

# **Logic Block Form Instructions**

**MC12-500**

---



**Implementation  
Hiway Gateway - 2**

***Logic Block  
Form Instructions***

**MC12-500  
9/95**

---

# Copyright, Trademarks, and Notices

Printed in U.S.A. — © Copyright 1995 by Honeywell Inc.

Revision 01 - September 15, 1995

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any indirect, special or consequential damages. The information and specifications in this document are subject to change without notice.

---

---

## **About This Publication**

This publication provides a guide for completing the Logic Block Configuration Form MC88-501 and the Logic Block Comments Entry Form MC88-502.

This publication supports Release 500.



---

# Table of Contents

---

**1 INTRODUCTION**

- 1.1 Purpose
- 1.2 Scope
- 1.3 Prerequisites
- 1.4 Related Documents

**2 OVERVIEW**

**3 SPECIFIC INSTRUCTIONS**

- 3.1 Logic Block Configuration Form
- 3.2 Logic Block Comments Entry Form



## INTRODUCTION Section 1

### 1.1 PURPOSE

This publication provides general and specific instructions for completing the Logic Block Configuration Form and the Logic Block Configuration Comments Form. The instructions include a sample Logic Block Configuration Form and a sample Comments Form. The Logic Block Configuration Form is designed to facilitate collection and organization of the data required to define logic blocks, which are functional entities executed in Multifunction Controllers. The logic block definitions (up to 128 for each Multifunction Controller) are part of the database that is configured for, and down-line loaded to, Multifunction Controllers (Figure 1-1).

### 1.2 SCOPE

This publication describes the method of defining logic blocks on 8 1/2 x 11 inch paper forms that facilitate later entry into the TDC 3000<sup>X</sup> system. It also describes the method of entering comments regarding logic block use on paper forms that facilitate entry into the system.

### 1.3 PREREQUISITES

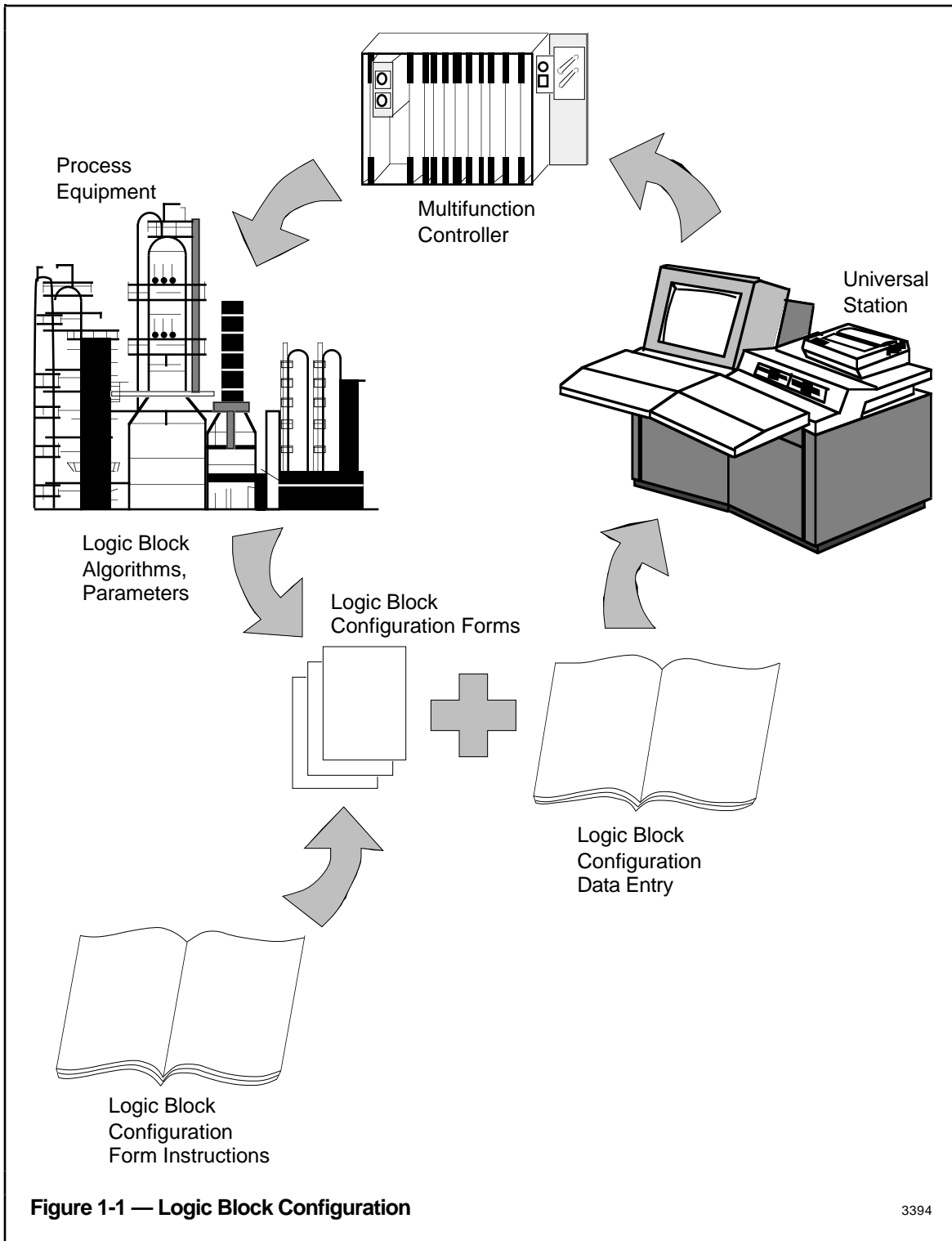
Before using this publication for filling in configuration and comments forms, the user should

