



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
CUAUTITLÁN**

**APLICACIÓN DE COMPUERTAS LÓGICAS
A MÁQUINAS ELÉCTRICAS CON EL PLC
ZELIO SOFT 2**

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECÁNICO ELÉCTRICISTA

PRESENTA

RICARDO JULIÁN RESENDIZ LAGUNA

ASESOR: ÁNGEL ISAÍAS LIMA GÓMEZ



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Dedicatoria

Si hay un ser llamado Dios le agradezco por sus protecciones y los dones que se me han dado en esta vida así como no perder la esperanza y la fe a lo largo de mi vida.

*A mis padres; María Elena Laguna Apolonio y Julián Reséndiz Alvares.
Por qué me han dado todo su apoyo, afecto, estímulo y se han sacrificado por mí para que consiga alcanzar mis metas .Les agradezco que hayan inculcado en mi valores, principios, que me han ayudado guiarme en la vida.*

*A mi hermano Michael Reséndiz Laguna
Que es parte de mis complementos con el que he pasado buenos momentos.*

A mis familiares en general con los que he compartido experiencias y buenos momentos así como su afecto.

Con los que ya no están con migo que desde allá arriba si hay un cielo me envían sus bendiciones.

A todos aquellos con los que he compartido buenos momentos y me han tendido la mano de amistad por citar algunos Cesar Martínez Jacob , Francisco Jahaziel ,Luis Miguel Torres, Rodolfo Jasso Galeana , Marco Vilchis Córdoba , Aldo ,Lupita ,Javier entre otros .

A todas aquellas flores que me he encontrado en el camino que me han motivado e inspirado.

*A mi asesor.
Al Ing. Ángel Lima que me brindo su tiempo, atención y orientación para llevar a cabo este trabajo y que muy gustosamente comparte sus conocimientos y muestra interés por el progreso de sus alumnos.*

*A la Universidad UNAM.
Por permitirme ser parte de esta gran institución en donde conocí a grandes amigos y excelentes personas ,me siento orgullosamente PUMA.*

ÍNDICE GENERAL

Objetivos.....	5
Introducción	6
Capítulo 1 Compuertas Lógicas	
1 Introducción a las compuertas lógicas.....	9
1.1 Operadores Lógicos.....	9
1.2 Compuertas Lógicas.....	10
1.3 Teoremas de las compuertas lógicas.....	13
1.4 Universalidad de las compuertas lógicas.....	15
1.5 Manejo de las funciones básicas de programación del software PLC ZELIO.....	17
1.6 Funcionamiento de menú y compuertas lógicas en diagrama de escalera con PLC ZELIO.....	22
1.7 Funcionamiento de menú y compuertas lógicas en diagrama de bloques con PLC ZELIO.....	25
1.8 Introducción a las tablas lógicas.....	29
1.9 Aplicación de ejemplo de compuertas lógicas de la tabla 1.6.....	29
1.10 Aplicación de ejemplo de compuertas lógicas de la tabla 1.7.....	35
Capítulo 2 Aplicación de principios básicos de control de compuertas lógicas.....	39
2 Función Relevador Contactor.....	39
2.1 Función Temporizador	
2.2 Aplicación de contactor temporizador en espera e inicio del diagrama	2.11.....49
2.3 Aplicación de contactor temporizador a la desconexión y reinicio del diagrama	2.12.....51
2.4 Aplicación de ejemplo semáforo relevadores y temporizadores del diagrama	2.13..... 53
Capítulo 3 Aplicación de operación a maquinas eléctricas con compuertas lógicas.....	55
3 Aplicación sincronización de 3 motores trifásicos con diferentes tiempos del diagrama 3.0	55
3.1 Aplicación sincronización de 3 motores trifásicos con diferentes tiempos con sentido inverso de giro del diagrama	3.1.....57
3.2 Aplicación de sincronización de tres motores mediante secuencias de operación con ciclos de repeticion del diagrama	3.2.....61

Figuras

1.1	Pantalla accediendo al icono del programa Zelio.....	17
1.2	Abrir programa nuevo.....	17
1.3	Selección de módulo de serie PLC.....	18
1.4	Selección de extensión de serie PLC	18
1.5	Seleccionando el tipo de lenguaje por bloques o escalera en el programa Zelio.....	18
1.6	Ventana de lenguaje de escalera.....	19
1.7	Indicaciones de escritura, correr programa y paro en el programa del software Zelio.....	19
1.8	Menú de transferencia de programa terminado.....	20
1.9	Menú y submenú de transferencia de programa al módulo PLC.....	20
1.10	Selección del puerto.....	20
1.11	Reconocimiento de puerto USB al módulo PLC.....	21
1.12	Transferencia del programa PC al módulo PLC.....	21
1.13	Accesando al módulo de programación BDF.....	25
1.14	Ecuación simplificada de la tabla 1.6 por suma se productos.....	30
1.15	Ecuación simplificada de la tabla 1.6 por productos de suma.....	33
1.16	Representación del módulo PLC del programa 1.4.....	34
1.17	Ecuación simplificada de la tabla 1.7 por suma de productos.....	37
1.18	Ecuación simplificada de la tabla 1.7 por productos de suma.....	37
2.1	Comando temporizador.....	43
2.2	Selección de tiempo comando temporizador.....	44
2.3	Selección de temporizador en lenguaje de escalera.....	45

Programas en el software Zelio

1.1	Compuerta NOT en diagrama de escalera.....	22
1.2	AND.....	22
1.3	NAND.....	23
1.4	OR.....	23
1.5	NOR.....	24
1.6	XOR.....	24
1.7	Compuerta NOT en diagrama de bloques.....	26
1.8	AND.....	26
1.9	NAND.....	27
1.10	OR.....	27
1.11	NOR.....	28
1.12	XOR.....	28
1.13	Acomodo de las compuertas lógicas del ejemplo 1.6 en suma de productos.....	31
1.13.1	Relacionando variables de las entradas L de la tabla 1.6.....	32
1.13.2	Termino de la relación de las variables con compuertas de la tabla 1.6.....	33
1.14	Tabla lógica 1.6 en productos de suma.....	34
1.15	Tabla lógica 1.7 en productos de suma.....	38
2.1	Función relevador contactor diagrama de escalera.....	39
2.2	Encendido del relevador.....	40
2.3	Apagado del relevador.....	41
2.4	Encendido del relevador diagrama de bloques.....	41
2.5	Apagado del relevador.....	42
2.6	Encendido función temporizador diagrama de escalera.....	43
2.7	Apagado reinicio función temporizador.....	44
2.8	Encendido función temporizador diagrama de bloques.....	46
2.9	Apagado reinicio función temporizador.....	47
2.10	Desactivación de función temporizador.....	48
2.11	Aplicación de relevador contactor en espera e inicio.....	49
2.12	Aplicación de relevador contactor a la desconexión y reinicio.....	51
2.13	Aplicación de ejemplo semáforo.....	53
3	Aplicación sincronización de 3 motores trifásicos con diferentes tiempos.....	55
3.1	Aplicación sincronización de 3 motores trifásicos con diferentes tiempos con sentido de giro	59
3.2	Aplicación de sincronización de tres motores mediante secuencias de operación con ciclos de repiticion	62

Tablas

1	Compuerta NOT.....	11
1.1	AND.....	10
1.2	NAND.....	11
1.3	OR.....	10
1.4	NOR.....	11
1.5	XOR.....	12

Diagramas de conexiones

1.1	Diagrama de conexión compuertas lógicas.....	21
1.2	Diagrama de la tabla lógica 1.6.....	35
1.3	Diagrama de la tabla lógica 1.7.....	38
2.2	Diagrama aplicación contactor temporizador en espera e inicio del programa : 2.11.....	50
2.3	Diagrama aplicación de contactor temporizador a la desconexión y reinicio 2.12.....	52
2.4	Diagrama aplicación de ejemplo semáforo 2.13.....	54
3	Diagrama aplicación sincronización de 3 motores trifásicos con diferentes tiempos del programa 3.0	56
3.1	Diagrama aplicación sincronización de 3 motores trifásicos con diferentes tiempos con sentido de giro inverso del programa 3.1.....	60
3.2	Aplicación de sincronización de tres motores mediante secuencias de operación con ciclos de repitición del programa 3.2	63
Glosario	8

OBJETIVOS

Generales

Dar a conocer que este equipo es capaz de realizar tareas de automatización , con mucha simplicidad en el manejo dar a conocer los beneficios que posee y las ventajas (tanto en costo económico como en espacio) así como desarrollar esta tesis como consulta para los alumnos de los laboratorios de máquinas eléctricas como consulta didáctica en la FES-CUAUTITLAN.

Específicos

1. Demostrar que con este equipo se pueden realizar con mucha facilidad , tareas de automatización como se realizan con un PLC.
2. Demostrar la programación de programas con compuertas logicas
3. Describir las funciones que tiene este equipo, para que el entendimiento sea rápido y efectivo.
4. Detallar los pasos necesarios que se requieren para poder programar este equipo.
5. Detallar claramente cada uno de los bloques de funciones que este equipo contiene.
6. Realizar varios proyectos reales en el cual se puede utilizar este equipo .

INTRODUCCIÓN

Con el gran crecimiento de la industria a nivel mundial, en la cual se desarrollan productos de consumo masivo con la mayor rapidez y con el menor costo posible, se ha tenido que implementar en dichas industrias métodos que ayuden a que esto sea una realidad. Es por ello que con el paso del tiempo se desarrolló e implementó la automatización de los procesos, con el muy conocido Controlador Lógico Programable o PLC, que hace que nuestros procesos requieran menos la intervención del ser humano, más de lo necesario y así se pueda agilizar la producción de algún producto en especial. Sin embargo, también el uso de estos controladores requiere un costo un tanto elevado y de personal muy calificado para la programación de dicho controlador, por lo que se está implementando el uso del relé inteligente o micro PLC, que tiene muchas (pero no todas) de las funciones que tiene un PLC normal, pero que con los elementos que cuenta es capaz de realizar procesos un tanto complejos, sin que se tenga que recurrir a la inversión de un PLC normal, que si se compararan los costos es menor al del PLC y su simplicidad para programarlo es también mucho menor y que se encuentra al alcance de cualquier persona que tenga poco (o nada) de conocimiento de lo que es programación compleja o avanzada.

Cada vez hay que ahorrar más tiempo y dinero, de la ingeniería hasta la operación, pasando por la puesta en servicio. Y las aplicaciones deben ofrecer siempre un *plus* decisivo de posibilidades, comodidad e ingenio técnico. Las funciones de control y maniobra juegan en este proceso un papel central, ya que deben ser a la vez simples e inteligentes. Simples en el manejo y en la operación, inteligentes para poder materializar las muchas posibilidades que se esperan de una aplicación con visión de futuro. El relé inteligente es un equipo con estas ventajas decisivas, gracias a la aplicación de una tecnología innovadora.

¿QUÉ ES EL RELÉ INTELIGENTE?

Es un módulo lógico que lleva integrados: control, unidad de mando y visualización con retroiluminación, fuente de alimentación, interfaz para módulo de programación (Card) y cable para PC, funciones básicas habituales pre programadas, por ejemplo para conexión retardada, desconexión retardada, relés de corriente, e interruptor de software, temporizador, marcas digitales y analógicas, entradas y salidas en función del modelo.

Con los relés inteligentes, se resuelven tareas de instalación y del ámbito doméstico, como por ejemplo alumbrado de escaleras, luz exterior, toldos, alumbrado de escaparates, etc. Así como la construcción de armarios eléctricos, máquinas y aparatos, como por ejemplo controles de puertas, instalaciones de ventilación, bombas de agua no potables, etc..

Así mismo se pueden utilizar para controles especiales en invernaderos o jardines de invierno, para el pre procesamiento de señales en controles y, mediante la conexión de un módulo de comunicaciones, por ejemplo ASI, para el control descentralizado de máquinas y procesos. Para las aplicaciones en serie en la construcción de máquinas pequeñas, aparatos y armarios eléctricos, así como en el sector de instalaciones, existen variantes especiales sin unidad de mando y visualización.

Los relés inteligentes o módulos lógicos, actualmente se utilizan para realizar procesos pequeños, porque se cree que no poseen la capacidad de realizar tareas complejas como los PLC's, pero poseen suficiente espacio en memoria como para controlar bandas transportadoras de llenados de botellas, o transferencias eléctricas automáticas, etc. Tal como lo hace un controlador lógico programable. Incluso la forma de programar es mucho más sencilla que con los controladores lógicos programables, y con estos módulos lógicos, no se necesita tener una licencia de *software*, ya que estos están más al alcance del usuario a un costo relativamente barato.

Los módulos lógicos son equipos que se pueden programar incluso sin el *software*, se puede realizar dicha acción directamente en el equipo, cosa que con un PLC no se puede hacer, incluso cuando se posee el *software* del relé inteligente o módulo lógico, los programas se pueden simular antes de descargarlo en los módulos, con lo que el programador puede estar seguro de que su proceso se realizará como se planeó. Con los ejemplos que se muestran, se puede ver la forma tan sencilla de realizar procesos complejos, y la representación de los programas en el *software* del equipo. Lo único que se necesita para poder programar los módulos lógicos son unos pocos conocimientos de compuertas lógicas y diagramación eléctrica.

Con los relés inteligentes se puede ahorrar tiempo y dinero cuando se diseñan Gabinetes de control, ya que por el espacio que ocupan reducen los costos de los mismos armarios, y su funcionalidad es tan grande que tareas complejas se pueden simplificar con la utilización de estos equipos.

Glosario

I	Entradas I de programa Zelio
Q	Salidas Q de programa Zelio
~	Negacion
•	Multiplicacion
[Agrupacion
+	Suma
NOT	Negacion
AND	Multiplicacion
NAND	Multiplicacion salida invertida negada
OR	Suma
Nor	Suma salida invertida negada
XOR	Exclusivo de igualdad

CAPÍTULO 1

COMPUERTAS LÓGICAS.

1. Introducción a las compuertas lógicas.

La potencia de los sistemas digitales está en la capacidad de sus componentes para tomar decisiones lógicas. Para esto debemos poder representar las proposiciones lógicas formuladas en lenguaje ordinario, con proposiciones simbólicas. Esto es asignarle un símbolo a la proposición.

En lógica las proposiciones son verdaderas o falsas, y para expresar su valor de verdad utilizaremos el símbolo "F" o "0" para falso y "V" o "1" para verdadero.

También representaremos las proposiciones en sí con ayuda de símbolos. Por ejemplo para simbolizar la proposición "la puerta está abierta" podríamos utilizar la letra P.

Si realmente la puerta está abierta podemos entonces decir que $P=1$. (o $P=V$)

Para cada proposición positiva existe una proposición negativa así podemos decir "la puerta NO está abierta" que representaremos como $P=0$. (o $P=F$). Para simbolizar esta proposición podemos hablar de P negada que representaremos como \bar{P} .

Las proposiciones solas no tienen mucho sentido si no se relacionan con otras para tomar decisiones. Así podemos reunir varias proposiciones lógicas para obtener una proposición compuesta. El valor de verdad de la proposición compuesta (verdadero o falso; 1 o 0) dependerá del valor de verdad de cada proposición componente y de la relación entre estas. La relación entre las proposiciones lógicas componentes viene dada por el operador lógico.

Los operadores lógicos primarios son el AND, el OR y el NOT

1.1 Operadores Lógicos

Operador lógico AND (conjunción lógica): Una proposición compuesta que utiliza este operador para relacionar sus proposiciones componente será verdad SI y SOLO SI las proposiciones componentes son verdaderas. Se simboliza con "." y al igual que en el álgebra convencional puede suprimirse. (AB , $A \cdot B$).

Ejemplo:

"José irá a la playa si el carro está listo Y el día es soleado"

Operador lógico OR (disyunción lógica): Una proposición compuesta que utiliza este operador será verdad si cualquiera de las proposiciones componentes es verdadera. Se simboliza con el signo "+". ($A+B$).

Ejemplo:

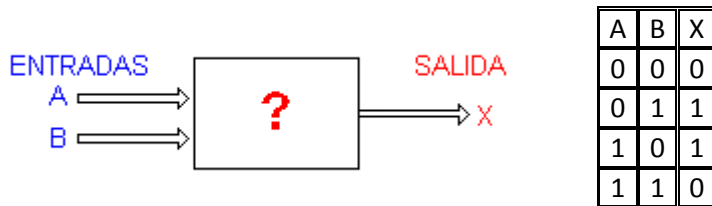
"La alarma sonará si se abre la puerta O se golpea el carro"

Operador lógico NOT (negación): Este operador se refiere a una sola proposición, negando su valor de verdad. Se representa con una barra sobre el símbolo que representa la proposición. (\bar{P})

Los operadores lógicos NOT, AND y OR se conocen como operadores lógicos básicos, puesto que cualquier función puede expresarse como una combinación de ellos.

Tablas de Verdad

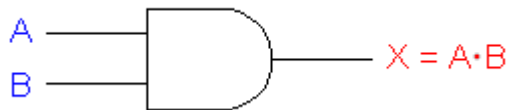
Para evaluar el valor de verdad de una proposición compuesta es muy útil usar una tabla de verdad. Esta es sencillamente una tabla que muestra el valor de la función de salida (proposición compuesta) para cada combinación de las variables de entrada (proposiciones componentes). En el siguiente circuito lógico de dos entradas la tabla muestra todas las combinaciones de los posibles niveles lógicos presentes en las entradas A y B y del correspondiente nivel de salida X.



A continuación se muestra las compuertas lógicas con la descripción como operan y las tablas con las señales de entradas y salidas representadas con unos y ceros.

1.2 Compuertas lógicas.

COMPUERTA AND

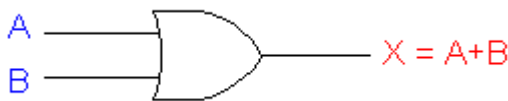


La compuerta AND es un dispositivo de dos o más entradas y una salida que cumple con la condición que la salida toma el valor lógico 1 si, y solo si todas las entradas valen 1.

TABLA DE VERDAD

A	B	A · B
0	0	0
0	1	0
1	0	0
1	1	1

COMPUERTA OR

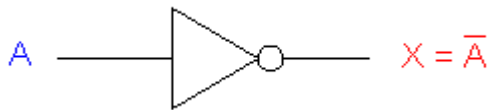


La compuerta OR es un dispositivo de dos o más entradas y una salida que cumple con la condición que la salida toma el valor lógico 1 si, y solo si una o más entradas valen 1.

TABLA DE VERDAD

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

COMPUERTA NOT



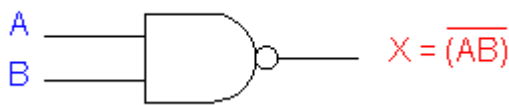
La compuerta NOT es un dispositivo de una entrada y una salida que cumple con la condición que la salida toma el valor lógico negado de la entrada.

TABLA DE VERDAD

A	\bar{A}
0	1
1	0

Las tres compuertas básicas enumeradas anteriormente pueden combinarse para realizar funciones lógicas más complejas. A continuación se muestran otras tres funciones importantes que se pueden realizar con compuertas, indicando su símbolo, su función y su tabla de verdad.

COMPUERTA NAND

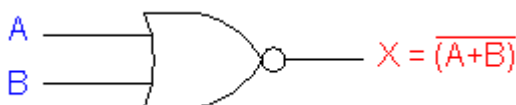


La compuerta NAND es un dispositivo de dos o más entradas y una salida que equivale a una compuerta AND seguida de negador (NOT AND = NAND). Cumple con la condición que la salida toma el valor lógico 0 si, y solo si todas las entradas valen 1. Si no la salida toma el valor 1. Se representa como una compuerta NAND seguida de un círculo que denota la negación

TABLA DE VERDAD

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

COMPUERTA NOR

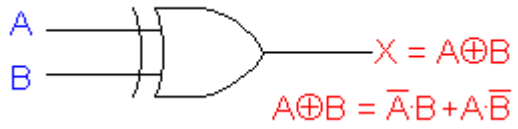


La compuerta NOR es un dispositivo de dos o más entradas y una salida equivalente a una compuerta OR seguida de un negador (NOT OR = NOR). Cumple con la condición que la salida toma el valor lógico 1 si, y solo si todas las entradas valen 0. Para las otras combinaciones la salida es 0. Se representa como una compuerta OR seguida de un círculo que denota la negación

TABLA DE VERDAD

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

COMPUERTA EXOR



La compuerta EXOR (or exclusivo) es un dispositivo de dos entradas y una salida que cumple con la condición que la salida toma el valor lógico 1 si, y solo si las entradas son diferentes.

TABLA DE VERDAD

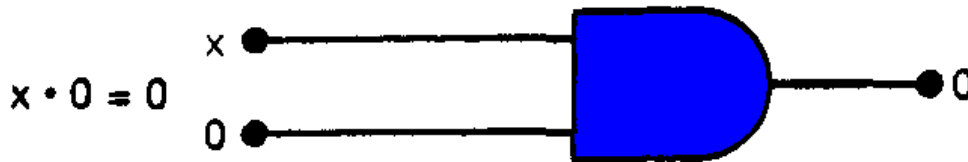
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

1.3 Teorema de compuertas lógicas Teoremas

Boléanos : Son un conjunto de reglas que nos pueden ayudar a simplificar las expresiones y los circuitos lógicos.

A continuación se muestran dichos teoremas.

En el teorema (1) se enuncia que si cualquier variable se opera con AND y con un 0 el resultado debe ser 0. Esto es fácil de recordar porque la operación AND es igual que la multiplicación común, en donde cualquier número que se multiplica por 0 es 0. Asimismo, se sabe que la salida de una compuerta AND será 0 siempre que cualquier salida sea 0, sin importar el nivel de la otra entrada.



El teorema (2) también es obvio en comparación con la multiplicación común.

El teorema (3) puede ser demostrado ensayando cada caso. Si $x=0$, entonces $0 \cdot 0 = 0$; si $x = 1$, entonces $1 \cdot 1 = 1$. Por lo tanto, $x \cdot x = x$.

El teorema (4) se puede demostrar en la misma forma. Sin embargo, también se puede razonar que en cualquier momento x o su inverso \bar{x} tiene que estar en el nivel 0 y por ende su producto AND siempre debe ser 0.

El teorema (5) es directo, ya que 0 sumado a cualquier número no afecta su valor, ya sea en la suma regular ó en una suma OR.

El teorema (6) estipula que si cualquier variable se opera con OR con 1, el resultado siempre será 1. Si verificamos esto para ambos valores de x ; $0 + 1 = 1$ y $1 + 1 = 1$. De manera equivalente se puede recordar que la salida de una compuerta OR será 1 cuando cualquier entrada sea 1, independientemente del valor de la otra entrada.

El teorema (7) se puede demostrar verificando ambos valores de x ; $0 + 0 = 0$ y $1 + 1 = 1$.

El teorema (8) se puede demostrar de forma similar, o simplemente podemos razonar que en cualquier momento x o \bar{x} debe estar en el nivel 1, de manera que siempre se opere con OR un 0 y un 1, lo cual da como resultado 1.

Teoremas con variables múltiples.

Los teoremas que se presentan a continuación implican más de una variable.

Los teoremas (9) y (10) se llaman leyes conmutativas. Estas leyes indican que no importa el orden en que se operen dos variables con OR o con AND, el resultado es el mismo.

$$9) x + y = y + x$$

$$10) x \cdot y = y \cdot x$$

Los teoremas (11) y (12) son leyes asociativas, las cuales afirman que se pueden agrupar las variables en una expresión AND o en una OR en cualquier forma que se desee.

$$11) x + (y + z) = (x + y) + z = x + y + z$$

$$12) x(yz) = (xy)z = xyz$$

El teorema (13) es la ley distributiva, la cual estipula que una expresión se puede desarrollar multiplicando término por término, como en el álgebra común.

$$13a) x(y + z) = xy + xz$$

$$13b) (w + x)(y + z) = wy + xy + wz + xz$$

Los teoremas anteriores son simple de entender pues obedecen al algebra común a diferencia de los que se muestran a continuación:

$$14) x + xy = x$$

$$15a) x + \emptyset xy = x + y$$

$$15b) \emptyset x + xy = \emptyset x + y$$

Teoremas de Morgan

Estos teoremas son de gran utilidad para simplificar expresiones en las que se invierte un producto o una suma de variables. Los teoremas son:

$$16) (x+y)' = x' \cdot y' \quad , \quad 17) (x \cdot y)' = x' + y'$$

Implicaciones del teorema de Morgan.

Considerando el teorema 16)

El lado izquierdo de la ecuación se puede tomar como la salida de una compuerta NOR cuyas entradas son x y y. Por otra parte, el lado derecho de la ecuación es el resultado de primero invertir x y y, y luego pasarlas a través de una compuerta AND. Estas representaciones son equivalentes como se ilustra en las figuras.

Ahora consideramos el teorema 17) El lado izquierdo de la ecuación se puede implementar con una compuerta NAND con entradas x y y. El lado derecho de la ecuación se puede llevar a cabo invirtiendo primero las entradas x y y luego pasándolas a través de una compuerta OR, estas representaciones son equivalentes y se muestran a continuación:

1.4 Universalidad de las compuertas NAND y NOR.

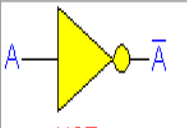
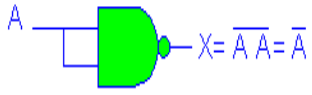
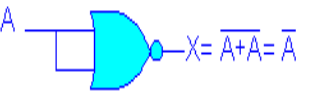
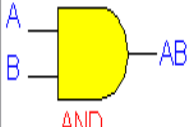
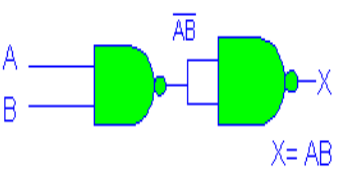
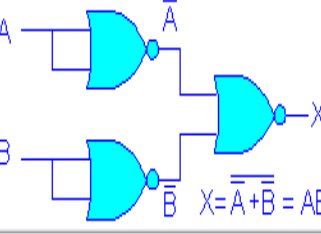
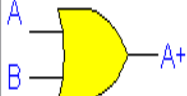
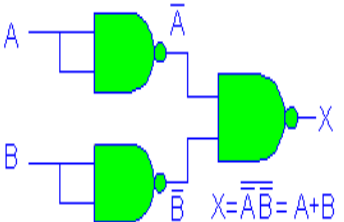
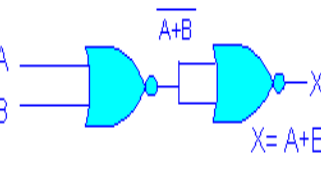
Estas compuertas se dicen que son "universales" puesto que con cada una de las dos familias podemos realizar todas las funciones lógicas.

En la tabla a continuación se muestran los operadores lógicos en función de solo compuertas NOR y solo compuertas NAND la bola delante de cada compuerta significa negación al pasar por dos negaciones la condición se vuelve valor positivo no negado.

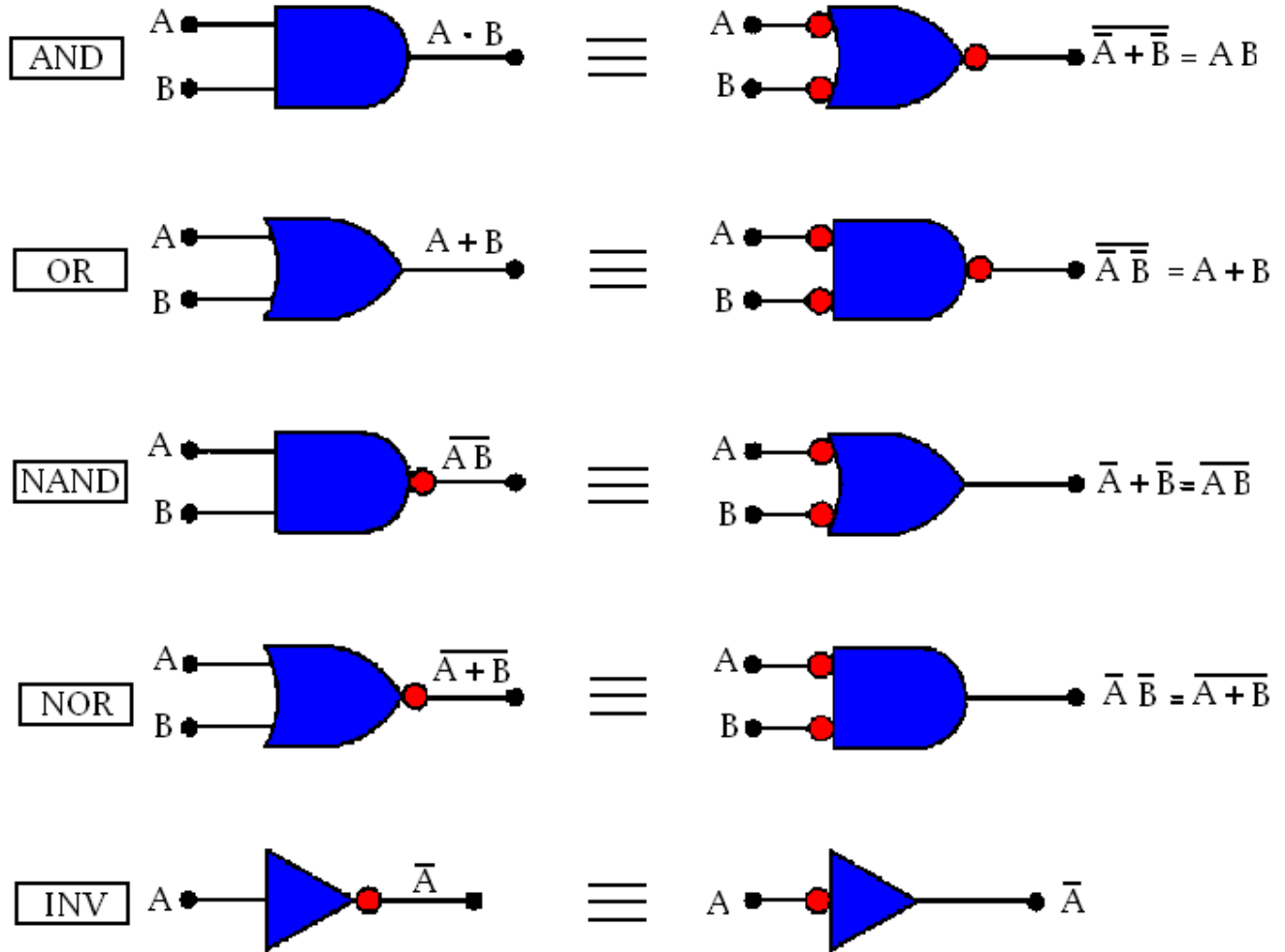
Universalidad de las compuertas NAND Y NOR

Estas compuertas se dicen que son "universales" puesto que con cada una de las dos familias podemos realizar todas las funciones lógicas.

