2-1	Circuito generador de paridad	2-1
2-2	Circuitos de detección y corrección de errores	2-3
	Circuito A: Detecta y corrige errores simples	2-3
	Circuito B: Utiliza "Software" y "Hardware" para la detección y corrección de errores	2-8
	Circuito C: Detecta y corrige errores simples en palabras de 16 bits	2-12
	Circuito D: Detecta y corrige errores simples y detecta los dobles	2-16

CONTENIDO (Continuación)

CAPITULO III

DISEÑO DE UN SISTEMA DE 16 KBYTES DE MEMORIA CON DETECCIÓN Y CORRECCION DE ERRORES.

3-1	Funcionamiento general del circuito	3-1
3-2	Ciclos de refresco y de lectura/escritura	3-5
	Ciclo de Refresco	3-5
	Ciclo de lectura/escritura	3-3
	Método de refresco de robo	3-11
3-3	Circuitos de indicación de error simple y doble	3-13

CAPITULO IV

CONSTRUCCION.

4-1	Construcción del circuito	4-1
4-1-1	Consideraciones sobre las fuentes de alimentación	4-2
4-1-2	Consideraciones sobre la distribución de las señales de datos y de control	4-3
4-2	Pruebas de funcionamiento	4-4
	Programa de inicialización	4-7
4-3	Mantenimiento	4-11

COMENTARIOS Y CONCLUSIONES.

Apendice A: Hojas de especificaciones

MC 3480	A - 1
MC 3242	A - 17
MCM 4116	A - 23

CONTENIDO (Continuación)

MC 8T26 A	A-31
MC 6871 B	A-37
74 S 280	A-39
74 LS 154	A-42

Apendice B:

Opciones	B-1
Fuente de voltaje	B-2
Configuración externa del equipo	B-4

Apendice C: Microcomputador MEK 6800 D2.

LISTA DE FIGURAS

1-1	Célula de memoria modelo afectada por el choque de una partícula alfa	1-4
1-2	Áreas sensitivas de una célula de memoria RAM dinámica ...	1-5
1-3	Variación de la fuente de voltaje V _{dd} , V _s , errores transitorios por hora	1-7
1-4	Variación de la frecuencia de operación, V _s , errores transitorios por hora	1-8
1-5	Algunos errores en un código de detección de errores	1-10
1-6	Distancias de un código de detección	1-17
1-7	Circuito que divide al polinomio $x^4 + x^3 + x^2 + 1$	1-34
2-1	Circuito A: Detecta y corrige errores simples	2-4
2-2	Circuito B: Utiliza "Software" y "Hardware" para la detección y corrección de errores	2-9
2-3	Circuito C: Detecta y corrige errores simples en palabras de 16 bits	2-13
2-4	Circuito D: Detecta y corrige errores simples y detecta los dobles	2-16
3-1	Diagrama de bloques	3-17
3-2	Diagrama de tiempos	3-18
3-3	Diagrama detallado	3-19
4-1	Distribución de elementos en la tarjeta de memoria	4-14
4-2-A	Contactos en la tarjeta 9610	4-15
4-2-B	Contactos en la tarjeta 9610	4-15

LISTA DE FIGURAS (Continuación)

B1	Fuente de voltaje	B - 3
B2	Configuración externa del equipo	B - 5
C1	Diagrama del microcomputador MEK 6800 D2	C - 3

LISTA DE TABLAS

1-1	Código Hamming para palabras de 8 bits	1-20
1-2	Código Hamming para palabras de 16 bits	1-21
1-3	División del polinomio $x^6 + x^5 + x^4$ para el polinomio $x^4 + x^3 + x^2 + 1$	1-35
4-1	Lista de elementos	4-16

CAPITULO PRIMERO
INTRODUCCION A LOS METODOS DE
DETECCION Y CORRECCION DE ERRORES

1.1. CAUSAS DE ERRORES EN CIRCUITOS DE MEMORIA

El estado lógico de las memorias dinámicas dependen de la carga almacenada en condensadores de las células de memoria. El nivel de carga determina el estado del bit de la célula: (1) lógico o (0) lógico.

La principal razón para que se produzcan errores transitorios en los circuitos de memoria dinámica es la exposición de estos circuitos a radiaciones de partículas alfa (núcleos de Helio 2 Protones y 2 Neutrones; con una masa atómica de 4) que alteran la carga almacenada en la célula produciendo errores en los cuales un bit es cambiado de 1 a 0 o viceversa, sin causar ningún daño permanente al circuito y sin dejar huella que pueda indicar que se ha producido un error.

Estos errores transitorios pueden sin embargo ser detectados y corregidos usando códigos y circuitos de Detección y corrección. Con objeto de reducir al mínimo el número de fallas en circuitos de memoria, se han realizado muchos experimentos y se han elaborado varios modelos que han permitido obtener datos estadísticos y encontrar técnicas para la producción y diseño de los circuitos integrados que mejoran su relación de erros (número de errores/hora de operación).

Por ejemplo la compañía Hughes Aircraft da una especificación de la relación de error de: 1 error en 5 millones de horas de operación en un sistema de 16 kbytes de memoria dinámica.

El flujo alfa, se define como el número de partículas alfa emitidas desde una superficie en la unidad de tiempo y se mide en: Partículas alfa por centímetro cuadrado por hora ($\alpha/\text{cm}^2/\text{h}$).

Las radiaciones alfa pueden encontrarse casi en cualquier parte debido a que provienen de sustancias que se encuentran en la naturaleza. Por ejemplo una tierra típica produce de 1 a 2 $\alpha/\text{cm}^2/\text{h}$.

Prácticamente todos los circuitos integrados de alta tecnología como memorias RAM, ROM o microprocesadores tienen, entre otros elementos que conforman su empaquetadura, cerámica y cuarzo; siendo estos elementos fuentes de partículas alfa. Estudios sobre estos materiales de cerámica nos dan un rango de flujo alfa desde 0.01 $\alpha/\text{cm}^2/\text{h}$. a 100 $\alpha/\text{cm}^2/\text{h}$, sin embargo la radiación alfa efectiva que un integrado experimenta, debido a los materiales de cerámica, encontrados en su empaquetadura es menor que 0.5 $\alpha/\text{cm}^2/\text{h}$.

El Uranio y el Torio son los mayores contaminantes radioactivos que se encuentran en los materiales de empaquetadura de semiconductores. Estos materiales emiten partículas alfa con energías que alcanzan hasta 8.78 millones de electrón voltios.

Estudios realizados muestran que partículas alfa de alrededor de 5 millones de electrón voltios ingresan en el silicio causando la mayoría de errores en memorias dinámicas.

Cuando las partículas alfa golpean y penetran la superficie del silicio, las partículas bajan suavemente y producen alrededor

de un millón de pares electrón - hueco de carga. La penetración es del orden de decenas de micrómetros.

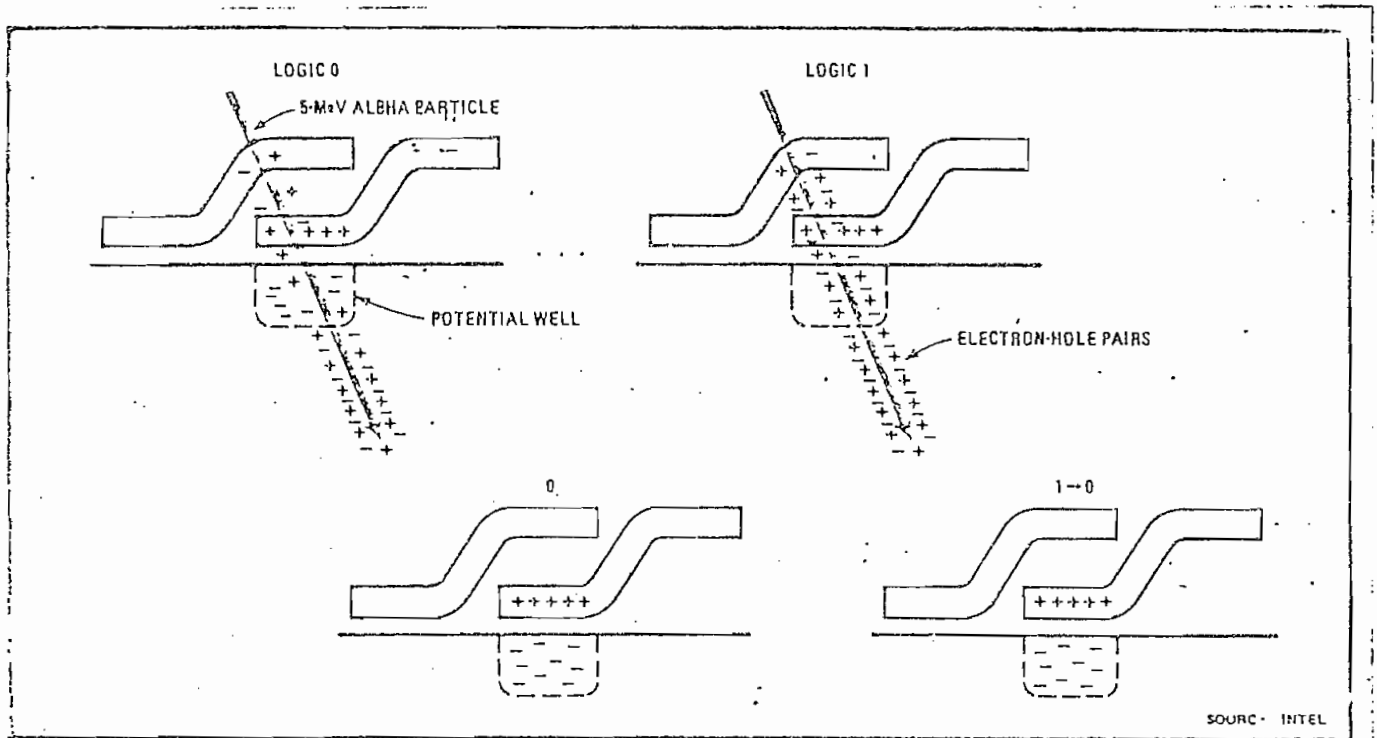
Dos modelos físicos generales se han propuesto para explicar los errores transitorios en memorias dinámicas. Uno de ellos basa su explicación en una célula modelo. En este modelo, la energía generada por una penetración alfa (1.4 millones de pares electrón - hueco generados a una profundidad de 25 μm .) es suficiente para compensar la carga existente en la célula de memoria. La figura 1-1 ilustra este modelo y representa el equivalente de una célula de memoria de un sistema de 16k de RAM Dinámica.

La célula de la figura (1-a) está en estado 0 lógico, su potencial está lleno con electrones.

La célula de la figura (1-b) está en estado 1 lógico con su potencial vacío (lleno de pares electrón - hueco).

Una partícula alfa con una energía de 5 millones de electrón voltios choca en cada célula y penetra a una profundidad de 25 μm . La célula en 0 lógico con su potencial lleno de electrones no es afectada. La célula en 1 lógico con su potencial previamente vacío, se llena con el exceso de carga y el nivel lógico de la célula cambia de 1 a 0.

En este modelo solo unos lógicos pueden ser convertidos en ceros lógicos. Sin embargo experimentos actuales en sistemas de 16k de memoria nos muestran que las partículas alfa pueden cambiar cualquier estado lógico.



1. Penetration model. In an early model for alpha-induced soft errors in dynamic random-access memories, a particle strikes cells in both the logic 0 and 1 state (top). The cell in a 0 state is unchanged, but the 1-state cell's potential well is filled, changing its state to 0 (bottom).

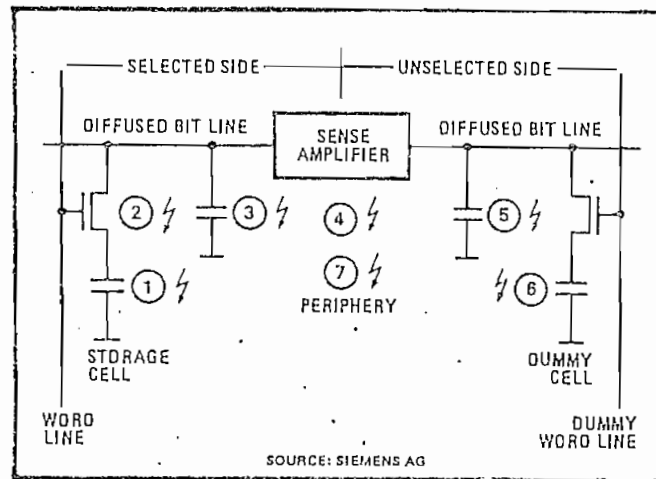
Fig. 1 - 1

Otro modelo basa su análisis en las áreas sensitivas de la célula de memoria en las que las partículas alfa chocan.

La figura 1 - 2 es una representación de las áreas sensitivas de la célula a partículas alfa, y es una representación del circuito simplificado equivalente de una célula de memoria RAM dinámica.

Dependiendo del lugar de la célula en los que chocan las partículas alfa, se pueden generar errores de 1 a 0 ó de 0 a 1 o ambos tipos de errores.

Fig. 1.2.



2. Line hits. The numbered dynamic-RAM locations have been shown to be sensitive to alpha-particle hits. Depending on the location, either type of binary error can be generated: bit-line hits on the selected side (3), for example, can cause erroneous 1s.

Fig. 1 - 2

Para medir la relación de error en memorias RAM dinámicas, un buen método consiste en acelerar los efectos de error en la memoria causado por radiaciones alfa.

Esto implica que el sistema de memoria se debe exponer a radiaciones alfa más fuertes que las radiaciones producidas por residuos en la empaquetadura del integrado.

Luego, la relación obtenida es extrapolada a los niveles de radiación alfa de la empaquetadura. La principal ventaja de este método es que todo el test se lo puede efectuar en pocas horas.

Para este test se usa una fuente con un fluido alfa que varía desde 50.000 a 50'000.000 $\alpha/cm^2/h.$, siendo de vital importancia la calibración de la intensidad del fluido alfa que choca contra

el circuito integrado, debido a que cualquier cálculo de interpolación depende de la precisión del punto inicial.

Una de las más importantes ventajas de los chequeos y mediciones de los errores transitorios es que permite variar los parámetros de la RAM que afectan dichos errores.

Dos de las más importantes variables son la fuente de voltaje ($v_{dd} = 12v$) y los ciclos de tiempo.

La fuente de voltaje controla la carga absoluta en una célula de RAM dinámica. Este voltaje también determina la diferencia de voltaje entre la célula y la célula de comparación (dummy) usada para determinar el estado de la célula durante ciclos de lectura y refresco.

El efecto de la variación de la fuente de voltaje v_{dd} se muestra en la figura 1-3. En un sistema de 16k de memoria RAM dinámica se ha variado la fuente v_{dd} desde su valor nominal 12 voltios a 9.6v en cuatro etapas. Los resultados de este test muestran que una reducción del 20% en la fuente de voltaje aumenta los errores en un orden de magnitud.

De esto se puede concluir que el aumento de la fuente de voltaje, aumenta la resistencia, a las partículas alfa, de la célula de memoria y de la célula de comparación.

Figura 1-3.

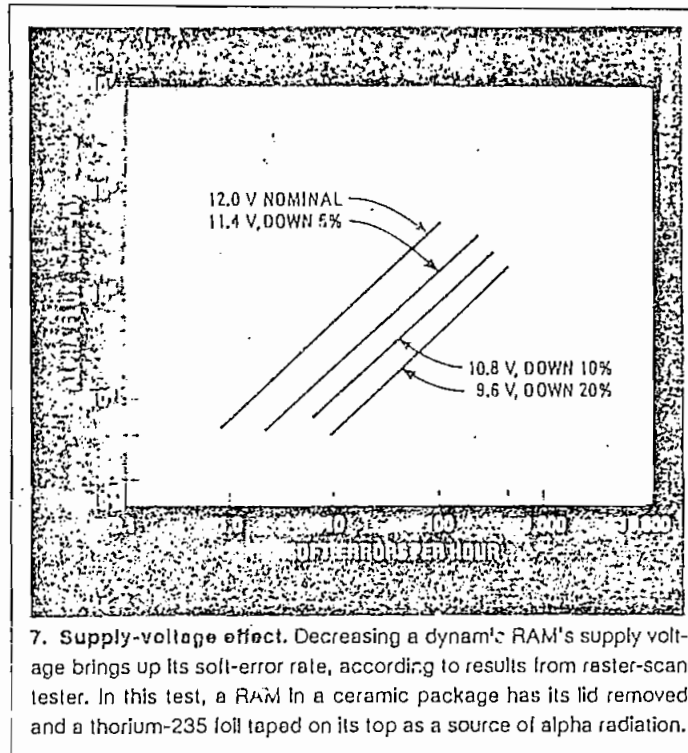


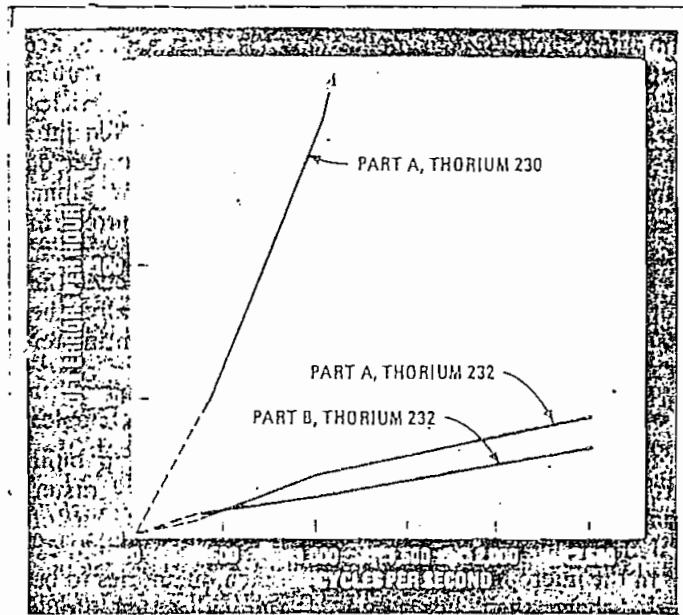
Fig. 1 - 3

Cada ciclo de refresco, lectura o escritura abre la posibilidad de que una partícula alfa pueda causar errores. De esta forma los ciclos de tiempo influyen en la relación de error. Para verificar esto se usaron dos diferentes fuentes de partículas alfa y se variaron los ciclos de tiempo de la RAM.

Un tipo de RAM se expuso a Torio 232 con un fluido alfa de $50.000 \alpha/\text{cm}^2/\text{h}$, y el segundo tipo de RAM fue expuesto a Torio 232 y Torio 230 ($273.000 \alpha/\text{cm}^2/\text{h}$).

Los resultados de este test, tal como se los muestra en la figura 1-4, indicaron que la relación de error disminuye con el

decrecimiento de la frecuencia de operación.



8. Varying cycle time. Results plotted for two dynamic RAM types show that slowing the rate of read, write, and refresh cycling reduces the RAM's soft-error rate. RAMs of type A were subjected to two levels of radiation from different sources in this test.

Fig. 1 - 4

La documentación anterior muestra como: aumentando el valor de la fuente de voltaje y disminuyendo la frecuencia de operación se disminuye la relación de error. Hay sin embargo otros factores como la empaquetadura, diseño y procesamiento de los circuitos integrados que también se consideran para reducir al mínimo los errores transitorios.

1.2. CODIGOS DE DETECCION DE ERRORES

Cuando se produce un error o una falla en un sistema, esta, no tiene un efecto inmediato porque sencillamente pudo ser que el banco o módulo donde se produjo la falla no estaba siendo ocupado.

Después de que la falla o el error hacen su efecto en el sistema, transcurre un período de tiempo hasta ser detectado, a este tiempo se lo llama: Tiempo Medio para Detección. En todos los sistemas confiables, existe la tendencia de reducir este tiempo a cero gracias a un hardware incluido en el sistema. Y esto es debido a que si no existe en el sistema un mecanismo que detecte fallas en un tiempo mínimo, se corre el peligro de que el error sea detectado cuando ya haya producido una catástrofe, y esta sea precisamente la que nos de la pauta para identificar esa falla.

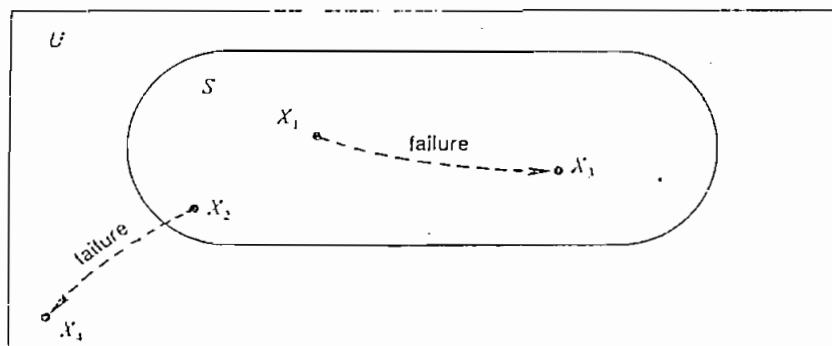
Luego de la detección, se tiene la diagnóstico y la reparación. Al tiempo medio entre la detección de un error y su reparación también se lo trata de reducir al máximo diseñando el sistema a base de módulos separados capaz de poder reemplazar módulos enteros y luego comenzar con la reparación particular de ese módulo.

Los códigos de detección de errores nos permiten detectar un error antes de que éste tenga su efecto en el sistema reduciendo al mínimo el tiempo medio para la detección.

Un código de detección de errores es un conjunto S de un conjunto universo U de vectores escogidos de tal manera que las fallas que afectan a los vectores X que están en S , producen vectores X' que no están en S .

Una palabra código es un vector en S ; una palabra no código es un vector que está en $U - S$.

Si X es una palabra código ($\Rightarrow X \in S$) y X' es un vector diferente producido por una falla, entonces X' es un error detectable si es una palabra no código ($\Rightarrow X' \in U - S$); y X' es un error no detectable si es una palabra código ($\Rightarrow X' \in S$). Estos conceptos se ilustran en la figura 1-5.



code words: X_1, X_2, X_3

noncode word: X_4

detectable error: $X_2 \rightarrow X_4$

undetectable error: $X_1 \rightarrow X_3$

Figure 2.1 Some errors in an error-detecting code.

Fig. 1 - 5

El peso de un error puede ser asociado con el número de fallas o bits errados simultáneos distintos necesarios para producir ese error, mientras que la distancia entre palabras código describe el número de fallas necesitadas para cambiar una palabra

código en otra y por lo tanto producir un error no detectable.

La información en un código de detección de errores es interpretada por un decodificador que distingue entre palabras código y palabras no código.

El decodificador puede simplemente notificar al sistema subsiguiente dentro de un sistema total que una palabra no código ha sido recibida, logrando de esta forma la detección de error.

También es posible para el decodificador intentar asociar una palabra no código con la palabra código transmitida que fue la más parecida antes del error, en cuyo caso la corrección de errores ha tomado lugar.

La política de detección o corrección de errores para un código particular depende solamente del diseño del decodificador y no del código en sí.

CODIGOS DE CHEQUEO DE PARIDAD. La información en un código de chequeo de paridad es codificada en vectores de un campo finito $GF(q)$ con q elementos (1).

Aunque existen códigos para cualquier valor de q solo nos interesa los que se forman con $q=2$ que representan códigos binarios,

Hay q^n vectores diferentes de la forma $X=(X_1, \dots, X_n)$, $X_i \in GF(q)$;

(1) Un campo finito es una estructura algebraica con un conjunto finito de elementos, una operación suma, y una operación multiplicación. El conjunto de números reales, la suma real y la multiplicación real forman un campo finito.

sin embargo, solamente un subconjunto S de q^k ($k < N$) vectores son palabras código.

Este subconjunto S se llama código (n,k) . La fracción k/n se llama la relación del código.

Expliquemos estos conceptos con un ejemplo:

Sea $q=2 \Rightarrow GF(q)$ tiene dos elementos (0 y 1), representando un código binario.

Sea $n=4 \Rightarrow$ existen $q^n = 2^4 = 16$ vectores diferentes de la forma $X = (X_1, X_2, X_3, X_4)$, siendo el $X_i = 0$ ó $X_i = 1$.

$X_0 = 0 0 0 0$	$X_8 = 1 0 0 0$
$X_1 = 0 0 0 1$	$X_9 = 1 0 0 1$
$X_2 = 0 0 1 0$	$X_{10} = 1 0 1 0$
$X_3 = 0 0 1 1$	$X_{11} = 1 0 1 1$
$X_4 = 0 1 0 0$	$X_{12} = 1 1 0 0$
$X_5 = 0 1 0 1$	$X_{13} = 1 1 0 1$
$X_6 = 0 1 1 0$	$X_{14} = 1 1 1 0$
$X_7 = 0 1 1 1$	$X_{15} = 1 1 1 1$

Sea $k = 3 \Rightarrow$ solo $q^k = 2^3 = 8$ vectores son palabras código. S es un subconjunto formado por estos 8 vectores, y se llama código $(4,3)$. La relación del código es $3/4$.

Los elementos de $GF(q)$ pueden combinarse en una operación suma, denotada por "@", o en una operación multiplicación, denotada por yuxtaposición.

Para $q = 2$ (elementos binarios 0,1) estas operaciones son respectivamente "OR EXCLUSIVO" y "AND".

Los vectores en una operación suma se combinan de la siguiente manera:

$$\text{Sea } X = (X_1, \dots, X_n)$$

$$Y = (Y_1, \dots, Y_n)$$

$$X \oplus Y = (X_1 \oplus Y_1, \dots, X_n \oplus Y_n)$$

La operación OR EXCLUSIVO es una suma en módulo 2:

$$0 \oplus 0 = 0$$

$$0 \oplus 1 = 1$$

$$1 \oplus 0 = 1$$

$$1 \oplus 1 = 0$$

y para GF(2) la suma \oplus es igual a la resta \ominus

Ejemplo:

$$X = (0, 0, 1, 1)$$

$$Y = (0, 1, 0, 1)$$

$$X \oplus Y = (0, 1, 1, 0)$$

También se define la multiplicación de un vector por un escalar de la siguiente manera:

$$cX = (cX_1, \dots, cX_n)$$

La operación AND es tal que:

$$0 \text{ AND } 0 = 0$$

$$0 \text{ AND } 1 = 0$$

$$1 \text{ AND } 0 = 0$$

$$1 \text{ AND } 1 = 1$$

Ejemplo:

$$X = (0, 1, 0, 1)$$

$$c = 0$$

$$cX = (0, 0, 0, 0)$$

DEFINICION. El peso hamming de un vector X , denotado por $w(X)$, es el número de componentes diferentes de cero de X .

DEFINICION. La distancia Hamming entre dos vectores X y Y , denotada por $d(X,Y)$, es el número de componentes en los cuales ellos difieren.

Por ejemplo, considérense los vectores de cuatro componentes binarios, $X = (1,0,1,1)$ y $Y = (0,1,1,0)$. Es claro ver que $w(x)=3$, $w(y)=2$, y $d(x,y)=3$.

Con las definiciones dadas demostraremos que para algún X y Y , $d(x,y)=w(X \oplus Y)$.

Sea $X=(x_1, x_2, x_3, \dots, x_n)$ y

$Y=(y_1, y_2, y_3, \dots, y_n)$

entonces $X \oplus Y = (x_1 \oplus y_1, x_2 \oplus y_2, x_3 \oplus y_3, \dots, x_n \oplus y_n)$

Puesto que el $X_i \oplus Y_i$ es 1 solo cuando el $X_i \neq Y_i$

(debido a la definición de la operación OR EXCLUSIVO)

El número de 1s de $X \oplus Y$ (definición de $w(X \oplus Y)$) es igual al número de componentes en los que X y Y difieren que corresponde a la definición de $d(X,Y)$ quedando así demostrado que $d(X,Y) = w(X \oplus Y)$

Para el ejemplo dado:

$$d(X,Y) = w(X \oplus Y) = w((1,1,0,1)) = 3$$

Cuando un error ha ocurrido, una palabra código X se transforma en un vector X' que lo podemos escribir como $X' = X \oplus E$ donde E

es un vector error diferente de cero. El peso de un error es el peso hamming de E . Por ejemplo, un error que transforma $X=(1,0,1,1)$ en $X' = (1,0,0,1)$ tiene el vector error $E = (0,0,1,0)$; el peso del error es 1.

Un error con peso 1 es referido como un error simple; un error con peso 2 es un error doble y así sucesivamente.

El número de errores que pueden ser detectados por un código está íntimamente relacionado con la distancia mínima del código, que se define a continuación.

DEFINICION. La distancia mínima de un código S es la mínima de las distancias hamming entre todos los posibles pares de palabras código en S .

Un código con distancia mínima $d + 1$ puede detectar todos los errores con peso d o menos. Un error no es detectable solamente si cambia alguna palabra código X en otra palabra código Y . Debido a que las palabras código están separadas por lo menos una distancia $d + 1$, se tendrá un error no detectable, solamente si el peso del error es $d + 1$ o más, como se muestra en la figura 1-6 para un código de 3 bits con $d = 1$ (distancia mínima = 2).

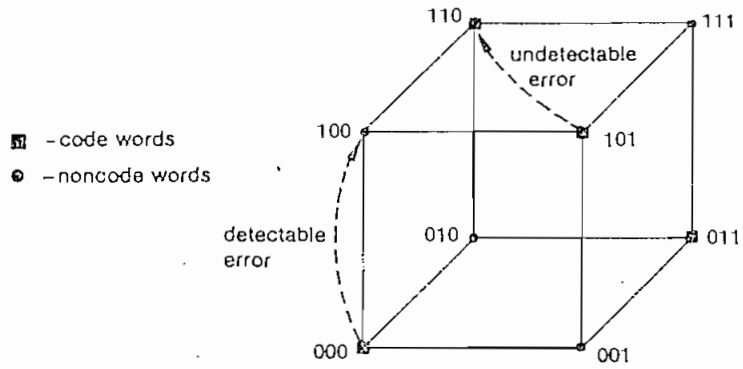


Figure 2.2 A distance-2 single-error-detecting code.

Fig. 1 - 6

1.3. CODIGO DE PARIDAD HAMMING PARA DETECCION Y CORRECCION DE ERRORES

El código Hamming fue publicado en 1950 y debe su nombre a su creador. Ha sido generalmente empleado en sistemas con pocos errores y resulta ideal en sistemas de memoria.

Para comprender mejor su funcionamiento, analicemos el problema con un enfoque real, particular para el presente trabajo.

Disponemos de un microcomputador con 8 bits de datos. La función del código a utilizarse es la de monitorear estos bits de alguna dirección dada en cada ciclo de lectura y escritura.

En el caso de que se produzca algún error simple (un solo bit errado) el sistema lo detecta y lo corrige automáticamente. Si se produce un error doble (dos bits errados) el sistema tan solo lo detecta, no lo corrige, y da una indicación de que ello ha ocurrido, encendiendo un led de indicación de error doble.

La probabilidad de no tener más de 2 errores puede ser razonablemente asumida para muchos sistemas.

Si trabajamos con bytes de 8 bits, un código binario de 4 bits resulta más que suficiente para detectar errores simples; pues con 4 bits se pueden designar 2^4 (16) posiciones diferentes.

Sin embargo como nuestro interés es también detectar errores dobles, necesitamos un bit adicional, requiriendo de este modo un total de 5 bits para el funcionamiento del código. A estos bits del código los llamaremos bits Hamming.

Para detectar y corregir errores simples el código debe tener una distancia mínima de 3. Adicionalmente si deseamos detectar

errores dobles el código debe tener una distancia mínima de 4.

Los bits hamming se los obtiene de la paridad par o impar de una relación determinada de los bits del byte, por esta razón también se los llama bits de paridad.

Existe la paridad para (even) y la paridad impar (odd).

El bit paridad en el caso de paridad impar es tal que el número de "unos" en la combinación byte-bit paridad es siempre impar. Así por ejemplo, si el byte es: 00110101 el bit paridad impar debe ser 1, en esa forma el número total de "unos" es impar. Si el byte fuese 11000111, el bit paridad impar debe ser 0.

El bit paridad en el caso de paridad par es tal que el número de "unos" en la combinación byte - bit paridad es siempre para. Así por ejemplo, un byte 00110101 produce un bit paridad 0 y un byte 11000111 produce un 1.

En esa forma el número total de unos en cada una de las combinaciones es siempre par.

Para generar un bit paridad, se suman en módulo dos los bits del byte. Esta suma en módulo dos corresponde a la operación OR EXCLUSIVO que la definimos en la sección 1-2, y recordando para esta operación se cumple que: $0 \oplus 0 = 0$; $0 \oplus 1 = 1$; $1 \oplus 0 = 1$; $1 \oplus 1 = 0$;

La suma en módulo dos de alguna combinación de los bits del byte es igual al "bit paridad par" de esa combinación y la misma suma más uno es el bit "paridad impar".

La suma en módulo dos más uno, corresponde al complemento de la operación OR EXCLUSIVO, denotándose como NOR EXCLUSIVO. Los símbolos para las operaciones OR y NOR EXCLUSIVO son respectivamente \oplus y $\bar{\oplus}$.

Las compuertas que los identifican son: $\Rightarrow \text{D}$ y $\Rightarrow \text{D}$ respectivamente.

CODIGO HAMMING.

El trabajo del código Hamming se explica en la tabla 1-1.

TABLA 1-1

Contenido del Bit		H0	H1	D0	H2	D1	D2	D3	H3	D4	D5	D6	D7
Posición del Bit		1	2	3	4	5	6	7	8	9	10	11	12
PALABRA APUNTADORA	P0 →	0	1	0	1	0	1	0	1	0	1	0	1
	P1 →	0	0	1	1	0	0	1	1	0	0	1	1
	P2 →	0	0	0	0	1	1	1	1	0	0	0	0
	P3 →	0	0	0	0	0	0	0	0	1	1	1	1

P0 =	H0	\oplus	D0	\oplus	D1	\oplus	D3	\oplus	D4	\oplus	D6
P1 =	H1	\oplus	D0	\oplus	D2	\oplus	D3	\oplus	D5	\oplus	D6
P2 =	H2	\oplus	D1	\oplus	D2	\oplus	D3	\oplus	D7		
P3 =	H3	\oplus	D4	\oplus	D5	\oplus	D6	\oplus	D7		

CODIGO HAMMING APLICADO A UNA PALABRA DE 8 BITS.

La tabla 1-2 explica el trabajo del mismo código para palabra de 16 bits.

T A B L A N° . 1 - 2

Contenido de Bit	H0	H1	D0	H2	D1	D2	D3	H3	D4	D5	D6	D7	D8	D9	D10	H4	D11	D12	D13	D14	D15
Posición del Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
$P_0 \rightarrow$	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$P_1 \rightarrow$	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
$P_2 \rightarrow$	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1
$P_3 \rightarrow$	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0
$P_4 \rightarrow$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

PALABRA
APUNTADORA

SCLO PROCESADORES DE 16 BITS.

P_0 H0 ⊕ D0 ⊕ D1 ⊕ D2 ⊕ D3 ⊕ D4 ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D8 ⊕ D9 ⊕ D10 ⊕ D11 ⊕ D12 ⊕ D13 ⊕ D14 ⊕ D15
 P_1 H1 ⊕ D0 ⊕ D2 ⊕ D3 ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D8 ⊕ D9 ⊕ D10 ⊕ D12 ⊕ D13
 P_2 H2 ⊕ D1 ⊕ D2 ⊕ D3 D7 ⊕ D8 ⊕ D9 ⊕ D10 ⊕ D14 ⊕ D15
 P_3 H3 ⊕ D4 ⊕ D5 ⊕ D6 ⊕ D7 ⊕ D8 ⊕ D9 ⊕ D10
 P_4 H4 ⊕ D11 ⊕ D12 ⊕ D13 ⊕ D14 ⊕ D15

La fila superior identificada como "CONTENIDO DEL BIT", muestra la estructura de una palabra de 8 bits de datos en la cual se han insertado los bits hamming. Los bits H_i son los bits paridad o hamming, y los 8 D_i son los bits de datos.

La segunda fila "POSICION DEL BIT" contiene números en base 10 que nos dan la posición ordinal de los bits.

Abajo de cada número está una columna que contiene un número binario de 4 bits el cual es igual al número de la posición del bit, y se la llama "PALABRA APUNTA DORA". (El bit superior P_0 es el menos significativo, y el bit inferior P_3 es el más significativo).

Las ecuaciones hamming $P_0 - P_1 - P_2 - P_3$ mostradas también en la tabla 1 son combinaciones OR EXCLUSIVO de los bits de datos y los bits hamming. Nótese que cada bit hamming H_i está presente solo en una de las ecuaciones y está asociado con una combinación diferente de los bits de datos en su ecuación.

Los bits hamming h_1-h_3 son generados como la paridad par de los bits de datos en su ecuación respectiva esto es: el bit hamming h_i será 0 o 1 de tal manera que el número total de "unos" en la respectiva ecuación sea par.

De esta manera en un ciclo de lectura de memoria los bits hamming producen en su ecuación correspondiente un 0 si no hay error o un 1 si los bits de datos o los bits hamming están errados. En este último caso las cuatro ecuaciones P_j tomadas en conjunto producen

una de las 12 palabras apuntadoras de 4 bits de la tabla 1. En otras palabras, cualquier error simple (un solo bit errado) produce una palabra apuntadora de 4 bits que indica la posición del bit errado y este conocimiento permite activar ya sea hardware (nuestro caso) o Software para corregir el bit errado por simple inversión.

Para insertar los bits hamming en la palabra de datos, se usa un orden estandar e invariante que no depende de la longitud de la palabra. Así, se nota que las posiciones de los bits hamming corresponden a los números obtenidos de la serie: $2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$. De esta manera, si tuviéramos una palabra de más bits, las posiciones de los bits hamming serían $2^4 = 16$, $2^5 = 32$, etc.

Esta invariabilidad tiene dos importantes y elegantes consecuencias. Primero, permite arreglar las ecuaciones, de tal manera que cuando un bit está errado, ellos producen el equivalente binario de la posición ordinal de ese bit en la palabra. Segundo, permite parar la secuencia de bits hamming y bits de datos (y las ecuaciones acompañantes) hasta el punto en el que ya se han incluido todos los bits de datos de la palabra; de esta manera usamos el menor número posible de bits hamming necesarios para detectar 1 bit errado.

Para detectar un error doble se añade al procedimiento un simple chequeo de paridad sobre todos los bits -hamming y datos- requiriéndose para esto un bit extra de paridad. La ecuación correspondiente es:

$$P_p = H_p \oplus H_0 \oplus H_1 \oplus D_0 \oplus H_2 \oplus D_1 \oplus D_2 \oplus D_3 \oplus H_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7$$

En igual forma que en las otras ecuaciones, H_p será 0 o 1 para lograr que el número total de "unos" en el segundo miembro de la ecuación, sea par. De esta manera, en un ciclo de lectura de memoria, si $P_p = 1$ implica que se ha producido un número impar de errores en los bits hamming o de datos, si $P_p = 0$ implica que se han producido cero errores o un número par de errores.

