

ANEXO - D
LOGICA BINARIA

Aplicada a diagramas en escalera y de bloques para la programación de un mini PLC

La lógica binaria fue desarrollada a principios del siglo XIX por el matemático George Boole para investigar las leyes fundamentales en que se basa el razonamiento humano. En un sistema binario sus variables solo pueden adoptar dos valores, **Verdadero** y **Falso**, o **Sí** y **No**, o **Cerrado** y **Abierto** (en electrónica se representan con **1** y **0** respectivamente), y ambos estados deben ser mutuamente excluyentes. También se le llama *álgebra booleana*.

Los circuitos eléctricos de control, los circuitos digitales, los circuitos con fluidos (hidráulicos y neumáticos) y los circuitos con luz (fibra óptica), se prestan muy bien a ser analizados con la lógica binaria, porque es fácil construir circuitos que adopten tales valores, tensión no-tensión, conectado no-conectado, abierto-cerrado, encendido-apagado, etc. La adaptación del álgebra de Boole a los computadores digitales fue presentada en 1938 por Claude Shannon de los Laboratorios Bell.

El álgebra booleana sigue ciertas reglas y permite simplificar problemas lógicos complicados. Otras herramientas para el análisis de los sistemas lógicos son las tablas de verdad y los mapas de Karnaugh; estos no los estudiaremos.

Operaciones binarias básicas

Afirmación: $a \Leftrightarrow A$

equivale a un interruptor normalmente abierto (NA)

Negación:

$$\text{NOT } a \Leftrightarrow \bar{A} \Leftrightarrow \text{-}a \Leftrightarrow \text{NO } a$$

equivale a un interruptor normalmente cerrado (NC)

Conjunción:

$$a \text{ AND } b \Leftrightarrow A \cdot B \Leftrightarrow a \wedge b \Leftrightarrow a \text{ Y } b$$

equivale a 2 o más interruptores NA en serie

Disyunción:

$$a \text{ OR } b \Leftrightarrow A + B \Leftrightarrow a \vee b \Leftrightarrow a \text{ O } b$$

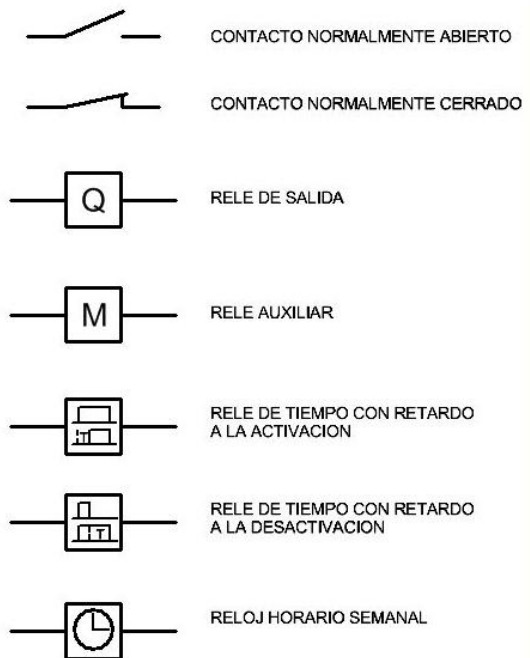
equivale a 2 o más interruptores NA en paralelo

Disyunción Exclusiva:

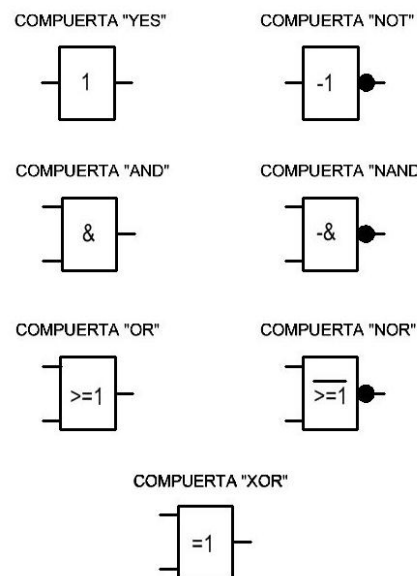
$$a \text{ XOR } b \Leftrightarrow \bar{A} \cdot B + A \cdot \bar{B} \Leftrightarrow A \times B \Leftrightarrow a \text{ X } b$$

equivale a un juego de interruptores serie-paralelo

Símbolos eléctricos

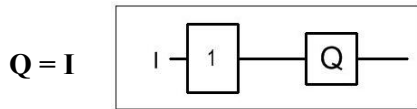
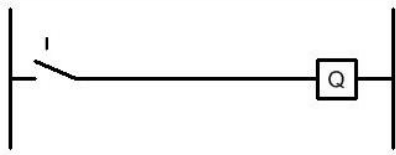