En la tabla a continuación se muestran los operadores lógicos en función de solo compuertas NOR y solo compuertas NAND.

	NAND	NOR
 <p>NOT</p>	 <p>$X = \overline{A \cdot A} = \overline{A}$</p>	 <p>$X = \overline{A + A} = \overline{A}$</p>
 <p>AND</p>	 <p>$X = \overline{\overline{A \cdot B}} = AB$</p>	 <p>$X = \overline{\overline{\overline{A + B}}} = AB$</p>
 <p>OR</p>	 <p>$X = \overline{\overline{A} \cdot \overline{B}} = A + B$</p>	 <p>$X = \overline{\overline{A + B}} = A + B$</p>

Representaciones alternas de compuertas lógicas. Se han introducido las cinco compuertas lógicas básicas (AND, OR, INVERSOR, NAND y NOR) y los símbolos lógicos estándar que se usan para



representarlas en diagramas de circuitos lógicos.

Descripción de semejanza de transformación de las compuertas lógicas.

En el lado izquierdo de la ilustración se muestra el símbolo estándar para cada compuerta lógica y en el lado derecho, el símbolo alternativo. El símbolo alternativo para cada una, puerta se obtiene a partir del símbolo estándar llevando a cabo lo siguiente:

1. Se invierte cada entrada y salida del símbolo estándar. Esto se hace agregando burbujas (círculos pequeños) en las líneas de entrada y salida que no tengan burbujas, y se remueven las que se encuentren allí.
2. Se cambia el símbolo de la operación de AND a OR, o de OR a NAND). (En el caso especial del INVERSOR, el símbolo de la operación no se cambia.)

Se deben destacar varios puntos con respecto a las equivalencias de los símbolos lógicos:

1. Las equivalencias se pueden extender a compuertas con cualquier número de entradas.
2. Ninguno de los símbolos estándar tiene burbujas en sus entradas, pero sí todos los símbolos alternos.

3. Los símbolos estándar y alternos para cada compuerta representan al mismo circuito físico: no hay diferencia en los circuitos que representan los dos símbolos,
4. Las compuertas NAND y NOR son compuertas de inversión, y por lo tanto, los símbolos estándar y alternos para cada una tendrán una burbuja, ya sea en la entrada o en la salida. Las compuertas AND y OR son compuertas no inversoras, por lo cual los símbolos alternos para cada una tendrán burbujas en las entradas y en la salida.

1.5 Manejo de las funciones básicas de programación del software PLC ZELIO

Entrar al programa ZELIO SOFT 2

Representación de las funciones básicas con programa PLC Zelio por diagrama de escalera.

Abrir con doble clic en el icono del programa Zelio desde escritorio

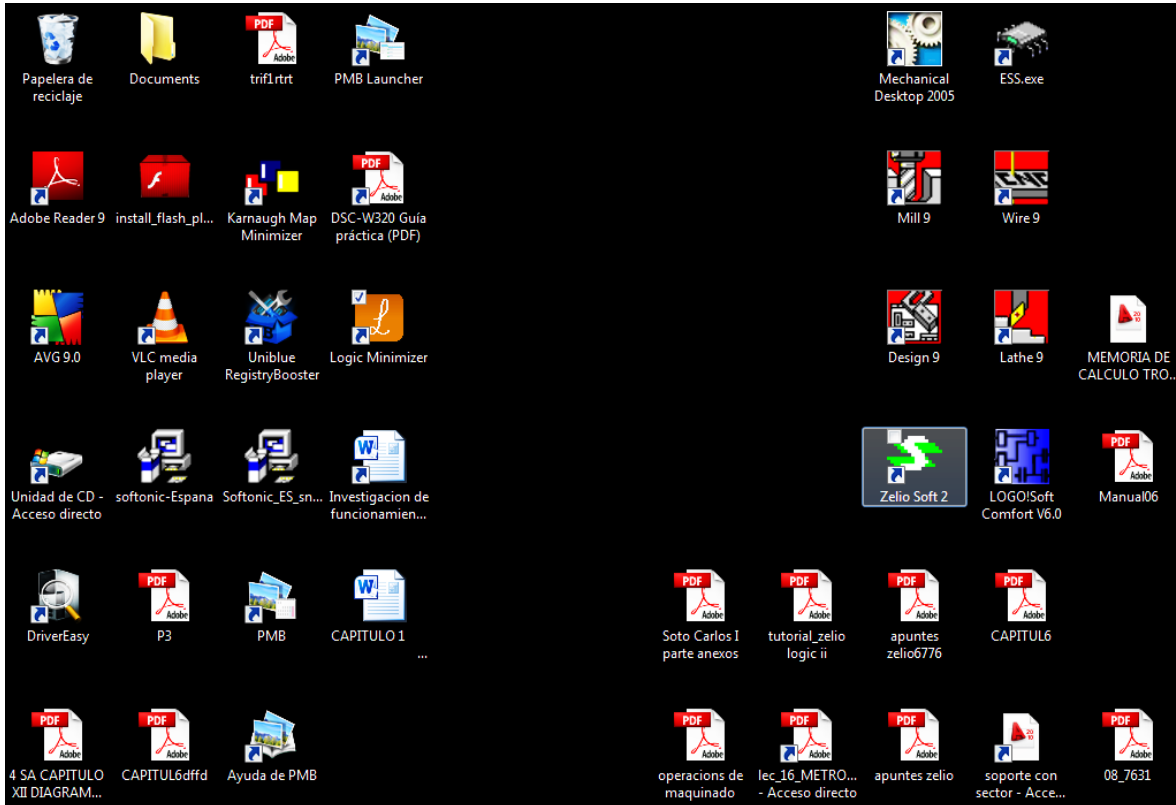


Fig 1.1

Abierto el programa pulsamos la pestaña de encabezado(Archivo),aparece un submenú pulsamos(Nuevo)

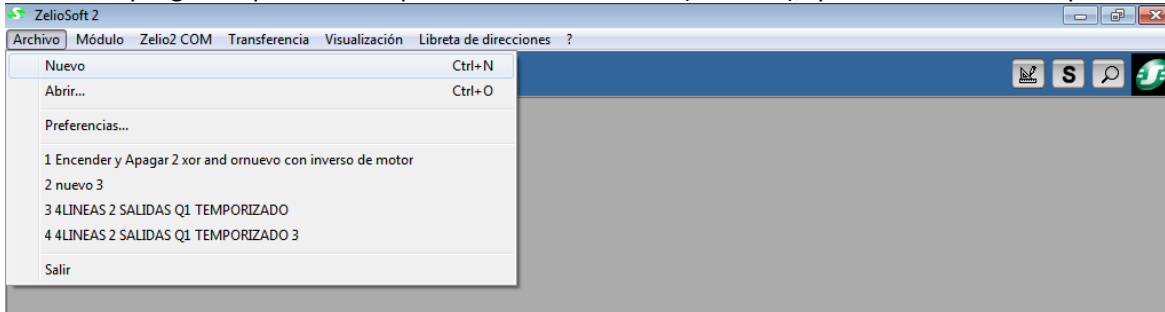


Fig 1.2

Se selecciona el tipo de módulo de serie del PLC con el que se está trabajando en este caso 6 entradas 4 salidas modelo SR3B101FU y en extensiones SR3XT1101FU.

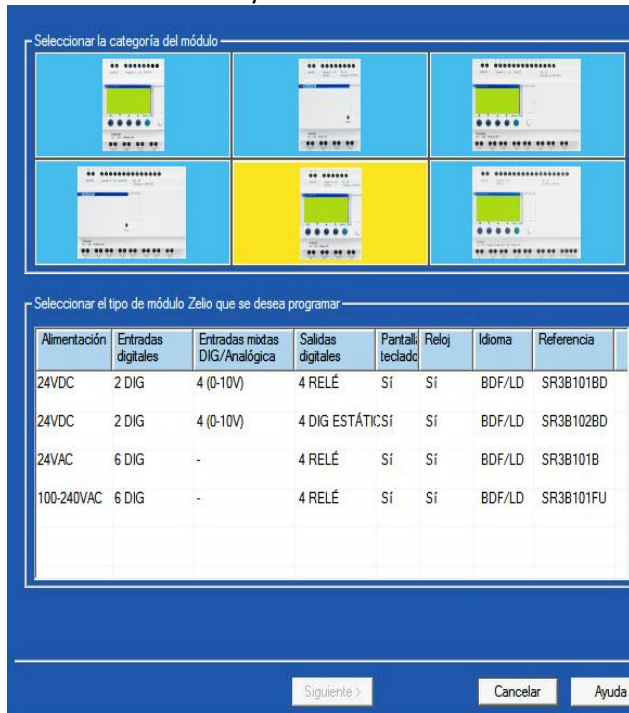


Fig 1.3

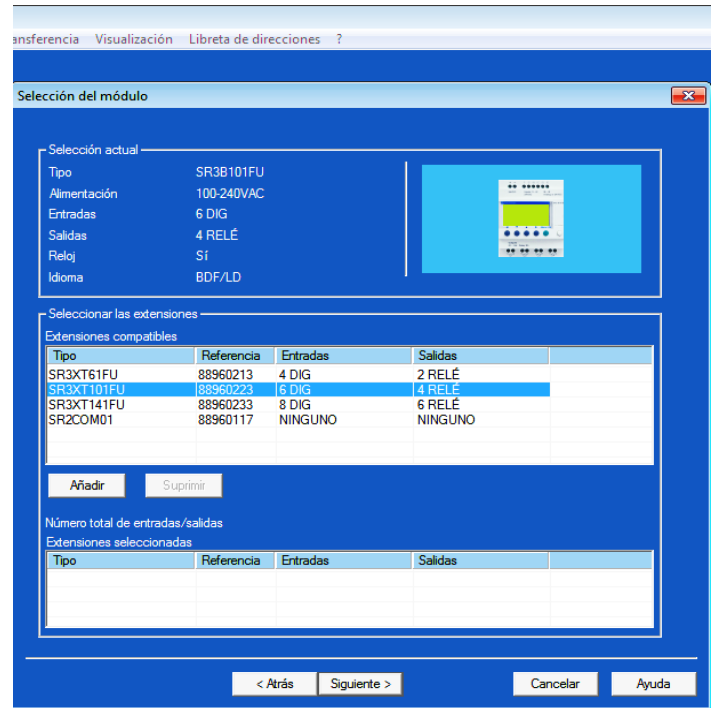


Fig 1.4

Posteriormente se selecciona el lenguaje con el que se quiere trabajar este programa dispone dos lenguajes uno de escalera y otro de diagrama de bloques lógicos para nuestro primer ensayo ocuparemos diagrama de escalera representando el funcionamiento de las funciones lógicas con sus entradas y salidas de respuesta como se muestra a continuación utilizando las funciones de diseño lógico (AND,NAND),(OR,NOR),NOT ,XOR.

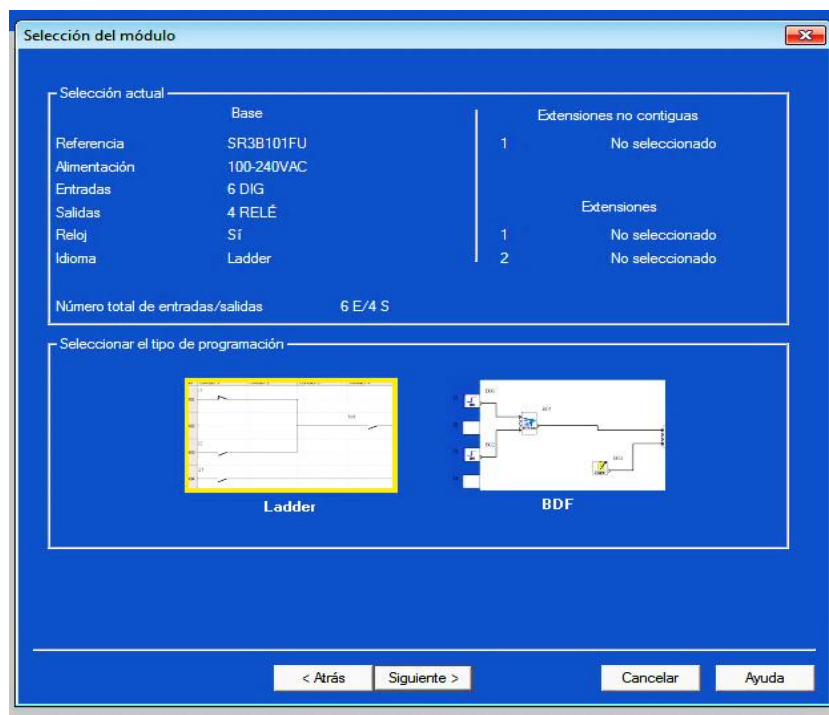


Fig 1.5

Seleccionando la ventana Ladder acceder a lenguaje de escalera o BDF lenguaje de bloques según sea el caso.

En la nueva ventana de este lenguaje de escalera tenemos del lado izquierdo extremo las líneas de conexión de franja gris de donde se conectan las entradas Dig (I) contactos de línea de seis entradas , que a su vez son controladas por el paso de corriente con los contactos y salidas DIG representadas en la pestaña de letra (Q) de 4 salidas las entradas y salidas Dig están representadas en el menú inferior extremo de la ventana en pestañas de color azul con las letras (I) y (Q) respectivamente, (M) representa relevadores auxiliares de Q1. Los Relés auxiliares marcados con una M se comportan exactamente igual que las Salidas Digitales (DIG)(Q), pero no poseen contacto eléctrico de salida. Se pueden utilizar como variables internas. Cualquier relé auxiliar se puede utilizar en el programa de forma indistinta como bobina o como contacto. Permiten memorizar un estado que se utilizará como el contacto asociado para las salidas(Q) mediante las entradas(I) al pulsar en el programa.

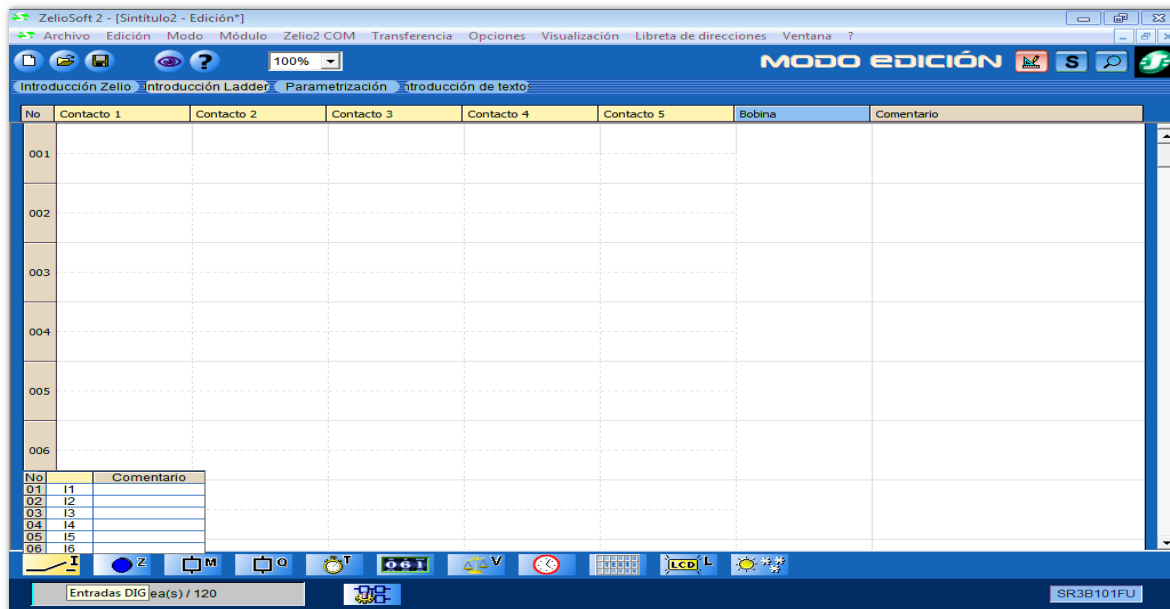


Fig 1.6

Cuando se está desarrollando un programa está activo en color rojo el botón con una escuadra y un lápiz en la parte superior derecha , cuando se termina el programa se simula apretando el botón adjunto descrito anteriormente representado por la letra (S) en negrita para evaluar el programa ,al pulsar este botón aparecen dos botones apagar y prender , RUN y STOP, para simular el programa en tiempo real se sacan las entradas y salidas Dig representadas por contactos pulsadores y focos de encendido y apagado que aparecen en unos cuadros con una bolita izquierda superior en la ventana del programa en la parte de en medio inferior del programa para ver la coherencia del programa de los parámetros establecidos se pulsa el botón representado por un ojo al lado del símbolo de interrogación en la parte superior izquierda del programa.

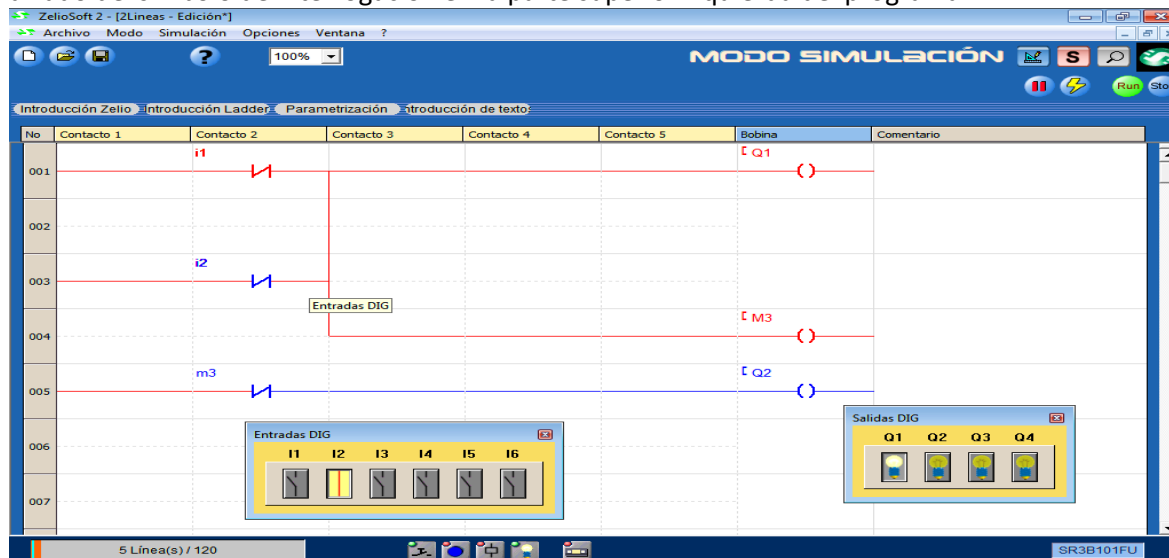


Fig 1.7

Transferencia del programa PC al modulo PLC ZELIO

Para cargar un programa al PLC se transfiere desde el menú (Transferencia) de la barra de comandos principales,

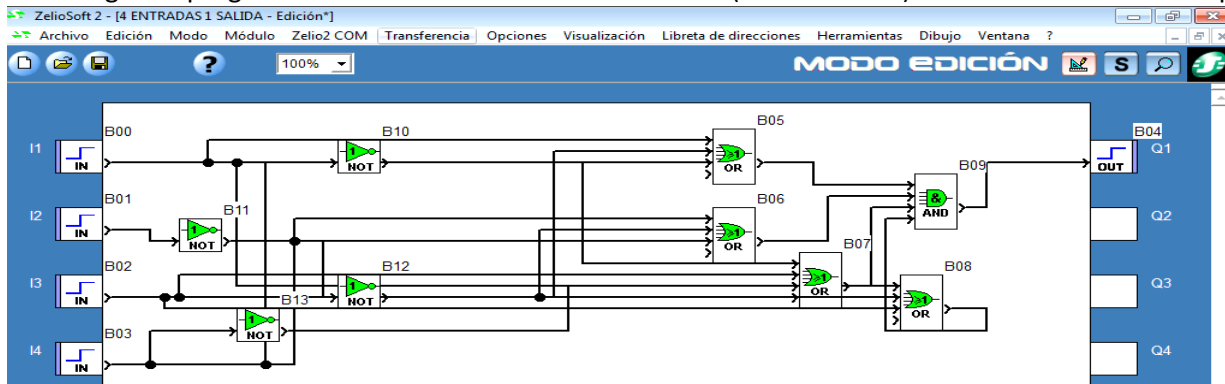


Fig 1.8

Se selecciona el submenú (Configuración de la comunicación) para seleccionar el puerto serial COM de la PC que sea compatible con el programa del PLC, se selecciona COM1,COM2,COM3,COM4 dependiendo de los puertos de la computadora por lo general se selecciona en automático el puerto dando clic en OK seleccionando el COM compatible.



Fig 1.9

Después de haber seleccionado el puerto COM de la interfaz entre el PLC y la PC se selecciona nuevamente el menú (Transferencia) se abre un submenú transferir programa se da clic en PC>Modulo para que reconozca el puerto USB de la PC al modulo PLC Zelio.

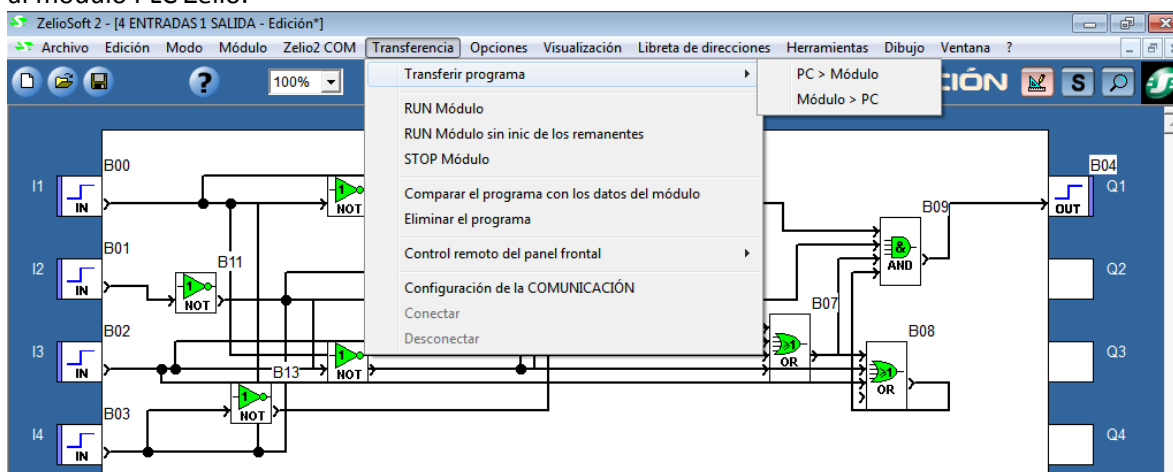


Fig 1.10

Después de dar clic en PC>Modulo aparece una ventana donde se da clic en Ok para que se transfiera el programa de la PC al módulo PLC se observa la transferencia en curso , se observa en la caratula del módulo físico parpadea la pantalla indicando que efectivamente se está transfiriendo el programa una vez transferido se retira el cable USB del puerto de la computadora y de la interface del módulo PLC.



Fig 1.11

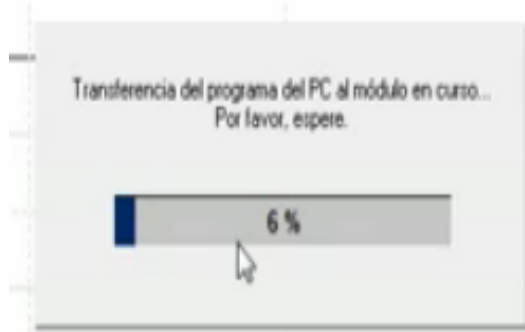


Fig 1.12

Diagrama eléctrico de Conexión PLC para control de compuertas Lógicas.

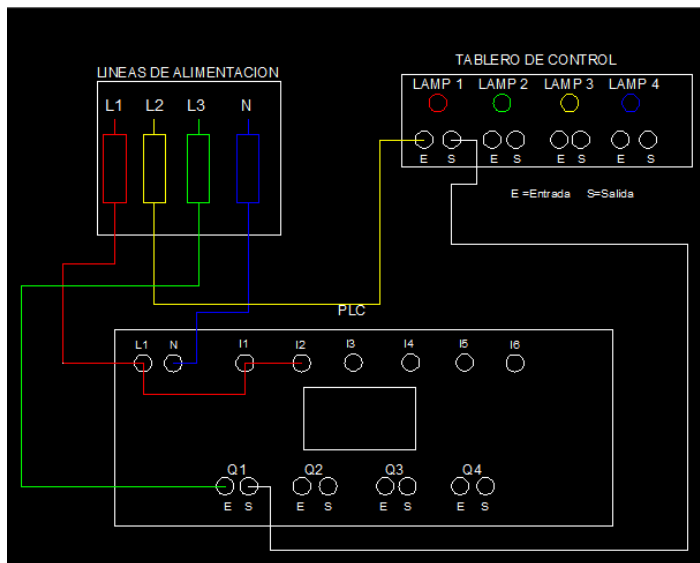
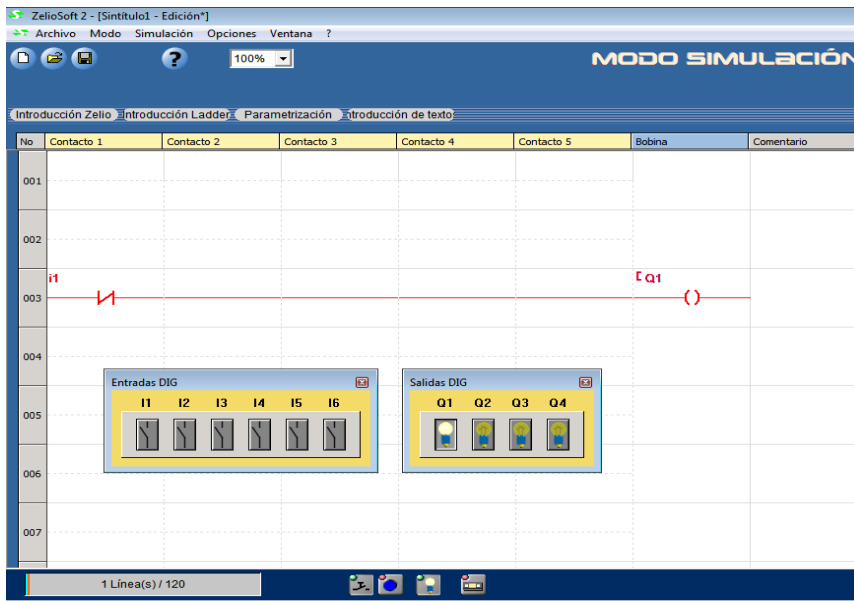


Diagrama 1.1

1.6 Funcionamiento de menús y compuertas lógicas en diagrama de escalera con PLC ZELIO

Compuerta NOT diagrama de escalera la compuerta NOT es un dispositivo de una entrada I1 y una salida Q1 que cumple con la condición que la salida toma el valor lógico negado de la entrada 0a1 y 1a0. Observación: Para cerrar contactos de entradas (I), relevadores auxiliares (M), salidas (Q) se da clic izquierdo sobre el símbolo del contacto y se selecciona normalmente cerrado o abierto.

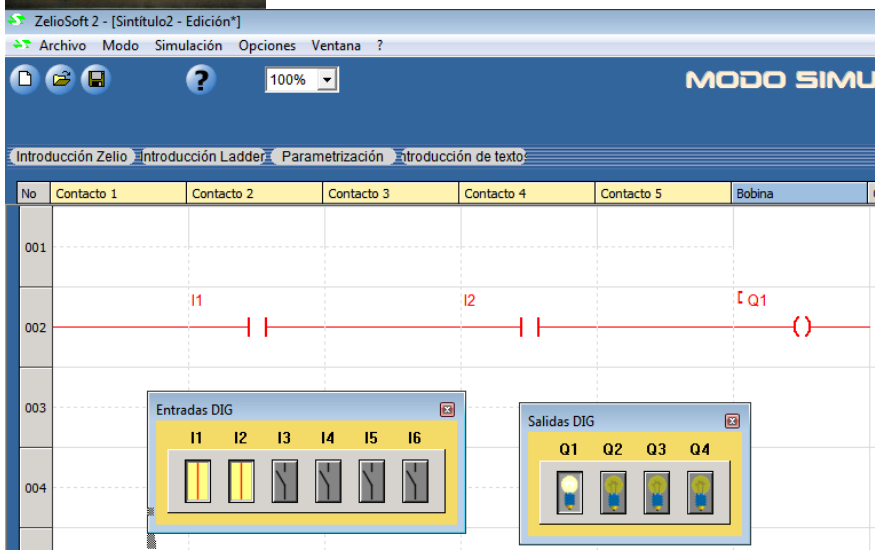


A	\bar{A}
0	1

Tabla 1

Programa 1.1

Compuerta AND representada en Zelio PLC diagrama de escalera esta compuerta funciona en serie opera como multiplicación por medio de dos condiciones en forma de señales por medio de contactos I1I2 al menos que los dos pulsos sean verdaderos se prende la salida representada por Q1 como bobina de salida.