Un bit errado en una palabra puede ser detectado y corregido, un error doble (2 bits errados) puede ser solamente detectado (no corregido) y la probabilidad de no tener más de 2 bits errados, puede ser razonablemente asumida para muchos sistemas. Sin embargo es necesario indicar que si se producen más de dos errores simultáneos, el sistema falla completamente, necesitándose para estos casos, códigos con mayor distancia mínima.

Antes de analizar algunos ejemplos, veamos el proceso de escritura y lectura; tanto de los bits de datos como de los bits hamming; en la memoria del computador.

ESCRITURA. Supongamos que una palabra de 8 bits de datos debe ser escrita en una localidad determinada de memoria. Previamente a esto, debemos obtener los bits hamming de acuerdo a las ecuaciones:

$$P_0 = H_0 \oplus D_0 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6$$

$$P_1 = H_1 \oplus D_0 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6$$

$$P_2 = H_2 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_7$$

$$P_3 = H_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7$$

$$P_p = H_0 \oplus H_1 \oplus D_0 \oplus H_2 \oplus D_1 \oplus D_2 \oplus D_3 \oplus H_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus H_p$$

Para las ecuaciones: P_0, P_1, P_2, P_3 ; los H_i (para escritura) son cero; obteniéndose P_0, P_1, P_2, P_3 . Luego, estos valores los reemplazamos en H_0, H_1, H_2, H_3 , logrando de esta manera que en cada ecuación el número total de "1" de la combinación datos - bit paridad sea par.

Para P_p la regla es la misma considerando que H_0, H_1, H_2, H_3 tienen el valor encontrado anteriormente.

De esta manera, se escribe en la memoria del computador los bits de datos y los 5 bits Hamming.

LECTURA. Supongamos que vamos a leer de la misma localidad de memoria analizada en la escritura. Se leen los bits de datos D_i y los bits hamming H_0, H_1, H_2, H_3, H_p . De las ecuaciones Hamming obtenemos P_0, P_1, P_2, P_3, P_p valores con los cuales el sistema puede saber la condición del error.

El mismo proceso se verá en forma circuital en el capítulo 2.

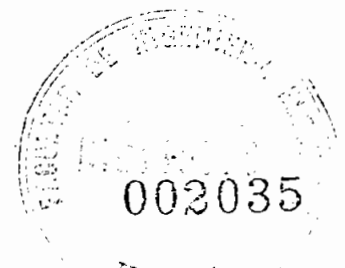
EJEMPLOS

- Supongamos que la palabra:

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	0

debe ser escrita en la memoria del computador.

El circuito hamming, en base a las ecuaciones correspondientes obtiene los bits:



$$H0 = D0 \oplus D1 \oplus D3 \oplus D4 \oplus D6$$

$$H0 = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$H1 = D0 \oplus D2 \oplus D3 \oplus D5 \oplus D6$$

$$H1 = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$H2 = D1 \oplus D2 \oplus D3 \oplus D7$$

$$H2 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$H3 = D4 \oplus D5 \oplus D6 \oplus D7$$

$$H3 = 0 \oplus 0 \oplus 0 \oplus 1 = 1$$

$$H_p = H0 \oplus H1 \oplus H2 \oplus H3 \oplus D0 \oplus D1 \oplus D2 \oplus D3 \oplus D4 \oplus D5 \oplus D6 \oplus D7$$

para H0, H1, H2 y H3 usamos los valores ya tabulados

$$H_p = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 0$$

Los valores de los Hi así encontrados, hacen de acuerdo a las ecuaciones que los Pi sean cero.

De esta manera, la palabra que se escribe en la memoria del computador en una localidad determinada es:

$$H0 \ H1 \ D0 \ H2 \ D1 \ D2 \ D3 \ H3 \ D4 \ D5 \ D6 \ D7 \ H_p.$$

$$0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0$$

Ahora cuando leemos de esa localidad de memoria, es probable que, por alguna de las razones expuestas en la sección 1-1, uno o más bits estén errados. Analicemos algunas posibilidades.

A. De la memoria del computador se obtiene la palabra:

$$H0 \ H1 \ D0 \ H2 \ D1 \ D2 \ D3 \ H3 \ D4 \ D5 \ D6 \ D7 \ H_p$$

$$0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0$$

Del circuito hamming se obtienen:

$$P0=H0 \oplus D0 \oplus D1 \oplus D3 \oplus D4 \oplus D6 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$P1=H1 \oplus D0 \oplus D2 \oplus D3 \oplus D5 \oplus D6 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$P2=H2 \oplus D1 \oplus D2 \oplus D3 \oplus D7 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$P3=H3 \oplus D4 \oplus D5 \oplus D6 \oplus D7 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 0$$

$$Pp=H0 \oplus H1 \oplus D0 \oplus H2 \oplus D1 \oplus D2 \oplus D3 \oplus H3 \oplus D4 \oplus D5 \oplus D6 \oplus D7 \oplus Hp$$

$$Pp= 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0$$

$$Pp= 0$$

El valor de $Pp=0$, nos indica que no hay error o existe un número par de errores.

La palabra apuntadora = $(P3 P2 P1 P0) = (0 0 0 0)$ nos da la posición ordinal 0, que nos indica que no hay bits errados.

En efecto, al comparar la palabra leída desde la memoria del computador y la palabra escrita en la misma, comprobamos que son exactamente iguales.

B. De la memoria del computador se obtiene la palabra:

$$H_0 \ H_1 \ D_0 \ H_2 \ D_1 \ D_2 \ D_3 \ H_3 \ D_4 \ D_5 \ D_6 \ D_7 \ H_p$$

$$0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0$$

=

Del circuito hamming se obtienen:

$$P0=H_0 \oplus D_0 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$P1=H_1 \oplus D_0 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$P2=H_2 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_7 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$$

$$P3=H_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7 = 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$Pp=H_0 \oplus H_1 \oplus D_0 \oplus H_2 \oplus D_1 \oplus D_2 \oplus D_3 \oplus H_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus H_p$$

$$Pp = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1$$

El valor del $P_p=1$ nos indica que existe un número impar de errores. La palabra apuntadora es $(P_3 P_2 P_1 P_0) = (1 0 1 0)$ que en base 10, representa el número 10. Por lo tanto, la posición ordinal en la que existe un error es la 10 que corresponde al bit D5.

Al comparar la palabra escrita y leída a y desde la memoria del computador, comprobamos que en efecto el bit D5 está errado.

C. De la memoria del computador se obtiene la palabra:

H0	H1	D0	H2	D1	D2	D3	H3	D4	D5	D6	D7	Hp
0	1	0	0	1	0	1	1	0	0	0	1	0
=												

Del circuito hamming se obtienen:

$$P_0 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$P_1 = 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$P_2 = 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$P_3 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 = 0$$

$$P_p = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0$$

$$P_p = 1$$

El valor de $P_p=1$ nos indica que existe un número impar de errores. La palabra apuntadora es $(P_3 P_2 P_1 P_0) = (0 1 0 0)$ que en base 10 representa el número 4. Por lo tanto, la posición ordinal en la que existe un error es la 4 que corresponde al bit H2.

Al comparar la palabra escrita y leída a y desde la memoria del computador, comprobamos que en efecto el bit H2 está errado.

D. De la memoria del computador se obtiene la palabra:

H0	H1	D0	H2	D1	D2	D3	H3	D4	D5	D6	D7	Hp
0	1	0	1	0	0	1	1	0	0	1	1	0
				=						=		

Del circuito hamming se obtienen:

$$P0 = 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$P1 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$P2 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$P3 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$Pp = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

El valor de $Pp=0$ indica que no hay error o que existe un número par de errores. Pero puesto que la palabra apuntadora ($P3 P2 P1 P0$) $= (1 1 1 0)$ no es cero, se concluye que existe un número par de errores y no se puede determinar la posición de los bits errados.

E. De la memoria del computador se obtiene la palabra:

H0	H1	D0	H2	D1	D2	D3	H3	D4	D5	D6	D7	Hp
0	1	0	1	1	1	1	1	0	0	1	0	0
				=						=	=	

Del circuito hamming se obtienen:

$$P0 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$P1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$P2 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$P3 = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$Pp = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

El valor del $Pp=1$ indica que existe un número impar de errores. La palabra apuntadora ($P3 P2 P1 P0$) $= (0 0 0 1)$, que en base 10 representa el número 1, nos da la posición ordinal del bit H0. Al comparar la palabra escrita y leída a y desde la memoria del

computador, comprobamos que el bit H0 no está errado y que en cambio los bits D2, D6, D7 si están errados; habiéndose producido un error triple que pasó a través del sistema completamente inadvertido, y para este ejemplo, produjo un error adicional al invertirse el bit H0.

Pero, como ya lo explicamos anteriormente, la probabilidad de tener más de dos errores, resulta muy pequeña.

1.4. OTROS CODIGOS

CODIGOS CICLICOS

Un código cíclico es un código de chequeo de paridad con la propiedad de que cualquier desplazamiento cíclico de una palabra código es también una palabra código.

Estos códigos tienen una estructura matemática muy pulida que hace que la codificación y la decodificación sea muy sencilla usando registros de desplazamientos de realimentación lineales, y por lo tanto son usados muy a menudo para chequeos de datos seriales.

Sea V un vector con n componentes comenzando con $n-1$, esto es:

$$V = (v_{n-1}, v_{n-2}, \dots, v_0)$$

Un desplazamiento cíclico de v hacia la izquierda en una posición se denota por $L(V)$, donde

$$L(V) = (v_{n-2}, \dots, v_i, v_0, v_{n-1})$$

Un desplazamiento de i lugares se denota por $(L^i(V))$, con la convención de que $L^0(V) = V$.

Las operaciones de vectores que usamos en secciones previas nos permiten combinar vectores pero no nos dan una manera conveniente de operar con los desplazamientos cíclicos de un vector. Para hacerlo, escribiremos un polinomio $V(x)$ de grado $n-1$ o menos correspondiendo a cada componente n del vector V . Los coeficientes de $V(x)$ corresponden a los coeficientes del vector V .

$$V(x) = (v_{n-1} x^{n-1} + v_{n-2} x^{n-2} + \dots + v_1 x + v_0)$$

El polinomio correspondiente a $L(V)$ es:

$$L(V(x)) = v_{n-2}x^{n-1} + \dots + v_0x + v_{n-1}$$

Si V es una palabra código, entonces el polinomio correspondiente a $V(x)$ se llama un código polinomial.

Debido a que por definición un código cíclico binario es un código de chequeo de paridad, tiene 2^k palabras código para algún $k < n$.

En un código cíclico binario (n,k) , (n =número de componentes del vector) existe un único código polinomial diferente de cero $G(x)$ de grado mínimo $n-k$. Cada código polinomial es múltiplo de $G(x)$, y cada múltiplo de $G(x)$ de grado $n-1$ o menos es un código polinomial. Ejemplo:

$$V(x) = M(x) G(x)$$

$V(x)$ es múltiplo de $G(x)$

Ahora, sea $V(x)$ algún código polinomial. Si dividimos $V(x)$ para $G(x)$, obtenemos un cociente $A(x)$ y un residuo $B(x)$. Esto es:

$$V(x) = A(x) G(x) + B(x)$$

donde $A(x)$ es algún polinomio y $B(x)$ tiene un grado menor que $G(x)$. Debido a que $V(x)$ es un código polinomial y $A(x) G(x)$, (un múltiplo de $G(x)$), es un código polinomial, entonces $B(x)$ debe ser un código polinomial. Pero $B(x)$ tiene un grado menor que $G(x)$, y $G(x)$ tiene el grado más pequeño de un código polinomial diferente de cero. Por lo tanto $B(x)$ debe ser cero y $V(x)$ es un múltiplo de $G(x)$ así:

$$V(x) = A(x) G(x)$$

resultando $V(x)$ divisible por $G(x)$

Al polinomio $G(x)$ se lo llama: Generador polinomial

La codificación y decodificación de un código cíclico requiere de una división por $G(x)$. Para codificar un mensaje dividimos $x^r M(x)$ por $G(x)$ y usamos el inverso del residuo como los bits de chequeo. Para la decodificación, dividimos el polinomio recibido $V(x)$ para $G(x)$; si el residuo no es cero, $V(x)$ no es un código polinomial debido a que no es un múltiplo de $G(x)$, y de esa forma se ha detectado un error. De esta forma el codificador y el decodificador pueden compartir el mismo circuito divisor.

La división para un polinomio arbitrario $G(x)$ puede ser lograda con registros de desplazamiento de realimentación lineal mostrados en la figura 1-7.

Cada cuadrado representa una célula de almacenamiento y cada "0" representa un sumador. Para $GF(2)$, elementos binarios, las células de almacenamiento son flip flops tipo D y los sumadores son compuertas OR EXCLUSIVO.

El circuito divide un polinomio $P(x)$ para $G(x)$. El coeficiente de orden más alto de $P(x)$ llega primero al circuito con el primer pulso de reloj, el segundo coeficiente con el segundo pulso y así sucesivamente hasta que el último coeficiente ha entrado al circuito.

El cociente ($Q(x)$) se produce en la salida serial con el primer coeficiente primero.

Los registros de desplazamiento son inicialmente puestos en cero y al final del proceso después de n pulsos de reloj estos contienen el residuo de la división.

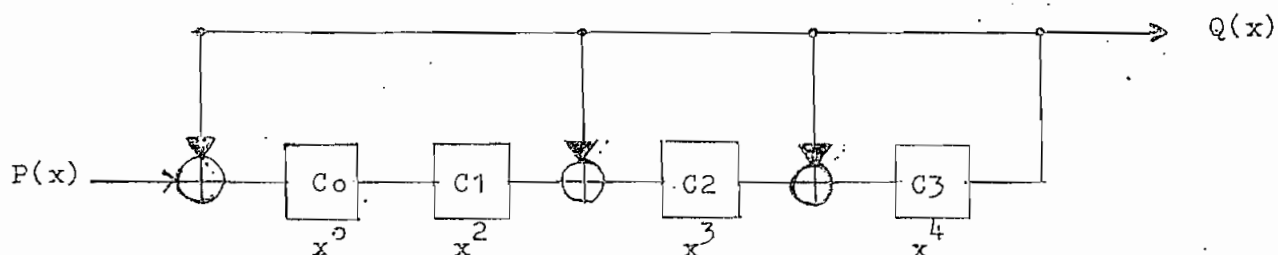


FIGURA 1-7: CIRCUITO QUE DIVIDE A $x^4 + x^3 + x^2 + 1$.

Nótese en la figura 1-7 que el coeficiente de x^1 es 0 y no se necesita un sumador.

La tabla 1-3 muestra la entrada, el estado y la salida del circuito cuando un polinomio

$$P(x) = x^6 + x^5 + x^4 \text{ se divide para } G(x) = x^4 + x^3 + x^2 + 1.$$

Para cada i se muestra el bit dividendo, el estado de los registros de desplazamiento, y el bit cociente antes y después de cada pulso de reloj.

TABLA 1-3

DIVISION DE $P(x) = x^6 + x^5 + x^4$ PARA $G(x) = x^4 + x^3 + x^2 + 1$

i	<u>ANTES DEL PUSO DE RELOJ</u>					<u>DESPUES DEL PULSO DE RELOJ</u>				
	Pi	C0	C1	C2	C3	qi	C0	C1	C2	C3
6	1	0	0	0	0	0	1	0	0	0
5	1	1	0	0	0	0	1	1	0	0
4	1	1	1	0	0	0	1	1	1	0
3	0	1	1	1	0	0	0	1	1	1
2	0	0	1	1	1	1	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0	0	1	0

(RESIDUO)

El residuo $R(x)$ es x^2 . Los bits del residuo (0 0 1 0) nos sirven como bits de chequeo. Para decodificación, como ya lo indicamos anteriormente, dividimos el polimONIO recibido $V(x)$ para $G(x)$; si el residuo no es cero se ha producido un error, y este ha sido detectado.

CAPITULO SEGUNDO

CIRCUITOS DE DETECCION Y CORRECCION DE ERRORES

2.1. CIRCUITO GENERADOR DE PARIDAD

En este capítulo supondremos que ya disponemos de un sistema de memoria acoplado a una computadora.


El sistema de memoria requerido para la detección y corrección de errores simples y detección de errores dobles, utilizando el código hamming; descrito en la sección 1-3; debe componerse de 8 ó 16 bits para los bits de datos y 5 ó 6 bits para los bits hamming de acuerdo a la longitud de la palabra del sistema que se está utilizando: palabra de 8 ó 16 bits.

El sistema a construirse, debe acoplarse a una microcomputadora de 8 bits de datos y 64 kbytes de direccionamiento (16 líneas de direcciones). Nosotros diseñaremos solo 16 kbytes de memoria y debemos usar 13 bits: 8 para los bits de datos y 5 para los bits hamming. Esto es: 16 kbytes X 13. En el capítulo 3 analizaremos el funcionamiento de este sistema de memoria.

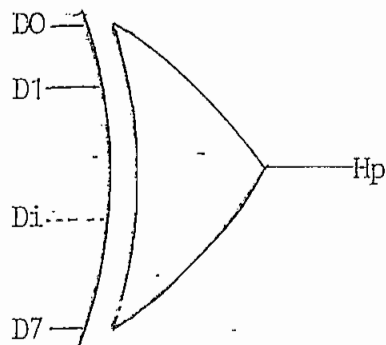
Antes de describir los circuitos de detección y corrección de errores, analicemos los circuitos generadores de paridad.

La tabla de verdad y la descripción de estos circuitos así como la de todos los circuitos usados en este trabajo son dadas en el apéndice A.

En la sección 1-3, habíamos visto que los bits hamming se los obtienen de la paridad par de alguna combinación de los bits del byte. Y que la paridad par no es sino la suma en módulo 2 de

esos bits, lográndose esta suma con la compuerta OR EXCLUSIVO cuyo símbolo es 

Por ejemplo obtengamos la paridad total (Hp) de una palabra de 8 bits : D0 - D7



$$H_p = D_0 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7$$

además recordando, el bit de paridad par es tal que el número de "1" en la combinación palabra-bit paridad es siempre par.

El circuito integrado 74LS280 realiza el mismo trabajo con características adicionales esto es: obtiene la paridad par o impar de 9 entradas diferentes.

En la siguiente sección veremos varios circuitos de detección y corrección de errores. Aceptando la nomenclatura:

$$R/\bar{W} = \begin{cases} 0L \Rightarrow \text{Escritura} \\ 1L \Rightarrow \text{Lectura} \end{cases} \quad \text{Nótese que: } R/\bar{W} \rightarrow \neg R/\bar{W}$$

2.2. CIRCUITOS DE DETECCION Y CORRECCION DE ERRORES

En esta sección describiremos varios circuitos de detección y co rrección de errores en palabras de 8 y 16 bits. Uno de los circuitos que indicaremos utiliza hardware y software para la detec ción y corrección de errores, los restantes utilizan únicamente hardware. El circuito construido en este trabajo corresponde al circuito D.

CIRCUITO A.-

El circuito A, se encuentra representado en la figura 2-1. Este circuito, detecta y corrige únicamente errores simples en palabras de 8 bits y solo utiliza hardware.

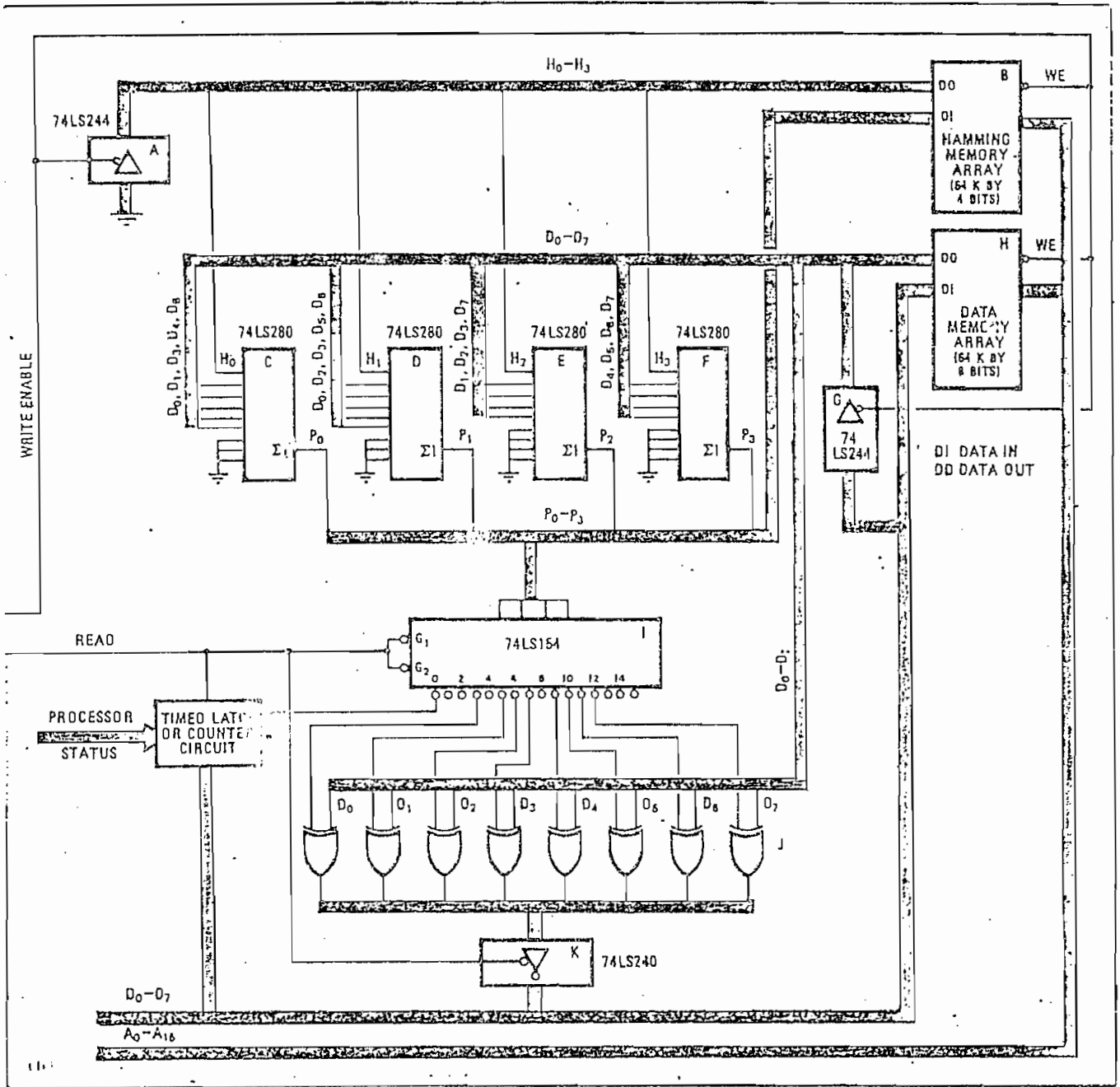


FIGURA 2-1

El sistema de memoria utilizado por este circuito, es un arreglo de 64 kbytes x 12, (4 bits hamming y 8 bits de datos).

Puesto que no se detectan errores dobles, no se utiliza el bit de paridad total y solo se requieren 4 bits hamming.

Nótese que el arreglo de memoria tiene líneas separadas de lectura y escritura, como es típico en memorias dinámicas, y que las líneas de salida D_0 tienen capacidad de 3 estados.

Supongamos que se va a realizar un ciclo de escritura: las líneas de direcciones seleccionan alguna localidad determinada de 8 bits de datos y la correspondiente localización de 4 bits de la memoria hamming.

Las líneas hamming H_0 - H_3 son puestas a tierra por el buffer A que es habilitado con la línea R/\bar{W} , que para escritura está en OL. La misma línea R/\bar{W} habilita el buffer G comunicando las líneas de datos, que a la vez son escritas en la memoria de datos, a los integrados generadores de paridad C, D, E, F. Esto permite obtener los bits de paridad de únicamente los datos (H_0 - H_3 son OL) para ser almacenados en la memoria hamming por medio de las líneas P_0 , P_1 , P_2 , y P_3 .

Esta condición es la que permite tener la condición de que el número de "1" en la combinación (bit paridad-datos) siempre sea par.

Nótese que las líneas P0-P3 no producen ninguna salida en el integrado I puesto que está deshabilitado con la línea (\bar{R}/W).

Nótese además que la memoria usada tiene para escritura la línea DO = Dout en alta impedancia.

Durante una operación de lectura: el buffer A se pone en circuito abierto y en las líneas H0-H3 se tienen los bits hamming escritos anteriormente en la memoria correspondiente. Estos, al igual que los bits de datos D0-D7 leídos desde la memoria correspondiente ingresan a los integrados de paridad para producir los bits de paridad P0-P3 que forman una palabra apuntadora (ver sección 1 - 3, tabla 1-1).

Para lectura, el integrado I está habilitado con la línea (\bar{R}/W). Las 4 líneas P0-P3, producen una de las 16 salidas del integrado I.

El número de salida, corresponde a la representación decimal del valor binario que se tiene en las líneas P0 a P3, y este valor a la vez corresponde a la posición ordinal del bit (ver tabla 1-1).

Si es que no se ha producido ningún error, las líneas P0-P3 serán todas 0 y la salida del integrado I será la 0, indicándonos que no ha existido error.

Nótese que las salidas del integrado I son invertidas, esta es: todas las salidas son "1" a excepción de aquella salida activa-

da que es "0" L.

Por lo tanto, debido a la tabla de verdad de las compuertas J, los datos D0-D7 leídos desde memoria son invertidos. El integrado K reinvierte los datos y estos llegan al procesador a través del bus de datos.

Si se produce algún error simple, la salida correspondiente a la posición ordinal del bit errado (tabla 1-1) es 0L, y la compuerta J correspondiente a ese bit no lo invierte. Sin embargo la compuerta K si lo hace, produciéndose la corrección de ese bit errado.

De esta forma, cualquier error simple detectado en un ciclo de lectura, es corregido antes de que la palabra leída pueda alcanzar el bus de datos. Sin embargo, a pesar de haber corregido el error, la memoria en sí, mantiene ese error. El contador indica do en el circuito puede ser chequeado periódicamente (cuando el procesador está libre de operación) por si hubo errores. Si ha ocurrido un error, es necesario leer cada palabra de memoria (en un ciclo de lectura automáticamente se produce la corrección) y reescribirla.

Debido a este frecuente acceso, se asume que la probabilidad de errores dobles, aunque finita, es poco probable.

CIRCUITO B.-

El circuito B se ilustra en la figura 2-2. Este circuito detecta y corrige errores simples y detecta errores dobles en palabras de 8 bits, el circuito utiliza hardware y software para la detección y corrección de errores.

Cuando un error se produce, el circuito genera una señal de interrupción para el procesador, y se ejecuta una rutina de software. Este programa o rutina lee los "latches" o retenedores de direcciones, datos y paridad, y esta información se utiliza para corregir errores simples y también para detectar errores dobles.

Este sistema requiere de un arreglo de memoria extra para almacenar el programa de corrección y se necesita que este arreglo sea mucho más confiable que el arreglo de memoria que está siendo chequeado, por que de otra manera, un error en el programa de corrección podría producir resultados incorrectos.

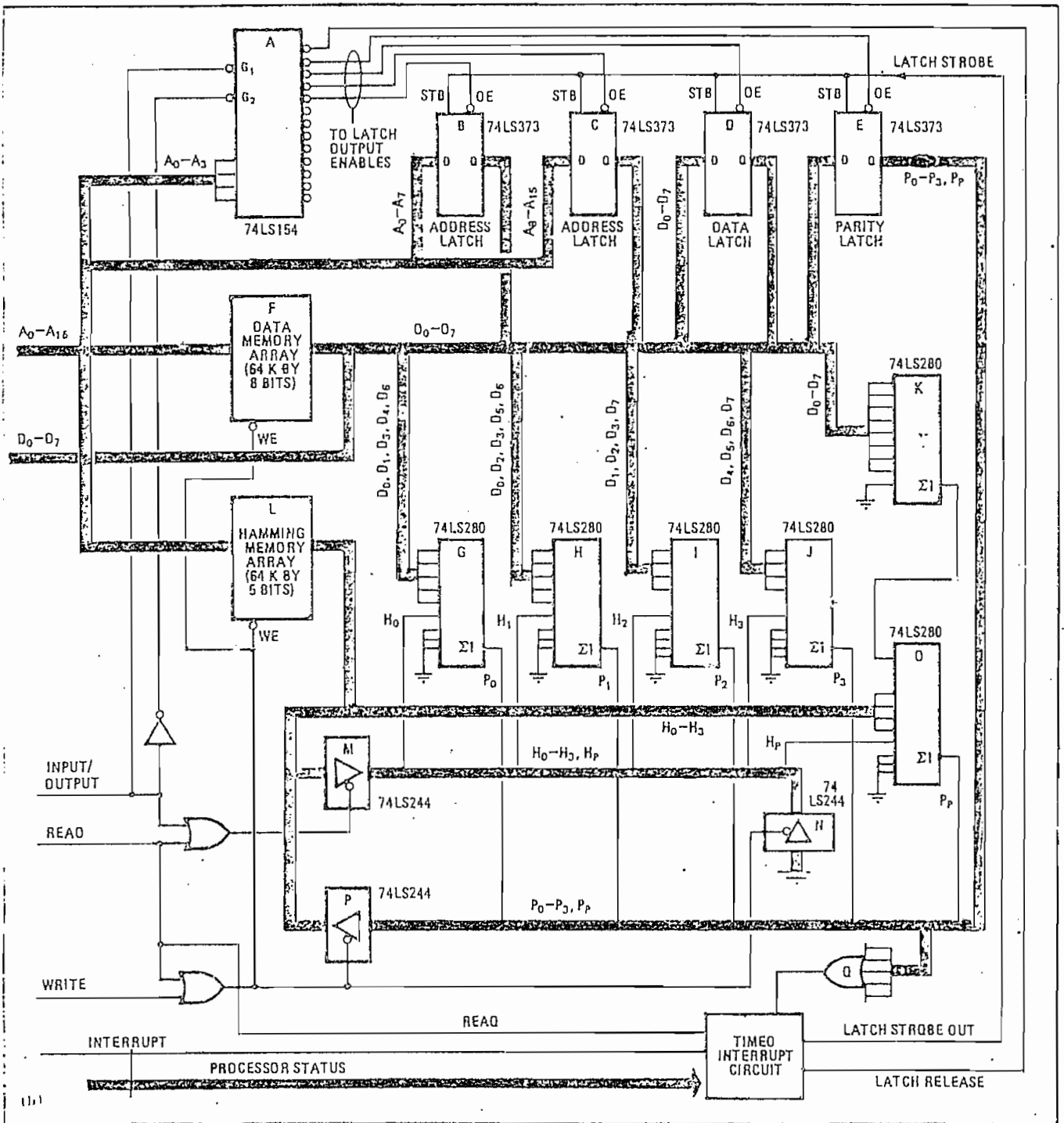


FIGURA 2-2

Para entender la secuencia de eventos, analicemos primero un ciclo de escritura:

El bus de direcciones selecciona una localidad de 8 bits de la memoria de datos y una localización correspondiente de 5 bits de la memoria hamming.

Mientras los 8 bits de datos son escritos en la memoria del CPU, la circuitería hamming genera una palabra de 4 bits hamming y un bit de paridad general que son almacenados en la memoria hamming.

Luego cuando los bits de datos son leídos desde la memoria, la circuitería hamming los suma con los bits hamming.

Si hay un error, una o más de las líneas de paridad PO-P3 se ponen en alto. Esto causa que la circuitería de interrupción, dé una señal (STROBE) para que se almacenen las palabras de direcciones, datos y paridad en sus respectivos "latches" y produce una interrupción al procesador que ejecuta un programa de interrupción el cual lee la información desde los "latches", la corrige y la reescribe en la memoria.

Más específicamente cuando una palabra de datos se escribe en memoria, las líneas HO-H3 y Hp (paridad total) son deshabilitadas de memoria por el buffer M de 3 estados y son puestas a tierra por el buffer N de 3 estados. Cada uno de los integrados de paridad G, H, I, J suman sus respectivos bits de datos con sus co-

respondientes bits hamming (h_i) que están puestos a tierra. Obteniéndose los bits de paridad P_0 - P_3 que son almacenados en la memoria hamming e ingresan al integrado 0 a través del buffer P que se encuentra habilitado.

Los integrados K y O obtienen la paridad total de D_0 - D_7 y H_0 - H_3 consiguiendo el bit P_p que también es almacenado en la memoria hamming.

Durante el ciclo de lectura de memoria, las líneas de paridad P_0 - P_3 y P_p son deshabilitadas de la memoria mientras que las líneas hamming H_0 - H_3 y H_p son deshabilitadas de tierra y habilitadas a memoria por el buffer M. Ahora los integrados de paridad G - J obtienen la paridad de sus respectivos datos y bits hamming según las ecuaciones P_0 - P_3 .

Los integrados K y O obtienen la paridad de D_0 - D_7 y H_0 - H_3 para obtener el bit de paridad total P_p .

Si no hay error, P_0 - P_3 y P_p son todos cero, pero si hay error, una o más de las líneas P_0 - P_3 estarán en 1L y se tendrá una salida de la compuerta Q.

En estas condiciones el circuito de interrupción envía una señal de interrupción al procesador y genera la señal "LATCH STROBE OUT", que habilita los 4 "latches" B-E que capturan respectivamente los bits de direcciones, datos y paridad.

El procesador enseguida ejecuta una subrutina que lee el contenido de los "latches". Cada uno de los "latches" son habilitados uno después de otro por el integrado A (decodificador de 4 a 16) el cual es direccionado como una puerta de I/O.

El programa, con la información recibida desde los "latches" corrige, los datos en base al código hamming y la palabra apuntada y los reescribe en memoria o detecta un error doble y da la indicación de que ello ha ocurrido. Finalmente la señal "LATCH RELEASE", desde el integrado A, resetea la condición de interrupción.

CIRCUITO C.-

El circuito C se ilustra en la figura 2-3. Este circuito detecta y corrige únicamente errores simples en palabras de 16 bits, utiliza solo hardware y su estructura es similar al circuito A con la diferencia de que este trabaja para palabras de 16 bits.

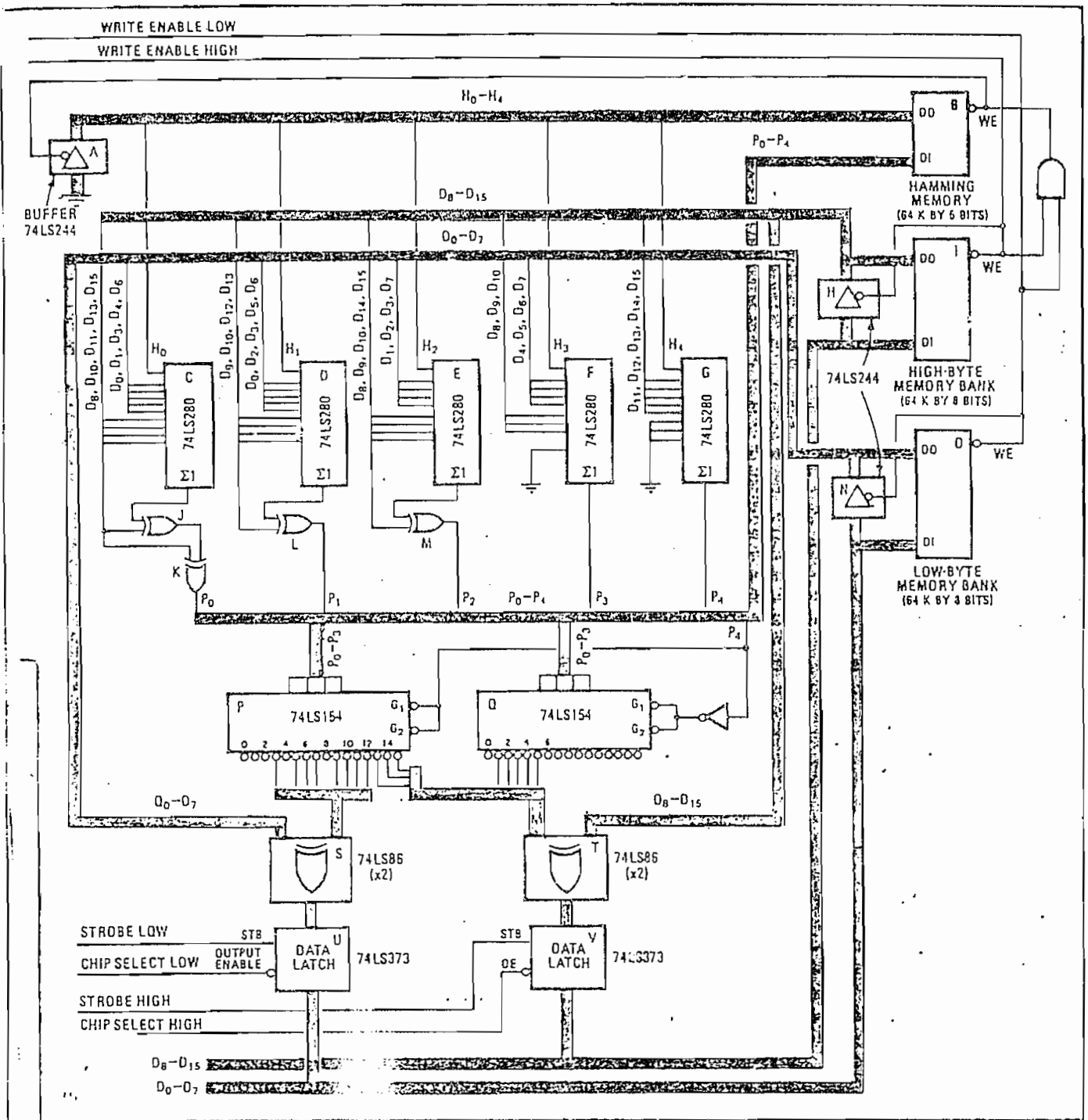


FIGURA 2-3

El sistema de detección y corrección de errores en palabras de 16 bits es más complicado que el similar para palabras de 8 bits, porque en primer lugar, se necesita resolver cinco ecuaciones hamming en lugar de cuatro; segundo, las operaciones del procesador son mucho más complejas debido a que este puede acceder separadamente a los 8 bits de orden bajo o alto, o puede acceder en forma global a los 16 bits de la palabra.

En el caso de una operación de escritura a un solo byte (8 bits), los datos en el otro byte deben ser primero leídos, corregidos, y luego combinados con los datos a ser escritos para generar los bits hamming.

Por ejemplo, supongamos que se desea escribir un byte en el banco de memoria alto (bloque I en la figura 2-3) pero no en el otro.

Primero, la memoria es accesada, pero algún circuito deshabilita la señal de escritura, causando que los datos sean leídos y, a través de los circuitos hamming, sean almacenados en los "latches" U y V. La salida del "latch" U se habilita poniéndose el byte de orden bajo en las líneas de datos D0-D7. En este proceso, este byte, en caso de que hubiese tenido algún error, ya fue corregido.

Sin embargo, la salida del "latch" V no se habilita (en este byte D8-D15, queremos escribir nueva información).

