Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

Product Family:	DL305	**Date**: 07.28.2021
Manual Number	D3-USER-M	
Revision and Date	1st Edition, Rev. D; January 2010	

07.28.2021

D3-HP is discontinued as of 06/2021. D3-HPP was discontinued 01/2018. Please consider the Productivity, BRX, or CLICK Series of PLC systems as a replacement.

07.16.2021

Changes to Chapter 5 & 6.

D3-16ND2-1 Discrete Input module has been discontinued as of 07/2021: Please consider the Productivity, BRX or CLICK Series of PLC systems as a replacement.

Errata Sheet

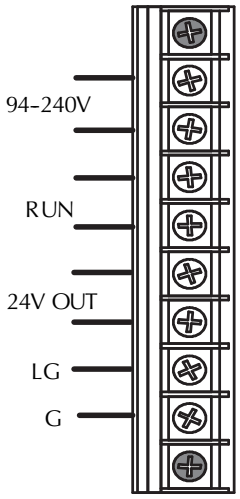
This Errata Sheet contains corrections or changes made after the publication of this manual.

08.2018

Changes to Chapter 2. Installation and Safety Guidelines

Page 2-12. Base Wiring

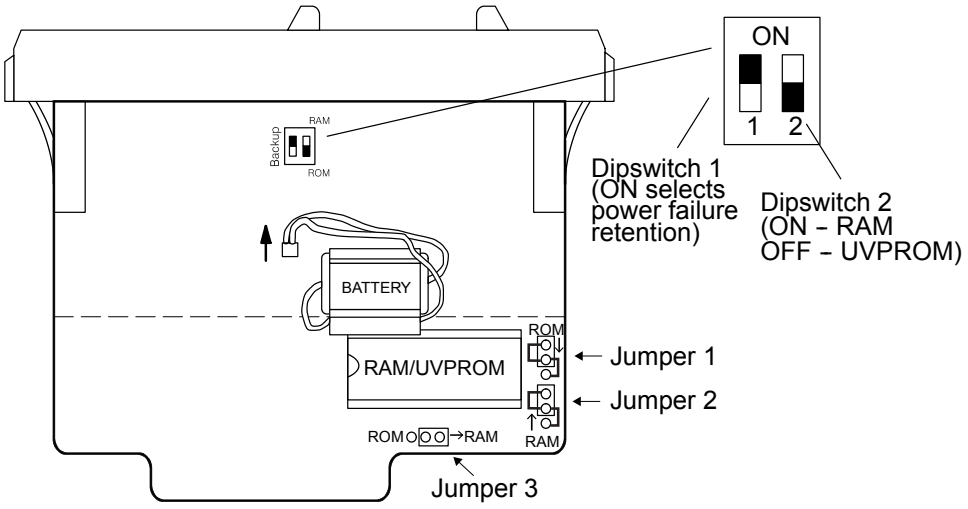
On newer 110-240 VAC bases, the terminal strip has been re-designed and does not have different terminals for 110VAC and 240VAC connections. If you have one of these re-designed bases the terminal strip will look like the graphic shown here.



Changes to Chapter 3. DL330/DL330P/DL340 CPU Specifications

Page 3-9. DL330/DL330P CPU Setup; Installing the UVPROM Option in the DL330/DL330P CPU.

The drawing showing the jumpers is missing the silkscreen jumper lines and arrows for jumpers 1, 2, and 3. Refer to this drawing.



Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

05.2018

Changes to Chapter 3. DL330/DL330P/DL340 CPU Specifications

Page 3-14. Battery Backup:

Please revise the first WARNING on this page (right above the battery replacement drawings) as follows:

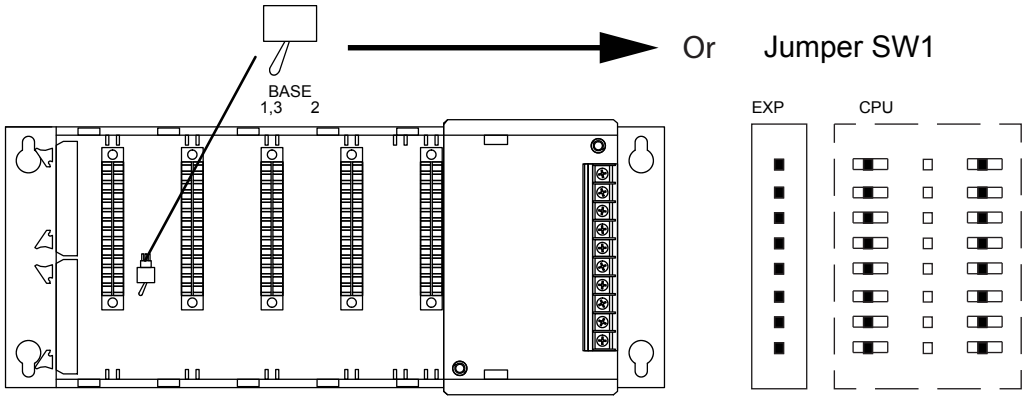
WARNING: If the battery is not installed or connected to the PC board, the 330 CPU will NOT notify you of the error. Be sure the battery is in place and the connector is firmly seated before you install the CPU into the base.

Changes to Chapter 4. Bases, Expansion Bases, and I/O Configuration

Page 4-16. Setting the Base Switches; 5 Slot Bases:

Replace the drawing of the 5 slot base with this one. Newer 5 slot bases have jumper switch SW1 instead of the toggle switch to set whether the base is a local CPU base or an expansion base.

NOTE: Older bases have a toggle switch to set the base as the CPU local base, the first expansion base, or the second (last) espansion base. Newer bases have the jumper SW1 in place of the switch.



Changes to Chapter 5. I/O Module Selection & Wiring Guidelines

Page 5-12. Fuse Protection

Replace the WARNING on this page with the following one:

WARNING: Modules which have soldered-in fuses or non-replaceable fuses are non-repairable and should be replaced with new modules.

Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

05.2018, cont'd

Changes to Chapter 11. Instruction Set; Timer, Counter, and Shift Register Instructions

Page 11-22. Timer (TMR) DL330/DL340 Only

Page 11-23. Counter (CNT) DL330/DL340 Only

Add this NOTE to both of these pages:

NOTE: *The counters and timers both time in Decimal and not in BCD. Presets for both are also interpreted as decimal data and not as BCD.*

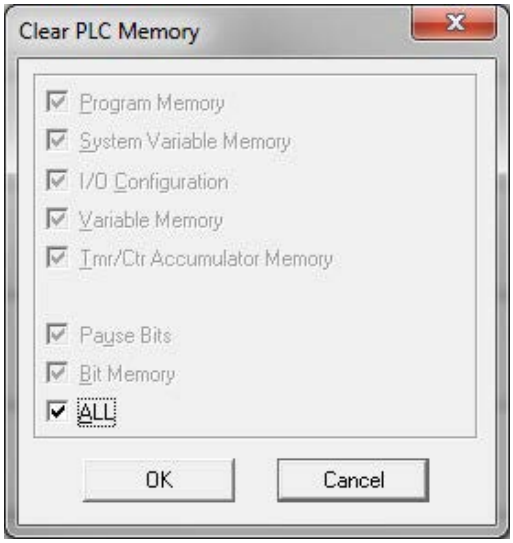
Changes to Chapter 13. Maintenance and Troubleshooting

Page 13-13. Add the following to the end of this chapter (right after END Instruction Placement):

Reset the PLC to Factory Defaults

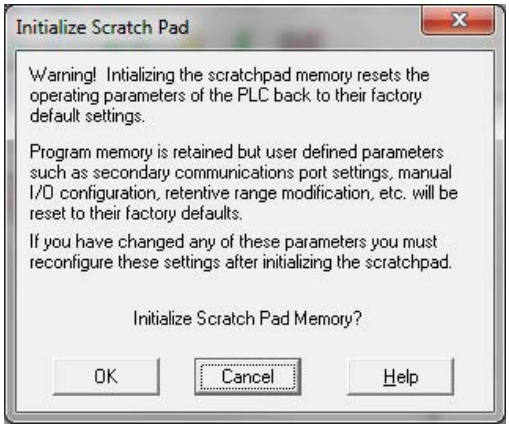
NOTE: *Resetting to factory defaults will not clear any password stored in the PLC.*

Resetting a DirectLogic PLC to Factory Defaults is a two-step process. Be sure to have a verified backup of your program using "Save Project to Disk" from the File menu before performing this procedure. Please be aware that the program as well as any settings will be erased and not all settings are stored in the project. In particular you will need to write down any settings for Secondary Communications Ports and manually set the ports up after resetting the PLC to factory defaults.



Step 1 – While connected to the PLC with DirectSoft, go to the PLC menu and select; "Clear PLC Memory". Check the "ALL" box at the bottom of the list and press "OK".

Step 2 – While connected with DirectSoft, go the PLC menu and then to the "Setup" submenu and select "Initialize Scratch Pad". Press "Ok".



NOTE: All configurable communications ports will be reset to factory default state. If you are connected via Port 2 or another configurable port, you may be disconnected when this operation is complete.

NOTE: *Retentive ranges will be reset to the factory settings.*

NOTE: *Manually addressed IO will be reset to factory default settings.*

The PLC has now been reset to factory defaults and you can proceed to program the PLC.

Errata Sheet

This Errata Sheet contains corrections or changes made after the publication of this manual.

06.13.2012

Changes to Chapter 3. DL330/DL330P/DL340 CPU Specifications

Page 3-3. CPU Specifications

DL340 CPUs DO NOT support overrides. In row 3, Supports Overrides, change the “Yes” in the DL340 column to “No”.

11

Instruction Set

In This Chapter. . . .

- Boolean Instructions
- Comparative Boolean Instructions
- Timer, Counter, and Shift Register Instructions
- Accumulator Load and Output Instructions
- Accumulator Logic Instructions
- Math Instructions
- Bit Operation Instructions
- Number Conversion Instructions
- Program Control Instructions
- Network Instructions
- Message Instructions

Handheld Programmer D3-HP & D3-HPP have been retired as of 03/2021 & 01/2018 respectively. Please consider Productivity, BRX, or CLICK series PLC systems as upgrades.

Introduction

The DL305 CPUs offer a wide variety of instructions to perform many different types of operations. This chapter shows you how to use these individual instructions. The following table provides a quick reference listing of the instruction mnemonic and the page(s) defining the instruction. (The mnemonics are very similar to the instruction names and should be easy to become familiar with in a short time.) For example STR NOT (Comparative) is the mnemonic for Store Not Equal. Each instruction definition will show in parentheses the keystrokes used to enter the instruction.

There are two ways to quickly find the instruction you need.

- If you know the instruction category (Boolean, Comparative Boolean, etc.) just use the header at the top of the page to find the pages that discuss the instructions in that category.
- If you know the individual instruction mnemonic, use the following table to find the page that discusses the instruction.

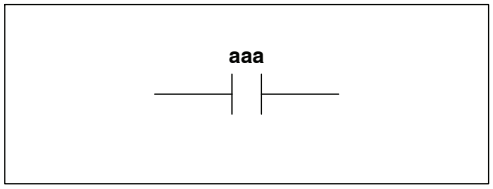
The DL330 and DL340 instructions sets are very similar. However, the DL330P instruction set has several differences. Some of the instructions in this chapter will be labeled “DL330/DL340 Only.” There may be an equivalent instruction for the DL330P CPU, but it may also work slightly differently. The DL330P instructions that operate differently from these instructions are discussed in Chapter 12. Make sure you review the instructions carefully to make sure you can use the instruction with your CPU.

NOTE: See Chapter 12 for RLL*PLUS* instructions.

Boolean Instructions

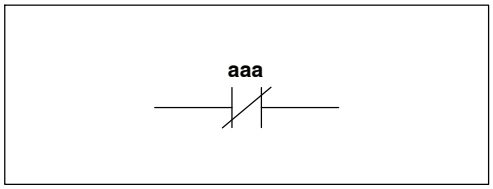
Store (STR)

The Store instruction begins a new rung or additional branch in a rung with a normally open contact. Status of the contact will be the same state as the associated image register point or memory location.



Store Not (STR NOT)

The Store Not instruction begins a new rung or additional branch in a rung with a normally closed contact. Status of the contact will be opposite the state of the associated image register point or memory location.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	aaa	aaa
Inputs / Outputs	000-167 700-767	000-177 700-767	000-167 700-767
Control Relays	160-373	160-373 1000-1067	160-174 200-77
Special Control Relays	374-377 770-777	374-377 770-777 1074-1077	175-177 770-777
Shift Register Bits	400-577	400-577	--

In the following Store example, when input 000 is on, output 010 will energize.

DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	0	ENT	
OUT	SHF	1	0	ENT

In the following Store Not example, when input 000 is off output 010 will energize.

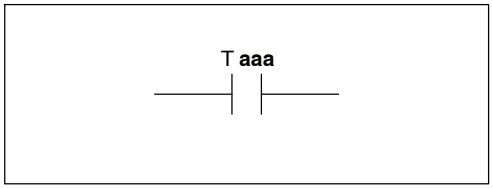
DirectSOFT Display

Handheld Programmer Keystrokes

STR	NOT	SHF	0	ENT
OUT	SHF	1	0	ENT

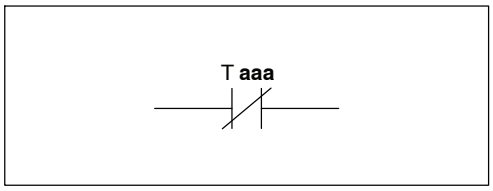
Store Timer (STR TMR) DL330/340 Only

The Store Timer instruction begins a new rung or additional branch in a rung with a normally open timer contact. The timer contact T aaa will be on when the timer current value is ≥ the preset value of the associated timer.



