Diseño de un controlador de lógica programable

Luis A. Ferrer
Edilberto Rodríguez
Candidatos a graduación

Sinopsis

En este artículo se presenta el diseño de un sistema de control de lógica programable. Aunque un diseño de este tipo requiere un procedimiento de varias etapas, estas etapas se agruparon en dos fases. La primera fase agrupa todo lo relacionado con el diseño y la implantación del equipo; la segunda fase cubre el diseño e implantación de la programación. Para la construcción del control de lógica se usó el entrenador ET-3400 de Heathkit y el microprocesador MC6800, ya que ambos se usan en el programa de estudios de ingeniería eléctrica de la Universidad Politécnica de Puerto Rico.

Abstract

This document shows how to design a programmable logic controller. This type of design involves several steps, but these steps were grouped in two subsequent phases. The first phase deals with the design and implementation of the hardware; the second phase covers the design and implementation of the software. The Heathkit ET-3400 trainer and the MC6800 microprocessor were used to build the programmable logic controller, because these tools are used in the electrical engineering curriculum at the Universidad Politécnica de Puerto Rico.

Introducción

Los sistemas de lógica basada en transistores poseen todas las ventajas de los circuitos electrónicos del estado sólido: son seguros, confiables, pequeños, rápidos y de costo módico. Su única falla es que no son fáciles de modificar. Ante la urgencia de alterarlos se necesita modificar el alambrado

de los dispositivos lógicos o cambiar los dispositivos. Estos cambios al equipo (hardware) son poco prácticos debido a su dificultad y al tiempo que consumen.

Durante los pasados años se ha popularizado una vía fundamentalmente diferente en la construcción de sistemas de lógica industrial. En este método las decisiones tomadas por el sistema las implantan instrucciones codificadas que residen en un dispositivo de memoria y las ejecuta un microprocesador. Si hay que modificar el sistema de control, solo se alteran las instrucciones. Este método se conoce como cambio de programación (software) y se implanta de manera sencilla y rápida con solo oprimir un teclado.

Un programa es una secuencia completa de las instrucciones codificadas que controlan el funcionamiento del sistema. Cuando se usa el método de cambio de programación, el sistema de lógica se conoce como un sistema programable. Si todos los componentes necesarios para el control se ensamblan como una unidad completa, el sistema se conoce como un control programable.

El diseño

El diseño de un sistema de control de lógica programable (PLC por sus siglas en inglés) requiere un procedimiento de varias etapas. Las etapas se dividieron en dos grupos:

a. Diseño e implantación del equipo

b. Diseño e implantación de la programación

La operación de un sistema PLC requiere que haya una comunicación con el usuario para que el sistema pueda recibir las instrucciones que determinan su función de control. La función de control es una representación del circuito digital usado para controlar un sistema específico según lo determina el usuario. Para representar un circuito digital se usan ecuaciones booleanas, las cuales muestran las variables de control y la lógica digital que hay entre estas variables. Esta forma de representar la función de control nos llevó a concluir que la manera adecuada de representar el circuito era usando el microprocesador MC6800, el cual se estudia en la Universidad Politécnica de Puerto Rico.

La primera etapa del diseño del equipo requiere determinar cuáles componentes y qué cantidad de éstos se usarán. El sistema debe proveer para la entrada de las ecuaciones booleanas, para la recepción de las señales de control y para el ajuste de los sistemas controlados. Para este propósito es necesario usar tres adaptadores de periferales de transmisión en paralelo (MC6821 o MC6520). También se requiere diseñar un decodificador para poder usar los adaptadores de los periferales. El decodificador es imprescindible para una comunicación apropiada entre el microprocesador y los adaptadores.

El sistema de computadora que se usó para construir el PLC se conoce como ET-3400 y posee un microprocesador MC6800. Para incorporar los tres adaptadores de los periferales a este sistema hay que regirse por el mapa de memoria que presenta la figura 1 y que pertenece al ET-3400.

MONITOR ROM	FFFF FC00
NO USADA	C1FF C170
INDICADORES	C16F C110
NO USADA	C10F C00F
TECLADO	C00E C003
NO USADA	C002 C000
256 BYTES RAM	01FF 0100
59 BYTES RAM (RESERVADOS)	00FF 00C5
197 BYTES RAM	00C4 0000

Figura 1. Mapa de la memoria del ET-3400

Para localizar los tres adaptadores se escogen los 12 espacios de memoria que estén libres de acuerdo al mapa de la memoria. Se escogieron estos 12 espacios comenzando con la dirección 7FF0 y finalizando en la 7FFB. Una vez se saben las direcciones en las cuales se encuentran los adaptadores

de los periferales y aplicando los métodos de la electrónica digital, entonces se diseñó el decodificador para estos adaptadores. A continuación se presenta un diagrama del decodificador (fig. 2).