- understand the fundamentals of TDC 3000<sup>X</sup> System configuration, particularly as they apply to the Multifunction Controller, and specifically to logic blocks. (Refer to the *Configuration Data Collection Guide*).
- determine, for a given Multifunction Controller, all process-related and special logic requirements, and record these requirements in lists and/or appropriate diagrams.

### 1.4 RELATED DOCUMENTS

The following publications should be available for reference while using this publication:

| Title                                  | Publication No. | Binder  |
|--|-----------------|---|
| Configuration Data<br>Collection Guide | SW12-500        | Implementation/Startup &<br>Reconfiguration - 2 |
| Logic Block Data Entry                 | MC11-500        | Implementation/Hiway Gateway - 2                |



## OVERVIEW Section 2

A Logic Block is similar in function to a logic element that can be implemented in dedicated hardware such as binary electronic circuitry. Logic Blocks are actually firmware entities in the Multifunction Controller that act on details of logic definition (algorithms, inputs, and outputs) that are entered into RAM by Logic Block Configuration Data Entry.

Each Multifunction Controller can process up-to-128 logic blocks. Any of the 128 logic blocks can be configured to execute any one of 12 algorithms. The 12 algorithms are described in Section 3, Table 3-3. They are listed by logic block type, which is determined by the number of inputs and outputs that can be configured for a given logic block.

The Engineer who is responsible for configuring logic blocks in a given Multifunction Controller should proceed generally as follows:

1. Determine the logic function requirements of the Multifunction Controller to be configured. Considerations should include interactions between the Multifunction Controller in question and other system components, particularly other Multifunction Controllers on the same C-Link, and interactions between logic-block functions and functions included in all programs that are intended for use in the Multifunction Controller.
2. Describe and record logic-block use in generic terms, using traditional methods such as logic diagrams, sequence diagrams, etc., that are familiar to the engineer who is doing the logic design.
3. For each of the (up-to-128) logic blocks, translate details of each logic element into logic conventions used for logic-block definition, and enter this data on the Logic Block Configuration Forms supplied with the system.
4. On the Logic Block Comments Sheet(s) provided with the system, describe functions of groups of interconnected logic blocks, and the relationships between functional groups of logic blocks. Because logic blocks are entered on the forms and into the system as single entities in which the intended use of related blocks may not be clear, it is important that a record of design intent is made at the time of original design. This is particularly important when design changes must be made significantly later than the original design, and by personnel other than the original designers.

### NOTE

The comments should be recorded on the sheets provided, because these sheets facilitate entry into the system (through the Text File Editor function) for convenient retrieval.

Graphic representations of four types of logic blocks, completed configuration and comment forms built from this data, and an associated Text Builder display are depicted in Figure 2-1.

