

ESTRUCTURA Y PROGRAMACIÓN DE ARREGLOS LÓGICOS GENÉRICOS (GAL'S) CON APLICACIÓN AL CONTROL DE UN BRAZO MECÁNICO

J. G. Mateos, J.M. Villegas, A. Hernández, E. Sánchez
jmateos@proton.ucting.udg.mx, jvillega@proton.ucting.udg.mx,
acorvera@proton.ucting.udg.mx, esanchez@proton.ucting.udg.mx

Universidad de Guadalajara, CUCEI
Departamento de Electrónica

Avenida Revolución # 1500, Puerta 10 y/o
Boulevard Marcelino García Barragán #1500
C.P. 44840, Guadalajara, Jal. México,
Tel/FAX: (013) 619-8471

RESUMEN

El trabajo que se presenta tiene como objetivo mostrar el funcionamiento del arreglo lógico genérico GAL (Generic Array Logic) aplicado al control de un brazo mecánico. Se diseña un circuito secuencial y se estructura la programación del circuito integrado GAL16V8, sus terminales de salida se usan para controlar el movimiento del brazo, a través de tres motores de 5 Vdc. Se producen así, tres grados de libertad, que son; pinzas de sujeción, brazo y soporte, respectivamente. El movimiento se activa con engranes dispuestos en un mecanismo "planetario". Los motores se energizan con una interface que se implementa con transistores de potencia. En general, los GAL, son circuitos integrados de lógica programable que utilizan tecnología MOSFET, éstos cuentan con un arreglo matricial de fusibles que inicialmente conectan todas las terminales de entrada del circuito, con todas las compuertas de un arreglo AND-OR. Además contiene una sección denominada Macroelda Lógica de Salida OLMC (Output Logic Macro Cell), que es una sección del GAL que incluye a los elementos necesarios para configurar sus salidas, de cuatro maneras lógicas distintas. Con los GAL's se proporciona una herramienta versátil en la solución de sistemas lógicos combinatorios o secuenciales. Los circuitos que se generan con esta técnica de diseño son pequeños y consecuentemente económicos.

ABSTRACT

The objective of the work that is presented is to show the operation of the generic array logic (GAL) arrangement applied to the control of a mechanical arm, a circuit is designed secuencial and the programming of the integrated circuit GAL16V8 is structured, their output terminals are used in order to control the movement of the arm, through three 5 Volts of C.D. motors, they take place so, three degrees of freedom, are produced; pincers of subjection, arm and support, respectively. The movement activates with willing gears in a mechanism "planetarium." The motors energize with an interface that is implemented with transistors of power. In general, the GAL, are circuits integrated with programmable logic that use MOSFET technology, these have matricial arrangement of fuses that initially connects all the input terminals from the circuit, with all the floodgates of an AND-OR arrangement, it also contains a section denominated output logic cell macro (OLMC), which is a section of the GAL that includes the necessary elements in order to configure their outputs, four different logical ways; with the GAL's a versatile tool in the solution of systems provide logical combinations or secuenciales. The circuits that are generated with this design technique are small and consequently economic.

1.0 CONTROL DEL BRAZO MECANICO.

Los brazos mecánicos existen en el mercado desde hace varios años. Por regla general los adquieren las universidades, los centros de investigación y las industrias [6]. La tarea del proyecto "**Estructura y Programación de Arreglos Lógicos Genéricos (GAL'S) con Aplicación al Control de un Brazo Mecánico**", es mostrar un brazo mecánico con tres grados de libertad, éste se controla a través de la programación de arreglos lógicos genericos (GAL). En términos de construcción y geometría, los brazos de robot educacionales siguen la misma estructura de los grandes manipuladores industriales, por lo general poseen menos ejes independientes de movimiento [6] y [7], más adelante se muestra el proceso de generar la obtención de expresiones matemáticas Booleanas mediante una lógica de diseño secuencial, para manipular los tres movimientos diferentes del brazo, se utilizan motores de 5 Vdc, dos de estos motores incluyen un sistema de caja reductora (sistema planetario) que disminuye el giro del motor a pequeños movimientos del eje de salida, con un par de giro muy alto, estos motores se usan para el movimiento de giro del sistema y del brazo que sube y desciende, respectivamente. El tercer motor permite el control de cierre y apertura de la tenaza de sujeción o pinzas de agarre (véase la figura 1).

