

Temporizador Programable en base a un dispositivo MAPL128.

Ing. Aquilino Cervantes Avila
Alumno de la Maestría del CINTEC-IPN

El trabajo con Dispositivos Lógicos Programables (PLD'S) se ha desarrollado ampliamente en este centro de estudios, por lo cual el objetivo de este trabajo es, por medio de un problema relativamente simple, introducir un dispositivo programable de gran potencial.

El trabajo con un Arreglo Múltiple de Lógica Programable ("Múltiple Array Programable Logic", MAPL), ofrece la ventaja de sustituir a múltiples dispositivos de lógica tradicional o de lógica programable; debido a esto, disminuye el área física en tarjeta, además de reducir el consumo de corriente de la fuente de poder empleada.

Hasta el momento se tiene noticia de 4 de estos dispositivos, siendo: MAPL's 128, 144, 244 y 268, fabricados por National Semiconductor. Si bien poseen ligeras variaciones en su configuración interna, la diferencia real la establece el número de compuertas que pueden sustituir y, por ende, el tamaño de su presentación física.

Para este proyecto se aplicará el dispositivo más sencillo de la serie, el MAPL 128. Para ello se realizará una descripción general de este dispositivo.

Características de los dispositivos MAPL128/144.

El MAPL128/144 pertenece a la tecnología EECMOS y posee una arquitectura de arreglos lógicos programables integrando múltiples FPLAS ("Field Programable Logic Array", Arreglo Lógico Programable en Campo), lo que permite facilitar el diseño de máquinas de estado, controladores, secuenciadores de microinstrucciones, interfaces y en general cualquier diseño lógico secuencial. Las características que posee son las siguientes:

Características generales de la serie MAPL128/144:

- o Alta densidad, con arquitectura PLA.
- o Velocidad de operación de 33,40 y 45 MHz dependiendo de la terminación en el dispositivo.
- o Bajo consumo de potencia $I_{cc} = 110$ mA máximo.
- o Borrable eléctricamente.
 - * 100 % de funcionalidad de prueba.
 - * Lógica instantáneamente reconfigurable
 - * Capacidad Mínima de 100 ciclos programación - borrado.
- o 27 macroceldas con registros tipo DE, JK, RS o T.
- o Precarga asíncrona, y capacidad de reinicio.
- o Reemplazo de múltiples dispositivos pal/pla.
- o Soportado por OPAL y por software popular dedicado al tema.
- o Celdas de seguridad que previenen el copiado.

La arquitectura MAPL es funcionalmente equivalente a un FPLA grande, teniendo un total de 128 productos términos. En realidad, el MAPL128 tiene ocho planos o páginas FPLAS, consistiendo cada

página de un arreglo AND y un arreglo OR, ambos programables.

Cada arreglo AND posee 58 entradas y 16 términos producto, los cuales pueden conectarse a cualquiera de las 54 sumas de términos, por lo que cada página tiene una configuración de 58 X 16 X 54.

Paginación

Tres registros internos manejan la paginación, teniendo solo una página activa en un momento dado. Al igual que los demás registros

internos, estos pueden usarse también para almacenar bits.

Las páginas se seleccionan en el momento de la aplicación; tal como las salidas de las macroceldas

Temporizador Programable en base a un dispositivo MAPL128.

Características del MAPL128

Terminales de alimentación	2
Entrada de reloj	1
Entrada de habilitación	1
Terminales de entrada	9
Terminales de salida	4
Terminales de entrada/salida	12
Total	28
Registros de página	3
Registros Internos*	8
Registros de salida	16
Total	27
Total de arreglos AND (entradas)	32
Total de arreglos OR (salidas)	54
Términos de transición lógicos	128
Control de términos	4
Suma de términos	432
Total de términos producto y suma	564

* Registros internos se refiere a que con estos se puede lograr la conexión interna, es decir, un nodo interno.

son controladas por reloj, así también lo son los registros de página. Lo anterior tiene el efecto de preseleccionar la página donde se encuentra el siguiente estado lógico. Debido a que el dispositivo posee arquitectura PIPELINE, la siguiente página no acumula retardo, con lo que la paginación aparece transparente al usuario.

La arquitectura que presenta el MAPL128/144 se muestra en la figura 1 .