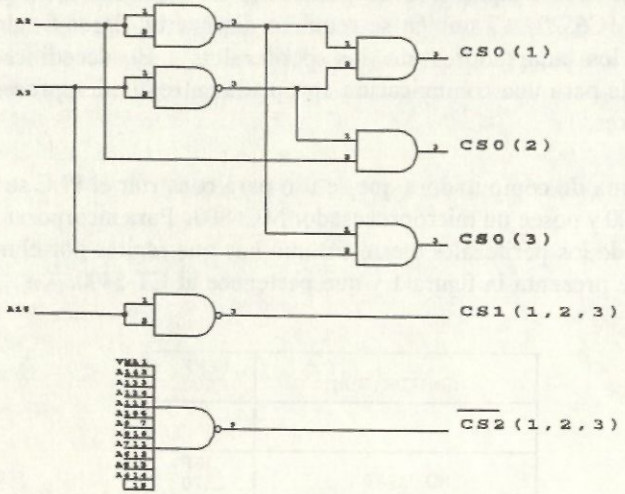

Figura 2. Diagrama del decoficador para los adaptadores

El diseño del decodificador y del circuito de comunicación con el usuario componen las partes más importantes del diseño del equipo en el sistema PLC-91. Para controlar de este circuito es necesario programar el microprocesador MC6800, tarea que es posible gracias al ET-3400.

El único problema que se encontró en el diseño de la programación fue la capacidad de memoria RAM que ofrece el ET-3400. El sistema PLC-91 usa 900 bytes para la programación y 200 bytes para las memorias usadas; esto es un total de 1.1 KB de memoria RAM. El ET-3400 ofrece 460 bytes de memoria RAM. Por esta razón se necesitó expandir la memoria RAM del ET-3400 añadiendo 2048 bytes. De esta forma se obtuvo un total de 2.5 KB, suficientes para la operación requerida.

Se usaron memorias del tipo 2016-A, las cuales usan 11 líneas de dirección para localizar 2048 (2^{11}) posiciones de memoria. Según el mapa de memoria en la figura 3, el cual incluye los espacios ocupados por los adaptadores de periferales, hay un espacio de 32,240 bytes libres entre las

direcciones 200 a 7FF0. Aquí se pueden escoger 2048 direcciones para poder localizar la nueva memoria.

MONITOR ROM	FFFF FC00
NO USADA	C1FF C170
INDICADORES	C16F C110
NO USADA	C10F C00F
TECLADO	C00E C003
NO USADA	C002 C000
ADAPTADORES DE LOS PERIFERALES	7FFB 7FF0
256 BYTES RAM	01FF 0100
59 BYTES RAM (RESERVADOS)	00FF 00C5
197 BYTES RAM	00C4 0000

Figura 3. Mapa de la memoria con los espacios para los adaptadores

Para que el microprocesador MC6800 se comunicara con la nueva memoria era necesario diseñar un decodificador para hacer posible esta interfase. Para diseñar el decodificador es útil referirse a la figura 4.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Figura 4. Especificaciones para el decodificador de la memoria expandida.

Mediante estas especificaciones se determinó el decodificador para la memoria expandida. Este se muestra en la figura 5.

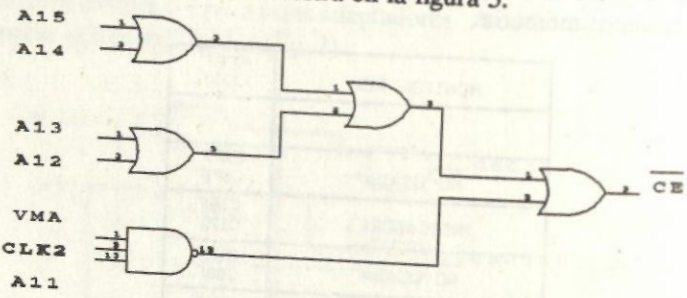

Figura 5. Diagrama del decodificador de la memoria expandida.

Además de los decodificadores de los adaptadores de los periferales y la memoria, se usó un decodificador para la señal de "Read Enable" (RE) del ET-3400. Esto es así ya que a la salida de la barra bidireccional del entrenador se conectan dos "buffers" triestados (3 state). Estos "buffers" están conectados como se muestra en la figura 6.