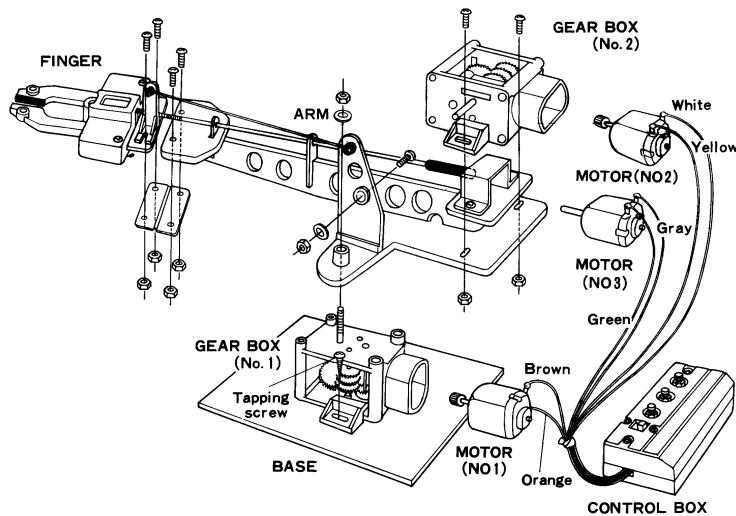


Figura 1. Geometría del brazo mecánico.

2.0 ANTECEDENTES.

La lógica programable se usa como una manera de “personalizar” los diseños lógicos, es decir, se diseñan pensando en el hardware propio. Los primeros PLD’S se programaron por máscara y se desarrollaron por fabricantes de computadoras, a principios de los sesentas llegó la lógica programable por fusibles y desde entonces ésta técnica, estuvo disponible tanto para pequeños como para grandes usuarios [1]. La tarea del diseñador lógico se simplifica con el uso del software, existen programas con los que actualmente se realizan tareas de diseño y minimización de lógica secuencial, esto es, representación Booleana de alto nivel, el software facilita la selección de dispositivos y ayuda a generar vectores de prueba para la simulación del circuito antes de su implementación final.

3.0 PROCEDIMIENTO DE DISEÑO

El diseño de un circuito secuencial empieza a partir de un conjunto de especificaciones y culmina en un diagrama lógico ó una lista de funciones Booleanas de las cuales se obtiene el diagrama final, el circuito se construye con el GAL16V8 [2], durante el análisis se determina una estructura de compuertas combinatorias, la cual conjuntamente con Flip Flops producen un circuito que cubre las características enunciadas. La cantidad de Flips Flops se determina por el número de estados necesarios que se encuentran presentes en el circuito, la lógica combinatoria se deriva de la tabla de estados, y una vez que se definen el tipo y número de Flips Flops, el proceso de diseño sufre una transformación, de ser un sistema secuencial, se modifica, para que se resuelva como un circuito combinatorio. Las expresiones lógicas se minimizan mediante mapas de Karnaugh. El GAL16V8 se programa con el compilador que ofrece el software OPAL [5], la programación se inicia al describir el archivo que indica el tipo de dispositivo a utilizar, la asignación de terminales, y las ecuaciones lógicas de las salidas respetan la sintaxis del compilador.