En este momento, la señal de escritura ya es habilitada y de esta forma el byte de orden alto D8-D15, desde el procesador, se escribe en la memoria de orden alto, mientras que el byte de orden bajo, desde el "latch" (ya corregido) se escribe en la memoria de orden bajo. Al mismo tiempo, las líneas H0-H4 son puestas a tierra por el buffer A de 3 estados y los bytes de orden alto y bajo son admitidos en los integrados de paridad C-G a través de los buffers H y N. Las salidas de paridad PO-P4 son entonces escritas en la memoria hamming.

Cuando se ejecuta un ciclo de lectura los bits hamming, leídos desde su memoria, y los bits de datos generan los bits de paridad PO-P4. Si se produjo algún error, una o más de estas líneas se pondrán en alto dándonos una palabra apuntadora según la tabla 1-2 ilustrada en la sección 1-3.

Las salidas de los decodificadores P y Q de 4 a 16, se combinan con los datos D0-D7 y D8-D15 en las compuertas OR-EXCLUSIVO para corregir cualquier error simple para finalmente poner en las líneas D0-D7 y D8-D15 los datos ya corregidos.

Debido a que los decodificadores 74LS154 solo tiene 16 bits de salida, se requieren dos de ellos. P4, cuando es 0, selecciona las salidas de orden bajo del integrado P y, cuando es 1, selecciona las salidas de orden alto del integrado Q y P.

Nótese que si el integrado P no está habilitado, todas sus sali-

das son 1L (desactivadas) y parte de sus salidas, sirven para combinarse con los datos de orden alto D8-D15.

Nótese además que los números de las salidas de los integrados P y Q corresponden a los números de posición de los bits de datos de la tabla 1-2 dada en la sección 1-3.

CIRCUITO D.-

El circuito D se ilustra en la figura 2-4. El circuito detecta y corrige errores simples y también detecta errores dobles en palabras de 8 bits.

En el presente trabajo, este circuito será construido y se probará el funcionamiento del código hamming. Los detalles de la construcción y las pruebas de funcionamiento serán explicadas en el Capítulo IV, mientras que el diseño y funcionamiento del sistema de memoria será explicado en el capítulo III.

El sistema de memoria utilizado por este circuito es un arreglo de 16 K x 13. Puesto que estamos detectando errores dobles se necesita el bit de paridad total P_p , requiriéndose en total 5 bits hamming; los restantes 8 bits son los de datos.

La memoria utilizada es del tipo dinámico, tiene las líneas de escritura y lectura separadas, teniendo las últimas la capacidad de 3 estados.

Cuando se produzca algún error simple (un solo bit errado) en un ciclo de lectura, el sistema lo detectará y lo corregirá automáticamente.

Cuando se produzca algún error doble (dos bits errados en la misma palabra) el sistema solo lo detectará y dará una indicación de que ello ha ocurrido, encendiendo el LED ilustrado en el diagrama del circuito.

Si se producen más de dos errores en la misma palabra, el sistema trabajará mal, produciendo resultados incorrectos. Pero, como ya se indicó anteriormente, la probabilidad de tener más de dos errores, en la misma palabra es muy pequeña.

Para entender el funcionamiento del circuito, veremos la secuencia de eventos en un ciclo de escritura y en uno de lectura de alguna localidad determinada de memoria.

Supongamos que se va a realizar un ciclo de escritura de datos

en la memoria del circuito. Las 14 líneas de direcciones A0-A13 seleccionan alguna localidad determinada de 8 bits de la memoria de datos y la correspondiente localización de 5 bits de la memoria hamming.

La línea R/\bar{W} habilita las salidas RO de los dos integrados 8T26 permitiendo que los datos DO-D7 sean escritos en la memoria de datos.

Las líneas hamming HO-H3 y Hp son puestas a tierra por el buffer L que se encuentra habilitado por la línea R/\bar{W} , la misma línea habilita el buffer B que comunica los datos DO-D7 con los integrados generadores de paridad: D, E, F, G, y C.

Nótese que las líneas DO de las memorias A y H se encuentran en alta impedancia (ciclo de escritura).

Los integrados D, E, F, y G generan los bits de paridad PO-P3, de acuerdo a las ecuaciones hamming, que son escritas en la memoria hamming a través del buffer M y alimentados al integrado generador de paridad J que obtiene, conjuntamente con la salida del integrado C, el bit paridad total P_p que también es escrito en la memoria hamming.

Nótese que las líneas PO-P3 no producen ninguna salida en el integrado N puesto que esta deshabilitado con la línea \bar{R}/W .

Durante una operación de lectura de memoria, el buffer L se pone

en circuito abierto y el buffer K comunica los bits hamming H0-H3 y Hp leídos desde memoria con las entradas de los integrados D, E, F, G y J respectivamente.

Al mismo tiempo los bits de datos D0-D7 leídos desde su memoria ingresan a los integrados D, E, F, G y C generando los 4 primeros, los bits de paridad P0-P3, mientras que los integrados C y J generan el bit de paridad total Pp.

Para lectura, el integrado N, decodificador de 4 a 16 74LS154, está habilitado con la línea \bar{R}/W y sus entradas P0-P3 forman una palabra apuntadora (Tabla 1-1 Sección 1-3).

Nótese que las salidas usadas del decodificador N corresponden a los números de posición de los bits de datos de la tabla 1-1 y recuérdese que estos números corresponden a la representación decimal de la palabra apuntadora P0-P3.

Si no se ha producido ningún error las líneas P0-P3 serán todas cero, y la salida activada del integrado N será la cero, indicándonos que no ha existido error.

Nótese que las salidas del integrado N son invertidas, esto es todas las salidas son 1L a excepción de aquella salida activada que es 0L. Por lo tanto, debido a la tabla de verdad de las compuertas Q, todos los datos D0-D7, leídos desde memoria, son invertidos. El integrado R reinvierte los datos y estos llegan al procesador a través de los integrados 8T26 con sus entradas Di activadas.

Si se produce algún error simple (un solo bit errado), la palabra apuntadora P0-P3 apuntará a ese bit y se activará (OL) la salida del integrado N correspondiente a la posición ordinal del bit errado (tabla 1-1).

De acuerdo a la tabla de verdad de las compuertas Q todos los bits de datos serán invertidos a excepción de aquel bit errado. El integrado R reinvierte los datos, produciéndose la corrección del error, y estos llegan al procesador a través de los integrados 8T26.

Nótese que el LED indicador de error doble siempre permanecerá apagado a menos que, en el punto del circuito indicado con $\overline{\text{NMI}}$ se tenga un OL, y esta condición debe cumplirse únicamente en un ciclo de lectura ($\overline{\text{R}}/\text{W}=0$) cuando se detecte un error doble.

Si se produce algún error doble, P_p será OL y una o más de las líneas P0-P3 será LL, produciendo de esta manera, a través de las compuertas O y P, la condición de OL en el punto indicado con $\overline{\text{NMI}}$, encendiendo el led indicador de error doble.

CAPITULO TERCERO

DISEÑO DE UN SISTEMA DE 16 KBYTES DE MEMORIA
CON DETECCION Y CORRECCION DE ERRORES

3.1 FUNCIONAMIENTO GENERAL DEL CIRCUITO

Puesto que en la sección 2-2 analizamos el funcionamiento del circuito de detección y corrección de errores, en esta, nos referimos más concretamente al sistema de memoria.

La figura 3-1 muestra el diagrama de bloques del circuito completo. Para la descripción del funcionamiento del sistema se hará referencia a esta figura conjuntamente con la figura 3-2 la cual muestra el diagrama de tiempos general del circuito.

Como se había mencionado anteriormente, necesitamos de un sistema de memoria RAM de 16 k x 13. Utilizaremos para ello 13 integrados UPD 416 o cualquiera de sus equivalentes: MCM 4116, MM5290, etc. Las especificaciones de este y otros integrados que utilizaremos en el diseño se da en mayor detalle en el apéndice A.

El integrado UPD 416-2 es una memoria dinámica de acceso randómico de alta velocidad (200ns para nuestro diseño) organizada como 16 k x 1 bits esto es 16384 bits.

Para direccionar los 16 kbytes de memoria necesitamos de 14 líneas de direcciones ($2^{14} = 16k$). Sin embargo se debe notar que los integrados de memoria poseen solamente 7 líneas de direcciones razón por la cual su empaquetadura contiene unicamente 16 patas lo cual es una ventaja por razones de tamaño del integrado.

Como es obvio, al tener los integrados de memoria sólo 7 líneas de direcciones, se necesita hacer un multiplexado de las 14 líneas y esto lo hace el Multiplexer MC 3242 con la línea de control ROW ENABLE, la cual si es 1 lógico permitirá el paso de A0 - A6 denominadas filas de direcciones y si es 0 lógico pasarán A7 - A13 denominadas columnas de direcciones.

El integrado de memoria está organizado como 128 filas y 128 columnas dando un total de $128 \times 128 = 16384$ células de memoria. Siendo la memoria del tipo dinámico se necesita realizar un ciclo de refresco de memoria máximo cada 2 milisegundos (Particular para el tipo de memoria usada), esto es $1/64$ de milisegundo para cada fila; requiriéndose para ello de un oscilador de por lo menos 64 KHZ.

Para el diseño del sistema se dispone de las líneas de: datos D0-D7, direcciones A0-A7, control R/W, reloj MEMORY CLOCK y habilitación CE provenientes del microcomputador MEK 6800 D2 al cual vamos a acoplar el sistema de memoria.

Las 14 líneas de direcciones A0-A13, necesarias para direccionar 16 kbytes, ingresan al circuito MC-3242 (ver figura 3-1) donde son multiplexadas, bajo control de la línea ROW ENAELE, a 7 líneas: primero las filas de direcciones A0-A6 y luego las columnas de direcciones A7-A13 ingresando estas 7 líneas a los integrados de memoria.

Las líneas de direcciones A14 y A15 ingresan al controlador de memoria MC 3480 y de las combinaciones de los estados lógicos de estas se selecciona una línea $\overline{\text{RAS}}$ de acuerdo a la siguiente tabla de verdad:

A15	A14	$\overline{\text{RAS}}$ Seleccionada	Rango de Memoria Seleccionado (HEX)
0	0	$\overline{\text{RAS}}$ 1	0000 - 3FFF
0	1	$\overline{\text{RAS}}$ 2	4000 - 7FFF
1	0	$\overline{\text{RAS}}$ 3	8000 - BFFF
1	1	$\overline{\text{RAS}}$ 4	C000 - FFFF

Cada una de estas líneas $\overline{\text{RAS}}$ podría ir conectada a un banco de 16 kbytes de memoria para tener un total de 64k, pero sólo vamos a diseñar 16k, conectando únicamente la línea $\overline{\text{RAS}}$ 1 y dejamos las otras en circuito abierto. De esta manera trabajaremos en el rango de memoria 0000-3FFF (HEX).

Tenemos además en las entradas del 3480, 5 señales de tiempo t1-t5 las cuales dan la sincronización que debe existir en el funcionamiento de todo el sistema. Esta sincronización consta de un control; para hacer el multiplexado de 14 a 7 líneas de direcciones en el 3242, de sincronización para realizar el ciclo de lectura/escritura, de sincronización para los ciclos de refresco, etc.

Para entender mejor el funcionamiento del circuito analicemos un ciclo de MPU o de microprocesador. La Figura 3-2 muestra los diagramas de tiempos.

Para este diseño en especial en el que se ha usado el método de refresco transparente (posteriormente se darán los detalles de éste método y de otro alternativo), se nota que en un ciclo de MPU se debe realizar un ciclo de refresco y uno de R/W.

El ciclo de refresco se realiza cuando la señal de reloj ϕ_2 del microprocesador es 0 lógico y el ciclo de R/W se realiza cuando ϕ_2 es 1 lógico y esto tiene que ser así porque el microprocesador lee o escribe de o a memoria únicamente cuando ϕ_2 es 1 lógico. Por la misma razón el método de refresco se llama "transparente" porque se realiza cuando ϕ_2 es 0 lógico y mientras ϕ_2 es 0 lógico el microprocesador no se relaciona con el sistema de memoria.

Para poder utilizar este método se necesita que la memoria usada sea bastante rápida y/o que el reloj del microprocesador no sea demasiado rápido. En nuestro diseño cumplimos con los dos requisitos pues los integrados de memoria usados (UPD 416 - 2) tienen un tiempo de acceso de 200 nano segundos y el reloj del microprocesador corresponde al circuito MC 6871 B cuya frecuencia de operación es de 614.4 khz., (esto es 1627.6 ns de período).

Se debe notar además que los ciclos de refresco se realizan con una frecuencia de 64 KHZ esto es 15625 ns o más grande que es un valor

aproximadamente 10 veces menor que la frecuencia del reloj por lo tanto no en todos los ciclos de MPU habrá ciclos de refresco. Respecto a esta condición, existe la posibilidad de eliminar el oscilador de 64 KHZ conectando la línea \bar{L} (señal invertida con respecto a $\emptyset 2$) a las entradas Ref Clk y $\overline{\text{count}}$ de los controladores de memoria (ver apéndice B para conectar el circuito bajo esta condición). Produciéndose con esta condición un ciclo de refresco cada ciclo de MPU. La desventaja que tiene esta posibilidad es el aumento del consumo de potencia por los continuos ciclos de refresco.

La figura 3-2 muestra la mayoría de las señales de tiempo que intervienen en el sistema y se ha incluido las correspondientes demoras de tiempo.

Las características eléctricas y de tiempos de las señales $\emptyset 1$, $\emptyset 2$, MEMORY CLOCK; provenientes del reloj del microprocesador MC6871B se encuentran más ampliamente detalladas en el apéndice A.

3.2. CICLOS DE REFRESCO Y DE LECTURA/ESCRITURA.

En la figura 3-2 se puede ver la secuencia de generación de las señales de tiempo para los ciclos de refresco y de lectura/escritura.

CICLO DE REFRESCO

Cuando se produce una transición positiva en las entradas REF. CLK (REFRESH CLOCK) y $\overline{\text{COUNT}}$ de los circuitos 3480 y 3242 respectivamente se inicia un ciclo de refresco de la siguiente manera: La salida $\overline{\text{REF REQ}}$ del 3480 se pone en 0 lógico, situación que indica Requerimiento de Refresco, este valor ingresa a la pata de datos del flip flop tipo D y cuando se produce la transición positiva de la señal \bar{L} en su entrada

de reloj CK1 el 0 lógico de la entrada pasa a la salida generando las señales de tiempo t_1 , t_2 , t_3 y t_4 con sus tiempos programados correspondientes, los cuales son generados por los circuitos monoestables 74121. Sus demoras de tiempo se calculan con la expresión $t=0.7RC$., obteniéndose para $t_1=90ns$ los valores de resistencia y condensador:

$R_1=2.57K\Omega$, $C_1=50PF$. Para $t_2=142ns$: $R_2=3.12K\Omega$, $C_2=65 PF$.

Para $t_3=185 ns$ = $R_3=3.36K\Omega$, $C_3=79PF$.

(Nótese que al producirse una transición positiva de \bar{L} y la señal $\overline{Ref Req}$ es uno lógico, no se generarán las señales de tiempo $t_1 - t_4$ situación que se puede observar en el siguiente ciclo de MPU mostrado en la Figura 3-2. También se debe notar que cuando se produce una transición positiva de la señal L aplicada a la entrada de reloj CK2 del flip flop tipo D el 0 lógico permanente de su entrada pasa a la salida generando siempre las señales $t_1 - t_4$ para los ciclos de lectura/escritura).

A la señal \bar{L} se le da una demora para producir la señal MC que ingresa a la pata MC del 3480 y la señal de salida \overline{MC} se aplica a la entrada REF GRANT (concesión de refresco) del mismo circuito.

Cuando la señal REF GRANT se pone en 1L hace que las salidas del 3480: REF EN (habilitación de refresco) se ponga en 1 lógico y ROW EN en 0 lógico.

El estado alto de la señal REF EN es aplicado a la entrada REF EN del 3242. Esta condición hace que sin importar el valor de su entrada ROW

EN, se tenga en sus salidas el valor de un registro interno de 7 bits, este registro de 7 bits interno en el 3242 es incrementado en 1 con cada pulso del oscilador de 64 KHZ aplicado en la entrada $\overline{\text{COUNT}}$ y su función es la de indicar al sistema cual de las 128 filas de los integrados de memoria debe ser refrescada, así después de 128 ciclos de refresco se habrán refrescado los 16 k x 1 células de memoria de todos los integrados.

Para los ciclos de refresco, unicamente nos interesan las señales t1 y t5. Las otras señales t2, t3, t4, a pesar de generarse no tienen ningún efecto en el sistema.

La transición positiva de la señal t1 hace que todas las líneas $\overline{\text{RAS}}$ (ROW ADDRESS SELECT) se pongan en 0 lógico habilitando de esta manera todos los bancos de memoria (pues toda la memoria debe ser refrescada) recordando que en nuestro diseño unicamente poseemos un banco de memoria de 16 kbytes.

La misma transición positiva de t1 obliga a que la línea $\overline{\text{REF REQ}}$ regrese al estado alto.

De esta manera se realiza el ciclo de refresco y al final de este, la transición positiva de t5, hace que el controlador 3480 ponga sus líneas: $\overline{\text{RAS}}$ y ROW EN en 1 lógico y REF EN 0 lógico inicializando el sistema para el siguiente ciclo correspondiente a uno de lectura/escritura.

CICLO DE LECTURA/ESCRITURA

Un ciclo de LECTURA/ESCRITURA se da inicio con la transición positiva de la señal del reloj L proveniente del microcomputador.

Como ya se explicó anteriormente, la transición positiva de esta señal conectada a la entrada de reloj del flip flop tipo D (CK1) permite que el nivel permanente de 0 lógico en su entrada pase a la salida generando las señales de tiempo t1 - t4.

La generación de las señales t1, t2, t3 debe ser en forma secuencial, pudiendo conectarse la señal t4 a t2 o a t3.

La condición inicial de las señales de tiempo para un ciclo de R/W es la siguiente: señal \overline{RAS} en 1 lógico, ROW EN en 1 lógico, \overline{CAS} en 1 lógico, REF EN en OL. Esta condición se logra con la transición positiva de t5 la cual también es generada gracias a la transición positiva de L.

Al estar las señales REF EN en 0 lógico y ROW EN en 1 lógico, a las salidas del multiplexer MC3242, o, lo que es lo mismo, en las entradas de direcciones de los integrados de memoria, tendremos las filas de direcciones A0-A6.

Hay que tener en cuenta que a pesar de tener las filas de direcciones en las entradas de los integrados de memoria, el circuito interno de estas, aún no reconoce el valor de sus entradas y solo lo hace cuando gracias a la transición positiva de t1 la línea \overline{RAS} 1 (ROW ADDRESS SELECT) se pone en OL.

Al ponerse \overline{RAS} 1 en OL, los integrados de memoria reconocen el contenido

de sus entradas de direcciones que hasta este momento son las filas de direcciones A0-A6, seleccionándose internamente una de las 128 filas de cada integrado de memoria.

Luego gracias a la transición positiva de t2 la línea ROW EN se pone en OL lo que obliga a que en las salidas del 3242 se tengan las columnas de direcciones A7-A13, pero igual que antes, los integrados de memoria no reconocen el valor de sus entradas de direcciones y esto no sucede hasta que gracias a la transición positiva de t3 la línea $\overline{\text{CAS}}$ (COLUMN ADDRESS SELECT) en la salida del 3480 se pone en OL seleccionándose internamente en los integrados de memoria una de las 128 columnas de la fila seleccionada en todos los integrados de memoria.

En este momento se tiene perfectamente especificada una de las 16384 localidades de memoria en la que se desea leer o escribir.

La transición positiva de t4 es el control para que la condición de la línea R/ $\overline{\text{W}}$ in en la entrada del 3480 se ponga en su salida R/ $\overline{\text{W}}$ out y ésta a la vez indica a los integrados de memoria si se trata de un ciclo de lectura (R) o uno de escritura ($\overline{\text{W}}$).

Se debe notar que de acuerdo a las especificaciones de los integrados de memoria UPD 416-2 mostradas en el apéndice A existe un tiempo mínimo tRP (ROW ADDRESS STROBE PRECHARGE TIME) durante el cual la señal $\overline{\text{RAS}}$ debe permanecer en 1 lógico. Este tiempo mínimo es de 120 ns. En la figura 3-2 se puede ver que tRP para el presente diseño es de 149 ns, valor que cumple con la especificación.

En el momento en el que la señal $\overline{\text{CAS}}$ baja a 0 lógico se tiene, como ya se indicó, completamente direccionada una localidad de memoria para escribir o leer de ella teniendo en cuenta el tiempo de acceso especificado (200 ns). (ver apéndice A).

El tiempo de acceso puede ser tomado desde la transición negativa de la señal $\overline{\text{CAS}}$ ($t_{\text{acc}} = 135\text{ns}$) o desde la transición negativa de la señal $\overline{\text{RAS}}$ ($t_{\text{acc}} = 200\text{ ns}$) todo depende de la diferencia de tiempo entre las transiciones negativas de las señales $\overline{\text{CAS}}$ y $\overline{\text{RAS}}$ denominado $t_{\text{RCD}} = 76\text{ns}$ para el presente diseño.

Las especificaciones dan un tiempo máximo para t_{RCD} y tiene un valor de $t_{\text{RCD max}} = 65\text{ ns}$ y se asume que si $t_{\text{RCD}} < t_{\text{RCD max}}$ el tiempo de acceso es igual a $t_{\text{RAC}} = 200\text{ns}$ y si $t_{\text{RCD}} > t_{\text{RCD max}}$ (nuestro caso) el tiempo de acceso es igual a $t_{\text{CAC}} = 135\text{ns}$.

Para la siguiente descripción del funcionamiento del sistema, vamos a asumir que se realiza un ciclo de lectura por cuanto este ciclo representa mayor complejidad con respecto a un ciclo de escritura.

Luego del tiempo de acceso los datos son válidos en las salidas de los integrados de memoria y para llegar a las entradas de datos del microprocesador, deben pasar por el circuito de detección y corrección de errores y por los "Buffers" 8T26, lo cual toma alrededor de 200ns de demora. La demora de estos circuitos para un ciclo de escritura es de alrededor de 176ns.

Tal como se puede ver en la figura 3-2 los datos se encuentran válidos 265ns antes de la transición negativa de $\emptyset 2$. El tiempo mínimo en el

que los datos deben estar válidos antes de la transición negativa de ϕ_2 es de 100ns, por lo tanto se cumple con la especificación y los datos pueden ser perfectamente leídos por el microprocesador.

Los datos dejan de ser válidos en las salidas de los integrados de memoria cuando la señal $\overline{\text{CAS}}$ sube a 1 lógico y esto lo hace después de la transición negativa de ϕ_2 .

La transición positiva de t_5 al final del ciclo de R/\overline{W} hace que el sistema se inicialice esto es: pone la línea $\overline{\text{RAS}}$ en 1 lógico, ROW EN en 1L, $\overline{\text{CAS}}$ en 1 lógico y R/\overline{W} en 1 lógico, en espera del siguiente ciclo de MPU.

METODO DE REFRESCO DE ROBO

El método de refresco de robo, a diferencia del método de refresco transparente, utiliza todo un ciclo de MPU para su realización. Esto lo logra enviando una instrucción de suspensión de actividades al microprocesador.

Este método resulta poco conveniente cuando la velocidad de operación del sistema es indispensable.

En el presente diseño se ha utilizado el método de refresco transparente el cual como ya se indicó pasa desapercibido para las operaciones del microprocesador pues en un ciclo de MPU se realiza un ciclo de Refresco y uno de lectura/escritura sin suspender las actividades del microprocesador.

Por otra parte el método de refresco de robo resulta indispensable cuando la frecuencia de operación del microprocesador es muy alta pues en ese caso no se alcanzaría a ejecutar un ciclo de refresco y uno lectura/

escritura en el mismo ciclo de MPU.

En algunas aplicaciones de microprocesadores en las que se necesita llevar un conteo del número de ciclos de MPU para gobernar alguna señal externa se utiliza el método de refresco transparente pues el método de refresco de robo produciría resultados incorrectos.

3.3. CIRCUITOS DE INDICACION DE ERROR SIMPLE Y DOBLE

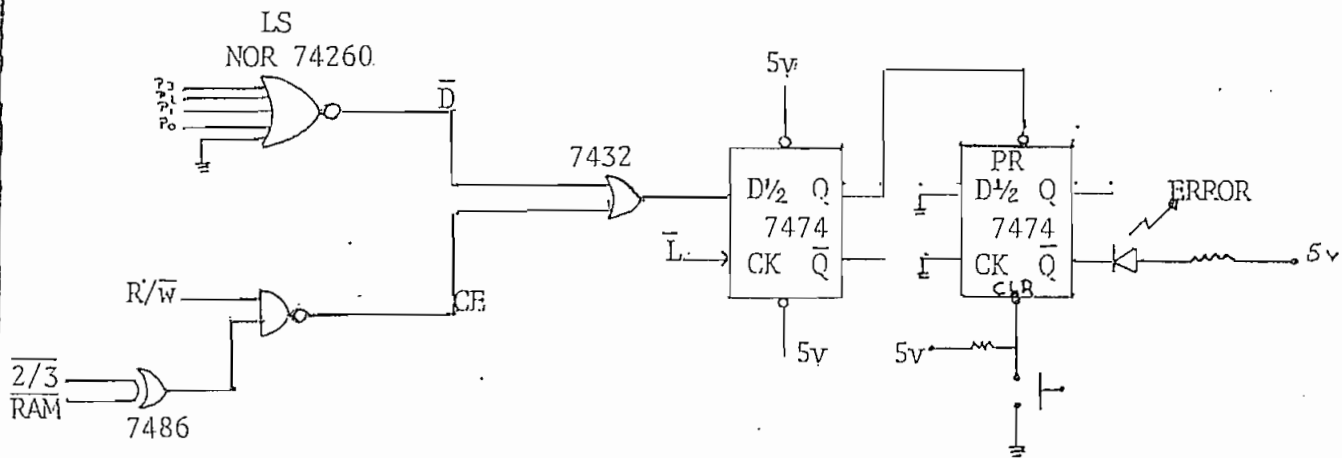
En la sección 2-2 - Circuito D se explica detalladamente el funcionamiento del sistema para detectar y corregir errores simples y detectar errores dobles.

En esta sección únicamente recordaremos las condiciones de: sin error, Error Simple y Error Doble dados por los estados lógicos de las líneas P₀ - P₃ - y P_p en las salidas de los integrados 74S280 (figura 3-1) en un ciclo de lectura de memoria:

Si no existe error, las líneas P₀ - P₃ y P_p son 0 lógico. Si existe error simple, la línea P_p es 1 lógico y por lo menos una de las líneas P₀ - P₃ es diferente de cero. La combinación de estas líneas forman una palabra apuntadora que indica al sistema cual es el bit errado.

Si existe error doble P_p es 0 lógico y por lo menos una de las líneas P₀ - P₃ será 1 lógico.

El circuito que indica la condición de error es el siguiente:



El led conectado a la salida \overline{Q} del 7474 debe encenderse unicamente bajo las siguientes condiciones:

- 1) Cuando el sistema se encuentra realizando un ciclo de lectura esto es cuando la señal R/\overline{W} es 1 lógico.
- 2) Cuando el sistema está trabajando en el rango de memoria comprendido entre 0000 y 3FFF (rango de memoria que posee detección y corrección de errores) lo cual pone la línea CE en 1 lógico.

Esta condición se logra con el OR EX de las señales $\overline{2/3}$ y \overline{RAM} provenientes del microcomputador.

La señal \overline{RAM} es la habilitación para los 8 kbytes primeros de memoria esto es de 0000 a 1FFF y la señal $\overline{2/3}$ es la habilitación para el rango de 2000 a 3FFF. Los diagramas y más información sobre el microcomputador se encuentran ilustrados en el apéndice C.

- 3) Cuando al cumplirse las dos primeras condiciones por lo menos una de las líneas P0 - P3 es 1 lógico en el momento en el que se produce una transición positiva de la señal \overline{L} . (instante en el que las salidas de memoria deben ser válidas y el microprocesador esta leyendo los datos).

Si se cumplen las tres condiciones, en la entrada de datos del primer flip flop tipo D 7474 tendremos un 0 lógico

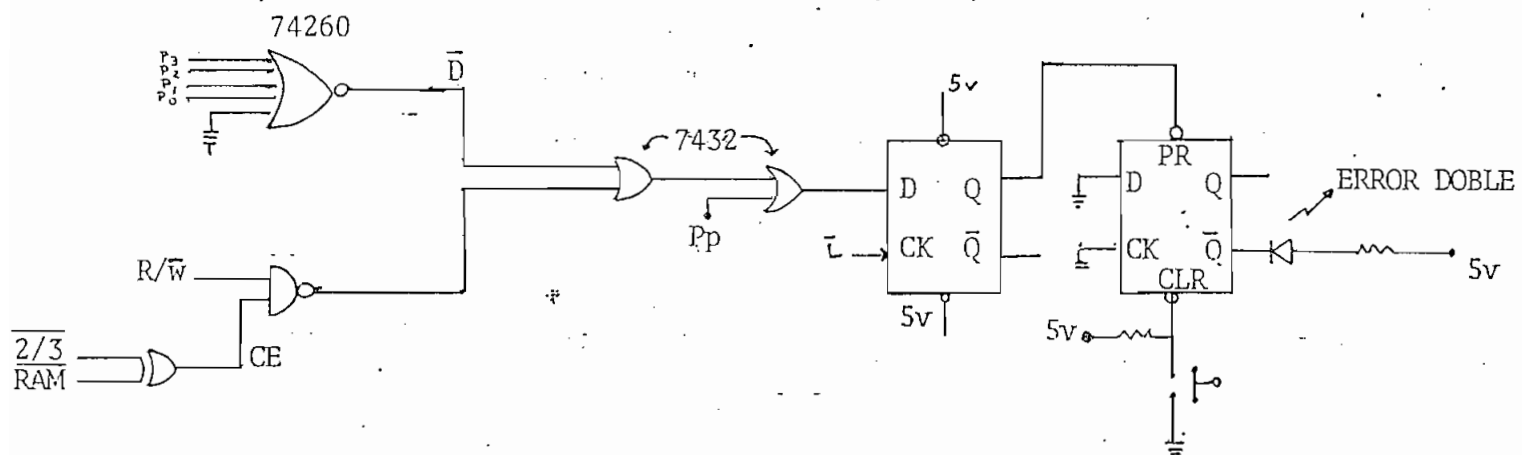
y con la transición positiva de \bar{L} pasa a la salida y entra al Preset del segundo flip flop lo que obliga a que la salida \bar{Q} se ponga en 0L y se mantenga en ese estado hasta que el interruptor en la entrada de CLEAR se activa poniendo la salida \bar{Q} nuevamente en 1 lógico apagando el LED.

El interruptor utilizado es el mismo que corresponde al interruptor de $\overline{\text{RESET}}$ del microcomputador.

Si cualquiera de las dos primeras condiciones no se cumple, en la entrada de datos del flip flop se tendrá un 1 lógico y no se prenderá el led.

Nótese que el conjunto de los 2 flip flop forman un circuito retenedor.

El circuito indicador de error doble es muy parecido al anterior con la siguiente diferencia:



Se ha añadido una compuerta OR con una de sus entradas conectada a la línea P_p (Figura 3-1).

El led indicador de error doble se enciende cuando a más de cumplirse las dos primeras condiciones para la indicación de error simple: se cumple que p_p sea OL y por lo menos una de las líneas $P_0 - P_3$ sea diferente de 0 al producirse la transición positiva de \bar{L} .

Si se tiene un error simple P_p será 1 lógico y el led de error doble no se encenderá.

Pero si se tiene un error doble tanto el led de error simple como el de doble se encenderán.

El diagrama del circuito completo y detallado se encuentra ilustrado en la Figura 3-3.

CAPITULO QUARTO

CONSTRUCCION

4.1.1 CONSIDERACIONES SOBRE LAS FUENTES DE ALIMENTACION

La potencia dinámica requerida por los sistemas de memoria es una función de la frecuencia de operación. Por lo tanto las fuentes de voltage utilizadas deben estar completamente dentro de las especificaciones, debiendo existir además adecuados desacoplamientos para suprimir el ruido de alta frecuencia resultado de las transiciones de corriente de los dispositivos. Esto asegura una óptima ejecución y confiabilidad del sistema.

Para lograr estos desacoplamientos se incluye en la tarjeta un condensador de 0.1 μF , 50V por cada integrado de memoria y por cada fuente (VCC(5V), VDD (12V), VBB (-5V)) conectando estos condensadores lo más cercanamente posible a las patas de fuente y tierra de cada integrado. Estas conexiones deben ser bastante anchas para reducir la inductancia.

Todos los caminos de fuente y tierra hacia los circuitos de memoria RAM deben correr horizontal y verticalmente hacia el arreglo. De esta manera se reduce la inductancia efectiva del sistema. En el presente diseño, esta situación se logra gracias a la configuración de la tarjeta 9610 utilizada.

Se deben incluir condensadores de desacoplamiento de volumen por cada fuente. En el presente diseño se utiliza uno de 100 μF , 15V por cada una de las fuentes de 12V y -5V.

Para la fuente de 5V que es la que tiene más carga se utiliza uno de 200 μF , 16V y 4 condensadores de tantalum de 47 μF .

4.1.2. CONSIDERACIONES SOBRE LA DISTRIBUCION DE LAS SEÑALES DE DATOS Y DE CONTROL

El diseño de un sistema de memoria debe ser hecho dejando buenos márgenes de tiempo tanto como se posible, pues, la combinación de los parámetros trabajando en el peor de los casos sacan las señales más allá de las especificaciones.

En sistemas de alta velocidad donde es difícil dejar márgenes de tiempo debido a que las señales están cercanas a los límites de las especificaciones de los circuitos de memoria, se utilizan resistencias de amortiguación en las líneas de datos, direcciones y control las cuales mejoran las formas de onda.

Cuando las líneas de reloj son demasiado largas y la frecuencia de operación es alta, existe la tendencia a que se produzcan reflexiones y estas a la vez produzcan oscilaciones las cuales se pueden controlar también con las resistencias de amortiguación.

La razón por la cual se pueden producir reflexiones es que debido a la alta frecuencia de operación las líneas de reloj se convierten en líneas de transmisión,

En el presente diseño, a pesar de que las señales de tiempo no están muy cercanas a los límites de las especificaciones, se han utilizado resistencias de 33Ω , $1/4 W$ para mejorar las formas de onda.

Las distancias de las líneas de datos, direcciones y control deben ser lo más cortas posible, por este motivo, a los controladores de memoria MC 3480 y MC 3242 se los ha distribuido entre los integrados de memoria.

4.2 PRUEBAS DE FUNCIONAMIENTO

En el momento de encender el equipo, tanto en los integrados de memoria Hamming como en los de datos se escribe información aleatoria, y como ya se explicó en la sección 2-2, la información que debe escribirse en la memoria Hamming depende de la palabra de datos.

Por esta razón antes de utilizar el sistema de memoria con detección y corrección de errores comprendido en el rango de direcciones de 0000 a 3FFF (HEX) es necesario

que se realicen ciclos de escritura en todas estas localidades, escribiendo cualquier parámetro sin importar su valor.

De esta manera se escribe en la memoria Hamming la información generada por los circuitos de paridad y dependiente de la palabra de datos.

Luego de este proceso de inicialización, el sistema de memoria construido puede ser utilizado de manera correcta pudiendo hacer uso de la información de los leds indicadores de error.

El proceso de inicialización mencionado, debería ejecutarse automáticamente en el momento de encendido de la máquina. Pero ello implicaría que el programa que logra la inicialización este grabado en el programa monitor del microcomputador. Esto sin embargo queda pendiente para un trabajo futuro, y no se lo ha realizado en el presente trabajo por cuanto el microcomputador sobre el que se ha acoplado el sistema de memoria construido, es un equipo de laboratorio sobre el cual existen otros proyectos que nuevamente modificarían el programa monitor.

Al no ejecutarse automáticamente el programa de inicialización, se lo debe ejecutar manualmente y para ello se ha grabado este programa en la EPROM U 10 del microcomputador (ver apéndice C). El inicio del programa está en la dirección C537 y se lo ejecuta con el comando C537 G. Una vez hecho esto, si es que los leds indicadores de error están encendidos, es necesario oprimir el interruptor del reset. Con esto los leds se apagarán, y se volverán a prender únicamente cuando exista algún error. Se puede chequear que en el rango de memoria 0000 - 3FFF se encuentra escrito un valor que es igual al byte menos significativo de la dirección correspondiente.

Cabe indicar que si no se realiza este proceso de inicialización de memoria los circuitos indicadores de error van a estar activados sin embargo, la memoria en sí va a trabajar correctamente, esto es debido a lo siguiente: si leemos de una localidad de memoria (en el rango 0000 - 3FFF) en la cual se escribió previamente un dato va a ser recuperada correctamente.

Si leemos de una localidad de memoria (en el mismo rango) en la cual no se ha escrito nada previamente lo

cual implica que para esa localidad de memoria no se generarán los bits Hamming, los circuitos de paridad indicarán 0, 1, o más errores, pero como no escribimos nada previamente no nos interesa el resultado de la palabra leída.

Para realizar el proceso de inicialización también existe la posibilidad de ingresar el programa en el mismo rango de memoria que se debe inicializar.

A continuación se detalla el programa que a la vez puede ser utilizado para comprobar el buen funcionamiento del sistema. El programa escribe desde la localidad 000F hasta la 3FFF un valor que es igual al byte menos significativo de la dirección correspondiente.

0000	CE	000F	LDX	000F	
0003	86	0F	LDA	0F	
0005	A7	00	A1	STA, X	00
0007	4C		INC	A	
0008	08		INX		
0009	8C	4000	CPX	4000	
000C	26	F7	BNE	A1	
000E	3F		SWI		

PROGRAMA DE INICIALIZACION

Una vez que se ha ingresado el programa en la memoria del computador lo ejecutamos con el comando 0000 G. Después de aproximadamente un segundo se exhibe 000E 3F que corresponde a la última instrucción del programa lo cual implica que el programa terminó su ejecución y se pueden chequear los resultados esperados.

Si los leds indicadores de error se encuentran encendidos, se debe oprimir el reset del microcomputador y a menos que se produzca un error estos no se volverán a encender.

Para verificar el hecho de que el sistema está detectando y corrigiendo errores simples correctamente, podemos simular la falla de un integrado de memoria simplemente extrayéndolo del sistema. Podemos extraer un integrado de memoria Hamming o uno de datos pero solo uno.

Al prender el equipo podemos comprobar que la información que se escribe en la memoria se puede recuperar correctamente lo cual implica que se está detectando y corrigiendo el error producido por haber extraído el integrado. En este caso el led indicador de error (ver Figura 4-1) permanecerá prendido permanentemente a pesar de haber ejecutado el programa de inicialización y a pesar de oprimir el interruptor del Reset lo cual nos indica que existe un error simple.

El led indicador de error doble permanecerá apagado.

Como ejemplo podemos extraer el integrado de memoria que corresponde a la línea de datos D5 y prender el equipo en esta condición.

Para tener escrito en memoria algún valor conocido efectúese el programa de inicialización y oprímase el interruptor del reset.

Al no existir el integrado de memoria que corresponde a la línea de datos D5, en esta línea se tendrá 1 lógico constante. Por esta razón cuando leemos el contenido de alguna localidad de memoria y se obtienen valores que tienen en el bit 5 un 1 lógico no se producirá la indicación de error.

