

CPLD para el Control de un Motor Paso a Paso

CPLD to control a stepper motor

Guillermo Tejada Muñoz¹

Facultad de Ingeniería Electrónica y Eléctrica, Universidad Nacional Mayor de San Marcos, Lima, Perú

Resumen— El presente trabajo describe el procedimiento por el cual dentro de un CPLD han sido insertados los circuitos lógicos secuenciales necesarios para hacer posible el funcionamiento de un motor paso a paso en secuencias de paso completo, medio paso, en sentidos de giro horario y antihorario y para diferentes velocidades. El trabajo es de utilidad para los profesionales y estudiantes de Ingeniería electrónica porque metodológicamente se diseñan circuitos lógicos secuenciales y se dan las pautas necesarias para el uso de los CPLDs.

Abstract— This paper describes the process by which within a CPLD have been inserted sequential logic circuits necessary to enable the operation of a stepper motor in full step sequences, half step, in directions of clockwise and counterclockwise and different speeds. The work is useful for professionals and students of electronic engineering because methodologically sequential logic circuits are designed and are given the necessary guidelines for the use of CPLDs.

Palabras Claves— PLD, CPLD, motor paso a paso, circuitos lógicos secuenciales, Quartus II, EPM7128SLC84-7, Altera University Program UP2 Education.

Keywords.— PLD, CPLD, stepper motor, sequential logic circuits, Quartus II, EPM7128SLC84-7 Altera University Program UP2 Education.

I. INTRODUCCIÓN

Un CPLD es un dispositivos cada vez más utilizado dentro de los equipos electrónicos, por ese motivo ni los docentes ni los estudiantes de la carrera de Ingeniería Electrónica debemos estar ajenos al manejo apropiado de estos dispositivos. Un CPLD se suministra al usuario sin que su

circuitería interna este configurada para cumplir una determinada función, la configuración del circuito dentro del CPLD lo especifica el diseñador según sus necesidades y puede ser reconfigurada cuantas veces lo crea necesario.

Por otro lugar, un motor paso a paso resulta de mucho interés para ser controlado debido a que es una buena alternativa en cualquier lugar donde se requiere movimientos con precisión. Puede ser utilizado con ventaja en aplicaciones donde se necesite un control en la rotación angular, velocidad, posición y sincronismo. Algunos ejemplos de aplicaciones se encuentran en: robótica, impresoras, plotters, equipo de oficina, controladores de posición en los discos duros, equipos médicos, fax, máquinas, automóviles y muchos más

En el presente trabajo, se hace una revisión teórica de los motores paso a paso y de los CPLD, principalmente el tipo de CPLD utilizado en el estudio, se detalla la metodología para diseñar los circuitos secuenciales que han sido ensamblados dentro del CPLD, también los procedimientos para programar el CPLD, el cual fue exitosamente probado al controlar el motor de paso para diversos tipos de funcionamiento. También se recomiendan algunos sitios de interés para fortalecer el aprendizaje en el manejo de la herramienta de software que programa al CPLD.

A. Motores Paso a Paso

Los motores paso a paso pueden verse como motores eléctricos sin conmutadores. Típicamente, todos los devanados del motor son parte del estator, y el rotor es un imán permanente o, en el caso de motores de reluctancia variable, es un bloque dentado de algún material magnéticamente blando.

Las conmutaciones deben ser manejadas externamente por un controlador electrónico, y típicamente, los motores y controladores están diseñados de modo que el motor puede mantenerse en cualquier posición fija así como girar de una posición a otra.

¹ Guillermo Tejada Muñoz, E-mail: gtejadam@unmsm.edu.pe

La mayoría de los motores paso a paso conocidos se pueden hacer avanzar a frecuencias de audio, lo que les permite girar muy velozmente. Con un controlador apropiado, se los puede hacer arrancar y detenerse en un instante en posiciones controladas [1].

Los motores paso a paso tienen un comportamiento del todo diferente al de los motores de corriente continua. En primer lugar, no giran libremente por sí mismos. Los motores paso a paso, como lo indica su nombre, avanzan girando por pequeños pasos. También difieren de los motores de CC en la relación entre velocidad y torque, un parámetro que también es llamado "par motor" y "par de giro". Los motores de CC no son buenos para ofrecer un buen torque a baja velocidad sin la ayuda de un mecanismo de reducción. Los motores paso a paso, en cambio, trabajan de manera opuesta: su mayor capacidad de torque se produce a baja velocidad.

Los motores paso a paso tienen una característica adicional: el torque de detención (que se puede ver mencionado también como "par de detención", e incluso par/torque "de mantenimiento"), que no existe en los motores de CC. El torque de detención hace que un motor paso a paso se mantenga firmemente en su posición cuando no está girando. Esta característica es muy útil cuando el motor deja de moverse y, mientras está detenido, la fuerza de carga permanece aplicada a su eje. Se elimina así la necesidad de un mecanismo de freno.

Si bien es cierto que los motores paso a paso funcionan controlados por un pulso de avance, el control de un motor paso a paso no se realiza aplicando en directo este pulso eléctrico que lo hace avanzar. Estos motores tienen varios bobinados que, para producir el avance de ese paso, deben ser alimentados en una adecuada secuencia. Si se invierte el orden de esta secuencia, se logra que el motor gire en sentido opuesto. Si los pulsos de alimentación no se proveen en el orden correcto, el motor no se moverá apropiadamente. Puede ser que zumbe y no se mueva, o puede ser que gire, pero de una manera tosca e irregular.

Esto significa que hacer girar un motor paso a paso no es tan simple como hacerlo con un motor de corriente continua, se requiere un circuito de control, que será el responsable de convertir las señales de avance de un paso y sentido de giro en la necesaria secuencia de energización de los bobinados.

Un motor paso a paso se define por los siguientes parámetros básicos:

1) Voltaje, tienen una tensión eléctrica de trabajo, este valor viene impreso en su

carcasa o por lo menos se especifica en su hoja de datos. Algunas veces puede ser necesario aplicar un voltaje superior para lograr que un determinado motor cumpla con el torque deseado, pero esto producirá un calentamiento excesivo y/o acortará la vida útil del motor.

