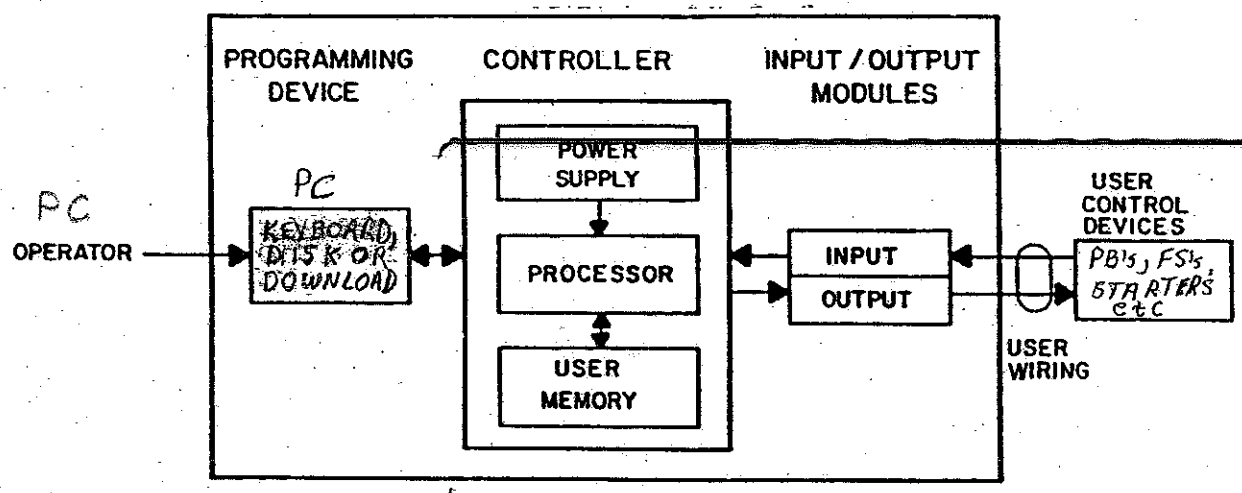
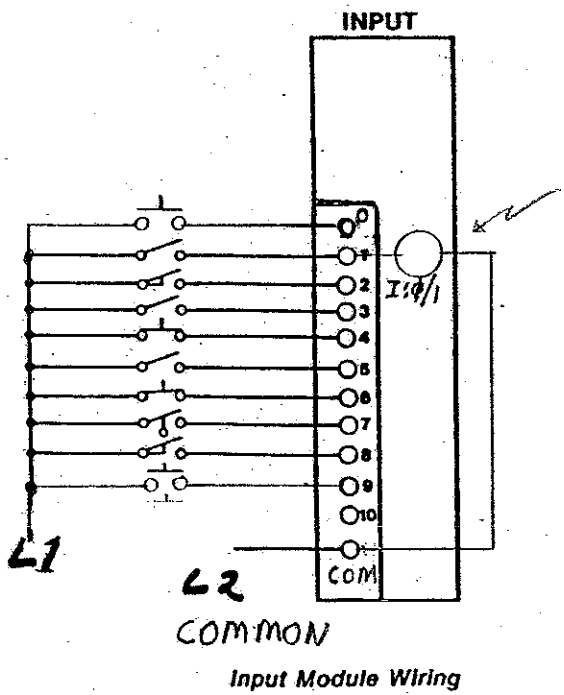
PLC



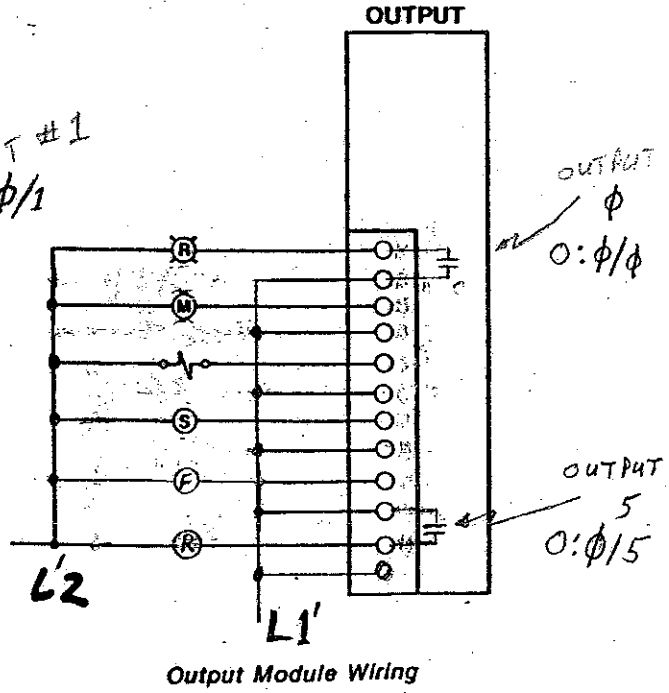
I/O, Controller, System Basic Block Diagram