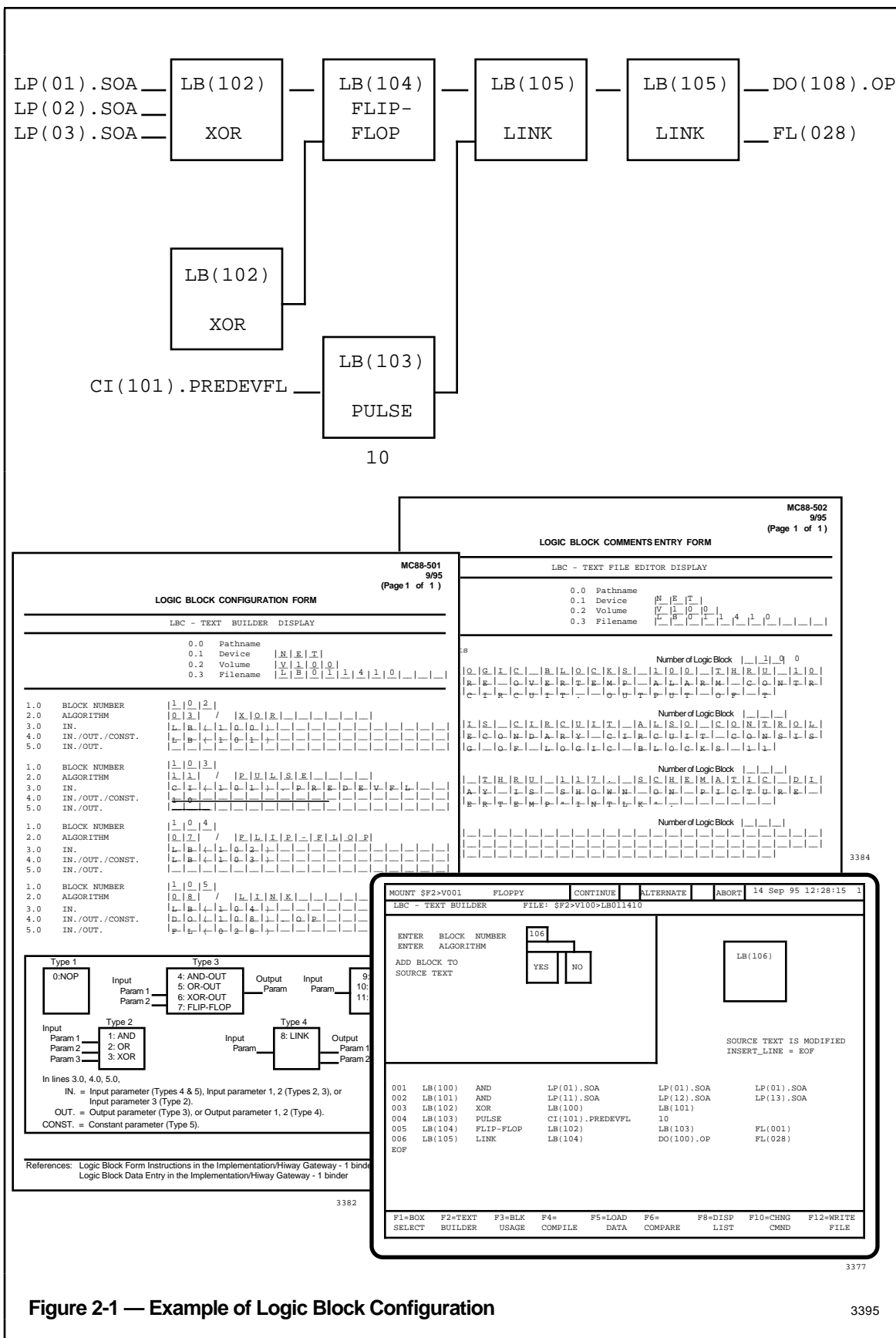


Figure 2-1 — Example of Logic Block Configuration

3395

## SPECIFIC INSTRUCTIONS Section 3

### 3.1 LOGIC BLOCK CONFIGURATION FORM

The Logic Block Configuration Form, shown on the following page, is printed on an 8 1/2 by 11 inch sheet. Each page is formatted to collect all the data required to define four logic blocks. The form is divided into three parts, in the following manner:

|                             |   |
|-----------------------------|---|
| Lines<br>0.0<br>thru<br>0.3 | Pathname—provides data entry areas for the system identifiers with which the logic block is associated (device, volume, and filename). This data is not actually part of the data required to define logic blocks; its primary function is to enable completed forms to be grouped for later entry of data into the system. |
| Lines<br>1.0<br>thru<br>5.0 | Configuration Lines—in which the word "lines" refers to data entry areas on the form and that correspond to the Logic Block Configurator - Text Builder display (refer to the <i>Logic Block Data Entry</i> publication for additional details). Each set of lines 1.0 through 5.0 defines a single logic block.            |
| Logic<br>Block<br>Types     | Graphic representations of the five different types of logic blocks, including available and/or required inputs and outputs, and a list of the algorithms included in each type.  |