- 2) Resistencia Eléctrica, determinará la corriente que consumirá el motor, y su valor afecta la curva de torque del motor y su velocidad máxima de operación.
- 3) Grados por paso, este es el factor más importante al elegir un motor paso a paso para un uso determinado. Este factor define la cantidad de grados que rotará el eje para cada paso completo. Una operación de medio-paso o semi-paso (half step) del motor duplicará la cantidad de pasos por revolución al reducir la cantidad de grados por paso. Cuando el valor de grados por paso no está indicado en el motor, es posible contar a mano la cantidad de pasos por vuelta, haciendo girar el motor y sintiendo por el tacto cada "diente" magnético. Los grados por paso se calculan dividiendo 360 (una vuelta completa) por la cantidad de pasos que se contaron. Las cantidades más comunes de grados por paso son: $0,72^\circ$, $1,8^\circ$, $3,6^\circ$, $7,5^\circ$, 15° y hasta 90° . A este valor de grados por paso usualmente se le llama la resolución del motor. En el caso de que un motor no indique los grados por paso en su carcasa, pero sí la cantidad de pasos por revolución, al dividir 360 por ese valor se obtiene la cantidad de grados por paso. Un motor de 200 pasos por vuelta, por ejemplo, tendrá una resolución de $1,8^\circ$ por paso [2].
- 4) Precisión del Paso, indica el porcentaje de error entre la posición real del eje y la posición teórica que tendría que tener.
- 5) Par de Detención, generan, sin alimentación, un par debido al imán permanente del motor que es máximo entre dos pasos. El valor dado es el valor máximo de este.
- 6) Par de Mantenimiento, indica el par máximo que podremos obtener del motor. El par depende de su construcción interna, la forma de los dientes y de la intensidad de campo del imán permanente.
- 7) Frecuencia de Resonancia, si sujetamos un motor paso a paso en una base rígida y lo acoplamos también de forma rígida a una carga y lo hacemos funcionar a una frecuencia cercana a la de resonancia del

sistema, tendremos que el motor perderá el paso. Este problema se puede tratar desde tres puntos de vista [3]:

- a) Controlar la resonancia en el mecanismo: Utilizar acoplamientos y soportes del motor basados en elastómeros que absorban las frecuencias de resonancia.
- b) Controlar la resonancia en el driver a bajo nivel: Un motor resonando induce una tensión alterna en las bobinas. Mediante una bobina externa podríamos eliminar esta frecuencia indeseable.
- c) Controlar la resonancia en alto nivel: El driver evita estas frecuencias.

Entre las ventajas de los motores paso a paso podemos citar las siguientes:

- 1) El ángulo de rotación es proporcional a los pulsos de entrada.
- 2) El motor tiene torque máximo cuando está enclavado (si las bobinas están energizadas).
- 3) Posición precisa y la repetición de los movimientos tienen una exactitud de 3 a 5% del paso y este error no es acumulativo de un paso a otro.
- 4) Excelente respuesta en el arranque, parada y reversa.
- 5) Muy fiable dado que no existe contacto de escobillas en el motor.
- 6) El motor responde a pulsos de entrada digitales, lo que permite un control de lazo abierto, haciendo un control más simple y barato.
- 7) Es posible lograr una velocidad de rotación muy baja en forma sincrónica con carga acoplada directamente sobre el eje.
- 8) Pueden tener un gran rango de velocidades de rotación, dado que la misma es proporcional a la frecuencia de los pulsos de entrada.

Mientras que entre sus desventajas podemos citar:

- 1) Puede ocurrir un fenómeno de resonancia si el motor no es controlado adecuadamente.
- 2) Muy difícil de operar a altas velocidades.

B. Aplicaciones y Tipos de Motores Paso a Paso

Un motor paso a paso puede ser una buena alternativa en cualquier lugar donde se requiera controlar el movimiento. Pueden ser utilizados con ventaja en aplicaciones donde se necesite un control en la rotación angular, velocidad, posición y sincronismo. Debido a la inherente ventajas enumeradas anteriormente, motores paso a paso

han encontrado su lugar en muchas aplicaciones diferentes. Algunos de estos incluyen impresoras, plotters, equipo de oficina, controladores de posición en los discos duros, equipos médicos, fax, máquinas, automóviles y muchos más [4].

Existen tres tipos básicos de motores paso a paso, ellos son: Reluctancia Variable, Imán Permanente e Híbrido.

- 1) Reluctancia Variable (RV), el motor paso a paso de Reluctancia Variable (RV), cuya sección transversal se muestra en la Fig. 1, es probablemente el más fácil entender desde un punto de vista estructural. La sección transversal es de un típico motor paso a paso RV. Este tipo de motor consiste de un rotor de hierro dulce multi dentado y un estator con bobinas devanadas. Cuando los devanados del estator se alimentan con corriente continua los polos se llegan a magnetizar. La rotación se produce cuando los dientes del rotor son atraídos a los polos del estator magnetizados.

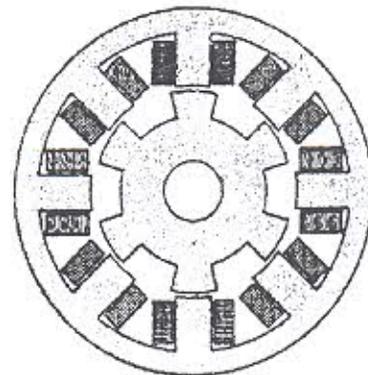


Fig. 1. Típico motor de reluctancia variable. Cada uno de los ocho polos del estator llevan un devanado concéntrico, mientras los seis polos del rotor no tienen devanado o imanes.

- 2) Imán Permanente (IP), el motor de paso a paso de imán permanente (IP), que se muestra en la Fig. 2, es de bajo coste y baja resolución con ángulos de paso típico de $7,5^\circ$ a 150° . Los motores IP como su nombre lo indica tiene imanes permanentes añadido a la estructura del motor. El rotor ya no tiene dientes como con el motor VR, en cambio está magnetizados con polos alternados norte y sur situados en una línea recta paralela al eje del rotor. Estos polos del motor magnetizados proporcionan un aumento en la intensidad del flujo magnético y debido a esto el motor IP muestra características mejoradas de torque en comparación con el tipo RV.

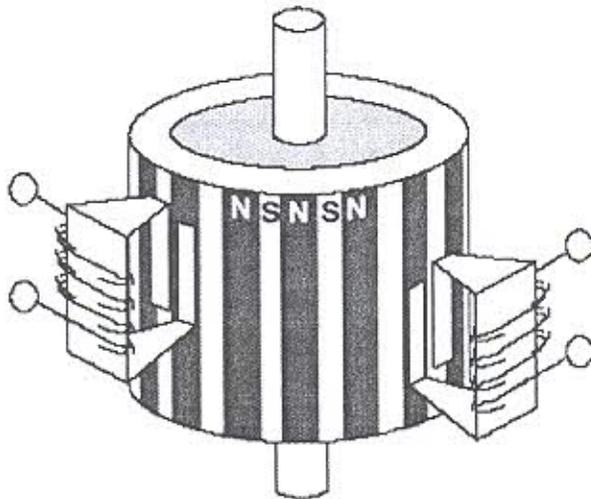


Fig. 2. Motor de imán permanente

- 1) Híbrido (HB), el motor Paso a Paso Híbrido (HB) híbrido, cuya sección transversal se muestra en la Fig. 3, es más caro que el motor paso a paso IP pero proporciona un mejor rendimiento con respecto a la resolución de cada paso, el torque y la velocidad. Típicamente presentan ángulos por cada paso en el rango de $3,6^\circ$ hasta 90° grados. El motor paso a paso híbrido combina las mejores características de los motores paso a paso IP y RV. El rotor es multi-dentado como en el motor RV y contiene un imán magnetizado axialmente concéntrico alrededor de su eje. Los dientes en el rotor proporcionan e igualan el flujo magnético, aumentando las características dinámicas del torque en comparación con los motores de paso RV e IP.