3.1 Planteamiento del problema, se define exactamente el problema a resolver y se plantea de una manera correcta, se establece la descripción en palabras del comportamiento del sistema, también suele acompañarse con un diagrama de estados, un diagrama de tiempos o cualquier otra información pertinente. La secuencia de movimientos a realizar se ejecuta según las necesidades del problema a resolver. En este caso se mueve un objeto de un lugar a otro. Primeramente obtengase una secuencia a partir de las variables que intervienen en el desarrollo del movimiento del sistema mecánico, esto es, la asignación de variables de estados que gobiernan la secuencia requerida.

Un motor controla el movimiento de abrir y cerrar las pinzas de sujeción, otro motor acciona el movimiento del brazo desde arriba y hacia abajo, y el tercero de ellos hace girar en forma rotacional el mecanismo de soporte de toda la estructura del brazo mecánico. Con referencia a las ecuaciones que se obtienen para su aplicación en el brazo de robot, y dada la existencia en el mercado, utilícase el GAL16V8, tres motores de 5 Vdc, y una interfaz de potencia de transistores para cada uno de los motores.

La secuencia del problema a resolver es la siguiente:

- 1 Se inicia la secuencia con los motores en reposo
- 2 Activación del cierre de las tenazas de sujeción
- 3 El brazo se habilita y sube
- 4 Girar la base de soporte de la estructura
- 5 Inversión de la secuencia de los pasos 4 al 1
- 6 La operación se repite de nueva cuenta

3.2 Tabla de estados: De la información dada del circuito se obtiene la tabla de estados, en la tabla se definen las características del comportamiento del problema a resolver. (Veáse la Tabla 1). La descripción de los movimientos de cada uno de los tres motores se simboliza con las letras, Q1 abre y Q2 cierra el motor de las pinzas de sujeción, Q3 sube y Q4 baja el brazo y finalmente Q5 gira a la derecha, Q6 se usa para realizar el giro a la izquierda del motor de la base de soporte de la estructura del brazo mecánico, Qn es el estado actual y Qn+1 es el estado siguiente.

TABLA 1

Qn						Qn+1					
Q1	Q2	Q3	Q4	Q5	Q6	Q1	Q2	Q3	Q4	Q5	Q6
0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	1	0	0	0	1	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0

Tabla 1. Asignación de estados del movimiento del brazo mecánico.

3.3 Mapas de Karnaugh: Enseguida se traduce la tabla de asignación de estados a un mapa de Karnaugh de 6 variables [8], con éstas se obtienen las ecuaciones lógicas Booleanas (Veáse la Tabla 2), al mismo tiempo se reduce la asignación de los estados y se encuentran las ecuaciones lógicas que gobiernan el control del brazo mecánico.

TABLA 2

Q1Q2Q3	Q4Q5Q6							
	000	001	011	010	110	111	101	100
000	D1	D4		D6				D2
001	D5							
011								
010								
110								
111								
101								
100	D3							

Tabla 2. Mapa de Karnaugh de 6 variables.

Las ecuaciones se definen para usar Flip Flops Tipo D [8] con la condición de un switch de arranque A:

$$D1 = \overline{Q1} * \overline{Q2} * \overline{Q3} * \overline{Q4} * \overline{Q5} * \overline{Q6} * A \quad (1)$$

$$D2 = \overline{Q1} * \overline{Q2} * \overline{Q3} * \overline{Q4} * \overline{Q5} * \overline{Q6} * A \quad (2)$$

$$D3 = \overline{Q1} * \overline{Q2} * \overline{Q3} * \overline{Q4} * \overline{Q5} * \overline{Q6} * A \quad (3)$$

$$D4 = \overline{Q1} * \overline{Q2} * \overline{Q3} * \overline{Q4} * \overline{Q5} * \overline{Q6} * A \quad (4)$$

$$D5 = \overline{Q1} * \overline{Q2} * \overline{Q3} * \overline{Q4} * \overline{Q5} * \overline{Q6} * A \quad (5)$$

$$D6 = \overline{Q1} * \overline{Q2} * \overline{Q3} * \overline{Q4} * \overline{Q5} * \overline{Q6} * A \quad (6)$$

3.4 El GAL16V8 posee las siguientes características:

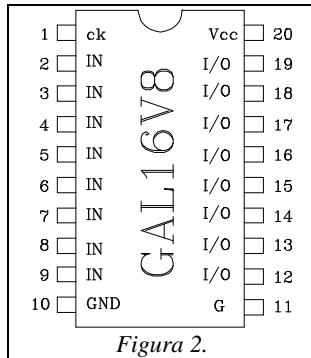


Figura 2.