Instructions for filling out the Logic Block Configuration Form are presented in Table 3-1. The instructions in this table are sequenced in the same order as they are presented on the form and on the Logic Block Configurator - Text Builder display, and each instruction is referenced by a line number that appears on the form.

Definitions of the entries required for the algorithm and its inputs and outputs are listed in Tables 3-2 through 3-4.

LOGIC BLOCK CONFIGURATION FORM

LBC - TEXT BUILDER DISPLAY

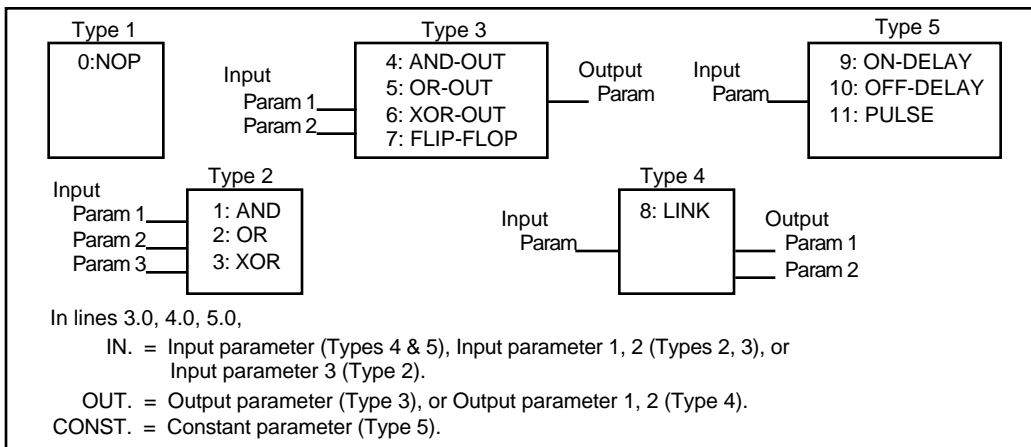
|     |          |                               |
|-----|----------|-------------------------------|
| 0.0 | Pathname |                               |
| 0.1 | Device   | N   E   T   _   _   _   _   _ |
| 0.2 | Volume   | V   1   0   0   _   _   _   _ |
| 0.3 | Filename | L   B   0   1   1   4   1   0 |

|     |                 |   |
|-----|-----------------|---|
| 1.0 | BLOCK NUMBER    | 1   0   2   |
| 2.0 | ALGORITHM       | 0   3   /   X   O   R   _   _   _   _                 |
| 3.0 | IN.             | L   B   (   1   0   0   )   _   _   _   _   _   _   _ |
| 4.0 | IN./OUT./CONST. | L   B   (   1   0   1   )   _   _   _   _   _   _   _ |
| 5.0 | IN./OUT.        | _   _   _   _   _   _   _   _   _   _   _   _         |

|     |                 |   |
|-----|-----------------|---|
| 1.0 | BLOCK NUMBER    | 1   0   3   |
| 2.0 | ALGORITHM       | 1   1   /   P   U   L   S   E   _   _   _   _                         |
| 3.0 | IN.             | C   I   (   1   0   1   )   _   P   R   E   D   E   V   F   L   _   _ |
| 4.0 | IN./OUT./CONST. | 1   0   _   _   _   _   _   _   _   _   _   _                         |
| 5.0 | IN./OUT.        | _   _   _   _   _   _   _   _   _   _   _   _                         |