INPUT LADDER DIAGRAM



Input Module Wiring

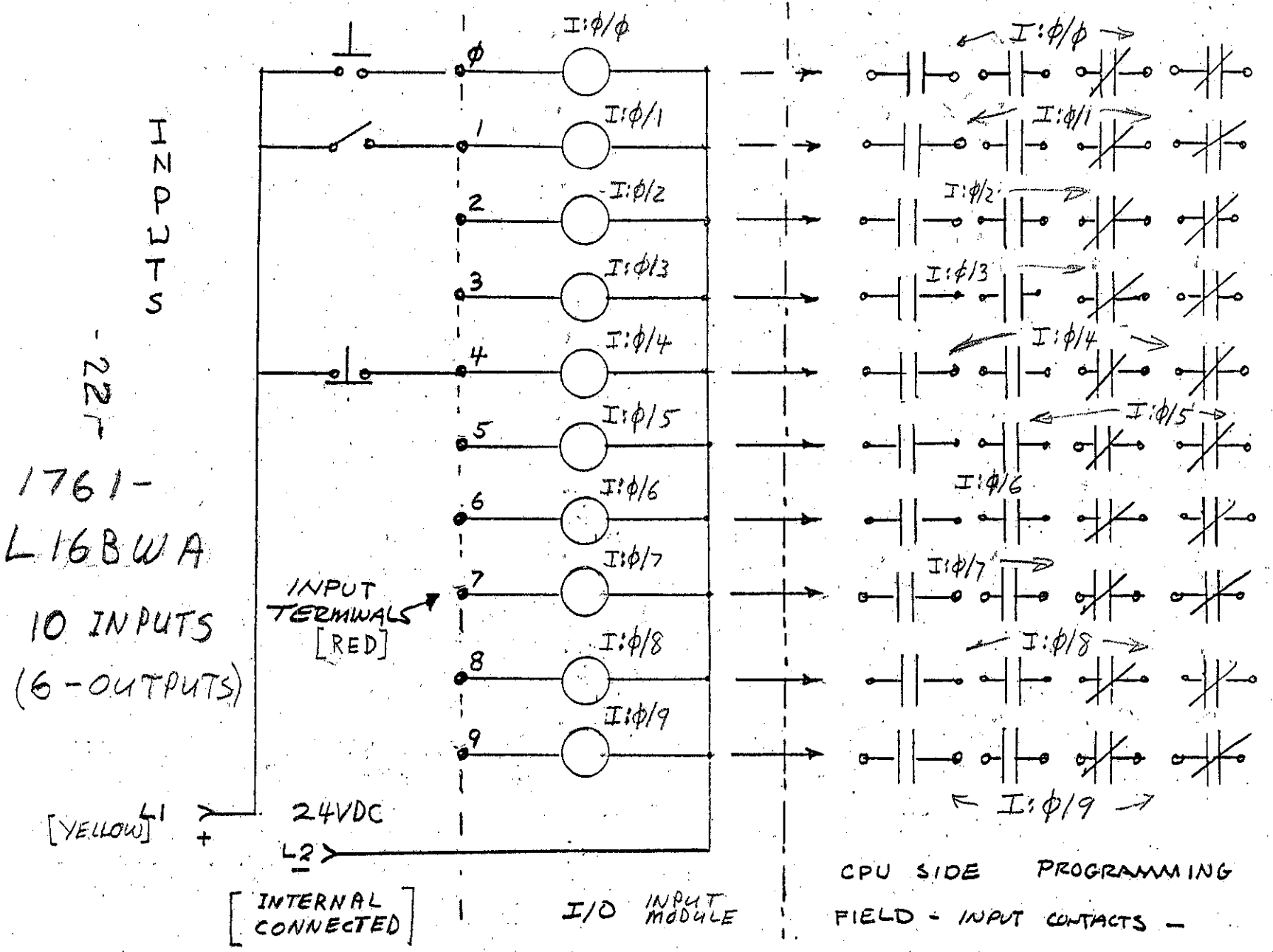