Store Not Timer (STR NOT TMR) DL330/340 Only

The Store Not Timer instruction begins a new rung or additional branch in a rung with a normally closed timer contact. The timer contact T aaa will be on when the timer current value is < the preset value of the associated timer.



Data Type		D3-330 Range	D3-340 Range	D3-330P Range*
		aaa	aaa	aaa
Timer	T	600-677	600-677	--

* See Chapter12 for similar RLL*PLUS* instructions

In the following Store Timer example, when the current value in timer 600 is ≥ the preset value output 017 will energize.

DirectSOFT Display

Handheld Programmer Keystrokes

STR	TMR	SHF	6	0	0	ENT	
OUT	SHF	1	7	ENT			

In the following Store Not Timer example, when the current value in timer 600 is < the preset value output 017 will energize.

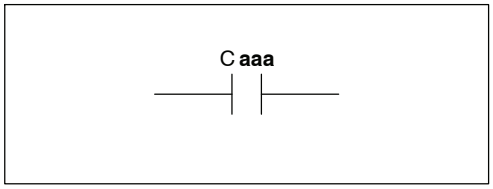
DirectSOFT Display

Handheld Programmer Keystrokes

STR	NOT	TMR	SHF	6	0	0	ENT
OUT	SHF	1	7	ENT			

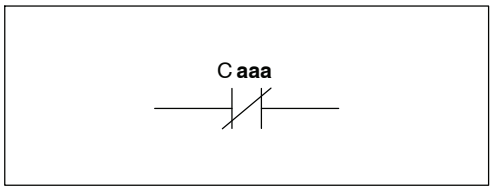
Store Counter (STR CNT) DL330/340 Only

The Store Counter instruction begins a new rung or additional branch in a rung with a normally open counter contact. The counter contact C aaa will be on when the counter current value ≥ the preset value of the associated counter.



Store Not Counter (STR NOT CNT) DL330/340 Only

The Store Not Counter instruction begins a new rung or additional branch in a rung with a normally closed counter contact. The counter contact C aaa will be on when the counter current value is < the preset value of the associated counter.



Data Type		D3-330 Range	D3-340 Range	D3-330P Range*
		aaa	aaa	aaa
Counter	C	600-677	600-677	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following Store Counter example, when the current value in counter 602 is ≥ the preset value output 015 will energize.

DirectSOFT Display

Handheld Programmer Keystrokes

STR	CNT	SHF	6	0	2	ENT
OUT	SHF	1	5	ENT		

In the following Store Not Counter example, when the current value in counter 602 is < the preset value output 015 will energize.

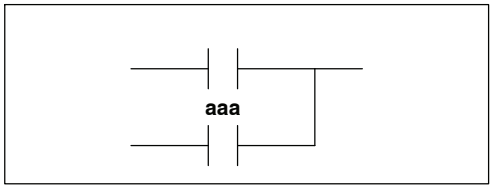
DirectSOFT Display

Handheld Programmer Keystrokes

STR	NOT	CNT	SHF	6	0	2	ENT
OUT	SHF	1	5	ENT			

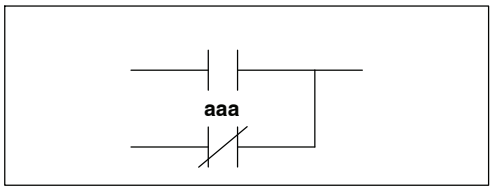
Or (OR)

The Or instruction logically ors a normally open contact in parallel with another contact in a rung. The status of the contact will be the same state as the associated image register point or memory location.



Or Not (OR NOT)

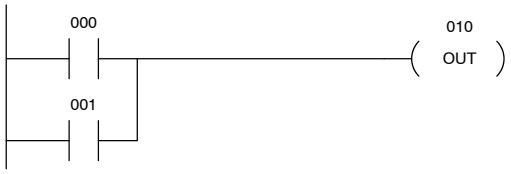
The Or Not instruction logically ors a normally closed contact in parallel with another contact in a rung. The status of the contact will be opposite the state of the associated image register point or memory location.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	aaa	aaa
Inputs / Outputs	000-167 700-767	000-177 700-767	000-167 700-767
Control Relays	160-373	160-373 1000-1067	160-174 200-77
Special Control Relays	374-377 770-777	374-377 770-777 1074-1077	175-177 770-777
Shift Register Bits	400-577	400-577	--

In the following Or example, when input 000 or 001 is on output 010 will energize.

DirectSOFT Display



Handheld Programmer Keystrokes

STR	SHF	0	ENT	
OR	SHF	1	ENT	
OUT	SHF	1	0	ENT

In the following Or Not example, when input 000 is on or 001 is off output 010 will energize.

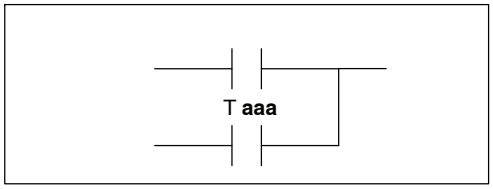
DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	0	ENT	
OR	NOT	SHF	1	ENT
OUT	SHF	1	0	ENT

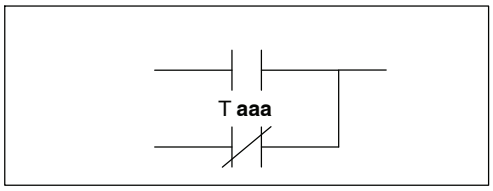
Or Timer (OR TMR) DL330/340 Only

The Or Timer instruction logically ors a normally open timer contact in parallel with another contact in a rung. The timer contact T aaa will be on when the timer current value is ≥ the preset value of the associated timer.



Or Not Timer (OR NOT TMR) DL330/340 Only

The Or Not Timer instruction logically ors a normally closed timer contact in parallel with another contact in a rung. The timer contact T aaa will be on when the timer current value is < the preset value of the associated timer.



Data Type		D3-330 Range	D3-340 Range	D3-330P Range*
		aaa	aaa	aaa
Timer	T	600-677	600-677	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following Or Timer example, when input 000 is on or the current value in T600 is ≥ the preset value output 010 will energize.

DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	0	ENT				
OR	TMR	SHF	6	0	0	ENT	
OUT	SHF	1	0	ENT			

In the following Or Not Timer example, when input 000 is on or the current value in T600 is < the preset value output 010 will energize.

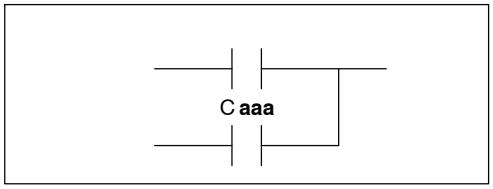
DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	0	ENT				
OR	NOT	TMR	SHF	6	0	0	ENT
OUT	SHF	1	0	ENT			

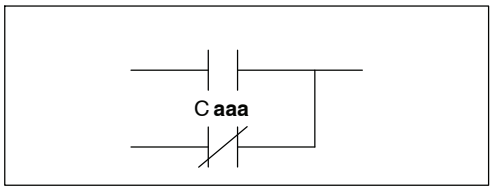
Or Counter (OR CNT) DL330/340 Only

The Or Counter instruction logically ors a normally open counter contact in parallel with another contact in a rung. The counter contact C aaa will be on when the counter current value is $\geq$ the preset value of the associated counter.



Or Not Counter (OR NOT CNT) DL330/340 Only

The Or Not Counter instruction logically ors a normally closed counter contact in parallel with another contact in a rung. The counter contact C aaa will be on when the counter current value is < the preset value of the associated counter.

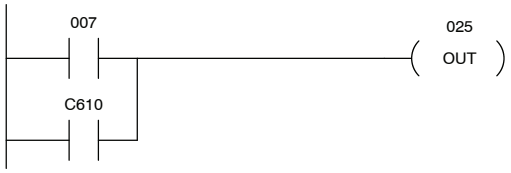


Data Type		D3-330 Range	D3-340 Range	D3-330P Range*
		aaa	aaa	aaa
Counter	C	600-677	600-677	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following Or Counter example, when input 007 is on or the current value in C610 is $\geq$ the preset value output 025 will energize.

DirectSOFT Display



Handheld Programmer Keystrokes

STR	SHF	7	ENT				
OR	CNT	SHF	6	1	0	ENT	
OUT	SHF	2	5	ENT			

In the following Or Not Counter example, when input location 007 is on or the current value in C610 is < the preset value output 025 will energize.

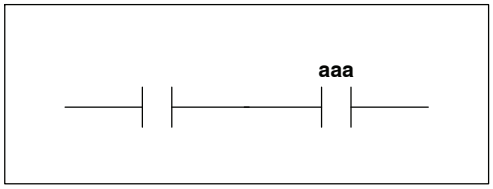
DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	7	ENT				
OR	NOT	CNT	SHF	6	1	0	ENT
OUT	SHF	2	5	ENT			

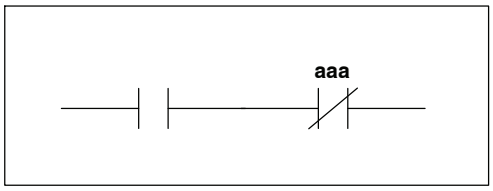
And (AND)

The And instruction logically ands a normally open contact in series with another contact in a rung. The status of the contact will be the same state as the associated image register point or memory location.



And Not (AND NOT)

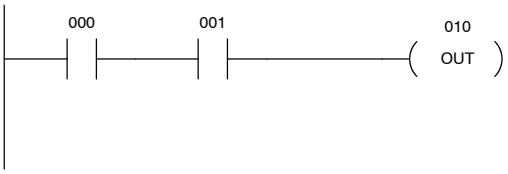
The And Not instruction logically ands a normally closed contact in series with another contact in a rung. The status of the contact will be opposite the state of the associated image register point or memory location.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	aaa	aaa
Inputs / Outputs	000-167 700-767	000-177 700-767	000-167 700-767
Control Relays	160-373	160-373 1000-1067	160-174 200-77
Special Control Relays	374-377 770-777	374-377 770-777 1074-1077	175-177 770-777
Shift Register Bits	400-577	400-577	--

In the following And example, when input 000 and 001 is on output 010 will energize.

DirectSOFT Display

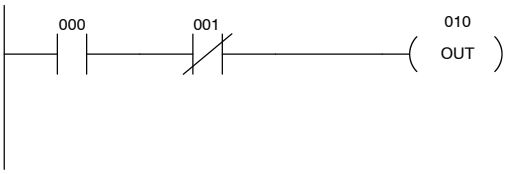


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
AND	SHF	1	ENT	
OUT	SHF	1	0	ENT

In the following And Not example, when input 000 is on and 001 is off output 010 will energize.

DirectSOFT Display

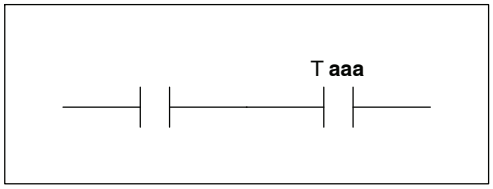


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
AND	NOT	SHF	1	ENT
OUT	SHF	1	0	ENT

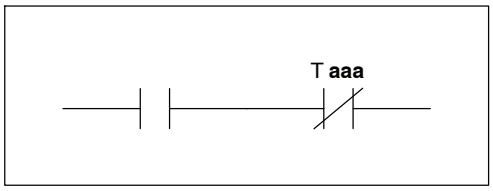
And Timer (AND TMR) DL330/340 Only

The And Timer instruction logically ands a normally open timer contact in series with another contact in a rung. The timer contact T aaa will be on when the timer current value ≥ the preset value of the associated timer.



And Not Timer (AND NOT TMR) DL330/340 Only

The And Not Timer instruction logically ands a normally closed timer contact in series with another contact in a rung. The timer contact T aaa will be on when the timer current value is < the preset value of the associated timer.

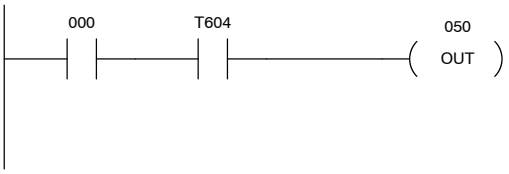


Data Type		D3-330 Range	D3-340 Range	D3-330P Range*
		aaa	aaa	aaa
Timer	T	600-677	600-677	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following And Timer example, when input 000 is on and the current value in timer 604 is ≥ the preset value output 050 will energize.

DirectSOFT Display

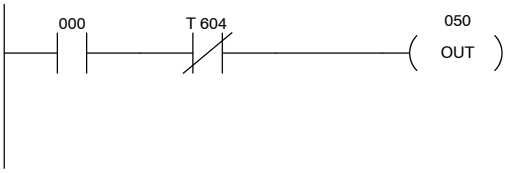


Handheld Programmer Keystrokes

STR	SHF	0	ENT			
AND	TMR	SHF	6	0	4	ENT
OUT	SHF	5	0	ENT		

In the following And Not Timer example, when input 000 is on and the current value in timer 604 is < the preset value output 050 will energize.

DirectSOFT Display

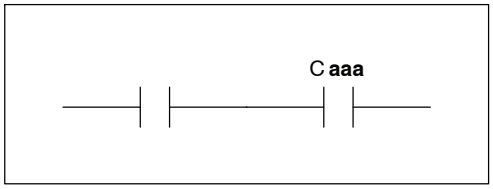


Handheld Programmer Keystrokes

STR	SHF	0	ENT				
AND	NOT	TMR	SHF	6	0	4	ENT
OUT	SHF	5	0	ENT			

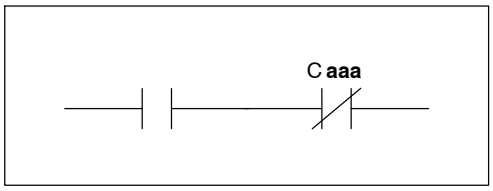
And Counter (AND CNT) DL330/340 Only

The And Counter instruction logically ands a normally open counter contact in series with another contact in a rung. The counter contact C aaa will be on when the counter current value is ≥ the preset value of the associated counter.