Por ejemplo los valores (en exadecimal) 2x, 3x, 6x, 7x, Ax, Bx, Ex, Fx, siendo x cualquier número exadecimal, poseen en el bit 5 un 1 lógico, y estos valores cuando son leídos de memoria no producirán la indicación de error a pesar de no existir el integrado correspondiente.

Cualquier otro valor leído de memoria si producirá la indicación de error, a la vez que éste es detectado y corregido.

Para verificar que el sistema detecta los errores dobles podemos extraer dos integrados de memoria y podemos chequear que tanto el led indicador de error como el led indicador de error doble (ver Figura 4-1) se encienden cuando de memoria se leen valores que dependen de los integrados extraídos.

Otra manera de simular la producción de un error sería con la ayuda de una fuente de radiación de partículas alfa las cuales

afectan el contenido de las células de memoria.

Cuando en funcionamiento normal del equipo se produce un error doble, cuya probabilidad como ya se ha explicado es muy pero muy reducida, el sistema, a más de encender el indicador de error, debería realizar una subrutina de interrupción la cual a la vez que suspende las actividades del microcomputador, pues los datos ya no son válidos, indica la localidad de memoria en la que se produjo ese error doble.

Esto se lograría conectando la misma salida del flip flop retenedor 7474, que va al cátodo del led indicador de error doble (Figura 3-3), a la línea $\overline{\text{NMI}}$ del microcomputador. Y a la vez teniendo grabado en el monitor del sistema la subrutina correspondiente.

Esta probabilidad no se le ha puesto en práctica en el presente trabajo por cuanto el microcomputador 6800 posee unicamente un nivel de interrupción $\overline{\text{NMI}}$, y este nivel ya ha sido utilizado para ejecutar la operación de rastreo "TRACE" (comando N).

4.3 MANTENIMIENTO

En esta sección se dan algunas sugerencias para solucionar algunos de los problemas que se podrían presentar en el sistema de memoria construido.

En caso de que no se pueda escribir ninguna información en el sistema de memoria, se deberá chequear en primer lugar que las fuentes de voltaje de +5V, +12V y -5V estén dentro de las especificaciones para los circuitos de memoria dados en el Apéndice A.

Si las fuentes de voltaje están dentro de las especificaciones o si estas han sido corregidas y se mantiene el problema es posible que los potenciómetros que gobiernan la duración de los tiempos t_1 , t_2 , t_3 y el oscilador de 64 KHZ hayan sido movidos. Para comprobar esto utilícese un osciloscopio y compruebe que las formas de onda obtenidas coinciden con las de la Figura 3-2.

Tal como se puede ver en la Figura 3-2, las transiciones positivas de las señales t_1 , t_2 , y t_3 son secuenciales y el tiempo que permanecen en OL son respectivamente 90, 142, y 186 ns y la diferencia entre las transiciones positivas son 46ns para $t_3 - t_2$ y 38ns para $t_2 - t_1$.

Para medir las señales t_1 , t_2 , y t_3 coloque la punta de prueba del osciloscopio respectivamente en los puntos: Pata 1 del circuito I8, Pata 1 del circuito I9 y Pata 1 del circuito I10 (Figura 4-1).

En caso de que estas señales esten fuera de las especificaciones ajústese P1, P2, y P3 (Figura 4-1) respectivamente.

Otra manera menos precisa para hacer este ajuste es la

siguiente: apagar el equipo, sacar la tarjeta de memoria del equipo y sin hacer ninguna modificación en la tarjeta medir la resistencia entre los terminales de los potenciómetros P1, P2 y P3 (Figura 4-1), los valores que deben obtenerse son: 3.6Ω , $1.918 \text{ K}\Omega$ y $2,79 \text{ K}\Omega$.

La señal del oscilador en la pata 3 del circuito U7 555 (Figura 4-1) debe ser mayor a 64KHZ es decir su período menor a 15.6 us. Un valor con el que se han desarrollado varias pruebas es 13.4 us.

Cuando el led indicador de error está encendido lo cual indica que se ha producido un error simple se debe probar el sistema con varias operaciones para chequear si es que el error es aleatorio o es permanente lo cual indicaría que un circuito de memoria ha fallado y debe ser reemplazado, en este caso debemos identificar cual es el integrado con problemas. El procedimiento que sigue nos ayuda para ello:

Vamos a chequear con la ayuda de un buen osciloscopio el valor que tienen las señales P0 - P3 en el momento de la transición positiva de \bar{L} (ver el circuito de indicación) de error en la Figura 3-3.

Para que las señales P0 - P3 sean claras es necesario que el microcomputador este realizando un lazo de operaciones en el rango de memoria con detección y corrección, para ello ejecutaremos el programa de inicialización con las siguientes modificaciones:

Cambiar la localidad	0002	de	OF	a	11
Cambiar la localidad	0004	de	OF	a	11
Cambiar la localidad	000E	de	3F	a	7E(JMP)
Escribir en la	000F	00			
Escribir en la	0010	00			

Las modificaciones realizadas hacen que el programa se ejecute indefinidamente.

Ejecutar el programa con el comando 0000 G.

Conectar el canal 1 del osciloscopio a la pata 10 del circuito I 45 7404, y el canal 2, sucesivamente en las patas 1, 2, 3, y 13 del circuito I43 que corresponden a las señales P3, P2, P1, P0 respectivamente.

Disparándose con el canal 1, chequear la transición positiva del canal 1 y observar el valor a ese tiempo del canal 2; la señal del canal 2 no va a estar bien definida porque tiene muchas variaciones a lo largo del tiempo; anotar los valores obtenidos en el canal 2 para las cuatro señales P3, P2, P1 y P0.

Estos valores lógicos forman en este orden una palabra apuntadora que nos indica el bit errado ya sea Hamming o de datos.

Referirse a la tabla 1-1 en el capítulo 1 para obtener, en base a la palabra apuntadora, el bit errado. Reemplazar el integrado de memoria correspondiente al bit errado por uno en buenas condiciones (Figura 4-1).



FIG. 4-2-A
LADO CON COMPONENTES



FIG. 4-2-B
LADO SIN COMPONENTES

TABLA 4 - 1

<u>ELEMENTO</u>	<u>ESPECIFICACION</u>			
C1	50 PF			
C2	65 PF			
C3	79 PF			
C4	0.001 MF	,	600 V	
C5	0.01 MF	,	50 V	
C6	0.1 MF	,	50 V	
C23	150 PF	,	50 V	
C26	6.8 MF	,	25 V	Tantalum
C27	100 MF	,	16 V	
C28	0.1 MF	,	50 V	
C29 - C32	47 MF	,		Tantalum
C33	220 MF	,	16 V	
C34	15 MF	,	25 V	Tantalum
C35	0.1 MF	,	50 V	
C36	100 MF	,	16 V	
C37	6.8 MF	,	25 V	Tantalum
C38	0.1 MF	,	50 V	
C41 - C53	0.01 MF	,	50 V	
C61 - C73	0.01 MF	,	50 V	
C81 - C93	0.01 MF	,	50 V	
F24	LED		(Indicador de error doble)	
F25	LED		(Indicador de error)	
P1	2K			
P2	2.7K			
P3	2.8K			
P4	100			
P5	2K			
R1 - R3	2.2 K	,	1/4 W	
R4	330	,	1/2 W	
R5	5.1 K	,	1/2 W	
R10 - R19	33	,	1/4 W	
R20 - R22	1.8 K	,	1/4 W	
R23	1 K	,	1/4 W	
R24	330	,	1/4 W	
R25	470	,	1/4 W	

TABLA 4 - 1 (Cont.)

<u>ELEMENTO</u>	<u>ESPECIFICACION</u>	
I 3	MC 3480	Controlada de memoria dinámica
I 4	MC 3242	Multiplexer de direcciones
I 5	74 LS 240	"Octal buffer" inversor
I 7	555	"Timer"
I 8 - I 10	74121	Multivibrador monoestable
I 12	7474	Flip flop tipo D
I 13 - I 14	8T26	"Quad Three-state bus transceiver"
I 15 - I 22	UPD 416-2	Memoria RAM dinámica de 16384 x 1
I 24 - I 28	UPD 416-2	"
I 23	74 LS 244	"Octal buffers" no inversor
I 29	74 LS 244	"
I 30 - I 35	74 S 280	Generador de paridad par/impar de 9 bits
I 36 - I 37	74 LS 244	
I 38	74154	Decodificador de 4 a 16
I 39	74 LS 244	
I 40 - I 41	7486	OR EX
I 42	74 LS 240	
I 43	74 LS 260	Nor de 5 entradas
I 44	74 LS 32	OR
I 45	7404	NOT
I 46	7408	AND
I 47	7486	OR EX
I 48	74 LS 00	NAND
I 49	7474	Flip flop tipo D
I 50	7474	Flip flop tipo D

COMENTARIOS Y CONCLUSIONES

COMENTARIOS Y CONCLUSIONES

Debido al avance de la tecnología dentro del campo de los sistemas de computación, se hace necesario aumentar su confiabilidad. Por esta razón, en los sistemas de memoria se incluyen sistemas de detección y corrección de errores.

En el presente trabajo se construye un sistema de 16 kbytes de memoria dinámica con detección y corrección de errores el cual se acopla al microcomputador 6800 D2.

El sistema detecta y corrige errores simples (1 bit errado) y detecta los dobles utilizando para ello el código hamming y 5 bits adicionales de memoria en la que se almacena información generada por circuitos de paridad y que está relacionada con la palabra de datos.

Los errores más generalizados que se pueden producir en los circuitos de memoria, son errores transitorios "soft error" los cuales pueden ser producidos por radiaciones alfa provenientes del material (cerámica, cuarzo) que conforma la empaquetadura del integrado de memoria. La relación de error obtenida en experimentos realizados, muestran que para un sistema de 16 kbytes de memoria dinámica se produce un error en 5 millones de horas de operación.

Por esta razón, la probabilidad de que se produzca más de un error es realmente muy pequeña, teniendo en cuenta este análisis, el sistema construído, logra la detección más no la corrección de errores dobles. Se debe tomar en cuenta que la complejidad de un código aumen-

ta considerablemente cuando se intenta corregir los errores dobles.

El código hamming utilizado requiere de 4 bits para la detección y corrección de errores simples. Esto es porque con 4 bits se pueden designar 2^4 (=16) localizaciones diferentes y para bytes de 8 (2^3) bits, un código de 4 bits es más que suficiente. Se debe notar que el código detecta y corrige errores simples tanto en los bits de datos como en los bits hamming, por esta razón un código de 3 bits no es suficiente.

Para lograr la detección de errores dobles, se utiliza un bit adicional para obtener la paridad global de los bits hamming y de datos. De esta manera, para un byte (8 bits), el número total de bits de chequeo usado es cinco.

Cabe indicar que con un bit adicional, el código hamming puede detectar y corregir errores en palabras de 16 bits.

Cuando se produce un error doble, el sistema lo detecta y debería parar la ejecución del microprocesador a la vez que se ejecuta una subrutina de interrupción la que indicaría la dirección en la que se produjo el error doble. Pero, como ya se indicó en el capítulo 4, el microprocesador 6800 tiene solamente un nivel de interrupción NMI y este ya está utilizado para ejecutar la operación de rastreo "Trace" la cual se realiza con el comando "N".

Por esta razón, cuando se produzca un error doble, el circuito indica

dor de error doble se activará, pero no se suspenderá la ejecución del microprocesador y se producirán resultados incorrectos.

Cuando se produzca un error triple, éste para el sistema será igual que un error simple y se producirán resultados incorrectos.

Para medir la relación de error en memorias RAM dinámicas, un método experimental consiste en acelerar los efectos de error en la memoria causados por radiaciones alfa.

Esto implica que el sistema de memoria, se debe exponer a radiaciones alfa más fuertes que las radiaciones producidas por residuos en la empaquetadura del integrado.

Luego la relación obtenida es extrapolada a los niveles de radiación alfa de la empaquetadura.

La ventaja de este método es que todo el test se lo puede ejecutar en pocas horas.

Para el diseño de los integrados de memoria RAM se realizan estos experimentos variando sus parámetros y observando las relaciones obtenidas.

De estos experimentos se concluye que con el aumento de la fuente de voltaje Vdd y con la disminución de la frecuencia de operación, se aumenta la resistencia de los integrados de memoria RAM a las partículas alfa. Esto es debido a que la fuente Vdd controla la carga abso-

luta de una célula de memoria RAM dinámica, este voltaje también determina la diferencia de voltaje entre la célula y la célula de comparación. Por otro lado, con la disminución de la frecuencia de operación, se disminuye también el consumo de potencia.

En el capítulo 4 se mencionó que sería conveniente incluir en el monitor del microcomputador el programa de inicialización el cual escribe un parámetro en el rango de memoria 0000 - 3FFF que contiene detección y corrección de errores (esto es para que la información de los bits hamming sea la generada por la palabra de datos, situación que no se cumple al momento de prender el equipo). Pues, también sería conveniente incluir en el mismo monitor un programa que realice un "lavado" de memoria, y que debería ejecutarse cuando el procesador está libre de operación.

El "lavado de memoria" consiste en leer de todas las localidades de memoria (si hay un error simple, automáticamente se produce la corrección de ese error) y reescribir esa información. Debido a este frecuente acceso, se asume que la probabilidad de errores dobles aunque finita es poco probable.

En el sistema se podría incluir registros retenedores adicionales o poseer un sistema de memoria de recuperación para almacenar la condición de P0-P3 y Pp cuando se produce un error.

De esta manera, bajo comandos del usuario se los podría decodificar y exhibir en indicadores luminosos con lo que se podría determinar

inmediatamente cuales fueron los bits errados.

Adicionalmente se podrían almacenar las direcciones en las que se producieron error.

Estos métodos serían muy aplicables sobre todo cuando se trata de obtener datos estadísticos de la producción de errores ayudando de esta manera a la mejora del diseño de los circuitos integrados.

Para el refresco de la memoria del sistema construido, se ha utilizado el método transparente.

Este método, a diferencia del método de refresco de robo, tiene la gran ventaja de no parar el microprocesador para realizar el refresco.

Esto tiene fundamental importancia cuando el sistema está diseñado para llevar un conteo de los ciclos de microprocesador ejecutados, por ejemplo en aplicaciones como en telecomunicaciones, este método resulta ideal. En cambio, el uso del método alternativo: Refresco de robo, produciría resultados incorrectos.

Sin embargo, hay que considerar que, debido a que en un ciclo de microprocesador, utilizando el refresco transparente; se debe ejecutar un ciclo de refresco y uno de lectura/escritura, la frecuencia del reloj del microprocesador no puede ser muy alta y a la vez, el tiempo de acceso de la memoria utilizada debe ser rápida.

En el diseño realizado, se cumple con las dos condiciones pues, el re

loj del microprocesador opera a una frecuencia de 614.4 KHZ (1627 ns) y el tiempo de acceso de la memoria utilizada (UPD 416-2) es de 200 ns.

Por otro lado, cuando en alguna aplicación se utiliza un procesador sumamente rápido cuyo período es comparable con el tiempo de acceso de la memoria, inevitablemente, se debe parar el microprocesador para realizar un ciclo de refresco, debiéndose utilizar para ello el método de refresco de robo.

Es decir, dependiendo de la aplicación del sistema, se escoje uno u otro tipo de refresco.

Se debe notar además que el método de detección y corrección de errores puesto en práctica en este trabajo, posee muchas demoras de tiempo a través de los varios circuitos TTL, razón por la cual, si el procesador hubiese sido más rápido, se debería haber pensado en el método de refresco de robo.

APENDICE A
HOJAS DE ESPECIFICACIONES

MOTOROLA SEMICONDUCTORS

PO BOX 20812 PHOENIX, ARIZONA 85086

MC3480

Advance Specifications and Applications Information

MEMORY CONTROLLER FOR 16 PIN 4K AND 16K DYNAMIC RAMS

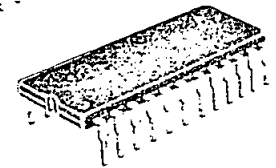
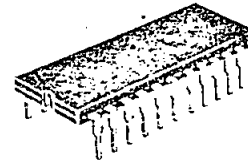
The memory controller chip is designed to greatly simplify the interface logic required to control the popular 16 pin 4K or 16K dynamic NMOS RAM in a microprocessor system such as the M6800. The controller will generate, on command from the microprocessor, the proper timing signals required to successfully transfer data between the microprocessor and the NMOS memories. The controller, in conjunction with an oscillator, will also generate the necessary signals required to insure that the dynamic memories are refreshed for the retention of data.

- Greatly Simplify the MPU-Dynamic Memory Interface
- Reduce Package Count and System Access/Cycle Times 30%
- Chip Enable for Expansion to Larger Word Capacity
- Generate 1 of 4 RAS Signals for an Optimum 16K/64K Memory System
- High Input Impedance for Minimum Loading of MPU Bus
- Schottky TTL Technology for High Performance
- Useful with 4K and 16K and Future Expanded Dynamic RAMs

DYNAMIC MEMORY CONTROLLER

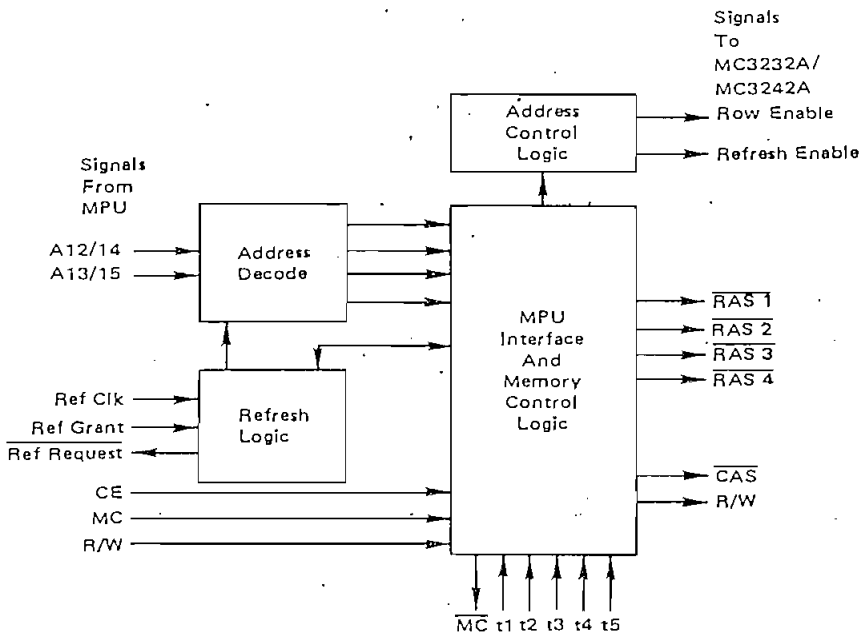
SCHOTTKY MONOLITHIC INTEGRATED CIRCUIT

L SUFFIX
CERAMIC PACKAGE
CASE 623



P SUFFIX
PLASTIC PACKAGE
CASE 649

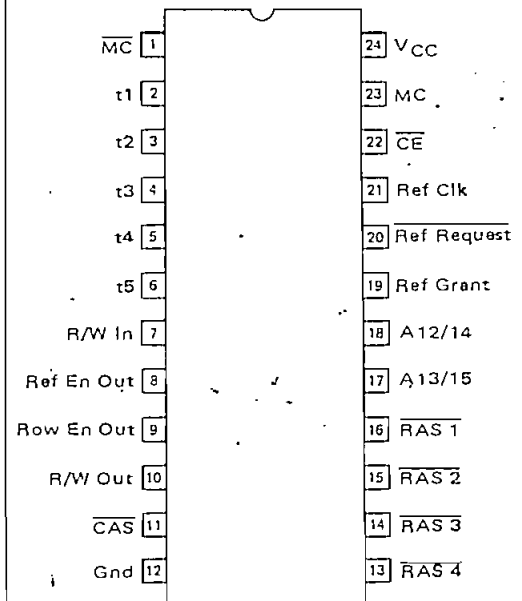
BLOCK DIAGRAM



Several methods may be employed to generate the required time delay:

1. One shots
2. High frequency counters
3. High frequency shift registers
4. Delay lines
5. Signals from MPU Clock

PIN CONNECTIONS



See Pin Descriptions

ABSOLUTE MAXIMUM RATINGS (Note 1)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	7.0	Vdc
Input Voltage	V_I	-0.5 to +7.0	Vdc
Output Voltage	V_O	-0.5 to +7.0	Vdc
Operating Ambient Temperature	T_A	0 to +70	°C
Storage Temperature	T_{stg}	-65 to +150	°C
Operating Junction Temperature	T_J		°C
Ceramic Package		175	
Plastic Package		150	

Note 1: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. They are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

Due to the advanced nature of this specification, final electrical limits are not yet given on all parameters. A final version may be obtained after October, 1978 by writing:

RECOMMENDED OPERATING CONDITIONS

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	+4.75 to +5.25	Vdc
Operating Ambient Temperature Range	T_A	0 to +70	°C

Motorola Linear IC Marketing — M250
P.O. Box 20912
Phoenix, AZ 85036

ELECTRICAL CHARACTERISTICS (Unless otherwise noted specifications apply over recommended power supply and temperature ranges. Typical values measured at $V_{CC} = 5.0$ V and $T_A = 25^\circ\text{C}$.)

Characteristic	Symbol	Min	Typ	Max	Unit
Input Voltage — Low Logic State	V_{IL}	—	—	0.8	V
Input Voltage — High Logic State	V_{IH}	2.0	—	—	V
Input Current — Low Logic State ($V_{IL} = 0.5$ V)	I_{IL}	—	—	-250	μA
Input Current — High Logic State ($V_{IH} = 2.7$ V) ($V_{IH} = 5.5$ V)	I_{IH}	—	—	40 100	μA
Input Clamp Voltages ($I_{IC} = 18$ mA)	V_{IC}	—	—	-1.5	V
Output Voltage — Low Logic State ($I_{OL} = 24$ mA for RAS, CAS, and R/W) ($I_{OL} = 8.0$ mA for Row En, Ref En, MC, Ref Req)	V_{OL}	—	—	0.5 0.5	V
Output Voltage — High Logic State ($I_{OH} = -1.0$ mA for RAS, CAS, and R/W) ($I_{OH} = -0.4$ mA for Row En, Ref En, and MC) ($I_{OH} = -0.2$ mA for Ref Req) (Note: Ref Req output has internal 5.0 k resistive pullup to V_{CC} .)	V_{OH}	3.0 2.4 2.4	— — —	— — —	V
Power Supply Current	I_{CC}	—	—	70	mA
Output Short-Circuit Current ($V_{OL} = 0$ V for Row En, Ref En, and MC)	I_{OS}	-10	—	-55	mA



SWITCHING CHARACTERISTICS Typical values measured at $V_{CC} = 5.0$ V, $T_A = 25^\circ\text{C}$

Characteristic	Symbol	Min	Typ	Max	Unit
Propagation Delay Times					
MC to $\overline{\text{MC}}$	tPMC	—	8	—	ns
t1 to $\overline{\text{RAS}}$	tPT1	—	30	—	
t2 to Row En	tPT2	—	30	—	
t3 to CAS	tPT3	—	30	—	
t4 to R/W	tPT4	—	30	—	
t5 to CAS, $\overline{\text{RAS}}$	tPT5A	—	30	—	
t5 to R/W, Row En, Ref En	tPT5	—	45	—	
Ref Clk to $\overline{\text{Ref Req}}$	tPCQ	—	45	—	
Ref Grant to Row En or Ref En	tPGR	—	30	—	
t1 to Ref Req (Refresh Cycle only)	tPTQ	—	30	—	
Setup Times					
Ref Clk before Ref Grant	tSCG	—	30	—	ns
A12, A13 before t1	tSAT	—	-10	—	
R/W Input before t4	tSRW	—	10	—	
$\overline{\text{CE}}$ before t1	tSCE	—	15	—	
Hold Times					
A12, A13 after t5	tHAT	—	10	—	ns
$\overline{\text{CE}}$ after t1	tHCE	—	0	—	
R/W after t4	tHRW	—	0	—	
Minimum Delay Times (see Note 2)					
t1 Low to High to t2 Low to High	tDEL (1-2)	—	30	—	ns
t1 Low to High to t4 Low to High	(1-4)	—	30	—	
t2 Low to High to t3 Low to High	(2-3)	—	30	—	
t3 Low to High to t5 Low to High	(3-5)	—	30	—	

Note 2: If delays between pins are less than the minimum specified, the succeeding outputs may not switch.

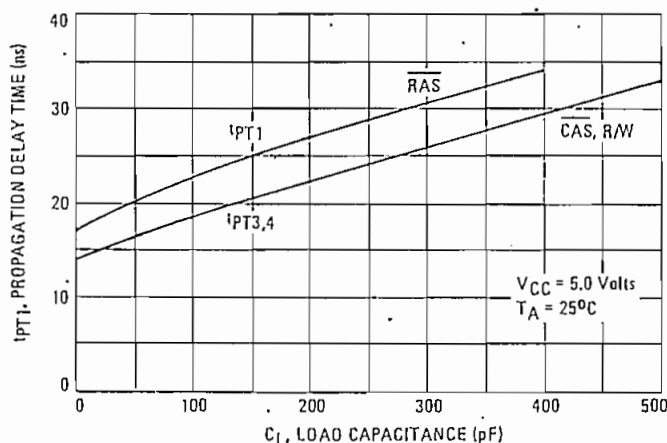
AC Loads (Note 3)

R/W and $\overline{\text{CAS}}$ Outputs	450 pF to Gnd*
$\overline{\text{RAS}}$ Outputs	150 pF to Gnd*
MC, Row En, Ref En, and Ref Req Outputs	15 pF to Gnd*

*Includes probe and jig capacitance.

NOTE 3: All outputs can drive larger capacitive loads than those shown with a small decrease in speed. See Figure 1.

FIGURE 1 — TYPICAL tPT1,3, and 4 (HIGH TO LOW) versus LOAD CAPACITANCE — RAS, CAS and R/W



Name	No.	Function
RAS1	16	Row Address Strobe pins which connect to each of the dynamic RAMs to latch in Row Address on memory chips. Decoded to 1 of 4 during R/W cycle. All 4 go low during refresh cycle.
RAS2	15	
RAS3	14	
RAS4	13	
CAS	11	Column Address Strobe pin which connects to each dynamic RAM to latch in column address.
R.W Out	10	This pin signals the dynamic RAM whether the RAM is to be read from or written into.
Row En	9	Row Enable output which goes to the MC3232A (MC3242A). It signals the Address Multiplexer that the lower half (Row Addresses) or the upper half (Column Addresses) of the address lines are to be multiplexed into the dynamic RAM address inputs. A Logic 1 on this output indicates the Row Addresses, and a Logic 0 indicates Column Addresses.
Ref En	8	Refresh Enable output. A Logic 1 signals the Address Multiplexer that a refresh cycle is to be done, and a Logic 0 indicates that address multiplexing should be done.
\overline{CE}	22	Chip Enable Input. A Logic 1 on this pin disables all chip functions, except that of Refresh and the \overline{MC} output. \overline{CE} must be low during t_1 low to high transition to initiate R/W cycle. Once t_1 is initiated, the cycle is independent of \overline{CE} .
R/W In	7	The Read/Write input pin receives information from the M6800 MPU as to the direction of data exchange in the dynamic RAM. It transmits a Logic 0 to the R/W output for a Write Cycle and a Logic 1 for a Read Cycle.
A12 (A14)	17	Upper Order Address lines from the M6800. These two inputs decode to four signals controlling the four RAS outputs. A14 and A15 apply to 16K RAMs.
A13 (A15)	18	
MC	23	Memory Clock input which comes from the microprocessor clock generator. This input must fall after the falling edge of t_1 to avoid a potential internal race condition.
\overline{MC}	1	The complement of MC. It is a buffered output which may be used to drive the circuitry creating the time delays used on inputs t_2 through t_5 .
t_1	2	These pins use external timing delays to sequentially select the outputs to be enabled. They are positive-edge triggered inputs. Assuming a Read/Write cycle is to be executed, a positive edge on t_1 forces a logic 0 on one of the four RAS outputs as determined by the A12/14, A13/15 inputs. After a delay, a positive edge on t_2 causes Row En to go to a Logic 0, providing address-multiplexing information to the MC3232A or MC3242A. t_3 enables the CAS output and it goes low. t_4 enables the R/W output and it goes low, assuming the R/W input was low. t_5 resets all the outputs to a Logic 1 (with the exception of \overline{MC} , Ref En, and Ref Req). The inputs t_1 , t_2 , t_3 , and t_5 are daisy-chained, so they must be sequentially driven to obtain the desired output signals. t_4 can be driven at any time after t_1 .
t_2	3	
t_3	4	
t_4	5	
t_5	6	
Ref Clk	21	The 32 kHz (64 kHz) Refresh Clock signals this pin that another refresh cycle is required. It is a positive-edge triggered input, and upon triggering, the Ref Req pin goes to a Logic 0.
Ref Req	20	The Refresh Request output acts as an input to the MPU system, requesting a refresh cycle. This output has a 5 k Ω pullup resistor to the V_{CC} supply to allow wire-ORing if desired.
Ref Grant	19	Through the Refresh Grant input, the MC6875 initiates a refresh cycle. This input is positive-edge triggered and is enabled only after the Ref Req pin has gone low. This allows the MC3480 to discern between a Refresh Grant or a DMA Grant even though they appear on the same line. When employing both dynamic memory (refresh) and DMA in a microprocessor-based system with a combined Refresh/DMA Request control on the clock, provision must be made for holding off a DMA request during a refresh period (and visa versa). If this provision is not made, clock stretching (cycle stealing) will continue indefinitely and dynamic microprocessor data will be lost. The positive edge on Ref Grant causes Row En output to go low and Ref En output to go high. This signals the MC3232A (MC3242A) that a refresh address is required. The refresh cycle occurs with the succeeding pulses on t_1 - t_5 . A positive edge on t_1 causes Ref Req to go high and all the RAS outputs to go low. A positive going edge on t_2 causes no change in the outputs, since it controls the address multiplexing (Row En) during the Read/Write cycles. There is no output change when t_5 and t_4 go high because no CAS or R/W signal is needed during refresh. A positive edge on t_5 resets the RAS to a Logic 1 state, and Ref En to a Logic 0 state, ready for the next Read/Write cycle.
V_{CC}	24	+5.0 V supply. A 0.1 μ F capacitor is recommended to bypass pin 24 to ground.
Gnd	12	System Ground.

*These outputs are designed to drive the highly capacitive inputs of multiple dynamic RAMs (150 pF for RAS outputs, and 450 pF for CAS and R/W outputs). Consequently, these outputs have no short circuit limit and must be handled accordingly. Note: All other outputs are LS TTL totem pole configuration unless otherwise noted.

TIME DELAY INFORMATION

TIMING REQUIREMENT CONSTRAINTS

- Δt_1 Minimum is determined by MPU Address Delay (t_{AD}), plus RAM Row Address Set-Up Time (t_{ASR}), minus MC3480 Propagation Delay (t_{PT1}).
- $\Delta t_2 - \Delta t_1$ Minimum is determined by RAM Row Address Hold Time (t_{RAH}) minus the minimum MC3232A/3242A Row Enable to Output Delay (t_{O0MIN}).
- $\Delta t_3 - \Delta t_2$ Minimum is determined by RAM Column Address Set-Up Time (t_{ASC} minimum) plus maximum MC3232A/3242A Row Enable to Output Delay (t_{O01MAX}).
- $\Delta t_4 - \Delta t_3$ No Minimum
- $\Delta t_5 - \Delta t_3$ Minimum is determined by RAM minimum CAS Pulse Width (t_{CAS}) or Access Time from CAS (t_{CAC}) plus Data Set-Up Time of MPU (t_{DSR}).
- $\Delta t_5 - \Delta t_4$ Minimum is determined by the RAM minimum Write Pulse Width (t_{WP}).

*The MC3480 has propagation delays also, but they cancel each other in these delay requirements.

Note: Also required in computing time delays are the various delays incurred by the particular delay scheme used; i.e., delays between $4 \times f_o$, $2 \times f_o$, and f_o from the MC6875 which are used as inputs or the gate delays of the gates used in Figures 4A through 4C.



TYPICAL APPLICATION 16K X 8-BIT MEMORY SYSTEM FOR M6800 MPU

Note: Numbers in parenthesis indicate part types or values for 16K x 1 RAMs

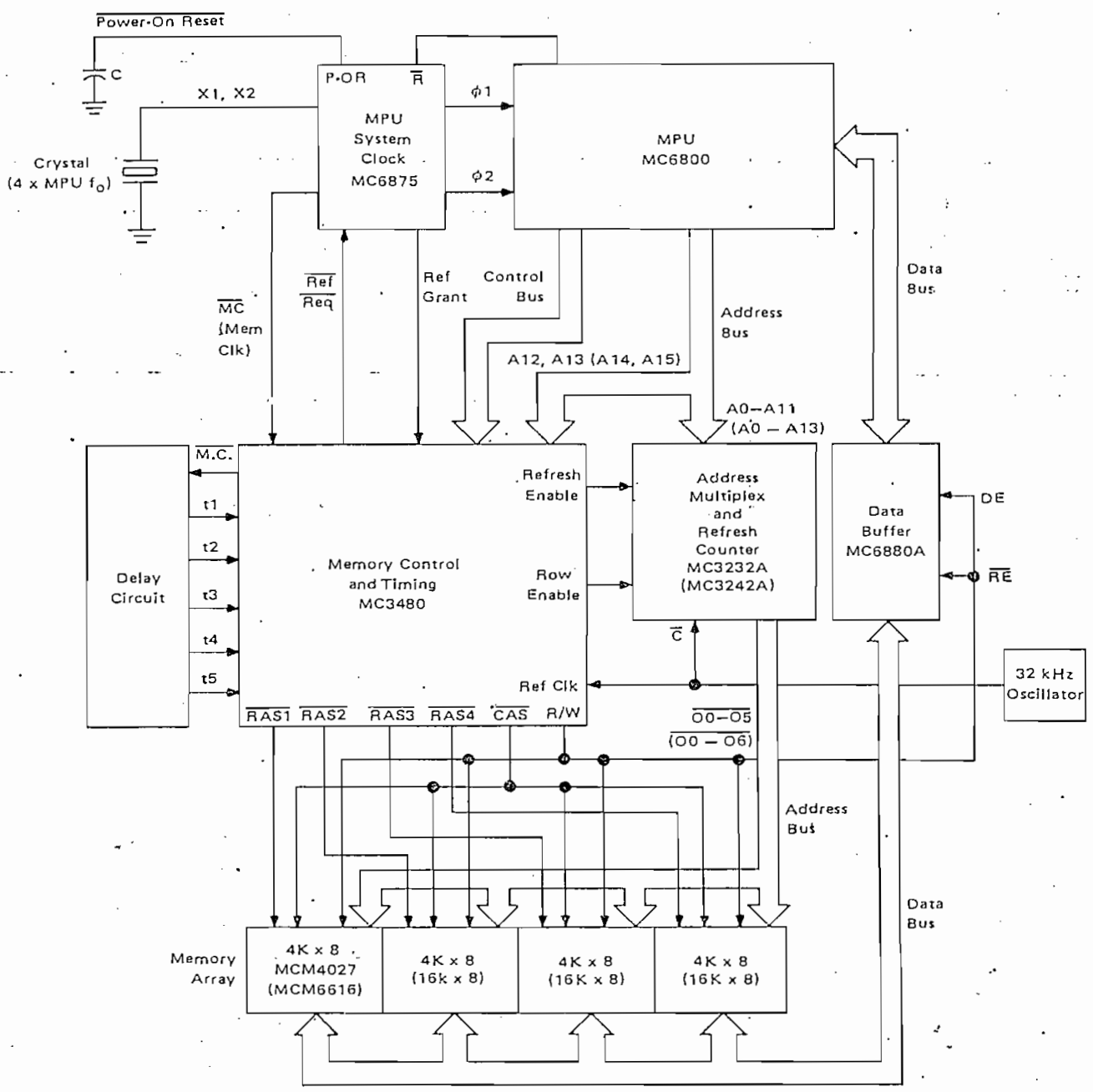


FIGURE 2 - READ/WRITE TIMING CYCLE

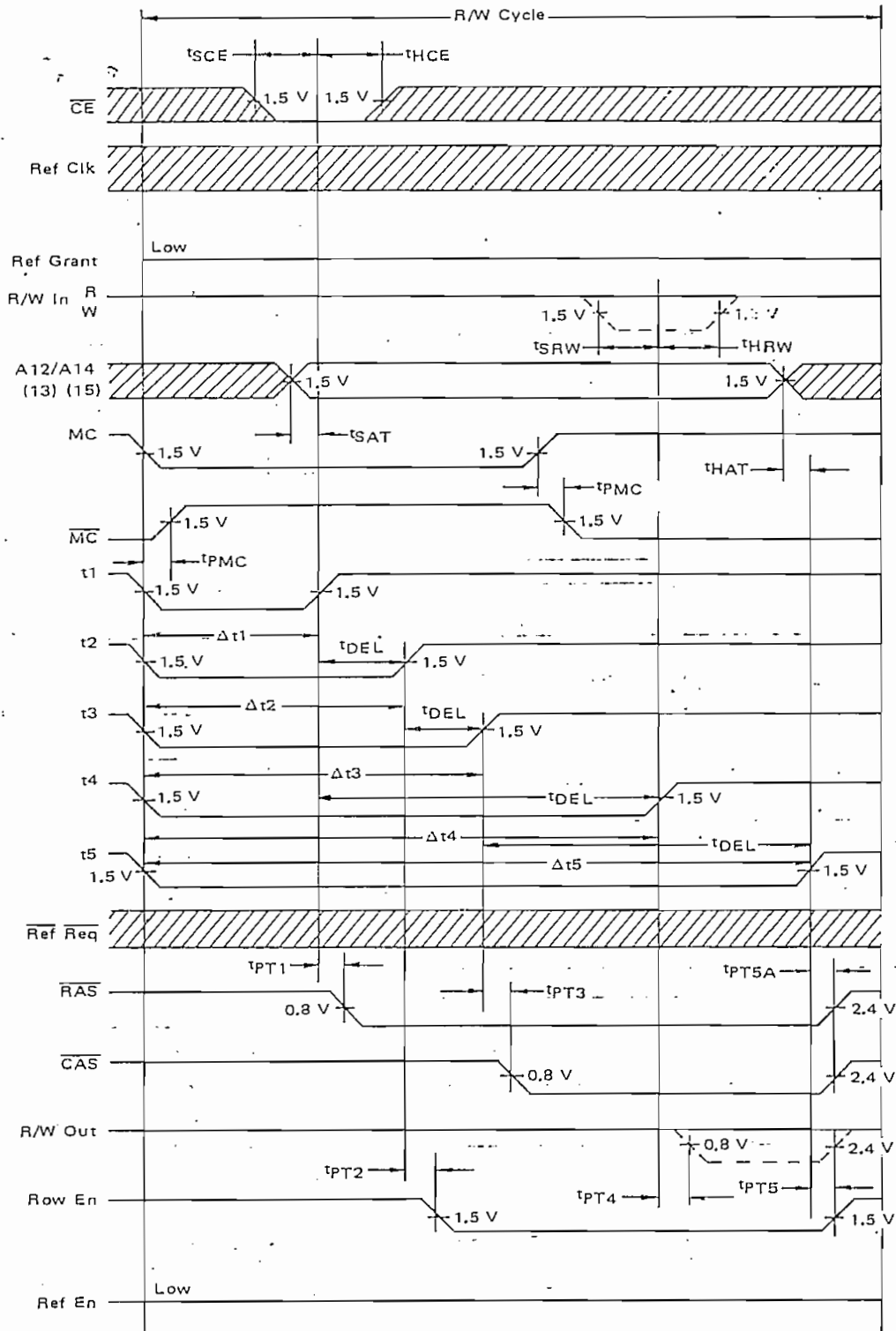
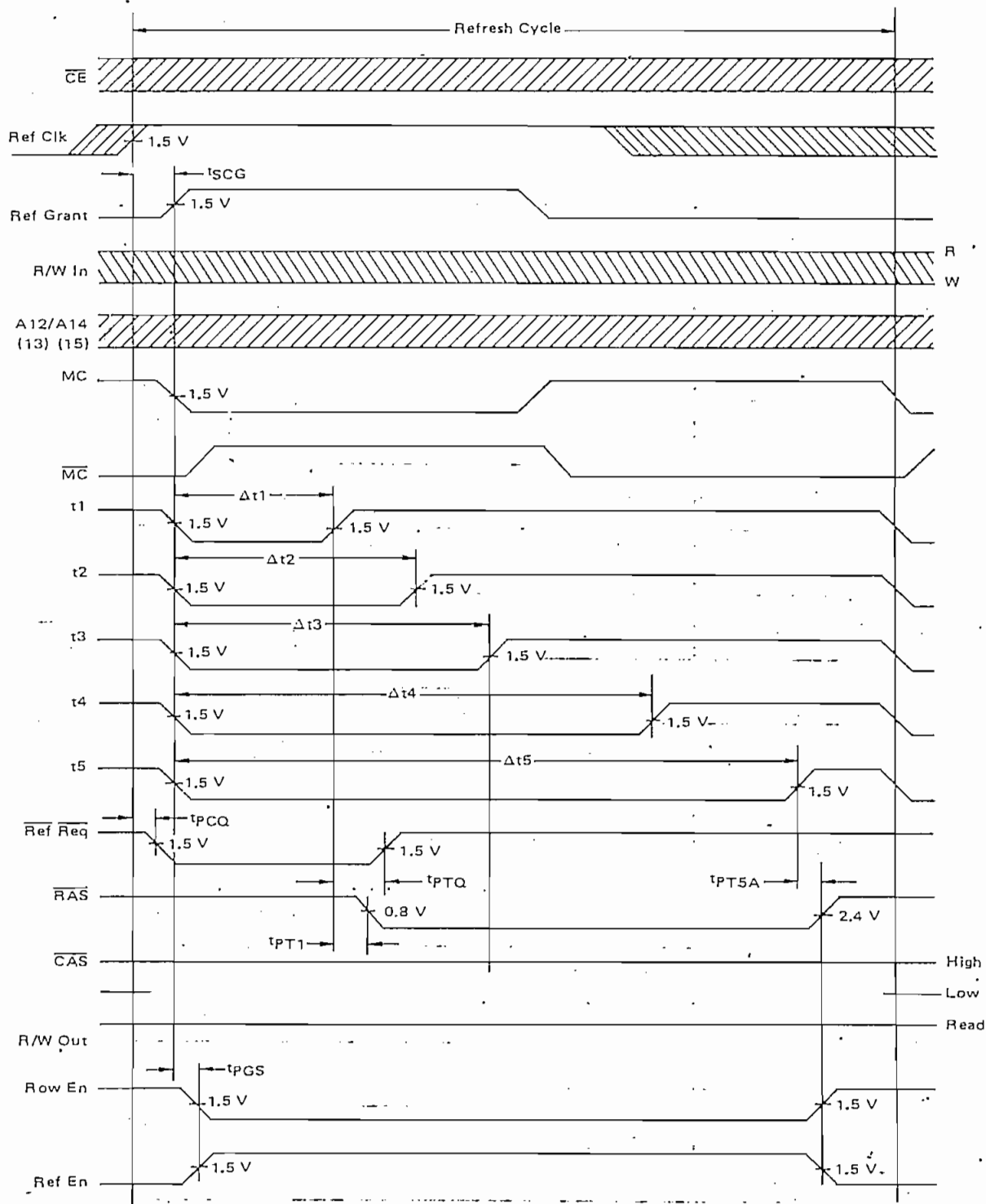


FIGURE 3 — REFRESH TIMING CYCLE



APPLICATIONS INFORMATION

GENERAL DESCRIPTION

The MC3480 uses five general timing inputs in place of a master clock with on-chip timing generation. This gives the system designer optimum flexibility in interfacing with the various microprocessor families and dynamic memories that are available. In simpler slow speed

systems, the timing signals required can be directly obtained from those available from the microprocessor. In systems requiring high speed memory/microprocessor cycle times, timing input t1-t5 can be obtained using delay lines or a range of techniques as shown in Figures 4 thru 8. It is only necessary to maintain the time delay relationships shown under time delay information.