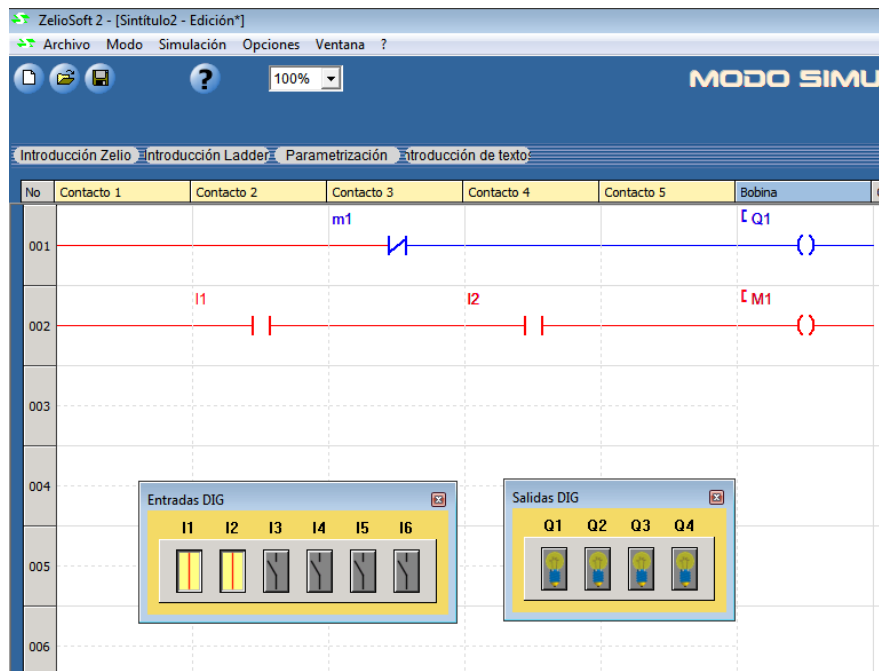


A	B	A·B
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 1.1

Programa 1.2

Compuerta NAND diagrama de escalera esta compuerta funciona igual que los valores de salida de la compuerta AND solo que va precedida por una compuerta inversora NOT delante de AND cambiando los valores de 0 a 1 y de 1 a 0, M1 representa una bobina de relevador auxiliar de salida invertida de los pulsos de I1xI2 pasando el pulso m1 que es el contacto de M1 a Q1() bobina .

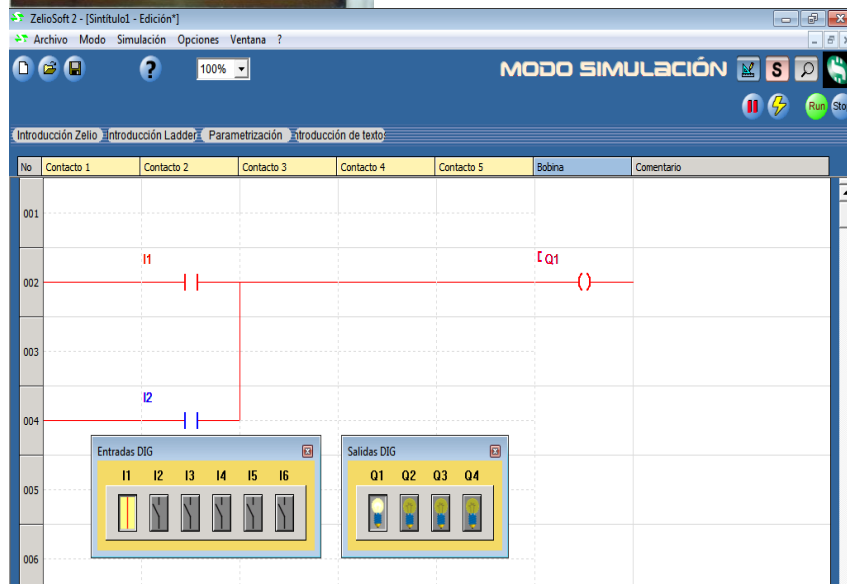


A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Programa 1.3

Tabla 1.2

Compuerta OR diagrama de escalera esta compuerta funciona en paralelo opera como suma por medio de dos condiciones con los contactos de entrada (I1+I2) al menos que una señal en el pulso sea verdadera en la salida vale 1 si en las dos es cero en la salida es cero ,si dos condiciones son 1 en la salida vale 1 representada por Q1() .

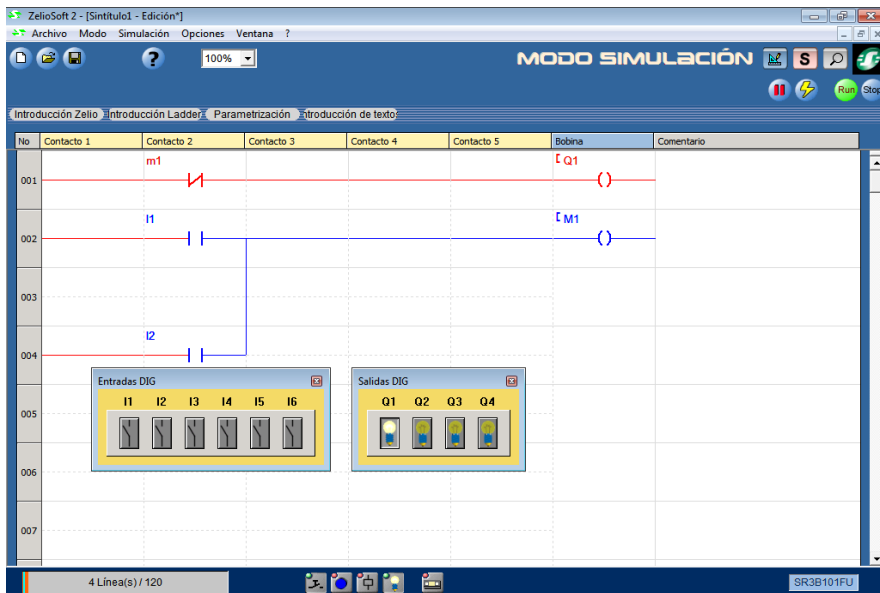


Programa 1.4

A	B	A+B
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 1.3

Compuerta NOR diagrama se escalera funciona igual que la compuerta OR en la salida tiene una compuerta inversora NOT teniendo como respuesta si alguno de los pulsos vale 1 en la salida vale 0 negado el valor vale 1 Q1, si las dos condiciones valen 1 en la salida valen 0 negado el valor vale 1 Q1, vale Q1 1 si las dos condiciones valen 0, M1 es una bobina auxiliar inversa de los contactos I1+I2 controlando el paso de la señal de Q1 con m1.

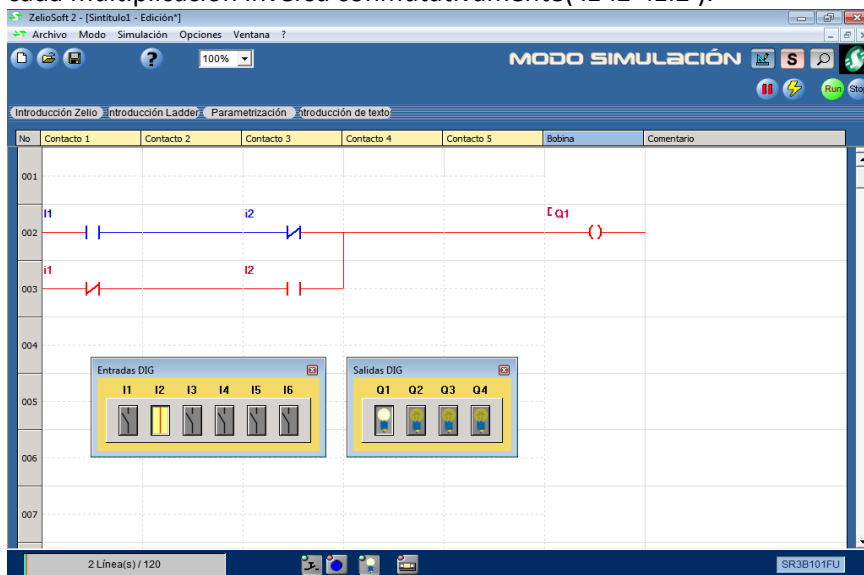


A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 1.4

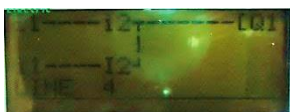
Programa 1.5

Compuerta XOR diagrama de escalera opera como una igualdad en donde una entrada I1 está inactiva y la otra entrada I2 está activa o desconectada, la salida está activa, si las dos entradas están activas, inactivas o desconectadas, la salida está inactiva representada por Q1. Al pulsar cualquiera de los dos contactos de entrada se activa el contacto controlador de la otra línea de entrada siendo diferentes los pulsos de control de cada entrada opera como la suma de dos multiplicaciones en donde dos de las variables son negadas en las multiplicaciones en cada multiplicación inversa conmutativamente ($I1'I2+I1I2'$).



A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 1.5



Programa 1.6

1.7 Funcionamiento de menús y compuertas lógicas en diagrama de bloques con PLC ZELIO

Ventana de diagrama de Bloques ventana del lado izquierdo representadas I1 a I6 se colocan las entradas del lado derecho Q1 a Q6 se colocan las salidas del programa.

Acceso

La función Entrada Todo o Nada (DIG) se encuentra disponible en la barra de funciones ENT representan la entrada L por IN tipo visualización en el estado Inactivo visualización en el estado Activo Entrada DIG Botón pulsador contacto.

Para realizar un programa BDF, las diferentes funciones que se insertarán en la hoja de cableado están disponibles en una barra de funciones. En cada una de las fichas de la barra de funciones se agrupa un tipo de función.

Cuando el ratón pasa sobre alguna de las fichas, el cuadro de diálogo muestra la lista de las variables disponibles estas variables son funciones predeterminadas con funciones específicas que contiene temporizadores, contadores comparadores son las más utilizadas entre otras representan funciones estándar predeterminadas.

La barra LOGIC contiene las compuertas lógicas (AND, NAND, OR, NOR, NOT, XOR) para diseño lógico de programas. Barra de funciones de las salidas SAL contiene las compuertas de salida los módulos lógicos cuentan con dos tipos de salidas DIG:

Las salidas estáticas para algunos módulos lógicos alimentados con una tensión continua.

Las salidas de relé para los módulos lógicos alimentados con una tensión alternativa o continua.

La visualización de los menús de comandos ENT, BDF, LOGIC, SAL se ubican en la parte inferior izquierda representadas por las casillas de color azul.

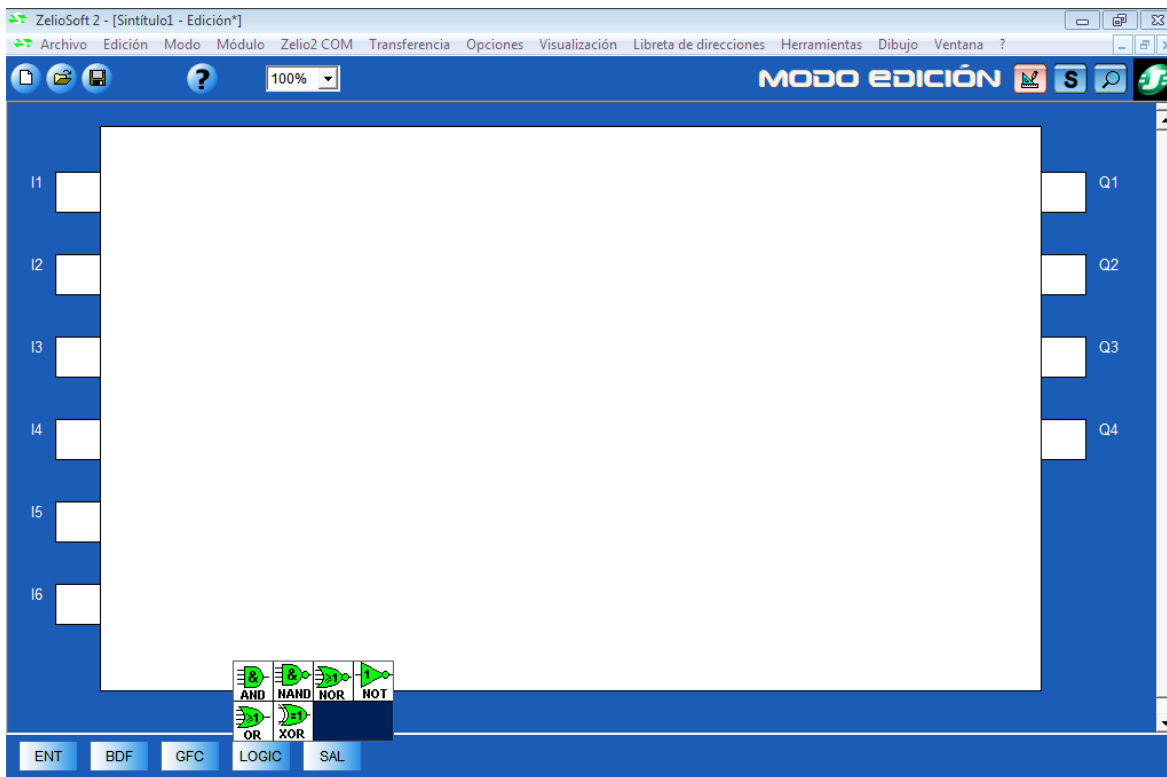
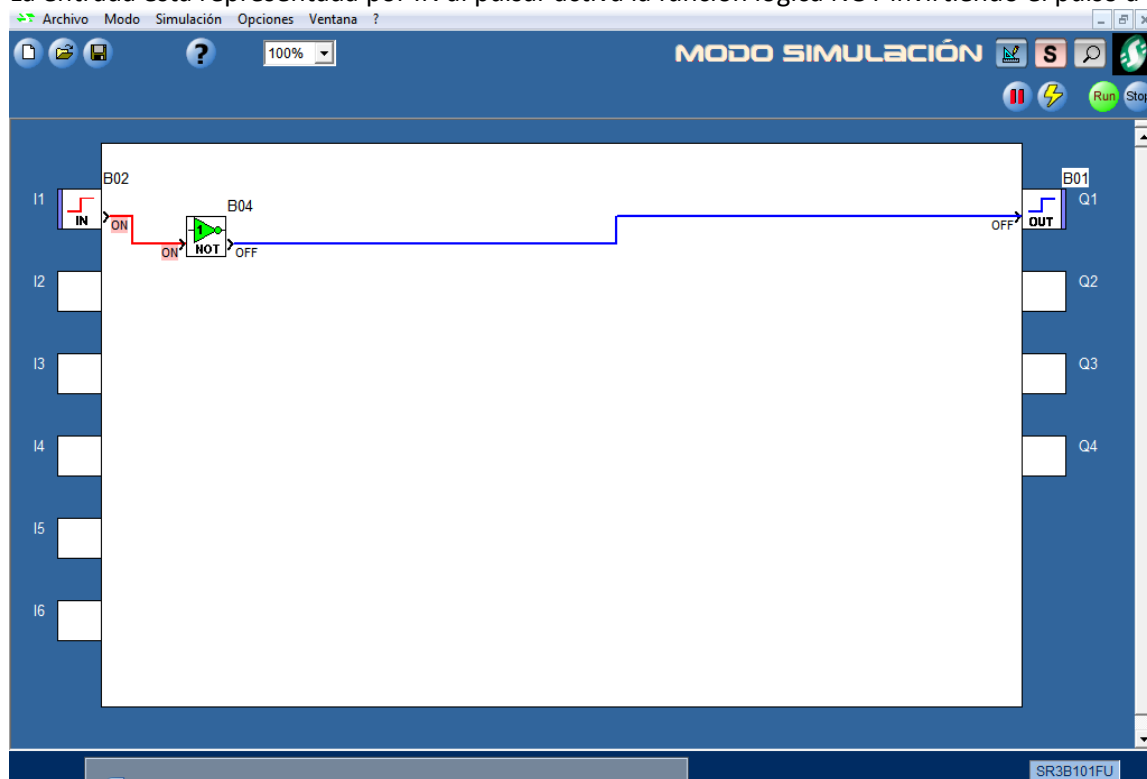


Fig 1.13

Función NOT diagrama de bloques.

La entrada esta representada por IN al pulsar activa la función lógica NOT invirtiendo el pulso a la salida OUT =Q1().



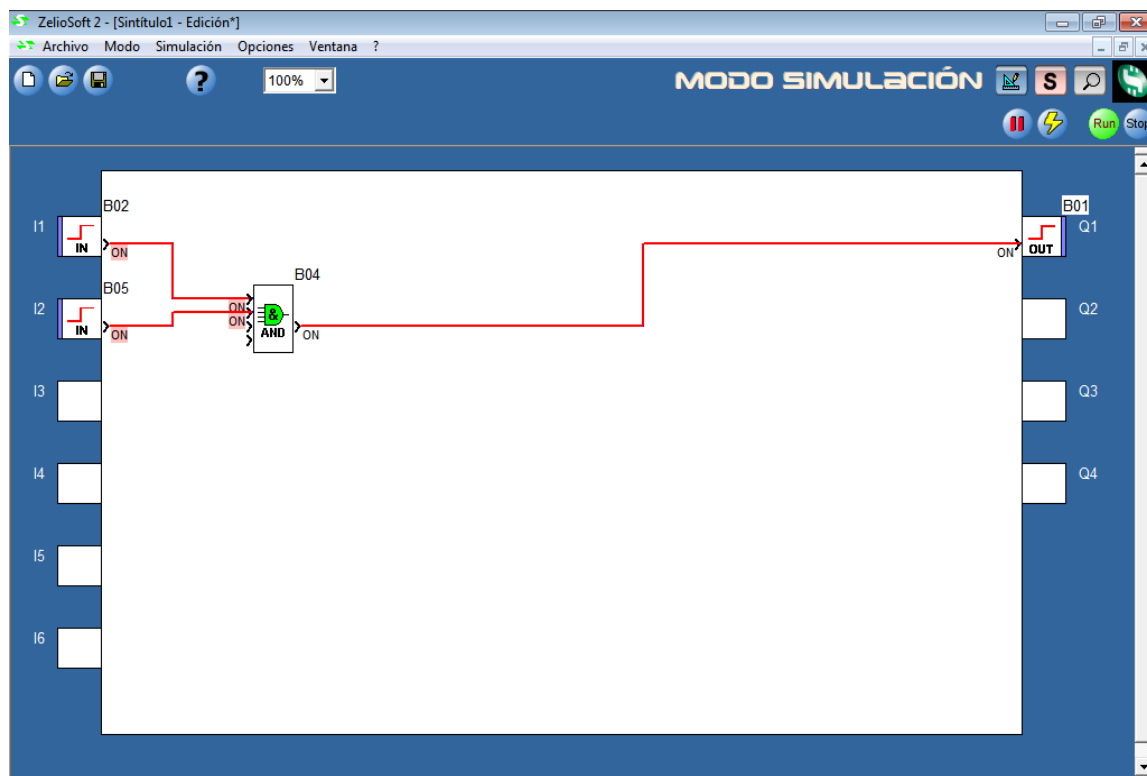
Programa 1.7

A	\bar{A}
0	1

Tabla 1

Función AND diagrama de bloques.

Las entradas están activadas por I1X12 por medio de los pulsadores IN conectados a la compuerta AND con salida Q1.



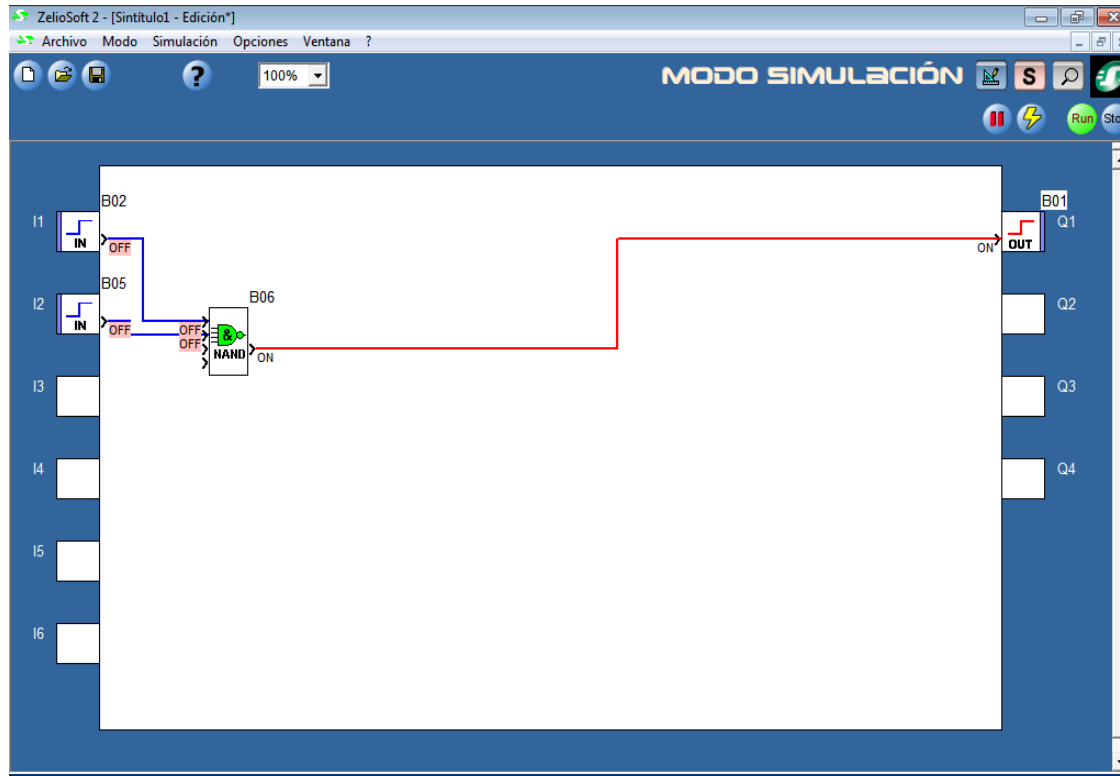
Programa 1.8

A	B	A-B
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 1.1

Función NAND diagrama de bloques.

Entradas activadas por I1X12 conectados a la función NAND con salida Q1.



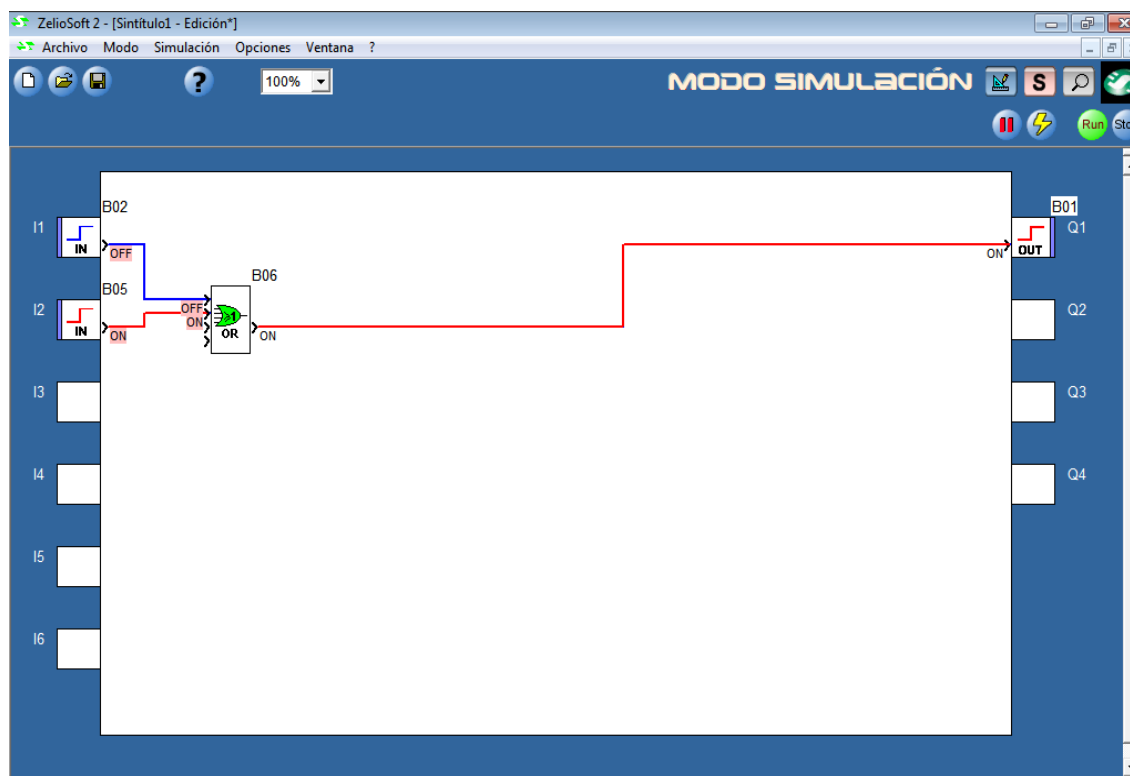
Programa 1.9

A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 1.2

Función OR diagrama de bloques.

Entradas activadas I1+I2 conectadas a compuerta OR con salida Q1.



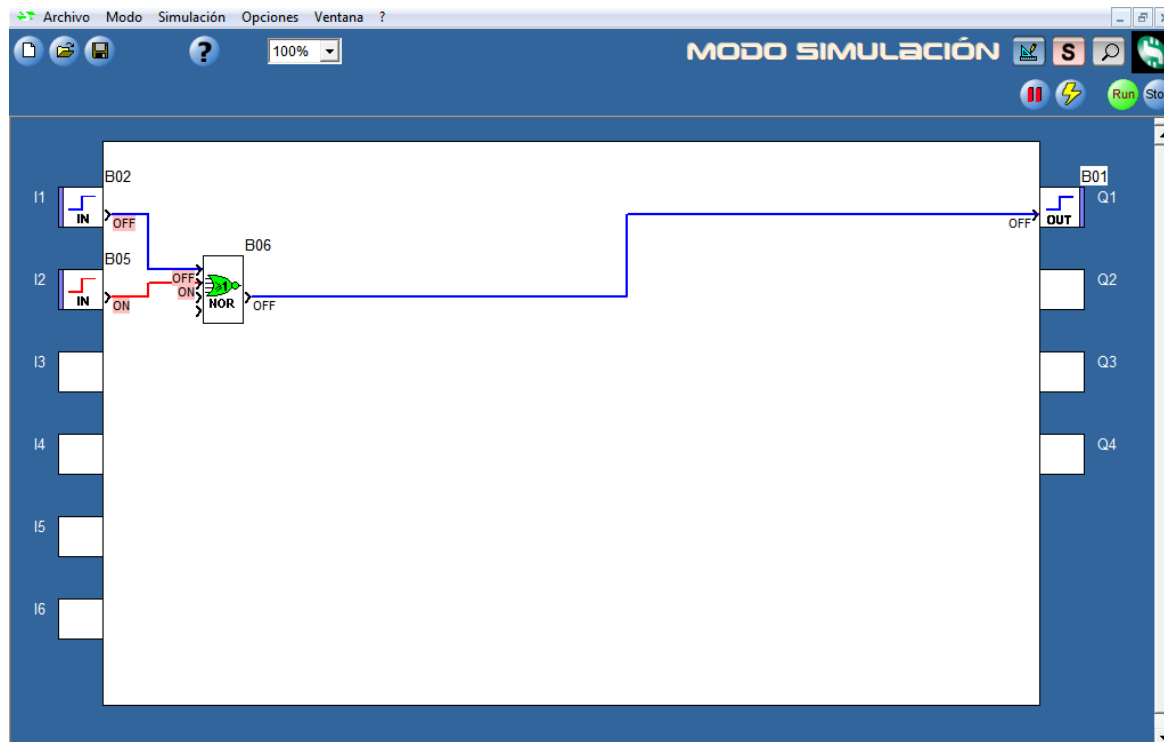
Programa 1.10

A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 1.3

Función NOR diagrama de bloques.

Entadas activas I1+I2 conectadas a la compuerta NOR con salida Q1.



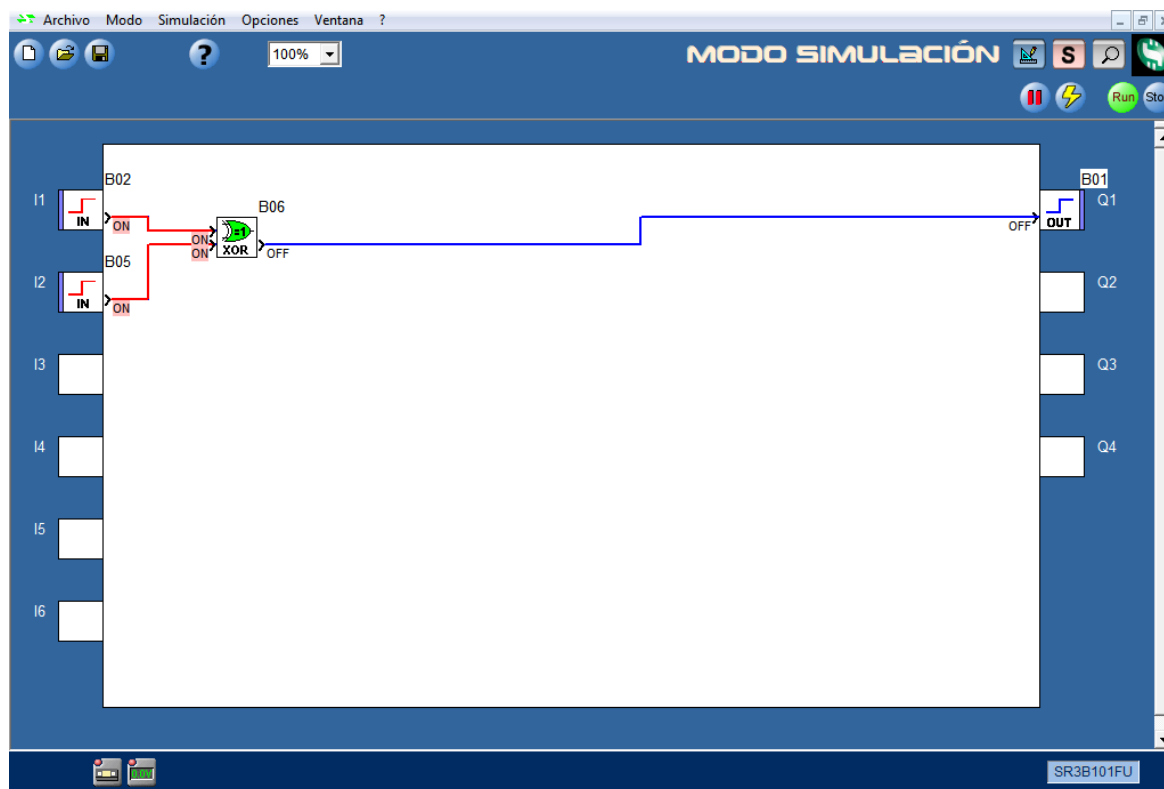
Programa 1.11

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 1.4

Funcion XOR diagrama de bloques.

