



# ***GE Fanuc Automation***

---

*Programmable Control Products*

## ***Series One™ PLC to Series 90™-30 PLC Translator***

*User's Manual*

GFK-0486

January, 1990

# WARNINGS, CAUTIONS, AND NOTES AS USED IN THIS PUBLICATION

## WARNING

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

## CAUTION

Caution notices are used where equipment might be damaged if care is not taken.

## NOTE

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware and software, nor to provide for every possible contingency in connection with installation, operation, and maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

1. Introduction	1
2. Unpack/Install	1
2.1. Packing List	1
2.2. Minimum Requirements	1
2.3. Physical Description	1
2.4. Pre-Installation Setup/Checkout	1
2.5. Installation	2
2.5.1. Floppy Disk Users	2
2.5.2. Hard Disk Users	2
2.6. Power Up/Verification	2
3. Setup	2
4. Operation	3
4.1. Main Menu	4
4.1.1. Config	4
4.1.1.1. Specifying the Series One Configuration	5
4.1.2. Program	8
4.1.2.1. Math Functions	8
4.1.2.2. Results	8
4.1.3. Exit	8
5. Integration	9
6. System Considerations	9
6.1. Memory Usage	10
6.1.1. %M Memory	10
6.1.2. %T Memory	11
6.1.3. %R Memory	12
6.2. Algorithms Used	12
6.2.1. Timers and Counters	12
6.2.2. Sequencers	12
6.2.3. Shift Register	12
6.2.4. Math Functions	13
6.3. Programmer Intervention	14
SUBTRACT FUNCTION	16
ADD FUNCTION	17
DIVIDE FUNCTION	18
MULTIPLY FUNCTION	19
<b>APPENDIX A. Sample Series One Original System</b>	<b>A-1</b>
<b>APPENDIX B. Series 90-30 System</b>	<b>B-1</b>

---

---

GFK-0486

<b>Figure</b>	1	Copyright and Disclaimer	3
	2	Main Menu	4
	3	CPU Selection Screen	5
	4	Initial Rack Display	6
	5	Discrete Input (DIN) Menu	6
	6	Completely Specified Rack	7
	7	%M Memory Usage by the Translator	10
	8	%T Memory Usage by the Translator	11
	9	%R Memory Usage by the Translator	12

## **1. Introduction**

In order to minimize the effort and expertise required to move Series One PLC applications to Series 90-30, GE Fanuc has developed this translator program.

Starting with a Logicmaster 1 print file of the original Series One program and the I/O module content of each slot in the Series One system, the translator will determine the required 90-30 hardware, and will convert the Series One program into a form usable by the Logicmaster 90-30 programming package.

Logic translation should be nearly 100%, but Series One Plus data operation (MATH) instructions will require some manual intervention. To the maximum extent possible, the areas requiring attention are flagged by special coils. The translator emulates the operation of the Series One Plus for the data instructions, but programming efficiencies can generally be realized by changing the approach to the application to take advantage of the instruction set of the 90-30 PLC.

This translator should be viewed as a tool which facilitates the conversion of the application, but the final responsibility for system operation must be accepted by the user.

## **2. Unpack/Install**

### **2.1. Packing List**

The translator package consists of this manual and a single 3 1/2 inch floppy disk containing the following files:

30XLAT.EXE  
MODULES.DBA  
EQUIV.DBA  
KEYS.DBA  
SAMPLE.TXT

### **2.2. Minimum Requirements**

- Workmaster II, Workmaster I, or other 100% IBM compatible.
- 256K RAM
- One floppy disk drive (3.5")
- DOS version 2.0 or greater
- The Distribution Diskette
- Print file of the original application from Logicmaster 1
- Layout of the Series One system, ie. I/O modules by slot
- Logicmaster 90-30 (Operation requires a Hard Disk)

### **2.3. Physical Description**

The translator is a software program and requires no physical description other than the PACKING LIST above.

### **2.4. Pre-Installation Setup/Checkout**

There are no pre-installation requirements.

## 2.5. Installation

### 2.5.1. Floppy Disk Users

The translator program may be executed from floppy disk. As always, you should make a working copy of the original by using the DOS COPY command.

EXAMPLE: With the original diskette in drive a: and a formatted diskette in drive b:, type

```
COPY A:*. * B:
```

to make your working copy .

### 2.5.2. Hard Disk Users

Substantial performance improvement will be obtained by installing the program on a hard disk. First create a directory (suggested name TRANSLAT) on the hard disk, and then copy all the files from the distribution disk into that directory.

EXAMPLE: With the original diskette in drive a: and a hard disk designated c: type the following:

```
MKDIR C:\TRANSLAT
```

```
COPY A:*. * C:\TRANSLAT
```

## 2.6. Power Up/Verification

In order to verify that the program files have been properly installed, default to the drive and directory which contains the files from the distribution diskette

EXAMPLES:

```
A: - for execution from a floppy disk
```

```
C:  
CD\TRANSLAT - for execution from the hard disk
```

then type

```
30XLAT
```

When the copyright and disclaimer screen (shown below) appears, you have properly completed the installation. If the screen does not appear, verify that all the files from the distribution diskette have been copied into the operating directory.

## 3. Setup

The translator has no specific setup requirements at this time.

## 4. Operation

In order to convert a Series One application to the 90-30 PLC, you must first copy the ASCII print file from Logicmaster 1 into the same directory as the files from the distribution diskette. (NOTE: Programs originally created using the Series One hand held programmer must first be imported into Logicmaster 1, conflicts resolved, and the ASCII print file created). In addition, you must know the Series One hardware configuration, ie. the contents of each slot.

From the Copyright and Disclaimer screen, press any key to continue to the MAIN MENU.