Compuertas lógicas (símbolos de bloques)



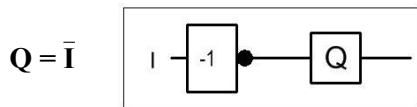
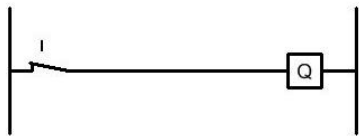
Circuitos binarios básicos y sus tablas de verdad

1. Interruptor sencillo NA – Compuerta YES



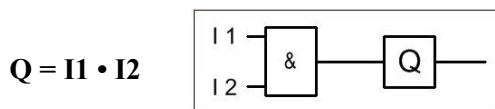
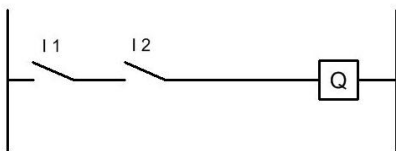
I	0	1
Q	0	1

2. Interruptor sencillo NC – Compuerta NOT



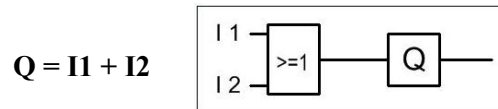
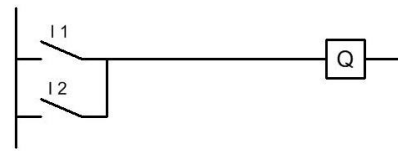
I	0	1
Q	1	0

3. Interruptores NA - NA en serie – Compuerta AND



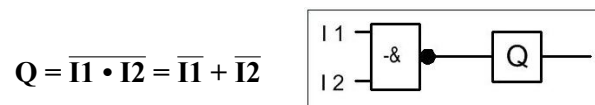
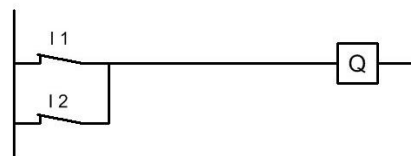
I1	0	0	1	1
I2	0	1	0	1
Q	0	0	0	1

4. Interruptores NA - NA en paralelo – Compuerta OR



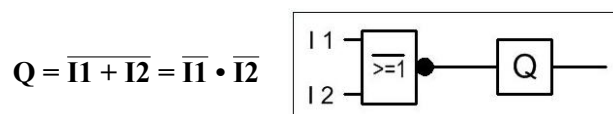
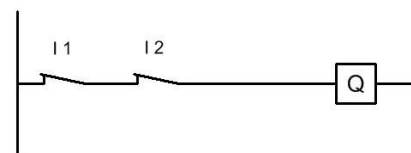
I1	0	0	1	1
I2	0	1	0	1
Q	0	1	1	1

5. Interruptores NC - NC en paralelo – Compuerta NAND



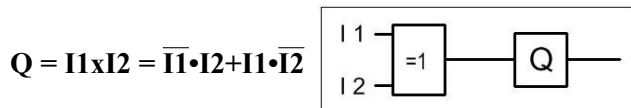
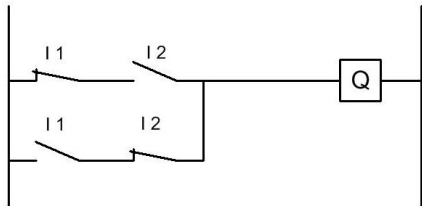
I1	0	0	1	1
I2	0	1	0	1
Q	1	1	1	0

6. Interruptores NC - NC en serie – Compuerta NOR



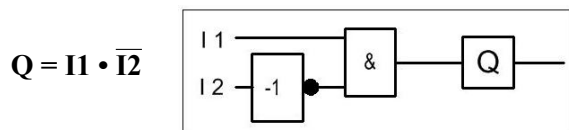
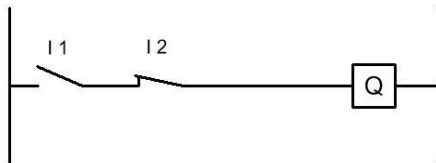
I1	0	0	1	1
I2	0	1	0	1
Q	1	0	0	0