Entradas activas I1,I2 con respuesta lógica XOR= (I1I2+I1I2) con salida Q1.



Programa 1.12

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 1.5

1.8 Introducción a las tablas lógicas.

La tabla de verdad es un sistema combinatorio de entradas representadas por (I) y una o más salidas representadas por Q(), el PLC no se le puede introducir la tabla de verdad en ese estado binario ya que la tabla tiene que ser simplificada o reducida a un lenguaje de compuertas lógicas que funciona con las compuertas AND, NAND, NOT, OR se hace la reducción de las tablas lógicas mediante los métodos algebraicos Karnaugh simplifica hasta seis variables en la entrada o Mc Cluskey resuelve de 5 variables en adelante, el resultado de estos métodos es una relación algebraica en una ecuación en donde la tabla a resolverse de cada método se resuelve agrupando ceros y unos, al agrupar ceros es por mini términos llamado también sumas de productos o agrupar unos es por maxiterminos o productos de sumas. En este ensayo no se enfatizara en la resolución de estos métodos solo se verá la relación del resultado simplificado algebraico de cada tabla lógica aplicado a los ejemplos de control citados ya que el conocimiento de resolución esta intrínseco o visto a lo largo de la carrera de IME en materias de Algebra 1,2,3 y una materia del área mecánica Electrónica Industrial en donde se sientan las bases y demostraciones de mencionados métodos, así como aparte hay diversos programas computacionales que los resuelven ahorrando tiempo de resolución en donde el Ingeniero solo se dedica al diseño de control de las variables de entrada y salida de un sistema eléctrico automatizado, no obstante al final del ensayo viene un CD de aplicaciones con programas que resuelven dichos métodos de fácil introducción y control que son explicados en el demo del CD para hacer más fácil la comprensión de esta tesis.

1.9 Aplicación de ejemplo de compuertas lógicas de la tabla 1.6

Aplicación de ejemplo de tablas lógicas con PLC ZELIO

Representación de una tabla de verdad de un sistema binario de 4 entradas I y una salida Q1 de valor arbitrario la finalidad de esta tabla es representarla en funcionamiento con el PLC ZELIO de tablas lógicas operadas con compuertas lógicas, el primer punto fue hacer el diseño de una tabla lógica de n entradas n salidas, para después obtener la ecuación algebraica simplificada y convertirla al lenguaje del PLC por diagrama de bloques o diagrama de escalera en el lenguaje del PLC con compuertas lógicas.

	D	C	B	A	
	I4	I3	I2	I1	Q1
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	1

Tabla 1.6

Se hace el análisis de transformación de la ecuación simplificada a lenguaje del programa PLC ZELIO Representación del resultado de la tabla de verdad anterior 4I, Q1 por suma de productos.

Se puede apreciar el resultado algebraico de la tabla de verdad con las 4 variables acomodadas según el algebra de resolucion equivalentes a $I1I2I3I4 = ABCD$, los corchetes [] significan agrupacion de una multiplicacion o suma dentro y fuera de los corchetes en este caso los corchetes representan la compuerta AND= [],el signo + representa la compuerta OR de suma +=OR ,el signo * representa un producto interno de dos variables de entrada *=x ,el signo ~ representa la negacion de una compuerta de entrada negada NOT ~≠ negacion.

Ecuacion simplificada de la trabla logica.

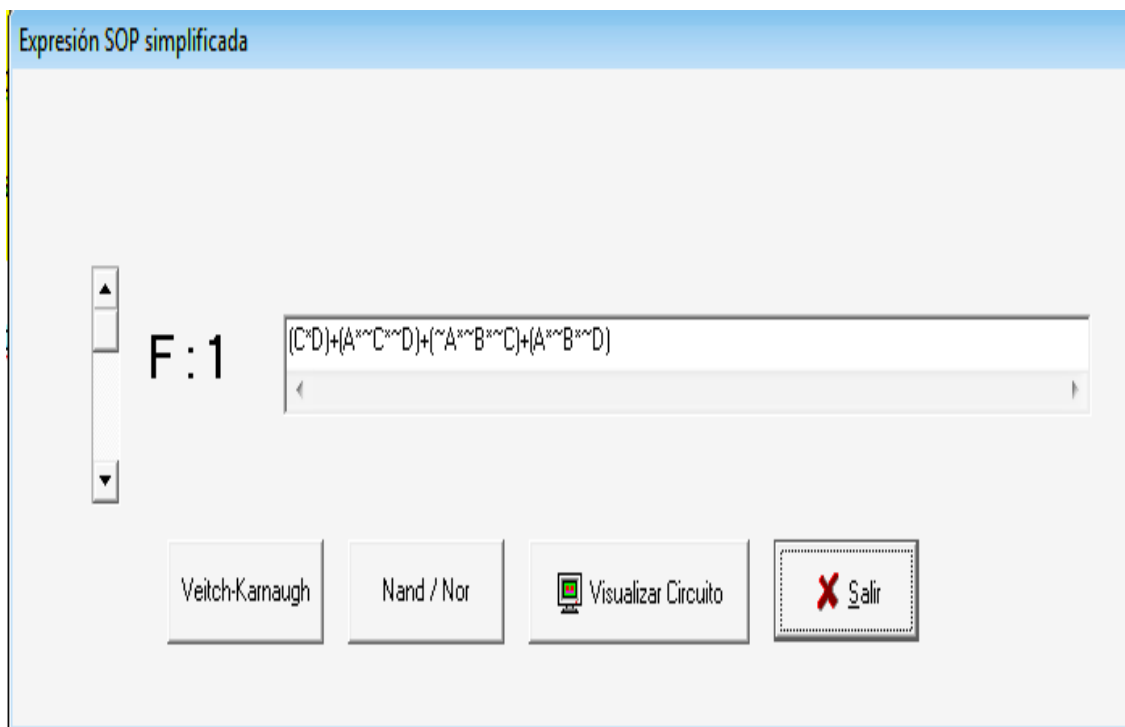
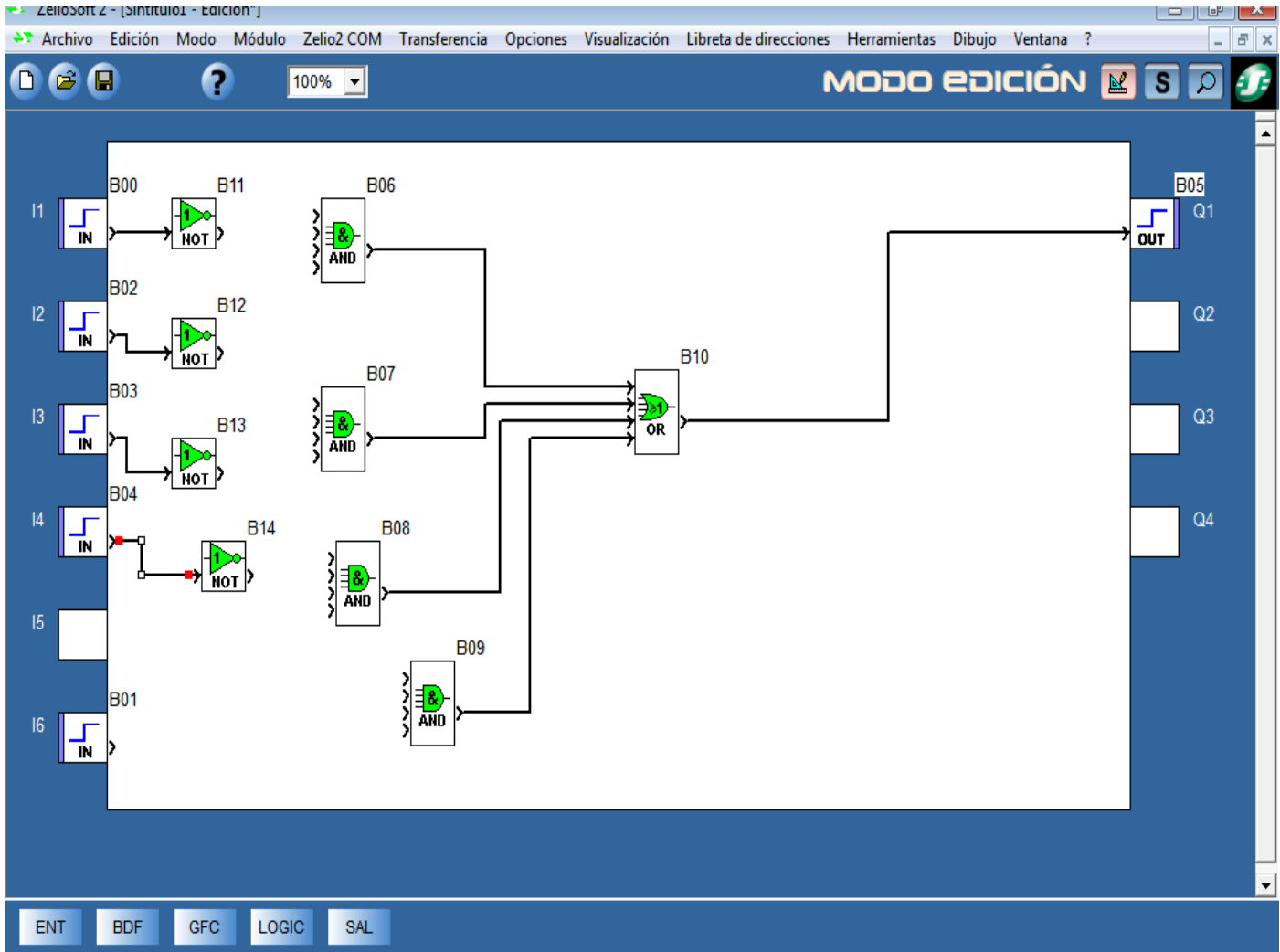


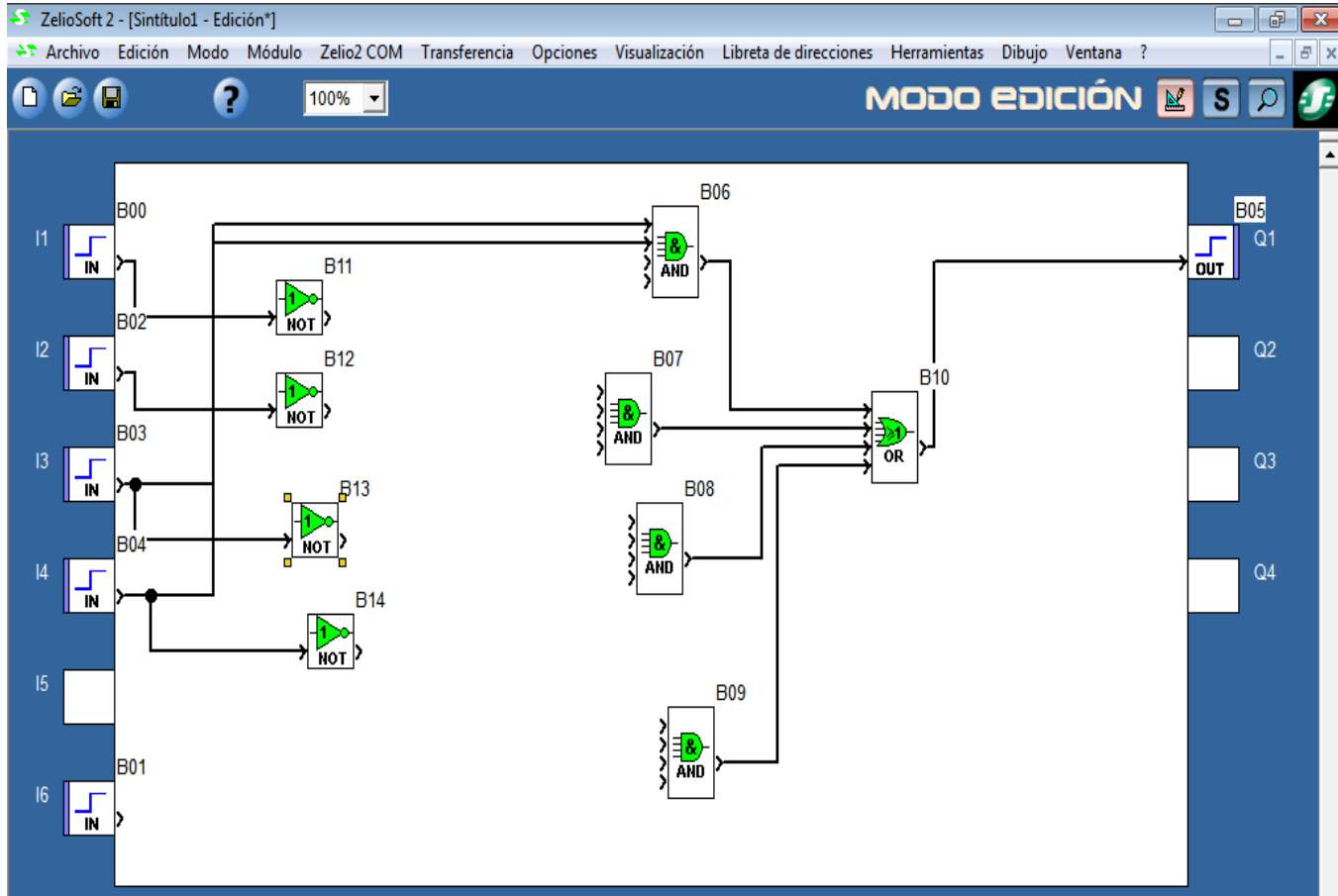
Fig 1.14

Una vez simplificada la tabla logica de verdad la ecuacion algebraica se convierte a diagrama de bloques o escalera al programa PLC ZELIO para su diseño y evaluacion en el programa del PLC para despues transferirlo al modulo fisico PLC. Para representar la ecuacion $[C * D] + [A * \sim C * \sim D] + [\sim A * \sim B * \sim C] + [A * \sim B * \sim D]$ Se sigue la secuencia original de las lineas de entrada I1I2I3I4=ABCD en el plc y como en la tabla de verdad en la entrada sin alterar su orden , despues se fija en la ecuacion que variable de letra tiene un negador se coloca un NOT delante de la variable negada, despues por medio de los corchetes se ven cuantos corchetes hay representando la compuerta AND se colocan en el esquema, se cuentan las sumas en este caso la compuerta OR tiene recepciones para 4 sumas de productos solo se utiliza una compuerta OR en el caso de las compuertas AND en este caso varian las entradas de 2 a 3 representadas dentro de cada corchete.



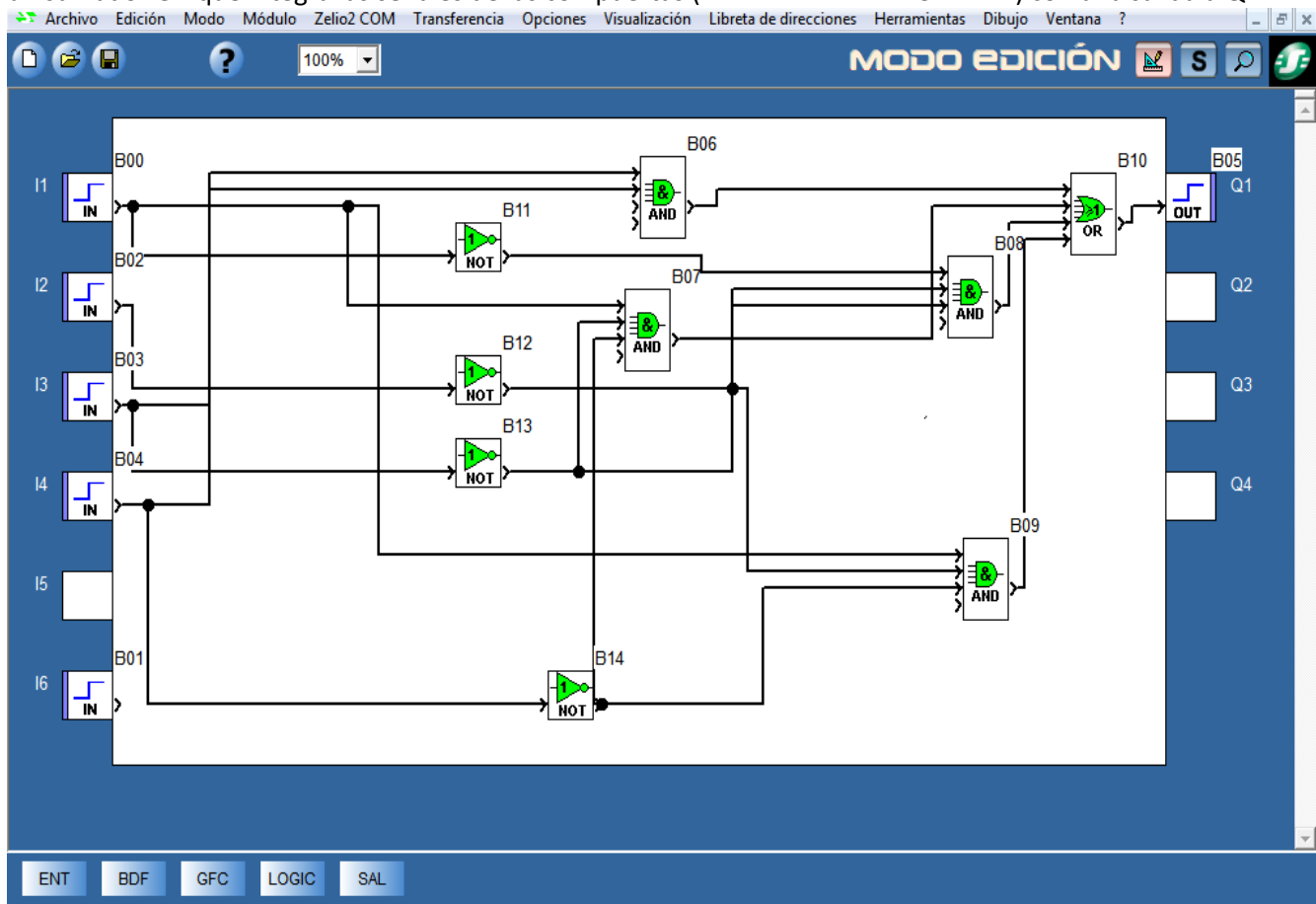
Programa 1.13

Relacionando las variables en el primer corchete se muestran las variables C y D como I3 y I4 multiplicadas se colocan en la primera compuerta AND, la variable C se jala desde la terminal IN de I3 a la compuerta AND la variable D se jala de la terminal IN de I4 a la primera AND, si una variable o variables están negadas en la ecuación se jalan desde la terminal NOT de cada variable I negada cabe destacar que de las terminales IN o líneas I y compuertas NOT su salida se pueden sacar muchas conexiones indefinidas a las compuertas lógicas dependiendo de la ecuación resultante. De esta misma manera se relacionan los demás corchetes restantes de productos con las compuertas AND que al final son sumadas por la compuerta OR con salida Q1.



Programa 1.13.1

Diagrama final de conexión de la tabla 4 líneas 1 salida en diagrama de boques PLC ZELIO en donde la segunda compuerta AND tiene las 3 variables multiplicadas ($Ax\sim Cx\sim D$)= $(I1X\sim I3X\sim I4)$ con sus respectivas salidas IN-NOT, la tercera compuerta AND ($\sim Ax\sim Bx\sim C$)= $(\sim I1x\sim I2x\sim I3)$ y la cuarta compuerta AND ($Ax\sim Bx\sim D$)= $(I1x\sim I2x\sim I4)$ las compuertas AND se suman por un sumador OR que integra las señales de las compuertas (AND1+AND2+AND3+AND4) con una salida a Q1.



Programa 1.13.2

Representación de la tabla 4I,Q1 por productos de suma $[A+\sim C+D] * [\sim B+\sim C+D] * [\sim A+C+\sim D] * [A+\sim B+C]$ en este caso las variables se alinean con las variables como en el anterior caso y se colocan negadores NOT en donde las variables ABCD sean negadas delante de sus entradas I1I2I3I4=ABCD

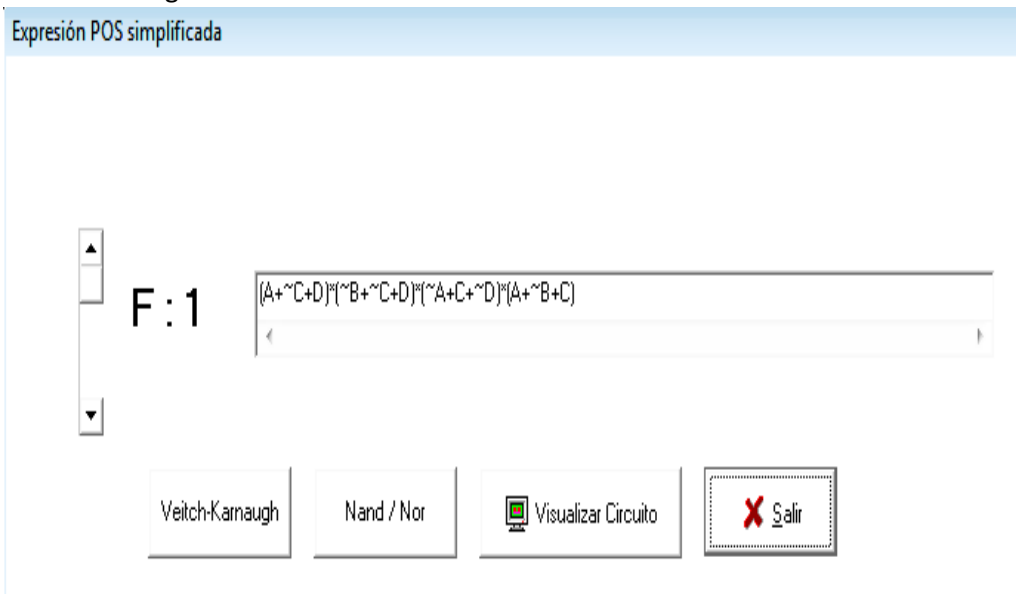
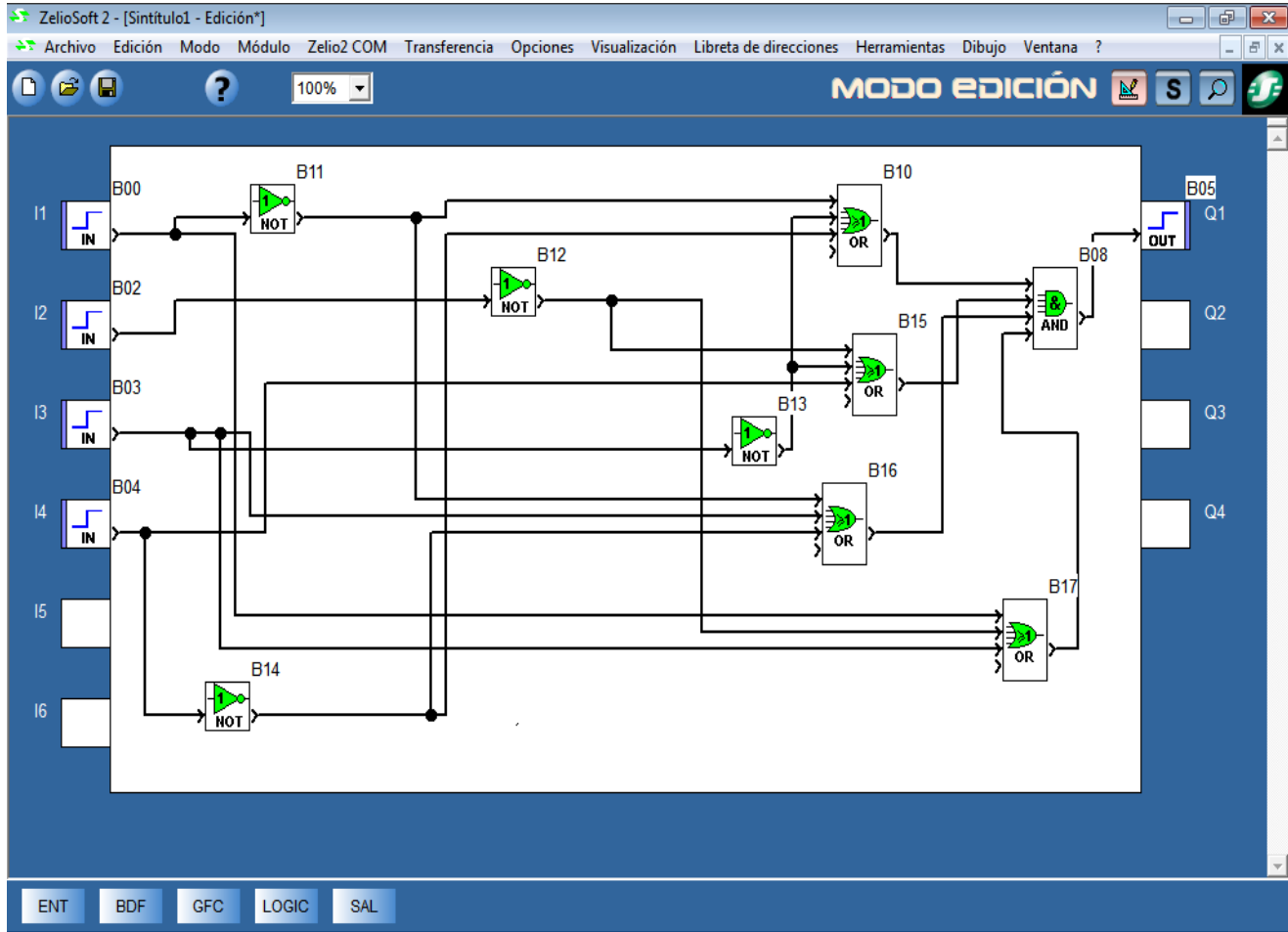


Fig 1.15
33

En este arreglo hay 4 compuertas OR sumadoras que se conectan a una sola compuerta AND que integra todas las señales de I1I2I3I4.



Programa 1.14

La tabla de verdad se cumple al pulsar los botones de entrada de línea (I1,I2,I3,I4) del modulo fisico del PLC equivalen a (I1I2I3I4) de la tabla logica con la salida Q1 que puede mandar la señal a una lampara,un mot or,actuador,etc.

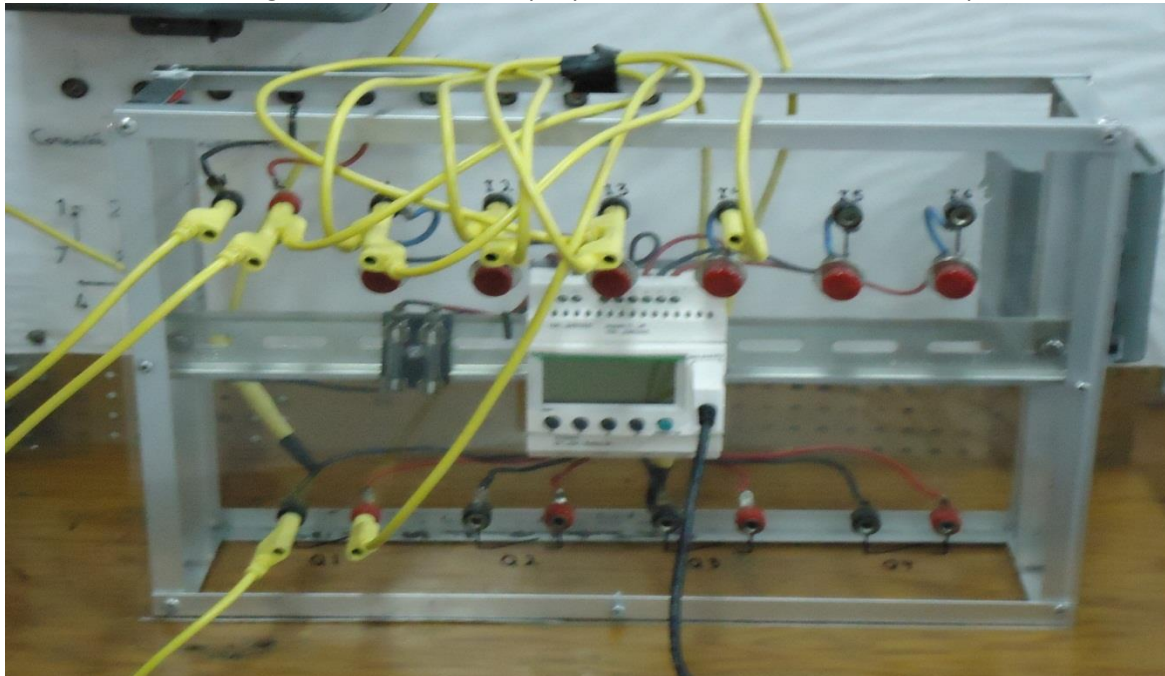


Fig 1.16

Diagrama de Alimentación y conexión del PLC de la tabla lógica 1.6 4I,Q1. En el lenguaje de diagrama de bloques con compuertas lógicas.