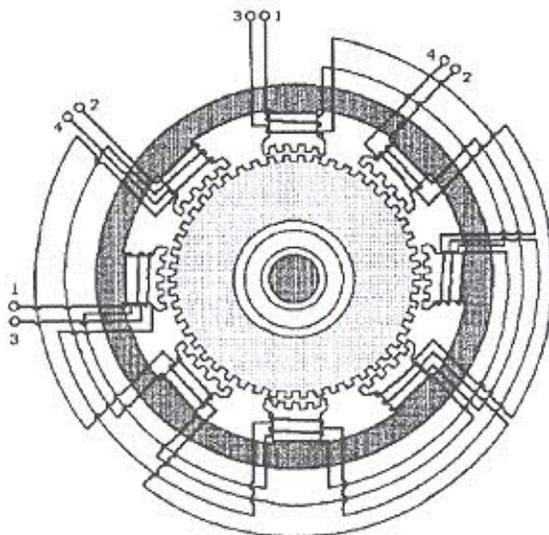


Fig. 3. Sección transversal de un motor híbrido, mostrando los segmentos del núcleo del rotor magnético y los polos del estator con su diagrama de cableado.

Los dos tipos más comúnmente usados de motores paso a paso son el de imán permanente y el híbrido. Si un diseñador no está seguro de cuál de los dos tipos se adapta a sus aplicaciones, entonces antes de seleccionar al híbrido deberá intentar evaluar como primera opción al de tipo IP ya que normalmente es varias veces menos caro que el híbrido. También existen algunos motores paso a paso especiales, uno de ellos es el motor de imanes de disco mostrado en la Fig. 4, el cual está diseñado como un disco con imanes de tierras raras. Este tipo de motor tiene algunas ventajas, tales como una baja inercia y un flujo magnético optimizado con ningún acoplamiento entre los dos devanados del estator, estas cualidades son esenciales en algunas aplicaciones [5].

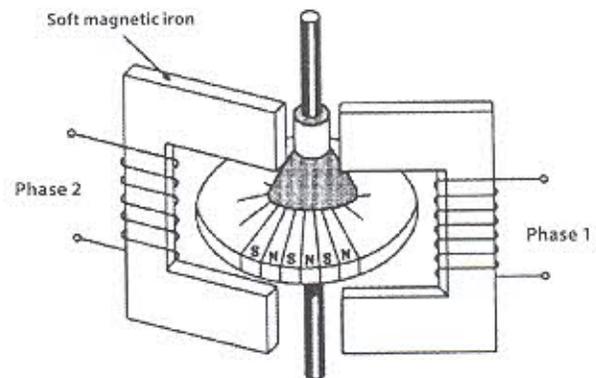


Fig. 4. Motor de imanes de disco

En las referencias [6] y [7] se pueden encontrar simulaciones en 3D que explican didácticamente la construcción y el funcionamiento de un motor paso a paso.

C. Dispositivos Lógicos Programables Complejos (CPLD)

Uno de los más trascendentales avances en la electrónica digital ha sido la introducción de los dispositivos lógicos programables (PLDs). Antes del desarrollo de los PLDs, los circuitos digitales se construyeron en integración a pequeña escala (SSI) e integración a media escala (MSI), estos dispositivos contenían puertas lógicas y otros circuitos digitales. Las funciones se determinaban en el momento de la fabricación y no podían ser cambiadas. Además, si un diseñador deseaba un dispositivo con una función en particular que no estaba en la lista de un fabricante, estaba obligado a utilizar varios dispositivos, algunos de los cuales podía contener funciones que no necesitaba, desperdiciando así el espacio en la tarjeta electrónica y tiempo de diseño. Los dispositivos lógicos programables proporcionan una solución a estos problemas.

Un PLD se suministra al usuario sin ninguna función lógica programada en absoluto, éstas son solo especificadas por el diseñador según sus necesidades. Dado que por lo general varias funciones se pueden combinar en el diseño interno del chip, el espacio utilizado en él puede reducirse. Además, si un diseño necesita ser cambiado, un PLD se puede reprogramar con la nueva información del diseño generalmente sin sacarlo del circuito. PLD es un término genérico, hay una amplia variedad de tipos de PLD, incluye: PAL (Arreglo Lógico Programable), GAL (Arreglo Lógico Genérico), EPLD (PLD Borrable), CPLD (PLD Complejo), FPGA (Arreglo de Puertas Programable en el Campo), así como otros.

Nos centraremos en CPLDs, aunque la terminología puede variar un poco, vamos a utilizar el término CPLD para referirnos a un dispositivo con varias secciones de PLDs interconectados en un solo chip [8].

La Fig. 5 Muestra el diagrama de bloques de CPLD (CPLD) MAX7000S de Altera. Un dispositivo de este tipo ha sido instalado en la Tarjeta de Desarrollo de Altera UP-1 que se ha utilizado en el presente estudio.

La estructura principal del MAX7000S es una serie de bloques de Arreglo Lógico (LABS), unidos por una matriz de interconexión programable (PIA).

Cada LAB tiene un grupo de 16 macroceldas que pueden distribuir, prestar o pedir prestado términos productos lógicos entre sí. Un único LAB tiene similares capacidad de un PLD de baja densidad, por lo que un CPLD como los MAX7000S puede ser pensado como una matriz de PALs o GALs interconectados entre si dentro de un solo Chip.

EPM7128S tiene 8 LABs, para un total de 8 * 16 = 128 macroceldas. Sin embargo, éstas no están disponibles para el usuario como E/S, el número de pines de E/S disponibles depende del tipo de paquete del dispositivo. Fig. 5, indica que cada LAB del dispositivo MAX7000S tiene de 6 a 16 pines de E/S. Para un EPM7128S en un encapsulado de 160 pines (paquete PQFP), hay 12 pines de E/S por LAB, para un total de 96 pines disponibles. Para el mismo dispositivo pero en un encapsulado PLCC de 84-pin, hay sólo 8 pines de I/O por LAB, para un total de 64 pines.

En la práctica, si un EPM7128SLC84 es programado dentro de la tarjeta electrónica hay solo 60 pines de I/O disponibles, cuando 4 pines son requeridos para la interfaz de programación. Las macroceldas que no están conectadas a los pines de del usuario de I/O pueden solo ser usadas para "lógica sepultada" o lógica que es interna solo al chip.