Macroceldas

En general, los dispositivos MAPL incluyen una estructura de macrocelda flexible, configurable por el usuario. La misma macrocelda registrada se emplea para las funciones de salida, retroalimentación o entrada/salida. Cada macrocelda puede ser configurada

requieren transformaciones por software o por lógica adicional. Todas las macroceldas se sincronizan por el flanco de subida de la señal de reloj.

Las macroceldas de salida presentan una señal de habilitación (OE) que permite llevar al dispositivo a un estado de alta impedancia. Es posible utilizar esta terminal en la función indicada o como entrada al arreglo AND; además, se tiene que las macroceldas de salida pueden ser configuradas como activas en alto o en bajo.

Las arquitecturas que presentan las macroceldas del MAPL 128 se muestran en las figuras 2 y 3.

NOTA: La macrocelda paginada es exactamente igual a la macrocelda interna. Por su parte, la macrocelda de solo salida, es igual a la de entrada/salida, excepto por la línea de retroalimentación.

como registro JK o D; ambos tipos se implementan en hardware y no

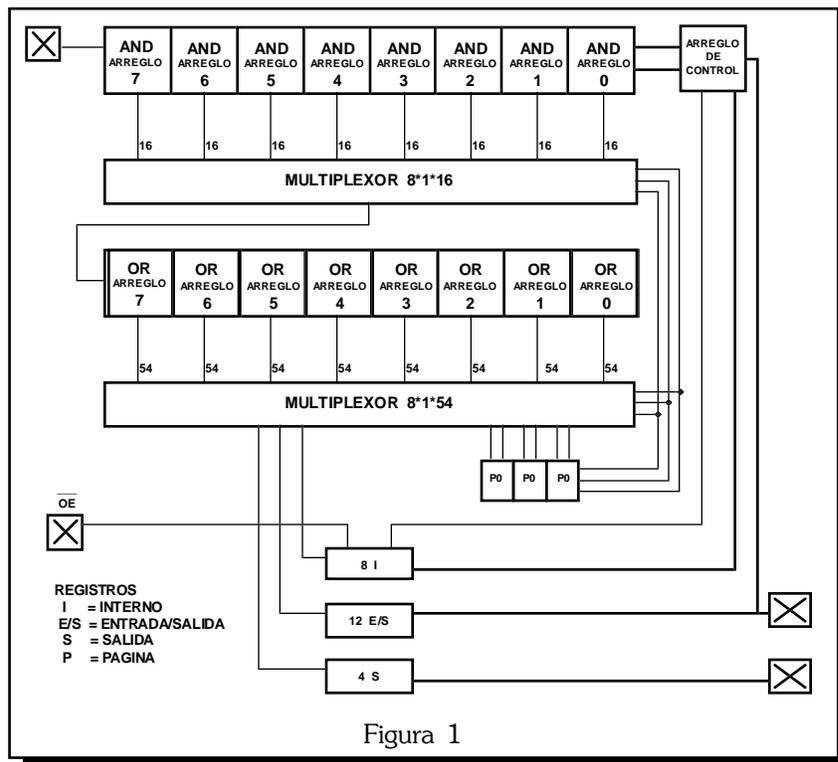


Figura 1

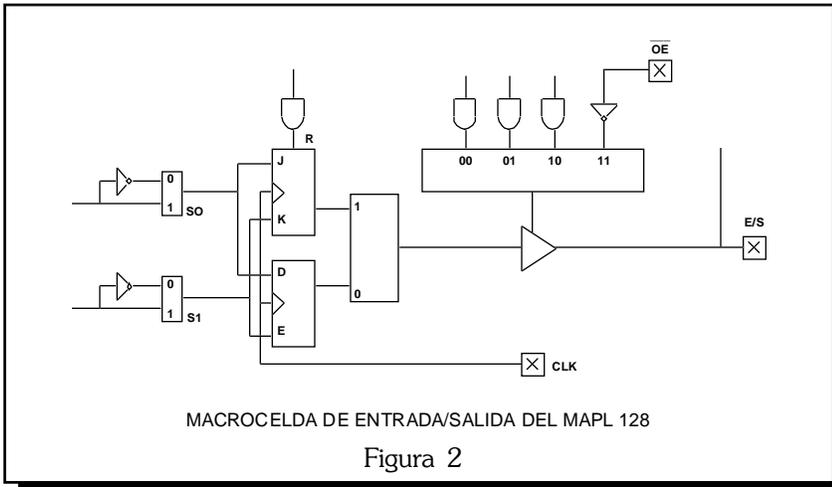


Figura 2

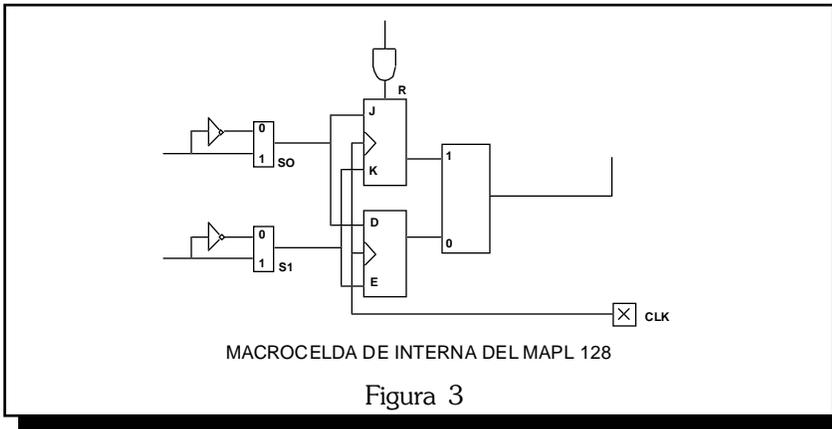


Figura 3

Descripción General del temporizador implementado

El circuito consta de: 4 contadores tipo ascendente - descendente y una unidad de control; 3 contadores trabajan en módulo 10 y el restante en módulo 6. La unidad de control se encarga de conectar los contadores en cascada para la forma descendente y de forma individual para la opción ascendente; el orden en que la unidad de control conecta los contadores, da como resultado el número 9:59:9 para cuenta máxima, generando así el mayor retardo. La programación de cada contador se realiza en forma individual; para esto se elige alguno de ellos y se lleva a la cuenta