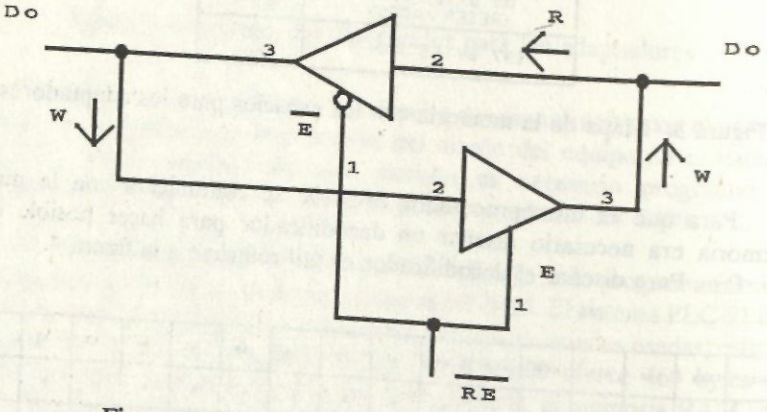

Figura 6. Digrama de conexión de los "buffers"

Cuando la señal de control **RE**=0, se activa el "buffer" de lectura y cuando RE=1, se activa el "buffer" de escritura. El estado normal de la señal de control es RE=1.

El microprocesador debe comunicarse con los adaptadores y la memoria que se conectan a la barra de datos extendida, esto es, después de los conectores de datos del panel. Para esta comunicación se requiere la señal de control RE para garantizar una transferencia de datos válida. Por tal motivo se construyó un decodificador para esta señal. En la construcción de este decodificador, tanto para la memoria extendida como para los adaptadores de los periferales, se usó la combinación de las señales de selección de cada uno de ellos con la señal de R/W (read/write) para producir una señal de control RE correcta. Para el decodificador de la señal RE es necesario considerar las líneas que seleccionan a los adaptadores de los periferales, la línea de selección de la memoria expandida CE y la señal R/W. Este se muestra en la figura 7.

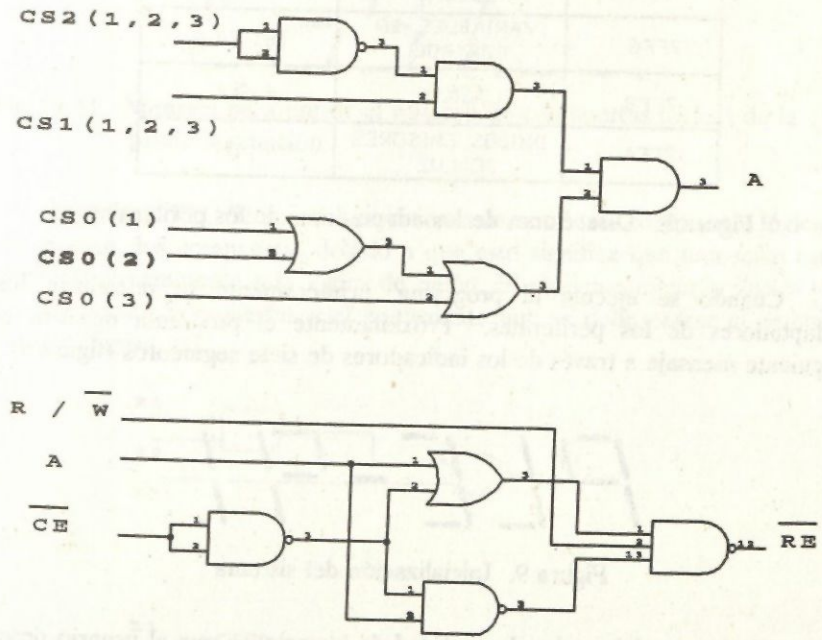

Figura 7. Diagrama del decodificador para la señal RE

Para el diseño de la programación se consideraron varios aspectos. Primeramente se determinó la dirección que se usó para las señales de control, los diodos emisores de luz (LEDs), las ecuaciones booleanas y el "push-button". Seguidamente se determinó cómo el usuario podía recibir las

instrucciones a seguir y cómo podía utilizar el teclado del ET-3400 para hacer las entradas de la cantidad de ecuaciones y compuertas lógicas que quiere usar para cada ecuación. La figura 8 muestra las direcciones de los adaptadores de los periferales.

DIRECCION	DESCRIPCION	# BITS ENTRADA O SALIDA
7FF0	SENALES DE CONTROL	8 (E)
7FF2	PUSHBUTTON	1 (E)
7FF4	VARIABLES DE FEEDBACK	8 (E)
7FF6	VARIABLES DE CONTROL	8 (E)
7FF8	LOGICA	4 (E)
7FFA	DIODOS EMISORES DE LUZ	8 (S)

Figura 8. Direcciones de los adaptadores de los periferales

Cuando se ejecuta el programa, primeramente se inicializan los adaptadores de los periferales. Próximamente el programa muestra el siguiente mensaje a través de los indicadores de siete segmentos (figura 9).