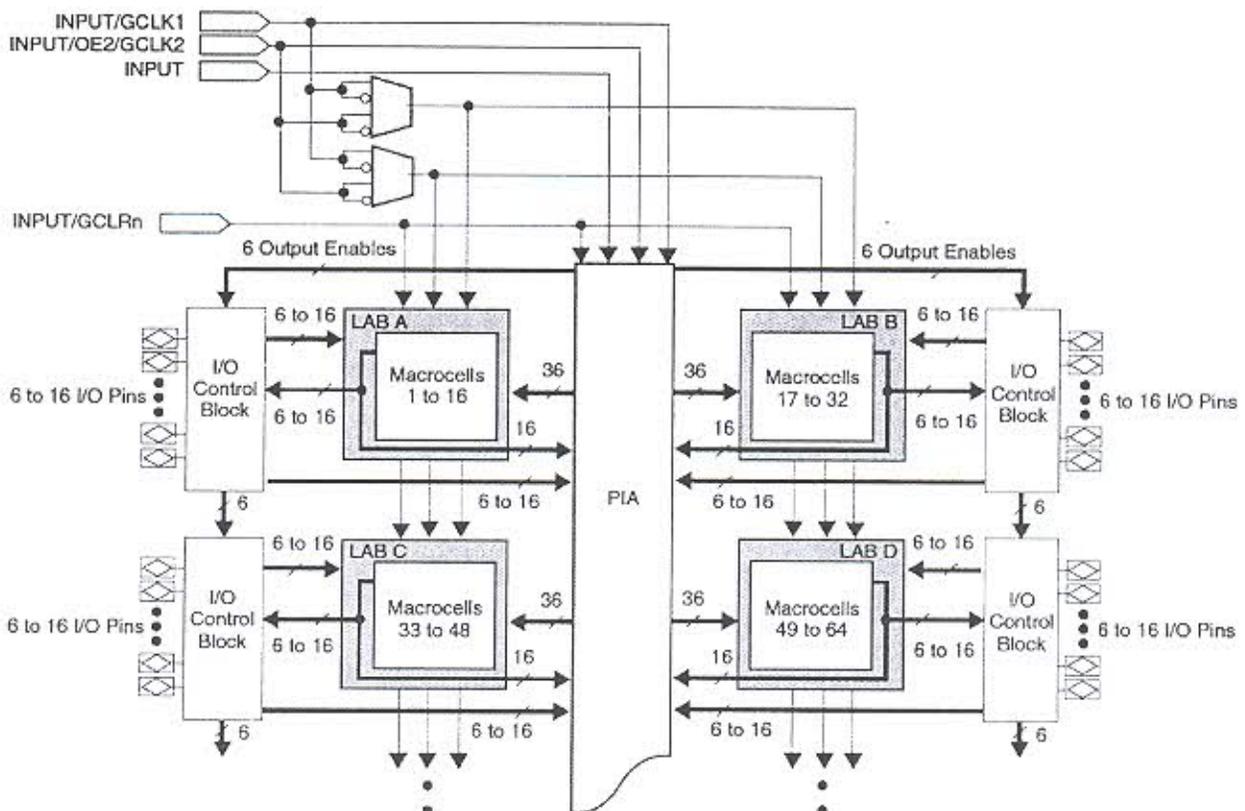


Figure 5. Diagrama de bloques del CPLD de Altera MAX 7000E and MAX 7000S

Tal como está implícito en la Fig. 5, todos los pines de I/O conectan hacia y desde sus asociados LABs un bloque de control de I/O (un circuito que controla conmutadores de tres estados de las señales en un pin de I/O). Las señales en los pines de I/O pueden también conectarse directamente a la PIA, donde están disponibles para su uso en otros LABs. Dieciséis líneas conectan las salidas de las macroceldas de cada LAB a la PIA, de nuevo para usarse por todo el dispositivo. El PIA se comunica a cada LAB a través de 36 líneas de productos para proporcionar conexiones desde otros LABs.

La familia MAX7000S tiene cuatro pines que pueden ser configurados como señales de control o entradas. *GCLK1* es un reloj global que es común a todas las macroceldas del dispositivo y puede ser usado como un reloj síncrono para todos los registros. *OEI* es un habilitador de salida que puede globalmente activar o desactivar la salida de los *tristates* de las macroceldas. *GCLRn* es un limpiador (clear) global que se activa en bajo. El cuarto pin de control puede ser configurado como una entrada, como pueden hacerlo los otros tres pines, o como un segundo clock global (*GCLK2*) o habilitador de salida (*OE2*). Si las funciones de control no son usadas, estos pines suman cuatro entradas al total disponible. Esta asignación puede ser realizada por el software de configuración utilizada durante el proceso de diseño.

La Fig. 6, muestra una macrocelda del dispositivo MAX7000S. La macrocelda es similar a un GAL o PAL en el sentido que provee funciones suma de productos, existe la opción de seleccionar estas salidas activas en alto o en bajo y seleccionar salidas de registros o combinacionales. Los registros de salidas pueden ser sincronizadas con uno de dos relojes globales o por un término producto desde la matriz AND.

La salida del registro puede ser borrado globalmente o mediante un término producto, mientras que puede ser preestablecido mediante un término producto. La macrocelda tiene cinco términos producto dedicados, esto es generalmente suficiente para implementar la mayoría de funciones lógicas, si son necesarios más términos productos, éstos pueden ser proveídos mediante expansores de lógicos compartida o expansores de lógica paralela.

Los expansores de lógica compartida no añaden más términos productos a una macrocelda, ellos hacen que la programación del LAB sea más eficiente al permitir que un término producto pueda ser utilizada en varias macroceldas del mismo LAB, como se observa en la Fig. 6, la salida del término producto se invierte logicamente y se retroalimenta a cualquiera de las entradas de otros términos productos. Dado que hay 16 macroceldas por LAB, el expansor de lógica compartida tiene un máximo de 16 términos productos.