FIGURE 4 - UNIVERSAL TIME DELAY USING MC6875

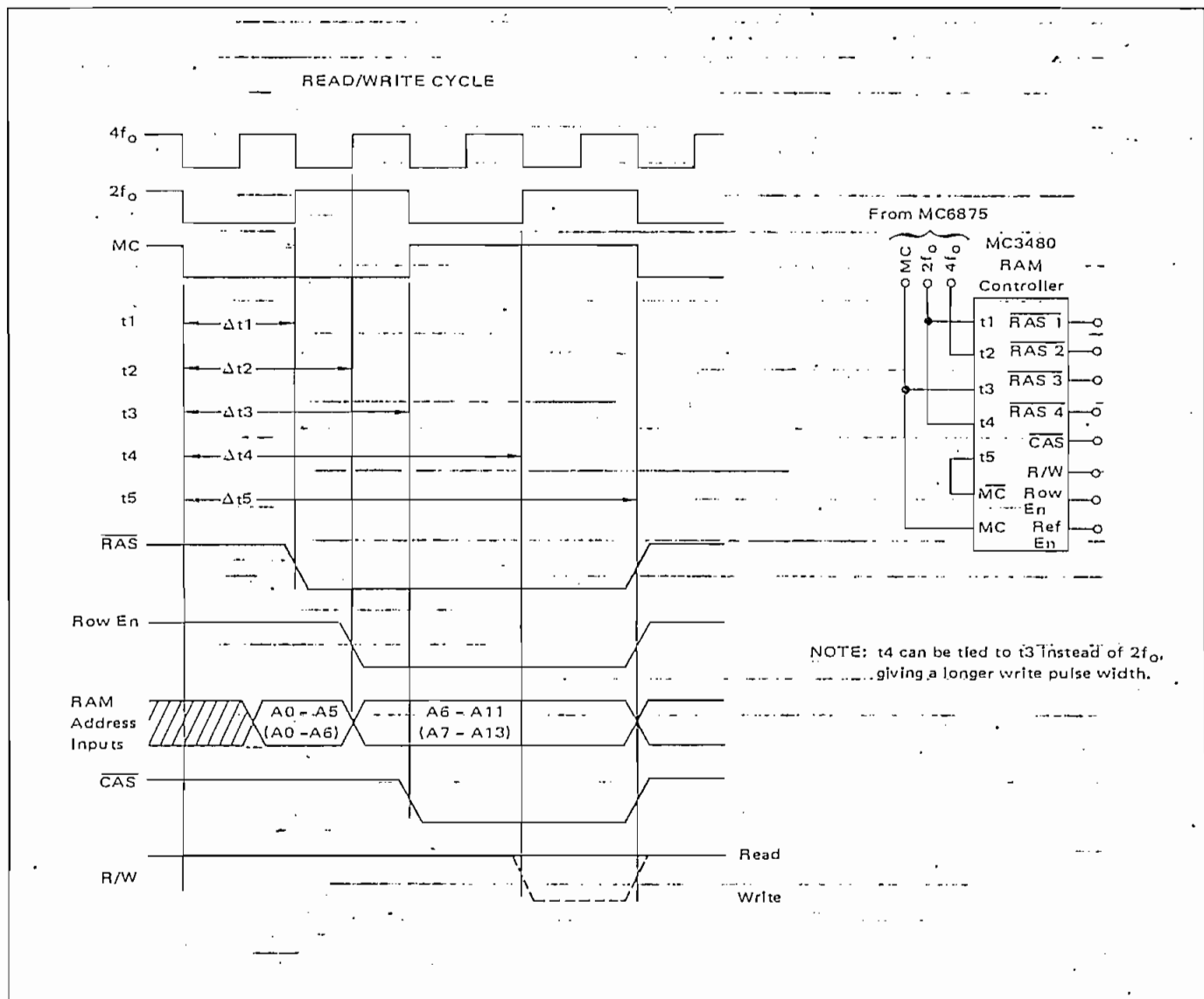


FIGURE 5 - ALTERNATE TIME DELAYS USING MC6875
(Read/Write Cycle Shown)

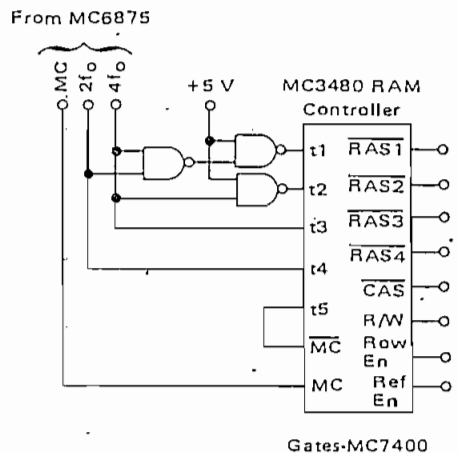
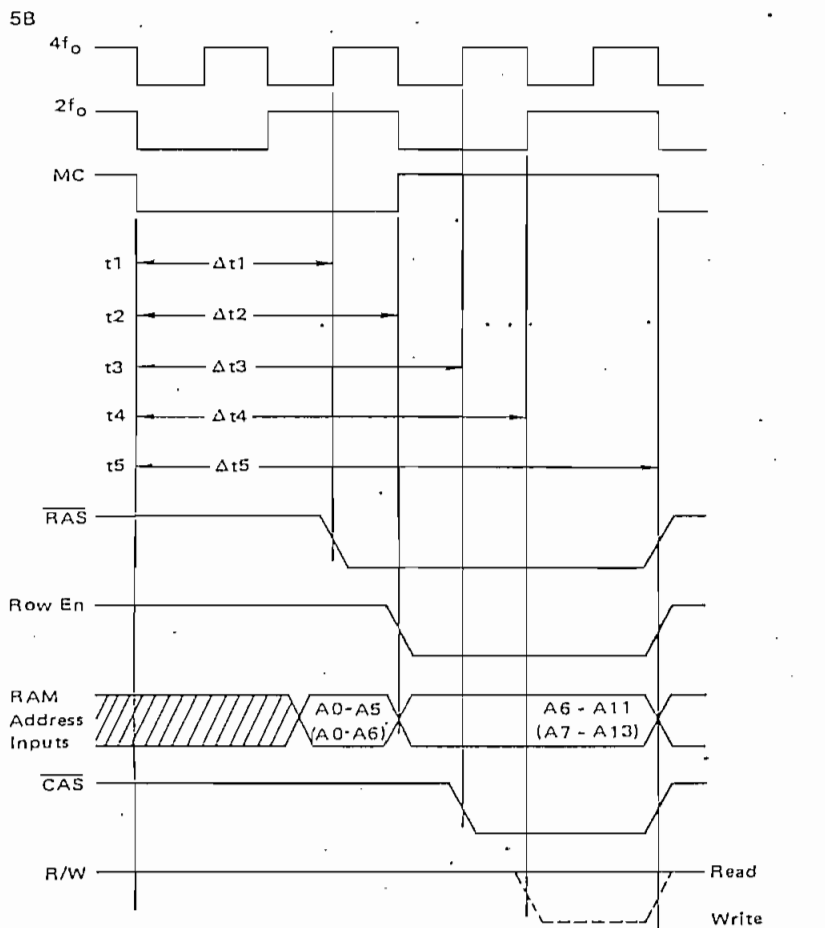
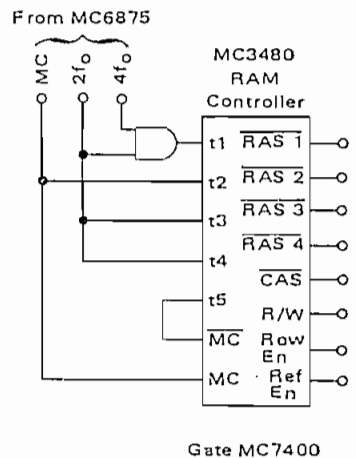
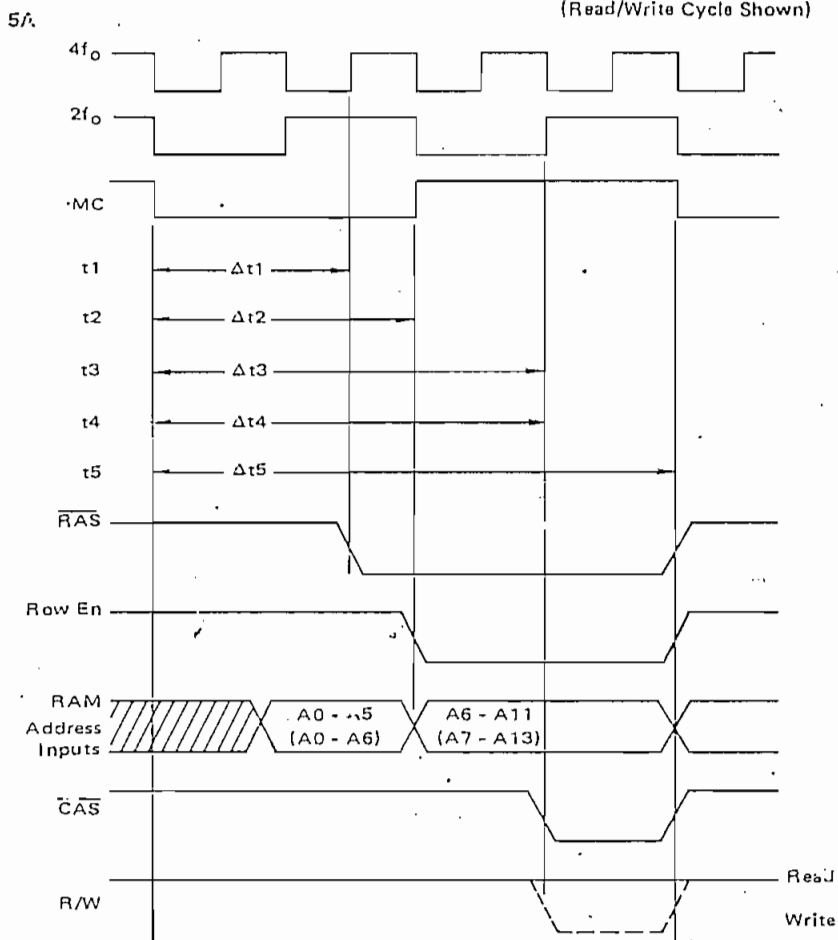


FIGURE 5C - ALTERNATE TIME DELAYS USING MC6875

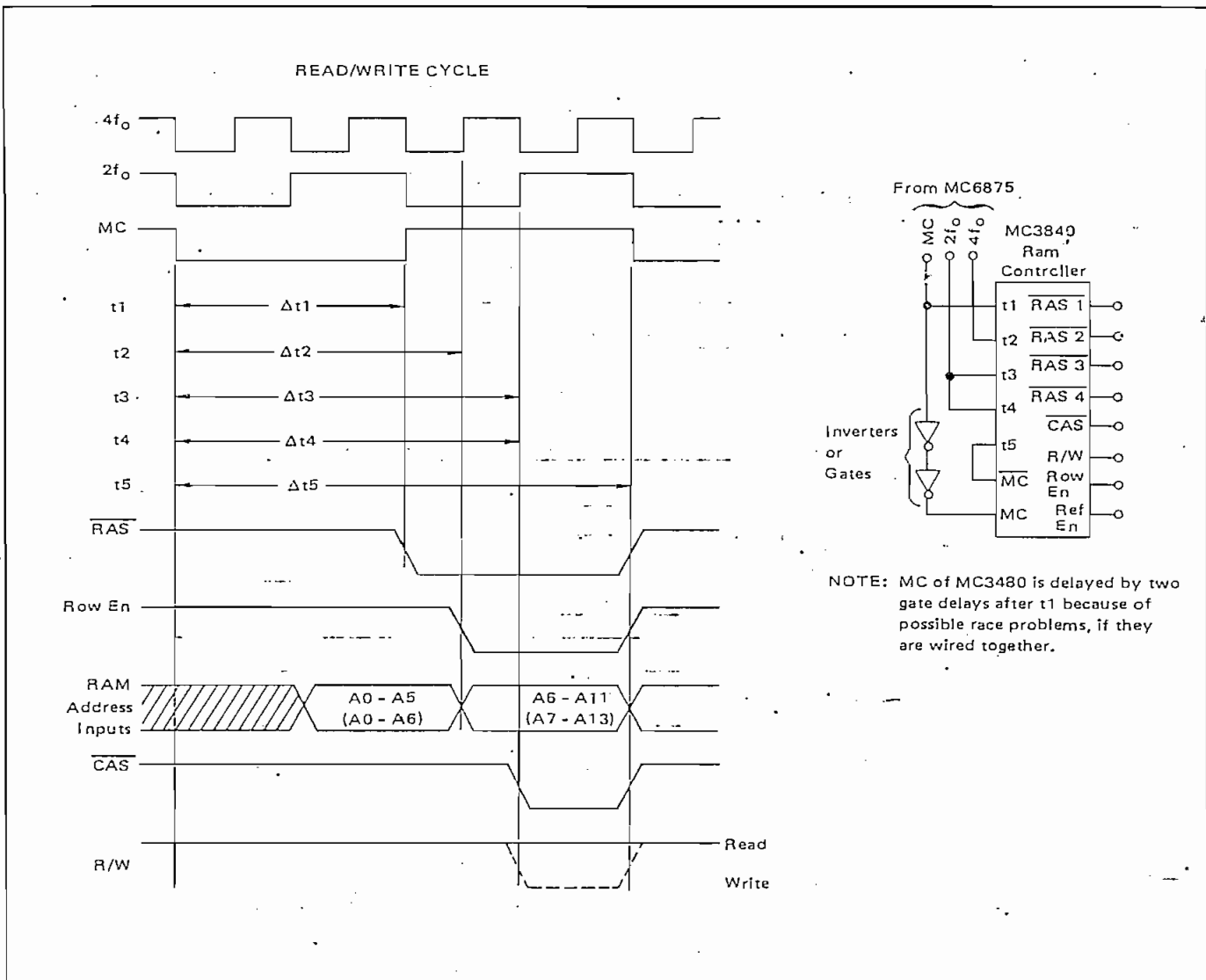


FIGURE 6 - ONE SHOT TIME DELAY METHOD

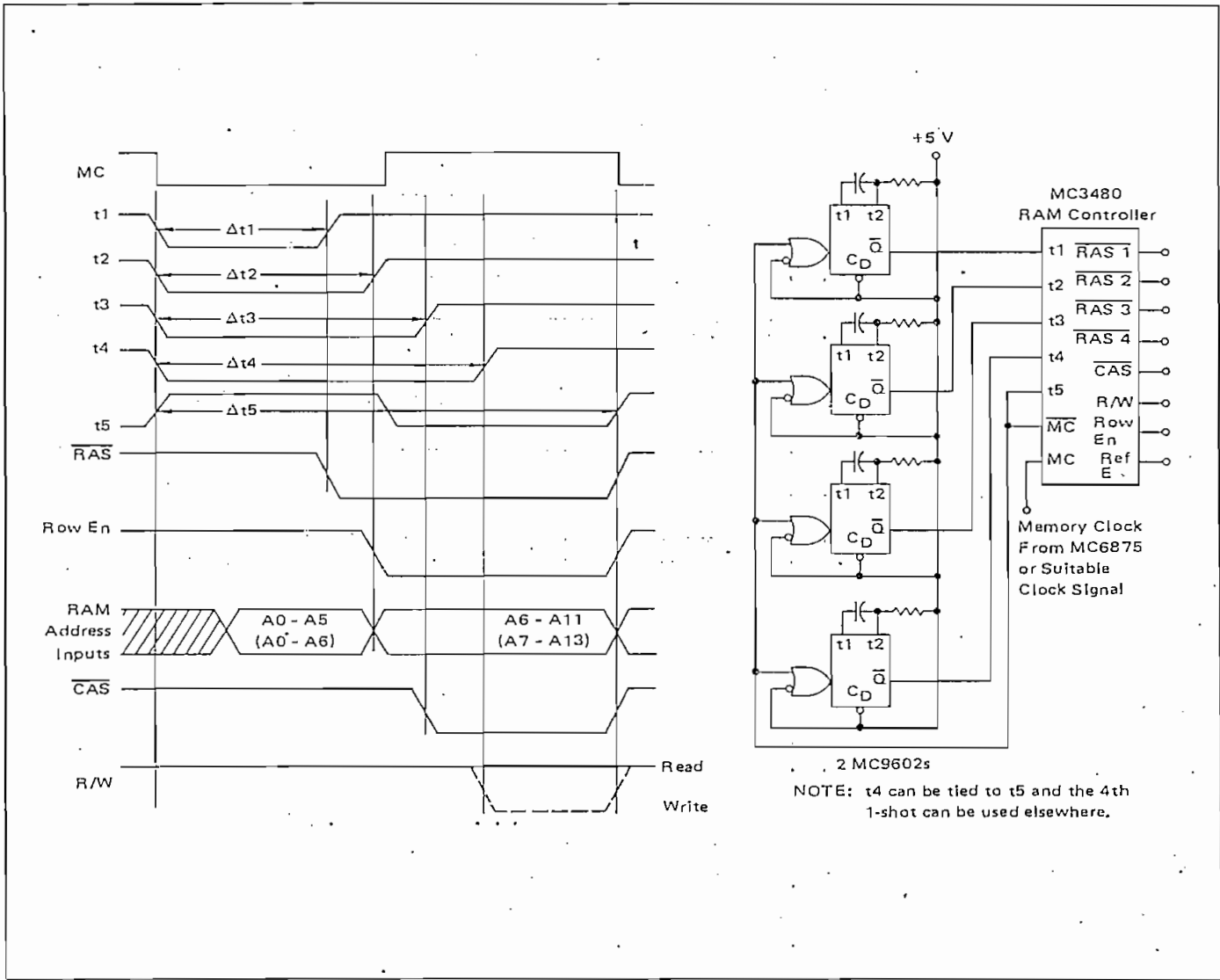


FIGURE 7 — DELAY LINE TIME DELAY METHOD

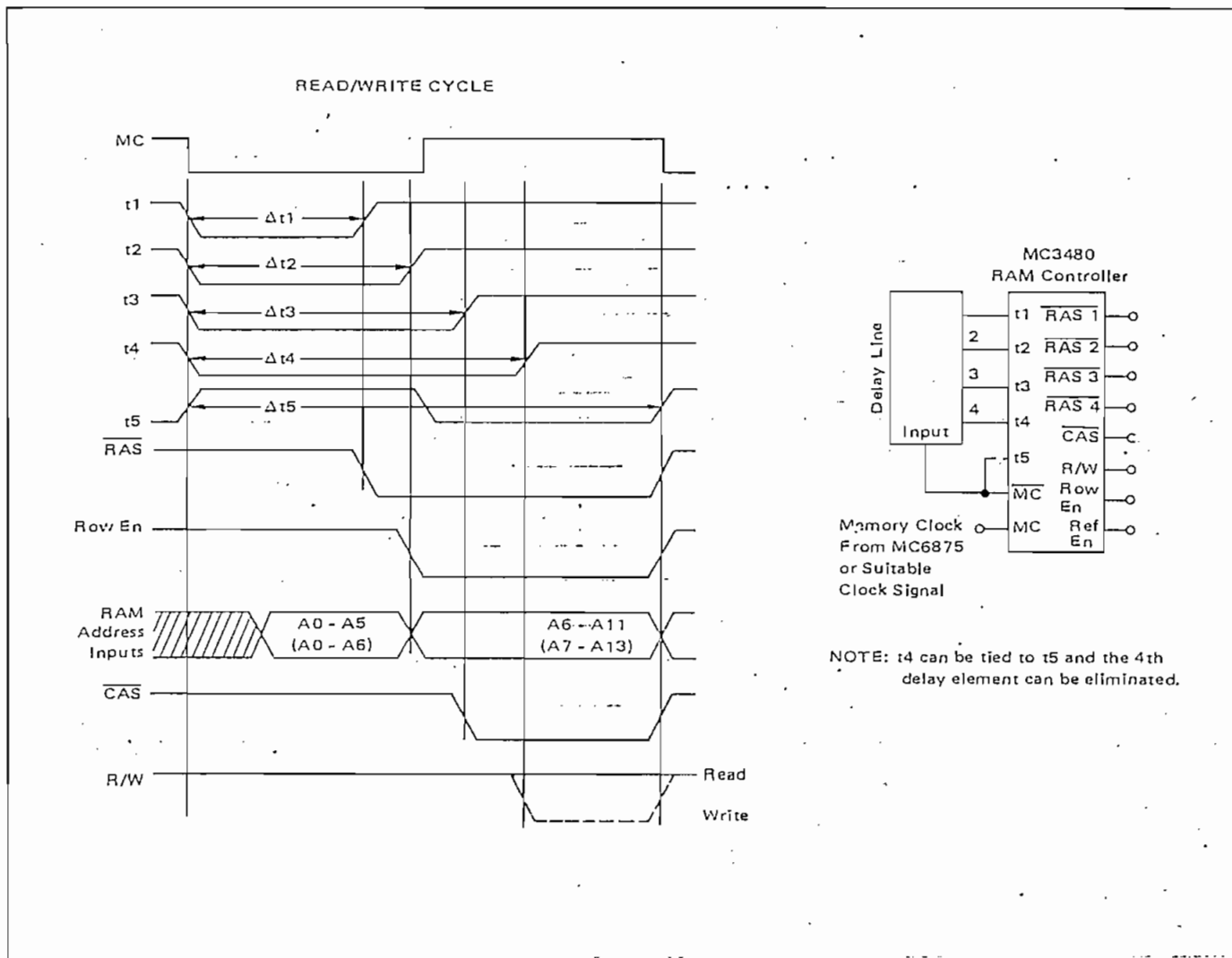
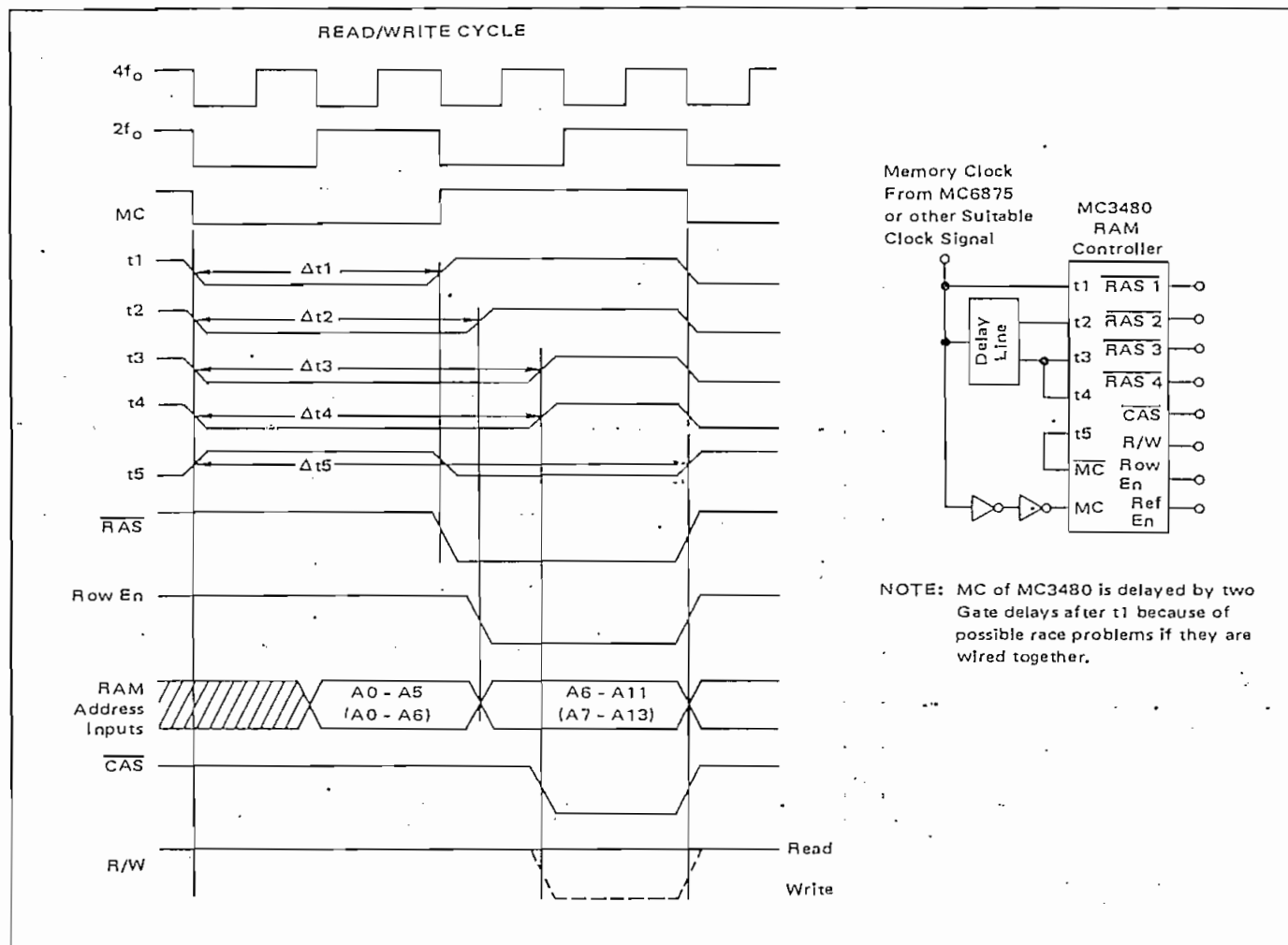


FIGURE 8 — DELAY LINE TIME DELAY (ALTERNATE METHOD)



REFRESH CONSIDERATIONS

The MC3480/MC3232A (MC3242A) memory control system can be used with either cycle steal or transparent refresh methods. Figure 9 shows one transparent technique employing refresh during $\phi 2$ low in an M6800 microprocessor-based system. Using this technique requires that the memory be capable of completing a Read/Write Cycle and a Refresh Cycle sequentially during the M6800 cycle. The minimum cycle time at the time of printing for dynamic multiplexed RAMs is 320 ns, therefore limiting the microprocessor to 1.56 MHz operation. The D flip-flops of Figure 9 produce a trigger at the beginning of both $\phi 1$ and $\phi 2$. For a 1.0 MHz system, the t1-t5 inputs should be adjusted so that the following conditions occur at the specified period after the beginning of a cycle:

- $\overline{\text{RAS}}$ falls at 100 ns (triggered by t1)
- Row In falls at 200 ns (triggered by t2)
- $\overline{\text{CAS}}$, R/W falls at 250 ns (triggered by t3)
- t5 rises at 450 ns.

A delay line could be used to generate t1-t5 in place of

the four monostables. For the 1.0 MHz system, it would require either two 5 tap delay lines with 50 ns per tap or a 10 tap line with 50 ns/tap. For use with a 600 kHz system, a delay line with 5 taps of 150 ns each could be used. For this case:

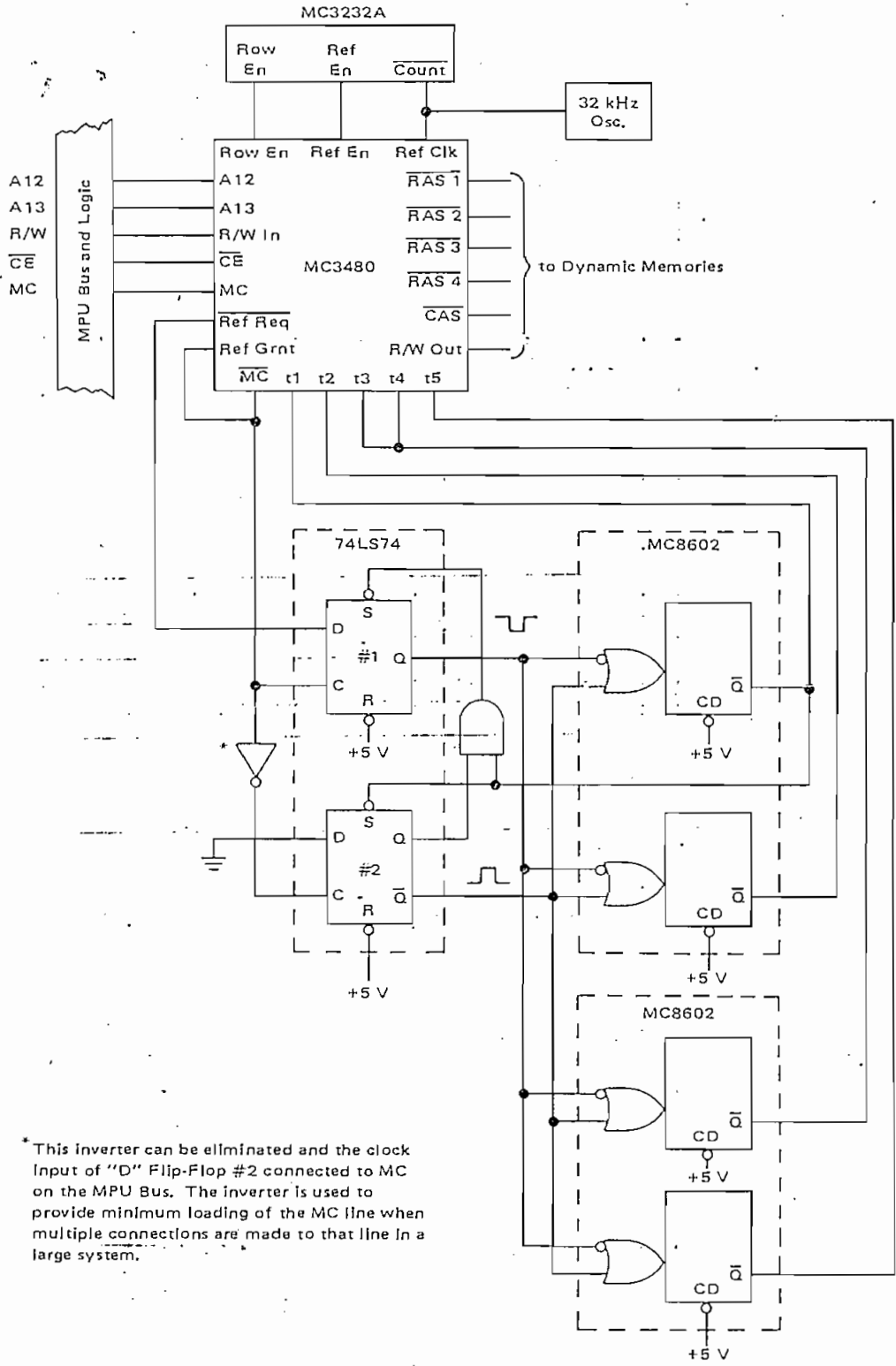
- $\overline{\text{RAS}}$ falls at 150 ns
- Row En falls at 300 ns
- $\overline{\text{CAS}}$, R/W falls at 450 ns
- t5 rises at 750 ns

Figure 10 shows typical refresh oscillator configurations for both 32 kHz (f_{REFmin} for 4K) and 64 kHz (f_{REFmin} for 16K). In the case of transparent refresh, if the designer is not concerned with power consumption, the refresh oscillator may be eliminated and the Ref Clk input connected to the MC input yielding a refresh every $\phi 1$.

For DMA operation combined with cycle stealing refresh, care must be taken not to allow a DMA request during a Refresh Request/Grant period and to hold off a refresh during a DMA operation. See comments under pin descriptions, Pin 19.