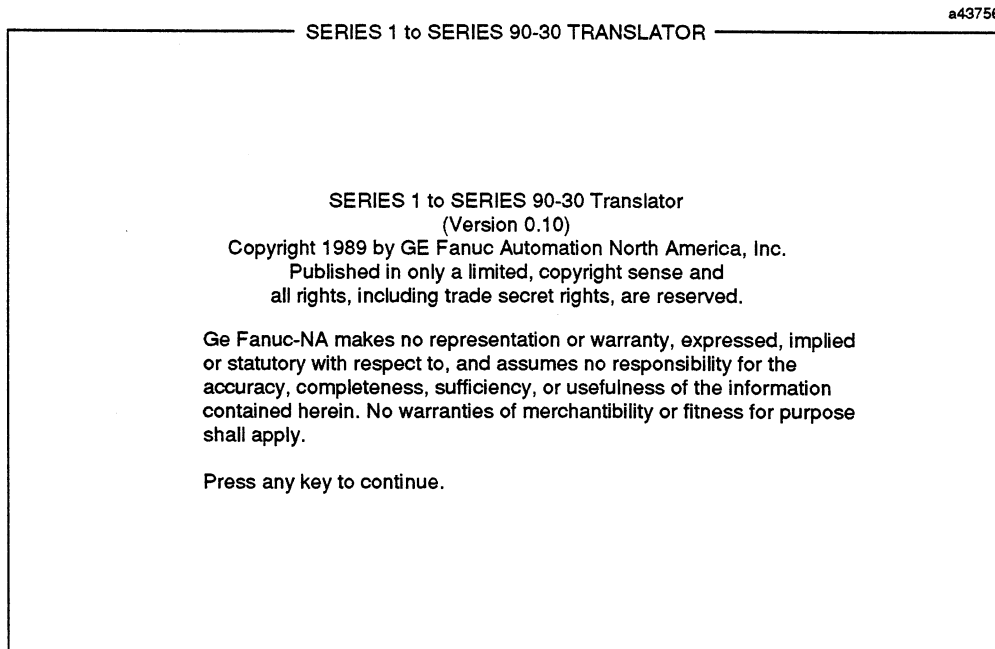
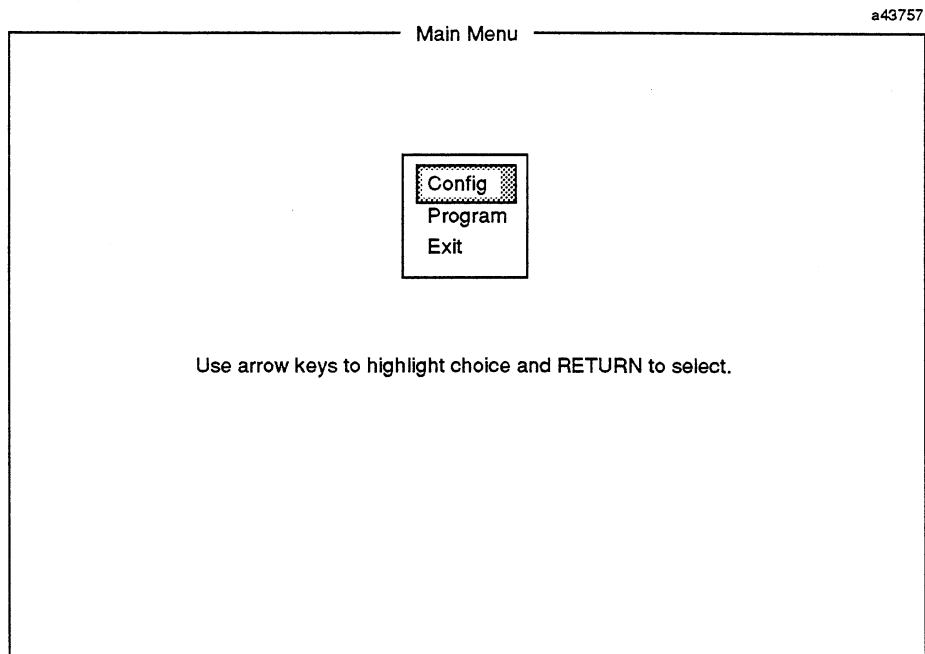


Figure 1. Copyright and Disclaimer

## 4.1. Main Menu



**Figure 2. Main Menu**

The MAIN MENU screen is shown above in Figure 2. Operation of each selection will be discussed in detail below. The SAMPLE file on the distribution diskette and the matching configuration in Appendix I will be used to illustrate the operation. *Tutorial instructions are shown in italics.*

### 4.1.1. Config

This selection is used to specify the hardware configuration of the Series One system to be converted. You will be prompted to specify a filename for the application. If the Series One print file is already on disk, you should specify that filename, without extension. (Print files from Logicmaster 1 have a .txt extension).

*Press ENTER to select CONFIG and specify the filename SAMPLE.*



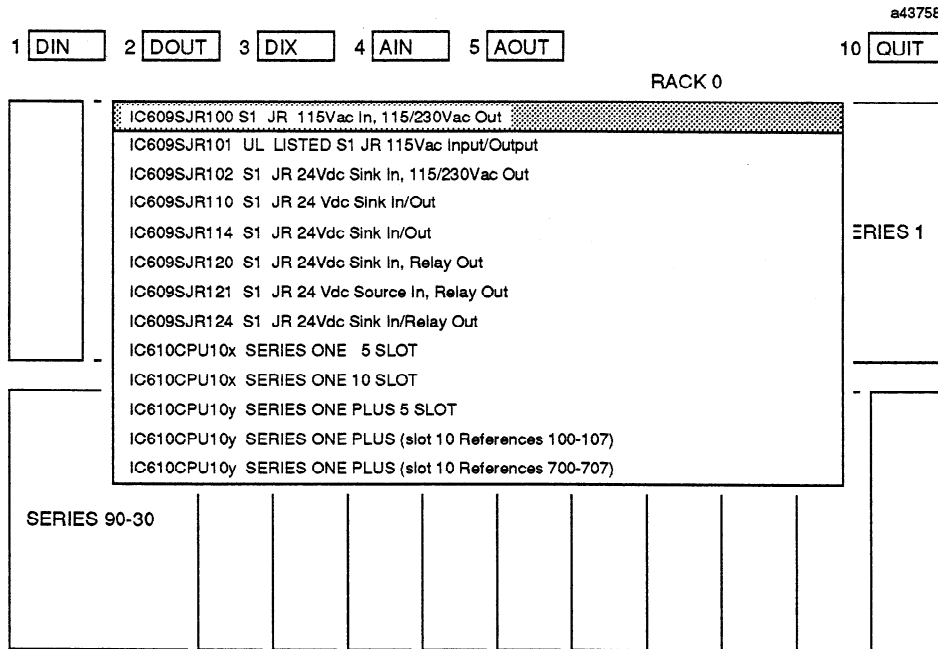


Figure 3. CPU Selection Screen

#### 4.1.1.1. Specifying the Series One Configuration

Following selection of CONFIG from the main menu and specification of a filename, the CPU Selection Screen (Figure 3) is displayed. A selection is made from the menu of possible Series One CPU's by using the CURSOR UP and CURSOR DOWN keys to highlight the appropriate item and pressing ENTER. Once a selection has been made, a graphic representation of the Series One rack is displayed on top of the screen, with a 90-30 rack displayed on bottom. (Note: if one of the Series One Junior selections was made, several slots of the 90-30 will already have been filled in).