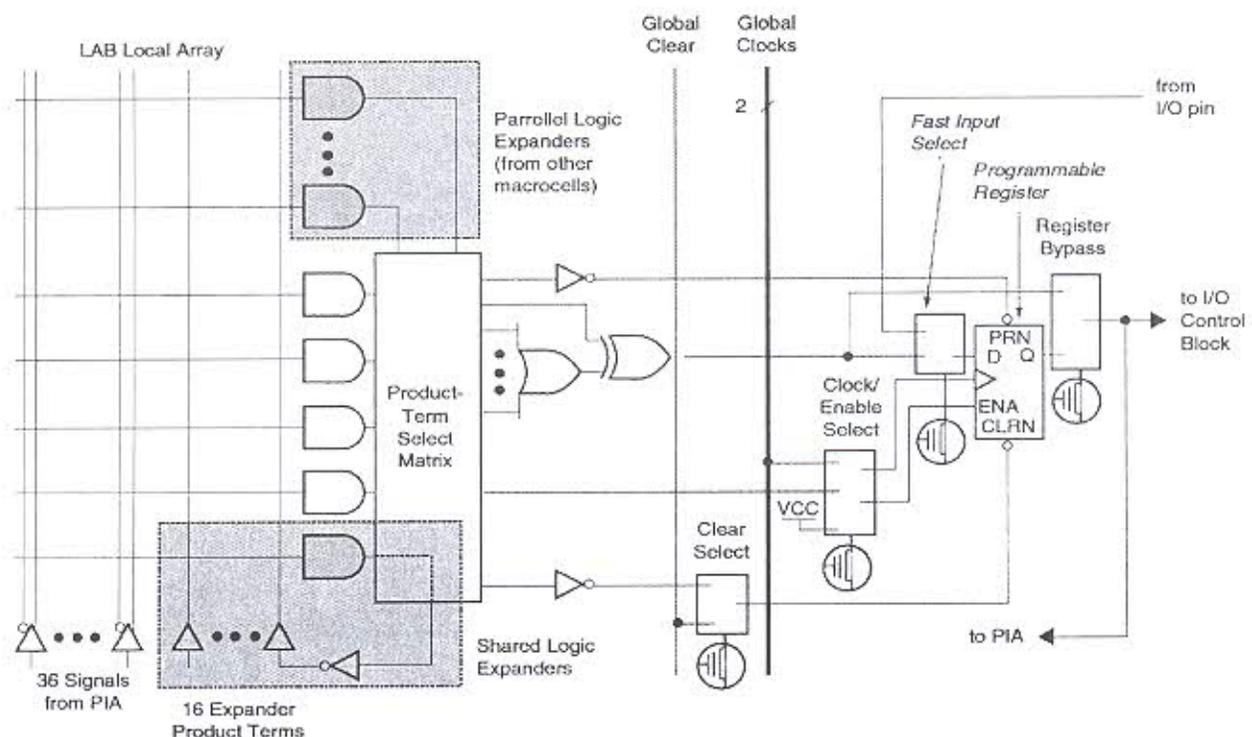


Fig. 6. Macrocelda de un CPLD MAX7000E y MAX7000S

Los expansores de lógica paralela permiten pedir prestado un máximo de 15 términos producto de otras macroceldas vecinas, de cada macrocelda restante pueden prestarse 5 terminos productos que no podrían ser utilizadas por aquellas macroceldas que la prestaron. Las funcionalidad de los expansores lógicos se ejecutan automáticamente mediante el software de programación de los CPLD [9].

Finalmente, podemos citar a grandes fabricantes como de CPLDs como [10]:

- Xilinx
- Altera
- Lattice Semiconductor
- Atmel
- Cypress Semiconductor

Mientras que ente las herramientas de software utilizadas para su programación podemos citar a:

- Altium Designer - Utilidad de diseño de Altera, Xilinx, Actel, y otras familias FPGA/CPLD
- Quartus - Utilidad de diseño para los dispositivos Altera.
- ISE - Utilidad de diseño para los dispositivos Xilinx.
- ispLEVER - Paquete de aplicaciones para los dispositivos Lattice.
- ModelSim - Simulador de diferentes lenguajes de programación hardware HDL/Verilog.

II. MÉTODOS Y RESULTADOS

Para encontrar a los circuitos secuenciales que controlan al motor paso a paso tanto en su funcionamiento de paso completo y de medio paso se utilizó la metodología para el diseño de circuitos secuenciales síncronos. Conociendo la secuencia lógica que deben tener las salidas del circuito, se encontró el Diagrama de Estados, la Tabla de Estados, la Tabla de Transición de Estados, los Mapas de Karnaugh de los circuitos combinacionales que activan las entradas de los Flip Flops y las salidas.

A. Método para Encontrar el Circuito Lógico para un Funcionamiento del Motor en Paso Completo

Para un funcionamiento de paso completo el Diagrama de Estados y la Tabla de Estados tipo Moore son mostradas en la Fig. 7 y Tabla I respectivamente. Los arcos que tienen el valor lógico 1 ($X=1$) corresponde al funcionamiento del motor en sentido horario, mientras que los arcos con valor lógico 0 ($X=0$) corresponde a un sentido antihorario.

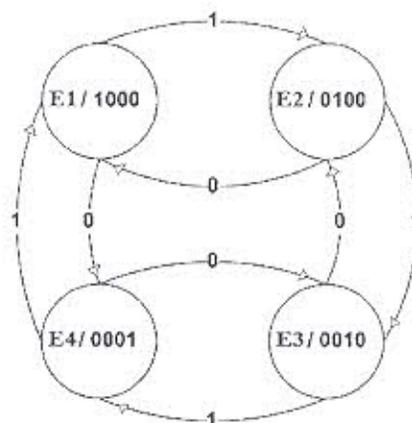


Fig. 7. Diagrama de Estados para el funcionamiento del motor a paso completo

TABLA I
TABLA DE ESTADOS

| Estados actuales | ENTRADA (X) | | SALIDAS ABCD |
|------------------|--------------------|--------------------|-----------------|
| | 0 | 1 | |
| | Estados siguientes | Estados siguientes | |
| E1 | E4 | E2 | 1000 |
| E2 | E1 | E3 | 0100 |
| E3 | E2 | E4 | 0010 |
| E4 | E3 | E1 | 0001 |

Como se tienen cuatro estados, entonces se tiene que utilizar dos Flip Flops (FFs), la Tabla II muestra la Tabla de Transición de Estados y se obtiene al sustituir a los estados de la Tabla I con los siguientes valores arbitrarios: E1=00, E2=01, E3=10, E4=11. Observar que $y1$ e $y2$ representa a los estados presentes de los Flip Flops, es decir a $Q_i(t)$, mientras que $Y1$ e $Y2$ representan a los estados siguientes de los Flip Flops, es decir a $Q_i(t+1)$.

TABLA II
TABLA DE TRANSICIÓN DE ESTADOS

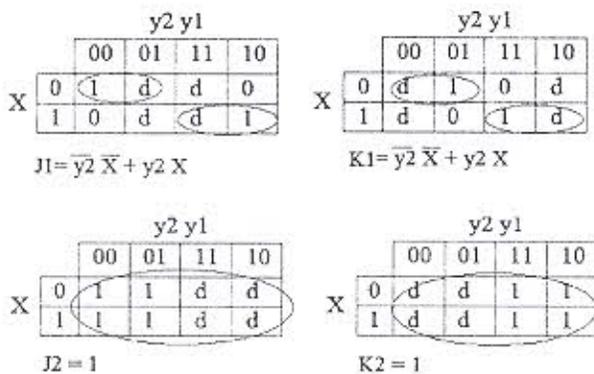
| y1 y2 | ENTRADA (X) | | SALIDAS ABCD |
|-------|---------------|-------|-----------------|
| | 0 | 1 | |
| | Y1 Y2 | Y1 Y2 | |
| 00 | 11 | 01 | 1000 |
| 01 | 00 | 10 | 0100 |
| 10 | 01 | 11 | 0010 |
| 11 | 10 | 00 | 0001 |

En la Tabla III se ha listado todas las posibilidades de cambio en las salidas de un FF JK de acuerdo al estado lógico en que se encuentren sus entradas J y K. Considere t como el tiempo antes de la activación de la entrada del reloj de los FF y $(t+1)$ luego de su activación.