7. Circuito Excluyente – Compuerta O exclusiva



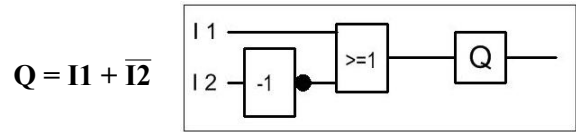
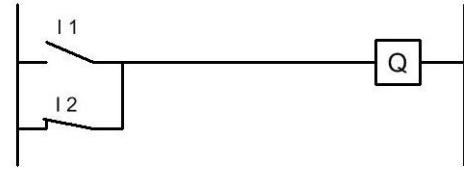
I1	0	0	1	1
I2	0	1	0	1
Q	0	1	1	0

8. Interruptores NA y NC en serie



I1	0	0	1	1
I2	0	1	0	1
Q	0	0	1	0

9. Interruptores NA y NC en paralelo



I1	0	0	1	1
I2	0	1	0	1
Q	1	0	1	1

Programación de PLC (autómatas)

Programmable Logic Controller

Los mini PLC se fabrican para voltajes específicos, por ejemplo, 120 VCA, 24 VCD, 240 VCA, 120 VCD. Cada modelo de PLC cuenta con un número determinado de entradas digitales y/o analógicas, y un número determinado de salidas. Existen también módulos de expansión.

Las entradas digitales requieren de algún nivel mínimo de voltaje para activarse (por ejemplo, 70 VCA en un PLC de 120 VCA), de lo contrario están desactivadas. Cada entrada se puede utilizar las veces que se requieran dentro del programa, ya que el PLC no requiere de más energía para usar una misma entrada varias veces en su programa lógico. Las entradas pueden ser activadas a través de un contacto exterior alimentado con el mismo voltaje de alimentación del PLC. Las entradas pueden ser de contacto mantenido o momentáneo (pulsador).

El PLC puede tener algunas entradas analógicas, las cuales pueden activarse con un nivel de voltaje programable, detectando así el voltaje de un sensor dentro de un rango, y cada una se activará o se desactivará según el nivel programado.

Las salidas pueden ser de relé (máximo 5 o 10A, y desde 12 VCD hasta 240 VCA), o pueden ser con un transistor de salida, estas normalmente se usarán para

cargas pequeñas que requieran mucha frecuencia de conmutación.

Un PLC tiene también otros elementos de programación con funciones análogas a elementos de control electromecánico o electrónico, tales como relés auxiliares, relés de tiempo con diferentes funciones, tales como retardo al energizar o al desenergizar o ambos, relojes semanales, reloj anual, contadores de pulsos, compuertas generadoras de pulsos, generadores de pulsos periódicos, comparadores, telerruptores y relés con memoria (latch).

Además para algunos de los elementos se puede definir que tengan o no remanencia (memoria de su estado) en caso de corte de energía.

El lenguaje de programación puede ser tipo diagrama en escalera (o eléctrico), como en el PLC Zelio de Schneider, o en bloques, como en el PLC Logo! De Siemens. Estos lenguajes son básicamente combinatorios (las salidas dependen de la combinación del estado de las entradas), y las secuencias se deben interpretar.

También existe el lenguaje gráfico de Control de Etapas de Transición GrafCET o *diagrama funcional*, con el cual se pueden representar secuencias de estado y ciclos completos del estado de un elemento. En un sistema secuencial el estado de un elemento depende de los estados anteriores.

El programa gráfico se puede introducir directamente con el teclado y la pantalla del mini PLC (si los tiene) o se puede hacer en un computador con Windows y transmitirlo con un cable a través del puerto USB.

Ejemplo 1

Diseñar el diagrama eléctrico para un mini PLC que accionará un circuito de luces exteriores y que:

- Sólo debe permitir el funcionamiento entre las 17:30 y las 5:30;
- Debe encender las luces todos los días a las 18:00;
- Permita encender o apagar las luces cuando se desee entre las 17:30 y las 5:30.