*Referring to the SAMPLE configuration shown in Appendix A, notice that it uses the 5 slot Series One CPU. Use the CURSOR DOWN key to highlight that selection and press ENTER.*

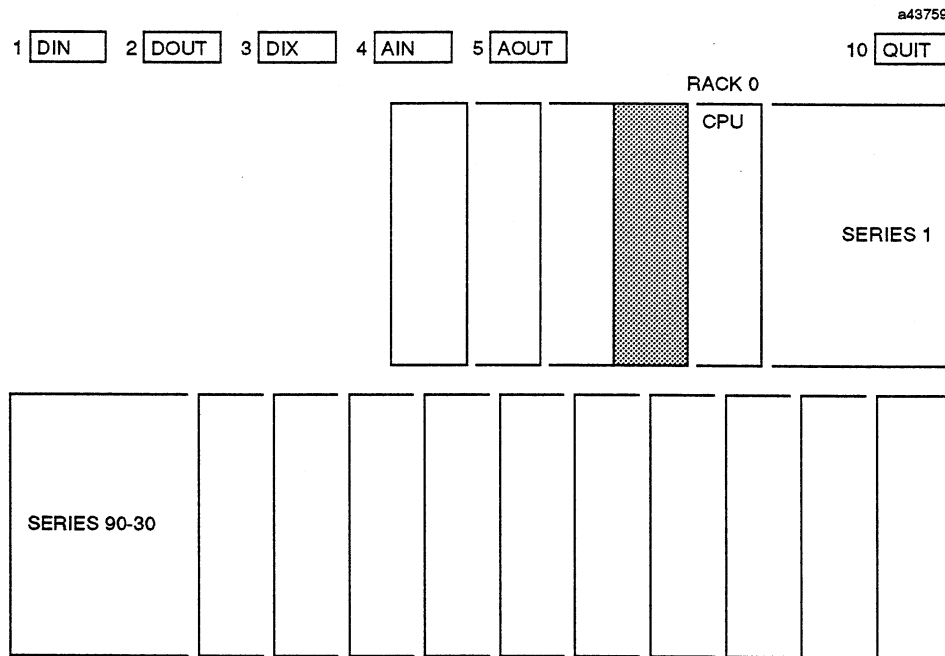


Figure 4. Initial Rack Display

If a Series One rack type CPU has been selected, one of the slots in the rack is highlighted (see INITIAL RACK DISPLAY FIGURE 4). The slot cursor may be moved with the CURSOR RIGHT and CURSOR LEFT keys. The function key mapping shown at the top of the screen corresponds to discrete in (DIN), discrete out (DOUT), mixed (MIXED), analog in (AIN), or analog out (AOUT).

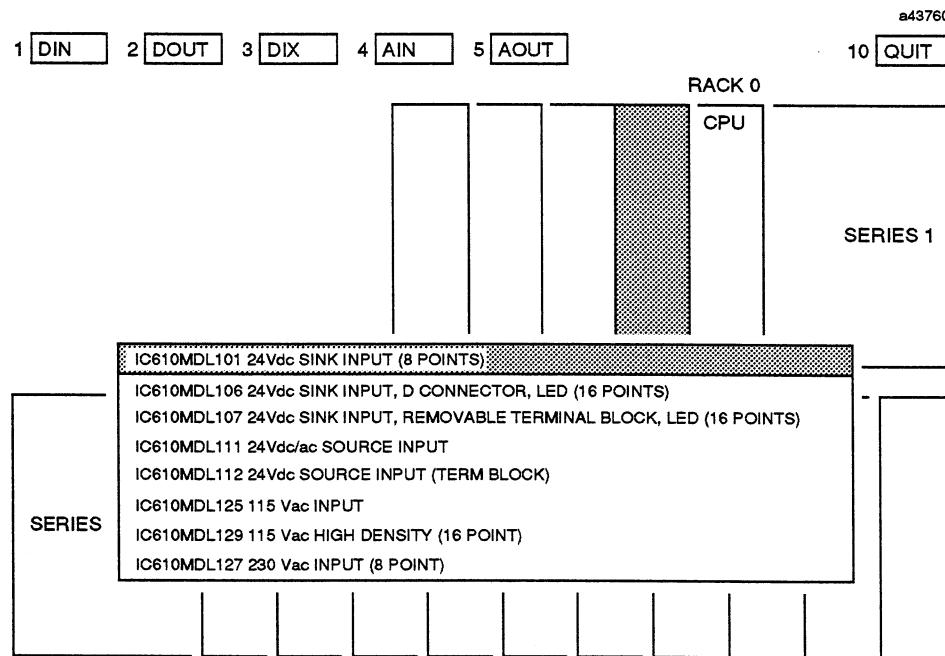


Figure 5. Discrete Input (DIN) Menu

GFK-0486

Each Series One slot is specified by moving the highlight to that slot, pressing the appropriate function key for the type module, and selecting from among the modules displayed in the subsequent menu. (Figure 5 shows the DIN menu). The corresponding 90-30 module is selected by the program and placed in the 90-30 rack, and addresses are mapped between the systems. The Series One slots may be specified in any order, but the 90-30 rack will always fill from left to right.

Continuing to follow the SAMPLE configuration in Appendix A, the first I/O slot (slot 2 in the CPU rack) in the Series One was a discrete input (Press F1 - DIN). It was further an 8 point SINK INPUT which happens to be the first menu item. Press ENTER to select it. If it had been another item, the CURSOR UP and CURSOR DOWN keys would be used to highlight the selection before pressing ENTER. Notice what happens: 1) the module type and octal addresses are shown in the Series One rack; 2) the corresponding module is placed in the Series 90-30 rack; 3) its starting address is shown as %I1; 4) the starting and ending octal addresses which are mapped from the Series One are displayed in the Series 90-30 slot. Fill in the rest of the first rack using the SAMPLE configuration.

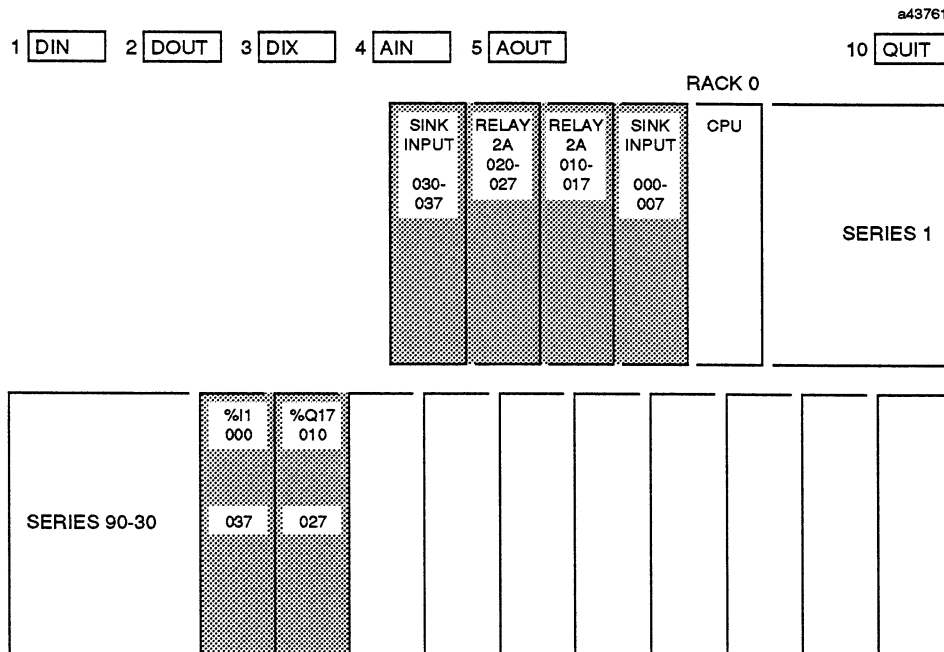


Figure 6. Completely Specified Rack

**NOTE**

(If sufficient spare points of the appropriate type are available from a previous selection, those points will be used rather than installing another module in the 90-30, ie. point usage is optimized).