TABLA III
ENTRADAS PARA UN FF JK

| | | | |
|-------------------|---------------------|----------|----------|
| $Q_i(t)$ y_i | $Q_i(t+1)$ Y_i | $J_i(t)$ | $K_i(t)$ |
| 0 | 0 | 0 | X |
| 0 | 1 | 1 | X |
| 1 | 0 | x | 1 |
| 1 | 1 | x | 0 |

De acuerdo a la Tabla II y Tabla III, se obtienen los mapas de Karnaugh de los circuitos combinacionales que excitan a las entradas de los Flip Flops, como se muestra a continuación:



Las salidas A, B, C, D solo dependen de los estados presentes entonces de la Tabla II se puede obtener la siguiente tabla lógica:

| $y_1 y_2 y_3$ | SALIDAS ABCD |
|---------------|-----------------|
| 0 0 0 | 1 0 0 0 |
| 0 0 1 | 1 1 0 0 |
| 0 1 0 | 0 1 0 0 |
| 0 1 1 | 0 1 1 0 |
| 1 0 0 | 0 0 1 0 |

De donde se obtiene:

$$A = \overline{y_1} \overline{y_2}$$

$$B = y_1 \overline{y_2}$$

$$C = y_1 y_2$$

$$D = y_1 y_2$$

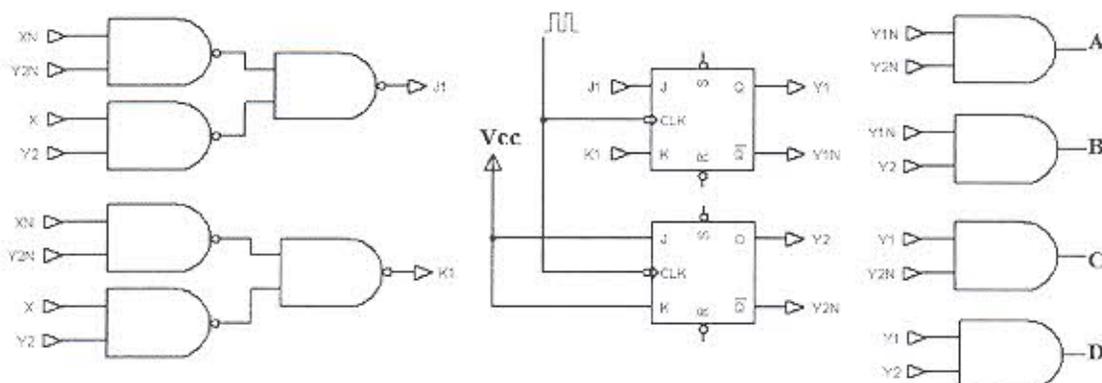


Fig. 8. Circuito lógico para el control del motor en paso completo

Finalmente, el circuito lógico de acuerdo a las expresiones deducidas se muestra en la Fig. 8.

B. Método para Encontrar el Circuito Lógico para un Funcionamiento del Motor en Medio Paso.

Para un funcionamiento de medio paso el Diagrama de Estados y la Tabla de Estados tipo Moore son mostradas en la Fig. 9 y Tabla IV respectivamente.

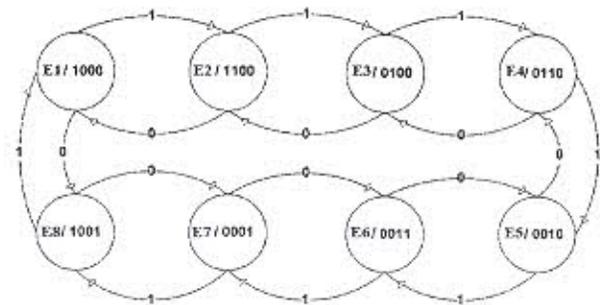


Fig. 9. Diagrama de Estados para el funcionamiento del motor en medio paso

TABLA IV
TABLA DE ESTADOS

| Estados actuales | ENTRADA (X) | | SALIDAS ABCD |
|------------------|-------------------------|-------------------------|-----------------|
| | Estados siguientes 0 | Estados siguientes 1 | |
| E1 | E8 | E2 | 1000 |
| E2 | E1 | E3 | 1100 |
| E3 | E2 | E4 | 0100 |
| E4 | E3 | E5 | 0110 |
| E5 | E4 | E6 | 0010 |
| E6 | E5 | E7 | 0011 |
| E7 | E6 | E8 | 0001 |
| E8 | E7 | E1 | 1001 |

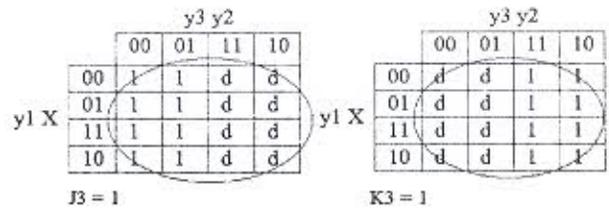
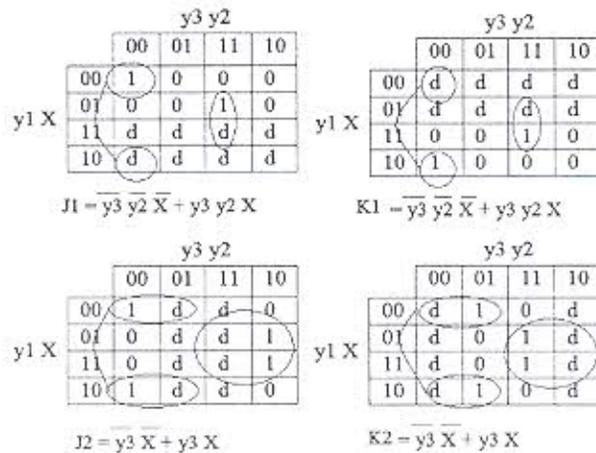
Como se ha obtenido ocho estados, entonces se tiene que utilizar tres Flip Fflops (FFs). La Tabla V es la Tabla de Transición de Estados, se obtiene al sustituir en la Tabla IV los estados con los siguientes valores arbitrarios: E1=000, E2=001, E3=010, E4=011, E5=100, E6=101, E7=110,

E8=111. Observar que y_1, y_2, y_3 representan a los estados presentes de los Flip Flops, es decir a $Q_i(t)$, mientras que Y_1, Y_2, Y_3 representan a los estados siguientes de los Flip Flops, es decir a $Q_i(t+1)$.