And Not Counter (AND NOT CNT) DL330/340 Only

The And Not Counter instruction logically ands a normally closed counter contact in series with another contact in a rung. The counter contact C aaa will be on when the counter current value is < the preset value of the associated counter.

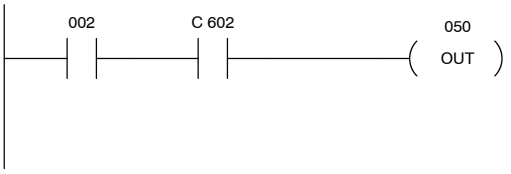


Data Type		D3-330 Range	D3-340 Range	D3-330P Range
		aaa	aaa	aaa
Counter	C	600-677	600-677	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following And Counter example, when input 002 is on and the current value in counter 602 is ≥ the preset value output 050 will energize.

DirectSOFT Display

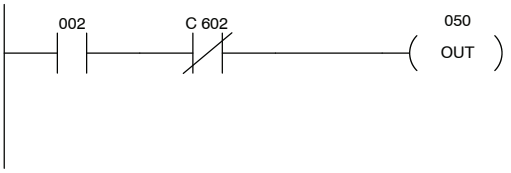


Handheld Programmer Keystrokes

STR	SHF	2	ENT			
AND	CNT	SHF	6	0	2	ENT
OUT	SHF	5	0	ENT		

In the following And Not Counter example, when input 002 is on and the current value in counter 602 is < the preset value output 050 will energize.

DirectSOFT Display

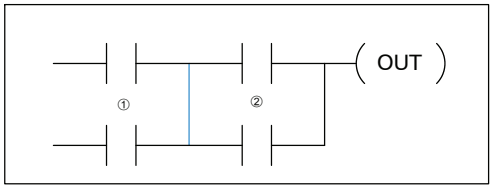


Handheld Programmer Keystrokes

STR	SHF	2	ENT				
AND	NOT	TMR	SHF	6	0	2	ENT
OUT	SHF	5	0	ENT			

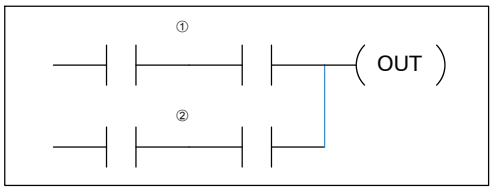
And Store (AND STR)

The And Store instruction logically ands two branches of a rung in series. Both branches must begin with the Store instruction.



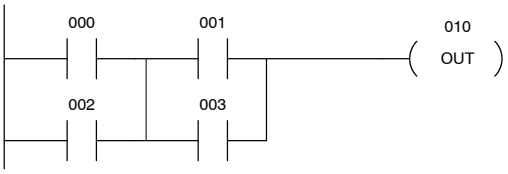
Or Store (OR STR)

The Or Store instruction logically ors two branches of a rung in parallel. Both branches must begin with the Store instruction.



In the following And Store example, the branch consisting of contacts 000 and 002 have been anded with the branch consisting of contacts 001 and 003.

DirectSOFT Display

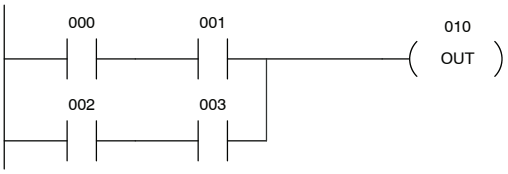


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
OR	SHF	2	ENT	
STR	SHF	1	ENT	
OR	SHF	3	ENT	
AND	STR	ENT		
OUT	SHF	1	0	ENT

In the following Or Store example, the branch consisting of 000 and 001 have been ored with the branch consisting of 002 and 003.

DirectSOFT Display

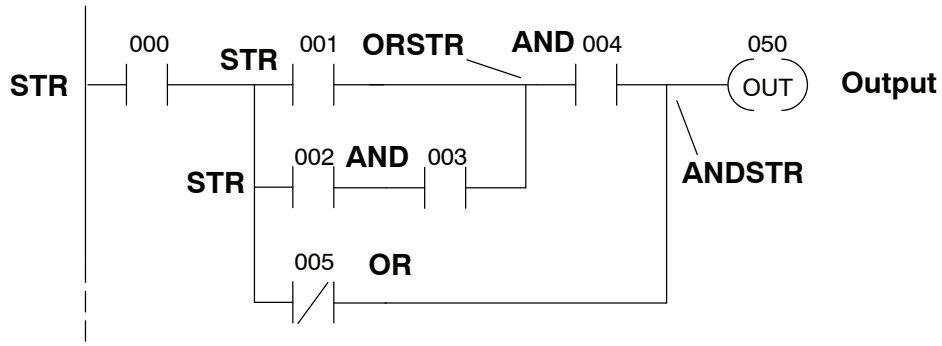


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
AND	SHF	1	ENT	
STR	SHF	2	ENT	
AND	SHF	3	ENT	
OR	STR	ENT		
OUT	SHF	1	0	ENT

There are limits to what you can enter with these simple boolean instructions. This is because the DL305 CPUs use an 8-level stack to evaluate the various logic elements. The stack is a temporary storage area that helps solve the logic for the rung. Each time you enter a Store instruction, the instruction is placed on the top of the stack. Any other instructions on the stack are pushed down a level. The And, Or, And Store, and Or Store instructions combine levels of the stack when they are encountered. Since the stack is only eight levels, an error will occur if the CPU encounters a rung that uses more than the eight levels of the stack.

The following example shows how the stack is used to solve simple boolean logic.



1 STR 000

1	STR 000
2	
3	
4	
5	
6	
7	
8	

2 STR 001

1	STR 001
2	STR 000
3	
4	
5	
6	
7	
8	

3 STR 002

1	STR 002
2	STR 001
3	STR 000
4	
5	
6	
7	
8	

4 AND 003

1	002 AND 003
2	STR 001
3	STR 000
4	
5	
6	
7	
8	

5 ORSTR

1	001 OR (002 AND 003)
2	STR 000
3	
.	
.	
8	

6 AND 004

1	004 AND [001 OR (002 AND 003)]
2	STR 000
3	
.	
.	
8	

7 OR 005

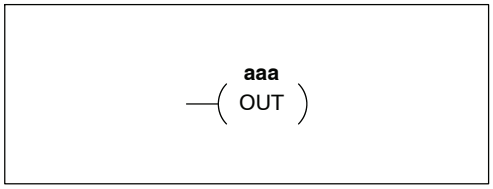
1	005 OR 004 AND [001 OR (002 AND 003)]
2	STR 000
3	
.	
.	
8	

8 ANDSTR

1	000 AND (005 OR 004) AND [001 OR (002 AND 003)]
2	
3	
.	
.	
8	

Out (OUT)

The Out instruction reflects the status of the rung (on/off) and outputs the discrete (on/off) state to the specified image register point or memory location. Multiple Out instructions referencing the same discrete location should not be used because only the last Out instruction in the program will control the physical output point.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	**aaa**	**aaa**
Outputs	000-167 700-767	000-177 700-767	000-167 700-767
Control Relays	160-373	160-373 1000-1067	160-174 200-77
Shift Register Bits	400-577	400-577	--

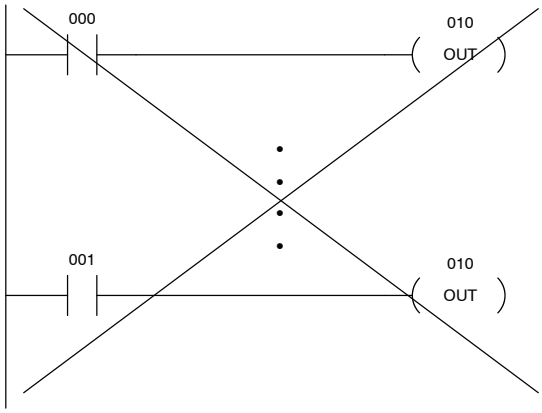
In the following Out example, when input 000 is on output 010 will energize.

DirectSOFT Display

Handheld Programmer Keystrokes

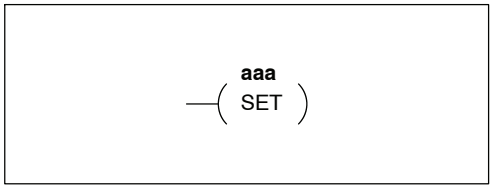
STR	SHF	0	ENT	
OUT	SHF	1	0	ENT

In the following Out example, two Out instructions using output 10 are used in the program. The status of output 010 being controlled by input 001 will override the instance of output 010 being controlled by input 000. The physical output would always be controlled by input 001.



Set (SET) DL330/340 Only

The Set instruction sets or turns on an output. Once the output is set it will remain on until it is reset using the Reset instruction. It is not necessary for the input controlling the Set instruction to remain on. The Set instruction is sometimes known as a latch. The Reset instruction is used to reset the output.



Reset (RST) DL330/340 Only

The Reset instruction resets or turns off an output. Once the output is reset it is not necessary for the input to remain on. The Reset instruction is sometimes known as an unlatch instruction.

Data Type	D3-330 Range	D3-340 Range	D3-330P Range*
	aaa	aaa	aaa
Outputs	000-167 700-767	000-177 700-767	--
Control Relays	160-373	160-373 1000-1067	--
Shift Register Bits	400-577	400-577	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following Set example, when input 000 is on output 010 will be energized.

DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	0	ENT	
SET	SHF	1	0	ENT

In the following Reset example, when input 001 is on output 010 will de-energize.

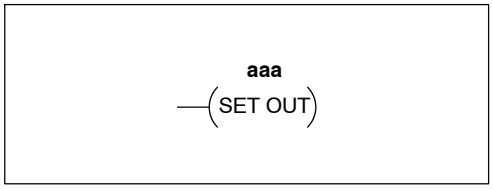
DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	1	ENT	
RST	SHF	1	0	ENT

Set Out (SET OUT)

The Set Out instruction reflects the status of the rung (on/off) and outputs the discrete (on/off) state to the specified image register location. This instruction is similar to the Out instruction except the output disable coil (special relay 376) will not override and disable the output. Multiple Set Out instructions referencing the same discrete location should not be used because only the last Set Out instruction in the program will control the physical output point.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	**aaa**	**aaa**
Outputs	000-167 700-767	000-177 700-767	000-167 700-767

In the following Set Out example, when input location 000 is on output 020 will energize. The output disable coil (special relay 376) will not override this output coil.

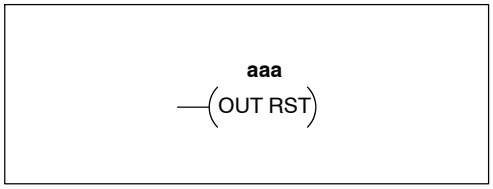
DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	0	ENT		
SET	OUT	SHF	2	0	ENT

Set Out Reset (SET OUT RST)

The Set Out Reset instruction is typically known as a one shot. When the input logic produces an off to on transition the output will turn on for one CPU scan.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	**aaa**	**aaa**
Outputs	000-167 700-767	000-177 700-767	000-167 700-767
Control Relays	160-373	160-373 1000-1067	160-174 200-77

In the following Set Out Reset example, when input 007 transitions from off to on, control relay 160 will energize for the remainder of the CPU scan.

DirectSOFT Display

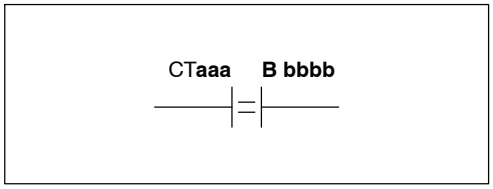
Handheld Programmer Keystrokes

STR	SHF	7	ENT				
SET	OUT	RST	SHF	1	6	0	ENT

Comparative Boolean Instructions

Store If Equal (STR) DL330/DL340 Only

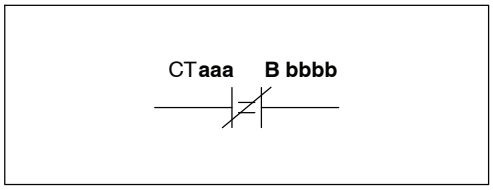
The Store If Equal instruction begins a new rung or additional branch in a rung with a normally open comparative counter contact. The contact will be on if the specified counter CT aaa = B bbbb.



CT

Store Not If Equal (STR NOT) DL330/DL340 Only

The Store Not If Equal instruction begins a new rung or additional branch in a rung with a normally closed comparative counter contact. The contact will be on if the specified counter CT aaa ≠ B bbbb.



Operand Data Type		D3-330 Range		D3-340 Range		D3-330P Range*	
	B	aaa	bbbb	aaa	bbbb	aaa	bbbb
Counters	CT	600-677	--	600-677	--	--	--
Data registers	R	--	400-577	--	400-577 700-777	--	--
Constant	K	--	0-9999	--	0-9999	--	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following Store If Equal example, when CT600 = 2510 output 012 will energize.

DirectSOFT Display

Handheld Programmer Keystrokes

STR	SHF	6	0	0	ENT
SHF	2	5	1	0	ENT
OUT	SHF	0	1	2	ENT

In the following Store Not If Equal example, when CT600 is ≠ the value in R400 output 020 will energize.