This process continues until the entire content of the original Series One rack has been specified. See Figure 6 for a completely specified rack.

If the original system has expansion rack(s), the CURSOR DOWN key is used to begin specification of the additional racks. CURSOR UP is used to return to the preceding rack.

Our SAMPLE system does include a 5 slot expansion rack. Use CURSOR DOWN, select the 5 slot expansion rack, and fill it with the modules as shown in the SAMPLE configuration.

Once the entire contents of the original Series One system has been specified, F10 should be pressed. Data will be stored in a CONFIGURATION file (ie. FILENAME.CFG) and will be used during the program translation process.

#### 4.1.2. Program

This selection is used to translate the Logicmaster 1 print file of the original program, which must by now be on the same disk drive and directory as the files from the distribution diskette. You will be prompted for the name of this file. As before, the filename must be specified without extension. If ENTER is pressed without entering a filename, a menu of all the print files (ie. those with a .txt extension) will be displayed. Selection is made by moving the cursor to the desired program file and pressing ENTER. If a CONFIGURATION file of the same name exists, translation of the program will begin immediately. Otherwise, you will be prompted for the name of the CONFIGURATION file to be used.

##### 4.1.2.1. Math Functions

If your original program contains Series One Plus data operations or math functions, you will be asked whether these should be translated. If you answer 'Y', the translator will convert the math operations by emulating Series One Plus, and will flag areas requiring attention (See section titled Operator Intervention). If you answer 'N', the ladder logic driving the math function will be converted and a flag coil MATH\_FUN will mark the first math function. The remaining math functions and any parallel coils in that rung will be skipped. The latter will require more effort on your part, but will allow you to take advantage of the 90-30 instruction set.

##### 4.1.2.2. Results

Two files are created in the translator directory. They both use the same filename as the Logicmaster 1 print file, but with different file extensions. The first, with a .CFG extension, is used with the 90-30 PLC Configuration Expert Program to create a customer quotation. Only GE Fanuc and our authorized distributors will have this program. The second, with a .MAP extension is an ASCII file containing the correspondence between the original Series One octal addresses, the new 90-30 references, and any NICKNAMES associated with them. You will probably find it useful to use the DOS PRINT command to produce a hardcopy of this file.

In addition, a 90-30 program folder (directory) will be created under the translator directory with the same name as the Logicmaster 1 print file, and three files will be created in that directory. The files are LMFOLDER.30, KEY1.DEF, and KEY2.DEF. These are a marker file, the PROGRAM teach-in file, and the CONFIGURATION teach-in file respectively. The folder and its files will be used by Logicmaster 90-30 to re-create the application.

*Continuing our tutorial, use the DOWN CURSOR key to highlight the PROGRAM selection and press ENTER. Specify the filename SAMPLE and press ENTER. Or, if you would like to see how the files menu works, press ENTER without specifying a filename then cursor to SAMPLE and press ENTER. If you are running on a Workmaster II, translation will take about 20 seconds. A Workmaster I will take over 1 minute.*

##### 4.1.2.3. Exit

This selection ends the translator program and returns to DOS.

## 5. Integration

The activities of the translator program are actually a front end to the Logicmaster 90-30 programming package. In order to continue, Logicmaster 90-30 must have been installed on the Workmaster or other computer and the folder (directory) with the three files created in the Results section above must be available. It is assumed from this point on that you are familiar with basic Logicmaster 90-30 operation.

The next step in the process of converting an application from Series One to Series 90-30 involves using the inherent capability of Logicmaster to read a "teach-in" file. The two .DEF files created by activities described above are used to automatically enter both the hardware configuration and logic into the Logicmaster 90-30 programmer.

First, start the Logicmaster 90-30 program and SELECT the program folder created by the translator. Both the CONFIGURATOR and the PROGRAMMER portions of the Logicmaster package will be used.

Once in the CONFIGURATOR, select I/O CONFIG and playback the "teach-in" file containing the configuration information by pressing ALT and the number "2". You should see the necessary I/O boards entered into the 90-30 rack together with their starting addresses. When the activity is finished, leave the CONFIGURATOR by pressing ZOOM OUT (ESC) twice, and replying 'Y' to the exit prompt.

*Select the tutorial folder SAMPLE and try the above. A SINK INPUT followed by 2 RELAY OUTPUTS, and another SINK INPUT board should be installed as in the Series 90-30 Sample Configuration figure below.*

Once in the PROGRAMMER, playback the "teach-in" file containing the logic information by pressing ALT and the number "1". The first screen activity you will observe is the teach-in file selecting FOLDER functions and disabling duplicate coil use checking. A beep will occur during this time. Next, you should see the ladder diagram and NICKNAMES being typed into the programmer. When the activity is finished, leave the PROGRAMMER by pressing ZOOM OUT (ESC), 2 times, and replying 'Y' to the exit prompt.

*Again using the SAMPLE folder try the above. You should see the program logic being entered into the folder. When it is complete, use the Logicmaster PRINT function to make a hard copy. You can use the DOS PRINT function to make a hard copy of the original program. Compare the two listings and become familiar with how the various elements are handled in the translation.*

### NOTE

Each Series One rung is identified in the resulting 90-30 program using a comment label. Since these consume 90-30 memory, you will probably want to delete the comments once your program is debugged.

At this point, assistance with the application translation is complete. There are probably items which MUST be resolved, and others which SHOULD be reviewed. These are explained in the section under PROGRAMMER INTERVENTION.