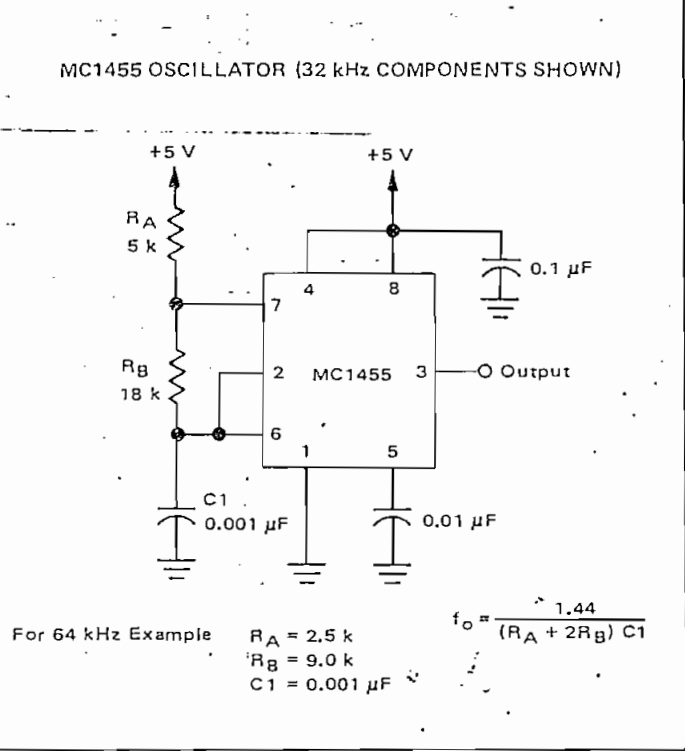
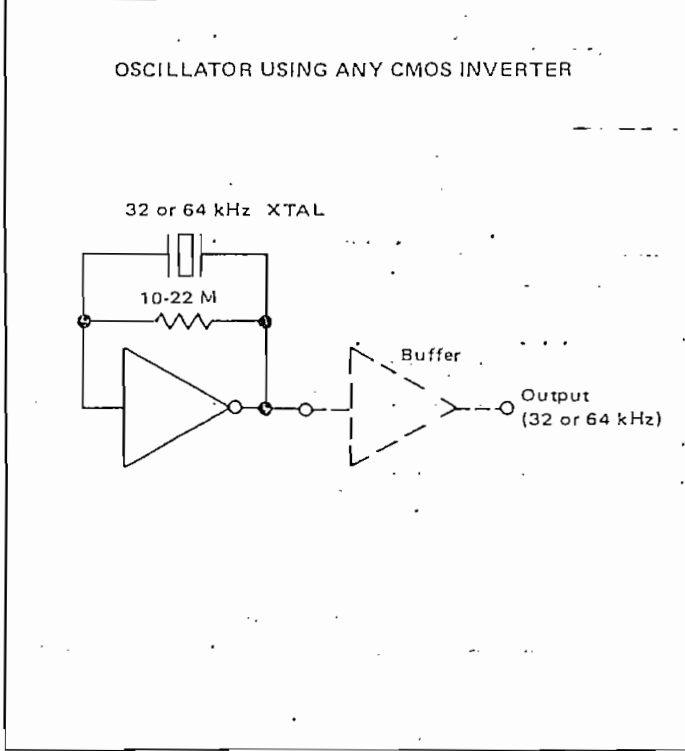
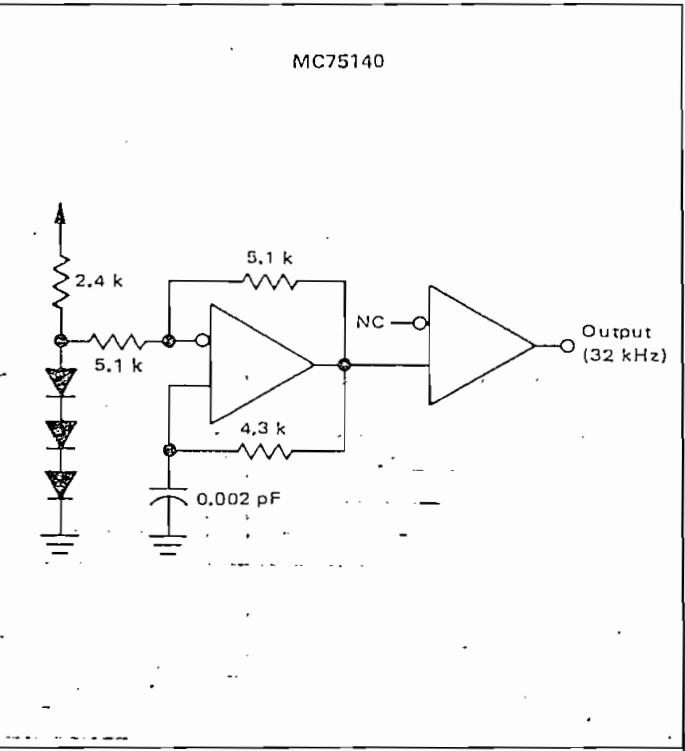
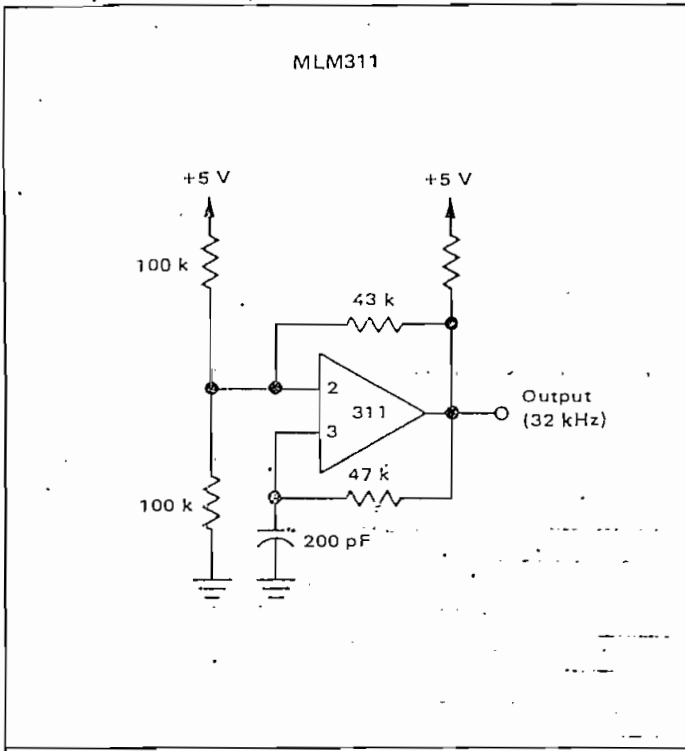


FIGURE 9 — EXAMPLE OF ϕ_2 LOW METHOD OF HIDDEN REFRESH USING MC3480 AND 4K RAMS

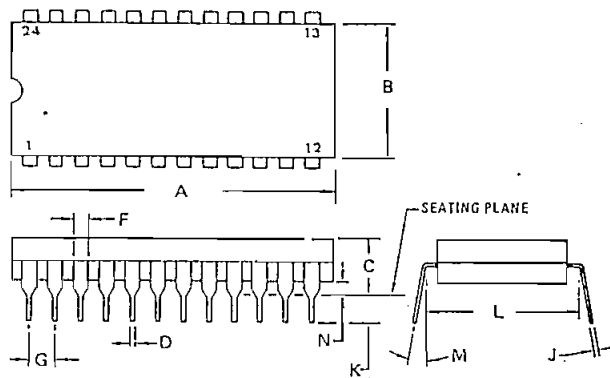


* This inverter can be eliminated and the clock input of "D" Flip-Flop #2 connected to MC on the MPU Bus. The inverter is used to provide minimum loading of the MC line when multiple connections are made to that line in a large system.

FIGURE 10 - SUGGESTED 32 kHz OSCILLATORS



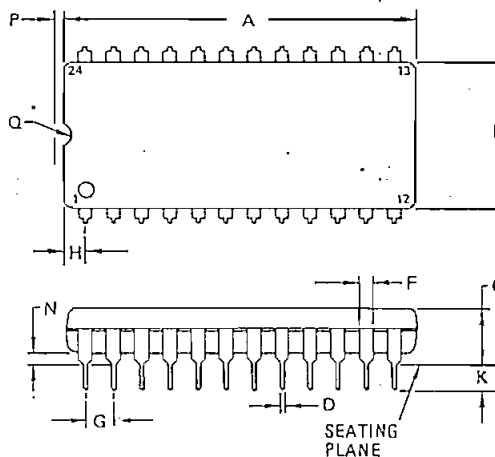
L SUFFIX
CERAMIC PACKAGE
CASE 623-03



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	31.24	32.26	1.230	1.270
B	12.70	13.72	0.500	0.540
C	4.06	5.59	0.160	0.220
D	0.41	0.51	0.016	0.020
F	1.27	1.52	0.050	0.060
G	2.54 BSC		0.100 BSC	
J	0.20	0.30	0.008	0.012
K	2.29	4.06	0.090	0.160
L	15.24 BSC		0.600 BSC	
M	0°	15°	0°	15°
N	0.51	1.27	0.020	0.050

- NOTES:
1. DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.
 2. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION. (WHEN FORMED PARALLEL)

P SUFFIX
PLASTIC PACKAGE
CASE 649-03



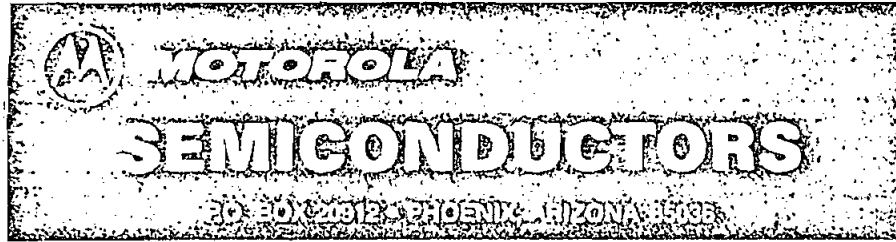
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	31.50	32.13	1.240	1.265
B	13.21	13.72	0.520	0.540
C	4.70	5.21	0.185	0.205
D	0.38	0.51	0.015	0.020
F	1.02	1.52	0.040	0.060
G	2.54 BSC		0.100 BSC	
H	1.65	2.16	0.065	0.085
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	14.99	15.49	0.590	0.610
M	-	10°	-	10°
N	0.51	1.02	0.020	0.040
P	0.13	0.38	0.005	0.015
Q	0.51	0.76	0.020	0.030

- NOTES:
1. LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.
 2. DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL.

Circuit diagrams utilizing Motorola products are included as a means of illustrating typical semiconductor applications; consequently, complete information sufficient for construction purposes is not necessarily given. The information has been carefully checked and

is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of Motorola Inc. or others.





MC3242A

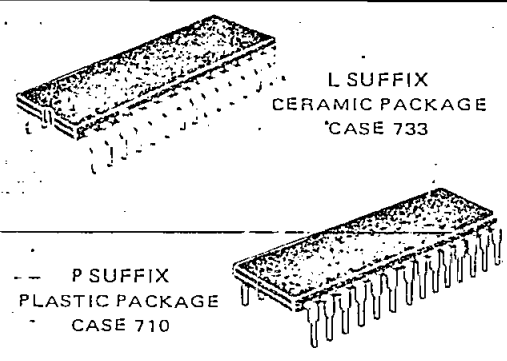
MEMORY ADDRESS MULTIPLEXER FOR 16K RAMS

The Motorola MC3242A is an address multiplexer and refresh counter for 16-pin 16K dynamic RAMs that require a 128-cycle refresh. It multiplexes fourteen system address bits to the seven address pins of the memory device. The MC3242A also contains a 7-bit refresh counter that is clocked externally to generate the 128 sequential addresses required for refresh. The high performance of the MC3242A will enhance the high speed of the N-channel RAMs such as the MCM4116.

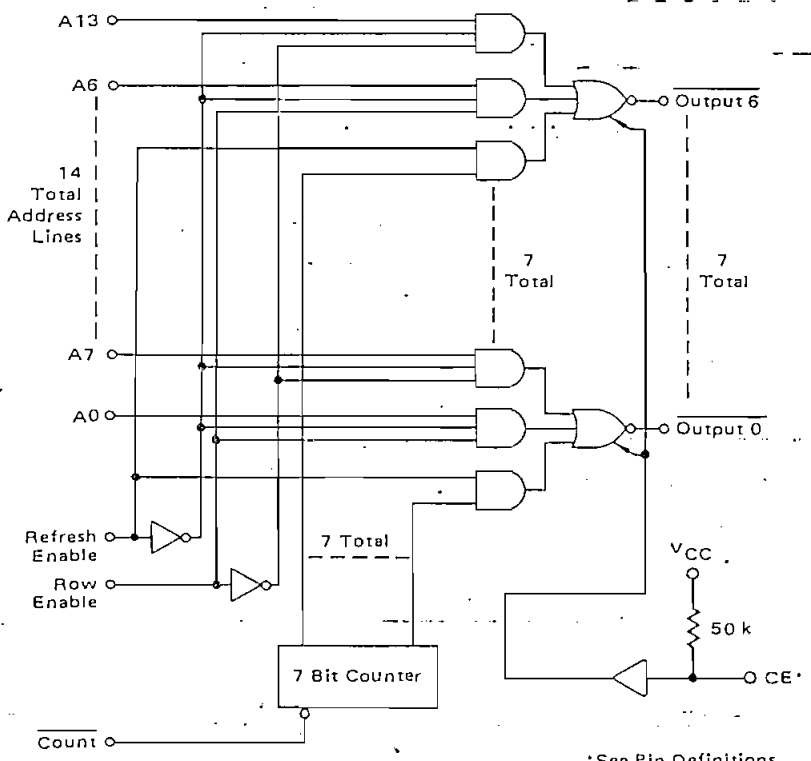
- Simplifies 16-Pin 16K Dynamic Memory Design
- Reduces Package Count
- 7-Bit Binary Counter for 128 Refresh Address
- Multiplexing: Row Address/Column Address/Refresh Address
- High Input Impedance for Minimum Loading of Bus:
 - $I_f = 0.25 \text{ mA Max}$
- Schottky TTL for High Performance Address Input to Output Delay —
 - $t_{AO} = 25 \text{ ns @ } C_L = 250 \text{ pF}$
- Second Source to Intel 3242
 (Detect Zero Function Not Included and Additional Chip Enable Feature Added at Pin 15)

MEMORY ADDRESS MULTIPLEXER AND REFRESH ADDRESS COUNTER

SCHOTTKY
 SILICON MONOLITHIC
 INTEGRATED CIRCUITS



LOGIC DIAGRAM



Count	1	28	VCC
Ref En	2	27	A6
Row En	3	26	A13
N.C.	4	25	A5
A1	5	24	A12
A8	6	23	A4
A2	7	22	A11
A9	8	21	A3
A0	9	20	A10
A7	10	19	O6
O0	11	18	O3
O2	12	17	O4
O1	13	16	O5
Gnd	14	15	CE*

Note: A0 Through A6 Are Row Addresses
 A7 Through A13 Are Column Addresses
 *See Pin Definitions

TRUTH TABLE AND DEFINITIONS

Refresh Enable	Row Enable	Output
H	X	Refresh Address (From Internal Counter)
L	H	Row Address (A0 through A6)
L	L	Column Address (A7 through A13)

Count — Advances Internal Refresh Counter

ORDERING INFORMATION

Device	Temperature Range	Package
MC3242AL	0 to 75°C	Ceramic DIP
MC3242AP	0 to 75°C	Plastic DIP

ABSOLUTE MAXIMUM RATINGS ($T_A = 25^\circ\text{C}$ unless otherwise noted.)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	-0.5 to +7.0	V
Input Voltage	V_I	-0.5 to +7.0	V
Output Voltage	V_O	-0.5 to +7.0	V
Output Current	I_O	100	mA
Operating Ambient Temperature	T_A	0 to +75	$^\circ\text{C}$
Storage Temperature	T_{stg}	-65 to +150	$^\circ\text{C}$
Junction Temperature	T_J		$^\circ\text{C}$
Ceramic Package		+175	
Plastic Package		+150	

"Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum ratings for extended periods may affect reliability.

ELECTRICAL CHARACTERISTICS (Unless otherwise noted, Min/Max values apply with $4.5\text{ V} \leq V_{CC} \leq 5.5\text{ V}$, $0^\circ\text{C} \leq T_A \leq 75^\circ\text{C}$; typical values apply with $V_{CC} = 5.0\text{ V}$, $T_A = 25^\circ\text{C}$.)

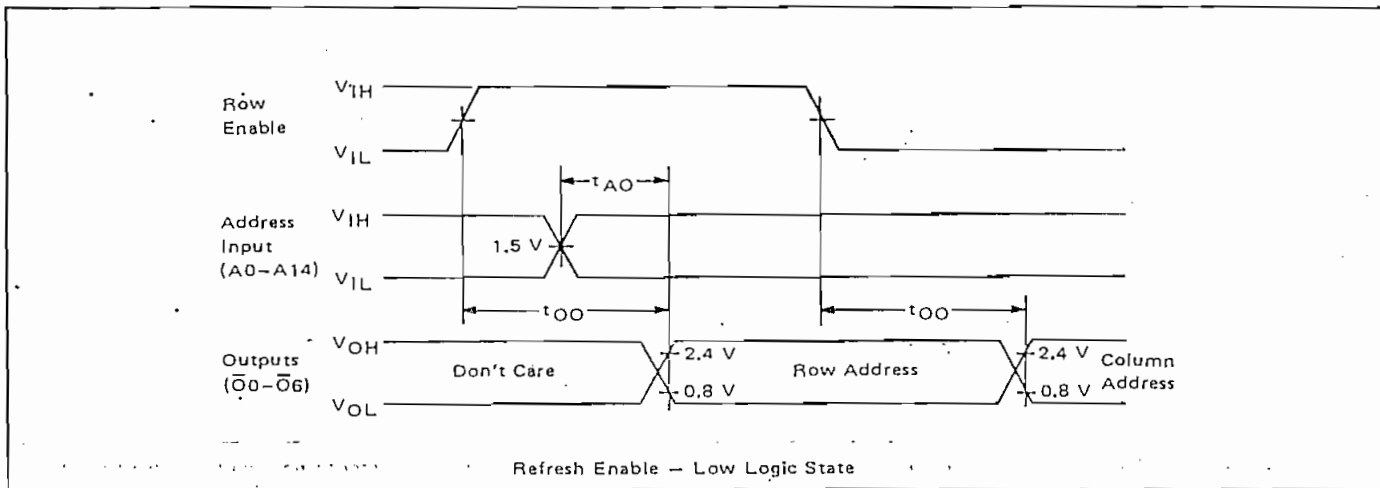
Characteristic	Symbol	Min	Typ	Max	Unit
Input Current, Low Logic State ($V_{IL} = 0.45\text{ V}$)	I_{IL}	—	-0.04	-0.25	mA
Input Current, High Logic State ($V_{IH} = 5.5\text{ V}$)	I_{IH}	—	—	10	μA
Input Voltage, Low Logic State	V_{IL}	—	—	0.8	V
Input Voltage, High Logic State	V_{IH}	2.0	—	—	V
Output Voltage, Low Logic State ($I_{OL} = 5.0\text{ mA}$)	V_{OL}	—	0.25	0.4	V
Output Voltage, High Logic State ($I_{OH} = -1.0\text{ mA}$)	V_{OH}	3.0	4.0	—	V
Input Clamp Voltage ($I_{IK} = -12\text{ mA}$)	V_{IK}	—	-0.8	-1.5	V
Power Supply Current ($V_{CC} = 5.5\text{ V}$)	I_{CC}	—	95	125	mA

SWITCHING CHARACTERISTICS (Unless otherwise noted, Min/Max values apply with $4.5\text{ V} \leq V_{CC} \leq 5.5\text{ V}$, $0^\circ\text{C} \leq T_A \leq 75^\circ\text{C}$; typical values apply with $V_{CC} = 5.0\text{ V}$, $T_A = 25^\circ\text{C}$.)

Characteristic	Symbol	Min	Typ	Max	Unit
Propagation Delay Times					
Address Input to Output (Load = 1 TTL, $C_L = 250\text{ pF}$)	t_{AO}	—	12	25	ns
(Load = 1 TTL, $C_L = 15\text{ pF}$, $V_{CC} = 5.0\text{ V}$, $T_A = 25^\circ\text{C}$)		—	6.0	9.0	
Row Enable to Output (Load = 1 TTL, $C_L = 250\text{ pF}$)	t_{OO}	12	27	41	ns
(Load = 1 TTL, $C_L = 15\text{ pF}$, $V_{CC} = 5.0\text{ V}$, $T_A = 25^\circ\text{C}$)		7	12	27	
Refresh Enable to Output (Load = 1 TTL, $C_L = 250\text{ pF}$)	t_{EO}	12	30	45	ns
(Load = 1 TTL, $C_L = 15\text{ pF}$, $V_{CC} = 5.0\text{ V}$, $T_A = 25^\circ\text{C}$)		7	14	27	
Count Pulse Width	t_{WC}	30	—	—	ns
Counting Frequency	f_C	5.0	10	—	MHz

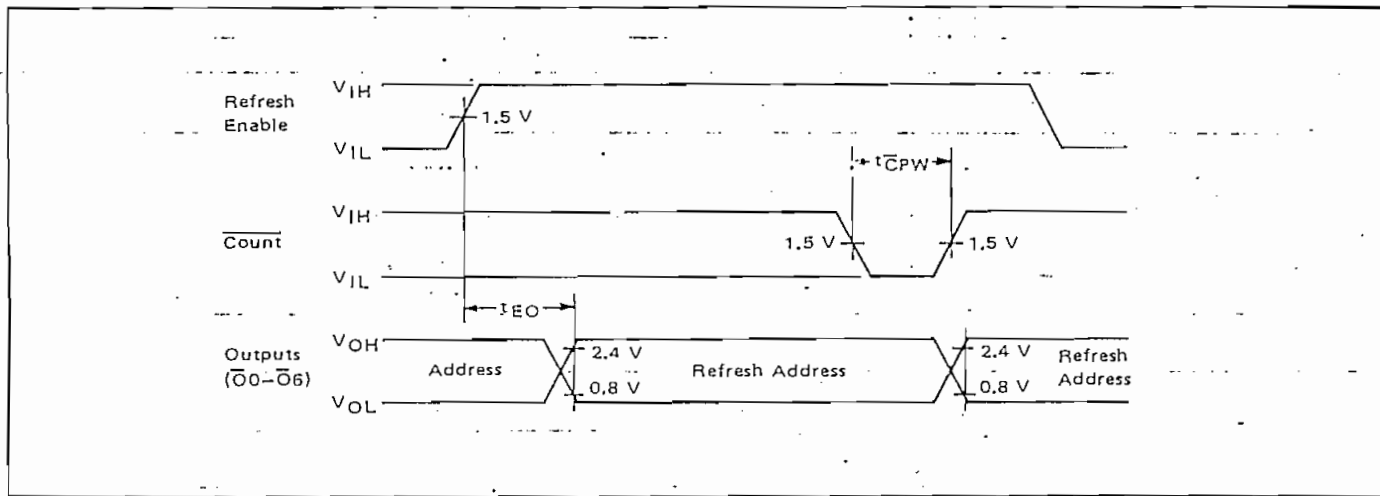


FIGURE 1 - AC WAVEFORMS WITH MCM4116 NORMAL CYCLE



Refresh Enable - Low Logic State

FIGURE 2 - REFRESH CYCLE



TYPICAL CHARACTERISTICS

FIGURE 3 - OUTPUT CURRENT versus OUTPUT LOW VOLTAGE

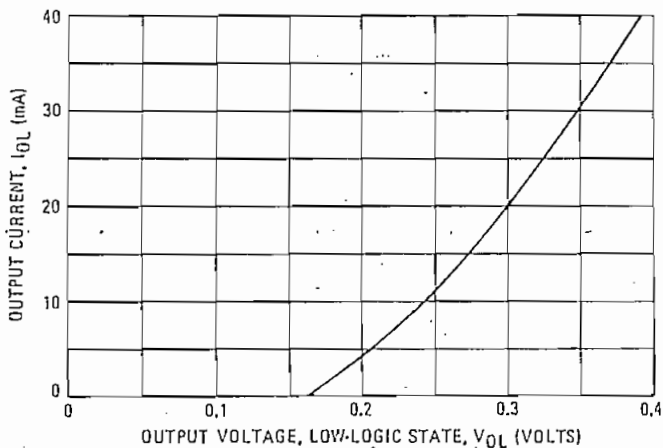
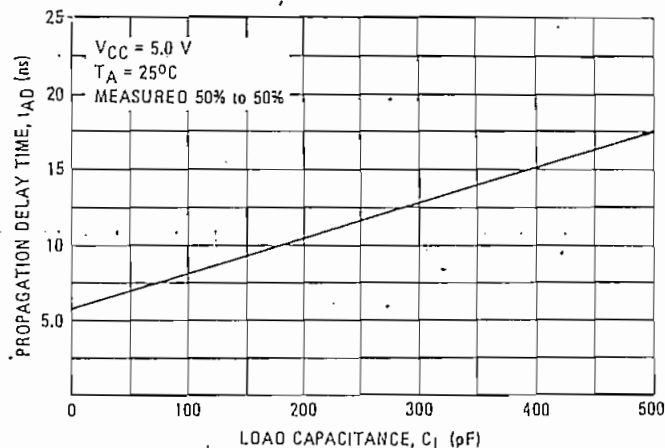


FIGURE 4 - PROPAGATION DELAY versus LOAD CAPACITANCE Row or Column Address to Output



Count Input — Pin 1

Active low input increments internal 6-bit counter by one for each count pulse in.

Refresh Enable Input — Pin 2

Active high input which determines whether the MC3242A is in refresh mode (H) or address enable (L).

A0–A6 Inputs — Pins 9, 5, 7, 21, 23, 27

Row address inputs.

A7–A13 Inputs — Pins 10, 6, 8, 20, 22, 26

Column address inputs.

$\bar{O}0$ – $\bar{O}6$ Outputs — Pins 11, 12, 13, 18, 17, 16, 19

Address outputs to memories: Inverted with respect to address inputs.

Gnd — Pin 14

Power supply ground.

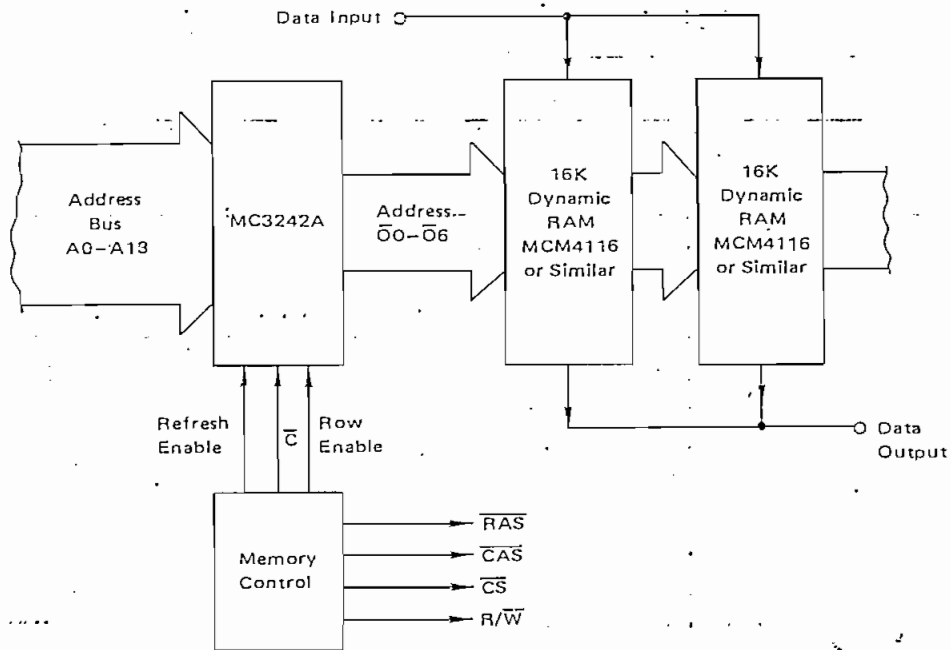
CE Input — Pin 15

Optional use, chip enable control pin. Left open, an internal 50 k Ω pullup resistor keeps this pin high and the MC3242A is a functional replacement for the Intel 3242 (without detect zero function). As an active input, when pulled low, all 3242A outputs go three-state. Regardless of Pin 15 (CE) condition, when power (V_{CC}) is removed, all 3242A outputs go three-state. In addition, the refresh address counter is reset to all 1s so that upon return of supply power, control of refresh addressing can be returned to the MC3242A (by pulling Pin 15 high) at a known address (i.e., all 1s). This option is available tested by consulting factory.

V_{CC} — Pin 28

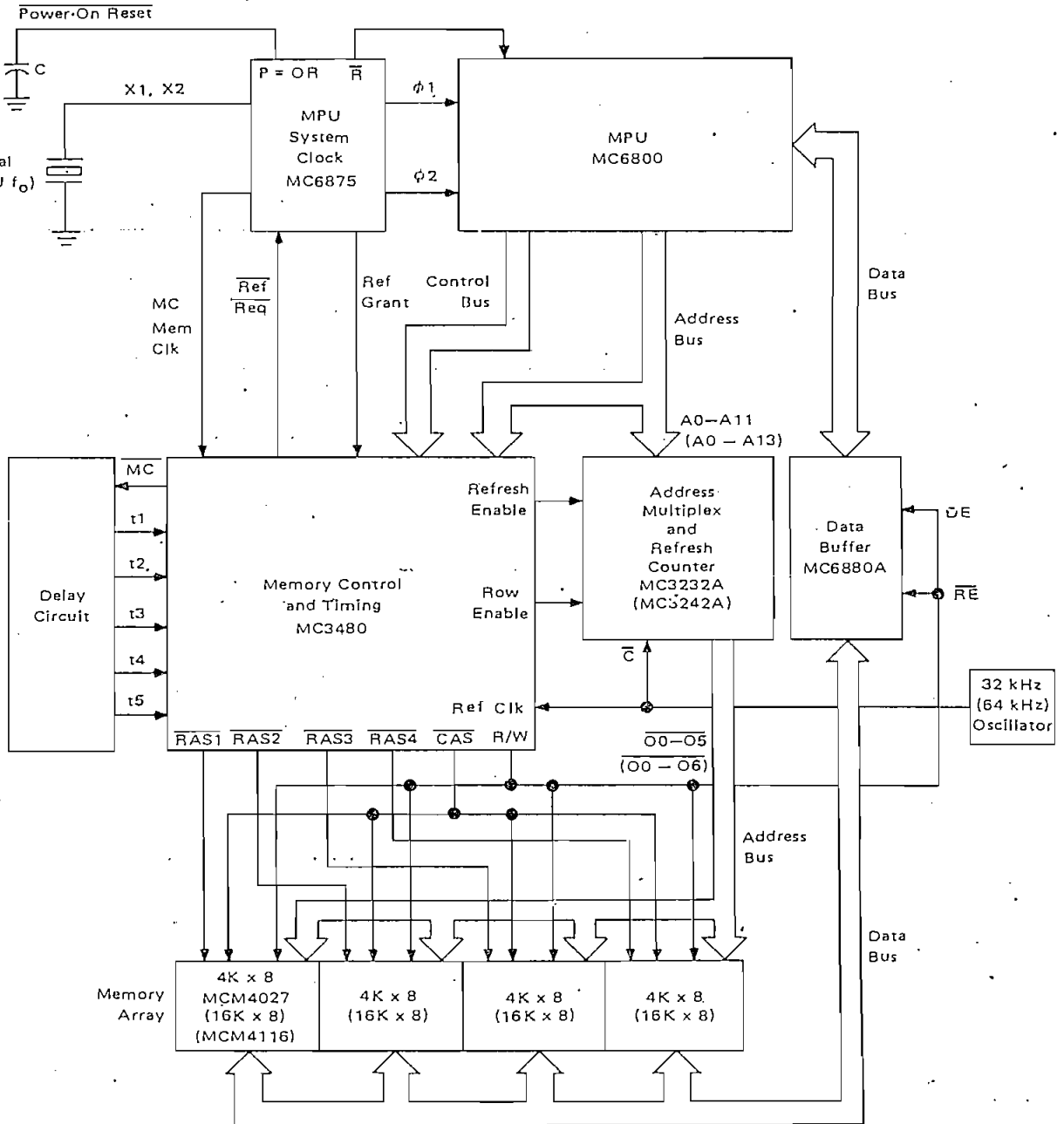
+5 V power supply input. Due to high capacitance drive capability, a 0.1 μ F capacitor should be used to ground along with careful V_{CC} and Gnd Bus layout.

GENERAL 16K DYNAMIC RAM
SIMPLIFIED BLOCK DIAGRAM

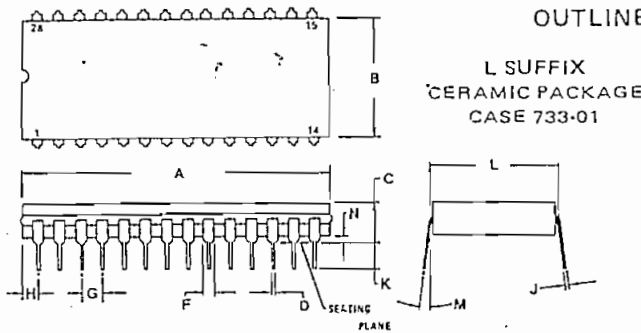


TYPICAL APPLICATION
16K X 8-BIT MEMORY SYSTEM FOR M6800 MPU

Note: Numbers in parentheses indicate part types or values for 16K x 1 RAMs



OUTLINE DIMENSIONS



L SUFFIX
CERAMIC PACKAGE
CASE 733-01

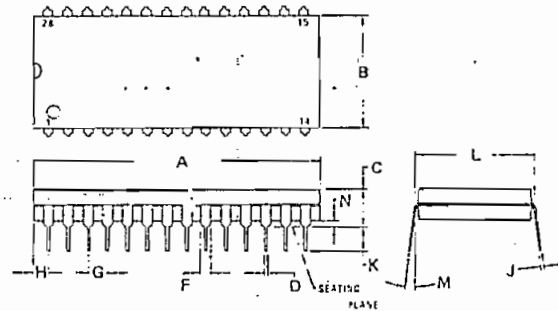
DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.83	37.59	1.450	1.480
B	12.70	13.46	0.500	0.530
C	5.08	5.84	0.200	0.230
D	0.38	0.56	0.015	0.022
F	1.27	1.65	0.050	0.065
G	2.54	BSC	0.100	BSC
H	2.03	2.29	0.080	0.090
J	0.20	0.30	0.008	0.012
K	3.18	4.06	0.125	0.160
L	15.24	BSC	0.600	BSC
M	5°	15°	5°	15°
N	0.51	1.27	0.020	0.050

NOTES:

- LEADS WITHIN 0.25 mm (0.010) DIA OF TRUE POSITION OF SEATING PLANE, AT MAXIMUM MATERIAL CONDITION.
- DIM L TO CENTER OF LEADS WHEN FORMED PARALLEL.
- DIM A AND B INCLUDES MENISCUS.

P SUFFIX
PLASTIC PACKAGE
CASE 710-02

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	36.45	37.21	1.435	1.465
B	13.72	14.22	0.540	0.560
C	3.94	5.08	0.155	0.200
D	0.35	0.56	0.014	0.022
F	1.02	1.52	0.040	0.060
G	2.54	BSC	0.100	BSC
H	1.65	2.16	0.065	0.085
J	0.20	0.38	0.008	0.015
K	2.92	3.43	0.115	0.135
L	15.24	BSC	0.600	BSC
M	0°	15°	0°	15°
N	0.51	1.02	0.020	0.040



THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation:

$$P_{D(T_A)} = \frac{T_{J(max)} - T_A}{R_{\theta JA}(Typ)}$$

Where: $P_{D(T_A)}$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than the sum of the products of the supply voltages and supply currents at the worst case operating condition.

$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section

T_A = Maximum Desired Operating Ambient Temperature

$R_{\theta JA}(Typ)$ = Typical Thermal Resistance Junction to Ambient

Motorola reserves the right to make changes to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others.



MOTOROLA Semiconductor Products Inc.

PO. BOX 30812 • PHOENIX, ARIZONA 85036 • A SUBSIDIARY OF MOTOROLA INC.



MCM4116

16,384-BIT DYNAMIC RANDOM ACCESS MEMORY

The MCM4116 is a 16,384 bit, high speed dynamic Random Access Memory designed for high-performance, low-cost applications in mainframe and buffer memories and peripheral storage. Organized as 16,384 one bit words and fabricated using Motorola's highly reliable N-channel double polysicon technology, this device provides speed, power, and density trade-offs.

By multiplexing row and column address inputs, the MCM4116 requires only seven address lines and permits packaging in Motorola's standard 16-pin dual in-line packages. This packaging technique allows high system density and is compatible with widely available automated test and insertion equipment. Complete address decoding is done on chip with address latches incorporated.

All inputs are TTL compatible, and the output is 3-state TTL compatible. The data output of the MCM4116 is controlled by the column address strobe and remains valid from access time until the column address strobe returns to the high state. This output scheme allows higher degrees of system design flexibility such as common input-output operation and two dimensional memory selection by decoding both row address and column address strobes.

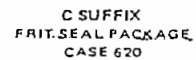
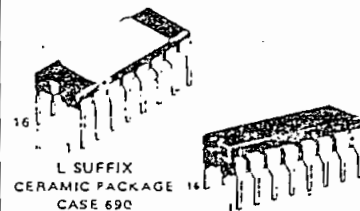
The MCM4116 incorporates a one transistor cell design and dynamic storage techniques, with each of the 128 row addresses requiring a refresh cycle every 2 milliseconds.

- Flexible Timing with Read-Modify-Write, RAS-Only Refresh, and Page-Mode Capability
- Standard 16 Pin Package
- 16,384 x 1 Organization
- 10% Tolerance on All Power Supplies
- All Inputs are Fully TTL Compatible
- Three-State Fully TTL-Compatible Output
- Common I/O Capability When Using "Early Write" Mode
- On-Chip Latches for Addresses and Data In
- Low Power Dissipation - 462 mW Active, 20 mW Standby (Max)
- Fast Access Time Options: 150 ns - MCM4116L-15, C-15
200 ns - MCM4116L-20, C-20
250 ns - MCM4116L-25, C-25
300 ns - MCM4116L-30, C-30
- Easy Upgrade from 16-Pin 4K RAMs
- Pin Compatible with 2117, 2116, 6616, μ PD416, and 4116

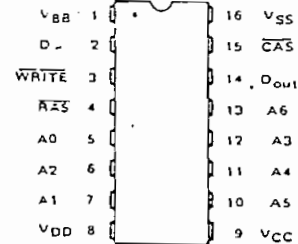
MOS

(N-CHANNEL)

16,384-BIT DYNAMIC RANDOM ACCESS MEMORY



PIN ASSIGNMENT



PIN NAMES

- A₀-A₆ Address Inputs
- CAS Column Address Strobe
- D_{in} Data In
- D_{out} Data Out
- RAS Row Address Strobe
- WRITE Read/Write Input
- V_{BB} Power (-5 V)
- V_{CC} Power (+5 V)
- V_{DD} Power (+12 V)
- V_{SS} Ground

ABSOLUTE MAXIMUM RATINGS (See Note 1)

Rating	Symbol	Value	Unit
Voltage on Any Pin Relative to V _{BB}	V _{in} , V _{out}	-0.5 to +20	Vdc
Operating Temperature Range	T _A	0 to +70	°C
Storage Temperature Range	T _{stg}	-65 to +150	°C
Power Dissipation	P _D	1.0	W
Output Current	I _{out}	50	mA

NOTE 1: Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device reliability.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit.

DC OPERATING CONDITIONS AND CHARACTERISTICS
(Full operating voltage and temperature range unless otherwise noted.)