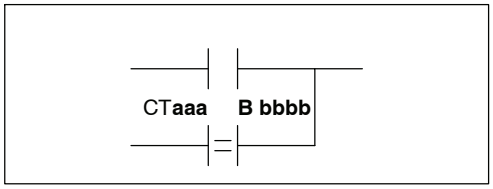
DirectSOFT Display

Handheld Programmer Keystrokes

STR	NOT	SHF	6	0	0	ENT
R	4	0	0	ENT		
OUT	SHF	0	1	2	ENT	

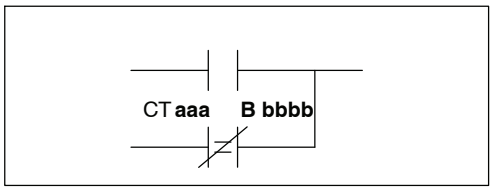
Or If Equal (OR) DL330/DL340 Only

The Or If Equal instruction connects a normally open comparative counter contact in parallel with another contact. The contact will be on if the specified counter CT aaa = B bbbb.



Or Not If Equal (OR NOT) DL330/DL340 Only

The Or Not If Equal instruction connects a normally closed comparative counter contact in parallel with another contact. The contact will be on if the specified counter
CT aaa ≠ B bbbb.

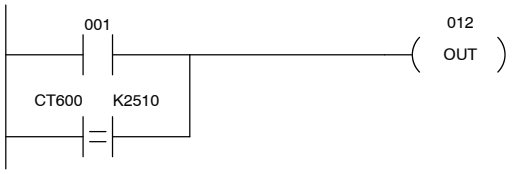


Operand Data Type		D3-330 Range		D3-340 Range		D3-330P Range*	
	B	aaa	bbbb	aaa	bbbb	aaa	bbbb
Counters	CT	600-677	--	600-677	--	--	--
Data registers	R	--	400-577	--	400-577 700-777	--	--
Constant	K	--	0-9999	--	0-9999	--	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following Or If Equal example, when input contact 001 is on or CT600 = 2510 output 012 will energize.

DirectSOFT Display

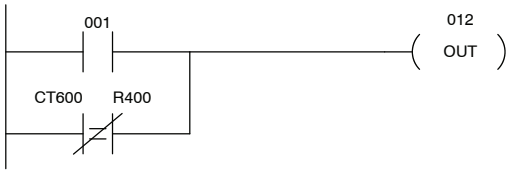


Handheld Programmer Keystrokes

STR	SHF	1	ENT		
OR	SHF	6	0	0	ENT
SHF	2	5	1	0	ENT
OUT	SHF	0	1	2	ENT

In the following Or Not If Equal example, when input contact 001 is on or CT600 ≠ the value in R400 output 012 will energize.

DirectSOFT Display

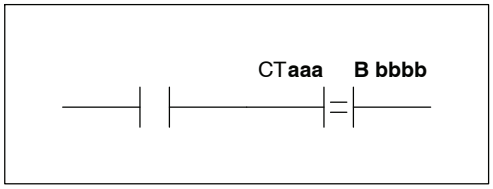


Handheld Programmer Keystrokes

STR	SHF	1	ENT		
OR	NOT	6	0	0	ENT
R	4	0	0	ENT	
OUT	SHF	0	1	2	ENT

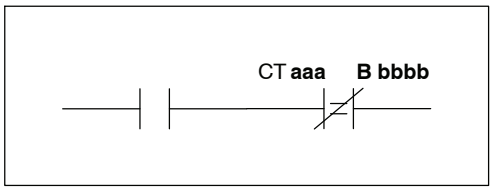
And If Equal (AND) DL330/DL340 Only

The And If Equal instruction connects a normally open comparative counter contact in series with another contact. The contact will be on if the specified counter CT aaa = B bbbb.



And Not If Equal (AND NOT) DL330/DL340 Only

The And Not If Equal instruction connects a normally closed comparative counter contact in series with another contact. The contact will be on if the specified counter CT aaa ≠ B bbbb.

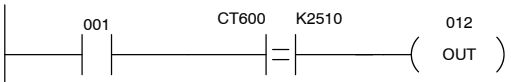


Operand Data Type		D3-330 Range		D3-340 Range		D3-330P Range*	
	B	aaa	bbbb	aaa	bbbb	aaa	bbbb
Counters	CT	600-677	--	600-677	--	--	--
Data registers	R	--	400-577	--	400-577 700-777	--	--
Constant	K	--	0-9999	--	0-9999	--	--

***** See Chapter 12 for similar RLL*PLUS* instructions

In the following And If Equal example, when input contact 001 is on and CT600 = 2510 the contact will turn on and output 012 will energize.

DirectSOFT Display

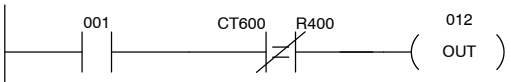


Handheld Programmer Keystrokes

STR	SHF	1	ENT		
AND	SHF	6	0	0	ENT
SHF	2	5	1	0	ENT
OUT	SHF	0	1	2	ENT

In the following And Not If Equal example, when input contact 001 is on and CT600 ≠ the value in R400 output 012 will energize.

DirectSOFT Display



Handheld Programmer Keystrokes

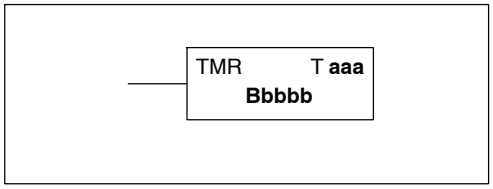
STR	SHF	1	ENT		
AND	NOT	6	0	0	ENT
R	4	0	0	ENT	
OUT	SHF	0	1	2	ENT

Timer, Counter, and Shift Register Instructions

Timer (TMR) DL330/DL340 Only

See Errata Sheet at the beginning of this file. The counters and timers both time in Decimal and not in BCD. Presets for both are also interpreted as decimal data and not as BCD.

The Timer instruction provides a single input timer with a 0.1 second increment (0-999.9 seconds) in the normal operating mode, or a 0.01 second increment (0-99.99 seconds) in the fast timer mode when relay 770 is turned on. The timer will time up to 9999 and stop. It will reset to zero when the input is turned off. The discrete bit associated with the timer will be on when the current value is equal to or greater than the preset value.

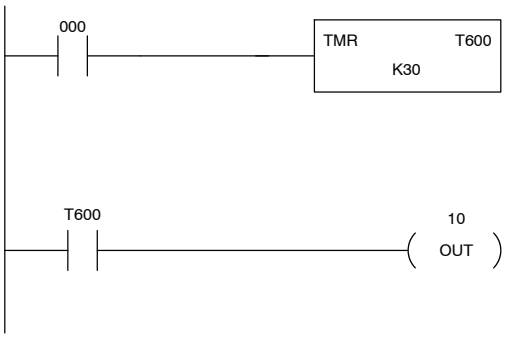


Operand Data Type		D3-330 Range		D3-340 Range		D3-330P Range	
	B	aaa	bbbb	aaa	bbbb	aaa	bbbb
Timers	T	600-677	--	600-677	--	--	--
Data registers	R	--	400-577	--	400-577 700-777	--	--
Constant	K	--	0-9999	--	0-9999	--	--

* See Chapter 12 for similar RLL*PLUS* instructions

In the following Timer example, timer 600 will begin timing up when input 000 turns on. The timer bit associated with timer 600 will turn on when the current value in timer 600 is $\geq$ the preset value K30 (3 seconds). When input 000 turns off the timer discrete bit and current value are reset.

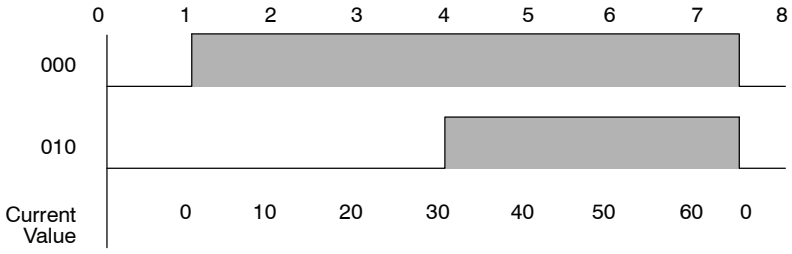
***Direct*SOFT Display**



Handheld Programmer Keystrokes

STR	SHF	0	ENT			
TMR	SHF	6	0	0	ENT	
K	3	ENT				

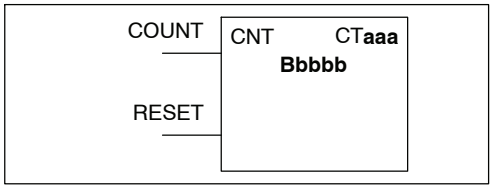
Timing Diagram



Counter (CNT) DL330/DL340 Only

See Errata Sheet at the beginning of this file. The counters and timers both time in Decimal and not in BCD. Presets for both are also interpreted as decimal data and not as BCD.

The Counter instruction provides a counter with a count and reset input. The range of this counter is 0-9999 and it will increment when the count input transitions from off to on. The counter is reset to 0 when you turn on the reset input. The counter bit associated with the counter will turn on when the current value is equal to or greater than the preset value.

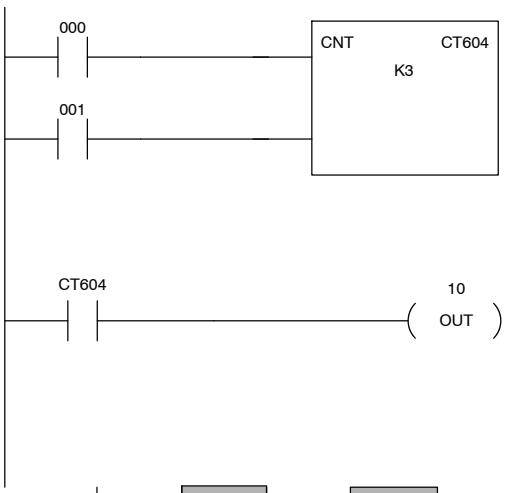


Operand Data Type		D3-330 Range		D3-340 Range		D3-330P Range	
	B	aaa	bbbb	aaa	bbbb	aaa	bbbb
Counters	CT	600-677	--	600-677	--	--	--
Data registers	R	--	400-577	--	400-577 700-777	--	--
Constant	K	--	0-9999	--	0-9999	--	--

* See Chapter 12 for similar RLL*PLUS* instructions

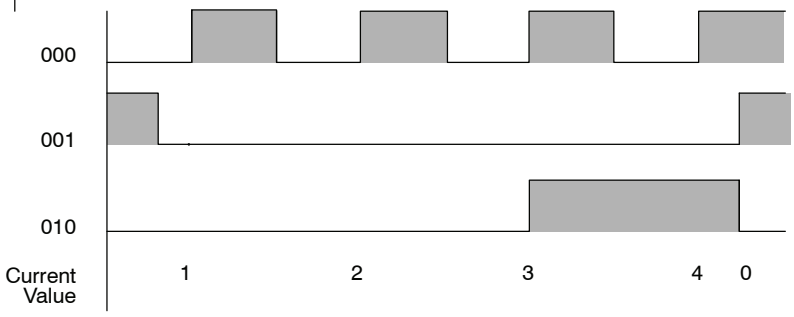
In the following Counter example, counter 604 will increment by one count when input 000 transitions from off to on. When input contact 001 is turned on the counter will reset to zero. The counter bit associated with counter 604 will turn on when the current value in counter 604 is ≥ the preset constant value K3 (3).

DirectSOFT Display



Handheld Programmer Keystrokes

STR	SHF	0	ENT		
STR	SHF	1	ENT		
CNT	SHF	6	0	4	ENT
SHF	3	ENT			

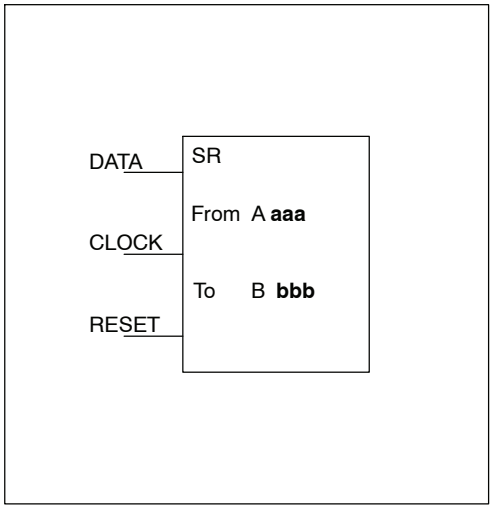


Shift Register (SR) DL330/340 Only

The Shift Register instruction shifts data through a predefined number of shift register bits. There are 128 bits allocated for use in shift registers. There is no limit to the number of shift registers which can be used in a program, however the total number of bits used cannot exceed 128.

The Shift Register has three contacts.

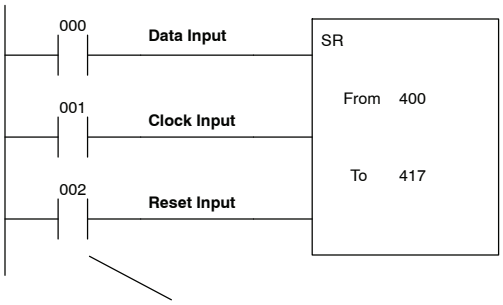
- Data — determines the value (1 or 0) that will enter the register
- Clock — shifts the bits one position on each off to on transition
- Reset —resets the Shift Register to all zeros.



With each off to on transition of the clock input, the bits which make up the shift register block are shifted by one bit position and the status of the data input is shifted into the starting bit position in the block. The direction of the shift depends on the entry in the From and To fields. From 400 to 407 would define a block of eight bits to be shifted from bit 400 to bit 407. From 407 to 400 would also define a block of eight bits, but would shift from bit 407 to bit 400. The maximum size of the shift register block is limited to 128 bits. There is no minimum block size.