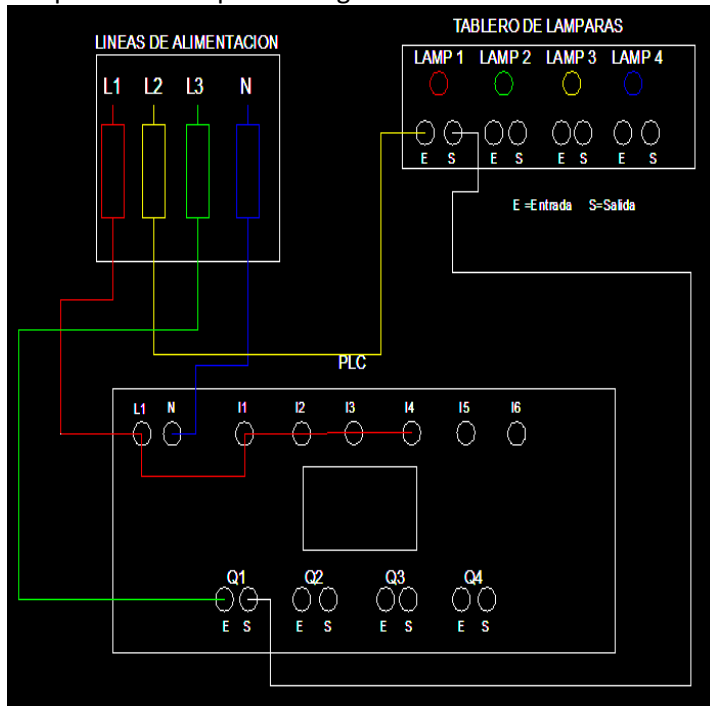


Diagrama 1.2

1.10 Aplicación de ejemplo de compuertas lógicas de la tabla 1.7

Ejemplo tabla lógica 1.7

En este ejemplo se aplica una aplicación en específico ya que se diseña un sistema de alarma para las dos puertas delanteras de un automóvil detectando con 1 en las salidas Q1,Q2 las condiciones de peligro para las dos puertas. Se encuentran 5 señales de activación de los sensores a las salidas de Q1,Q2 mediante los faros, las dos puertas delanteras y la cajuela.

La alarma debe de activarse cuando se presentan cualquiera de las siguientes condiciones.

- 1-Los faros están prendidos mientras el encendido no esta funcionando
- 2-La puerta está abierta mientras el encendido está funcionando.La indicación es independiente para cada puerta y se debe mostrar en el panel frontal.
- 3-Cuando este la cajuela abierta mientras el encendido esta funcionando

Primero se definen las variables de entrada y salida del circuito:

I5=Encendido(1 activado,0 desactivado)

I4=faros(1 encendido,0 apagados)

I3=puerta izquierda(1abierta,0 cerrada)

I2=puerta derecha(1 abierta,0 cerrada)

I1=cajuela(1 abierto, 0 cerrada)

Q1=alarma puerta izquierda(1 activada,0desactivada)

Q2=alarma puerta derecha(1 activada, 0 desactivada)

Tabla funcional: Tomando en cuenta las variables de entrada y salida y las condiciones para activarse las alarmas, se realiza la tabla funcional:

Dec	Encendido I5	Faros I4	Puerta izq I3	Puerta der I2	Cajuela I1	Alarma Q1	Alarma Q2
0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0
2	0	0	0	1	0	0	0
3	0	0	0	1	1	0	0
4	0	0	1	0	0	0	0
5	0	0	1	0	1	0	0
6	0	0	1	1	0	0	0
7	0	0	1	1	1	0	0
8	0	1	0	0	0	1	1
9	0	1	0	0	1	1	1
10	0	1	0	1	0	1	1
11	0	1	0	1	1	1	1
12	0	1	1	0	0	1	1
13	0	1	1	0	1	1	1
14	0	1	1	1	0	1	1
15	0	1	1	1	1	1	1
16	1	0	0	0	0	0	0
17	1	0	0	0	1	1	1
18	1	0	0	1	0	0	1
19	1	0	0	1	1	1	1
20	1	0	1	0	0	1	0
21	1	0	1	0	1	1	1
22	1	0	1	1	0	1	1
23	1	0	1	1	1	1	1
24	1	1	0	0	0	0	0
25	1	1	0	0	1	1	1
26	1	1	0	1	0	0	1
27	1	1	0	1	1	1	1
28	1	1	1	0	0	1	0
29	1	1	1	0	1	1	1
30	1	1	1	1	0	1	1
31	1	1	1	1	1	1	1

Tabla 1.7

Se observa que las combinaciones de entrada 0-7, el encendido y los faros están apagados por tanto las alarmas están desactivadas. Para las combinaciones 8-14, el encendido está apagado pero están encendidos los faros, por tanto se activan las dos alarmas. Para las combinaciones 16-31 el encendido está activado y entonces se debe considerar las distintas condiciones para que se active una o las dos alarmas.

Después de definir las variables del sistema se resuelve la tabla lógica a su expresión algebraica simplificada en este caso se resolvió por suma de productos.

Expresion simplificada para Q1

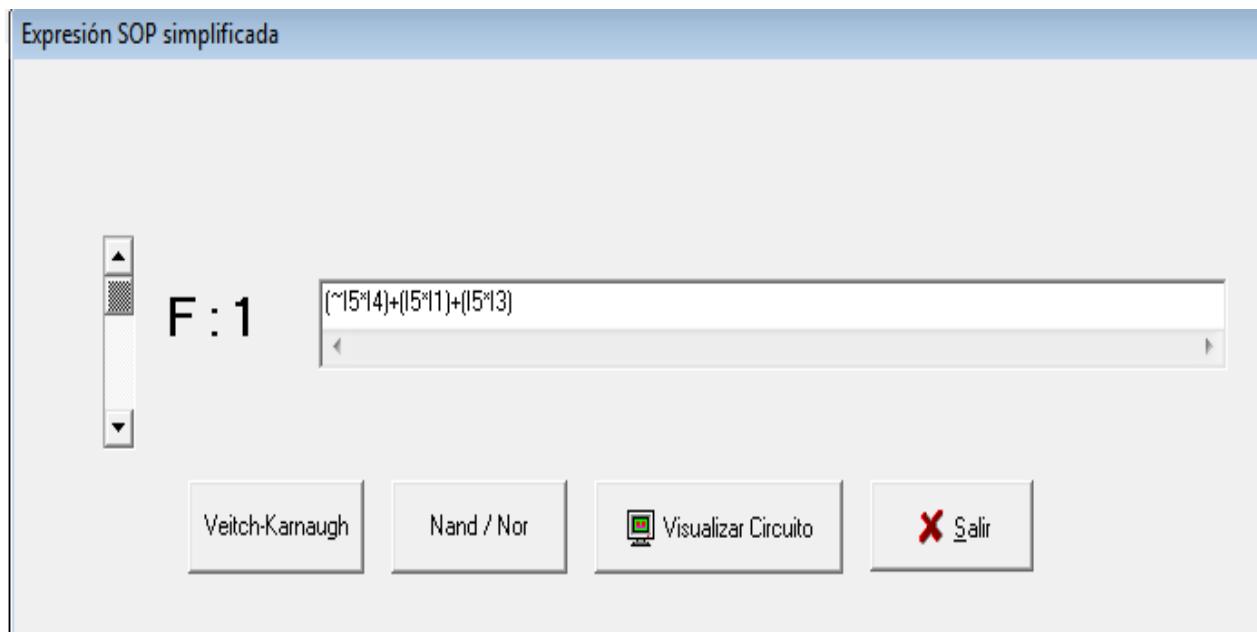


Fig 1.17

Expresion simplificada para Q2

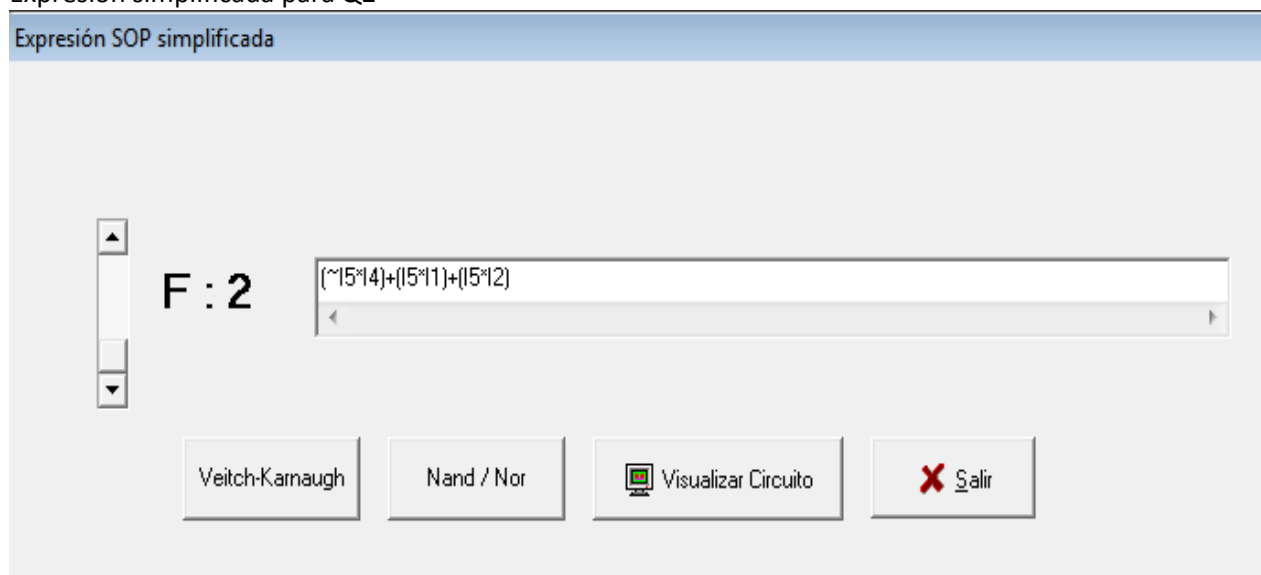
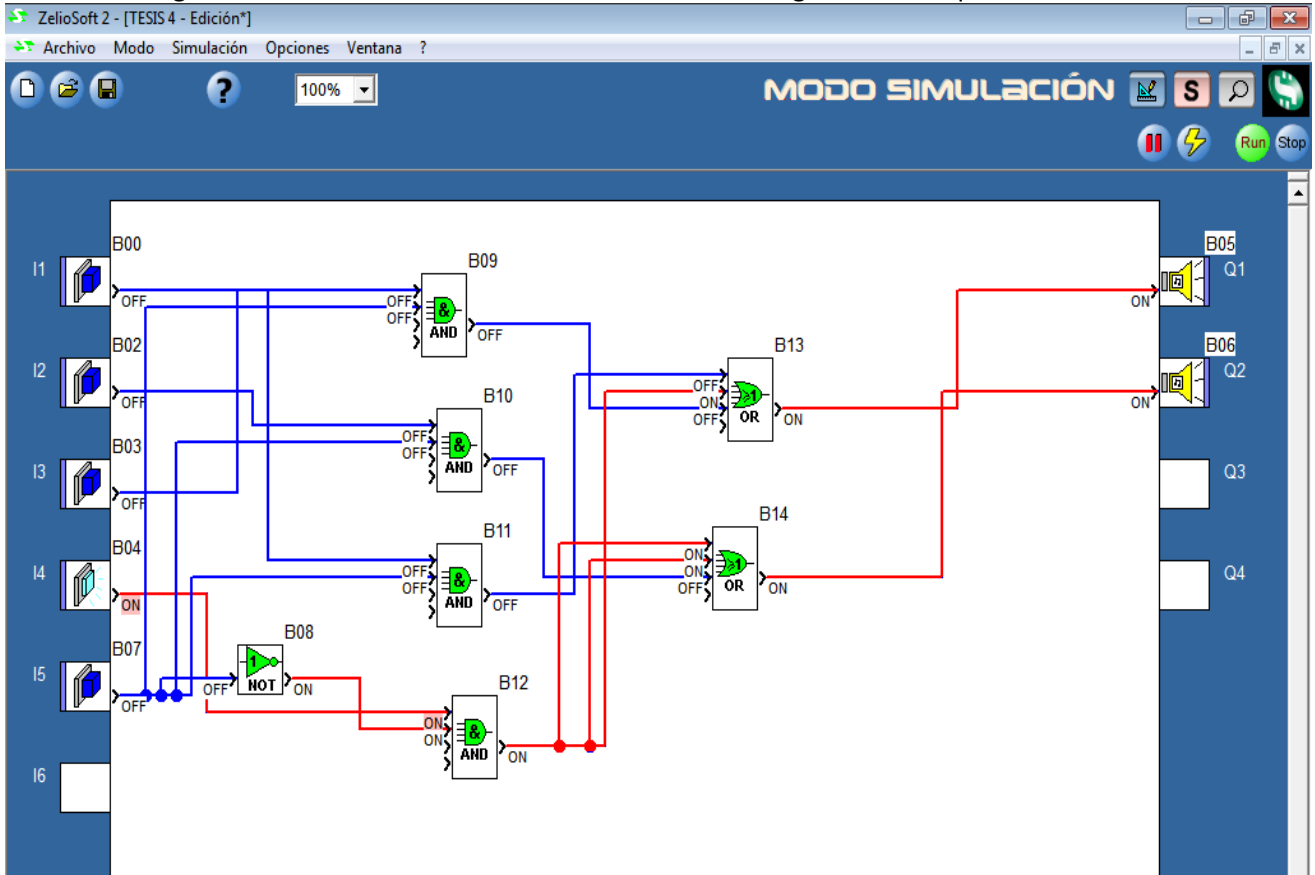


Fig 1.18

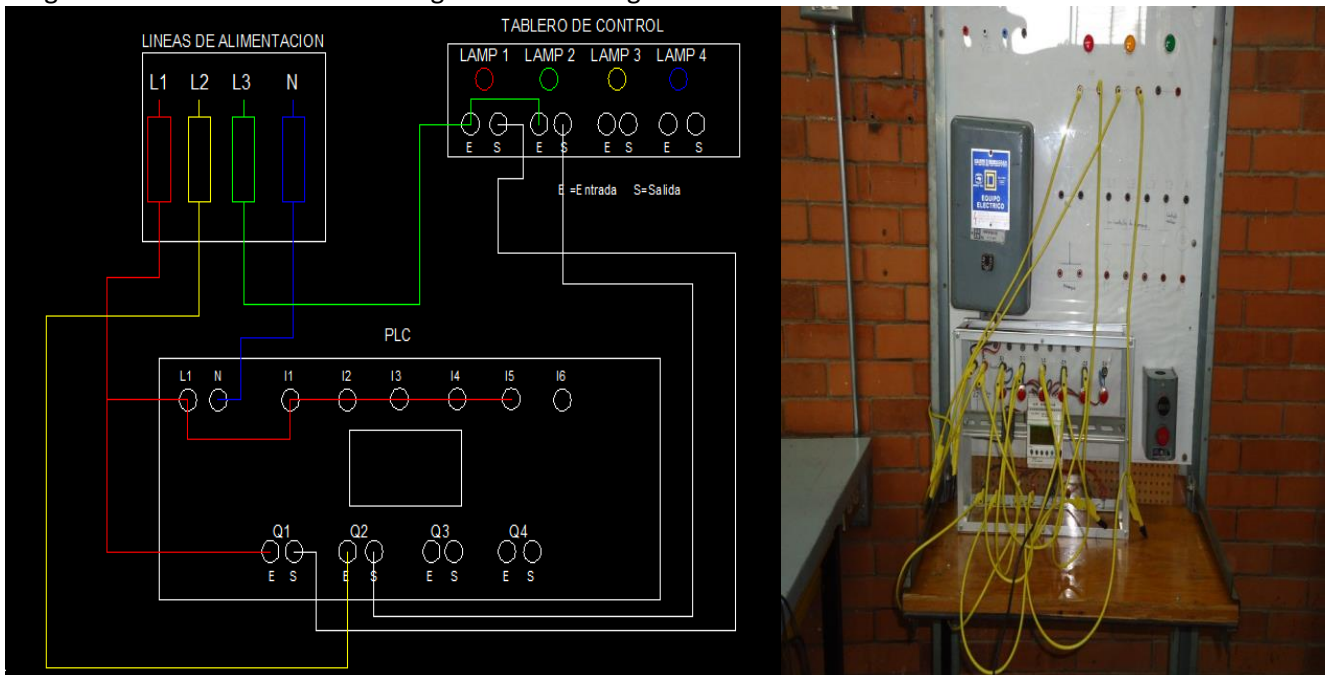
Las ecuaciones algebraicas resultantes de la simplificación de la tabla 1.7 se relacionan entre si para formar un sistema en común convirtiéndolo en diagrama de bloques en el software del programa del PLC Zelio teniendo como bloques AND en común como conexiones a las salidas de las expresiones $(\sim I5 \cdot I4)$ y $(I5 \cdot I1)$ para Q1 y Q2 las expresiones $(I5 \cdot I3)$ $(I5 \cdot I2)$ son independientes en la conexión para cada salida Q1 y Q2 .

La ecuación algebraica resultante de la tabla 1.7 se convierte a diagrama de bloques .



Programa 1.15

Diagrama de conexión de la tabla lógica 1.7 Diagrama 1.3

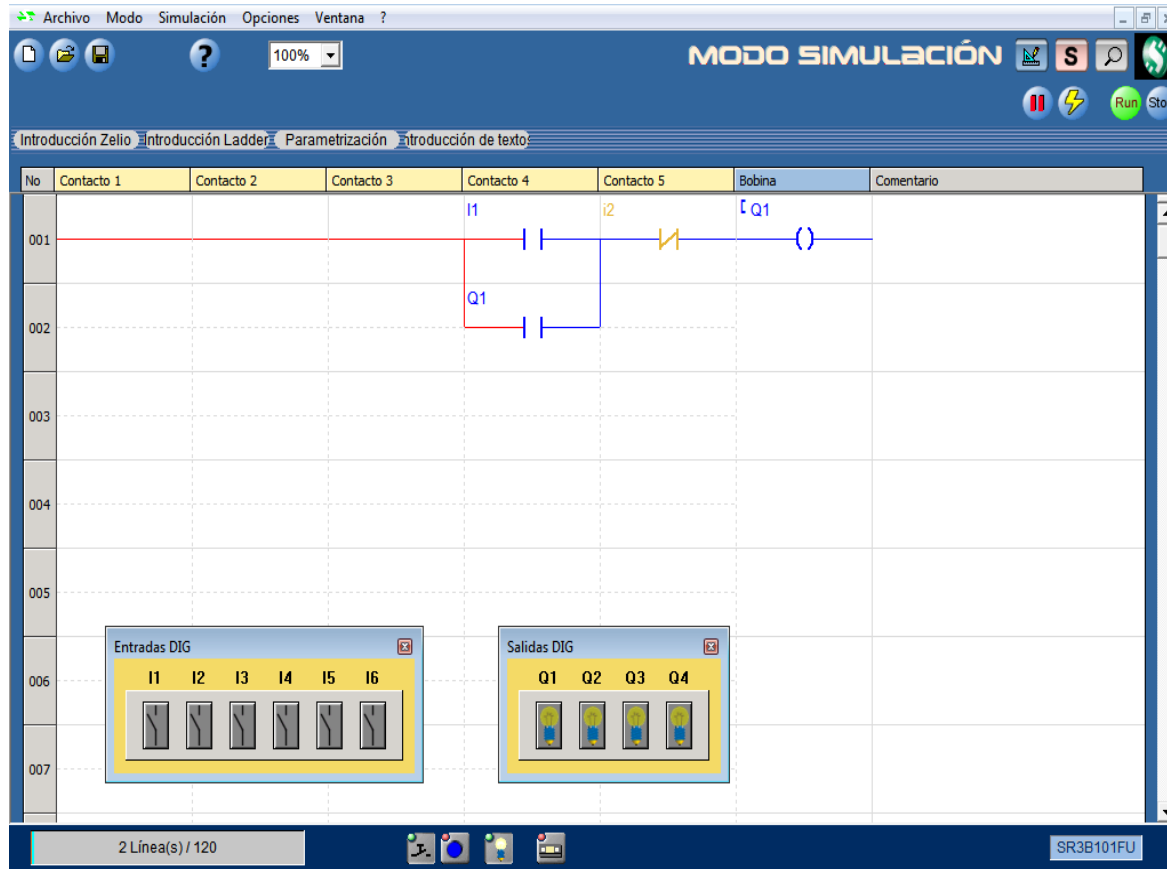


CAPÍTULO 2

2 Función Relevador con PLC ZELIO

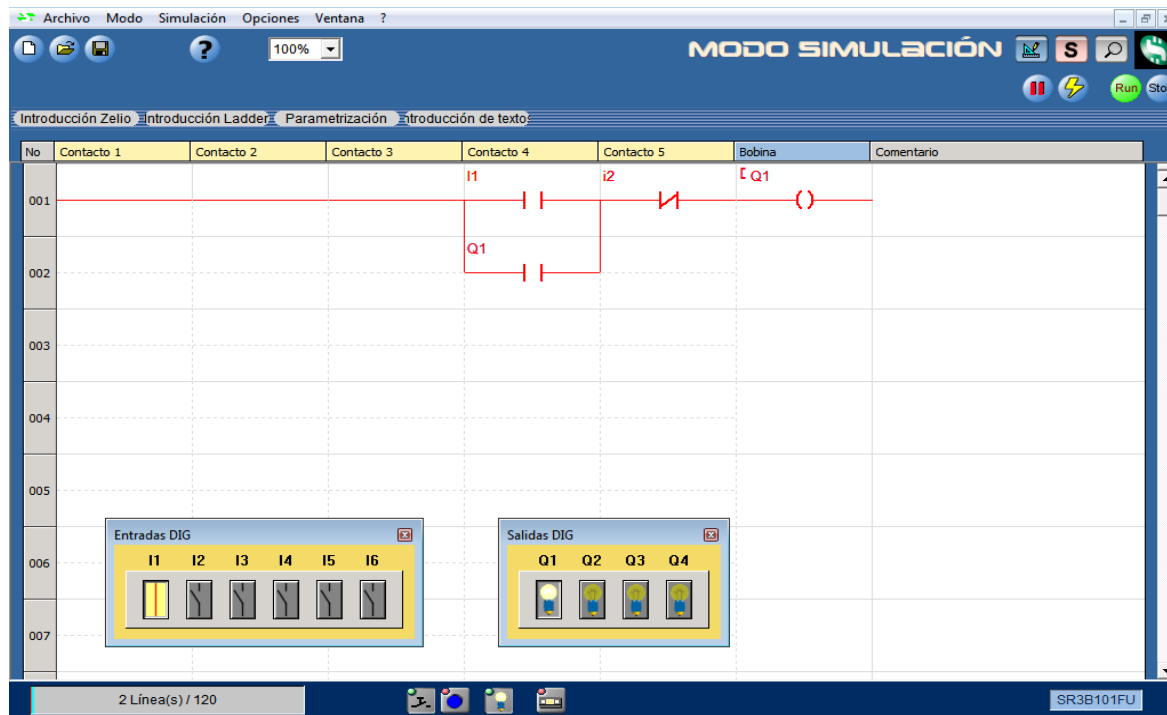
Función relevador diagrama de escalera .

En el diagrama mostrado se representa el funcionamiento del relevador encendido y apagado que tiene muchas aplicaciones básicas en un diseño de control automatizado en este diagrama eléctrico las entradas de control (I1xI2) están en serie funcionan como una compuerta AND I1 normalmente abierto I2 normalmente cerrado como contactores,(I1+Q1) operan paralelo como una compuerta OR,Q1 debajo de I1 funciona como contactor de la salida de la bobina Q1: ().



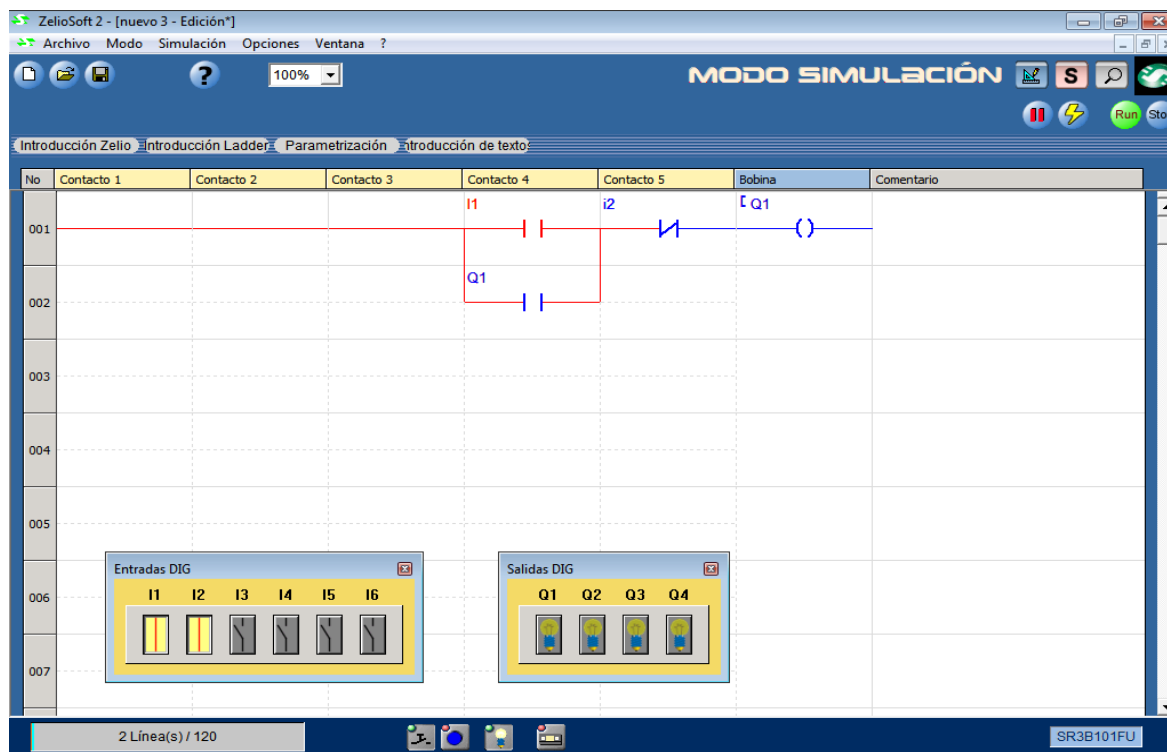
Programa 2.1

Al pulsar L1 en el recuadro de abajo en amarillo entradas DIG pasa la corriente por de bajo de I1 a Q1 pasando por I2 contacto normalmente cerrado prendiendo Q1 (⊕)



Programa 2.2

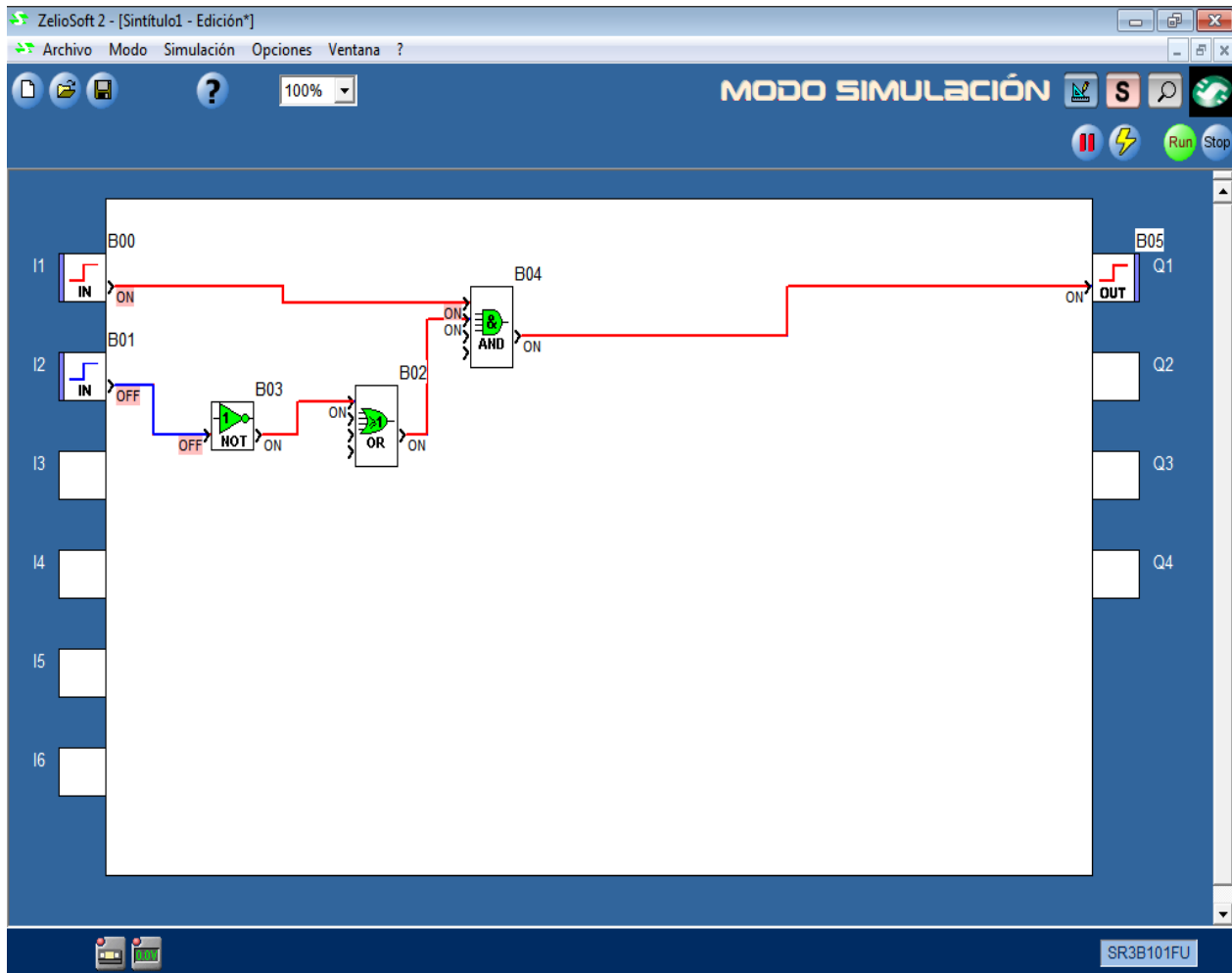
Al pulsar I2 en las entradas DIG recuadro amarillo se desactiva Q1 contactor de la bobina Q1() interrumpiendo la señal de I1 en este diagrama se representa las compuertas AND y OR simultáneamente operando en conjunción.