|     |                 |   |
|-----|-----------------|---|
| 1.0 | BLOCK NUMBER    | 1   0   4   |
| 2.0 | ALGORITHM       | 0   7   /   F   L   T   P   =   F   L   O   P         |
| 3.0 | IN.             | L   B   (   1   0   2   )   _   _   _   _   _   _   _ |
| 4.0 | IN./OUT./CONST. | L   B   (   1   0   3   )   _   _   _   _   _   _   _ |
| 5.0 | IN./OUT.        | _   _   _   _   _   _   _   _   _   _   _   _         |

|     |                 |   |
|-----|-----------------|---|
| 1.0 | BLOCK NUMBER    | 1   0   5   |
| 2.0 | ALGORITHM       | 0   8   /   L   T   N   K   _   _   _   _             |
| 3.0 | IN.             | L   B   (   1   0   4   )   _   _   _   _   _   _   _ |
| 4.0 | IN./OUT./CONST. | D   O   (   1   0   8   )   .   O   P   _   _   _   _ |
| 5.0 | IN./OUT.        | F   L   (   0   2   8   )   _   _   _   _   _   _   _ |



REFERENCES: Logic Block Form Instructions in the Implementation/Hiway Gateway - 1 binder  
 Logic Block Data Entry in the Implementation/Hiway Gateway - 1 binder

Table 3-1 — Entries for Logic Block Configuration Form

| LINE                              | NAME            | DATA ENTRY   | REMARKS  |
|-----------------------------------|-----------------|--|--|
| <b>LBC - TEXT BUILDER DISPLAY</b> |                 |  |  |
| 0.0                               | Pathname        |  | Enter User Pathname  |
| 0.2                               | Volume          | \$Fn (floppy drive n)<br>XXXX (user-defined in<br>Network configuration)   |  |
| 0.3                               | Filename        | LBxyyzz, where<br>xx = Hiway Number<br>(01-20)<br>yy = Box Number<br>(01-31)<br>zz = Serial Number<br>(00-99)  | Configured in Network.<br><br>Box Number of Multifunction<br>Controller.<br>User number for revision<br>level of this filename.  |
| 1.0                               | BLOCK NUMBER    | Enter three digits, from<br>001 through 128.   | Block Number must be<br>unique within a given<br>Multifunction Controller.<br>Logic blocks that are inputs<br>must have a block number<br>lower than that of the blocks<br>for which they are inputs.  |
| 2.0                               | ALGORITHM       | Enter one or two digits<br>(0 through 11) that cor-<br>respond to the desired<br>algorithm or enter algorithm<br>name.   | Algorithm numbers/names<br>are shown on forms. Numbers,<br>names and descriptions<br>are in Table 3-2.   |
| 3.0                               | IN.             | Enter up-to-18 characters<br>to select the desired input<br>parameter.   | In Line 3.0, IN. = Input<br>parameter (number un-<br>specified in Types 4 and 5; 1<br>in Types 2 and 3. Allowable<br>input parameters are in Table<br>3-3.   |
| 4.0                               | IN./OUT./CONST. | Enter:<br>- up-to-18 characters to<br>select the desired input<br>parameter (IN.),<br>or<br>- up-to-17 characters to<br>select the desired output<br>parameter, (OUT.),<br>or<br>- an integer number<br>between 0001 and 9999<br>to specify a time constant<br>(CONST.). | In Line 4.0, IN. = Input<br>parameter 2 (Types 2 and<br>3); OUT. = Output parameter<br>(Type 3), or Output<br>parameter 1 (Type 4);<br>CONST. = Constant parameter<br>(Type 5).<br>Allowable parameters for<br>- Inputs are in Table 3-3<br>- Outputs are in Table 3-4<br><br>Constant parameter specifies<br>time in seconds. |
| 5.0                               | IN./OUT.        | Enter:<br>- up-to-18 characters<br>to select the desired<br>input parameter (IN.),<br>or<br>- up-to-17 characters<br>to select the desired<br>output parameter<br>(OUT.).  | In Line 5.0, IN. = Input<br>parameter 3 (Type 2);<br>OUT = Output parameter<br>(unspecified in Type 3,<br>Output parameter 2 in<br>Type 4).<br>Allowable parameters for<br>- Inputs are in Table 3-3<br>- Outputs are in Table 3-4   |