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Typ	Max	Unit	Notes
Supply Voltage	V _{DD}	10.8	12.0	13.2	Vdc	1
	V _{CC}	4.5	5.0	5.5	Vdc	1, 2
	V _{SS}	0	0	0	Vdc	1
	V _{BB}	-4.5	-5.0	-5.5	Vdc	1
Logic 1 Voltage, $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, $\overline{\text{WRITE}}$	V _{IHC}	2.7	-	7.0	Vdc	1
Logic 1 Voltage, all inputs except $\overline{\text{RAS}}$, $\overline{\text{CAS}}$, $\overline{\text{WRITE}}$	V _{IH}	2.4	-	7.0	Vdc	1
Logic 0 Voltage, all inputs	V _{IL}	-1.0	-	0.8	Vdc	1

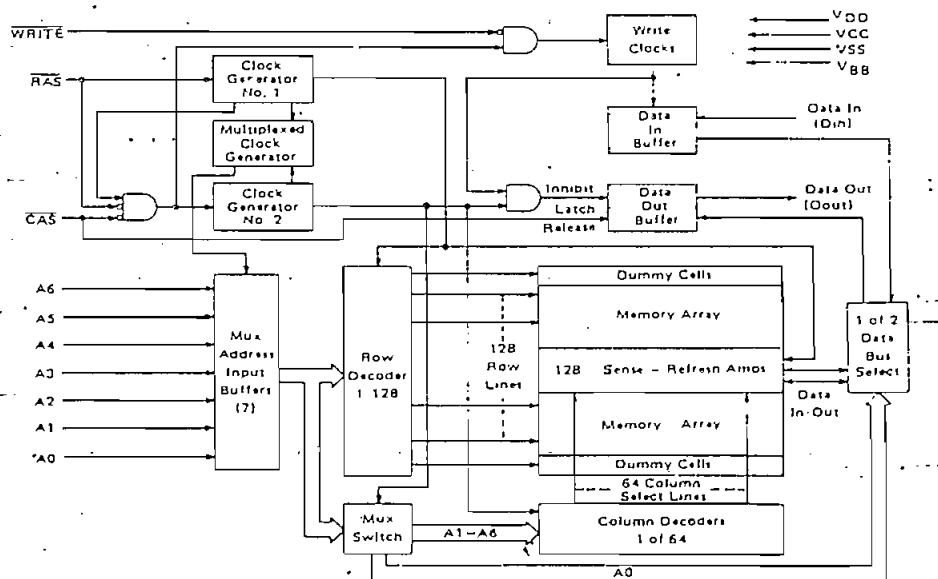
DC CHARACTERISTICS (V_{DD} = 12 V ± 10%, V_{CC} = 5.0 V ± 10%, V_{BB} = -5.0 V ± 10%, V_{SS} = 0 V, T_A = 0 to 70°C)

Characteristic	Symbol	Min	Max	Units	Notes
Average V _{DD} Power Supply Current	I _{DD1}	-	35	mA	4
V _{CC} Power Supply Current	I _{CC}	-	-	mA	5
Average V _{BB} Power Supply Current	I _{BB1,3}	-	200	μA	
Standby V _{BB} Power Supply Current	I _{BB2}	-	100	μA	
Standby V _{DD} Power Supply Current	I _{DD2}	-	1.5	mA	6
Average V _{DD} Power Supply Current during "RAS only" cycles	I _{DD3}	-	27	mA	4
Input Leakage Current (any input)	I _{IHL}	-	10	μA	
Output Leakage Current	I _{OL1}	-	10	μA	6
Output Logic 1 Voltage @ I _{out} = 5 mA	V _{OH}	2.4	-	Vdc	2
Output Logic 0 Voltage @ I _{out} = 4.2 mA	V _{OL}	-	0.4	Vdc	

NOTES:

- All voltages referenced to V_{SS}. V_{BB} must be applied before and removed after other supply voltages.
- Output voltage will swing from V_{SS} to V_{CC} under open circuit conditions. For purposes of maintaining data in power down mode, V_{CC} may be reduced to V_{SS} without affecting refresh operations. V_{OH} minimum specification is not guaranteed in this mode.
- Several cycles are required after power up before proper device operation is achieved. Any B cycles which perform refresh are appropriate.
- Current is proportional to cycle rate; maximum current is measured at the fastest cycle rate.
- I_{CC} depends upon output loading. The V_{CC} supply is connected to the output buffer only.
- Output is disabled (open circuit) and $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ are both at a logic 1.
- 0 V < V_{out} < +5.5 V.
- Capacitance measured with a Boonton Meter or effective capacitance calculated from the equation: C = $\frac{Q}{\Delta V}$.

BLOCK DIAGRAM



MCM4116

AC OPERATING CONDITIONS AND CHARACTERISTICS (See Notes 3, 9, 14)
(Read, Write, and Read-Modify-Write Cycles)

RECOMMENDED AC OPERATING CONDITIONS

(V_{DD} = 12 V ± 10%, V_{CC} = 5.0 V ± 10%, V_{BB} = -5.0 V ± 10%, V_{SS} = 0 V, T_A = 0 to 70°C.)

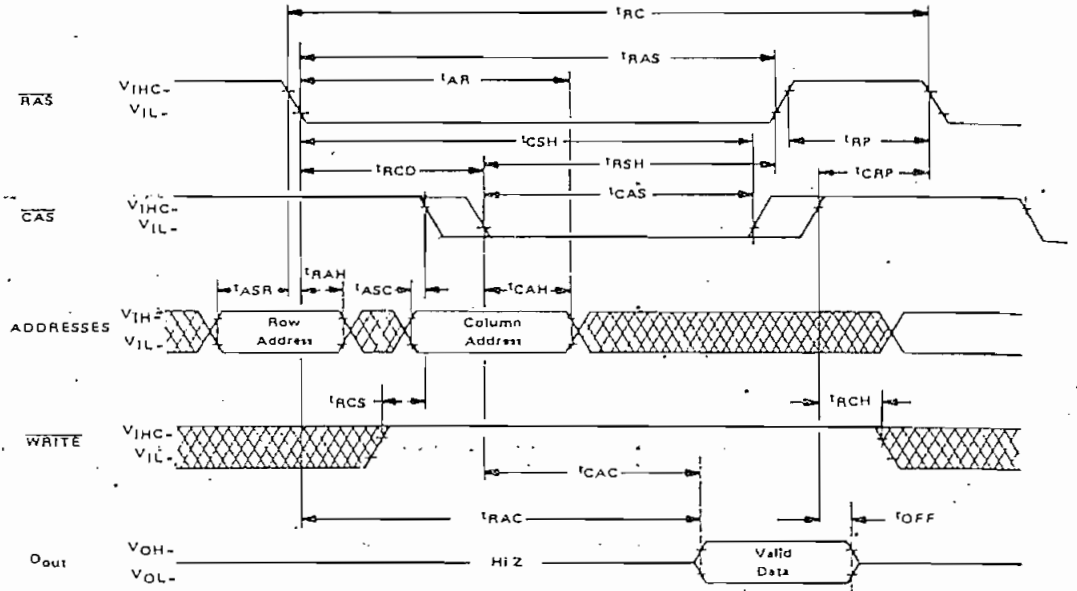
Parameter	Symbol	MCM4116-15		MCM4116-20		MCM4116-25		MCM4116-30		Units	Notes
		Min	Max	Min	Max	Min	Max	Min	Max		
Random Read or Write Cycle Time	t _{RC}	375	-	375	-	410	-	480	-	ns	
Read Write Cycle Time	t _{RWC}	375	-	375	-	515	-	660	-	ns	
Access Time from Row Address Strobe	t _{RAC}	-	150	-	200	-	250	-	300	ns	10, 12
Access Time from Column Address Strobe	t _{CAC}	0	90	-	135	-	165	-	200	ns	11, 12
Output Buffer and Turn-off Delay	t _{OFF}	0	40	0	50	0	60	0	60	ns	
Row Address Strobe Precharge Time	t _{RP}	100	-	120	-	150	-	180	-	ns	
Row Address Strobe Pulse Width	t _{RAS}	150	10,000	200	10,000	250	10,000	300	10,000	ns	
Column Address Strobe Pulse Width	t _{CAS}	90	10,000	135	10,000	165	10,000	200	10,000	ns	
Row to Column Strobe Lead Time	t _{RCD}	20	60	25	65	35	85	60	100	ns	13
Row Address Setup Time	t _{ASR}	0	-	0	-	0	-	0	-	ns	
Row Address Hold Time	t _{RAH}	20	-	25	-	35	-	60	-	ns	
Column Address Setup Time	t _{ASC}	-10	-	-10	-	-10	-	-10	-	ns	
Column Address Hold Time	t _{CAH}	45	-	55	-	75	-	100	-	ns	
Column Address Hold Time Referenced to RAS	t _{AR}	105	-	120	-	160	-	200	-	ns	
Transition Time (Rise and Fall)	t _T	3.0	35	3.0	50	3.0	50	3.0	50	ns	14
Read Command Setup Time	t _{RCS}	0	-	0	-	0	-	0	-	ns	
Read Command Hold Time	t _{RCH}	0	-	0	-	0	-	0	-	ns	
Write Command Hold Time	t _{WCH}	45	-	55	-	75	-	100	-	ns	
Write Command Hold Time Referenced to RAS	t _{WCR}	105	-	120	-	160	-	200	-	ns	
Write Command Pulse Width	t _{WP}	45	-	55	-	75	-	100	-	ns	
Write Command to Row Strobe Lead Time	t _{RWL}	60	-	80	-	100	-	180	-	ns	
Write Command to Column Strobe Lead Time	t _{CWL}	60	-	80	-	100	-	180	-	ns	
Data in Setup Time	t _{DS}	0	-	0	-	0	-	0	-	ns	15
Data in Hold Time	t _{DH}	45	-	55	-	75	-	100	-	ns	15
Data in Hold Time Referenced to RAS	t _{DHR}	105	-	120	-	160	-	200	-	ns	
Column to Row Strobe Precharge Time	t _{CRP}	-20	-	-20	-	-20	-	-20	-	ns	
RAS Hold Time	t _{RSH}	100	-	135	-	165	-	200	-	ns	
Refresh Period	t _{RFSH}	-	2.0	-	2.0	-	2.0	-	2.0	ms	
WRITE Command Setup Time	t _{WCS}	-20	-	-20	-	-20	-	-20	-	ns	
CAS to WRITE Delay	t _{CWD}	70	-	95	-	125	-	180	-	ns	16
RAS to WRITE Delay	t _{RWD}	120	-	160	-	210	-	280	-	ns	16
CAS Precharge Time (Page mode cycle only)	t _{CP}	60	-	80	-	100	-	100	-	ns	
Page Mode Cycle Time	t _{PC}	170	-	225	-	275	-	325	-	ns	
CAS Hold Time	t _{CSH}	150	-	200	-	250	-	300	-	ns	

Parameter	Symbol	Typ	Max	Units	Notes
Input Capacitance (A0-A5), D _{in}	C _{I1}	4.0	5.0	pF	9
Input Capacitance RAS, CAS, WRITE	C _{I2}	8.0	10	pF	9
Output Capacitance (O _{out})	C _O	5.0	7.0	pF	7, 9

NOTES: (continued)

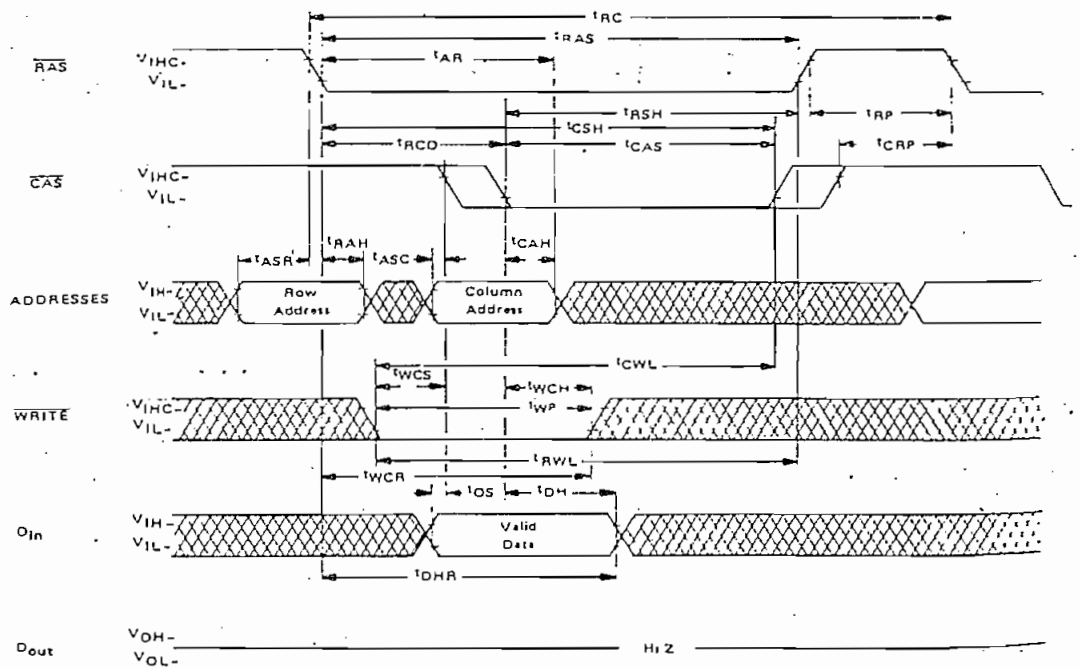
- AC measurements assume t_T = 5.0 ns.
- Assumes that t_{RCD} + t_T < t_{RCD} (max).
- Assumes that t_{RCD} + t_T > t_{RCD} (max).
- Measured with a load circuit equivalent to 2 TTL loads and 100 pF.
- Operation within the t_{RCD} (max) limit ensures that t_{RAC} (max) can be met. t_{RCD} (max) is specified as a reference point only, if t_{RCD} is greater than the specified t_{RCD} (max) limit, then access time is controlled exclusively by t_{CAC}.
- V_{IHC} (min) or V_{IH} (min) and V_{IL} (max) are reference levels for measuring timing of input signals. Also, transition times are measured between V_{IHC} or V_{IH} and V_{IL}.
- These parameters are referenced to CAS leading edge in random write cycles and to WRITE leading edge in delayed write or read-modify-write cycles.
- t_{WCS}, t_{CWD} and t_{RWD} are not restrictive operating parameters. They are included in the data sheet as electrical characteristics only. If t_{WCS} > t_{WCS} (min), the cycle is an early write cycle and the data out pin will remain open circuit (high impedance) throughout the entire cycle; if t_{CWD} > t_{CWD} (min) and t_{RWD} > t_{RWD} (min), the cycle is a read-write cycle and the data out will contain data read from the selected cell. If neither of the above sets of conditions is satisfied the condition of the data out (at access time) is indeterminate.

READ CYCLE TIMING



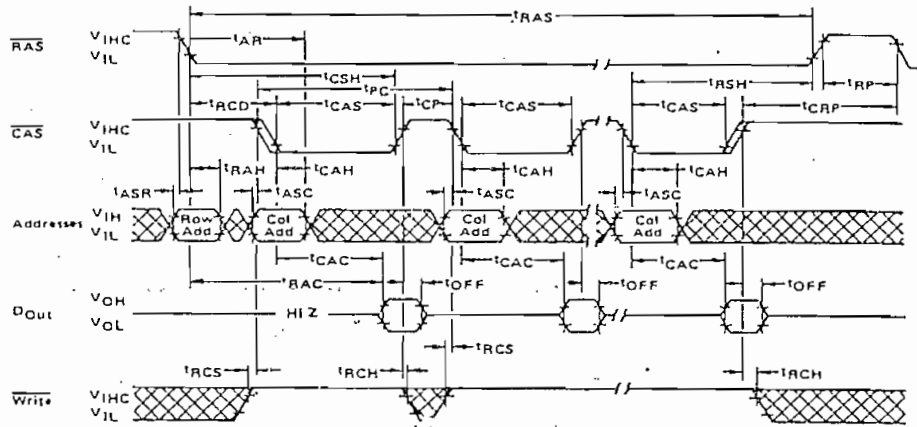
5

WRITE CYCLE TIMING

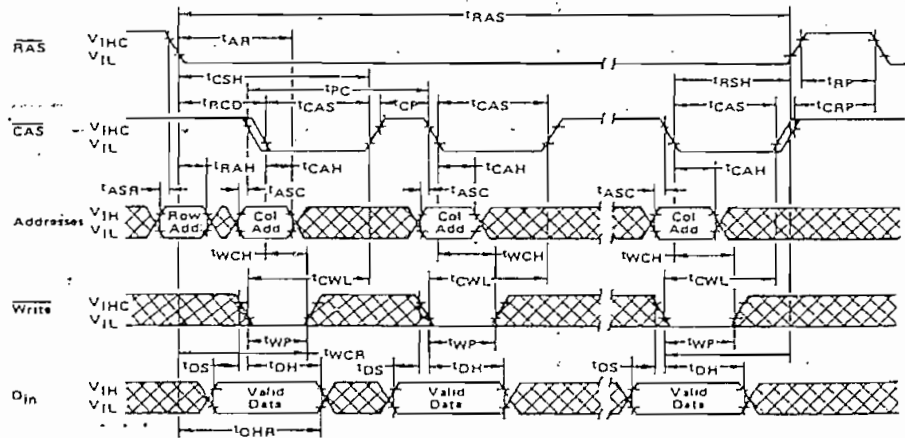


PAGE MODE READ CYCLE

5

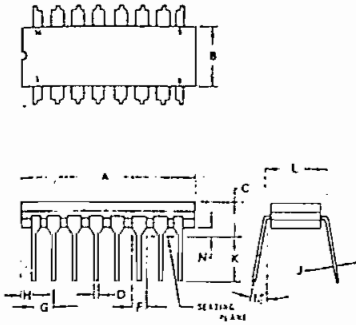


PAGE MODE WRITE CYCLE



OUTLINE DRAWINGS

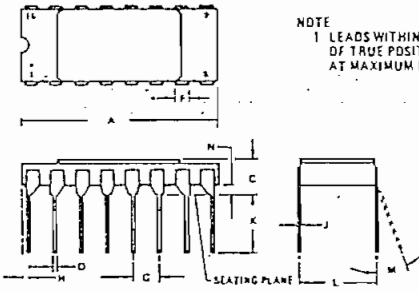
CASE 820-06



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	19.05	19.94	0.750	0.785
B	6.27	7.11	0.245	0.280
C	3.94	5.08	0.155	0.200
D	0.38	0.51	0.015	0.020
E	0.89	1.52	0.035	0.060
F	2.54 BSC		0.100 BSC	
G	0.38	1.52	0.015	0.060
H	0.70	0.30	0.008	0.012
I	3.18	5.08	0.125	0.200
J	7.37	8.13	0.290	0.320
K	-	15°	-	15°
L	0.51	1.27	0.020	0.050

- LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
- PACKAGE INDEX NOTCH IN LEAD NOTCH IN CERAMIC OR INK DOT
- DIM "A" AND "B" (620 06) DO NOT INCLUDE GLASS RUN OUT
- DIM "L" TO INSIDE OF LEADS (MEASURED 0.51 mm (0.020) BELOW BODY)

CASE 690-09

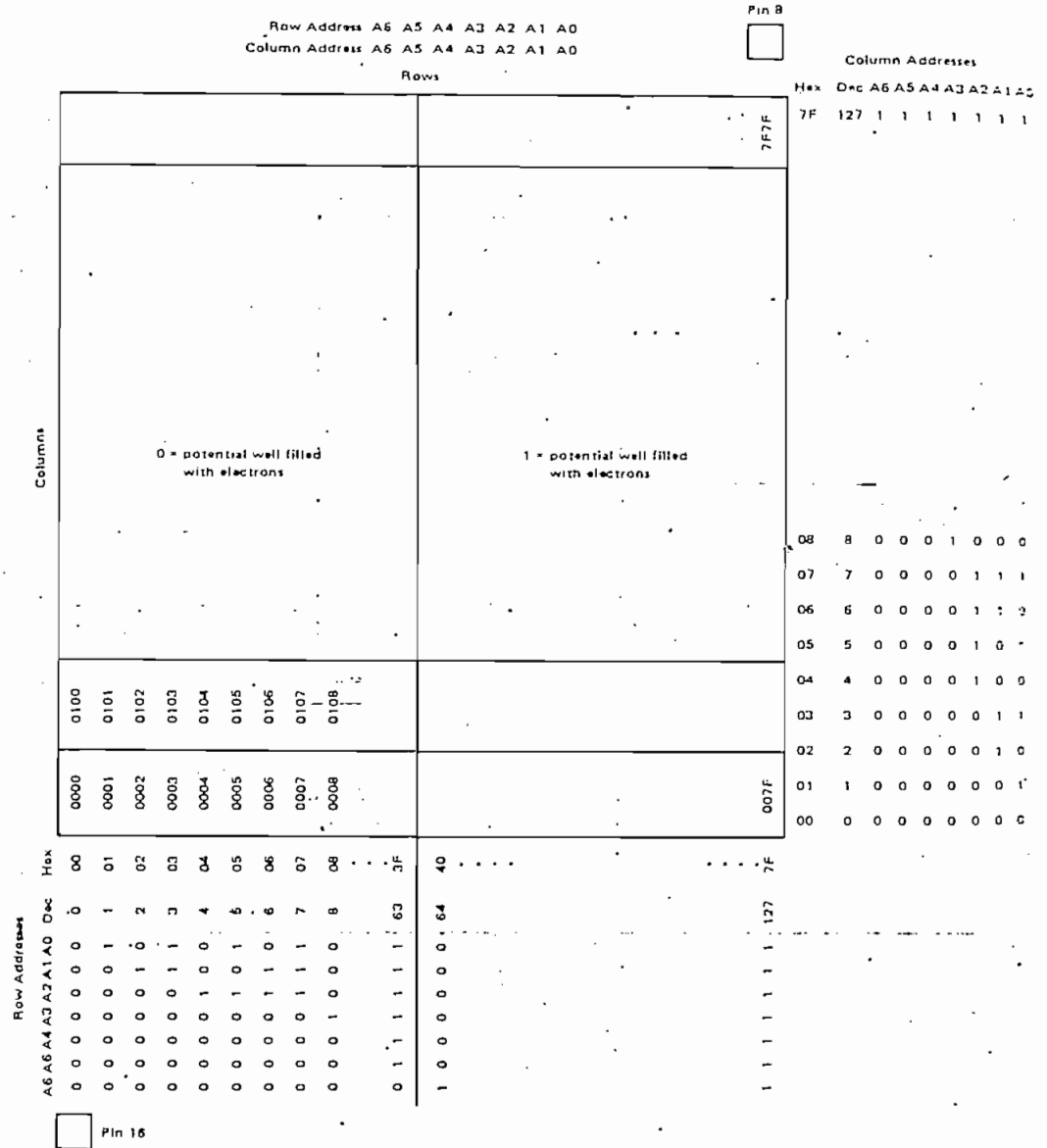


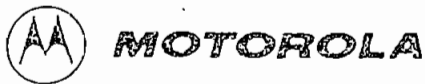
NOTE
1 LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	20.07	20.57	0.790	0.810
B	2.67	3.94	0.105	0.155
C	0.38	0.51	0.015	0.021
D	1.27	1.52	0.048	0.060
E	2.54 BSC		0.100 BSC	
F	0.76	1.27	0.030	0.050
G	0.70	0.31	0.008	0.012
H	3.05	4.83	0.120	0.190
I	7.62 BSC		0.300 BSC	
J	-	10°	-	10°
K	0.64	1.52	0.025	0.060

MCM4116 BIT ADDRESS MAP

5





QUAD THREE-STATE BUS TRANSCEIVER

This quad three state bus transceiver features both excellent MOS or MPU compatibility, due to its high impedance PNP transistor input, and high speed operation made possible by the use of Schottky diode clamping. Both the -48 mA driver and -20 mA receiver outputs are short-circuit protected and employ three-state enabling inputs.

The device is useful as a bus extender in systems employing the M6800 family or other comparable MPU devices. The maximum input current of 200 μ A at any of the device input pins assures proper operation despite the limited drive capability of the MPU chip. The inputs are also protected with Schottky-barrier diode clamps to suppress excessive undershoot voltages.

The MC8T26A is identical to the NE8T26A and it operates from a single +5 V supply.

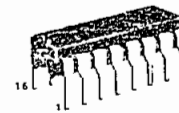
- High Impedance Inputs
- Single Power Supply
- High Speed Schottky Technology
- Three-State Drivers and Receivers
- Compatible with M6800 Family Microprocessor

MC6880A MC8T26A

This device may be ordered under either of the above type numbers.

QUAD THREE-STATE BUS TRANSCEIVER

MONOLITHIC SCHOTTKY
INTEGRATED CIRCUITS

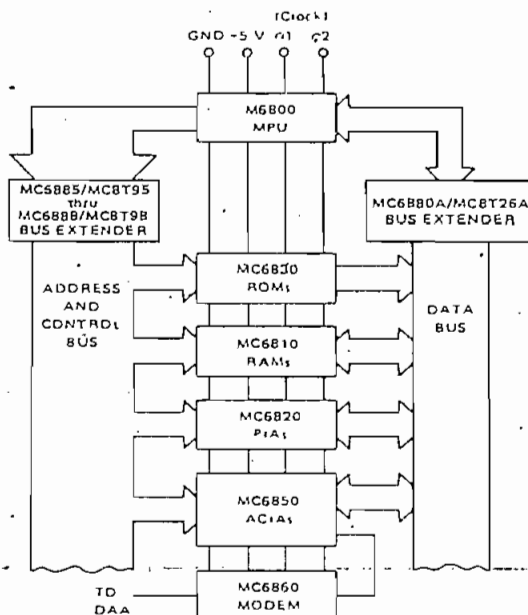


L SUFFIX
CERAMIC PACKAGE
CASE 620

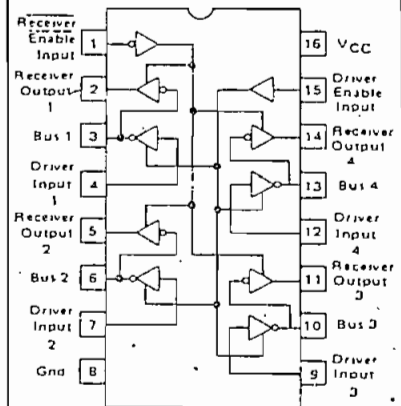


P SUFFIX
PLASTIC PACKAGE
CASE 648

MICROPROCESSOR BUS EXTENDER APPLICATION



PIN CONNECTIONS - MC6880A MC8T26A



ORDERING INFORMATION

Device	Alternate	Temperature Range	Package
MC6880AL	MC8T26AL	0 to 75°C	Ceramic DIP
MC6880AP	MC8T26AP	0 to 75°C	Plastic DIP



MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
Power Supply Voltage	V _{CC}	8.0	Vdc
Input Voltage	V _I	5.5	Vdc
Junction Temperature	T _J		°C
Ceramic Package		175	
Plastic Package		150	
Operating Ambient Temperature Range	T _A	0 to +75	°C
Storage Temperature Range	T _{stg}	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (4.75 V < V_{CC} < 5.25 V and 0°C < T_A < 75°C unless otherwise noted)

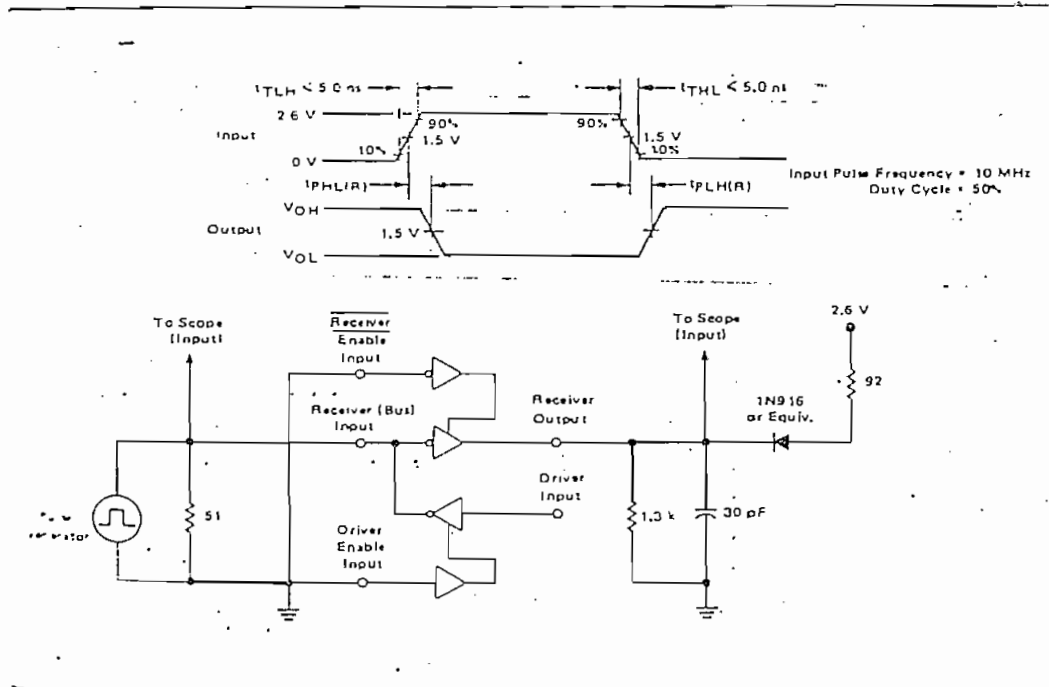
Characteristic	Symbol	Min	Typ	Max	Unit
Input Current - Low Logic State (Receiver Enable Input, V _{IL(RE)} = 0.4 V) (Driver Enable Input, V _{IL(DE)} = 0.4 V) (Driver Input, V _{IL(D)} = 0.4 V) (Bus (Receiver) Input, V _{IL(B)} = 0.4 V)	I _{IL(RE)} I _{IL(DE)} I _{IL(D)} I _{IL(B)}	-	-	700 700 -700 -200	μA
Input Disabled Current - Low Logic State (Driver Input, V _{IL(D)} = 0.4 V)	I _{IL(D)} DIS	-	-	-25	μA
Input Current - High Logic State (Receiver Enable Input, V _{IH(RE)} = 5.25 V) (Driver Enable Input, V _{IH(DE)} = 5.25 V) (Driver Input, V _{IH(D)} = 5.25 V) (Receiver Input, V _{IH(B)} = 5.25 V)	I _{IH(RE)} I _{IH(DE)} I _{IH(D)} I _{IH(B)}	-	-	25 25 25 100	μA
Input Voltage - Low Logic State (Receiver Enable Input) (Driver Enable Input) (Driver Input) (Receiver Input)	V _{IL(RE)} V _{IL(DE)} V _{IL(D)} V _{IL(B)}	-	-	0.55 0.45 0.65 0.85	V
Input Voltage - High Logic State (Receiver Enable Input) (Driver Enable Input) (Driver Input) (Receiver Input)	V _{IH(RE)} V _{IH(DE)} V _{IH(D)} V _{IH(B)}	2.0 2.0 2.0 2.0	-	-	V
Output Voltage - Low Logic State (Bus (Driver) Output, I _{OL(B)} = 48 mA) (Receiver Output, I _{OL(R)} = 20 mA)	V _{OL(B)} V _{OL(R)}	-	-	0.5 0.5	V
Output Voltage - High Logic State (Bus (Driver) Output, I _{OH(B)} = 10 mA) (Receiver Output, I _{OH(R)} = 2.0 mA) (Receiver Output, I _{OH(R)} = 100 μA, V _{CC} = 5.0 V)	V _{OH(B)} V _{OH(R)}	2.4 2.4 3.5	3.1 3.1	-	V
Output Disabled Leakage Current - High Logic State (Bus (Driver) Output, V _{OH(B)} = 2.4 V) (Receiver Output, V _{OH(R)} = 2.4 V)	I _{OH(B)} I _{OH(R)}	-	-	100 100	μA
Output Disabled Leakage Current - Low Logic State (Bus Output, V _{OL(B)} = 0.5 V) (Receiver Output, V _{OL(R)} = 0.5 V)	I _{OLL(B)} I _{OLL(R)}	-	-	100 -100	μA
Input Clamp Voltage (Driver Enable Input I _{IC(DE)} = 12 mA) (Receiver Enable Input I _{IC(RE)} = +12 mA) (Driver Input I _{IC(D)} = -12 mA)	V _{IC(DE)} V _{IC(RE)} V _{IC(D)}	-	-	-1.0 -1.0 -1.0	V
Output Short Circuit Current, V _{CC} = 5.25 V (1) (Bus (Driver) Output) (Receiver Output)	I _{OS(B)} I _{OS(R)}	-50 -30	-	150 75	mA
Power Supply Current (V _{CC} = 5.25 V)	I _{CC}	-	-	87	μA

(1) Only one output may be short-circuited at a time.

SWITCHING CHARACTERISTICS (Unless otherwise noted, specifications apply at $T_A = 25^{\circ}\text{C}$ and $V_{CC} = 5.0\text{ V}$)

Characteristic	Symbol	Figure	Min	Max	Unit
Propagation Delay Time from Receiver (Bus) Input to High Logic State Receiver Output	$t_{PLH}(R)$	1	—	14	ns
Propagation Delay Time from Receiver (Bus) Input to Low Logic State Receiver Output	$t_{PHL}(R)$	1	—	14	ns
Propagation Delay Time from Driver Input to High Logic State Driver (Bus) Output	$t_{PLH}(D)$	2	—	14	ns
Propagation Delay Time from Driver Input to Low Logic State Driver (Bus) Output	$t_{PHL}(D)$	2	—	14	ns
Propagation Delay Time from Receiver Enable Input to High Impedance (Open) Logic State Receiver Output	$t_{PLZ}(RE)$	3	—	15	ns
Propagation Delay Time from Receiver Enable Input to Low Logic Level Receiver Output	$t_{PZL}(RE)$	3	—	20	ns
Propagation Delay Time from Driver Enable Input to High Impedance Logic State Driver (Bus) Output	$t_{PLZ}(DE)$	4	—	20	ns
Propagation Delay Time from Driver Enable Input to Low Logic State Driver (Bus) Output	$t_{PZL}(DE)$	4	—	25	ns

FIGURE 1 — TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY FROM BUS (RECEIVER) INPUT TO RECEIVER OUTPUT, $t_{PLH}(R)$ AND $t_{PHL}(R)$





MC6880A, MC8T26A

FIGURE 2 - TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM DRIVER INPUT TO BUS (DRIVER) OUTPUT, $t_{PLH(D)}$ AND $t_{PHL(D)}$

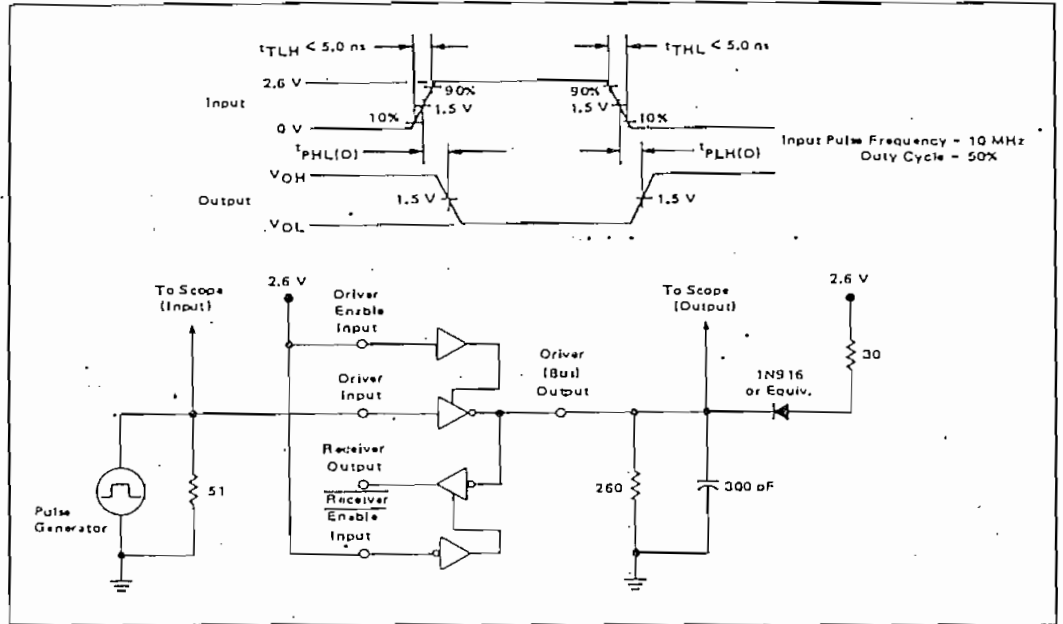


FIGURE 3 - TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIME FROM RECEIVER ENABLE INPUT TO RECEIVER OUTPUT, $t_{PLZ(RE)}$ AND $t_{PZL(RE)}$

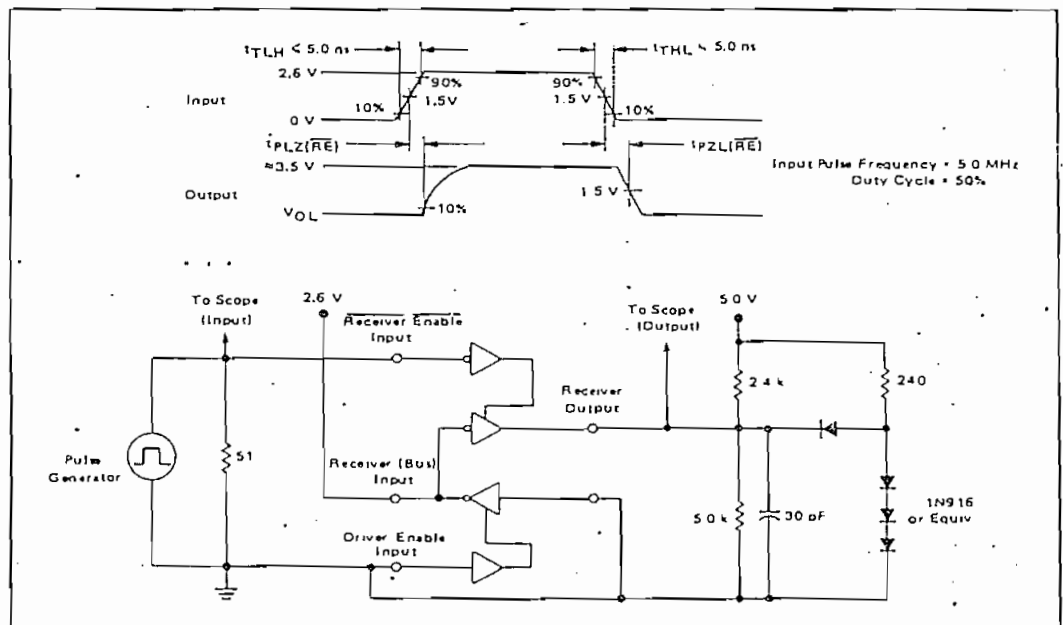


FIGURE 4 - TEST CIRCUIT AND WAVEFORMS FOR PROPAGATION DELAY TIMES FROM DRIVER ENABLE INPUT TO DRIVER (BUS) OUTPUT, $t_{PLZ(DE)}$ AND $t_{PLZ(IE)}$