deseada (cuenta en forma ascendente), una vez que la cuenta ha sido seleccionada, se marca el inicio de cuenta y los contadores inician su actividad en forma descendente.

Existen dos líneas de control, PROG Y CHG: la primera se utiliza para lograr la programación, mientras que la segunda para la selección del contador deseado. Una vez que el sistema se programa con ambas señales, se marca el inicio de cuenta descendente, a partir del

PROG	CHG	FUNCION
0	0	Inicio de cuenta descendente
0	1	Cuenta ascendente para el contador elegido
1	0	Cambio de contador
1	1	Cuenta descendente

Una vez que la cuenta termina, una línea señala el fin de la cuenta ENDCOUNT.

valor preestablecido en la programación. Esto se resume en la siguiente tabla.

Se presenta además una señal de entrada que permite la inhibición de una nueva cuenta descendente (NEW), o el abandono de la cuenta actual; si esta señal se aterriza, después que el temporizador llega al fin de su cuenta descendente (0000) el circuito continúa con una nueva cuenta descendente a partir de 9 : 59 : 9, y así en forma sucesiva. Si se desea tener solo una cuenta, entonces la señal de ENDCOUNT se conecta en forma directa a la terminal NEW. De esta forma, la cuenta se detiene en cero y permite programar al sistema nuevamente, si así se desea, o reiniciar la cuenta en que se encontraba el temporizador en el momento en que se detuvo.

El diagrama a bloques del temporizador se muestra en la figura 4.

Programa de aplicación

Para este proyecto se empleó la herramienta de programación OPAL, proporcionado por National Semiconductor. El método que se usó para la solución del problema es el de tablas de verdad. Dado que el OPAL no soporta dos tipos de tablas diferentes (si se desea verificar el ejemplo), es necesario trabajar con métodos alternativos, o compilar en forma parcial las diferentes tablas, y utilizar los módulos EQN.

Temporizador Programable en base a un dispositivo MAPL128.