Solución

Salidas:

Q1 Contactor del circuito de luces

Entradas:

I1 Pulsador de conexión manual

I2 Pulsador de desconexión manual

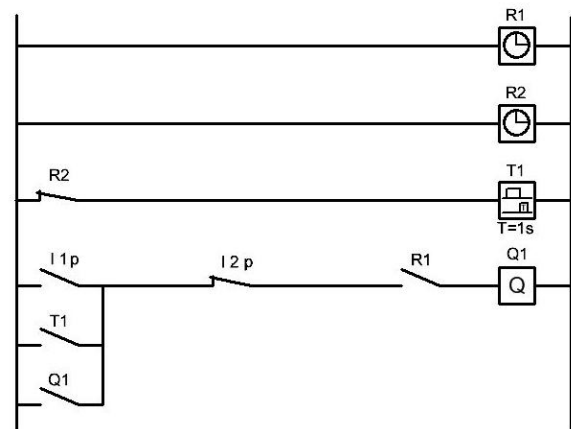
Otros:

R1 Reloj semanal ajustado L-K-M-J-V-S-D de 17:30 a 5:30

R2 Reloj semanal ajustado L-K-M-J-V-S-D de 18:00 a 18:05

T1 Relé de tiempo con impulso a la desactivación ajustado a 1s.

Diagrama de programación lógica en escalera o eléctrico (Ej. 1)



Ya que este es un circuito secuencial y de contactos momentáneos, no es muy fácil ni práctico construir una tabla de verdad para Q1.

Ejemplo 2

Diseñar el diagrama lógico para un mini PLC que accionará una bomba de agua de riego y que:

- Debe hacer funcionar la bomba todos los días entre las 8 pm y las 10 pm;
- Permita desconectar para mantenimiento, en invierno, o cuando suena la alarma;
- Permita operar manualmente fuera de horario;
- Si no hay agua en el tanque de captación la bomba debe apagarse y debe enviarse una señal de alarma, la cual debe durar 15 segundos y apagarse definitivamente.

Solución

Salidas:

- Q1 Contactor de la bomba de agua
- Q2 Sirena de alarma

Entradas:

- I1 Contacto de desconexión del sistema
- I2 Interruptor de nivel del tanque de captación
- I3 Contacto de conexión manual

Otros:

- R1 Reloj semanal ajustado L-K-M-J-V-S-D de 20:00 a 22:00
- T1 Relé de tiempo con retardo a la activación ajustado a 15s.

Diagrama de programación lógica en escalera o eléctrico (Ej. 2)

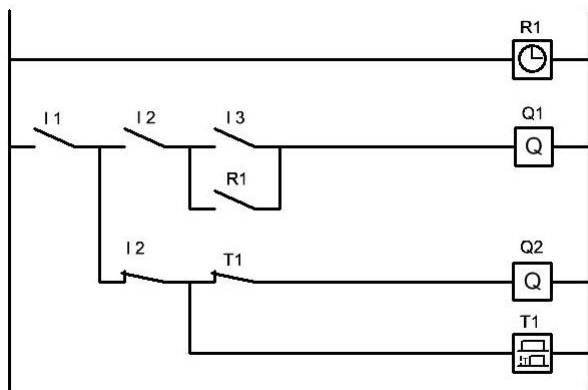


Diagrama de programación lógica en bloques (Ejemplo 2)

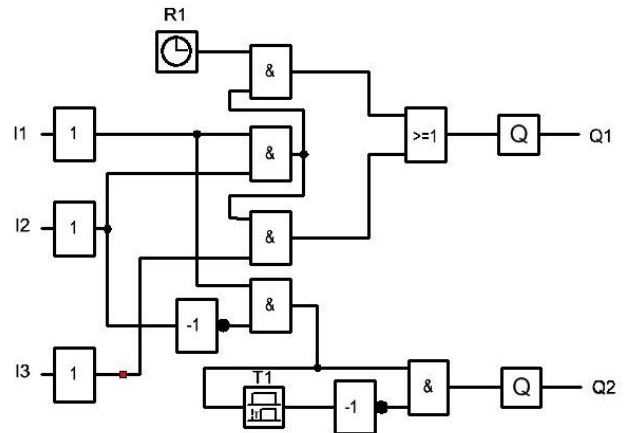


Tabla de verdad para Q1 (Cuando I1=0 es trivial: Q1=0)

I1	1	1	1	1	1	1	1	1
I2	0	0	0	0	1	1	1	1
I3	0	0	1	1	0	0	1	1
R1	0	1	0	1	0	1	0	1
Q1	0	0	0	0	0	1	1	1

Tabla de verdad para Q2 (Cuando I1=0 es trivial: Q2=0)

I1	1	1	1	1	1	1	1	1
I2	0	0	0	0	1	1	1	1
I3	0	0	1	1	0	0	1	1
T1	0	1	0	1	0	0	0	0
Q2	1	0	1	0	0	0	0	0