Table 3-2 — Logic Block Algorithms

| ALGORITHM                      |           | FUNCTION               | COMMENTS  |
|--------------------------------|-----------|------------------------|---|
| NO.                            | NAME      |                        |   |
| Type 1 - No Input/No Output    |           |                        |   |
| 0                              | NOP       | No operation           | No operation, but NOP blocks have logic block numbers like other logic blocks.  |
| Type 2 - Three Input/No Output |           |                        |   |
| 1                              | AND       | Logical AND            | All configured inputs must be active (1-state) to produce active state of the block. At least two inputs must be configured. Block state is available for test by programs or as input to another logic block, but is not usable for direct control action (as for interlocks). |
| 2                              | OR        | Logical OR             | Any configured input must be active (1-state) to produce active (1-state) block state. At least one input must be configured. Block state is available for test by programs or as input to another logic block, but isn't usable for direct control action (e.g. interlocks).   |
| 3                              | XOR       | Exclusive OR           | Only one of configured inputs may be active (1-state) to produce active block state. At least two inputs must be configured. Block state is available for test by programs or as input to another logic block, but is not directly usable for control (as for interlocks).      |
| Type 3 - Two Input/One Output  |           |                        |   |
| 4                              | AND-OUT   | Logical AND            | Comments for AND algorithm (above) apply, except that block state is available through the specified output parameter for control action.   |
| 5                              | OR-OUT    | Logical OR             | Comments for OR algorithm (above) apply, except that block state is available through the specified parameter for control action.   |
| 6                              | XOR-OUT   | Logical OR             | Comments for XOR algorithm (above) apply, except that block state is available through the specified output for control action.   |
| 7                              | FLIP-FLOP | Binary storage element | Input parameter 1 active (1-state) sets the flip-flop logic block and element specified output parameter to active state (1-state). Input parameter 2 active resets to 0-state.   |
| Type 4 - One Input/Two Output  |           |                        |   |
| 8                              | LINK      | Signal distribution    | Input parameter active (1-state) causes both output parameters to be active.  |
| Type 5 - One Input/Constant    |           |                        |   |
| 9                              | ON-DELAY  | Delay                  | Input parameter going active causes block state to go active after the delay time (seconds) specified by the Constant parameter. The block state is available for test by programs or as an input to another logic block, but is not directly available for control action.     |
| 10                             | OFF-DELAY | Delay                  | Input parameter going active causes block state to go inactive after the delay time (seconds) specified by the Constant parameter. Block state is available to programs or as an input to another logic block, but is not directly available for control action.                |
| 11                             | PULSE     | Pulse                  | Input going active causes block to be active for the time (seconds) specified by the Constant parameter. Block state is available for test by programs and as input to another logic block, but is not directly usable for control action.                                      |

Table 3-3 — Input Parameters

| POINT TYPE         | FORMAT      | PARAMETER   | VALUE  | DESCRIPTION   |
|--------------------|-------------|---|--|---|
| Modulating Control | LP(xx).zz   | xx : slot number<br>zz : variable code                        | 01-16<br>PTINAL<br>ALHIFL<br>ALLOFL<br>SOA<br>SOB<br>LM<br>OMAN<br>OAUTO<br>OCAS<br>OCOM<br>INITMAN<br>PMAN<br>PAUTO<br>PCAS<br>PCOM<br>PROG | Alarm State<br>Hi alarm<br>Lo alarm<br>DO 1 (LO)<br>DO 2 (HI)<br>Loop-module manual mode<br>Manual mode<br>Auto mode<br>Cascade mode<br>Computer mode<br>Initialization manual mode<br>Program manual mode<br>Program auto mode<br>Program cascade mode<br>Program computer mode<br>Program attribute |
| Digital input      | DI(xxyy).zz | xx : slot number<br>yy : subslot number<br>zz : variable code | 01-16<br>01-16<br>PV<br>PTINAL   | Digital input state<br>Alarm state  |
| Digital output     | DO(xxy).zz  | xx : slot number<br>y : subslot number<br>zz : variable code  | 01-16<br>1-8<br>OP   | Digital output state  |
| Counter input      | CI(xxy).zz  | xx : slot number<br>y : subslot number<br>zz : variable code  | 01-16<br>1-4<br>STATE<br>RESET<br>STRTSTOP<br>PRESETFL<br>PREDEVFL   | Counter active state<br>same as STATE<br>same as STATE<br>Alarm state<br>Pre-alarm state  |
| Analog input       | AI(xx).zz   | xx : AI point number<br>zz : variable code                    | 01-16<br>STATE<br>RESET<br>STRTSTOP<br>PTINAL<br>ALHIFL<br>ALLOFL  | Accumulation active state<br>same as STATE<br>same as STATE<br>Alarm state<br>HI alarm<br>LO alarm  |
| Timer              | TM(xx).zz   | xx : timer number<br>zz : variable code                       | 01-32<br>STATE<br>RESET<br>STRTSTOP<br>TIMOUTFL  | Timer active state<br>same as STATE<br>same as STATE<br>Time-up alarm   |
| Flag               | FL(xxx)     | xxx : flag number   | 001-256  |   |
| Logic block        | LB(xxx)     | xxx : LB number   | 001-128  |   |
| Others             |             | Time Out Gate<br>Soft Failure<br>Partial Failure              | TOG<br>SFTFL<br>PRTFL  |   |