TABLA V
TABLA DE TRANSICIÓN DE ESTADOS

| y1 y2 y3 | ENTRADA (X) | | SALIDAS ABCD |
|----------|-------------|----------|-----------------|
| | 0 | 1 | |
| | Y1 Y2 Y3 | Y1 Y2 Y3 | |
| 000 | 111 | 001 | 1000 |
| 001 | 000 | 010 | 1100 |
| 010 | 001 | 011 | 0100 |
| 011 | 010 | 100 | 0110 |
| 100 | 011 | 101 | 0010 |
| 101 | 100 | 110 | 0011 |
| 110 | 101 | 111 | 0001 |
| 111 | 110 | 000 | 1001 |

De acuerdo a la Tabla V y la Tabla III, se obtienen los mapas de Karnaugh de los circuitos combinacionales que excitan a las entradas de los FFs.



Las salidas A, B, C, D solo dependen de los estados presentes, entonces de la Tabla V se puede obtener la siguiente tabla lógica:

| y1 y2 y3 | SALIDAS ABCD |
|----------|-----------------|
| 000 | 1000 |
| 001 | 1100 |
| 010 | 0100 |
| 011 | 0110 |
| 100 | 0010 |
| 101 | 0011 |
| 110 | 0001 |
| 111 | 1001 |

De donde se obtiene:

$$A = \overline{y_1} \overline{y_2} + y_1 y_2 y_3$$

$$B = \overline{y_1} y_3 + \overline{y_1} y_2$$

$$C = \overline{y_1} y_2 y_3 + y_1 \overline{y_2}$$

$$D = y_1 y_2 + y_1 y_3$$

Finalmente, el circuito lógico de acuerdo a las expresiones deducidas se muestra en la Fig. 10.

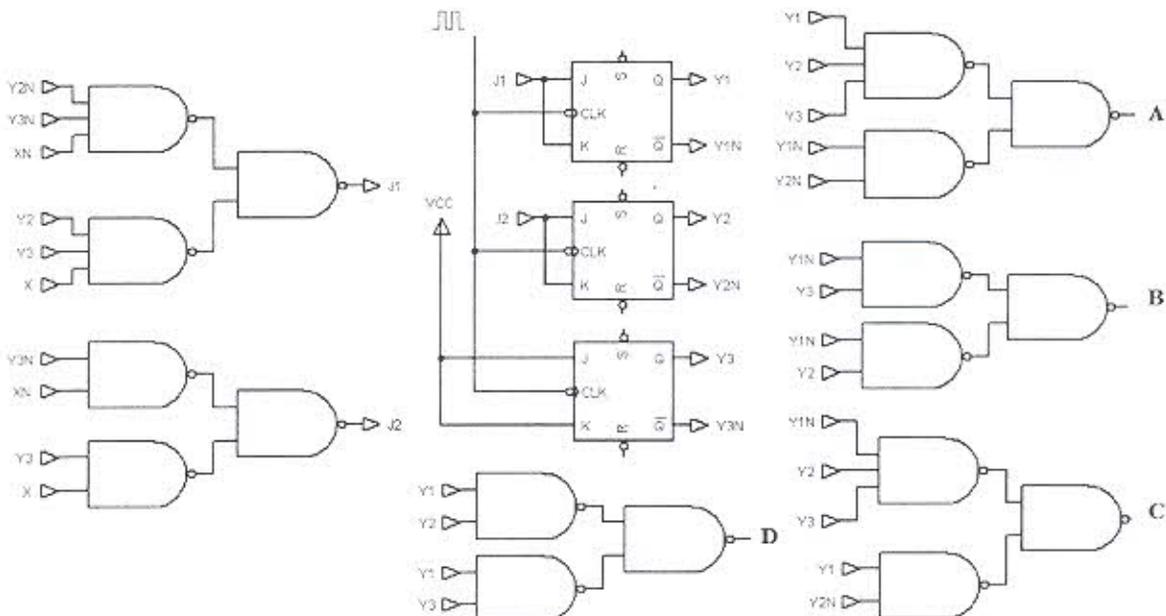


Fig. 10. Circuito lógico para el control del motor en medio paso

C. Método para Programar al CPLD

El CPLD EPM7128SLC84-7 fue el utilizado, este dispositivo se encuentra dentro de la tarjeta de desarrollo *University Program UP2* de Altera. La cual también posee un FPGA. Para la configuración interna del CPLD se utilizó el software de distribución gratuita Quatus II versión 9.1, instalada en una computadora Pentium IV. La conectividad entre la computadora y la tarjeta UP2 se estableció mediante el puerto paralelo. Para el aprendizaje detallado del manejo del QUARTUS se sugiere los videos del sitio de la referencia [11].

Utilizando el editor de esquemáticos del Quartus se dibujaron los circuitos de la Fig. 9 y 10 y otros adicionales. El CPLD luego de su programación fue capaz de generar las señales de salida ABCD que posibilitó al motor operar a paso completo, medio paso y en ambos casos variar también el sentido de giro, así como también variar su velocidad que se consigue al dividir la frecuencia del reloj que alimenta a los circuitos. La Fig. 11, muestra un diagrama de bloques del circuito total ensamblado en el CPLD, el bloque Circuito Paso Completo y el Circuito Medio Paso se corresponden con los circuitos lógicos de las Fig. 9 y 10. El bloque Divisor de Frecuencia está constituido por seis contadores

módulo 16 cada uno conectados en cascada, lo que permite dividir la frecuencia del reloj (25.175Mhz) entre 16777216 veces como máximo. Este bloque permite obtener por Q20, Q21, Q22 Y Q23 frecuencias de 12, 6.0, 3.0 y 1.5Hz respectivamente, cualquiera de las frecuencias puede ser seleccionada mediante la combinación binaria de V1 Y V0 del selector del MUX de 4 a 1, con lo que podemos tener cuatro velocidades diferentes para el motor. El modo de funcionamiento de paso o medio paso es seleccionado mediante el valor lógico de P que selecciona en la salida del MUX 2 a 1 las salidas ABCD del paso completo o la del medio paso.

La Fig. 12, muestra el esquema completo constituido por las etapas de control, aislamiento, potencia y el motor. La etapa de control lo constituye el CPLD quien provee las señales ABCD necesarias para todos los modos de funcionamiento del motor. La etapa de aislamiento entre la de control y potencia está constituida por los optoacopladores 4N33. La etapa de potencia se implementó con transistores 2N2222 con diodos antiparalelo para eliminar los picos transitorios producidos por la carga inductiva del motor. Se utilizó el motor paso a paso NMB PM42L-048-RCC1 de imán permanente, de 48 pasos por rotación (de 7.5° por paso).