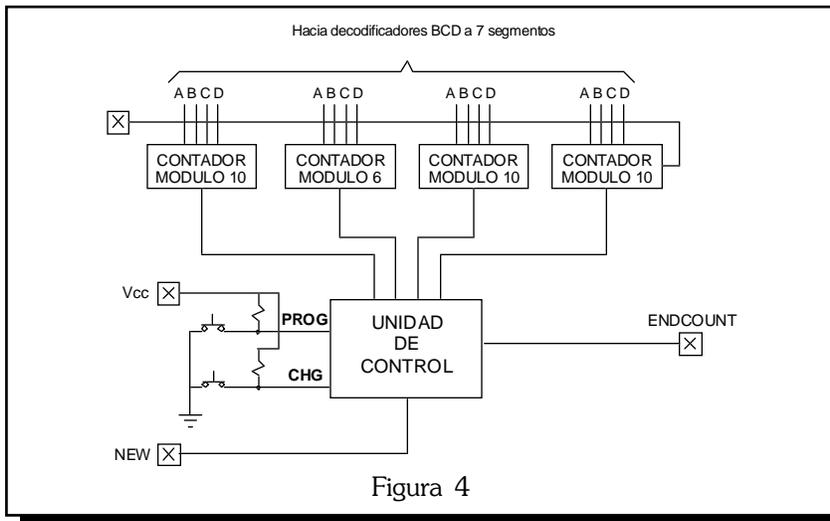


Figura 4

En esta aplicación se maneja paginación manual para lograr un mejor control sobre el sistema. Si en el sistema se deja que el OPAL realice la paginación automática, es probable que se marque errores en la compilación, esto es debido a que el OPAL intenta aprovechar al máximo el espacio por página, por lo que es posible que alguna ecuación sobrepase el número máximo de términos producto y no sea posible crear el archivo JED para el programa. A continuación se tiene una explicación del programa, y en cuadros anexos el listado del mismo.

Explicación general del programa

El programa ocupa las 8 páginas del MAPL, esto es, cada contador ocupa dos páginas, una de ellas se destina para la cuenta ascendente y otra más para la cuenta descendente. La unidad de control se organiza dentro de las 8 páginas y va ligada en forma directa con el control de la paginación.

Si se observa en forma detenida la manera en que se logra la conexión de los contadores en cascada, se podrá ver que ésta se da gracias precisamente a que las pá-

ginas se encuentran multiplexadas, así la conmutación de páginas es la que realiza la conexión indicada y debido a que todas las salidas son registradas, se observa el efecto de captura (HOLD).

También es necesario indicar que el MAPL 128 posee únicamente 12 salidas configurables también como entradas; si consideramos que el temporizador emplea 4 contadores con un total de 15 salidas, se aprecia la necesidad de obtener la retroalimentación de alguna manera para compensar las tres salidas faltantes. Para evitar un aca-

```

BEGINHEADER
                                IPN
                                CENTRO DE INVESTIGACION TECNOLÓGICA EN COMPUTACION
                                EDIFICIO DE GRADUADOS DE UPIICSA
                                CALLE TE No. 950 COL. GRANJAS MEXICO MEXICO D.F. CP. 08400
                                ALUMNO: AQUILINO CERVANTES AVILA

.DESCRIPCION:
    El siguiente programa implementa un temporizador programable en un MAPL 128

ENDHEADER

BEGINDEFINITION

    Device MAPL128;

    Inputs   clk,chg,prog,new;           { señales de control }

    Outputs  (jk,hold) endcount,al0,al1,al2; { señales solo salida }

    Feedbacks (jk,hold) sl0,sl1,sl2,sl3;   { forman al contador menos significativo }
    Feedbacks (jk,hold) ml0,ml1,ml2,ml3;   { forman al contador que sigue al menos significativo }
    Feedbacks (jk,hold,buried) l0,l1,l2;   { forman al contador que sigue al más significativo }
    Feedbacks (jk,hold) s0,s1,s2,s3;      { forman al contador más significativo }
    Feedbacks (buried) p0,p1,p2;          { registros internos de paginación }

ENDEFFINITIONS

BEGINEQUATIONS

    al0 = l0;                            { se ocupan tres registros internos (l0, l1 y l2) para el contador }
    al1 = l1;                            { módulo 6, y se conectan a tres salidas (al0, al1 y al2) }
    al2 = l2;

                                { se garantiza que ENDCOUNT está activa solo al término }
                                { de la cuenta descendente. }

    aux1 = sl0 * /p0 * /p1 * p2;
    aux2 = /sl1 * /sl2 * /sl3 * /ml0 * /ml1 * /ml2 * /ml3;

    endcount = aux1 * aux2 * /l0 * /l1 * /l2 * /s0 * /s1 * /s2 * /s3;

ENDEQUATIONS

BEGINTRUTH_TABLE                { tabla de verdad para el contador ascendente }
                                { menos significativo }

    TTIN chg,prog,sl3,sl2,sl1,sl0,p2,p1,p0;
    TTOUT sl3,sl2,sl1,sl0,p2,p1,p0;

    01  ———  000 ???? 001  { si chg,prog están en 01 o en 11, sin importar el }
    11  ———  000 ???? 000  { contenido de sl3-sl2, captura su valor y hace un }
    
```

```

00 — 000 ??? 100 { cambio de página a 001 o 000 respectivamente }
                { La combinación chg,prog = 00 provoca un inicio }
                { de cuenta descendente }

10 0000 000 0001 000
10 0001 000 0010 000
10 0010 000 0011 000 { la combinación chg,prog = 10 permite al contador }
10 0011 000 0100 000 { continuar con su cuenta }
10 0100 000 0101 000
10 0101 000 0110 000
10 0110 000 0111 000
10 0111 000 1000 000 { si se efectúa el cambio de página a 001, se busca }
10 1000 000 1001 000 { el siguiente contador ascendente por programar }
10 1001 000 0000 000
    
```