**Note:** The Boolean complement of any Input Parameter is specified by placing a minus sign (-) before the Input Parameter Code. For example, the code -LP(01).PTINAL, when true, means that Loop 01 is **not** in alarm.

Table 3-4 — Output Parameters

| POINT TYPE         | FORMAT     | PARAMETER                       | VALUE      | DESCRIPTION                    |
|--------------------|------------|---------------------------------|------------|--------------------------------|
| Modulating Control | LP(xx).zz  | xx : slot number                | : 01-16    |                                |
|                    |            | zz : variable code              | : SOA      | : DO 1 (LO) request            |
|                    |            |                                 | : SOB      | : DO 2 (HI) request            |
|                    |            |                                 | : OMAN     | : Manual mode request          |
|                    |            |                                 | : OAUTO    | : Auto mode request            |
|                    |            |                                 | : OCAS     | : Cascade mode request         |
|                    |            |                                 | : OCOM     | : Computer mode request        |
|                    |            |                                 | : PMAN     | : Program manual-mode request  |
|                    |            |                                 | : PAUTO    | : Program auto-mode request    |
|                    |            |                                 | : PCAS     | : Program cascade-mode request |
|                    | : PCOM     | : Program computer-mode request |            |                                |
| Digital Output     | DO(xxy).zz | xx : slot number                | : 01-16    |                                |
|                    |            | y : subslot number              | : 1-8      |                                |
|                    |            | zz : variable code              | : OP       | : Digital output request       |
| Counter Input      | CI(xxy).zz | xx : slot number                | : 01-16    |                                |
|                    |            | y : subslot number              | : 1-4      |                                |
|                    |            | zz : variable code              | : RESET    | : Start/reset request          |
|                    |            |                                 | : STRTSTOP | : Restart/stop request         |
| Analog Input       | AI(xx).zz  | xx : AI point number            | : 01-16    |                                |
|                    |            | zz : variable code              | : RESET    | : Start/reset request          |
|                    |            |                                 | : STRTSTOP | : Restart/stop request         |
| Timer              | TM(xx).zz  | xx : timer number               | : 01-32    |                                |
|                    |            | zz : variable code              | : RESET    | : Reset/stop request           |
|                    |            |                                 | : STRTSTOP | : Start/stop request           |
| Flag               | FL(xxx)    | xxx : flag number               | : 001-256  |                                |

**Note:** A continuous output is specified by placing an asterisk (\*) before the Output Parameter Code, and a momentary output is specified by not including the asterisk. For example, \*FL(001) specifies continuous output to Flag 001, while FL(001) specifies a momentary output.

### 3.2 LOGIC BLOCK COMMENTS ENTRY FORM

The Logic Block Comments Entry Form, shown on the following page, is also printed on an 8 1/2 by 11 inch sheet. The form is divided into two parts, in the following manner:

Lines 0.0 thru 0.3 Pathname—provides data entry areas for the system identifiers (device, volume, and filename) with which the set of logic blocks (up-to-128 for a given Multifunction Controller) is associated. Pathname data is not actually part of logic-block comments that are entered into the system; it is intended only to identify each Comments form with a particular Multifunction Controller and data-base file.

1.0 Comment Lines—in which user comments relating to logic blocks for a given Multifunction Controller are recorded. After logic-block configuration data has been entered into the system (through the Logic Block Configurator function), these comments can be entered into the system through the Text File Editor function. The comments are available to the user for reference

- on the forms on which they are entered.
- on the Text File Editor display.
- on text listings (printouts) obtained through the Text File Editor.