Operand Data Type	D3-330 Range		D3-340 Range		D3-330P Range	
	aaa	bbbb	aaa	bbbb	aaa	bbbb
Shift Register Bits	400-577	400-577	400-577	400-577	--	--

DirectSOFT Display



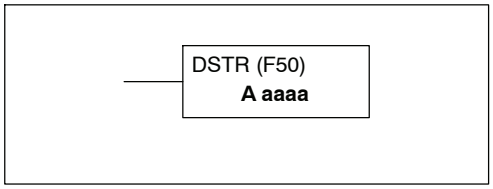
Handheld Programmer Keystrokes

STR	SHF	0	ENT		
STR	SHF	1	ENT		
STR	SHF	2	ENT		
SR	SHF	4	0	0	ENT
SHF	4	1	7	ENT	

Accumulator Load and Output Instructions

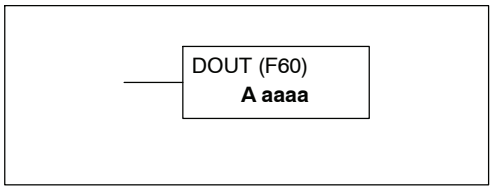
Data Store DSTR (F50)

The Data Store (F50) is a 16-bit instruction that loads the value of a 16-bit register, two consecutive 8-bit registers (specify starting location), or a 4-digit BCD value into the accumulator.



Data Out DOUT (F60)

The Data Out (F60) is a 16-bit instruction that copies the 16-bit value in the accumulator to a 16-bit reference or two consecutive 8-bit registers (specify starting location).

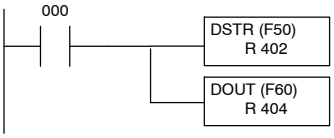


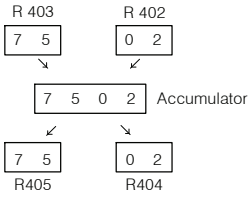
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
*Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

* A constant is not a valid data type for the DOUT (F60) instruction.

In the following example, when input 000 is on the value (7502) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The value in the accumulator is output to data registers R404 and R405 using the Data Out (F60) instruction.

***Direct*SOFT Display**



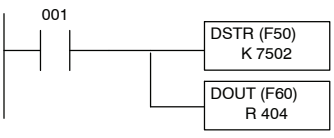


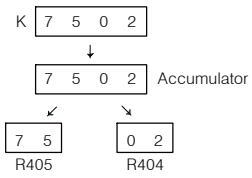
Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	6	0	ENT	
R	4	0	4	ENT

In the following example, when input 001 is on the BCD constant value K7502 is loaded into the accumulator using the Data Store (F50) instruction. The value in the accumulator is output to data registers R404 and R405 using the Data Out (F60) instruction.

***Direct*SOFT Display**



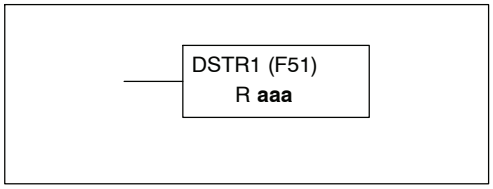


Handheld Programmer Keystrokes

STR	SHF	1	ENT		
F	5	0	ENT		
SHF	7	5	0	2	ENT
F	6	0	ENT		
R	4	0	4	ENT	

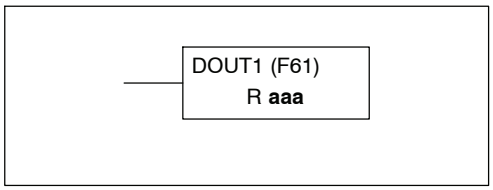
Data Store 1 DSTR (F51)

The Data Store 1 (F51) is an 8-bit instruction that loads the value from a specified 8-bit register into the lower 8 bits of the accumulator. The upper 8 bits of the accumulator are set to zero.



Data Out 1 DOUT (F61)

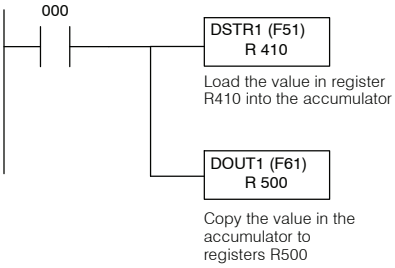
The Data Out 1 (F61) is an 8-bit instruction that copies the value in the lower 8 bits of the accumulator to a specified 8-bit register.

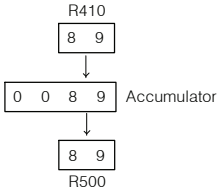


Data Type		D3-330 Range	D3-340 Range	D3-330P Range
		aaaa	aaaa	aaaa
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Data Registers	R	400-577	400-577 700-777	400-577

In the following example, when input 000 is on the value (89) in R410 is loaded into the lower 8 bits of the accumulator using the Data Store 1 (F51) instruction. The value in the least significant 8 bits of the accumulator is output to data register R500 using the Data Out 1 (F61) instruction.

***Direct*SOFT Display**



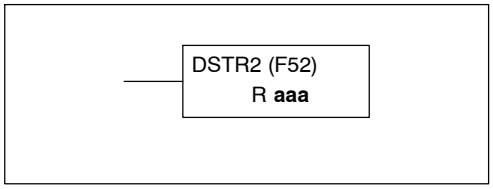


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	1	ENT	
R	4	1	0	ENT
F	6	1	ENT	
R	5	0	0	ENT

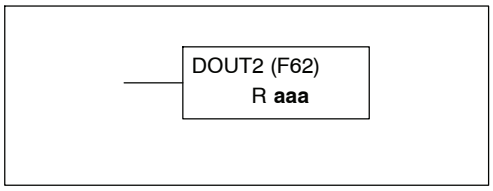
Data Store 2 DSTR (F52)

The Data Store 2 (F52) is a 4-bit instruction that loads the value of the most significant 4 bits of a specified 8-bit register into the least significant 4 bits of the accumulator. The remaining 12 bits of the accumulator are set to zero.



Data Out 2 DOUT (F62)

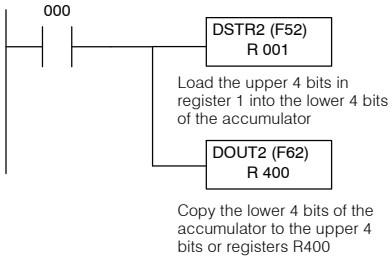
The Data Out 2 (F62) is a 4-bit instruction that copies the value in the least significant 4 bits of the accumulator into the most significant 4 bits of a specified 8-bit register. The lower 4 bits of the register are not altered .

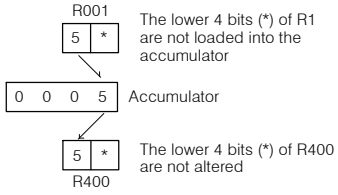


Data Type		D3-330 Range	D3-340 Range	D3-330P Range
		aaaa	aaaa	aaaa
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-27
Shift Registers	R	040-056	040-56	--
Stages	R	--	--	100-116
Data Registers	R	400-577	400-577 700-777	400-577

In the following example, when input 000 is on the most significant 4 bits of R1 are loaded into the lower 4 bits of the accumulator using the Data Store 2 (F52) instruction. The value in the least significant 4 bits of the accumulator is output to most significant 4 bits of data register R400 using the Data Out 2 (F62) instruction.

DirectSOFT Display



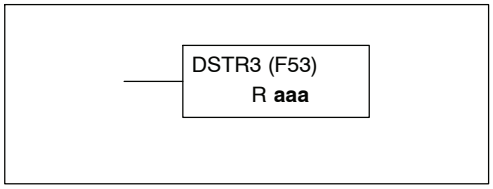


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	2	ENT	
R	1	ENT		
F	6	2	ENT	
R	4	0	0	ENT

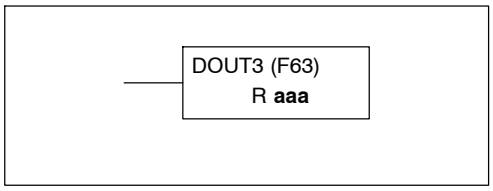
Data Store 3 DSTR (F53)

The Data Store 3 (F53) is a 4-bit instruction that loads the value of the least significant 4 bits of a specified 8-bit register into the least significant 4 bits of the accumulator. The upper 12 bits of the accumulator are set to zero.



Data Out 3 DOUT (F63)

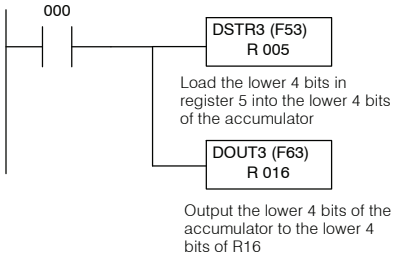
The Data Out 3 (F63) is a 4-bit instruction that copies the value in the least significant 4 bits of the accumulator to the least significant 4 bits of a specified 8 bit register. The upper 4 bits of the register are not altered.

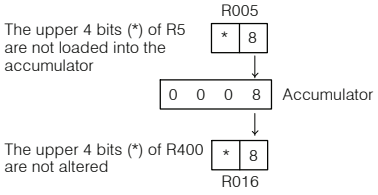


Data Type		D3-330 Range	D3-340 Range	D3-330P Range
		aaaa	aaaa	aaaa
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Data Registers	R	400-577	400-577 700-777	400-577

In the following example, when input 000 is on the least significant 4 bits of R005 are loaded into the accumulator using the Data Store 3 (F53) instruction. The data in the least significant 4 bits of the accumulator is output to the least significant 4 bits of R016 using the Data Out 3 (F63) instruction.

***Direct*SOFT Display**



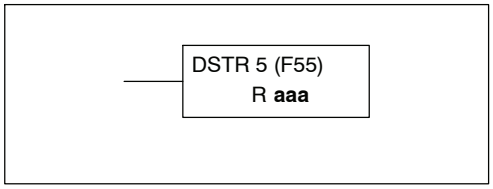


Handheld Programmer Keystrokes

STR	SHF	0	ENT
F	5	3	ENT
R	5	ENT	
F	6	3	ENT
R	1	6	ENT

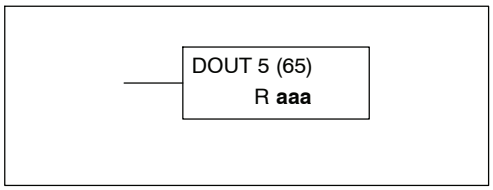
Data Store 5 DSTR (F55)

The Data Store 5 (F55) is a 16-bit instruction that loads the value of 16 image register locations for a specified 16 point input module into the accumulator.



Data Out 5 DOUT (F65)

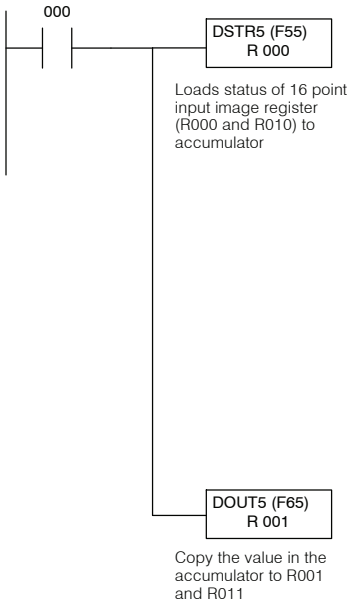
The Data Out 5 (F65) is a 16-bit instruction that outputs the 16 bit value in the accumulator to the image register of a specified 16 point output module.

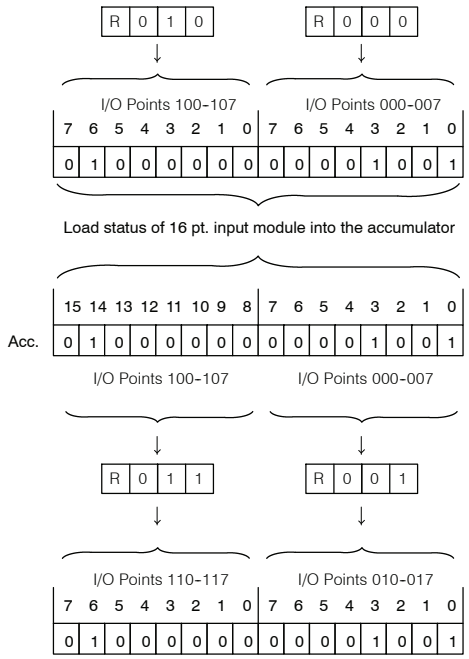


Data Type		D3-330 Range	D3-340 Range	D3-330P Range
		aaaa	aaaa	aaaa
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075

In the following example, when input 000 is on the binary status of a 16 point I/O module in slot 1 (R000 and R010) is loaded into the accumulator using the Data Store 5 (F55) instruction. The value in the accumulator is copied to I/O register locations in slot 2 (R001 and R011) using the Data Out 5 (F65) instruction.