El microcircuito consta de 20 terminales.
 La fuente de alimentación es; Vcc = 5 Volts ± 5%
 Consumo de corriente 90 mA.
 Frecuencia máxima del reloj para los FF, 41.6M Hertz
 Vih, voltaje de entrada en nivel alto = 2 Volts.
 Voh, voltaje de salida en nivel alto = 2.4 Volts.
 Vil, voltaje de entrada en nivel bajo = 0.8 Volts.
 Vol, voltaje de salida en nivel bajo 0.5 Volts.
 Temperatura de operación de 0 a 75 °C.
 Admite 16 variables de entrada diferentes y
 Ocho (8) variables de salida diferentes.

En su estructura interna se encuentra un plano AND programable a la entrada, es programable por medio de MOSFET'S [1], [2] y [3] de compuertas flotantes.

El GAL16V8 es un dispositivo de la familia de los Dispositivos Lógicos Programables PLD'S, contiene importantes mejoras sobre sus antecesores, que lo hacen versátil y funcional, la principal ventaja es su reprogramabilidad, es decir, el GAL16V8 ofrece la opción de borrar las ecuaciones grabadas en él mediante pulsos eléctricos, su celda básica es una EECMOS, que le proporciona la característica de borrable, además ofrece un bajo consumo de corriente.

3.5 Uso de la computadora para introducir las ecuaciones obtenidas en un archivo que se grabara con extensión "file.eqn". El circuito se programa con el compilador OPAL. En primer término se crea un archivo ASCII con las ecuaciones solución, mismo que se graba con la extensión "file.eqn", sin olvidar que se trata de una secuencia lógica secuencial, esto es, a las salidas se les anteponen dos puntos (:) después del signo de igual, se definen las entradas y las salidas de acuerdo a las variables empleadas, no olvidar definir la entrada OE, esta habilita las salidas (tri-state) del GAL16V8. Enseguida se muestra el formato del archivo "file.eqn".

```
;Brazo mecánico con tres motores
chip brazo1 gal16v8
CLK A 3 4 5 6 7 8 9 Gnd
OE 12 12 Q6 Q5 Q4 Q3 Q2 Q1 Vcc
equations
Q1:=/Q1*/Q2*/Q3*/Q4*/Q5*/Q6*/A
Q2:=/Q1*/Q2*/Q3*/Q4*/Q5*/Q6*/A
Q3:=Q1*/Q2*/Q3*/Q4*/Q5*/Q6*/A
Q4:=/Q1*/Q2*/Q3*/Q4*/Q5*/Q6*/A
Q5:=/Q1*/Q2*/Q3*/Q4*/Q5*/Q6*/A
Q6:=/Q1*/Q2*/Q3*/Q4*/Q5*/Q6*/A
```


3.8 Si la simulación produce resultados satisfactorios, se graba en el GAL16V8 el archivo "file.jed" en un programador de dispositivos programables y queda listo para la aplicación. Finalmente se conecta el circuito de control y de potencia del brazo mecánico, se observa en la figura 4 un circuito buffer (74LS125) [4] necesario para obtener la corriente de disparo de los transistores de potencia que gobiernan el sentido de giro de los motores de 5 Vdc, con el 74125 se produce la señal necesaria en los trasistores, para obtener el movimiento de avance y retroceso en las pinzas de sujeción, brazo y base de soporte, respectivamente. Se conectan además leds como indicadores puntuales del proceso, en cada una de las salidas para observar el comportamiento del sistema (véase la figura 4).