Comments are not executed in the system and, therefore, have no effect on system operation. Comment entries are optional; they can include any information that is meaningful to the user. Comments are very important when logic-block configuration must be correlated with system functions (such as programming) and/or must be modified as a result of revisions to process requirements. Clear and comprehensive comments are particularly important when logic-block modifications and correlation with other system functions must be done by personnel other than those who initially do logic block configuration.

Instructions for filling out the Logic Block Comments Entry Form are listed in Table 3-5.



Table 3-5 — Entries for Logic Block Comments Form

| LINE | NAME               | DATA ENTRY  | REMARKS  |
|------|--------------------|---|--|
|      |                    | <b>TEXT FILE EDITOR DISPLAY</b>   |  |
| 0.0  | <b>Pathname</b>    |   | Pathname entries are not required for logic-block configuration, but are recommended to facilitate data entry.   |
| 0.1  | Device             | NET (History Module) or \$Fn (floppy drive n)   |  |
| 0.2  | Volume             | XXXX (user-defined in Network configuration)  |  |
| 0.3  | Filename           | LBxxyyzz, where<br>xx = Hiway Number (01-20).<br>yy = Box Number (01-31).<br><br>zz = Serial Number (00-99) | Configured in Network. Box Number of Multifunction Controller. User number for revision level of this filename.  |
| 1.0  | <b>Comments</b>    |   |  |
|      | No. of Logic Block | Enter the number of the logic block to which the comment applies.   | Comments can apply to logic blocks before or after the comment, or to any group of logic blocks. This number is only a reference for locating comments in a Text Editor listing. |
|      |                    | For each comment line, enter any character string (alphanumerics, symbols, and spaces).                     | The first two characters of each comment line, when entered into the system via the Text File Editor function must be hyphens, as preprinted on the comments form.               |



# READER COMMENTS

Honeywell IAC Automation College welcomes your comments and suggestions to improve future editions of this and other publications.

You can communicate your thoughts to us by fax, mail, or toll-free telephone call. We would like to acknowledge your comments; please include your complete name and address

**BY FAX:** Use this form; and fax to us at (602) 313-4108

**BY TELEPHONE:** In the U.S.A. use our toll-free number 1\*800-822-7673 (available in the 48 contiguous states except Arizona; in Arizona dial 1-602-313-5558).

**BY MAIL:** Use this form; detach, fold, tape closed, and mail to us.

Title of Publication: **Logic Block Form Instructions** Issue Date: **9/95**

Publication Number: **MC12-500**

Writer: **J. Danner**

**COMMENTS:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**RECOMMENDATIONS:** \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

NAME \_\_\_\_\_ DATE \_\_\_\_\_  
TITLE \_\_\_\_\_  
COMPANY \_\_\_\_\_  
ADDRESS \_\_\_\_\_  
CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_  
TELEPHONE \_\_\_\_\_ FAX \_\_\_\_\_

(If returning by mail, please tape closed; Postal regulations prohibit use of staples.)

Communications concerning technical publications should be directed to:

Automation College  
Industrial Automation and Control  
Honeywell Inc.  
2820 West Kelton Lane  
Phoenix, Arizona 85023-3028

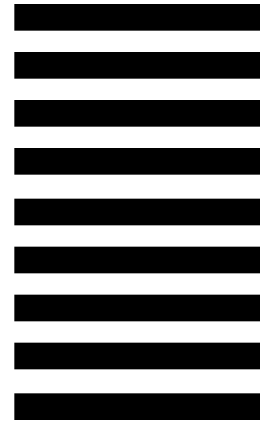
FOLD

FOLD

From: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_



NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE USA



Cut Along Line

**BUSINESS REPLY MAIL**  
FIRST CLASS      PERMIT NO. 4332      PHOENIX, ARIZONA

POSTAGE WILL BE PAID BY ....

**Honeywell**

Industrial Automation and Control  
2820 West Kelton Lane  
Phoenix, Arizona 85023-3028

Attention: Manager, Quality

FOLD

FOLD

Additional Comments:



**Honeywell**

---

**Industrial Automation and Control**  
Honeywell Inc.  
16404 North Black Canyon Highway  
Phoenix, Arizona 85023-3033

*Helping You Control Your World*