ENDTRUTH_TABLE

BEGINTRUTH_TABLE { tabla de verdad para el contador descendente }
 { menos significativo }

TTIN new,chg,prog,sl3,sl2,sl1,sl0,p2,p1,p0;
 TTOUT sl3,sl2,sl1,sl0,p2,p1,p0;

```

111 — 100 ??? 000 { si new,chg y prog se encuentran en 1, se }
                { captura el contenido de sl3-sl0 y salta a la }
                { página 000 (donde es posible la precarga) }

011 0000 100 1001 101
011 1001 100 1000 100
011 1000 100 0111 100
011 0111 100 0110 100 { para que el contador trabaje es necesario la }
011 0110 100 0101 100 { combinación de bits 011 dada por new, chg }
011 0101 100 0100 100 { y prog en forma respectiva. Este contador se }
011 0100 100 0011 100 { localiza en la página 100 }
011 0011 100 0010 100
011 0010 100 0001 100
011 0001 100 0000 100
    
```

ENDTRUTH_TABLE

BEGINTRUTH_TABLE { tabla de verdad para el contador ascendente- }
 { descendente que sigue al menos significativo }

TTIN chg,prog,ml3,ml2,ml1,ml0,p2,p1,p0;
 TTOUT ml3,ml2,ml1,ml0,p2,p1,p0;

```

01 — 001 ??? 011 { si chg,prog están en 01 o en 11, sin importar el }
11 — 001 ??? 001 { contenido de sl3-sl2, captura su valor y hace un }
00 — 001 ??? 100 { cambio de página a 011 o 001 respectivamente }
                { La combinación chg,prog = 00 provoca un inicio }
                { de cuenta descendente }

10 0000 001 0001 001
10 0001 001 0010 001
10 0010 001 0011 001
10 0011 001 0100 001 { el contador ascendente se encuentra en la página }
10 0100 001 0101 001 { 001 y trabaja con la combinación 01 dada por chg }
10 0101 001 0110 001 { y prog }
10 0110 001 0111 001
10 0111 001 1000 001
10 1000 001 1001 001
10 1001 001 0000 001

11 0000 101 1001 110
11 1001 101 1000 100
11 1000 101 0111 100
11 0111 101 0110 100 { el contador descendente se encuentra en la página }
11 0110 101 0101 100 { 101 actualiza registros y regresa al contador descen- }
11 0101 101 0100 100 { dente menos significativo. Si existe un cambio de }
11 0100 101 0011 100 { estado 0000 a 1001, se busca el contador siguiente }
11 0011 101 0010 100 { en la página 110 }
11 0010 101 0001 100
11 0001 101 0000 100
    
```

ENDTRUTH_TABLE

reos externos, y además el uso probable de lógica adicional, el contador módulo 6 se efectúa por medio de 3 registros internos conectados, también en forma interna, a 3 registros de sólo salida, solucionando con esto el problema.

De esta forma, si el MAPL cuenta con 16 terminales de salida, resta indicar que la última salida se utiliza para marcar el fin de cuenta.

Conclusiones

El trabajo con el MAPL 128 sirve como introducción para el manejo de otros dispositivos más complejos, tales como GATE ARRAY's, FPGA's, ASIC's, etc. Sin embargo, aunque en este ejemplo no se utiliza, es necesario usar la técnica de "pruebabilidad", que permite verificar si un proyecto dentro de un dispositivo programable se encuentra trabajando en forma adecuada.