***Direct*SOFT Display**





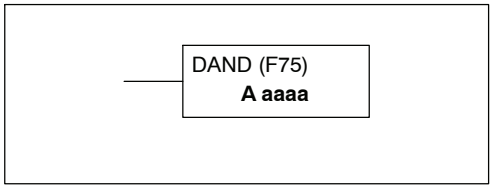
Handheld Programmer Keystrokes

STR	SHF	0	ENT
F	5	5	ENT
R	0	ENT	
F	6	5	ENT
R	1	ENT	

Accumulator Logic Instructions

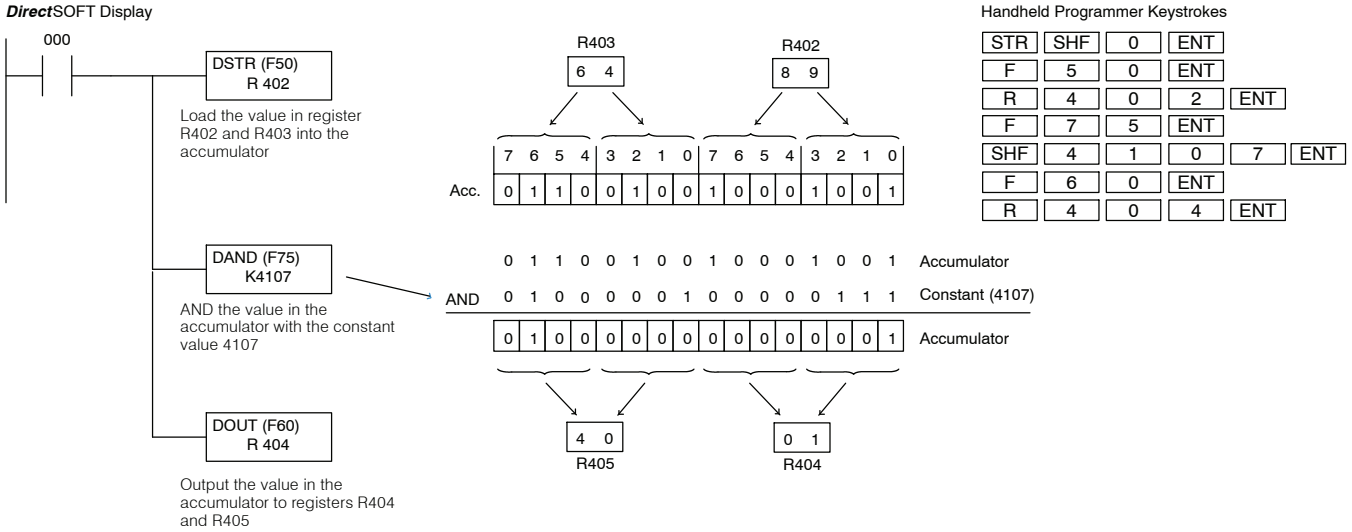
Data And DAND (F75)

The Data And (F75) is a 16-bit instruction that logical ands the value in a 16-bit reference, two consecutive 8-bit registers (specify starting location), or a 4-digit BCD constant with the value in the accumulator. The result resides in the accumulator.



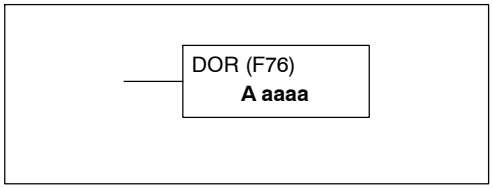
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	aaaa	aaaa	aaaa
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

In the following example, when input 000 is on the value(6489) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The data in the accumulator is logically anded with the constant K4107 with the result residing in the accumulator. The value in the accumulator is output to data register R404 and R405 using the Data Out (F60) instruction.



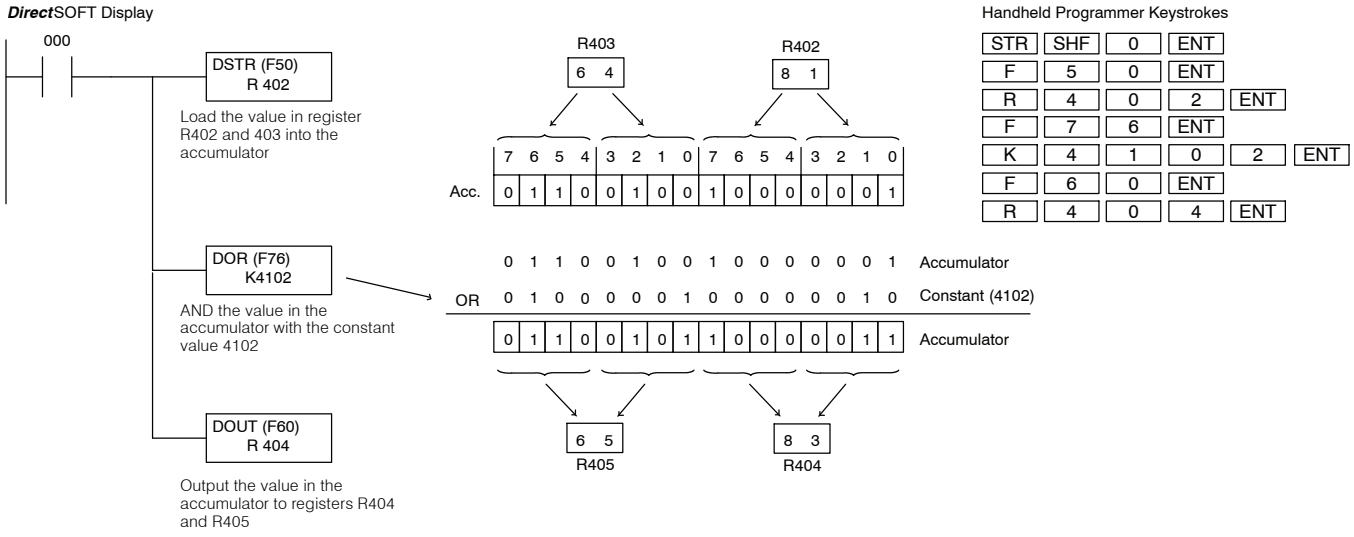
Data Or DOR (F76)

The Data Or (F76) is a 16-bit instruction that logically ors the value in a 16-bit reference, two consecutive 8-bit registers, (specify starting location) or a 4-digit BCD constant with the value in the accumulator. The result resides in the accumulator.



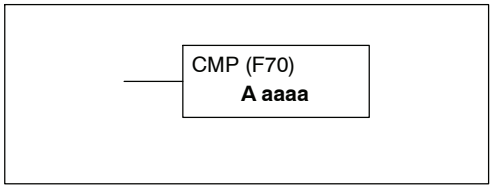
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

In the following example, when input 000 is on the value (6481) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The data in the accumulator is logically ored with the constant K4102 with the result residing in the accumulator. The value in the accumulator is output to data registers R404 and R405 using the Data Out (F60) instruction.



Compare CMP (F70)

The Compare (F70) is a 16-bit instruction that compares the value in a 16-bit reference, two consecutive 8-bit registers (specify starting location), or a 4-digit BCD against the value in the accumulator. Discrete bit flags are used to indicate if the result of the comparison was greater than, equal to, or less than the value in the accumulator.

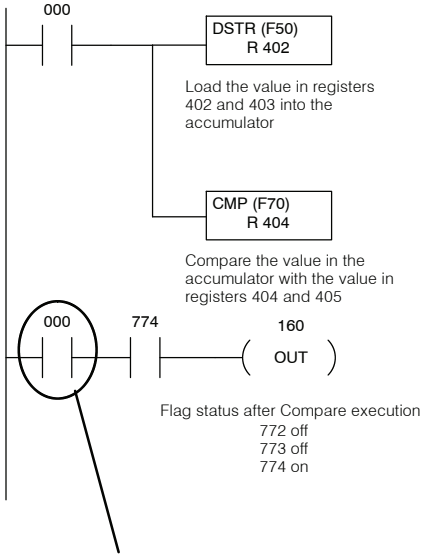


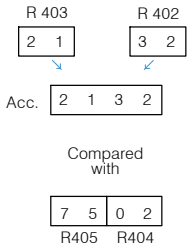
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-27
Shift Registers	R	040-056	040-56	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

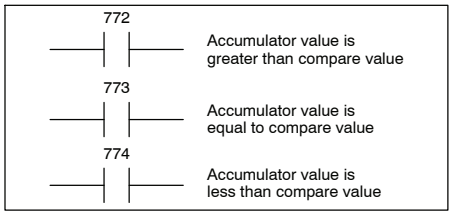
Discrete Bit Flags	Description
772	Will be on if the accumulator value is greater than the compare value
773	Will be on if the accumulator value is equal to the compare value
774	Will be on if the accumulator value is less than the compare value

In the following example, when input 000 is on the value (2132) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The data in the accumulator is compared to value in data registers R404 and R405 using the Compare (F70) instruction. Discrete status flag 774 is used to indicate if the accumulator is less than the compare value in this example.

***Direct*SOFT Display**







Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	7	0	ENT	
R	4	0	4	ENT
STR	SHF	0	ENT	
AND	SHF	7	7	4 ENT
OUT	SHF	1	6	0 ENT

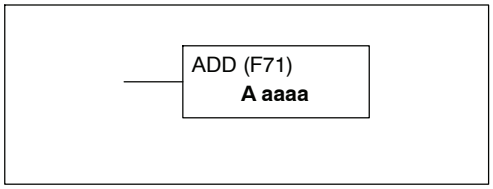
NOTE: Input 000 has been used to interlock output 160. This is done since an earlier comparison could result in status flag 774 coming on when this particular comparison is not being executed thereby providing the opportunity for an unexpected output signal on output 160.

It is a common mistake to just use the status flags without interlocking to control outputs in a program but, status flags 772 – 774 can change several times during the same scan. Just as you should not use the status flags by themselves to control outputs, you also should not monitor status flags within the program. Instead you should monitor the interlocked outputs controlled by the status flags.

Math Instructions

Add
ADD (F71)

The Add (F71) is a 16-bit instruction that adds the value of a 16 bit reference, two consecutive 8-bit registers (specify starting location), or a 4-digit BCD value with the value in the accumulator. The result resides in the accumulator. Discrete bit flags are used to indicate if the result had a carry digit or if the result was was zero.

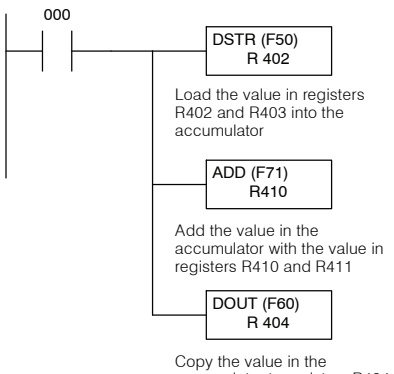


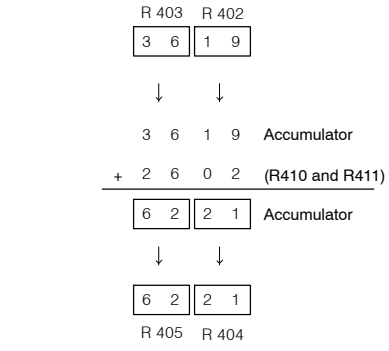
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

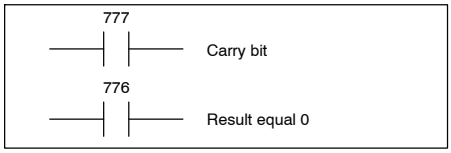
Discrete Bit Flags	Description
775	Will be on if the operation results in a carry
776	Will be on if the result is 0

In the following example, when input 000 is on the value (3619) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The Add instruction (F71) adds the value (2602) in R410 and R411 to the value in the accumulator. The result in the accumulator is then copied to data registers R404 and R405 with the Data Out (F60) instruction.

DirectSOFT Display







Handheld Programmer Keystrokes

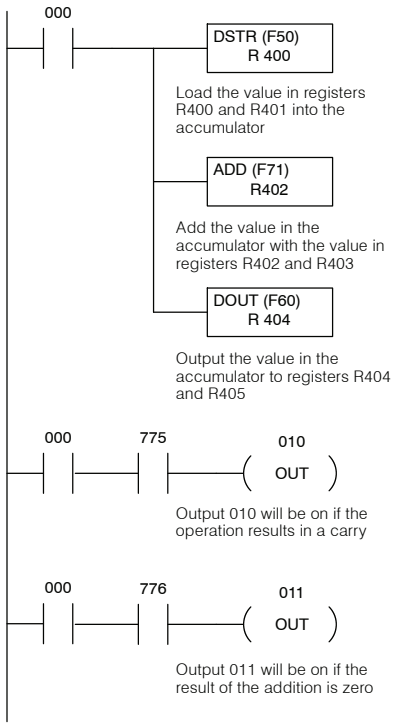
STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	7	1	ENT	
R	4	1	0	ENT
F	6	0	ENT	
R	4	0	4	ENT

Add Example

The following examples demonstrate how the discrete status flags are used to indicate if the result of the add has produced a number which exceeds the capacity of the accumulator. Remember, the accumulator has a 4 digit maximum. When a calculation produces a number larger than 4 digits, part of this number is lost. The following table shows different values being used in the logic example below. Notice how the discrete status flags change.

	Registers for DSTR Instruction R401/R400	Registers for ADD Instruction R403/R402	Registers for DOUT Instruction R405/R404	Discrete Status Flag 775	Discrete Status Flag 776
Example 1	500	400	0900	off	off
Example 2	5000	5000	0000	on	on
Example 3	5050	5000	0050	on	off

DirectSOFT Display



Handheld Programmer Keystrokes

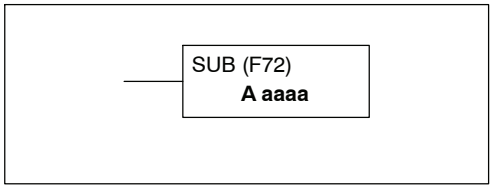
STR	SHF	0	ENT		
F	5	0	ENT		
R	4	0	0	ENT	
F	7	1	ENT		
R	4	0	2	ENT	
F	6	0	ENT		
R	4	0	4	ENT	
STR	SHF	7	7	5	ENT
OUT	SHF	0	1	0	ENT
STR	SHF	7	7	6	ENT
OUT	SHF	0	1	1	ENT

NOTE: An input has been used to interlock the outputs on the last two rungs. This is done since an earlier math instruction could result in the status flag coming on when this particular math instruction is not being executed thereby providing the opportunity for an unexpected output signal.