## 6. System Considerations

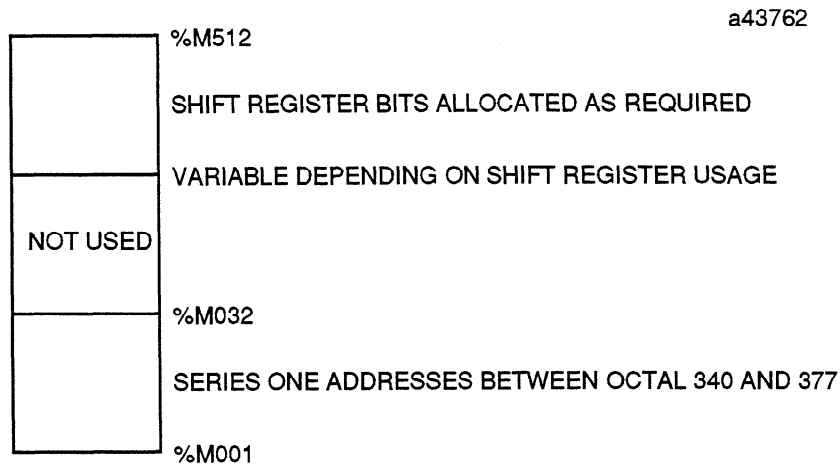
As configured by the translator, each Series 90-30 slot is addressed on a 16 bit boundary, ie. 1, 17, 33, etc., without respect to whether an input or output board will occupy the slot. Real Series One I/O points will be mapped automatically to their counterparts in the 90-30 system. Using the 90-30 CONFIGURATOR, the board locations may be changed within the rack, but their assigned starting addresses must not be changed.

## 6.1. Memory Usage

Memory assignments are made with the intention that the translated program might be run on the smallest CPU in the 90-30 family (the -311).

### 6.1.1. %M Memory

The %M memory map of the 90-30 system is shown in Figure 7. Shift register bits are allocated 16 bits at a time, working down from the 512 limit of this memory type. Series One references between 340 - 377 (retentive coils) are mapped up from %M001 to %M032.



**Figure 7. %M Memory Usage by the Translator**

6.1.2. %T Memory

The %T memory map of the 90-30 system is shown in Figure 8. Series One references 160 through 337 (non retentive coils) are mapped up from %T001 through %T112. Series One references from 000 to 159 which are used as internals are mapped down beginning at %T240 in the order in which they are encountered in the original program. The translator uses %T memory above 241 for math flags. See usage explanation under section titled Math Functions.

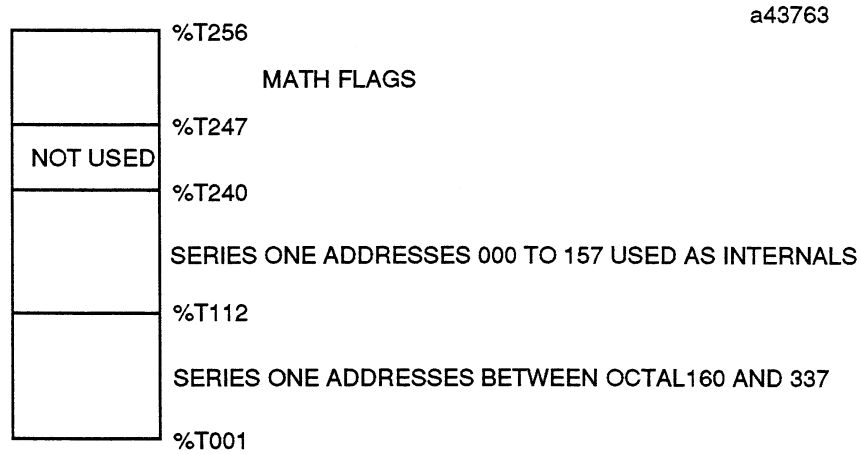


Figure 8. %T Memory Usage by the Translator

### 6.1.3. %R Memory

Registers are allocated by the program beginning with %R0001 and working up, in the order in which registers are encountered in the original Series One program. Registers 301 and 302 are reserved as the equivalent of the Series One ACCUMULATOR; 303 and 304 are the equivalent of the AUX ACCUMULATOR. Registers between 319 and 512 are used for TIMER AND COUNTER functions which require 3 registers each.

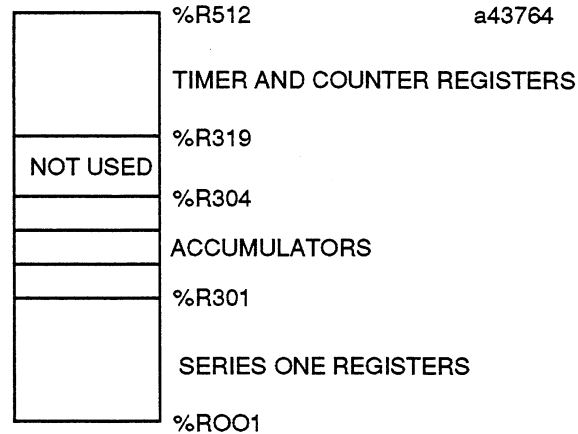


Figure 9. %R Memory Usage by the Translator

## 6.2. Algorithms Used

To the maximum extent possible, the translator attempts to emulate the operation of the Series One PLC. In the case of contacts and coils, the translation is straight forward. Other elements warrant some explanation.

### 6.2.1. Timers and Counters

Series 90-30 timers and counters do NOT have a coil associated with the element, but since the Series One does, each occurrence of a timer or counter is brought out to an internal coil. Use of contacts from the original Series One timer or counter will use the internal 90-30 coil.

### 6.2.2. Sequencers

Counters in the Series One PLC are frequently used together with the STEP function to effect a sequencer. The translator will implement the counter in the normal way. The STEP function will be replaced with a separate rung which includes a compare function brought out to an internal coil. Contacts from the internal coil will replace the STEP references throughout the program.

### 6.2.3. Shift Register

This is one of the more difficult program elements to translate from Series One to Series 90-30. Series One shift registers can start with any bit in the 400 - 477 series; can manipulate a group of any length; and operates directly on the bits in question. Series 90-30 must start with a bit on a word boundary; must manipulate words, ie. 16 bits or multiple thereof; does NOT operate directly on the bits in question, but takes word(s) as input, performs the shift on an internal image, ie. leaving the input unchanged, and



GFK-0486

places the result at the specified output reference. The translator will handle all these complexities, but an example will aid in understanding.

If a Series One shift register of 7 bits, using a START bit 400 and a STOP bit 406 were translated, the logic goes as follows:

1. Allocate 16 bits (since less than 16 are used) from the top of %M memory, ie. bits 497 - 512.
2. Select a SHL\_WORD instruction since the Series One bits are in natural order, ie. not reversed.
3. Right justify by mapping bits in order

Series One	Series 90-30
400	= 497
401	= 498
402	= 499
403	= 500
404	= 501
405	= 502
406	= 503

Bits 504 through 512 in the 90-30 will be wasted or unused in this case.

```

<----- SHIFT LEFT
*****
*512      unused          *503 * 502 *501 *500 *499 *498 *497*
*****
    
```

If the Series One bits had been reversed, ie. START 406 and STOP 400, step 1 above remains the same ; a SHR\_WORD instruction would have been selected in step 2 ; bits would be left justified in step 3 according to the following:

406	= 512
405	= 511
404	= 510
403	= 509
402	= 508
401	= 507
400	= 506

Bits 497 through 505 in the 90-30 will be wasted or unused in this case.

```

SHIFT RIGHT ---->
*****
*512 *511 *510 *509 *508 *507 *506 *      unused          497*
*****
    
```

In both cases, the IN and OUT for the selected 90-30 function will be specified as %M497.