OUTPUT LADDER DIAGRAM



Output Module Wiring

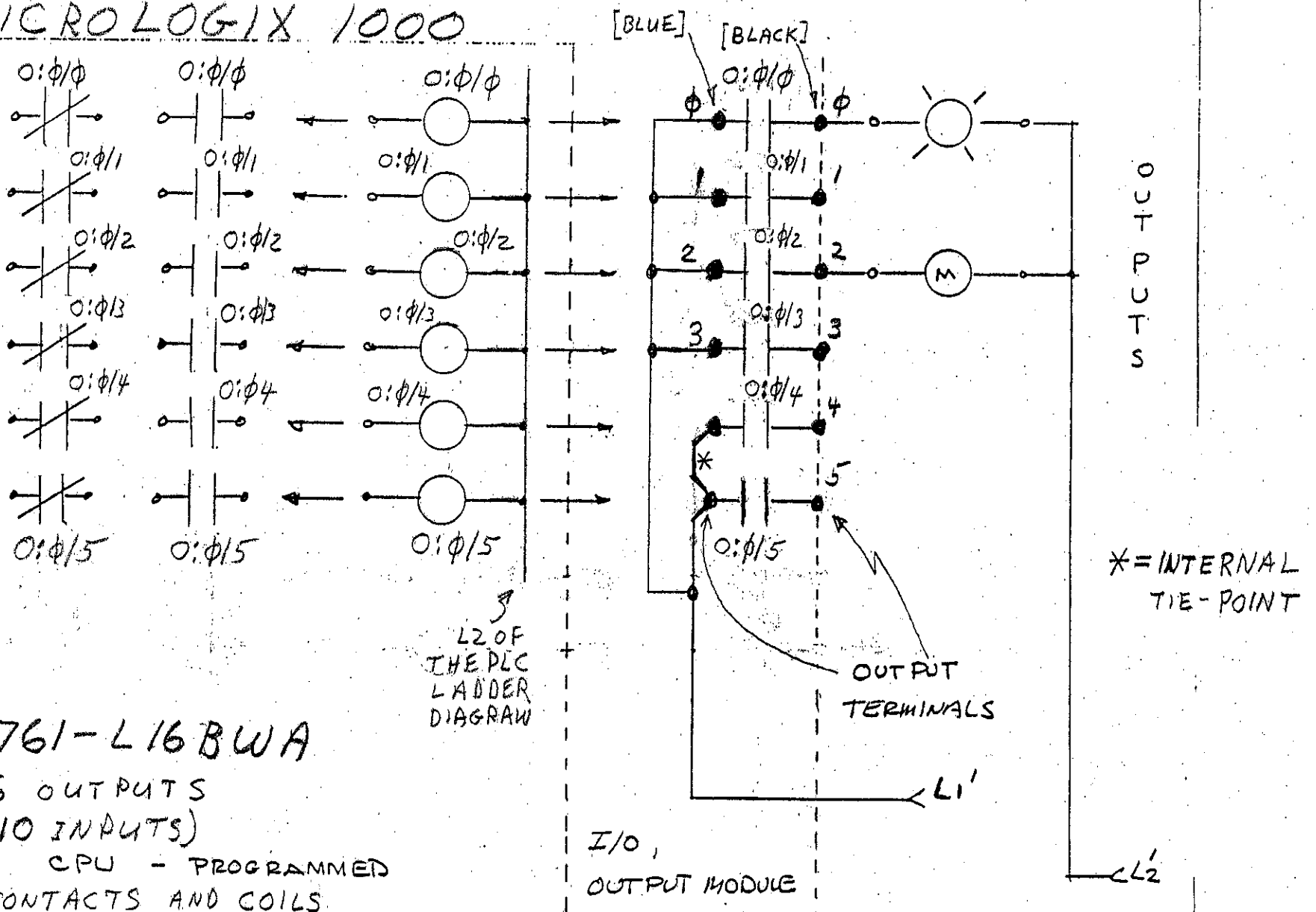
INPUT 1761-L16BW.