Programa 2.3

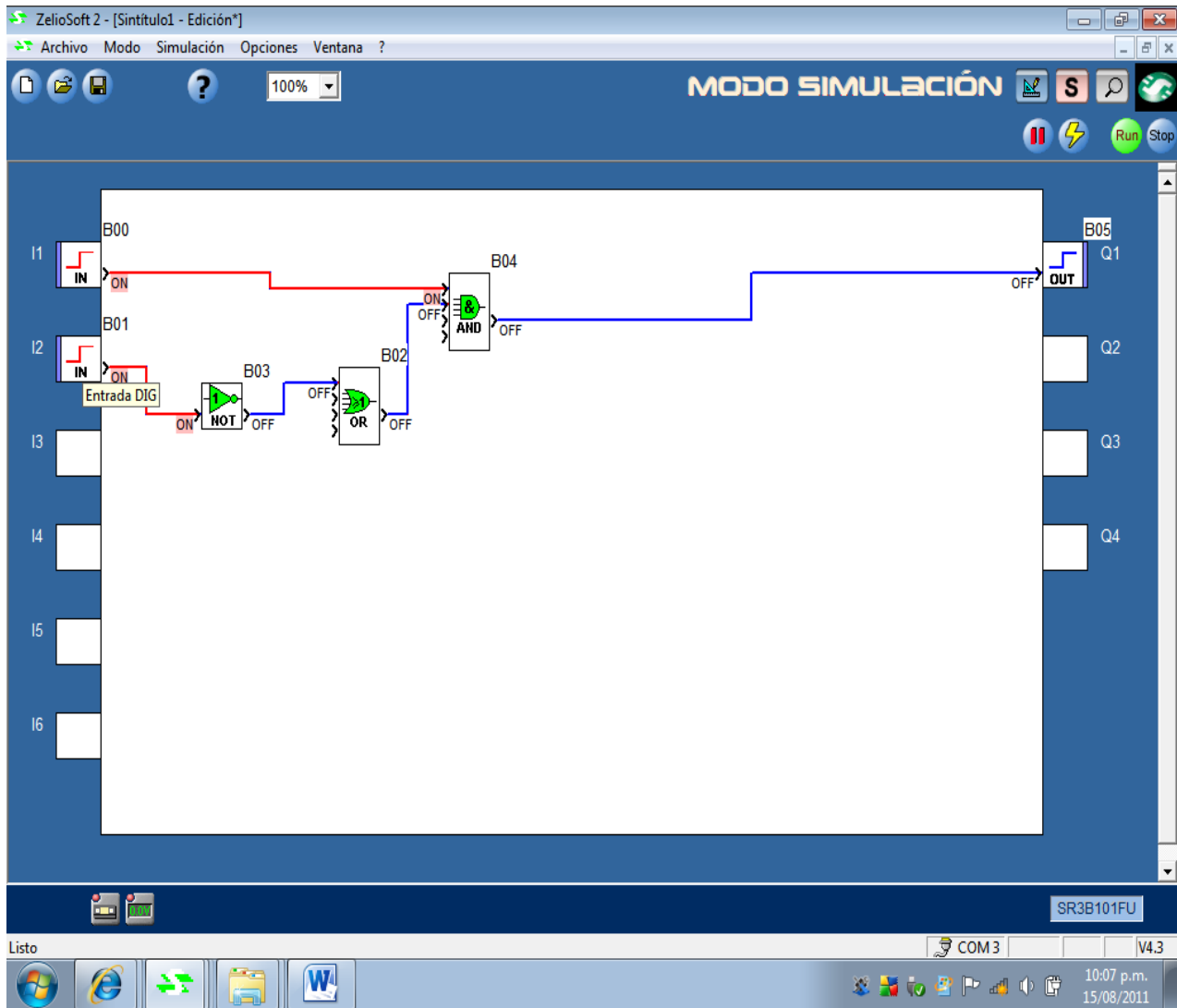
Funcion Relevador diagrama de bloques diagrama lógico.

Al pulsar I1 la compuerta OR activa el pulso en paralelo con I1 pasando la corriente a I2 normalmente cerrado mediante la compuerta NOT activando Q1 =OUT



Programa 2.4

Al pulsar I2 se corta el pulso en la compuerta OR que hace de contacto en paralelo con I1 interrumpiendo el paso de corriente a I2 normalmente cerrado desactivando Q1.



Programa 2.5

2.1 Función Temporizador con PLC ZELIO.

Temporizador de diagrama de escalera.

El temporizador se utiliza para activar y desactivar una condición lógica mediante un relevador de contactos al activar una línea de entrada L será activada y desactivada mediante un arreglo con el contacto de la bobina Q() controlada con dos contactos de tiempo (t) con su salida de respuesta(T) a los extremos de entrada y salida del relevador encendido, apagado. Los contactos (t) y la bobina (T) de respuesta se seleccionan de la casilla temporizadores con un reloj marcado, las T seguida de un número y una letra son contactos, la T sola en la segunda casilla es la bobina de los contactos (t).

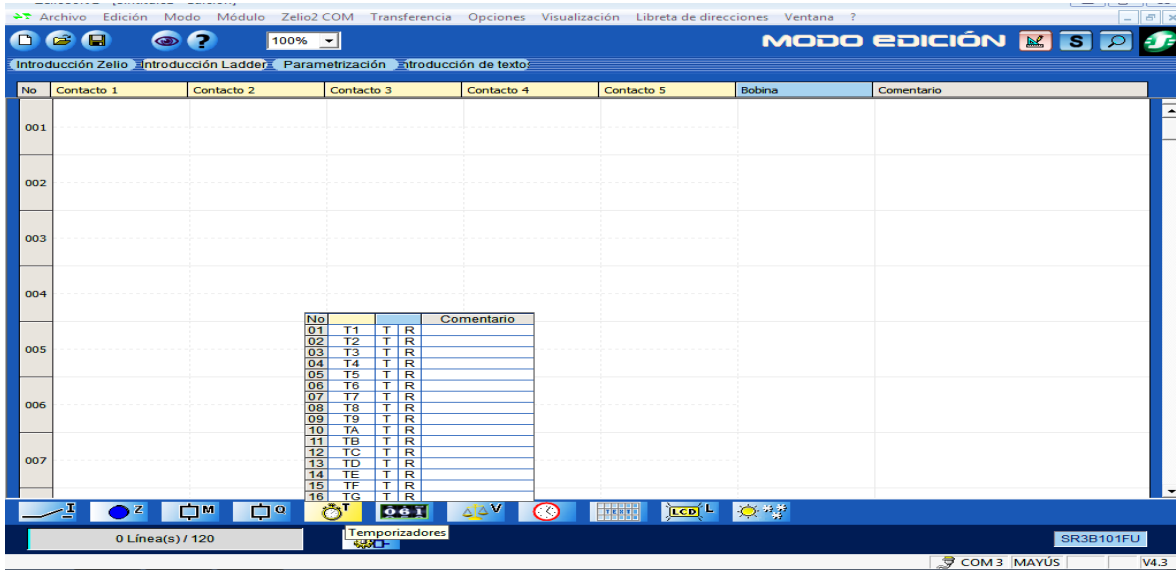
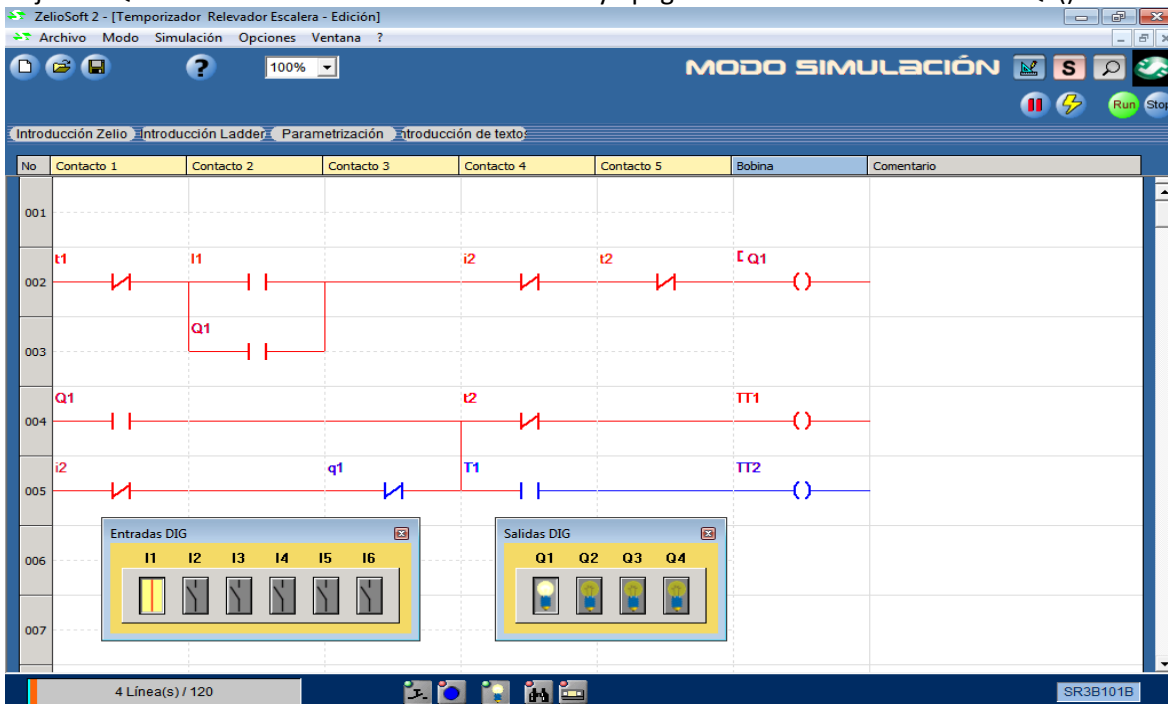


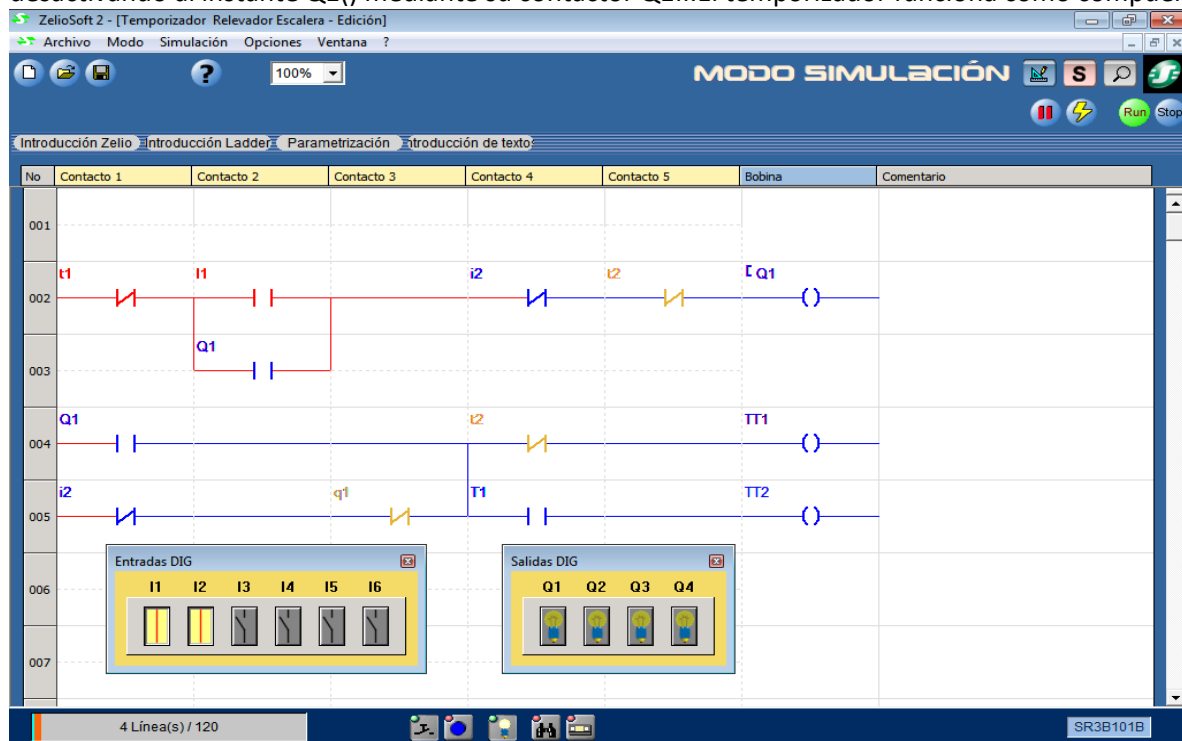
Fig 2.1

Funcionamiento del temporizador ;al pulsar I1 del relevador de contactos se activa la bobina TT1 tiempo de entrada de respuesta activada por Q1 contactor dejando pasar la corriente a i2 pasando a t2 como tiempo de salida una vez terminado el tiempo de t2 de apagado se reinicia el conteo de TT1 automático de encendido dejando Q1II enclavado siendo cíclico encendido y apagado automático de la bobina Q1().



Programa 2.6

Al pulsar I2 del relevador se interrumpe el conteo de apagado y encendido automatico desactivando TT1 y TT2 desactivando al instante Q1() mediante su contactor Q1II.El temporizador funciona como compuerta OR y XOR



Programa 2.7

Para dar tiempo se encendido y apagado se da clic izquierdo a la bobina TT2 y TT1 para seleccionar el tiempo y el tipo de respuesta .

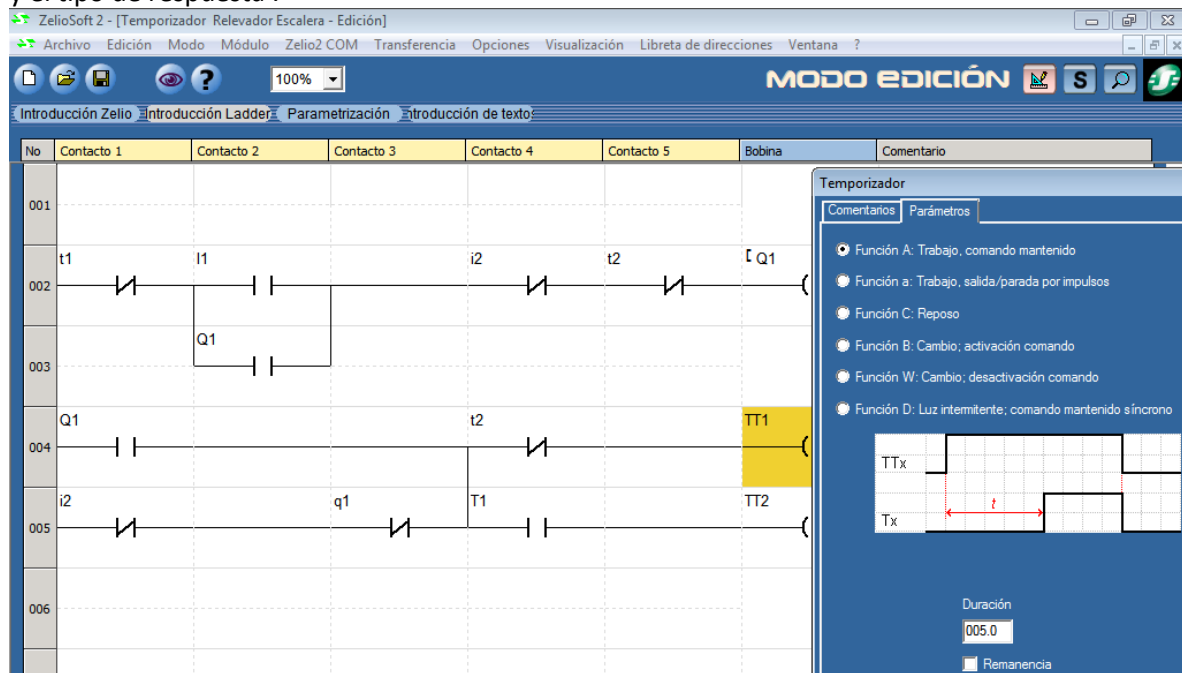


Fig 2.2

Temporizador Diagrama de bloques.

Funciona de manera semejante al temporizador de escalera, en esta forma los temporizadores ya están de manera de funciones predeterminadas solo para dar tiempo de entrada y salida de tiempo TIMER A/C este tipo de temporizadores se conectan como entrada a la salida del relevador lógico y con salida a Q1().

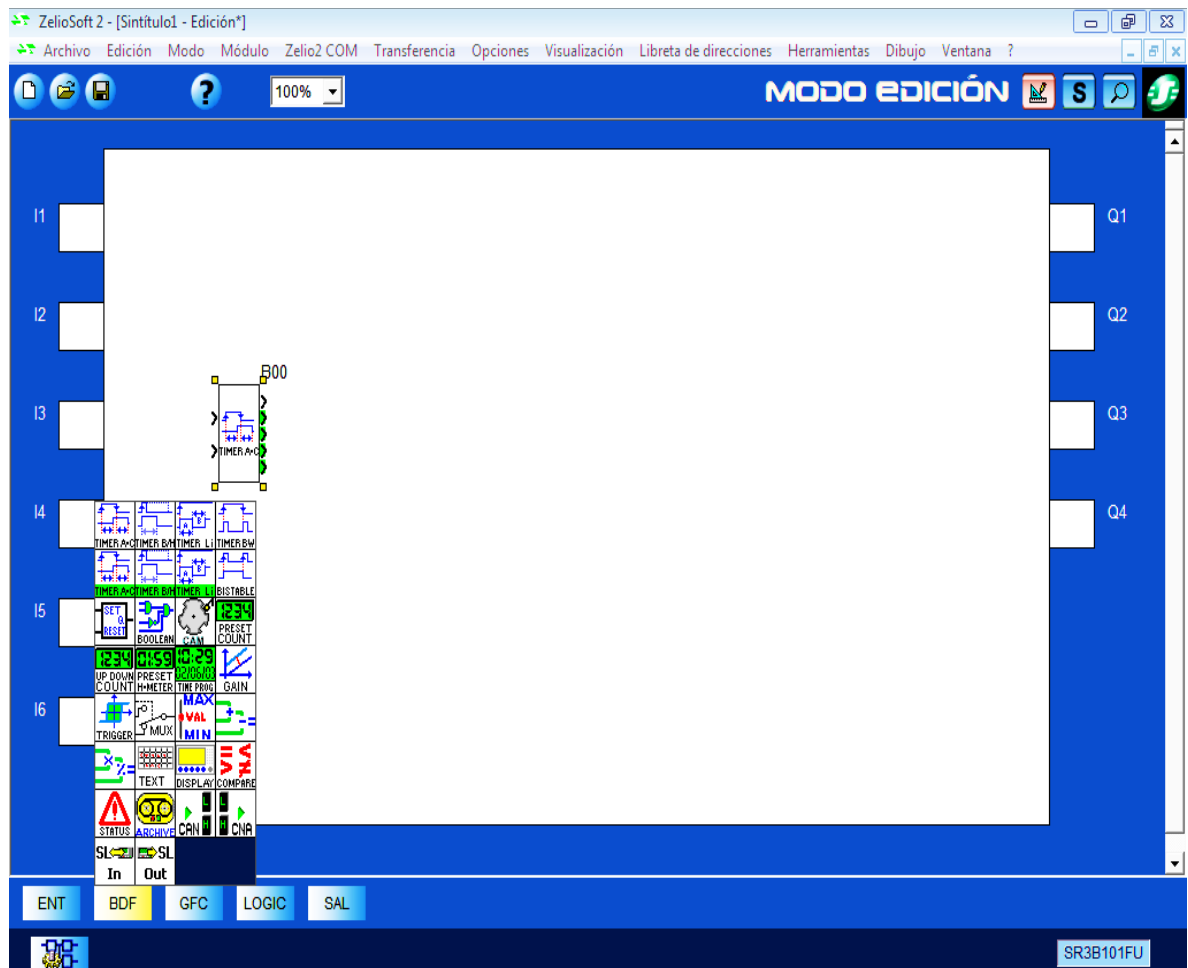
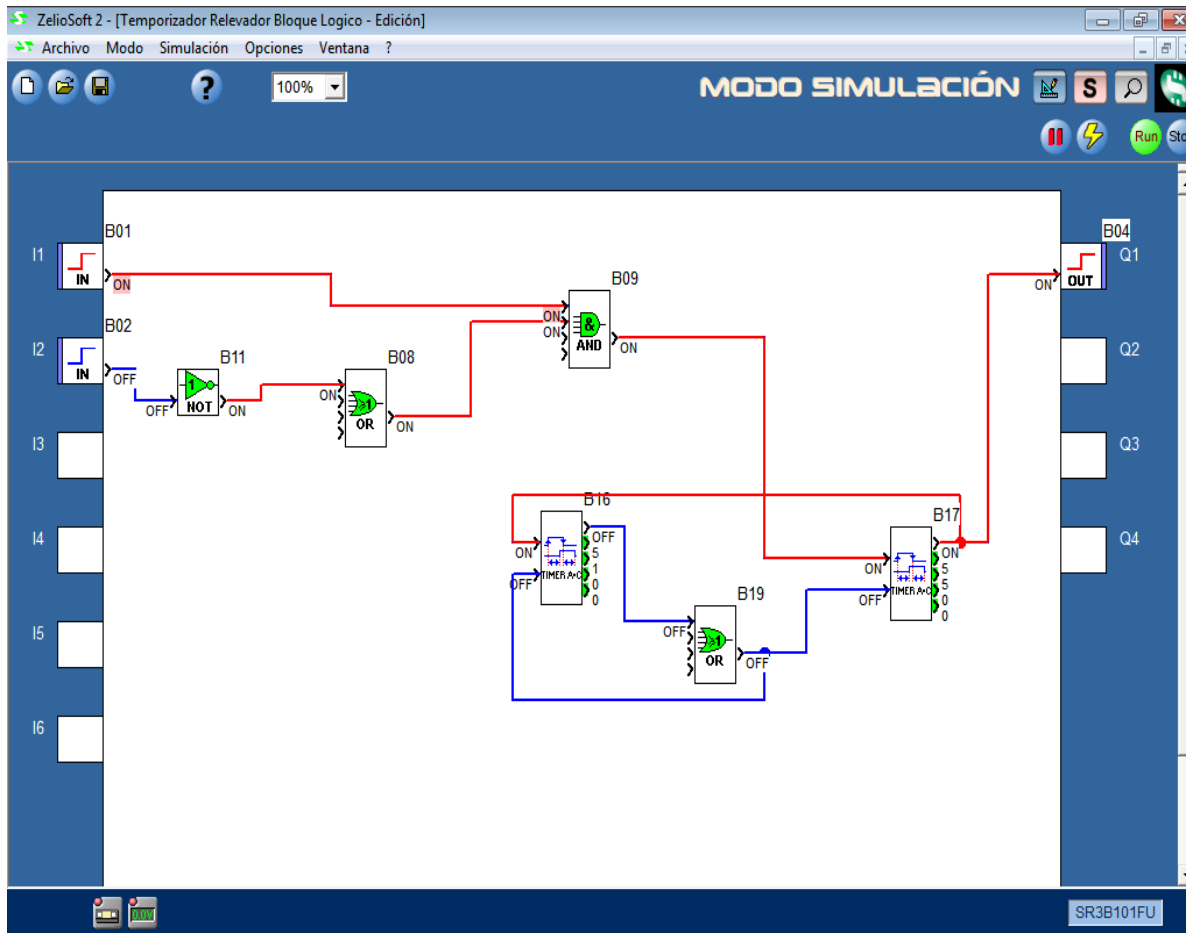


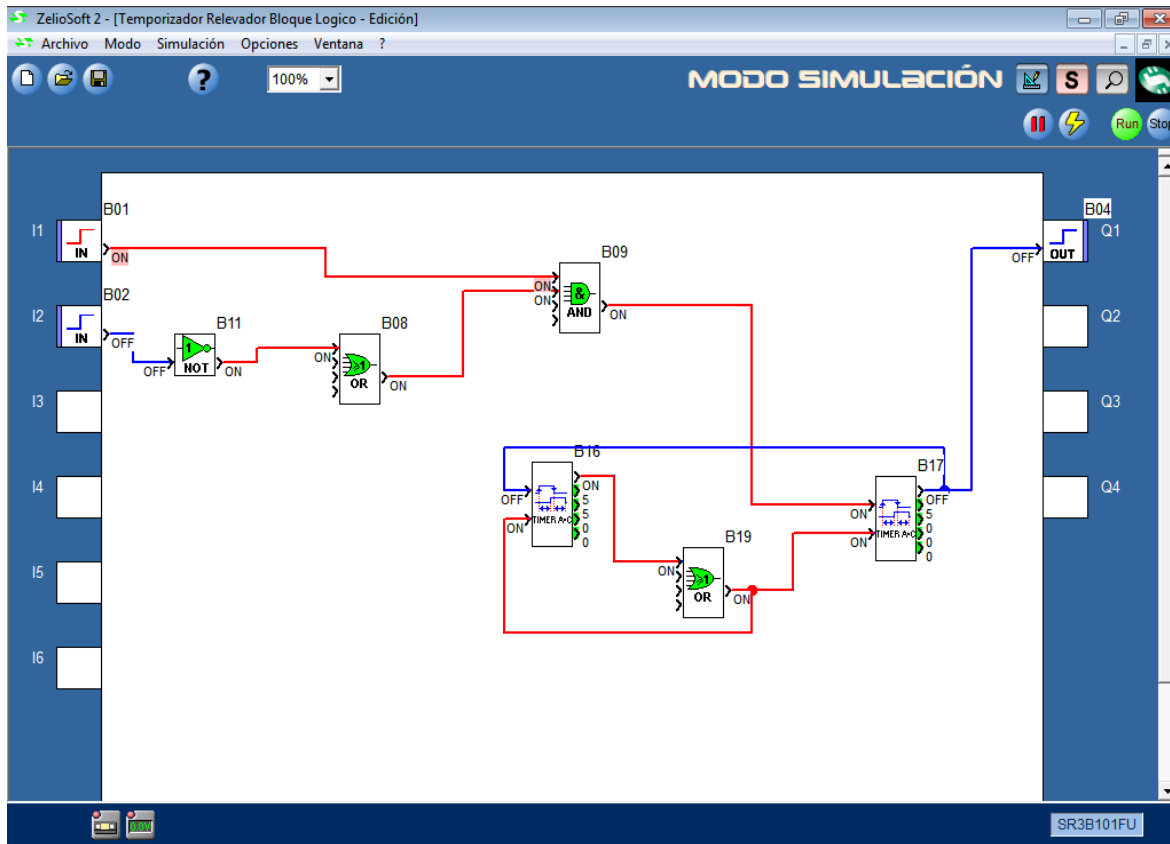
Fig 2.3

Funcionamiento temporizador diagrama de bloques;al pulsar I1 del relevador la terminal del relevador se conecta a la temporizador B17 ON contando 5 segundos de encendido y después el mismo relevador B17 al prender Q1 manda la señal a la entrada del temporizador B16 conteo 5 segundos para mandar la señal por medio de la compuerta OR la señal de apagado al temporizador B17.



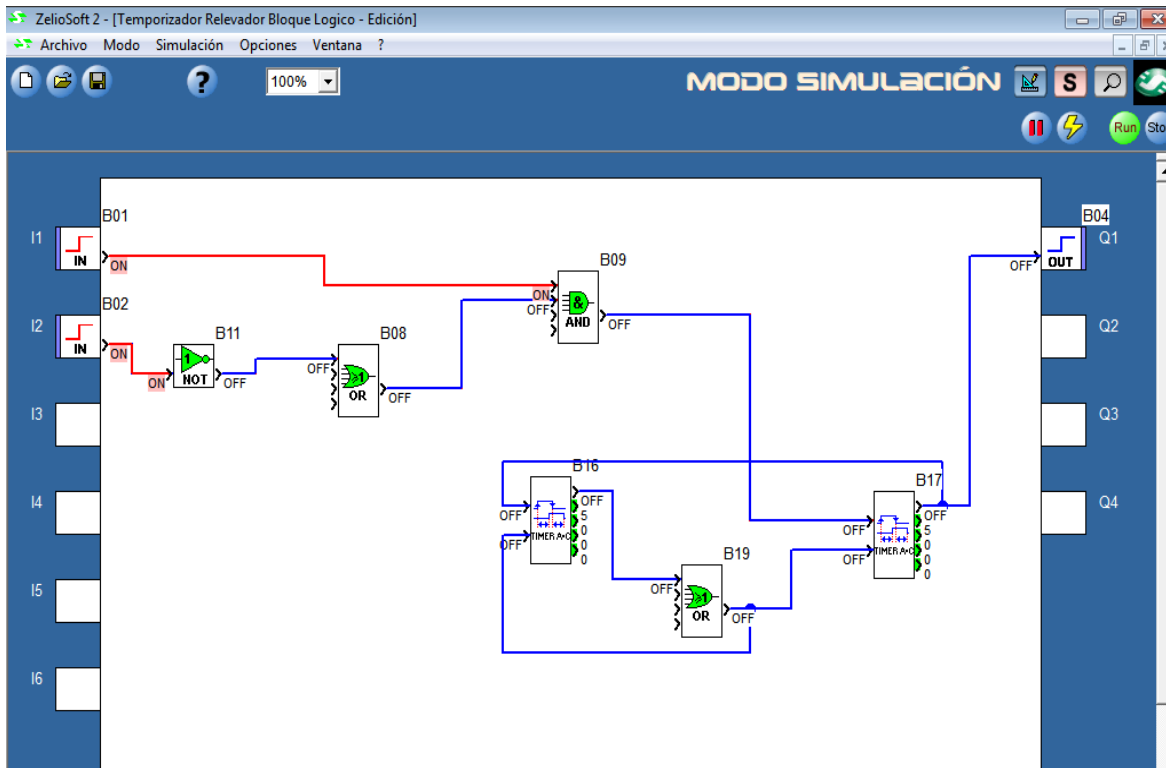
Programa 2.8

El temporizador B16 al mandar la señal por la compuerta OR se retro alimenta para su propia desactivación esperando la señal del temporizador B17. Cabe destacar que la compuerta OR es la única que interactúa con este tipo de arreglo de temporizadores ya que la compuerta XOR necesita de dos entradas activas para su funcionamiento mientras la OR en estos casos solo necesita de una entrada activa.