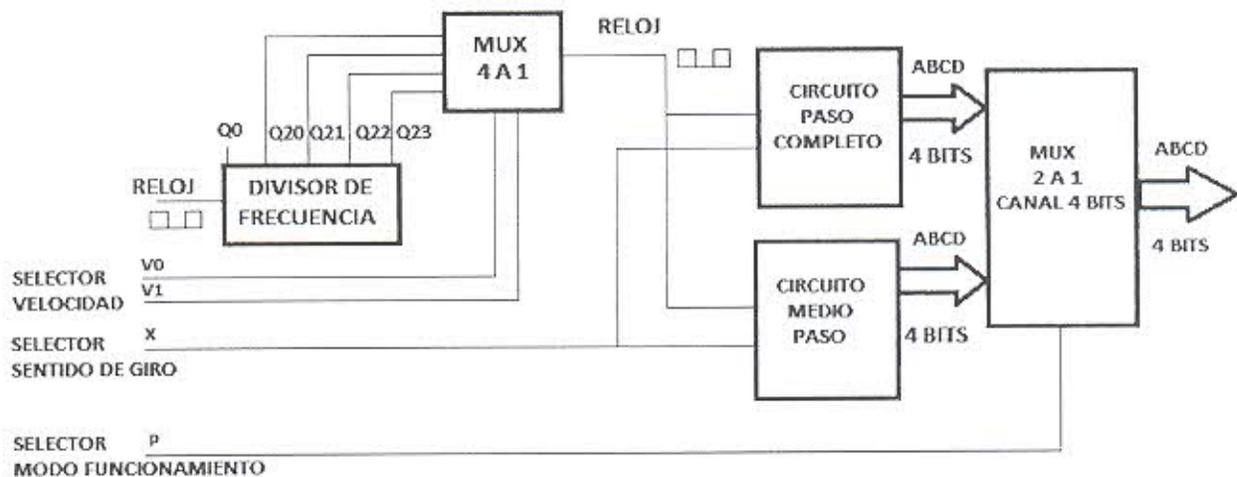


Fig.11. Diagrama de bloques del circuito lógico ensamblado dentro del CPLD

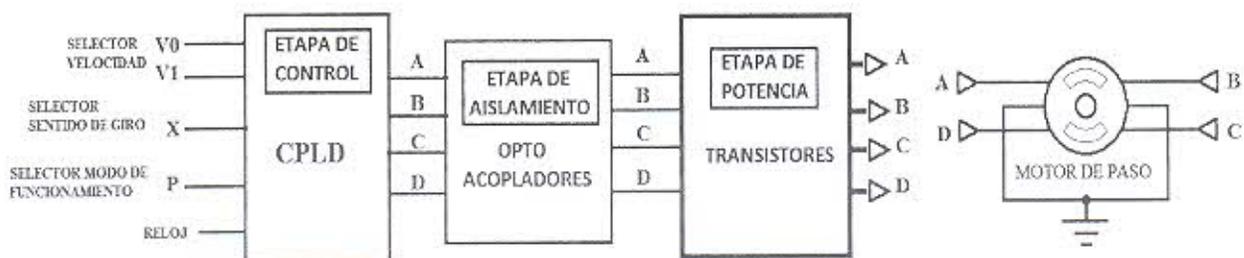


Fig. 12. Diagrama de bloques del circuito que incluye etapa de control, aislamiento, potencia y motor

Finalmente, en la foto de la Fig. 13 se muestra el circuito implementado, se observa la tarjeta de desarrollo UP2 en donde se ubica el dispositivo CPLD EPM7128SLC84-7 de Altera, luego a la izquierda se puede observar la placa de pruebas, de color blanco, en donde se encuentran los optoacopladores 4N33 y los transistores 2N2222 con diodos antiparalelos, más a la izquierda observamos el motor de paso NMB PM42L-048-RCC1 en cuya base hemos colocado un transportador para poder medir la deflexión en grados por cada paso de giro.

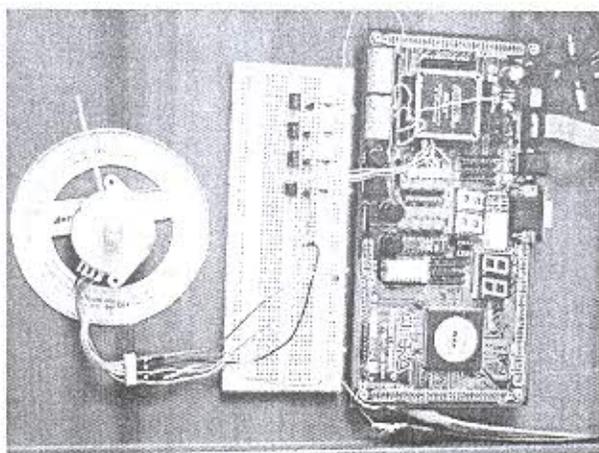


Fig. 13. Fotografía en donde se demuestra el funcionamiento del motor

III CONCLUSIONES

Se ha cumplido con creces los objetivos del estudio que solo consideraba en la propuesta la simulación del circuito, no solo se ha simulado el funcionamiento del circuito con las herramientas de simulación disponible en QUARTUS, sino además se ha logrado demostrar físicamente el funcionamiento del circuito ensamblado dentro del CPLD EPM7128SLC84-7 que ha sido capaz de controlar a un motor en sus modalidades de paso completo, medio paso, con sentido de giro horario y antihorario y con cuatro velocidades distintas, también se ha validado la metodología de diseño para circuitos secuenciales síncronos.

AGRADECIMIENTOS

Agradezco al profesor Victor Alva Saldaña por sus sugerencias y a los alumnos Gustavo Marlo Cuba Escardon y Angel Bastidas Cerazo por el apoyo en la implementación del circuito.

REFERENCIAS

- [1] Douglas W. Jones, Control of Stepping Motors a Tutorial, The University of IOWA Department of Computer Science, <http://homepage.cs.uiowa.edu/~jones/step/>, fecha de consulta: agosto 2012.
- [2] Eduardo J. Carletti, Motores Paso a Paso Características Básicas, http://robots-argentina.com.ar/MotorPP_basico.htm, fecha de consulta: julio 2012.
- [3] MCBtec, Motores Paso a Paso (PAP), <http://www.mcbtec.com/MotorPAPBasico.pdf>.
- [4] Industrial Circuits Application Note Stepper Motor Basics, <http://www.solarbotics.net/library/pdflib/pdf/motorbas.pdf>, fecha de consulta: setiembre 2012.
- [5] Stepper Motor and Its Driver, Reference of EAS 5407 Mechatronics Design Project Spring 2003, fecha de consulta: julio 2012.
- [6] How the Stepper motors are made and how they operate, <http://www.youtube.com/watch?v=MHdz3c6KLrg>, fecha de consulta: noviembre 2012.
- [7] PCB Heaven, http://www.pcbheaven.com/wikipages/How_Stepper_Motors_Work/; fecha de consulta: noviembre 2012.
- [8] Digital Design with CPLD and VHDL, Robert Dueck, Delmar Cengage Learning; 2 edition (June 8, 2004).
- [9] Digital Systems Design with FPGAs and CPLDs, Ian Grout, Newnes; 1 edition (April 9, 2008).
- [10] Complex programmable logic device; http://en.wikipedia.org/wiki/Complex_programmable_logic_device; fecha de consulta: enero 2013.
- [11] Diseño Lógico con QUARTUS; http://www.youtube.com/watch?v=0jSD0C_Kmug; fecha de consulta: abril 2012.