ALLEN-BRADLEY MICRO LOGIX 1000



ALLEN-BRADLEY MICROLOGIX 1000

OUTPUT 1761-L16BWA



1761-L16BWA

6 OUTPUTS
(10 INPUTS)
CPU - PROGRAMMED
CONTACTS AND COILS

Program Files

Program files contain controller information, the main ladder program, interrupt subroutines, and any subroutine programs. These files are:

- **System Program** (file 0) – This file contains various system related information and user-programmed information such as processor type, I/O configuration, processor file name, and password.
- **Reserved** (file 1) – This file is reserved.
- **Main Ladder Program** (file 2) – This file contains user-programmed instructions defining how the controller is to operate.
- **User Error Fault Routine** (file 3) – This file is executed when a recoverable fault occurs.
- **High–Speed Counter Interrupt** (file 4) – This file is executed when an HSC interrupt occurs. It can also be used for a subroutine ladder program.
- **Selectable Timed Interrupt** (file 5) – This file is executed when an STI occurs. It can also be used for a subroutine ladder program.
- **Subroutine Ladder Program** (files 6 – 15) – These are used according to subroutine instructions residing in the main ladder program file or other subroutine files.

Data Files

Data files contain the status information associated with external I/O and all other instructions you use in your main and subroutine ladder program files. In addition, these files store information concerning processor operation. You can also use the files to store "recipes" and look–up tables if needed.

These files are organized by the type of data they contain. The data file types are:

- **Output** (file 0) – This file stores the state of the output terminals for the controller.
- **Input** (file 1) – This file stores the status of the input terminals for the controller.
- **Status** (file 2) – This file stores controller operation information. This file is useful for troubleshooting controller and program operation.
- **Bit** (file 3) – This file is used for internal relay logic storage.
- **Timer** (file 4) – This file stores the timer accumulator and preset values and status bits.
- **Counter** (file 5) – This file stores the counter accumulator and preset values and the status bits.
- **Control** (file 6) – This file stores the length, pointer position, and status bits for specific instructions such as shift registers and sequencers.
- **Integer** (file 7) – This file is used to store numeric values or bit information.

Addressing Data Files

For the purposes of addressing, each data file type is identified by a letter (identifier) and a file number.

File Type	Identifier	File Number
Output	O	0
Input	I	1
Status	S	2
Bit	B	3
Timer	T	4
Counter	C	5
Control	R	6
Integer	N	7

The addresses are made up of alphanumeric characters separated by delimiters. Delimiters include the colon, slash, and period.

Specifying Logical Addresses

The format of a logical address, `xf:e`, corresponds directly to the location in data storage.

Where:	Is the:		
x	File type:	O—output I—input S—status B—binary	T—timer C—counter R—control N—integer
f	File #:	0—output 1—input 2—status 3—binary	4—timer 5—counter 6—control 7—integer
:	Element delimiter: numbers	Colon or semicolon delimiter separates file and structure/word	
e	Element number: 0 to:	0—output 1—input 32—status 31—binary	39—timer 31—counter 15—control 104—integer

You assign logical addresses to instructions from the highest level (element) to the lowest level (bit). Addressing examples are shown in the table below.

To specify the address of a:	Use these parameters:①
Word within an integer file	N 7:2 File Type File Number
Word within a timer file	T 4 : 7 . ACC File Type File Number File Delimiter Structure Number Delimiter Word
Bit within an integer file	N 7 : 2 / 5 File Type File Number File Delimiter Word Number Bit Delimiter Bit Number
Bit within a bit file	B 3 / 31 Bit Delimiter Bit Number Bit files are bit stream continuous files, and therefore you can address them in two ways: by word and bit, or by bit alone.
Bit within a control file	R 6 : 7 / D N File Type File Number File Delimiter Structure Number Delimiter Mnemonic

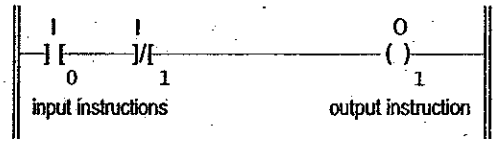
① The addressing parameters required when using the HHP differ slightly from those required when using the programming software.