### 6.2.4. Math Functions

Series 90-30 registers 301 and 302 are designated to act as the Series One accumulator; registers 303 and 304 act as the aux accumulator. All math functions will act on these registers. Bits are assigned as equivalents to the Series One Plus flags according to the following table:

## Special Math Function Coil

%T248	M_START	- EACH MATH RUNG USES THIS COIL
-------	---------	---------------------------------

## Series One Equivalent Math Flags

90-30 REFERENCE	FLAG INDICATION	SERIES ONE REFERENCE
%T242	GREATER THAN	772
%T243	EQUAL	773
%T244	LESS THAN	774
%T245	CARRY	775
%T246	ZERO	776
%T247	OVERFLO	777

Since the Series One Plus BCD math accumulator was limited to 4 BCD digits, the register equivalent in the 90-30 is corrected if 9999 is exceeded, and the OVERFLOW flag is set.

The Series One Plus compare function is replaced by three separate compares which set the appropriate flags above.

### 6.3. Programmer Intervention

Due to the substantial differences in the math functions between the two PLC's, manual intervention will probably be required in this area. Flag usage is shown in the following table:

#### Untranslated Function Flags

%T256	XLATERR	- INDICATES A TRANSLATION ERROR
%T255	MATH_FUN	- INDICATES A MATH FUNCTION
%T254	DECODE	- INDICATES A DECODE FUNCTION
%T253	ENCODE	- INDICATES A ENCODE FUNCTION
%T252	BCD_BIN	- INDICATES A BCD TO BINARY FUNCTION
%T251	BIN_BCD	- INDICATES A BINARY TO BCD FUNCTION
%T250	D_OUT	- INDICATES A SERIES ONE D_OUT FUNCTION
%T249	D_STORE	- INDICATES A SERIES ONE D_STR FUNCTION

Using the SEARCH capability of Logicmaster 90-30, each of the above flags must be located and appropriate logic substituted to duplicate the necessary function from the corresponding area of the Series One program.

XLATERR probably indicates use of a FUN20 or High Speed Counter. MATH\_FUN only appears if you are NOT translating the math functions.

The translator treats all moves into and out of the accumulator as 16-bit moves, but then flags the instruction by adding a coil with nickname D\_STR or D\_OUT. If your program used any of the 4 or 8 bit Series One instructions, you must perform the masking and shifting operations.

The translator performs no conversions, ie. BCD\_BIN and BIN\_BCD instructions are merely flagged. Realizing that all math in the Series 90-30 is performed in binary (as opposed to the Series One Plus BCD math), you must add the appropriate data conversion instructions for your application.

Likewise, the ENCODE and DECODE instructions from the Series One Plus are merely flagged. The program user must determine the appropriate action to take at these points in the program.

As an aid to your further understanding, Series One Plus math functions and their corresponding 90-30 implementations as performed by the translator are included in the Appendix. You will note that one of two approaches is used depending on flag usage in your Series One program. When flags are used, the algorithm emulates all of the flags associated with that function in the Series One Plus.