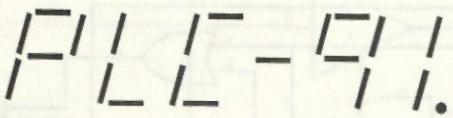

Figura 9. Inicialización del sistema

El sistema debe saber la cantidad de ecuaciones que el usuario desea usar. Dependiendo del sistema que se quiere controlar, se pueden usar de una a ocho ecuaciones. El sistema le indica al usuario que entre la cantidad de ecuaciones a usarse de la siguiente forma (figura 10).

Mediante el uso del teclado del ET-3400 el usuario entra el número de ecuaciones a usarse. Primero se oprime el número cero y luego la cantidad de ecuaciones. De cometerse un error el sistema presenta nuevamente esta

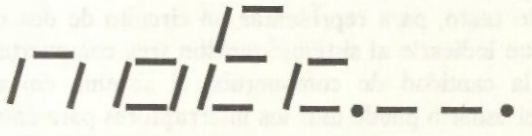

Figura 10. Mensaje del sistema para entrar el número de ecuaciones a usarse

pantalla. Una vez se entra este parámetro, se procede a solicitar la cantidad de compuertas lógicas de la primera ecuación. El sistema presenta un mensaje indicándole al usuario que entre la cantidad de compuertas lógicas que va a usar en la ecuación Y1 (figura 11).

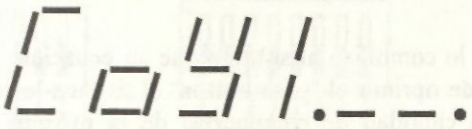

Figura 11. Mensaje para entrar el número de compuertas lógicas de la primera ecuación

El sistema PLC-91 permite ecuaciones de una ó tres compuertas lógicas. No se usan dos compuertas debido a que esto significa que una señal está entrando directamente a la lógica de unión. El siguiente ejemplo ilustra un circuito de dos compuertas y el equivalente que se debe entrar al sistema PLC-91 (figura 12).

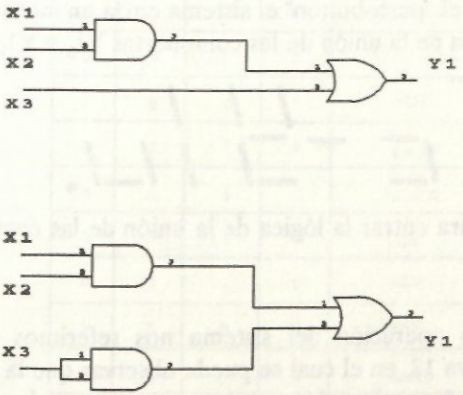

Figura 12. Circuito de dos compuertas y el equivalente que se entra al sistema

Como se puede observar, los circuitos resultan en la misma ecuación booleana. Por lo tanto, para representar un circuito de dos compuertas el usuario tiene que indicarle al sistema que son tres compuertas. Una vez el usuario entra la cantidad de compuertas, el sistema envía un mensaje indicando que el usuario puede usar los interruptores para entrar la primera ecuación (figura 13).

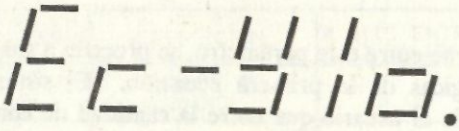

Figura 13. Mensaje que se recibe luego de entrar el número de compuertas

Si el usuario le comunicó al sistema que su ecuación era de una sola compuerta, luego de oprimir el "push-button" el sistema le envía un mensaje para que entre la cantidad de compuertas de la próxima ecuación. Si la ecuación es de tres compuertas el sistema le pide al usuario que entre la ecuación de la próxima compuerta (figura 14).

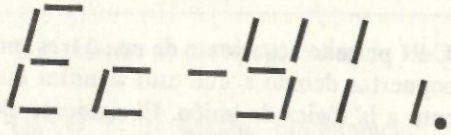

Figura 14. Mensaje para entrar la ecuación de la próxima ecuación

Luego de oprimir el "push-button" el sistema envía un mensaje para que el usuario entre la lógica de la unión de las compuertas $Y1_0$ y $Y1_1$ (figura 15).

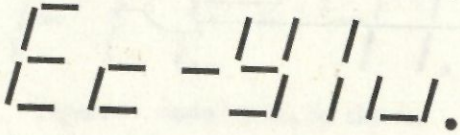

Figura 15. Mensaje para entrar la lógica de la unión de las compuertas $Y1_0$ y $Y1_1$.