You can also address at the bit level using mnemonics for timer, counter, or control data types. The available mnemonics depend on the type of data. See chapters 7 through 13 for more information.

Applying Ladder Logic to Your Schematics

The logic you enter into the micro controller makes up a ladder program. A ladder program consists of a set of instructions used to control a machine or a process.

Ladder logic is a graphical programming language based on electrical relay diagrams. Instead of having electrical rung continuity, ladder logic is looking for logical rung continuity. A ladder diagram identifies each of the elements in an electromechanical circuit and represents them graphically. This allows you to see how your control circuit operates before you actually start the physical operation of your system.

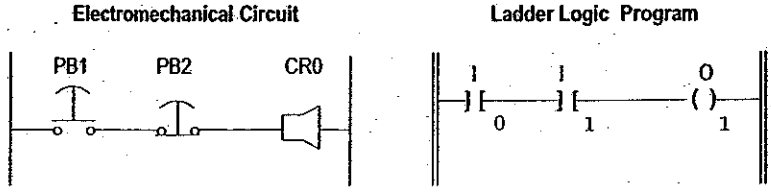


In a ladder diagram each of the input devices are represented in series or parallel combinations across the rung of the ladder. The last element on the rung is the output that receives the action as a result of the conditional state of the inputs on the rung.

Each output instruction is executed by the controller when the rung is scanned and the conditions on the rung are true. When the rung is not scanned or the logic conditions on the rung do not create a true logic path, the output is not executed.

The software allows you to enter a ladder logic program into the micro controller.

In the following illustration, the electromechanical circuit shows two pushbuttons wired in series with an alarm horn. PB1 is a normally open pushbutton, and PB2 is normally closed. This same circuit is shown in ladder logic by two contacts wired in series with an output. Contact I/0 and I/1 are examine–if–closed instructions.[1]



[1] Contact I1 would be an examine–if–open instruction ( ]/[ ) if PB2 was a normally open electromechanical circuit.

The following tables show how the above circuits operate. The tables represent all the possible conditions for the electromechanical circuit, each of the inputs, and the resulting output state in each of the associated ladder programs.

Electromechanical Circuit State Table

PB1 (Condition)	PB2 (Condition)	Alarm Horn (Action)
not pushed	not pushed	silent
not pushed	pushed	silent
pushed	not pushed	alarm
pushed	pushed	silent

Instruction Program State Table

I0 (Condition)	I1 (Condition)	O1 (Action)
0	0	silent
0	1	silent
1	0	silent
1	1	alarm

Examine if Closed (XIC)

Execution Times (µsec) when:

True	False
1.54	1.72

From the Instruction Type display, press:

Use the XIC instruction in your ladder program to determine if a bit is On. When the instruction is executed, if the bit addressed is on (1), then the instruction is evaluated as true. When the instruction is executed, if the bit addressed is off (0), then the instruction is evaluated as false.

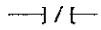
Bit Address State	XIC Instruction
0	False
1	True

Examples of devices that turn on or off include:

- a push button wired to an input (addressed as I1:0/4)
- an output wired to a pilot light (addressed as O0:0/2)
- a timer controlling a light (addressed as T4:3/DN)

Entering the Instruction

Examine if Open (XIO)



Execution Times (µsec) when:

True	False
1.54	1.72

From the Instruction Type display, press:

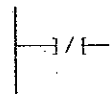
Use an XIO instruction in your ladder program to determine if a bit is Off. When the instruction is executed, if the bit addressed is off (0), then the instruction is evaluated as true. When the instruction is executed, if the bit addressed is on (1), then the instruction is evaluated as false.

Bit Address State	XIO Instruction
0	True
1	False

Examples of devices that turn on or off include:

- motor overload normally closed (N.C.) wired to an input (I1:0/10)
- an output wired to a pilot light (addressed as O0:0/4)
- a timer controlling a light (addressed as T4:3/DN)

Entering the Instruction



Output Energize (OTE)

—( )—

Execution Times (µsec) when:

True	False
4.43	4.43

Use an OTE instruction in your ladder program to turn On a bit when rung conditions are evaluated as true.