It is a common mistake to just use the status flags without interlocking to control outputs in a program but, the status flags can change several times during the same scan. Just as you should not use the status flags by themselves to control outputs, you also should not monitor status flags within the program. Instead you should monitor the interlocked outputs controlled by the status flags.

Subtract SUB (F72)

The Subtract (F72) is a 16-bit instruction that subtracts the value in a 16-bit register, two consecutive 8-bit registers (specify starting location), or a 4-digit BCD value from the value in the accumulator. The result resides in the accumulator. Discrete bit flags are used to indicate if the result had a borrow digit or the result was zero.

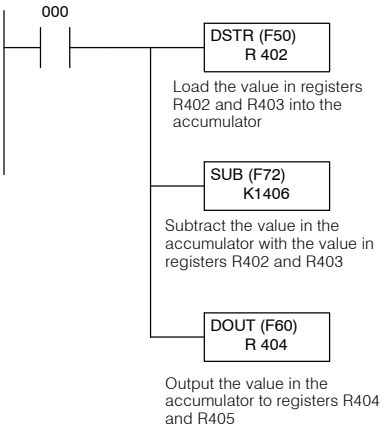


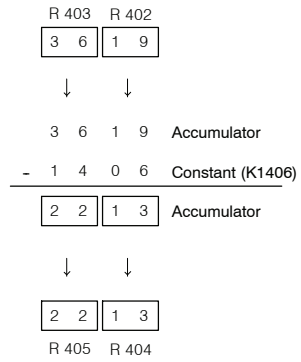
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

Discrete Bit Flags	Description
775	Will be on if the result if a borrow digit occurred
776	Will be on if the result is 0

In the following example, when input 000 is on the value (3619) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The constant value K1406 is subtracted from the value in the accumulator using the Subtract (F72) instruction. The result in the accumulator is then copied to data registers R404 and R405 using the Data Out (F60) instruction.

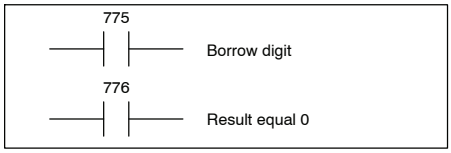
***Direct*SOFT Display**





Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	7	2	ENT	
SHF	1	4	0	6 ENT
F	6	0	ENT	
R	4	0	4	ENT



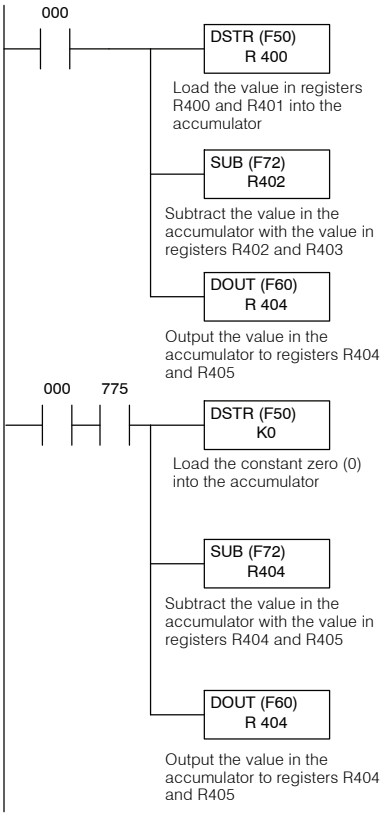
Subtract Example

The following examples demonstrate how the discrete status flags are used to indicate if the result of the Subtraction is a 0 or required a borrow digit. The following table shows different values being used in the logic example below. Notice how the discrete status flags change for each example.

	Registers for DSTR Instruction R401/R400	Registers for SUB Instruction R403/R402	Registers for DOUT Instruction R405/R404	Discrete Status Flag 775	Discrete Status Flag 776
Example 1	6050	5000	1050	off	off
Example 2	7050	7050	0000	off	on
Example 3	5000	6000	9000*	on	off

***** The DL305 cannot process negative numbers. When the number being subtracted from the accumulator is larger than the number in the accumulator, a borrow digit occurs and the subtraction is completed. The value in the accumulator does not represent the difference between the two numbers. To get the difference between the two numbers in Example 3 the result (9000) in the accumulator is subtracted from 0. The final result is 1000, the difference between 6000 and 5000.

DirectSOFT Display



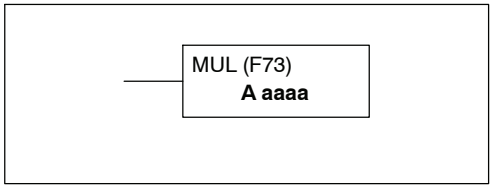
Handheld Programmer Keystrokes

STR	SHF	0	ENT		
F	5	0	ENT		
R	4	0	0	ENT	
F	7	2	ENT		
R	4	0	2	ENT	
F	6	0	ENT		
R	4	0	4	ENT	
STR	SHF	0	ENT		
AND	SHF	7	7	5	ENT
F	5	0	ENT		
R	4	0	4		
F	7	2	ENT		
K	0	ENT			
F	6	0	ENT		
R	4	0	4	ENT	

NOTE: It is a common mistake to just use the status flags without interlocking to control outputs in a program, but the status flags can change several times during the same scan. Just as you should not use the status flags by themselves to control outputs, you also should not monitor status flags within the program. Instead you should monitor the interlocked outputs controlled by the status flags.

Multiply MUL (F73)

The Multiply (F73) is a 16-bit instruction that multiplies the value in a 16-bit register, two consecutive 8-bit registers, or a 4-digit BCD value by the value in the accumulator. The least significant four digits of the result are stored in the accumulator and the most significant four digits are stored in the auxiliary accumulator (R575 and R577). A discrete bit flag is used to indicate if the result was zero.

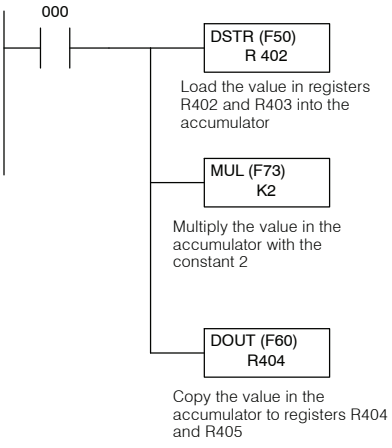


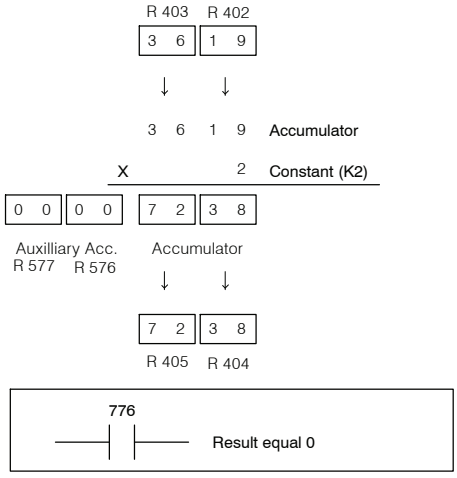
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

Discrete Bit Flags	Description
776	Will be on if the result is 0

In the following example, when input 000 is on the value (3619) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The data in the accumulator is multiplied with the constant K2 with the result residing in the accumulator and auxiliary accumulator (R576 and R577) using the Multiply (F73) instruction. The value in the accumulator is output to data registers R404 and R405 using the Data Out (F60) instruction.

DirectSOFT Display





Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	7	3	ENT	
SHF	2	ENT		
F	6	0	ENT	
R	4	0	4	ENT

Multiply Example

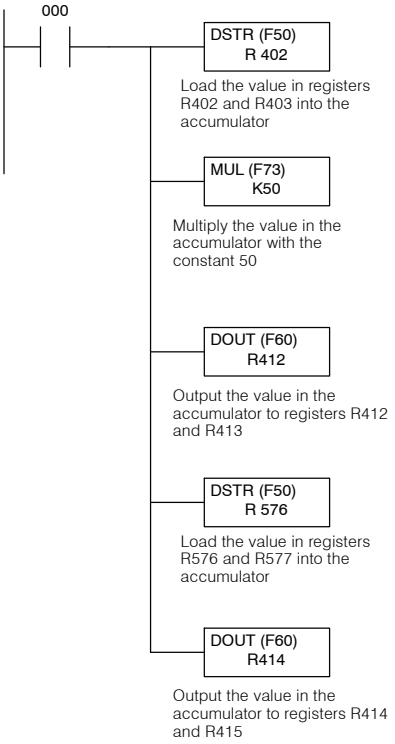
The multiply instruction allows you to multiply two 4-digit numbers together. The result is located in the accumulator and the auxiliary accumulator (R576 and R577) when necessary. The accumulator holds the lower 4 digits of the result and the auxiliary accumulator holds the upper 4 digits.

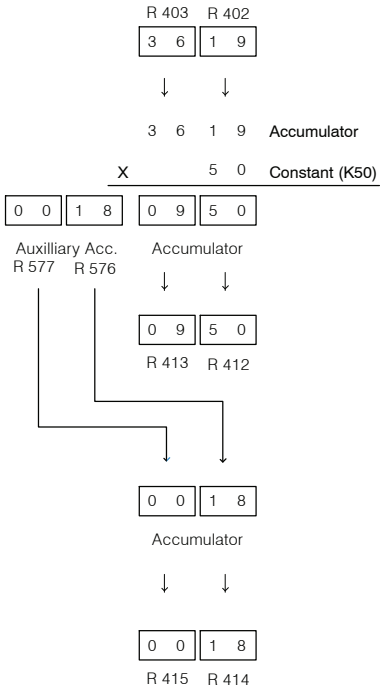
Whenever possible multiplications resulting in more than 4 digits should be avoided since the DL305 instruction set can only manipulate a maximum of two consecutive 8-bit registers (4 digits) at one time.

If the result of a multiplication is greater than 4 digits, the application program must be written to compensate for the instruction set 4 digit maximum for data manipulation. The example below shows how the auxiliary accumulator is used to store a result with more than 4 digits and how to access the upper 4 digits.

The example below shows how the auxiliary accumulator is used to process numbers larger than 4 digits when the multiplication instruction is used.

DirectSOFT Display



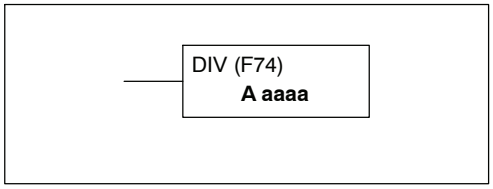


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	7	3	ENT	
SHF	5	0	ENT	
F	6	0	ENT	
R	4	1	2	ENT
F	5	0	ENT	
R	5	7	6	ENT
F	6	0	ENT	
R	4	1	4	ENT

Divide DIV (F74)

The Divide (F74) is a 16-bit instruction that divides the value in the accumulator by the value in a 16-bit register, two consecutive 8-bit registers, or a 4-digit BCD value. The integer portion of the result is stored in the accumulator and the decimal fraction is stored in the auxiliary accumulator, R576 and R577. Discrete flags are used to indicate if the dividend or divisor is zero or if only the divisor is zero.

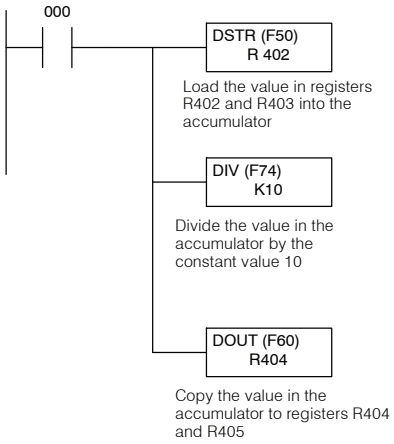


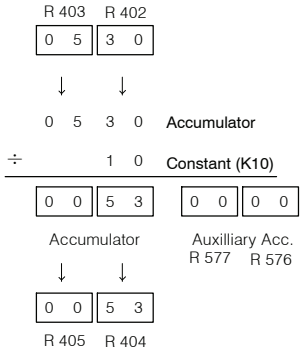
Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

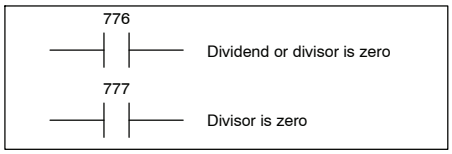
Discrete Bit Flags	Description
776	Will be on if the dividend or divisor is zero
777	Will be on if the divisor is zero

In the following example, when input 000 is on the value (530) in R402 and R403 is loaded into the accumulator using the Data Store (F50) instruction. The data in the accumulator is divided by the constant 10 (K10). The result in the accumulator and is copied to data registers R404 and R405 using the Data Out (F60) instruction. The remainder is in the auxiliary accumulator (R576 and R577).