Programa 2.9

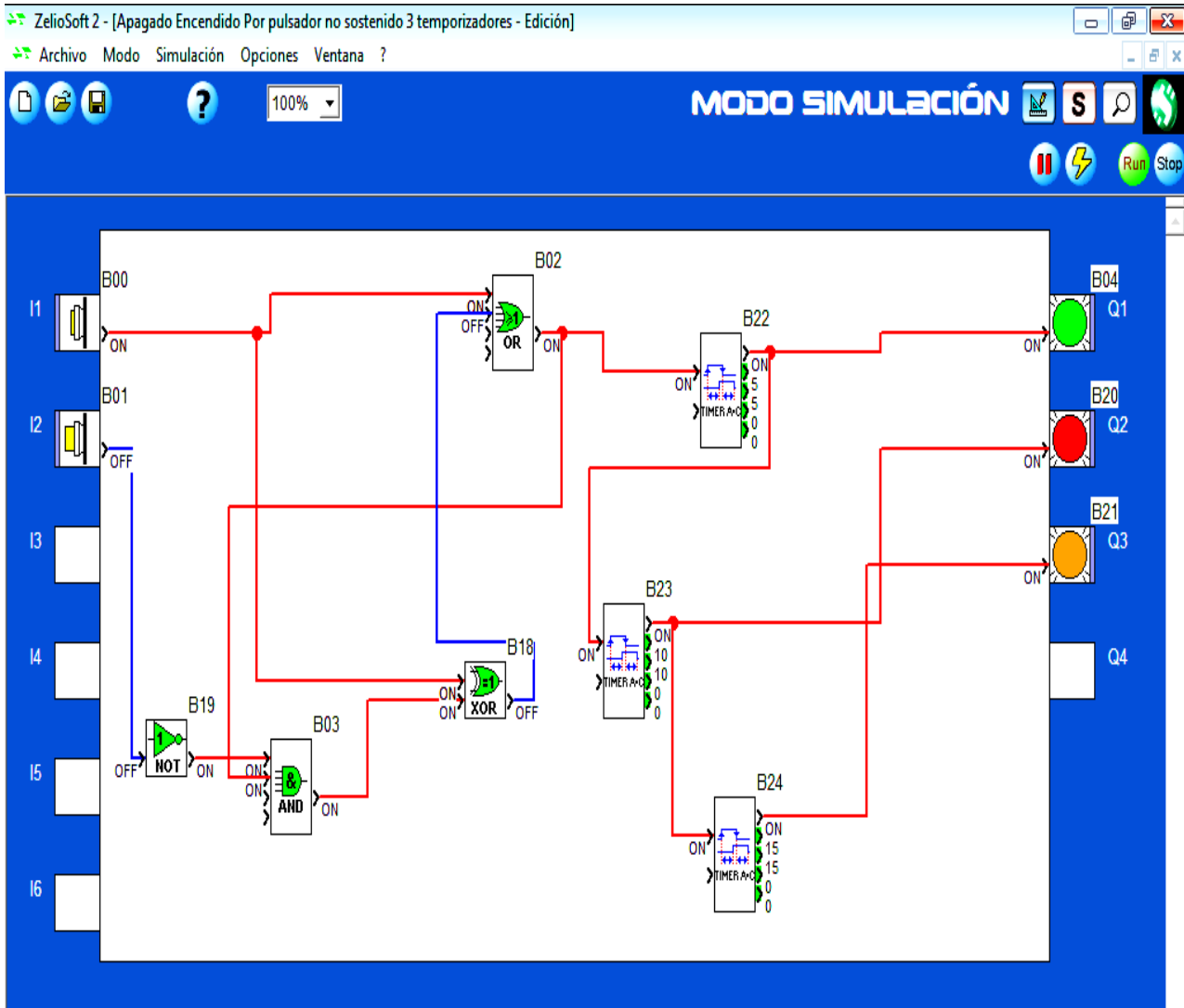
Al pulsar I2 del temporizador se desactivan simultáneamente los temporizadores deteniendo su conteo puesta a cero.



Programa 2.10

2.2 Aplicación de ejemplo función relevador ,temporizador en espera inicio .

En este ejemplo la función del relevador contactor controla el inicio de arranque del temporizador mediante la la señal de entrada I1 que activa el conteo de espera del temporizador mediante un arreglo de temporizadores que va activando en secuencia por tiempos el encendido de tres lámparas como se muestra en el programa 2.11 al pulsar I1 el contactor activa el conteo del primer temporizador que enciende la primera lámpara a los 5 segundos de espera una vez se enciende la primera lámpara inicia el conteo del segundo temporizador que enciende la segunda lámpara a los 10 segundos el tercer temporizador inicia el conteo después de encendida la segunda lámpara ,la tercera lámpara se enciende a los 15 segundos ,al presionar I2 se detiene la secuencia de los temporizadores y se desconectan las lámparas automáticamente.



Programa 2.11

Diagrama de alimentación y conexión del programa relevador temporizador espera inicio 3 lámparas.

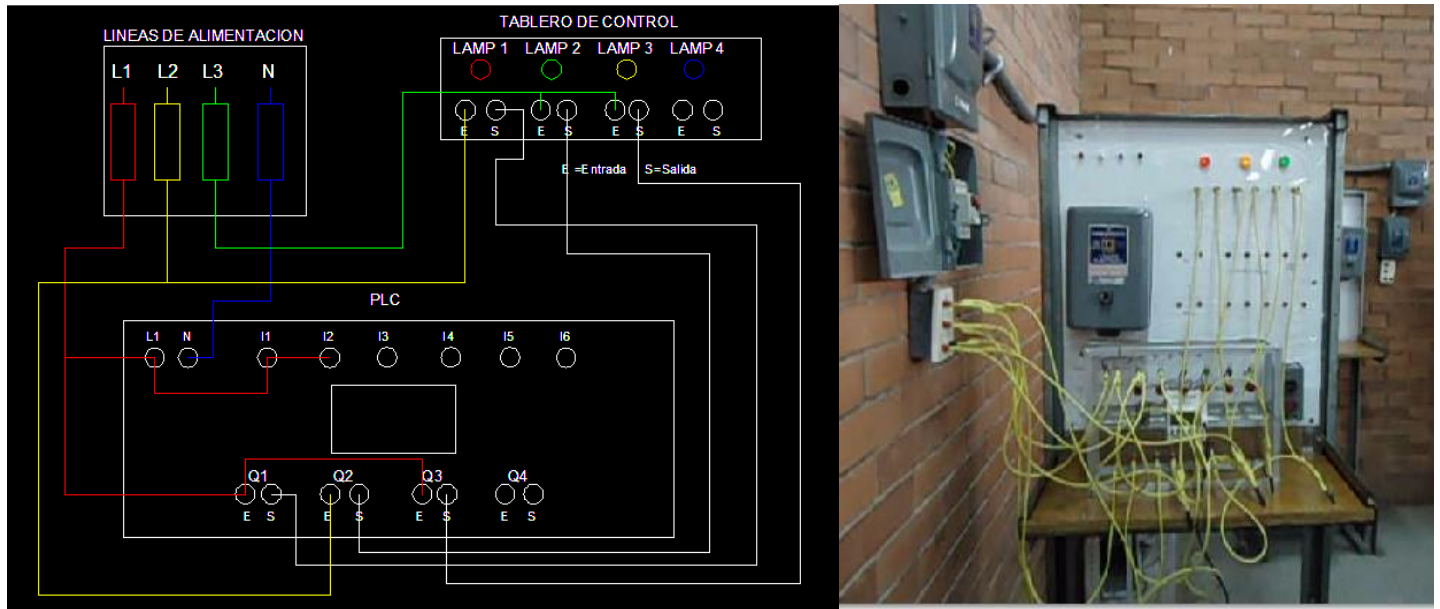
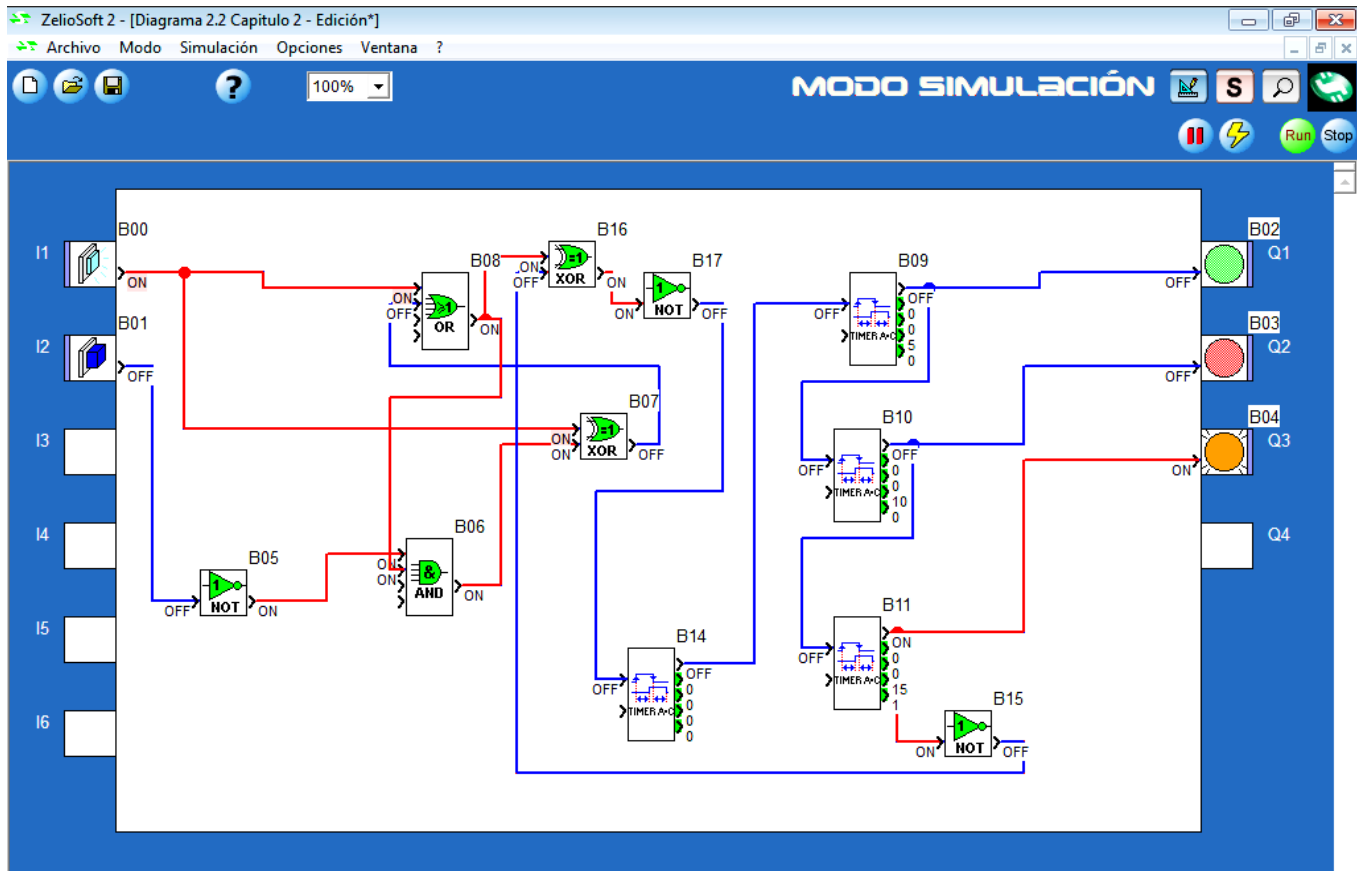


Diagrama 2.11

2.3 Aplicación de ejemplo función relevador temporizador desconexión reinicio.

En este ejemplo se tiene un arreglo de 3 lámparas al presionar I1 a la entrada el relevador contactor activa las tres lámparas al mismo tiempo junto con los tres temporizadores de cada lámpara a controlar el tiempo la primera lámpara se desactiva al conteo a los 5 segundos ,la segunda lámpara a los 10 segundos la tercera a los 15 segundos una vez terminado el ciclo de apagado de las tres lámparas se vuelven a encender simultáneamente las tres lámparas al mismo tiempo automáticamente para repetir la secuencia de conteo y apagado de cada lámpara para detener el ciclo de los temporizadores de presiona I2 parando la secuencia del programa quedando encendidas las tres lámparas sin apagar o apagadas dependiendo del estado del programa.



Programa 2.12

Diagrama de alimentación y conexión del programa relevador temporizador desconexión reinicio 3 lamparas.

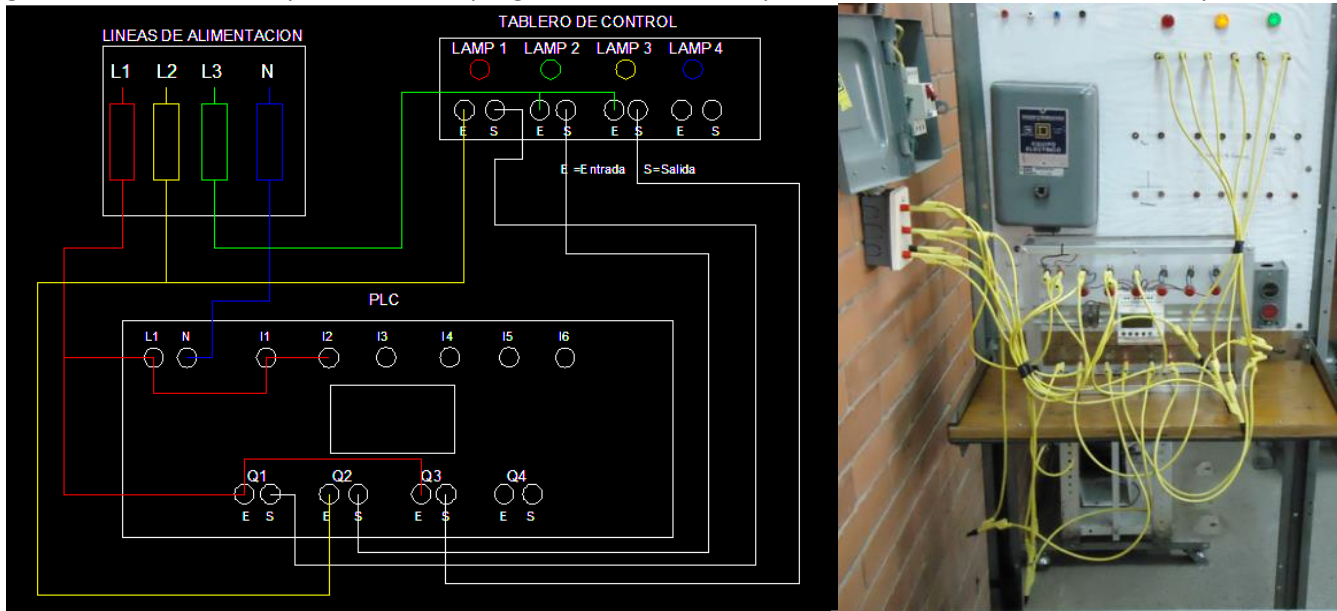
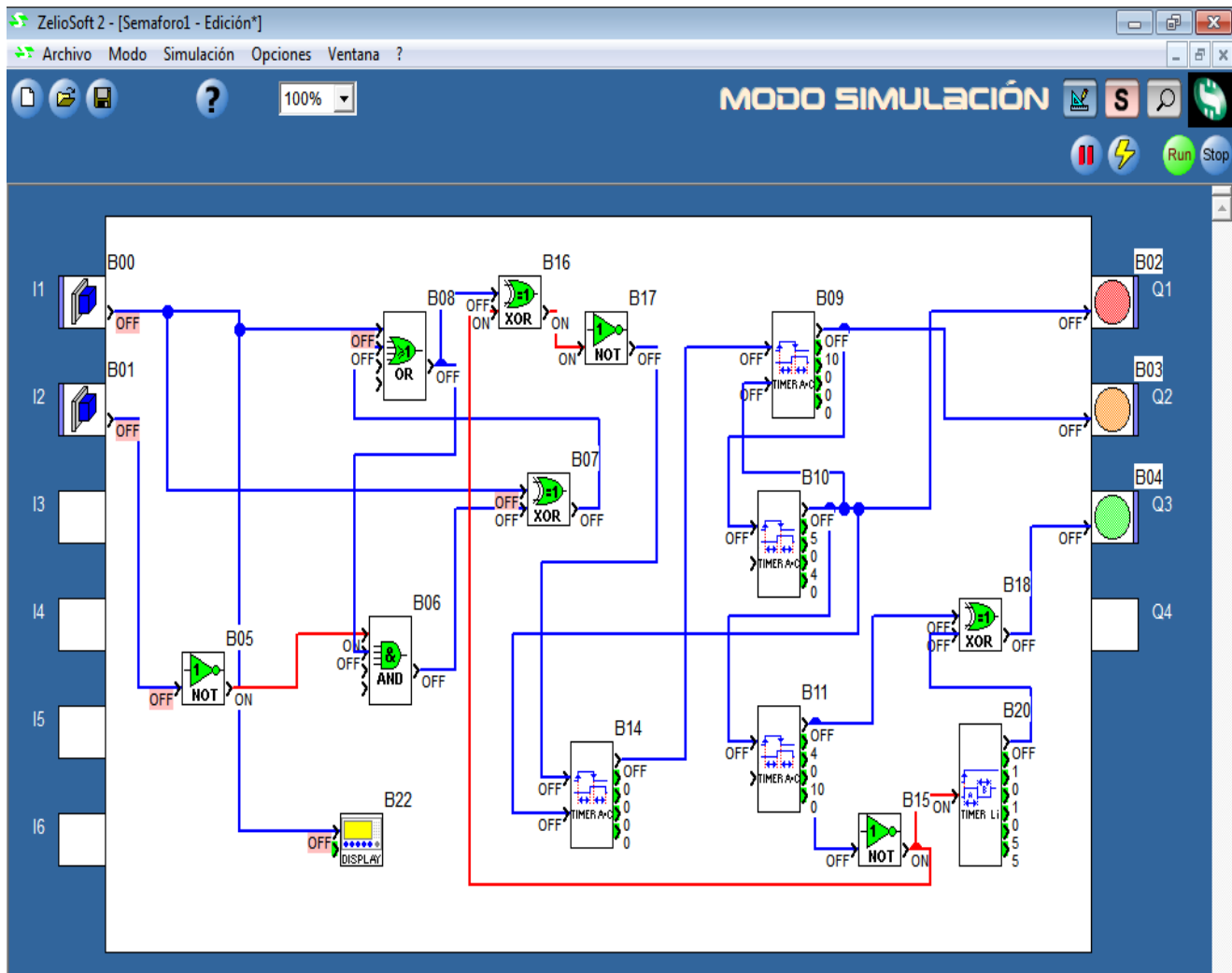


Diagrama 2.12

2.4 Aplicación de ejemplo semáforo

En este ejemplo se tiene un arreglo de tres lámparas que simulan la secuencia de tiempo de señalización de las lámparas de un semáforo. La secuencia se inicia al pulsar I1 de la entrada del PLC iniciando la señalización con 5 destellos en la lámpara verde para pasar a la lámpara naranja como preventivo al término de los destellos, la lámpara roja se enciende a los 5 segundos de conteo de apagado de la lámpara naranja, la lámpara verde se enciende después de 4 segundos de conteo de apagado de la lámpara roja, la lámpara verde se mantiene encendida 10 segundos para terminar el conteo e iniciar con los destellos y volver a repetir el ciclo de la secuencia al pulsar I2 se interrumpe el ciclo y se apagan las lámparas.



Programa 2.13

Diagrama de alimentación y conexión del programa semáforo.

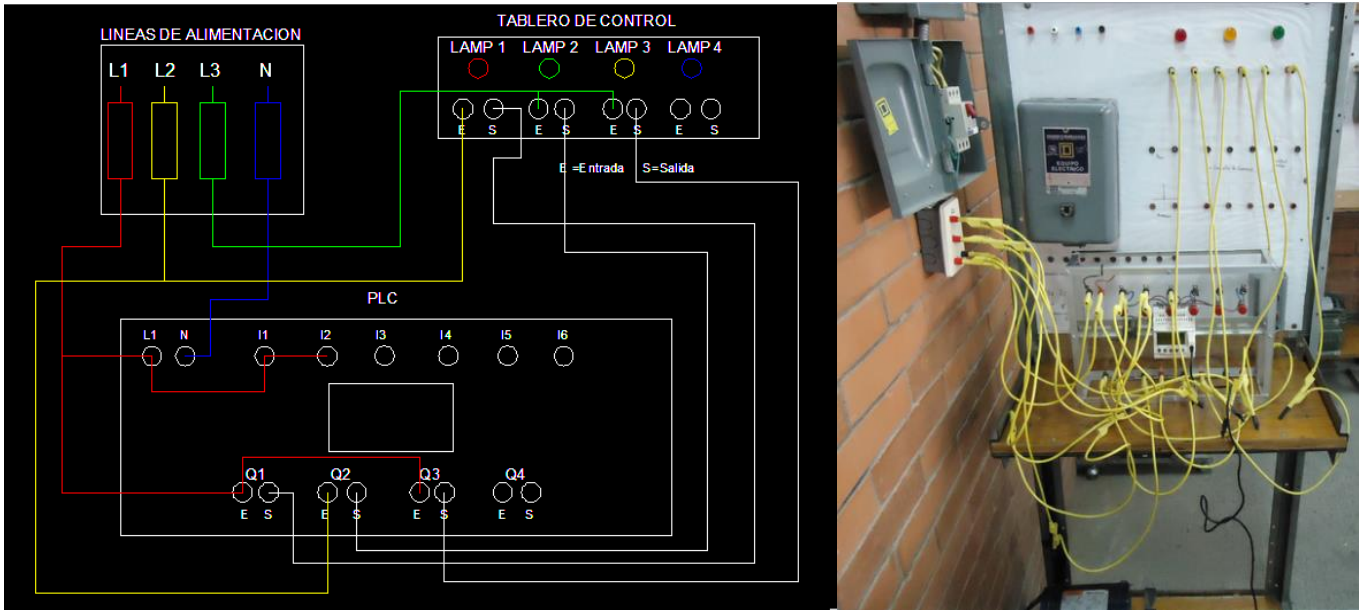


Diagrama 2.13

CAPÍTULO 3 Aplicación de operación a maquinas eléctricas con compuertas lógicas.

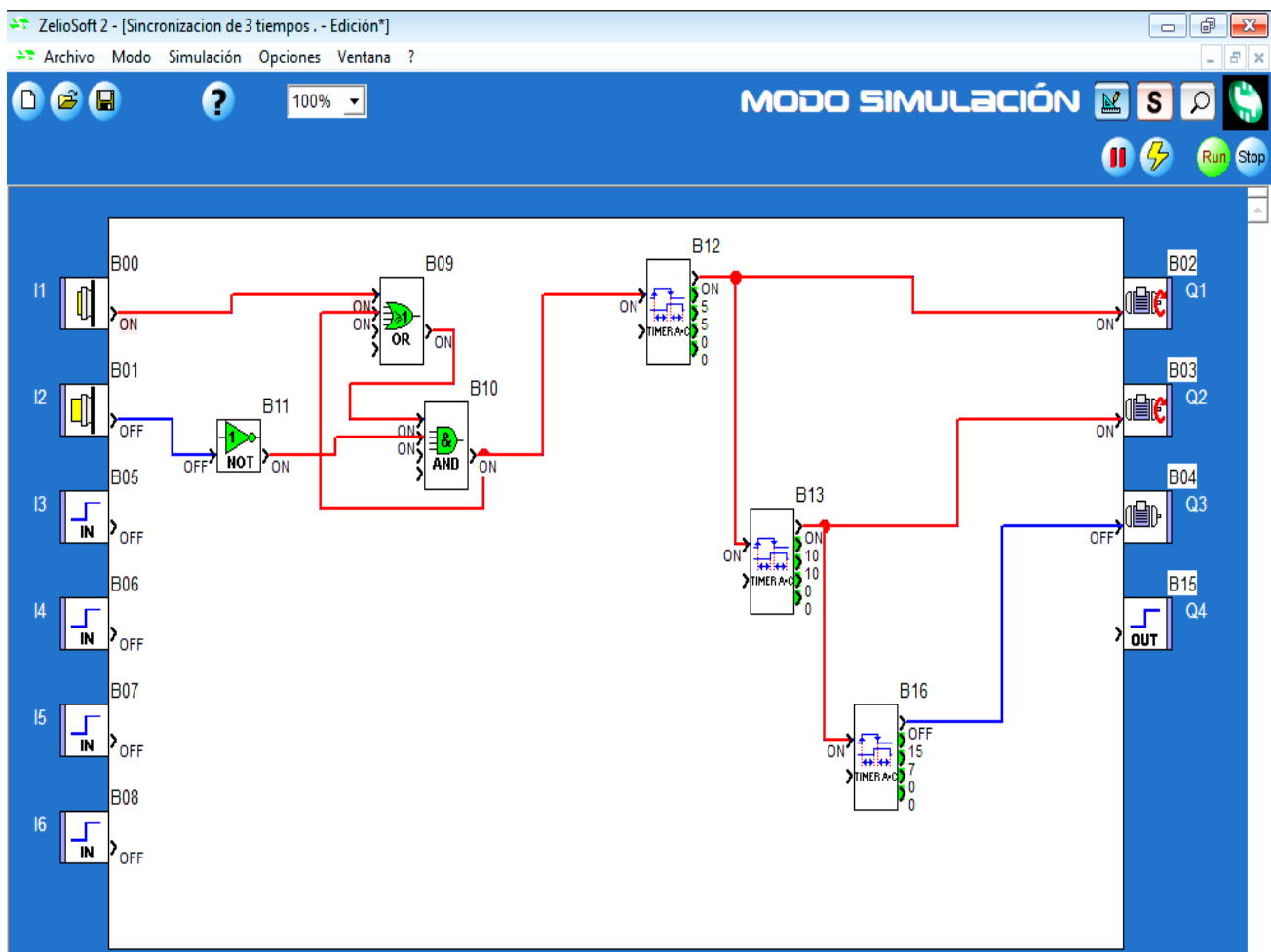
3 Aplicación sincronización de 3 motores trifásicos con diferentes tiempos.

En este ejemplo de aplicación se ponen en operación el arranque de tres motores trifásicos sincronizados con diferentes tiempos en el arranque controlados por tres temporizadores como función de aplicación que vienen en el programa del PLC Zelio en el arranque.

En el programa de control para esta aplicación se tienen las entradas I1 que activa los tres temporizadores que ponen en marcha los tres motores en diferentes tiempos de arranque por medio del enclavamiento que hacen las compuertas AND ,OR,NOT al recibir el pulso de I1 transfiriendo el pulso al arreglo de los tres temporizadores que activan las salidas Q1,Q2,Q3 poniendo en marcha a los motores en diferentes tiempos de arranque.

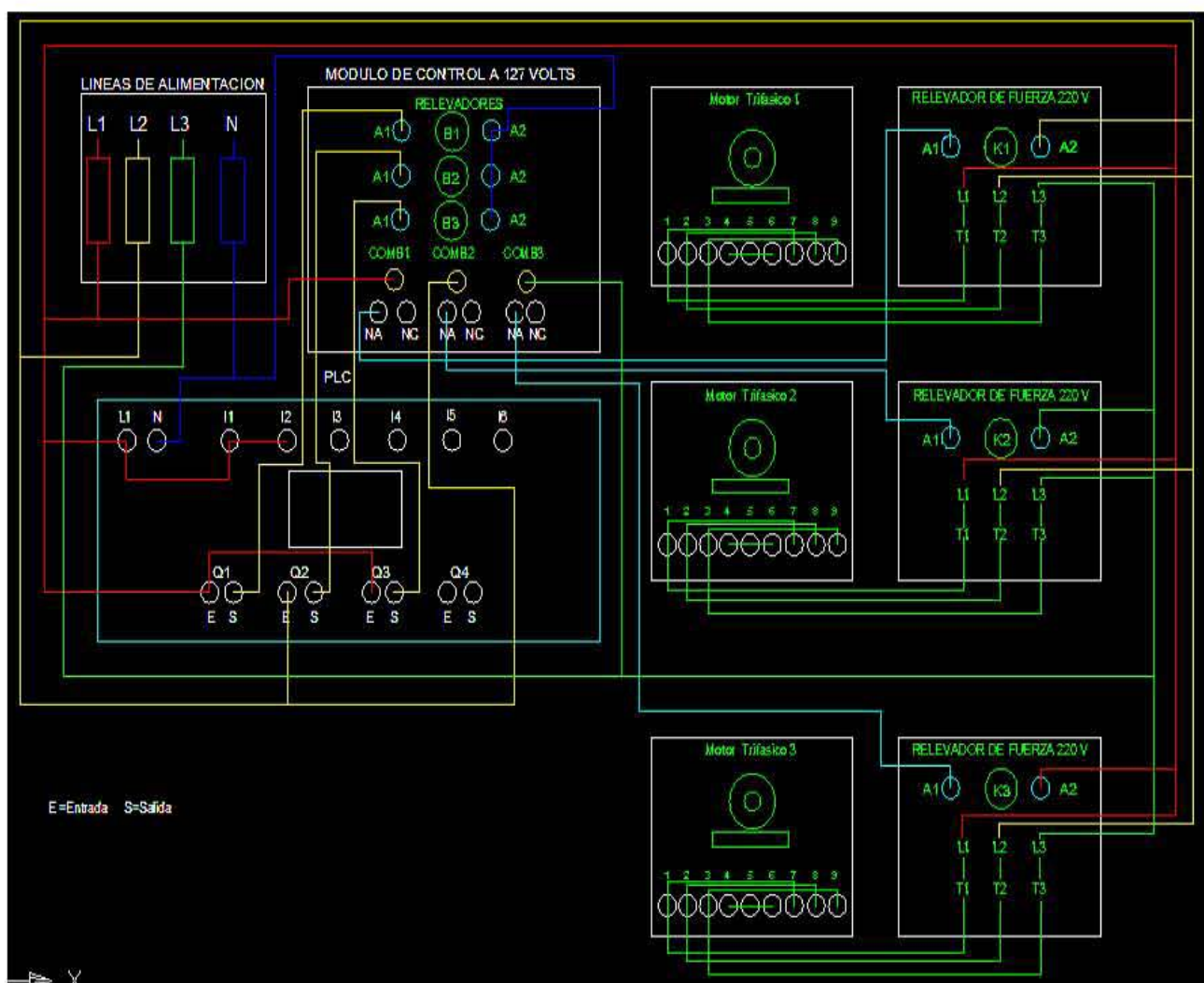
Al pulsar I1 se activan los tres temporizadores por tiempos el primer temporizador activa a los 5 segundos el primer motor recibido el pulso de I1 ,el segundo temporizador se activa a los 10 segundos acabando el conteo del primer temporizador poniendo en marcha el segundo motor, el tercer temporizador se activa a los 15 segundos arrancando el tercer motor.