SUBTRACT FUNCTION

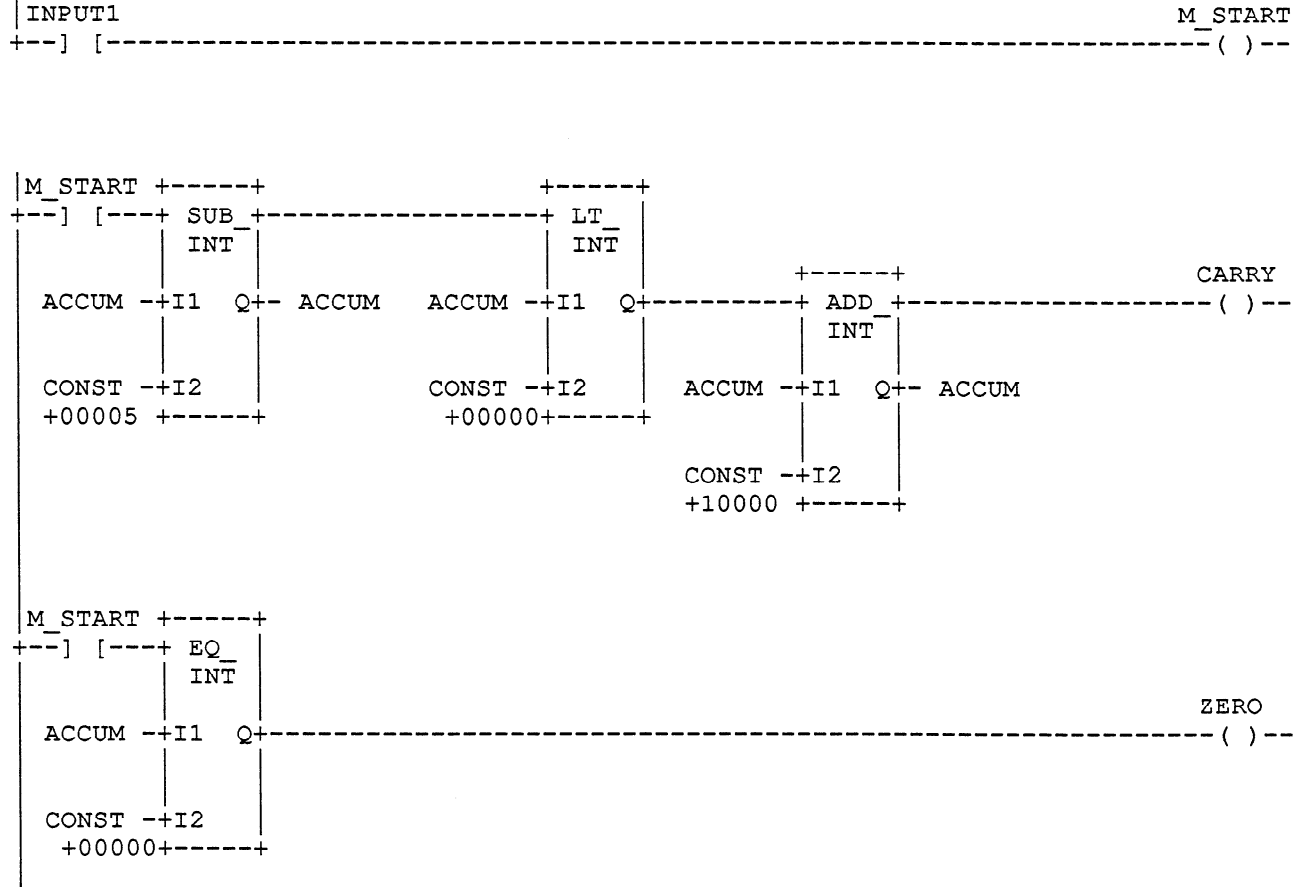
SERIES ONE PLUS RUNG



SERIES 90-30 EQUIVALENT  
WITH NO CARRY OR ZERO FLAG USAGE



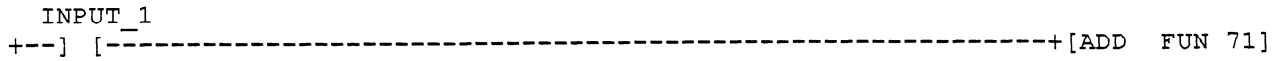
SERIES 90-30 EQUIVALENT  
WITH CARRY OR ZERO FLAG USAGE



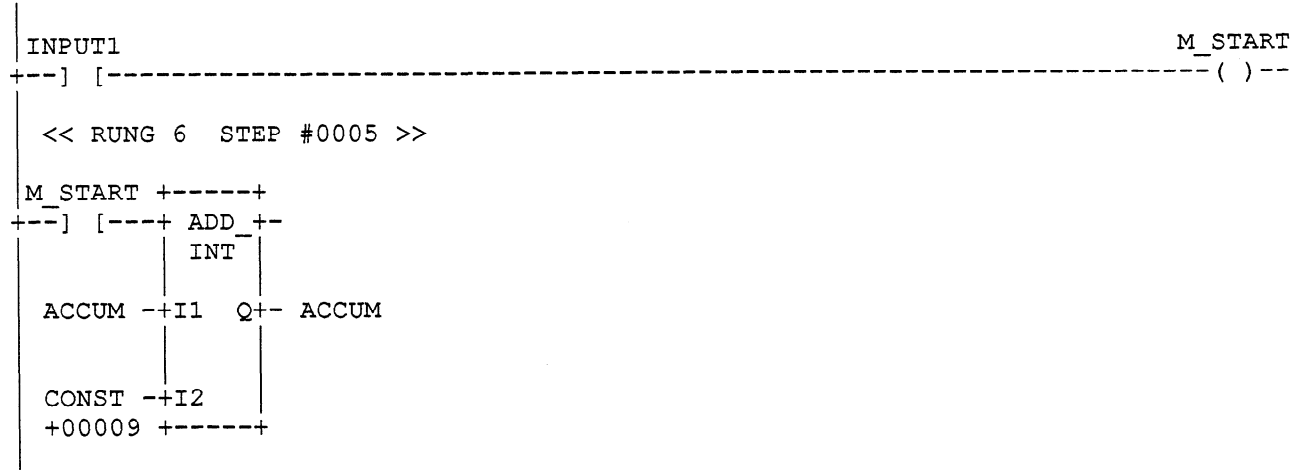
GFK-0486

**ADD FUNCTION**

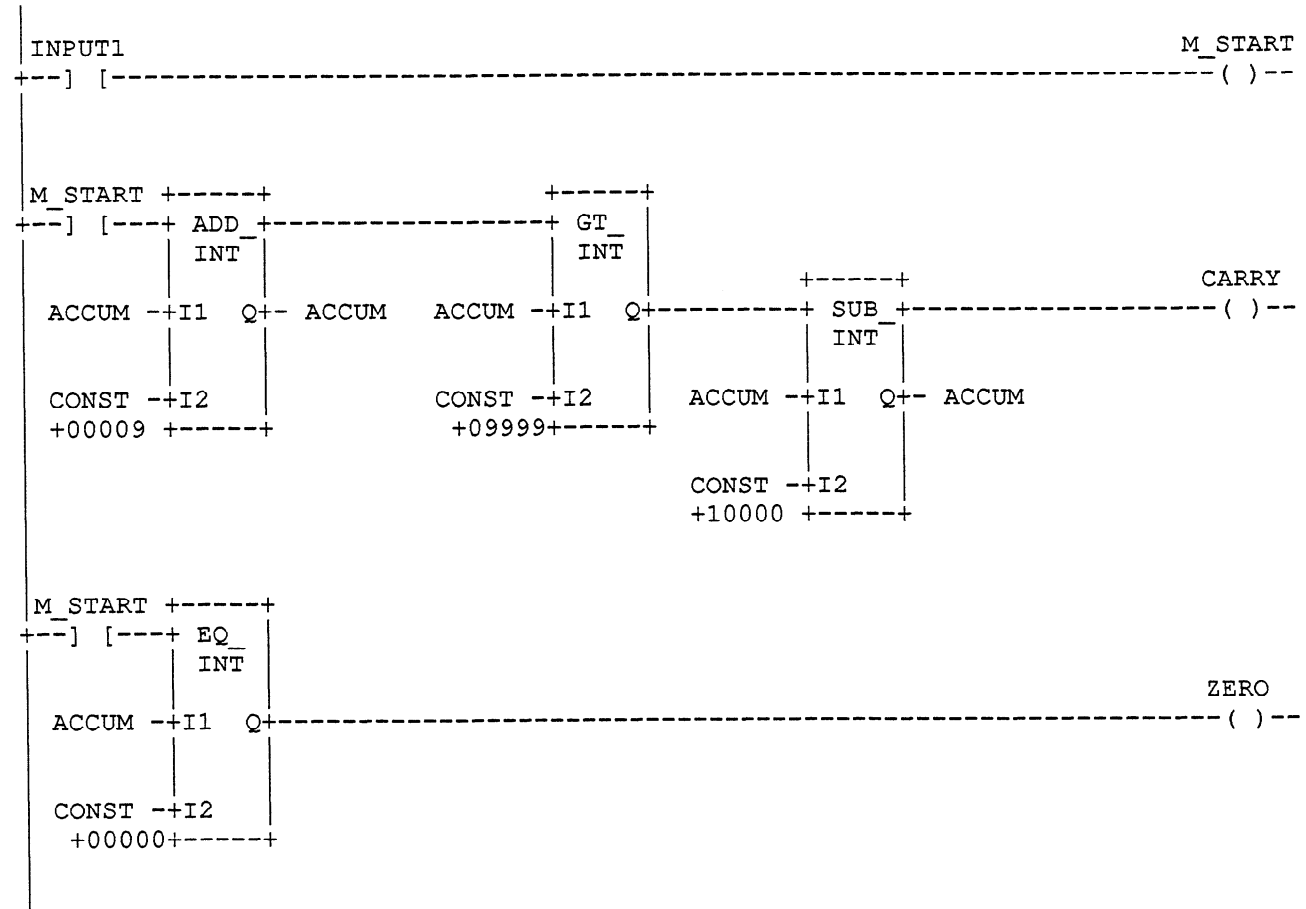
SERIES ONE PLUS RUNG



SERIES 90-30 EQUIVALENT  
WITH NO CARRY OR ZERO FLAG USAGE



SERIES 90-30 EQUIVALENT  
WITH CARRY OR ZERO FLAG USAGE

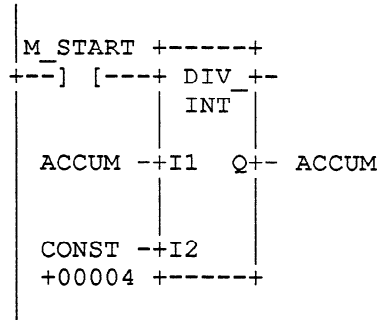


**DIVIDE FUNCTION**

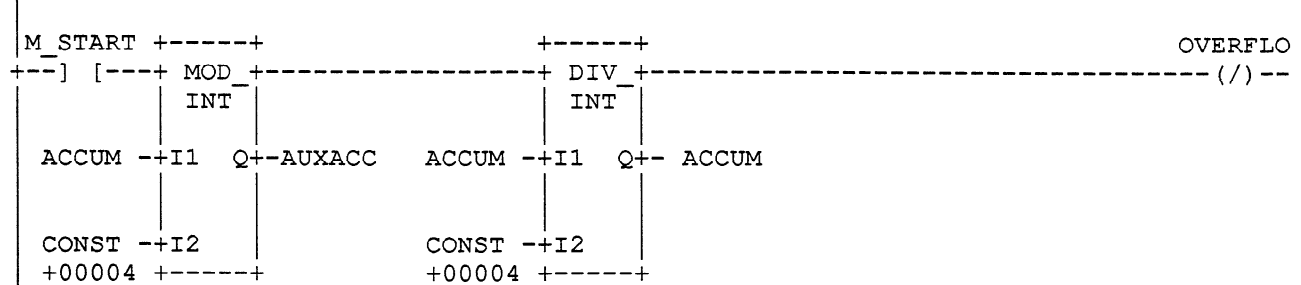
SERIES ONE PLUS RUNG