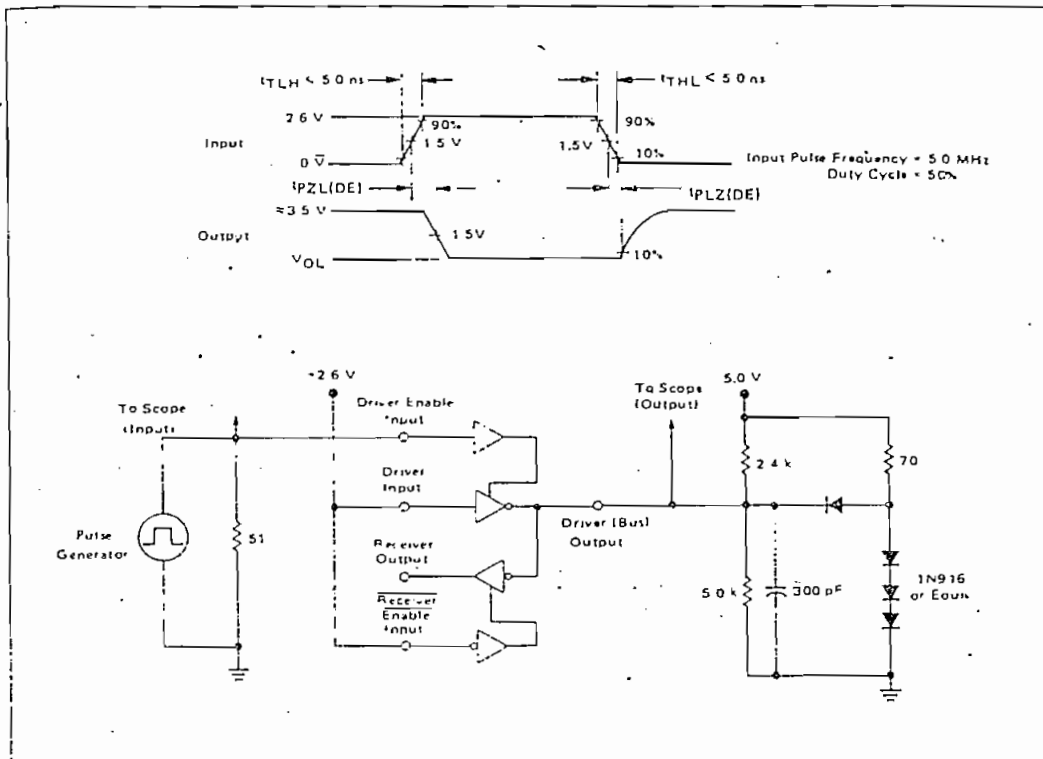
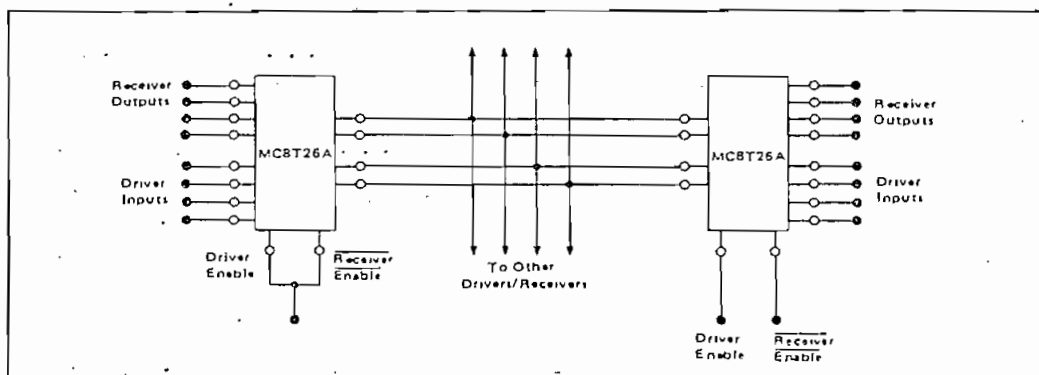
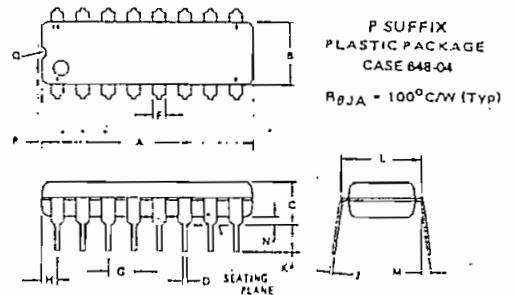
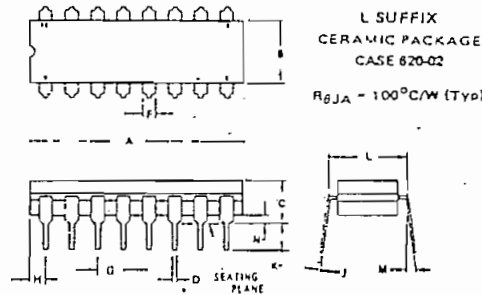


FIGURE 5 - BIDIRECTIONAL BUS APPLICATIONS



MC6880A, MC8T26A



- NOTES:
- LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION
 - PKG INDEX NOTCH IN LEAD NOTCH IN CERAMIC OR INX DOT
 - DIM "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

- NOTES:
- LEADS WITHIN 0.13 mm (0.005) RADIUS OF TRUE POSITION AT SEATING PLANE AT MAXIMUM MATERIAL CONDITION.
 - DIMENSION "L" TO CENTER OF LEADS WHEN FORMED PARALLEL

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	13.05	19.81	0.750	0.780
B	6.22	6.98	0.245	0.275
C	4.06	5.08	0.160	0.200
D	0.38	0.51	0.015	0.020
F	1.40	1.65	0.055	0.065
G	2.54 BSC		0.100 BSC	
H	0.51	1.14	0.020	0.045
J	0.20	0.30	0.008	0.012
K	3.14	4.06	0.125	0.160
L	7.37	7.87	0.290	0.310
M	—	15°	—	15°
N	0.51	1.02	0.020	0.040

DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	—	22.10	—	0.870
B	6.10	6.60	0.240	0.260
C	—	5.08	—	0.200
D	0.38	0.52	0.015	0.021
F	—	1.78	—	0.070
G	2.54 BSC		0.100 BSC	
H	0.38	2.41	0.015	0.095
J	0.20	0.30	0.008	0.015
K	2.92	—	0.115	—
L	7.62 BSC		0.300 BSC	
M	0°	15°	0°	15°
N	0.51	—	0.020	—
R	—	8.26	—	0.325

THERMAL INFORMATION

The maximum power consumption an integrated circuit can tolerate at a given operating ambient temperature, can be found from the equation:

$$PD(T_A) = \frac{T_{J(max)} - T_A}{R_{\theta JA}(Typ)}$$

Where $PD(T_A)$ = Power Dissipation allowable at a given operating ambient temperature. This must be greater than

the sum of the products of the supply voltages and supply currents at the worst case operating condition

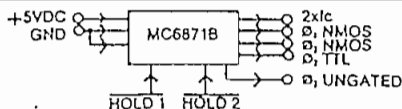
$T_{J(max)}$ = Maximum Operating Junction Temperature as listed in the Maximum Ratings Section

T_A = Maximum Desired Operating Ambient Temperature

$R_{\theta JA}(Typ)$ = Typical Thermal Resistance Junction to Ambient

MC6871B

alternate function microprocessor clock
250 kHz to 2.5 MHz



specifications

Rating	Symbol	Value	Unit
Supply Voltage	V_{cc}	$5.00 \pm 5\%$	Vdc
Operating Temperature Range	T_a	0 to +70	°C
Storage Temperature	T_{stg}	-55 to +125	°C
Power Supply Drain (max)	I_{cc}	100	mA

ELECTRICAL CHARACTERISTICS ($V_{cc} = 5.0 \pm 5\%$, $V_{ii} = 0$, $T_a = 0^\circ$ to 70° C, unless otherwise noted)

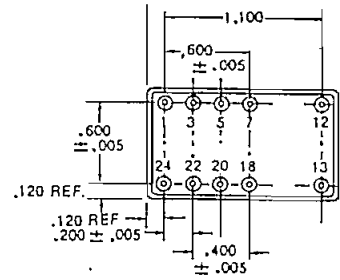
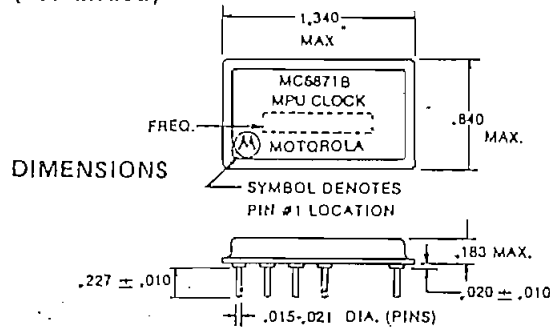
Characteristic	Symbol	Min	Typ	Max	Unit
Frequency					
Operating Frequency	f_c	250		2.5	MHz
Frequency stability (inclusive of calibration tolerance at -25° C, operating temperature, supply voltage change, load change, aging, shock and vibration)			± 0.1		%
NMOS Outputs at 1.0 MHz Operation**					
Pulse Width (measured at $V = -0.3V$ dc level)	T_{OH}	430			ns
	T_{OH}	450			ns
Logic Levels	V_{OL}	$V_{ii} + .1$		$V_{ii} + .3$	Vdc
	V_{OL}	$V_{ii} + .3$		$V_{ii} + .1$	Vdc
Rise and Fall Times	t_r	5	12	50	ns
	t_f	5	12	50	ns
Overshoot/Undershoot Logic "1"		$V_{ii} - .5$		$V_{ii} + .5$	Vdc
Logic "0"	V_{OI}	$V_{ii} + .5$		$V_{ii} + .5$	Vdc
Pulse duration of any overshoot or undershoot	T_{DS}			40	ns
Period @ 0.3V dc Level	t_{cc}		100		us
Edge Timing @ $V = 0.3V$ dc	T_{λ}	940			ns
NMOS Relationship @ $-0.5V$ dc	t_{OL}	0		8.0	us
TTL Outputs					
Relative to 0, NMOS @ 0.3V dc					
0, TTL @ 1.4V dc	T_r	15	30	45	ns
	T_f	10	25	40	ns
0, Ungated @ 1.4V dc	T_c	30	50	70	ns
	T_f	20	40	60	ns
2x1c @ 1.4V dc	T_i	40	80	120	ns
Logic Levels	V_{OH}	2.4	3.2		Vdc
	V_{OI}		.3	.4	Vdc
Rise and Fall Times .4V and 2.4V	t_r			15	ns
2.4V and .4V	t_f			15	ns
Logic "0" Sink (/Gate)	I_{OL}			-1.6	mA
Logic "1" Source (/Gate)	I_{OH}			+40	uA
Current Output Shorted	I_{SC}	-18		-57	mA
Load					
NMOS—Load Capacity 0, 0,	C_{NLOS}	80	120	160	pf
TTL—No. of Loads				5	till
TTL—Load Capacity	C_{TLL}			50	pf
Logic inputs** ("0" Level applies HOLD)					
Holds 0, NMOS "High", 0, NMOS "Low", 0, TTL "Low"	HOLD 1	-2		+4	Vdc
Holds 0, NMOS "Low", 0, NMOS "High", 0, TTL "High"	HOLD 2	-2		+4	Vdc

* Inic specified test load
 ** Must be externally held at "1" level (2.4V min., 5.0V max.) if not used
 *** Apply the following parameters for frequencies other than 1 MHz
 $t_{OH} = 0.5 (P-140)$ ns
 $t_{OL} = 0.5 (P-100)$ ns
 $T_r = (P-60)$ ns
 where P=desired period of operation in nanoseconds

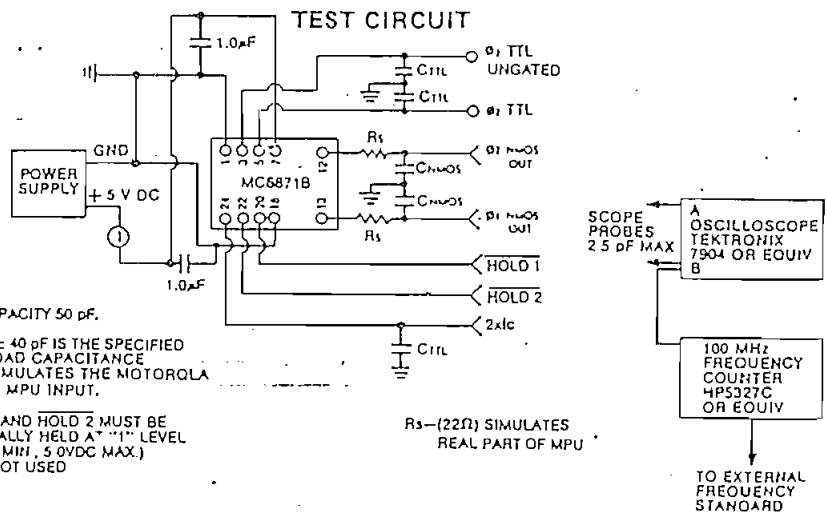
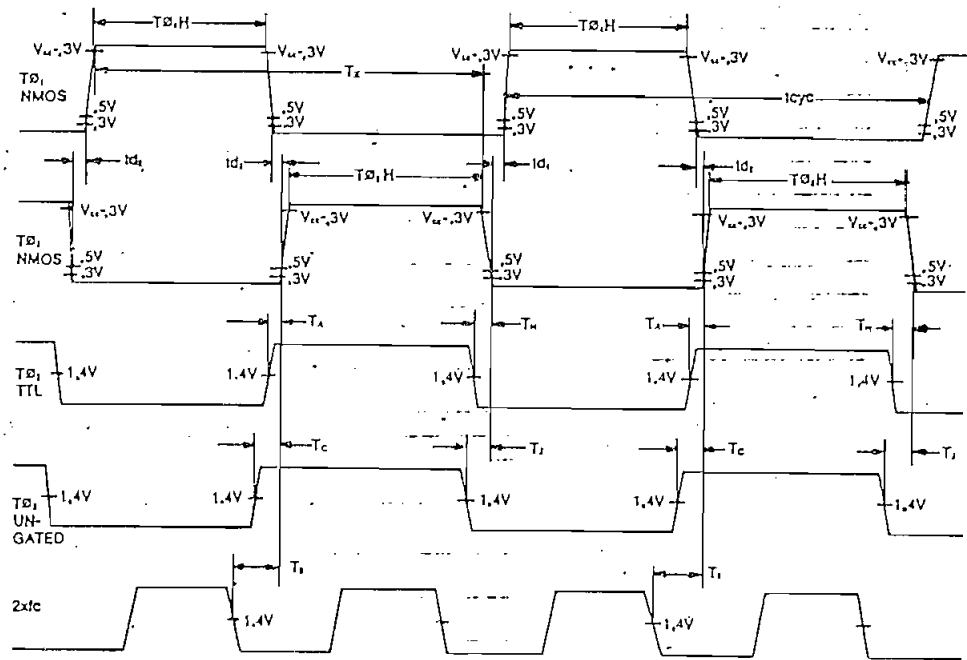
PIN	CONNECTION
1	GND
3	0, TTL UNGATED
5	0, TTL
7	V_{cc} (+5VDC)
12	0, NMOS
13	0, NMOS
18	GND
20	HOLD 1
22	HOLD 2
24	2x1c

Note: 4x1c available on request
 Note: All dimensions are in inches

MC6871 (continued)



WAVEFORM TIMING
ALL TIME IN NANoseconds.



TTL
MSI

TYPES SN54S280, SN74S280
9-BIT ODD/EVEN PARITY GENERATORS/CHECKERS

BULLETIN NO. DL-S 7211829, DECEMBER 1972

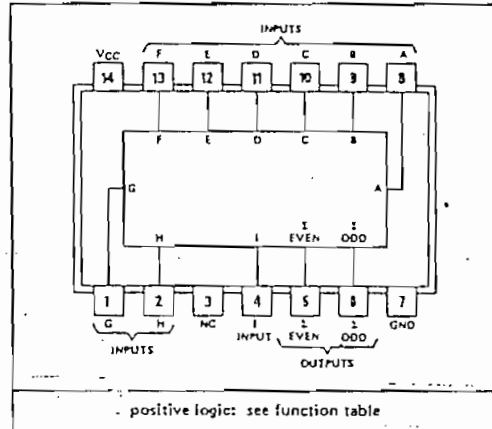
- Generates Either Odd or Even Parity for Nine Data Lines
- Cascadable for n-Bits
- Can Be Used to Upgrade Existing Systems using MSI Parity Circuits
- Typical Data-to-Output Delay of Only 14 ns

FUNCTION TABLE

NUMBER OF INPUTS A THRU I THAT ARE HIGH	OUTPUTS	
	Σ EVEN	Σ ODD
0, 2, 4, 6, 8	H	L
1, 3, 5, 7, 9	L	H

H = high level, L = low level

J OR N DUAL-IN-LINE OR W FLAT PACKAGE (TOP VIEW)



NC—No internal connection

description

These universal, monolithic, nine-bit parity generators/checkers utilize Schottky-clamped Series 54S/74S TTL high-performance circuitry and feature odd/even outputs to facilitate operation of either odd or even parity applications. The word-length capability is easily expanded by cascading as shown under typical application data.

The 'S280' can be used to upgrade the performance of most systems utilizing the '180 parity generator/checker. Although the 'S280 is implemented without expander inputs, the corresponding function is provided by the availability of an input at pin 4 and the absence of any internal connection at pin 3. This permits the 'S280 to be substituted for the '180 in existing designs to produce an identical function even if 'S280's are mixed with existing '180's.

The 'S280 is fully compatible with most other TTL and DTL circuits. Input buffers are provided so that each input represents only one normalized Series 54S/74S load, and full fan-out to 10 normalized Series 54S/74S loads is available from each of the outputs at low logic levels. A fan-out to 20 normalized Series 54S/74S loads is provided at high logic levels to facilitate connection of unused inputs to used inputs. Typical power dissipation is 335 milliwatts.

The SN54S280 is characterized for operation over the full military temperature range of -55°C to 125°C ; the SN74S280 is characterized for operation from 0°C to 70°C .

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage (see Note 1)	7 V
Input voltage	5.5 V
Operating free-air temperature range: SN54S280	-55°C to 125°C
SN74S280	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

TYPES SN54S280, SN74S280

9-BIT, ODD/EVEN PARITY GENERATORS/CHECKERS

recommended operating conditions

	SN54S280			SN74S280			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-1			-1	mA
Low-level output current, I_{OL}			20			20	mA
Operating free-air temperature, T_A	-55		125	0		70	$^{\circ}$ C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS ¹	MIN	TYP ²	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage				0.8	V
V_I Input clamp voltage	$V_{CC} = \text{MIN}; I_I = -18 \text{ mA}$			-1.2	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = 0.8 \text{ V}, I_{OH} = -1 \text{ mA}$		2.5 2.7	3.4 3.4	V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = 0.8 \text{ V}, I_{OL} = 20 \text{ mA}$			0.5	V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 5.5 \text{ V}$			1	mA
I_{IH} High-level input current	$V_{CC} = \text{MAX}, V_I = 2.7 \text{ V}$			50	μ A
I_{IL} Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.5 \text{ V}$			-2	mA
I_{OS} Short-circuit output current ³	$V_{CC} = \text{MAX}$			-40 -100	mA
I_{CC} Supply current	$V_{CC} = \text{MAX}$, See Note 2	SN54S280	67	99	mA
		SN74S280	67	105	mA
	$V_{CC} = \text{MAX}, T_A = 125^{\circ}\text{C}$, See Note 2	SN54S280N		94	mA

¹For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

²All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$.

³Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.

NOTE 2: I_{CC} is measured with all inputs grounded and all outputs open.

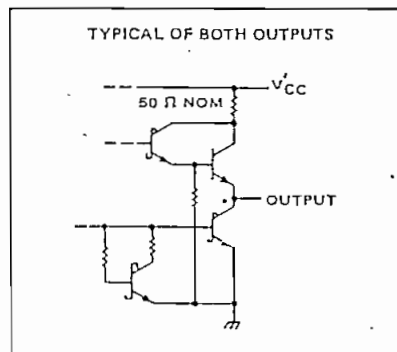
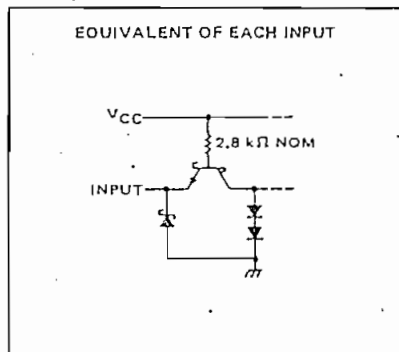
switching characteristics, $V_{CC} = 5 \text{ V}, T_A = 25^{\circ}\text{C}$

PARAMETER ¹	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH}	Data	Σ Even	$C_L = 15 \text{ pF}, R_L = 180 \Omega$, See Note 3		14	21	ns
t_{PHL}					11.5	18	
t_{PLH}	Data	Σ Odd			14	21	ns
t_{PHL}					11.5	18	

¹ t_{PLH} = propagation delay time, low-to-high-level output; t_{PHL} = propagation delay time, high-to-low-level output

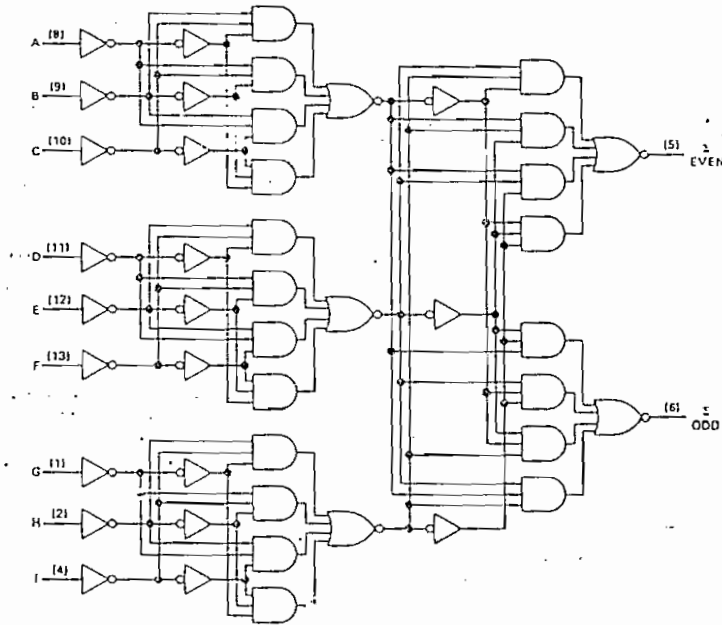
NOTE 3: Load circuit and voltage waveforms are shown on page 148.

schematics of inputs and outputs



TYPES SN54S280, SN74S280 9-BIT ODD/EVEN PARITY GENERATORS/CHECKERS

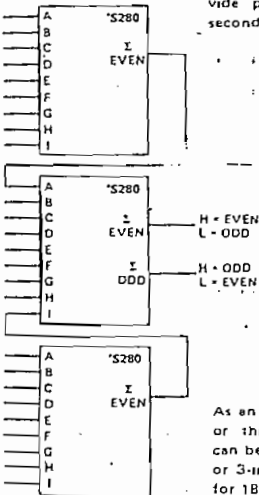
functional block diagram



TYPICAL APPLICATION DATA

25-LINE PARITY/GENERATOR CHECKER

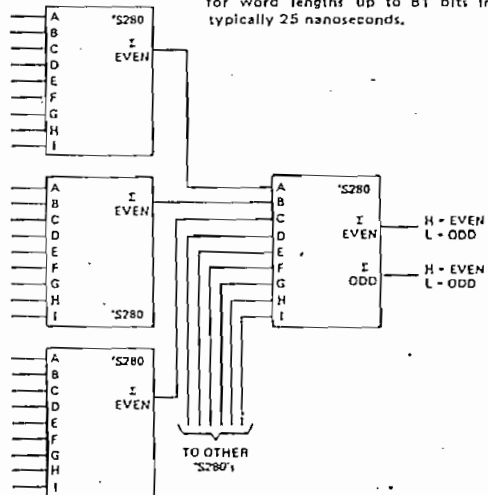
Three 'S280's can be used to implement a 25-line parity generator/checker. This arrangement will provide parity in typically 25 nanoseconds.



As an alternative, the outputs of two or three parity generators/checkers can be decoded with a 2-input ('S86) or 3-input ('S135) exclusive-OR gate for 18- or 27-line parity applications.

81-LINE PARITY/GENERATOR CHECKER

Longer word lengths can be implemented by cascading 'S280's. As shown here, parity can be generated for word lengths up to 81 bits in typically 25 nanoseconds.



TTL
MSI

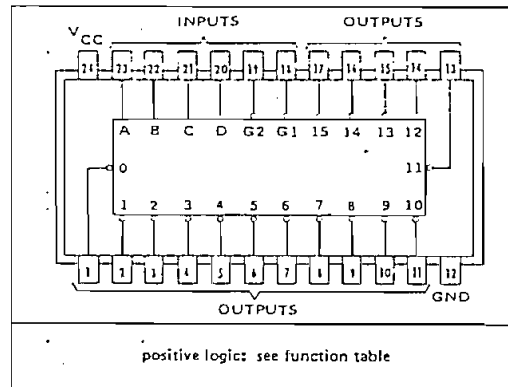
TYPES SN54154, SN54L154, SN74154, SN74L154 4-LINE-TO-16-LINE DECODERS/DEMULTIPLEXERS

BULLETIN NO. DL-S 7211805, DECEMBER 1972

- '154 is Ideal for High-Performance Memory Decoding
- 'L154 is Designed for Power-Critical Applications
- Decodes 4 Binary-Coded Inputs into One of 16 Mutually Exclusive Outputs
- Performs the Demultiplexing Function by Distributing Data From One Input Line to Any One of 16 Outputs
- Input Clamping Diodes Simplify System Design
- High Fan-Out, Low-Impedance, Totem-Pole Outputs
- Fully Compatible with Most TTL, DTL, and MSI Circuits



'154 ... J, N, OR W PACKAGE
'L154 ... J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

TYPE	TYPICAL AVERAGE PROPAGATION DELAY		TYPICAL POWER DISSIPATION
	3 LEVELS OF LOGIC	STROBE	
'154	23 ns	19 ns	170 mW
'L154	45 ns	38 ns	85 mW

description

Each of these monolithic, 4-line-to-16-line decoders utilizes TTL circuitry to decode four binary-coded inputs into one of sixteen mutually exclusive outputs when both the strobe inputs, G1 and G2, are low. The demultiplexing function is performed by using the 4 input lines to address the output line, passing data from one of the strobe inputs with the other strobe input low. When either strobe input is high, all outputs are high. These demultiplexers are ideally suited for implementing high-performance memory decoders. For ultra-high-speed systems, SN54S138/SN74S138 and SN54S139/SN74S139 are recommended.

These circuits are fully compatible for use with most other TTL and DTL circuits. All inputs are buffered and input clamping diodes are provided to minimize transmission-line effects and thereby simplify system design.

Series 54 and 54L devices are characterized for operation over the full military temperature range of -55°C to 125°C; Series 74 and 74L devices are characterized for operation from 0°C to 70°C.

TYPES SN54154, SN74154

4-LINE-TO-16-LINE DECODERS/ DEMULTIPLEXERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	5.5 V
Operating free-air temperature range: SN54154 Circuits	-55°C to 125°C
SN74154 Circuits	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

	SN54154			SN74154			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-800			-800	μ A
Low-level output current, I_{OL}			16			16	mA
Operating free-air temperature, T_A	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	SN54154			SN74154			UNIT
		MIN	TYP	MAX	MIN	TYP‡	MAX	
V_{IH} High-level input voltage		2			2			V
V_{IL} Low-level input voltage				0.8			0.8	V
V_I Input clamp voltage	$V_{CC} = \text{MIN}, I_I = -12 \text{ mA}$			-1.5			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = 0.8 \text{ V}, I_{OH} = -800 \mu\text{A}$	2.4	3.4		2.4	3.4		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN}, V_{IH} = 2 \text{ V}, V_{IL} = 0.8 \text{ V}, I_{OL} = 16 \text{ mA}$		0.2	0.4		0.2	0.4	V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX}, V_I = 5.5 \text{ V}$			1			1	mA
I_{IH} High-level input current	$V_{CC} = \text{MAX}, V_I = 2.4 \text{ V}$			40			40	μ A
I_{IL} Low-level input current	$V_{CC} = \text{MAX}, V_I = 0.4 \text{ V}$			-1.6			-1.6	mA
I_{OS} Short-circuit output current§	$V_{CC} = \text{MAX}$	-20		-55	-18		-57	mA
I_{CC} Supply current	$V_{CC} = \text{MAX}$, See Note 2		34	49		34	56	mA

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.

‡ All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^\circ \text{C}$.

§ Not more than one output should be shorted at a time.

NOTE 2: I_{CC} is measured with all inputs grounded and all outputs open.

switching characteristics, $V_{CC} = 5 \text{ V}, T_A = 25^\circ \text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH} Propagation delay time, low-to-high-level output, from A, B, C, or D inputs through 3 levels of logic	$C_L = 15 \text{ pF}, R_L = 400 \Omega,$ See Note 3		24	36	ns
t_{PHL} Propagation delay time, high-to-low-level output, from A, B, C, or D inputs through 3 levels of logic			22	33	ns
t_{PLH} Propagation delay time, low-to-high-level output, from either strobe input			20	30	ns
t_{PHL} Propagation delay time, high-to-low-level output, from either strobe input			18	27	ns

NOTE 3: Load circuit and voltage waveforms are shown on page 148.

TYPES SN54L154, SN74L154 4-LINE-TO-16-LINE DECODERS/ DEMULTIPLEXERS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, V_{CC} (see Note 1)	7 V
Input voltage	5.5 V
Operating free-air temperature range: SN54L154 Circuits	-55°C to 125°C
SN74L154 Circuits	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

	SN54L154			SN74L154			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-400			-400	μ A
Low-level output current, I_{OL}			8			8	mA
Operating free-air temperature, T_A	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage				0.8	V
V_I Input clamp voltage	$V_{CC} = \text{MIN.}, I_I = -12 \text{ mA}$			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}, V_{IH} = 2 \text{ V}, V_{IL} = 0.8 \text{ V}, I_{OH} = -400 \mu\text{A}$	2.4	3.4		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}, V_{IH} = 2 \text{ V}, V_{IL} = 0.8 \text{ V}, I_{OL} = 8 \text{ mA}$		0.2	0.4	V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}, V_I = 5.5 \text{ V}$			1	mA
I_{IH} High-level input current	$V_{CC} = \text{MAX.}, V_I = 2.4 \text{ V}$			20	μ A
I_{IL} Low-level input current	$V_{CC} = \text{MAX.}, V_I = 0.4 \text{ V}$			-0.8	mA
I_{OS} Short-circuit output current §	$V_{CC} = \text{MAX.}$	-9		-29	mA
I_{CC} Supply current	$V_{CC} = \text{MAX.},$ SN54L154		17	25	mA
	See Note 2 SN74L154		17	28	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.

‡ All typical values are at $V_{CC} = 5 \text{ V}, T_A = 25^\circ \text{C}$.

§ Not more than one output should be shorted at a time.

NOTE 2: I_{CC} is measured with all inputs grounded and all outputs open.

switching characteristics, $V_{CC} = 5 \text{ V}, T_A = 25^\circ \text{C}$

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
t_{PLH} Propagation delay time, low-to-high-level output, from A, B, C, or O inputs through 3 levels of logic	$C_L = 15 \text{ pF}, R_L = 800 \Omega,$ See Note 3		48	72	ns
t_{PHL} Propagation delay time, high-to-low-level output, from A, B, C, or D inputs through 3 levels of logic			44	66	ns
t_{PLH} Propagation delay time, low-to-high-level output, from either strobe input			40	60	ns
t_{PHL} Propagation delay time, high-to-low-level output, from either strobe input			36	54	ns

NOTE 3: Load circuit and voltage waveforms are shown on page 148.

APENDICE B

- OPCIONES
- FUENTE DE VOLTAJE
- CONFIGURACION EXTERNA DEL EQUIPO

APENDICE B

OPCIONES

En el Capítulo IV se explicó la posibilidad de eliminar el oscilador de 64 KHZ, el cual genera los ciclos de refresco una vez en aproximadamente 10 ciclos de MPU. Realizando la modificación que se detalla a continuación se logra que se realice un ciclo de refresco en cada ciclo de MPU.

Con esta modificación, el consumo de potencia se incrementa debido a los continuos ciclos de refresco.

Para el procedimiento que sigue, ver las figuras 3-3 y 4-1.

Paso 1. Eliminar la conexión entre la pata 20 del circuito I3 (MC 3480) y la pata 2 de circuito I12 (7474). (Señal $\overline{\text{REF REQ}}$).

Paso 2. Conectar a tierra la pata 2 del circuito I12 (7474).

Paso 3. Eliminar la conexión entre la pata 3 del circuito I7 (555) y las patas 21 y 1 de los circuitos I3 (MC 3480) y I4 (MC 3242) respectivamente. (Señal de salida del oscilador de 64 (KHZ)).

Paso 4. Conectar la pata 10 del circuito i45 (7404) (señal $\overline{\text{L}}$) a las patas 21 y 1 de los circuitos I3 (MC 3480) y I4 (MC 3242) respectivamente.

FUENTE DE VOLTAJE.

APENDICE B (Cont.)

CONFIGURACION EXTERNA DEL EQUIPO

La configuración externa del equipo se la puede ver en el dibujo de la página siguiente.

Se nota que existen 4 conectores de 86 contactos, 2 de los cuales están ocupados con la tarjeta del microprocesador y con la tarjeta de 16 kbytes de memoria. Los 2 restantes quedan libres para usos futuros.

Es importante notar que el orden de las tarjetas no es intercambiable, esto es únicamente debido a las conexiones de las fuentes de alimentación.

La tarjeta del microprocesador, está más cercana a la fuente de voltaje, mientras que, la tarjeta de memoria se encuentra más alejada.

También es importante notar que los lados que contienen los elementos de las tarjetas, se encuentran orientados hacia la fuente de voltaje.

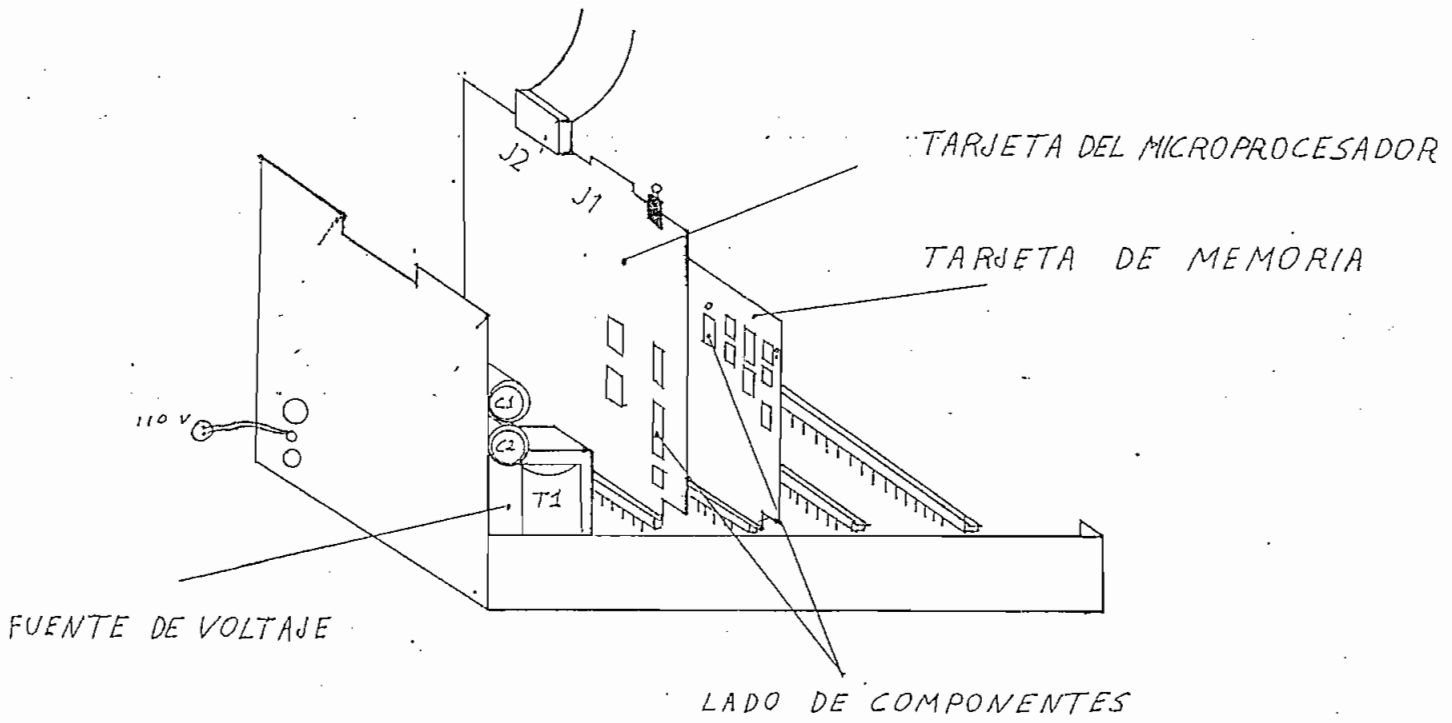
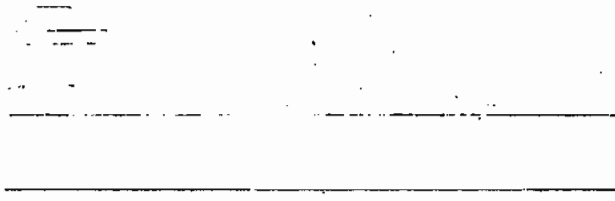


FIGURA B-2

APENDICE C

DIAGRAMA DEL MICROCOMPUTADOR MEK 6800 D2



APENDICE C

A continuación se ilustra el diagrama del microcomputador MEK 6800 D2 al cual se ha acoplado el sistema de memoria construido.

Para poder acoplar el sistema de memoria al microcomputador se requiere que en este último estén realizadas las siguientes modificaciones las cuales, a pesar de no encontrarse ilustradas en el diagrama, ya están realizadas.

- C.1. Los circuitos U₄ y U₅ correspondientes a la "BUFFERS" 8T26, e indicados en el diagrama como opcionales, deben estar instalados.
- C.2. Los circuitos U₁ y U₂ correspondientes a los circuitos 8T97 e indicados como opcionales, deben estar instalados.
- C.3. El circuito U₃ indicado en el diagrama como opcional debe tener puentes en las patas: 2 y 3, 4 y 5, 6 y 7, 9 y 10; y; 11 y 12.
- C.4. La pata número 4 del circuito U₇, correspondiente a la señal $\overline{\text{RAM}}$ debe estar sin conexión o lo que es lo mismo en 1 lógico.

El circuito U₇, es el integrado 7430 que corresponde a una compuerta NAND de 8 entradas. Su salida habilita a los "buffers" 8T26.

Sin conectar la señal $\overline{\text{RAM}}$ a la pata 4 de U₇, su salida puede habilitar los "buffers" 8T26 en el rango 0000 -3FFF y otros rangos que no corresponden a los datos por las señales que ingresan al circuito U₇.

C.5. La memoria RAM incluida en la tarjeta del microcomputar debe trabajar en el rango 4000 - 4180.

Para esto es necesario que la habilitación de los circuitos RAM: U14, U16, U18, U19 en su pata número 15, sea la señal $\overline{4/5}$ la cual dá habilitación en el rango 4000 - 5FFF..

NOTA: Si no se cumple con las modificaciones indicadas en los puntos C4 y C5 y se trata de utilizar los dos sistemas de memoria, esto es, la memoria RAM en la tarjeta del microcomputador y la memoria del sistema construido, se puede causar un daño grave al equipo.

C.6. El circuito U10 debe corresponder a una EPROM 2716 (con las modificaciones necesarias para su conexión). En esta EPROM se almacena el programa de inicialización en la localidad C537.

BIBLIOGRAFIA

Alan Heimlich y Joel Korelitz, "Memory Finds and Fixes Errors to Raise Reliability of Micro Computer". Revista: Electronics / Enero 3, 1980; Pg. 169 - 172.

Mark Brodsky, "Hardening RAMs Against Soft Errors". Revista: Electronics / Abril 24, 1980, Pg. 117 - 122.

Ernest L. Wall, "Applying the Hamming Code to Microprocessor - based systems". Revista: Electronics / Noviembre 23, 1979; Pg. 103 - 110.

Motorola, The Complete Motorola Micro Computer Data Library.

Advanced Micro Devices, An 2960 Series Dynamic Memory Support Handbook.

John Wakerly, Error Detecting Codes, Self-Checking Circuits and Applications.

Mostek, Memory Data Book and Designers Guide, 1979.