Al pulsar I2 se desactivan los temporizadores poniendo fin a la marcha de los tres motores simultáneamente.



Programa 3.0

Diagrama de alimentación y conexión del programa 3.0



3.1 Aplicación sincronización de 3 motores trifásicos con diferentes tiempos con sentido inverso de giro

En esta aplicación se pone en operación el arranque de tres motores sincronizados activados por tres temporizadores en diferente tiempo de arranque .

En el programa de aplicación se tiene la entrada I1 que es el paro del sistema del programa desactivando la compuerta AND B07 desactivando el pulso a la compuerta OR B13 y a los temporizadores mediante la compuerta NOT B09 desactivando el enclavamiento en el arreglo de las compuertas OR B08,NOT B09 y AND B07 .

La entrada I2 activa el enclavamiento transfiriendo el pulso a la compuerta OR B13 a los temporizadores activando las salidas Q1,Q2,Q3 en diferentes tiempos poniendo en marcha los tres motores en sentido horario ,resaltando que la entrada I2 no es de pulso

sostenido en el enclavamiento en la botonera del PLC para ello se tiene la retroalimentación de la salida de la compuerta AND B07 a la entrada de la compuerta OR B08 a manera de memoria de mantener el pulso de activación en el enclavamiento se para la marcha de los motores al pulsar I1.

La entrada I3 activa el enclavamiento de las compuertas B07, B08, B09 a la compuerta B13 para activar por otra rama del sistema a la salida Q4 que activa el sentido inverso de los tres motores al activarse las salidas Q1,Q2,Q3 para activar Q4, se activa el pulso de I3 a la compuerta OR B14 de la salida de esta a la entrada de la compuerta AND B15 de la salida a la entrada de la compuerta AND B16 para cerrar el circuito de activación de la salida de la compuerta OR B13 se conecta a la entrada de la compuerta ANDB15 y para guardar el pulso de activación no sostenido se transfiere de la salida de la compuerta AND B15 a la entrada de la compuerta OR B14, se pulsa I1 parando el programa.

En esta aplicación se pone en operación el arranque de tres motores sincronizados activados por tres temporizadores en diferentes tiempos de arranque.

En el programa de aplicación se tiene la entrada I1 que es el paro del sistema desactivando la compuerta ANDB07, ORB08, NOTB09 que hacen el enclavamiento de las entradas I2,I3 a los temporizadores B10, B11,B12.

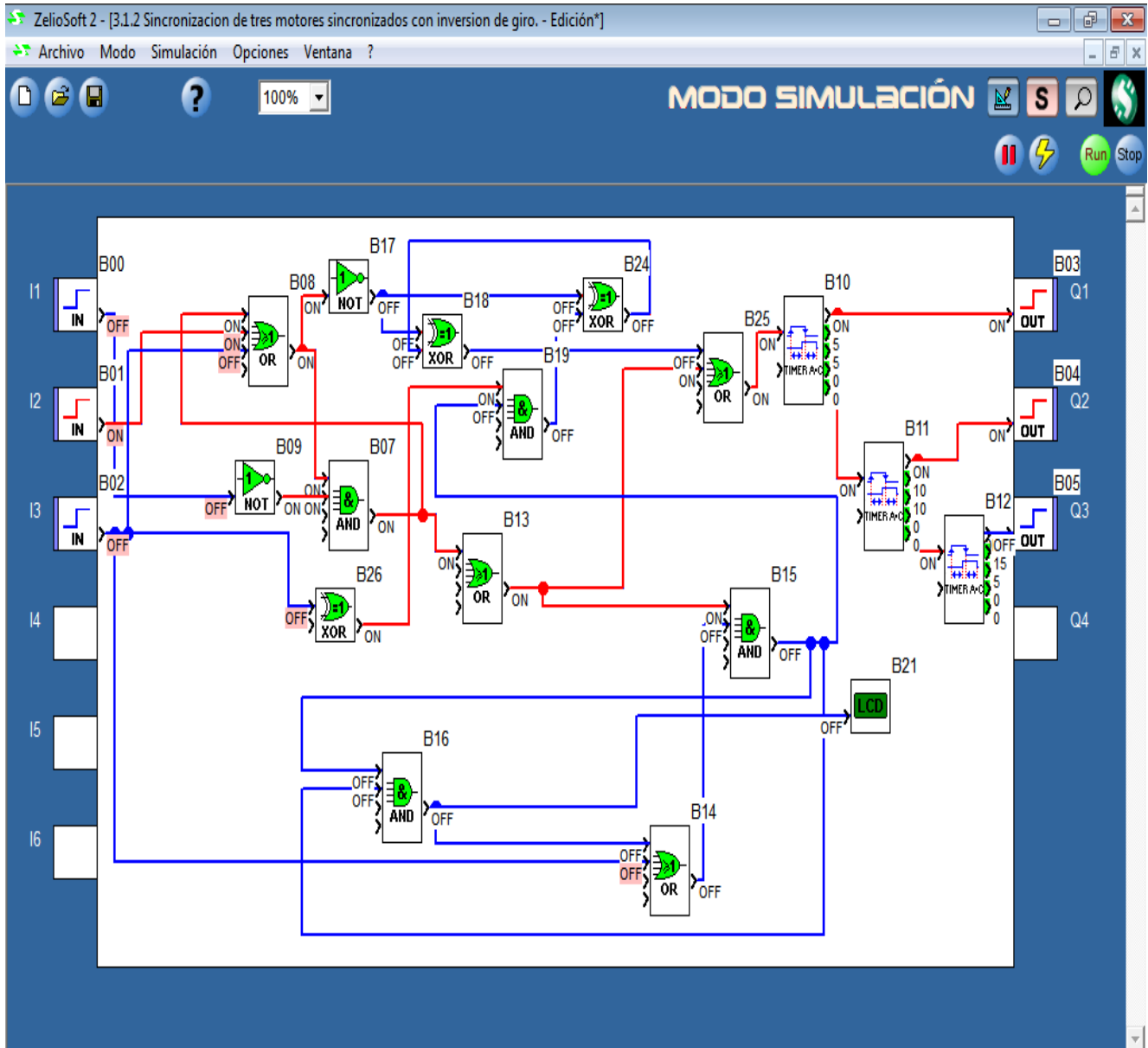
La entrada I2 activa el enclavamiento transfiriendo el pulso de la salida del enclavamiento de la compuerta AND B07 a la compuerta ORB13 ,a la compuerta ORB5 activando el arreglo de los temporizadores B10,B11 B12, activando la marcha de los motores en tiempos diferentes de arranque con las salidas Q1,Q2,Q3 después de un tiempo los tres motores en marcha se pulsa I1 para poner fin l la marcha de los motores.

La entrada I3 activa el enclavamiento transfiriendo el pulso de la compuerta AND B07 a la compuerta ORB13 ,a la compuerta ORB5 activando el arreglo de los temporizadores B10,B11

B12,activando la marcha de los motores en sentido inverso en tiempos diferentes de arranque con las salidas Q1,Q2,Q3 después de un tiempo los tres motores en marcha se pulsa I1 para poner fin l la marcha de los motores.

Al activar I3 se activa la pantalla LCD L21 como indicador del sentido inverso del motor por medio de las compuertas OR14 a AND15 a la compuerta AND16.

Las compuertas NOTB17, XORB18, XORB24, XORB26 ,ANDB19 es un arreglo de seguridad que interactúa con las entradas I2,I3 que impide la transferencia de pulso en el control de los temporizadores afectando a las salidas Q1,Q2,Q3.



Programa 3.1

Diagrama de alimentacion y conexión del programa 3.1

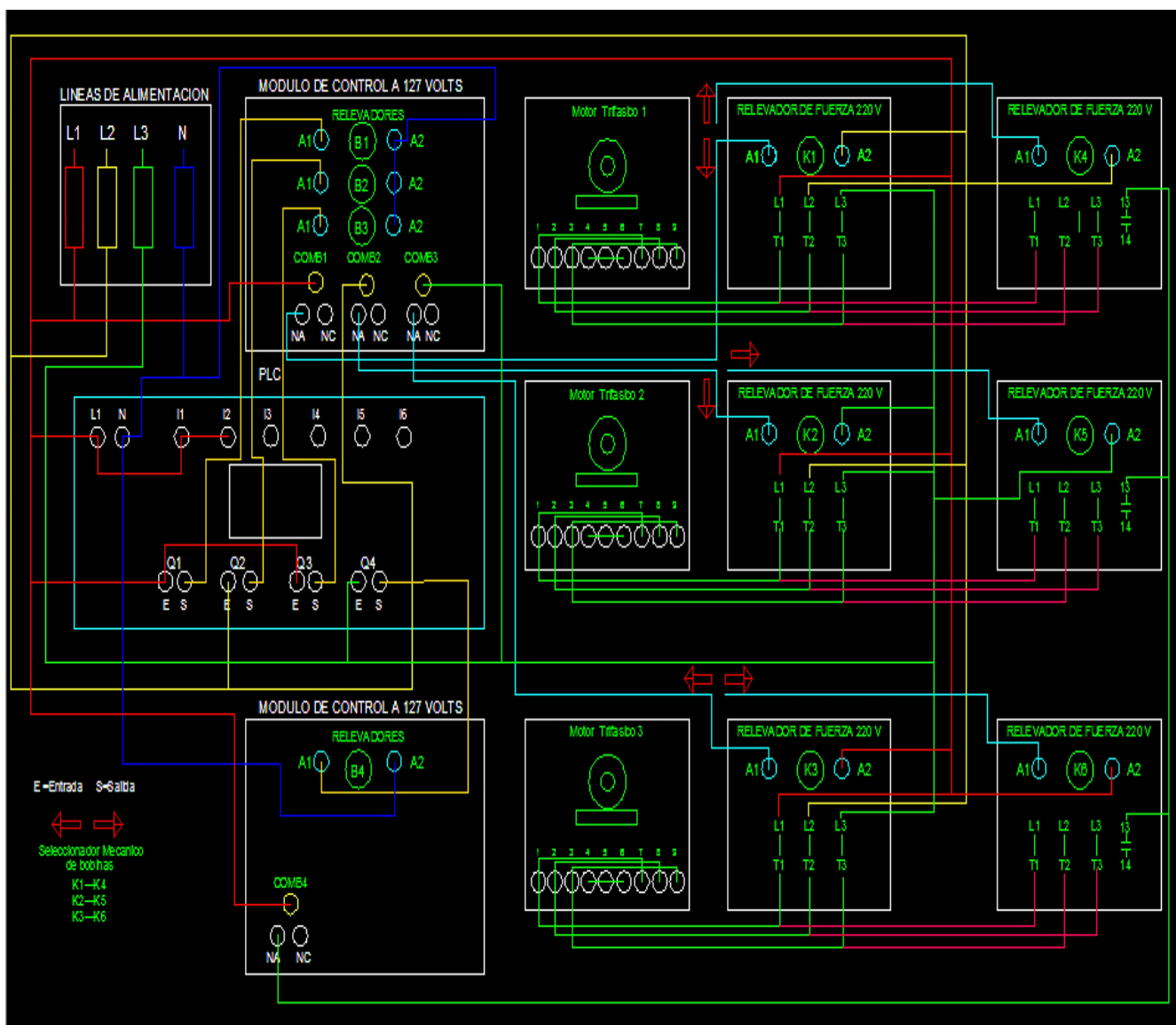


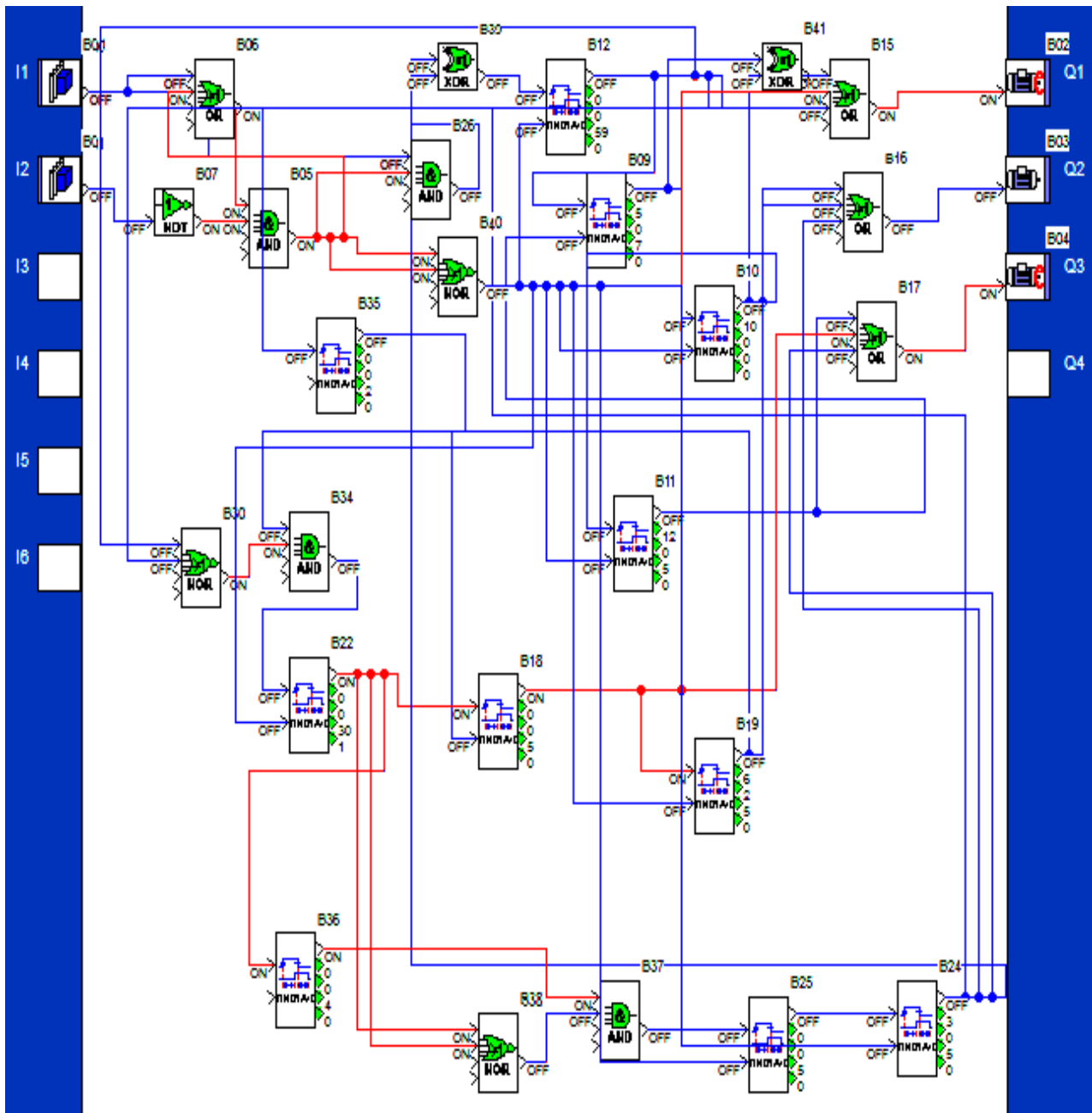
Diagrama 3.1

3.2 Aplicación de sincronización de tres motores mediante secuencias de operación con ciclos de repetición.

En esta aplicación se pone el arranque de tres motores en sincronización en diferentes secuencias de operación, este programa se activa por I1 y se desactiva la secuencia en cualquier punto del programa al pulsar I2 al pulsar I1 se activan las 3 salidas Q1, Q2, Q3 en diferentes tiempos repitiendo el ciclo 2 veces, posteriormente, el programa cambia a otra secuencia activando Q1, Q3 al mismo tiempo al desactivarse se activa Q2 repitiendo el ciclo tres veces, en la última secuencia se activan Q1, Q2, Q3 al mismo tiempo solo una vez, después de un tiempo se repite el programa desde la primera secuencia ciclicamente hasta pulsar I2 en cualquier punto del programa se detiene todo el sistema.

En el Programa al pulsar I1 se activa el arreglo de enclavamiento con las compuertas OR B06, NOT B07 y AND B05 de la salida de esta va a la entrada AND B26 de la salida XOR B39 se conecta al temporizador B12 este temporizador es el que contiene todo el tiempo de los ciclos de la primera secuencia mandando la señal a los temporizadores B09, B10, B11 conectados a las compuertas OR B15, B16, B17 estas compuertas reciben las señales de las diferentes partes de control partes de control del programa.

Al terminar el conteo de desconexión del temporizador B11 tiene un regreso de desconexión al temporizador B09 repitiendo nuevamente el ciclo al terminar por segunda vez el conteo de desconexión del temporizador B11 el contador B12 interrumpe la señal a la compuerta NOR B40 activando la salida de esta a la entrada de la compuerta AND B34 esta compuerta tiene una espera de desconexión del temporizador B12 para que mande el pulso del temporizador B35 a la compuerta AND B34 para activar el temporizador B22 que es el que tiene el tiempo completo de la segunda secuencia activando el temporizador B18 que activa Q1 y Q3 al mismo tiempo a su desconexión activa el temporizador B19 que activa Q2 a su desconexión activa el temporizador B18 repitiendo el ciclo 3 veces al terminar el conteo el temporizador B22 desactiva el temporizador B36 y a la compuerta NOR B38 activando la compuerta AND B37 a la entrada a su salida activa el temporizador B25 transfiriendo al temporizador B24 activando Q1, Q2, Q3 al mismo tiempo una sola vez al acabar su tiempo a la conexión desactiva el temporizador B25 al acabarse su tiempo de desconexión del temporizador B34 activa la compuerta XOR B39 repitiendo nuevamente las secuencias y los ciclos al presionar I2 se detiene la secuencia y los ciclos parando el programa.



Programa 3.2

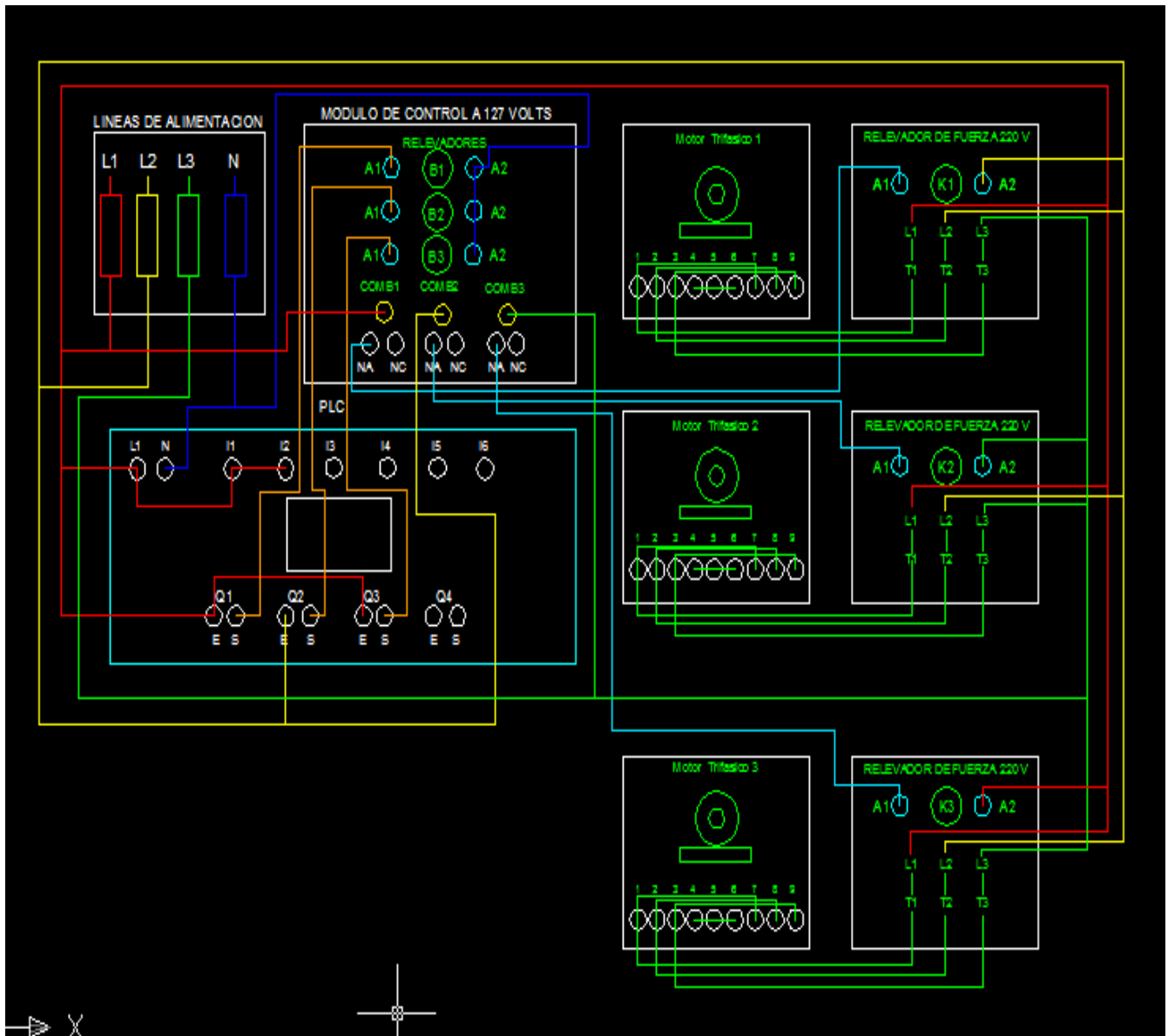


Diagrama 3.2

Diagrama de alimentación y conexión del programa 3.2

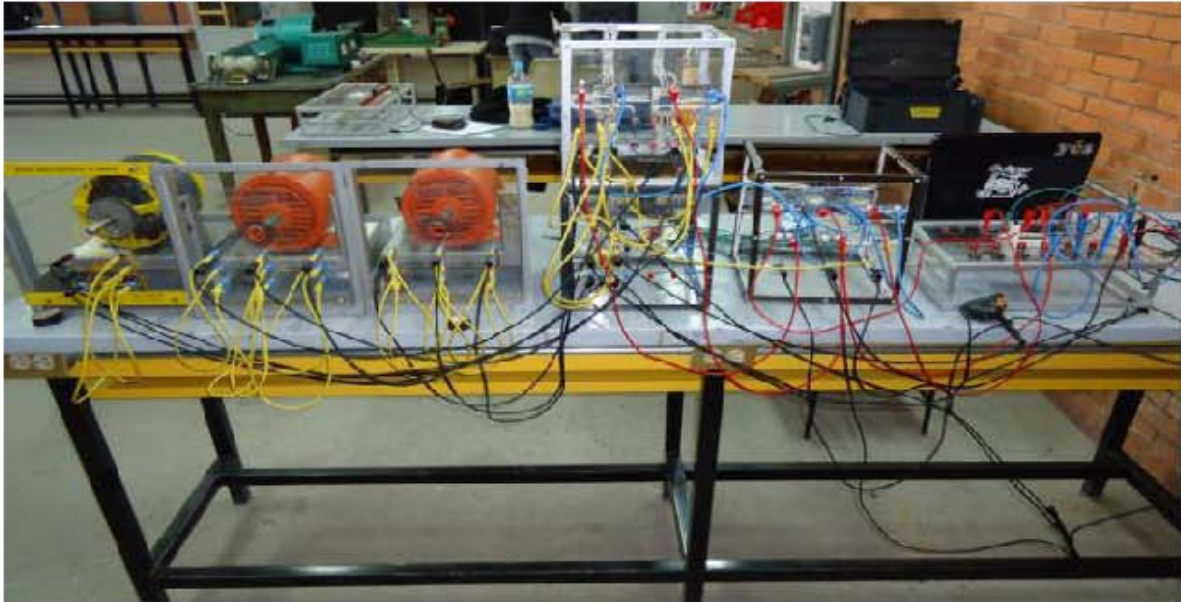


Diagrama Electrico 3.2

Conclusiones

Esta investigación es de mucha importancia ya que es una de las pocas y primeras aportaciones con bases y fundamentos al manejo y control de los módulos inteligentes PLC, para la carrera de Ingeniería Mecánica Eléctrica de la FES-C4 y a la UNAM para que los estudiantes de las Ingenierías tengan la información y el conocimiento de cómo manejar este tipo de dispositivos de una manera y metodología más fácil de comprender.

Al final de la investigación con este módulo lógico se tuvieron las siguientes ventajas:

Menor tiempo empleado en la elaboración de proyectos debido a que:

- No es necesario dibujar el esquema de contactos.
- No es necesario simplificar las ecuaciones lógicas, ya que, por lo general, la capacidad de almacenamiento del módulo de memoria es lo suficientemente grande.
- La lista de materiales queda sensiblemente reducida, y al elaborar el presupuesto correspondiente eliminaremos parte del problema que supone el contar con diferentes proveedores, distintos plazos de entrega, etc.

- . Posibilidad de introducir modificaciones sin cambiar el cableado y añadir aparatos.
- . Mínimo espacio de ocupación
- . Menor coste de mano de obra de la instalación
- . Economía de mantenimiento. Además de aumentar la fiabilidad del sistema, al eliminar contactos móviles, los mismo autómatas pueden detectar e indicar averías.
- . Posibilidad de gobernar varias máquinas con un mismo autómata.
- . Menor tiempo para la puesta de funcionamiento del proceso al quedar reducido el tiempo de cableado.
- . Si por alguna razón la maquina queda fuera de servicio, el autómata útil para otra maquina, o sistema de producción.

APÉNDICE

Programas:

Programa Compuerta NOT Diagrama de escalera	1.1.....	pag 22
Compuerta AND	1.2.....	22
Compuerta NAND	1.3.....	23
Compuerta OR	1.4.....	23
Compuerta NOR	1.5.....	24
Compuerta XOR	1.6.....	24
Programa Compuerta NOT Diagrama de Bloques	1.7.....	26
Compuerta AND	1.8.....	26
Compuerta NAND	1.9.....	27
OR	1.10.....	27
NOR	1.11.....	28
XOR	1.12.....	28
Programa de Aplicación Compuertas Lógicas de la tabla 1.6		
	1.13.....	31
	1.13.1.....	31
	1.13.2.....	32
	1.14.....	34
Programa de Aplicación Compuertas Lógicas de la tabla 1.7		
	1.15.....	38
Programa Función Relevador		
Diagrama de Escalera	2.1.....	39
	2.2.....	40
	2.3.....	40
Diagrama de Bloques	2.4.....	41
Programa Función Temporizador		
Diagrama de Escalera		
	2.6.....	43
	2.7.....	44
Diagrama de Bloques		
	2.8.....	46
	2.9.....	47
	2.10.....	48
Programa Aplicación Función Relevador, Temporizador espera e inicio	2.11.....	49
Programa Aplicación Función Relevador, Temporizador desconexión ,reinicio	2.12.....	51
Programa Aplicación de ejemplo Semáforo	2.13.....	53
Programa Aplicación Sincronización de tres motores trifásicos con diferentes tiempos de operación	3.1.....	58
Programa Aplicación Sincronización de tres motores trifásicos con diferentes tiempos de operación con sentido inverso de giro	3.2.....	59
Programa Aplicación Sincronización de tres motores trifásicos mediante secuencias de operación con ciclos de repetición	3.3.....	62

Bibliografía

- 1- Albert P.Maluino
Donald P.Leach
Alfaomega Principios y aplicaciones digitales
- 2- Balcells, J y Romeral, J.L.
Autómatas programables. Marcombo. 1997
- 3- Bocksnick, Bernd
Fundamentos de la técnica de mando. Festo. 1990
- 4- Manual Edition 03/2009 Logo Simens STEP 7
- 5- Michell, G.
Autómatas programables. Marcombo. 1990
- 6- M.Morris Mano Charles R.Kime
Prentice Hall Fundamentos de diseño lógico
- 7- PLC de Siemens
Alfaomega Una manera fácil de programar.
- 8- Porras, A y Montanero, A. P.
Autómatas programables. McGraw-Hill. 1990
- 10- Roger L. Tokheim
McGraw_Hill Principios digitales.
- 11- Schneider
Documentación técnica TSX Micro
- 12- Simon, A.
Autómatas programables. Paraninfo. 1988.
- 13- Sistema de Automatizacion Simens S7 200
- 14- Ubaldo Ordaz Garcia
Edmundo Rios Miranda
Damian Ordaz Montiel
Trillas Controladores Lógicos Programables