An example of a device that turns on or off is an output wired to a pilot light (addressed as O0:0/4).

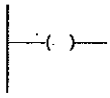
OTE instructions are reset when:

- You enter or return to the REM Run or REM Test mode or power is restored.
- The OTE is programmed within an inactive or false Master Control Reset (MCR) zone.

Important: A bit that is set within a subroutine using an OTE instruction remains set until the subroutine is scanned again.

Entering the Instruction

From the Instruction Type display, press:



Output Latch (OTL) and Output Unlatch (OTU)

—(L)—

—(U)—

Execution Times (µsec) when:

	True	False
OTL	4.97	3.16
OTU	4.97	3.16

OTL and OTU are retentive output instructions. OTL can only turn on a bit, while OTU can only turn off a bit. These instructions are usually used in pairs, with both instructions addressing the same bit.

Your program can examine a bit controlled by OTL and OTU instructions as often as necessary.

ATTENTION: Under fatal error conditions, physical outputs are turned off. Once the error conditions are cleared, the controller resumes operation using the data table value of the operand.

Using OTL

When you assign an address to the OTL instruction that corresponds to the address of a physical output, the output device wired to this screw terminal is energized when the bit is set (turned on or enabled).

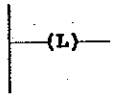
When rung conditions become false (after being true), the bit remains set and the corresponding output device remains energized.

When enabled, the latch instruction tells the controller to turn on the addressed bit. Thereafter, the bit remains on, regardless of the rung condition, until the bit is turned off (typically by a OTU instruction in another rung).

Entering the Instruction

From the Instruction Type display, press:

BIT F1 OTL -(L)- F4



Using OTU

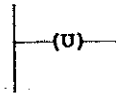
When you assign an address to the OTU instruction that corresponds to the address of a physical output, the output device wired to this screw terminal is de-energized when the bit is cleared (turned off or disabled).

The unlatch instruction tells the controller to turn off the addressed bit. Thereafter, the bit remains off, regardless of the rung condition, until it is turned on (typically by a OTL instruction in another rung).

Entering the Instruction

From the Instruction Type display, press:

BIT F1 OTU -(U)- F5



>RSLogix 500<

MLogix 1000™ COMMANDS AND ADDRESS FORMAT

INPUTS: (BIT INSTRUCTIONS)

—| |— XIC = EXAMINE IF CLOSED (N.O.)

—|/|— XIO = EXAMINE IF OPEN (N.C.)

ADDRESS SET UP I: XXX / CHANNEL #

XXX ⇒ CORRESPONDS TO THE PHYSICAL LOCATION OF THE PARTICULAR INPUT i.e. TRACK #, SLOT #, PORT

- IN HOUSE SYSTEM OF ADDRESSING (TYPING FORMAT) ⇒ I: Ø / CHAN #
  - * 1761-L16BWA ⇒ Ø TO 9
  - * 1761-L32BWA ⇒ Ø TO 19

EX PB @ CHANNEL 5 ⇒ I: Ø / 5

LIMIT SWITCH @ CHANNEL Ø ⇒ I: Ø / Ø

NOTE THE ABOVE ADDRESSING SCHEME ALLOWS ACCESS TO A "WORD" (16 BIT) OR A "BIT" OF A WORD (OTHER ADDRESSING EXISTS)

WORD ADDRESS: N7:2

BIT OF A WORD: N7:2/5 ← BIT #

μLOGIX™ 1000

OUTPUTS: (BIT INSTRUCTIONS)

—| |— ⇒ XIC —|/|— XIO

—( )— ⇒ OTE = OUTPUT ENERGIZE

—(L)— ⇒ OTL = OUTPUT LATCH

—(U)— ⇒ OTU = OUTPUT UNLATCH

ADDRESS SYNTAX ⇒ O:φ/CHANNEL#

1761-L16BWA ⇐ φ TO 5

φ TO 11 ⇒ 1761-L32BWA

EX

O:φ
—(L)—
φ

⇒ LATCH OUTPUT AT OUTPUT CHANNEL #φ

O:φ
—( )—
4

⇒ NON-LATCHING OUTPUT AT OUTPUT CHAN # 4

MLOGIX1000™

INTERNAL COILS: OTE, OTL, OTU

NONLATCH — LATCH/UNLATCH

ADDRESSING:
BIT ADDRESS FORMAT (BIT FILE)