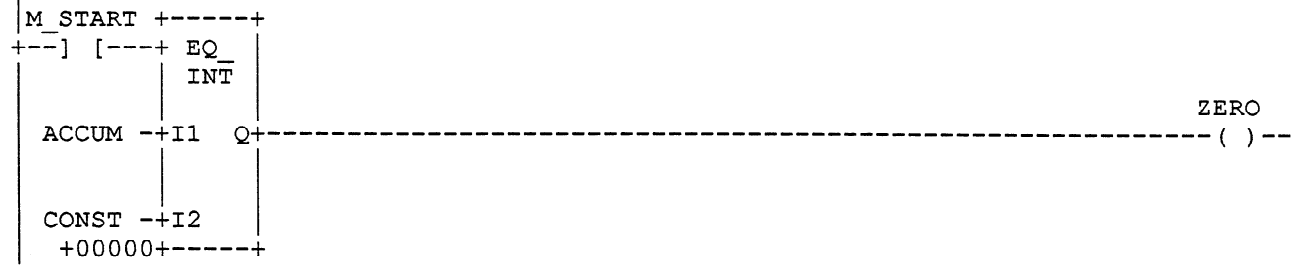
SERIES 90-30 EQUIVALENT  
WITH NO ZERO OR OVERFLOW FLAG USAGE



SERIES 90-30 EQUIVALENT  
WITH ZERO OR OVERFLOW FLAG USAGE



OVERFLO  
( / )--



ZERO  
( )--

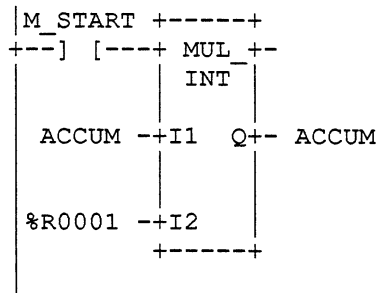
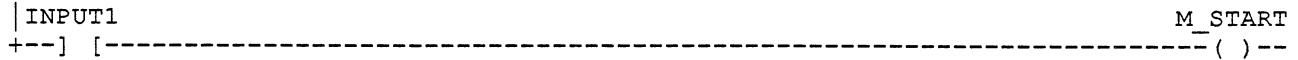
GFK-0486

MULTIPLY FUNCTION

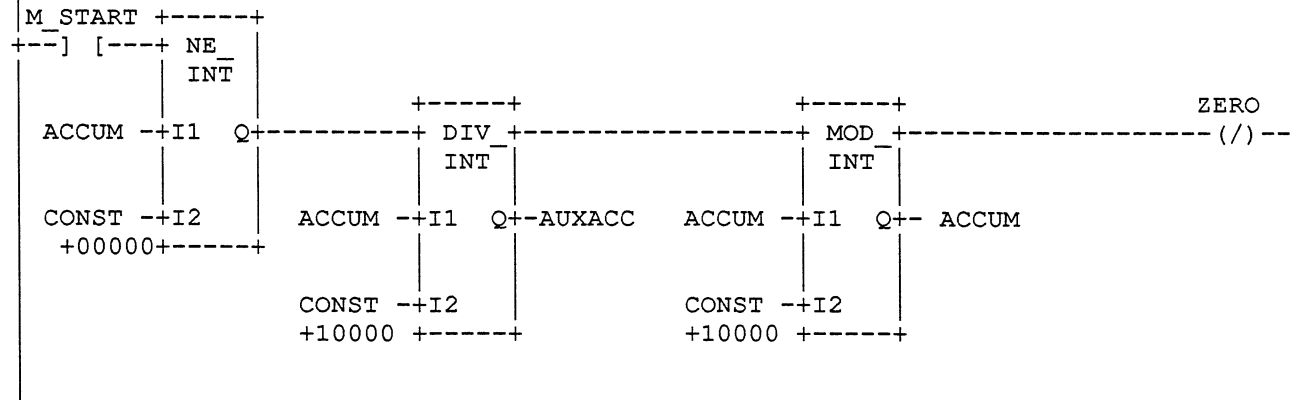
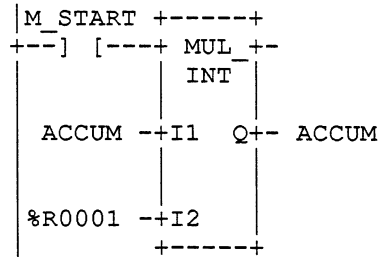
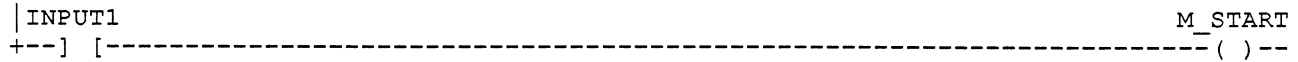
SERIES ONE PLUS RUNG



SERIES 90-30 EQUIVALENT  
WITH NO ZERO FLAG USAGE



SERIES 90-30 EQUIVALENT  
WITH ZERO FLAG USAGE







## Appendix A Sample Series One Original System