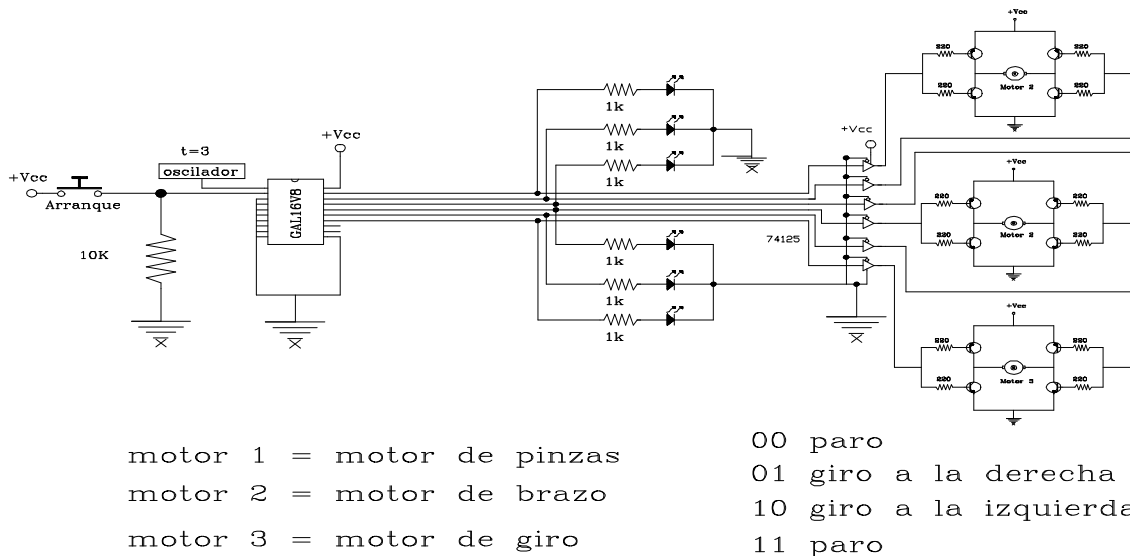


Figura 4. GAL Aplicado al control de un brazo mecánico.

3.9 Por último se realizan pruebas eléctricas físicas del diseño final.

4.0 RESULTADOS

Los resultados de este trabajo son; se minimizan expresiones booleanas secuenciales, se controla el movimiento de un brazo mecánico con tres grados de libertad, se diseña una interfáz con transistores de potencia, el circuito emplea el GAL16V8; se emplea el compilador OPAL para producir las expresiones lógicas secuenciales que se graban en el GAL, con ésto se hace práctico y flexible la programación de GAL'S, obteniéndose minimización de costos, al ahorrar en la adquisición de componentes y en la reducción del dimensionado físico del ensamble; se producen simulaciones gráficas de las señales de control que describen el comportamiento dinámico del sistema, lo cual ayuda a facilitar la búsqueda de la estabilidad en el prototipo final.

5.0 REFERENCIAS BIBLIOGRAFICAS

- [1] Programmable Logic Devices Databook and Design Guide, National Semiconductor, 1990.
- [2] Programmable Logic Design Guide, National Semiconductor, May 1986.
- [3] Programmable Logic Comercial/Military, Handbook and Data Book 1986-1987, Advanced Micro Devices.
- [4] Programmable Logic, INTEL 1993
- [5] OPAL Junior User's Guide/Manual, National Semiconductor 1991.
- [6] Robótica Práctica, Tecnología y Aplicaciones, José Ma. Angulo, Paraninfo, 1986.
- [7] Curso de Robótica, José Ma. Angulo, Rafael Avilés, Paraninfo, 1989.
- [8] Teoría de Conmutación y Diseño Lógico, Hill Peterson, Mc.Graw Hill, 1979