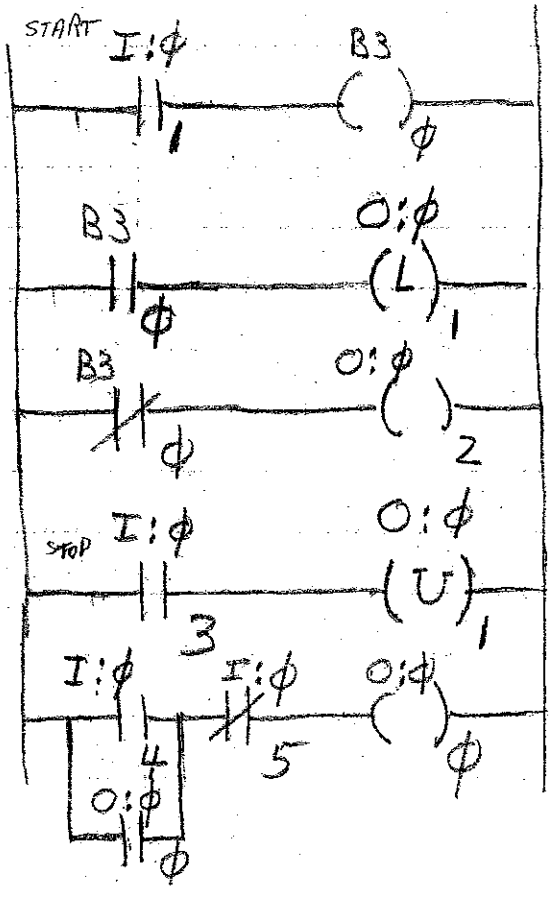
B3/ COIL #

↑ 0 TO 511 (512 BITS)

EXAMPLE
B3/0 = INTERNAL COIL #0
B3/14 = INTERNAL COIL #14

• OF COURSE CONTACT(S) OF A INTERNAL COIL HAVE THE SAME ADDRESS

EXAMPLE

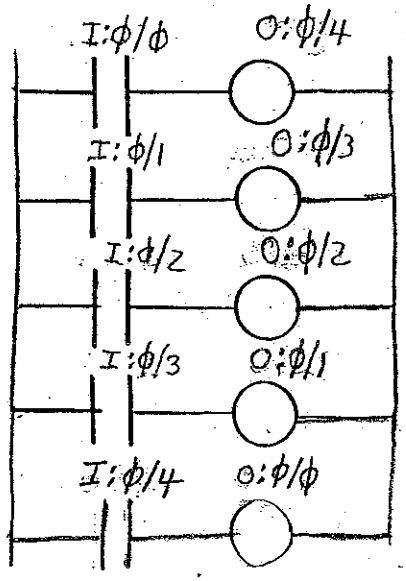


INPUT #1, INTERNAL COIL 0

EXERCISE 1-

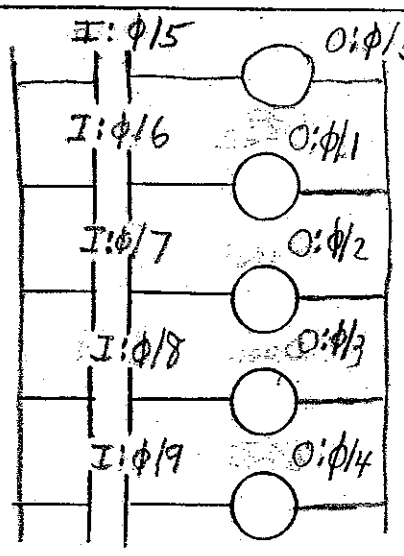
-I/O TEST PROGRAM -

TEST 1



INPUT #0 CONTROLS OUTPUT #4

TEST 2



CLEAR LADDER DIAGRAM, START NEW PROGRAM

INPUT #9 CONTROLS OUTPUT #4

NOTE: PLC ALSO HAS "INTERNAL" COILS AVAILABLE FOR CONTROL RELAY FUNCTIONS. ADDRESS FOR THE "INTERNAL COILS" RANGE FROM B3/0 TO B3/511.