Lo más interesante de este ejemplo es, sin duda, el manejo de la paginación. Las páginas dentro del MAPL se encuentran multiplexadas, por lo cual solo un plano está activo a la vez. Debido a esta característica es posible insertar los decodificadores de BCD a 7 segmentos dentro del propio MAPL. Sin embargo, debido a que se necesita multiplexar las líneas del decodificador (si se desean tener líneas independientes, no son suficientes las salidas del dispositivo), sería necesario el aumentar un latch para cada contador y agregar al menos dos señales (con las que se tienen cuatro combinaciones) de control para indicar en que momento se encuentran válidos los datos de qué contador. El trabajo, finalmente, resulta en agregar circuitos tipo latch o colocar decodificadores, optándose por lo segundo. El ejem-

```

BEGINTRUTH_TABLE      { tabla de verdad para el contador ascendente- }
                      { descendente que sigue al más significativo }

TTIN chg,prog,l2,l1,l0,p2,p1,p0;
TTOUT l2,l1,l0,p2,p1,p0;

01  — 010  ??? 011      { si chg,prog están en 01 o en 11, sin importar el }
11  — 010  ??? 010      { contenido de l2 - l0, captura su valor y hace un }
00  — 010  ??? 100      { cambio de página a 011 o 010 respectivamente }
                      { La combinación chg,prog = 00 provoca un inicio }
                      { de cuenta descendente }

10 0000 010 0001 010
10 0001 010 0010 010
10 0010 010 0011 010
10 0011 010 0100 010      { el contador ascendente se encuentra en la página }
10 0100 010 0101 010      { 010 y trabaja con la combinación 01 dada por chg }
10 0101 010 0110 010      { y prog }
10 0110 010 0111 010
10 0111 010 1000 010
10 1000 010 1001 010
10 1001 010 0000 010

11 0000 110 1001 111
11 1001 110 1000 100
11 1000 110 0111 100
11 0111 110 0110 100      { el contador descendente se encuentra en la página }
11 0110 110 0101 100      { 110 actualiza registros y regresa al contador descen- }
11 0101 110 0100 100      { dente menos significativo. Si existe un cambio de }
11 0100 110 0011 100      { estado 0000 a 1001, se busca el contador siguiente }
11 0011 110 0010 100      { en la página 111 }
11 0010 110 0001 100
11 0001 110 0000 100

ENDTRUTH_TABLE

BEGINTRUTH_TABLE      { tabla de verdad para el contador ascendente- }
                      { descendente que sigue al más significativo }

TTIN chg,prog,s3,s2,s1,s0,p2,p1,p0;
TTOUT s3,s2,s1,s0,p2,p1,p0;

01  — 011  ??? 000      { si chg,prog están en 01 o en 11, sin importar el }
11  — 011  ??? 011      { contenido de l2 - l0, captura su valor y hace un }
00  — 011  ??? 100      { cambio de página a 011 o 010 respectivamente }
                      { La combinación chg,prog = 00 provoca un inicio }
                      { de cuenta descendente }

10 0000 011 0001 011
10 0001 011 0010 011
10 0010 011 0011 011
10 0011 011 0100 011      { el contador ascendente se encuentra en la página }
10 0100 011 0101 011      { 011 y trabaja con la combinación 01 dada por chg }
10 0101 011 0110 011      { y prog. }
10 0110 011 0111 011
10 0111 011 1000 011
10 1000 011 1001 011
10 1001 011 0000 011

11 0000 111 1001 100
11 1001 111 1000 100
11 1000 111 0111 100
11 0111 111 0110 100
11 0110 111 0101 100      { el contador descendente se encuentra en la página }
11 0101 111 0100 100      { 110 actualiza registros y regresa al contador descen- }
11 0100 111 0011 100      { dente menos significativo. Este contador es el más }
11 0011 111 0010 100      { significativo y no necesita llevar acarreo a otro contador }
11 0010 111 0001 100
11 0001 111 0000 100

ENDTRUTH_TABLE

```

plo descrito, con los decodificadores integrados, también se desarrolló y está a disposición de quien lo solicite. El paquete de OPAL, se trabajó en una computadora 80386DX a 40 MHz.

Bibliografía

- [1] Programable Logic Devices Data Book and Design Guide. National Semiconductor, 1993.
- [2] OPAL TM. Manual de Usuario.