Para observar la operación del sistema nos referimos al diagrama esquemático de la figura 12, en el cual se puede observar que la ecuación Y1 tiene tres compuertas. A continuación presentamos las ecuaciones ($Y1_0$, $Y1_1$, $Y1_u$) que el usuario debe entrar para este ejemplo en específico.

$$Y1_0 = X_1 \cdot X_2$$
$$Y1_1 = X_3 \cdot X_3$$
$$Y1_U = (\cdot)$$

Las variables de control ($X_1 ... X_8$), las variables de retroalimentación ($Y_1 ... Y_8$) y la lógica están localizadas respectivamente en los interruptores que presenta la figura 16.

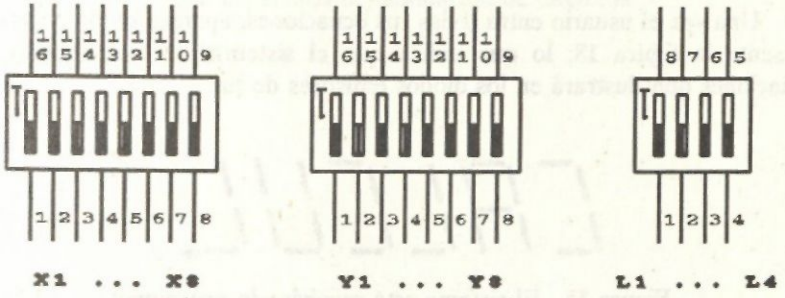

Figura 16. Interruptores del sistema

Oprimiendo los interruptores necesarios se obtienen las variables de control y retroalimentación necesarias. Sin embargo, para la lógica hay que referirse a la figura 17.

L1	L2	L3	L4	LOGICA
0	0	0	0	— — — —
0	0	0	1	NOT
0	0	1	0	EXOR
0	0	1	1	NEXOR
0	1	0	0	OR
0	1	0	1	NOR
1	0	0	0	AND
1	0	0	1	NAND

Figura 17. Lógica usada en el ejemplo de la figura 12

Al oprimir los interruptores indicados ($L_1 ... L_8$) se obtiene la lógica deseada para cada compuerta y para la lógica de unión.

La cantidad de ecuaciones, el número de compuertas en las ecuaciones y las ecuaciones booleanas se guardan en una estiba (stack) de memoria. La estiba se usa luego por varias sub-rutinas para determinar el estado lógico de las variables de control y retroalimentación para hacer la lógica correspondiente entre ellos.

Una vez el usuario entra todas sus ecuaciones, aparece el mensaje que presenta la figura 18; lo que indica que el sistema está resolviendo las ecuaciones que ilustrará en los diodos emisores de luz.

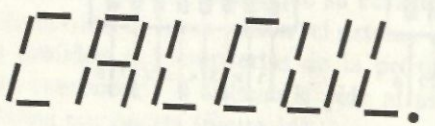

Figura 18. El sistema está resolviendo ecuaciones

El sistema va resolviendo ecuación por ecuación de forma sistemática. Escoge la primera ecuación, entra en una sub-rutina que determina cuáles variables usa esa ecuación. Primero determina las variables de control y sus estados lógicos correspondientes y luego determina las variables de retroalimentación y sus respectivos estados. Los estados de todas las variables se guardan en una estiba de memoria. Estos estados pueden ser 00 ó FF que representan 0 y 1 binario, respectivamente. Luego de tener todos los estados entra en la sub-rutina de lógica, la cual determina la lógica para resolver las variables. Próximamente el programa determina si la ecuación es de una ó tres compuertas. Si es de una compuerta guarda el resultado en la estiba de los resultados de las ecuaciones. Si es de tres compuertas, entonces resuelve la segunda ecuación con el mismo método o sub-rutina y luego los resultados de estas dos compuertas se unen en una lógica de unión. Ese es el resultado de la ecuación, el cual se envía a la estiba de los resultados. Una vez resueltas todas las ecuaciones, los resultados se envían a los correspondientes diodos emisores de luz.

Esperamos que los estudiantes de la Universidad Politécnica que estén interesados en el campo de las aplicaciones de microprocesadores puedan

llevar a cabo investigaciones y experimentos con la finalidad de mejorar y expandir el sistema aquí presentado. Los aspectos de mayor interés en este sentido son:

- la pantalla del sistema
- el teclado
- la programación de ROM
- la expansión de la memoria RAM
- los periferal de almacenamiento de datos
- la capacidad de simulación de circuitos