DirectSOFT Display







Handheld Programmer Keystrokes

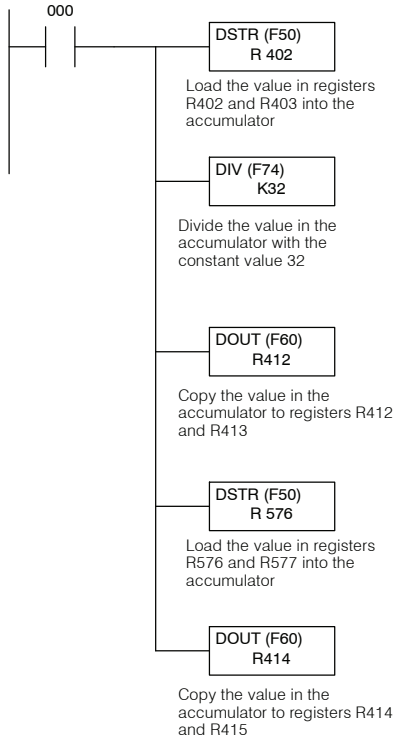
STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	7	4	ENT	
SHF	1	0	ENT	
F	6	0	ENT	
R	4	0	4	ENT

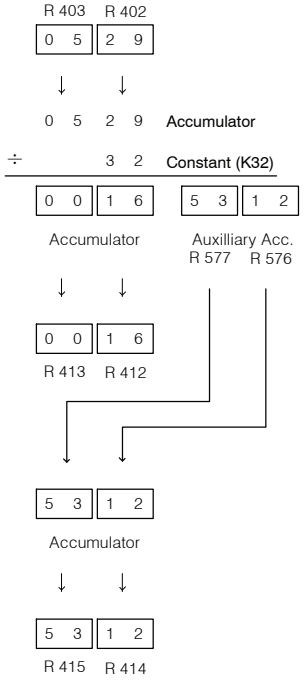
Divide Example

The divide instruction allows you to divide the value in the accumulator by 4 digits maximum. The divide instruction uses the accumulator for the integer value of the result and the auxiliary accumulator (R576 and R577) for fraction. The instruction set only allows manipulation on two consecutive registers at a time. For example, if the result was a 4 digit number with a remainder it would have to be treated like two 4-digit numbers in the program. Manipulating numbers over 4 digits should be avoided whenever possible. If it cannot be avoided the application program must be written to compensate for the 4-digit maximum for data manipulation.

The example below shows how the auxiliary accumulator is used to store the fractional portion of the result and how to access the remainder.

***Direct*SOFT Display**





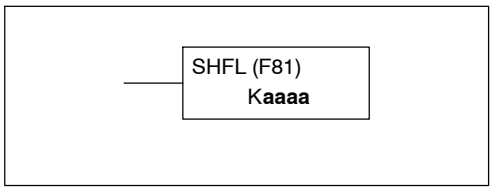
Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	0	ENT	
R	4	0	2	ENT
F	7	4	ENT	
SHF	3	2	ENT	
F	6	0	ENT	
R	4	1	2	ENT
F	5	0	ENT	
R	5	7	6	ENT
F	6	0	ENT	
R	4	1	4	ENT

Bit Operation Instructions

Shift Left SHFL (F81)

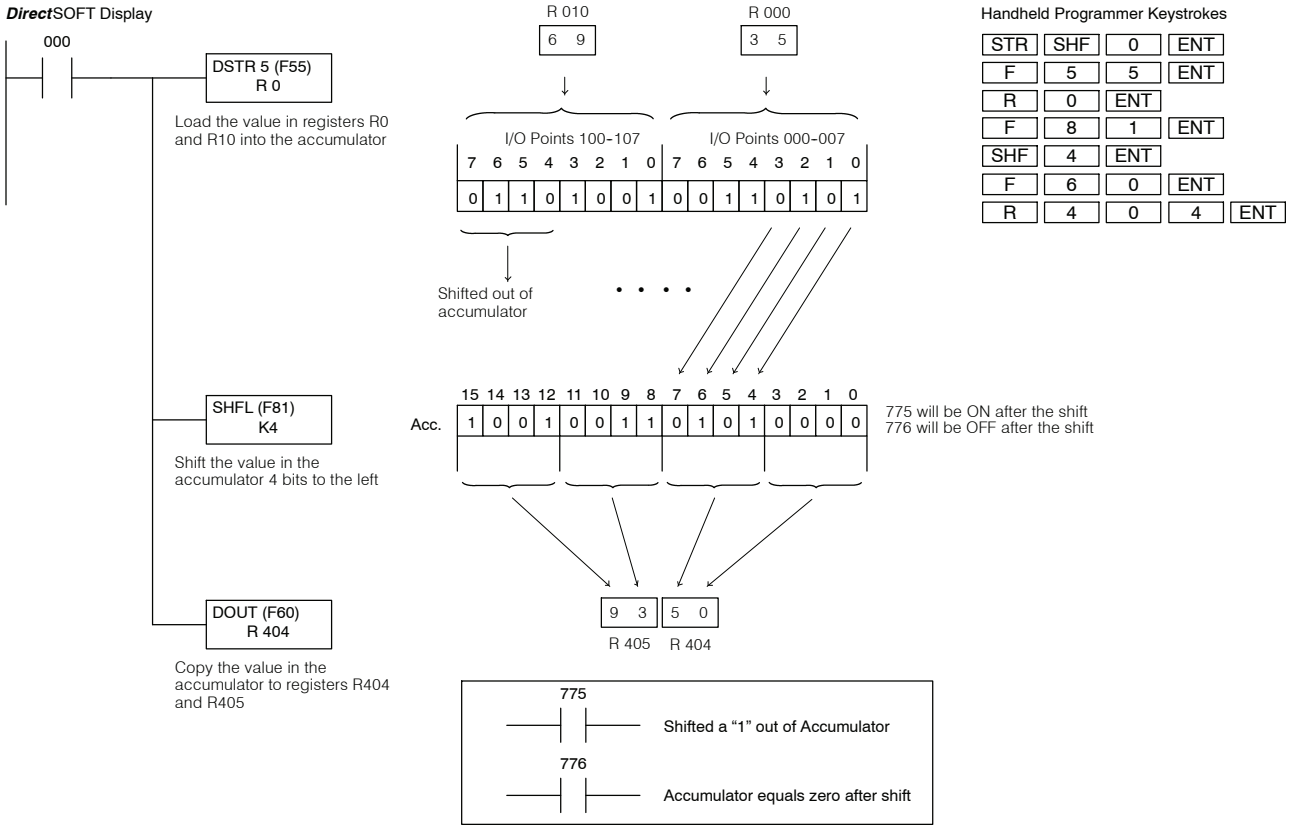
The Shift Left (F81) is a 16-bit instruction that shifts the value in the accumulator a specified number of bits (15 maximum) to the left. Discrete bit flags are used to indicate if a "1" was shifted out of the accumulator or if the accumulator equals "0" after the shift.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	**aaaa**	**aaaa**
Constant (4-digit BCD) K	1-16	1-16	1-16

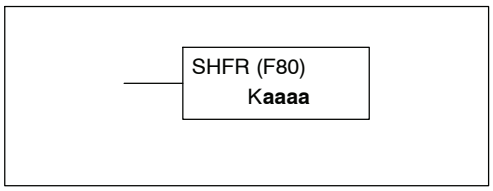
Discrete Bit Flags	Description
775	Will be on if a"1" was shifted out of the accumulator.
776	Will be on if the accumulator equals zero after the shift instruction is executed.

In the following example, when input 000 is on the value in R000 and R010 is loaded into the accumulator using the Data Store 5 (F55) instruction. The bit pattern in the accumulator is shifted to the left 4 bit positions using the Shift Left (F81) instruction with the result resides in the accumulator. The value in the accumulator is copied to data registers R404 and R405 using the Data Out (F60) instruction.



Shift Right SHFR (F80)

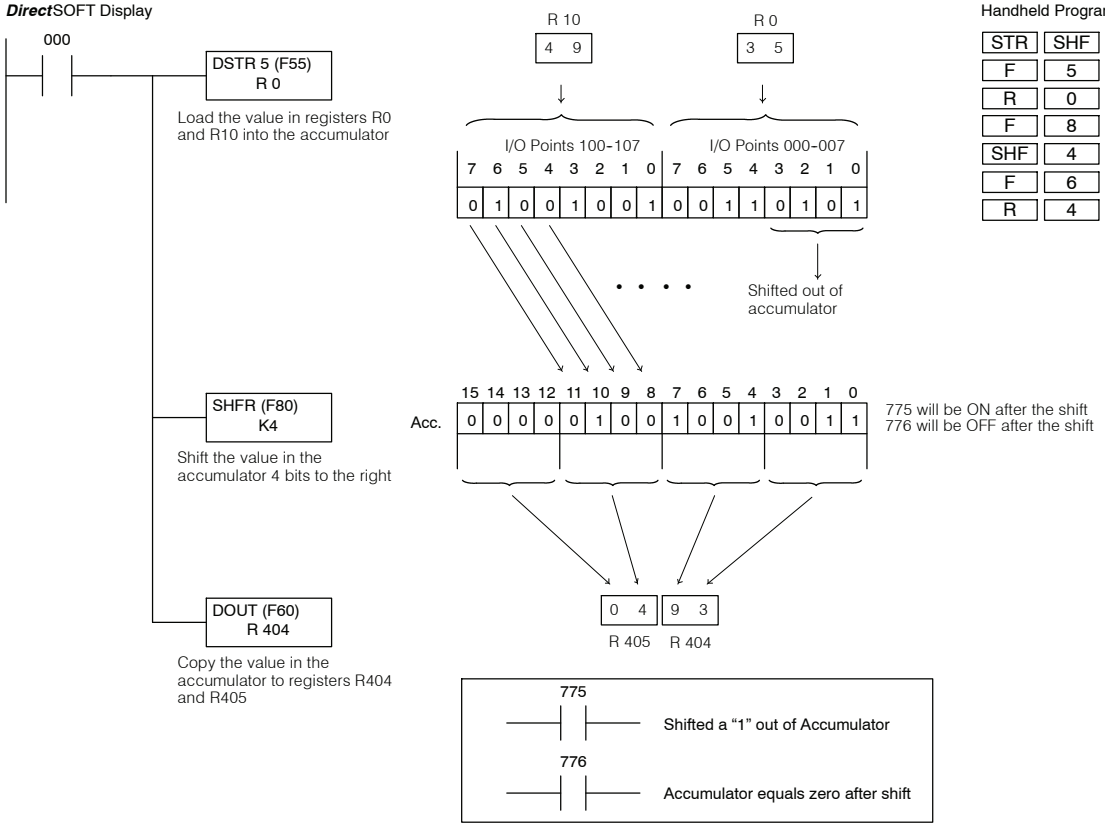
The Shift Right (F80) is a 16-bit instruction that shifts the value in the accumulator a specified number of bits (15 maximum) to the right. Discrete bit flags are used to indicate if a "1" was shifted out of the accumulator or if the accumulator equals "0" after the shift.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	**aaaa**	**aaaa**
Constant (4-digit BCD) K	1-16	1-16	1-16

Discrete Bit Flags	Description
775	Will be on if a"1" was shifted out of the accumulator.
776	Will be on if the accumulator equals zero after the shift instruction is executed.

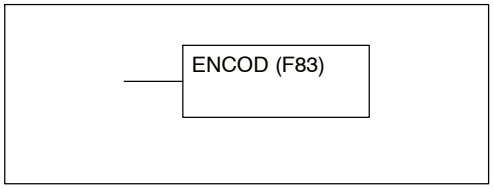
In the following example, when input 000 is on the value in R000 and R010 is loaded into the accumulator using the Data Store 5 (F55) instruction. The bit pattern in the accumulator is shifted 4 bit positions using the Shift Right (F80) instruction and the result resides in the accumulator. The value in the accumulator is copied to data registers R404 and R405 using the Data Out (F60) instruction.



Number Conversion Instructions

Encode
ENCOD (F83)

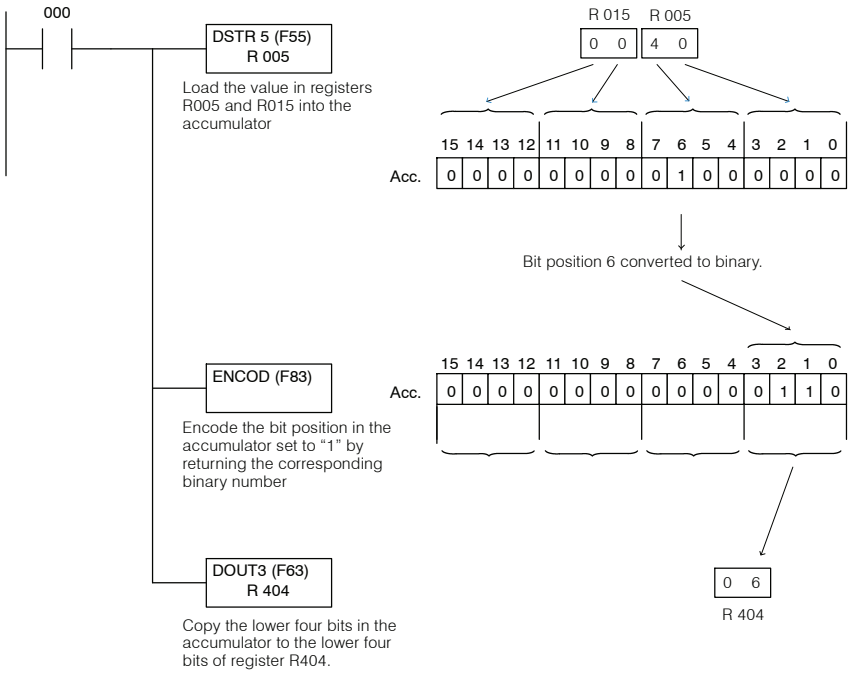
The Encode instruction encodes the accumulator bit position that contains a 1 by returning the corresponding binary representation. If the most significant bit is set to HEX F (decimal 15), the binary value 15 is returned to the accumulator. If the accumulator value is 0000 or 0001 a zero will be returned to the accumulator. If there is more than one bit position set to a "1" the least significant "1" will be encoded. The discrete bit flag 777 is used to indicate if there were multiple 1s in the accumulator.



Discrete Bit Flags	Description
777	Will be on if there was more than one bit position set to a "1" in the accumulator.

In the following example, when input 000 is on the 16-bit binary pattern from registers R005 and R015 is loaded in the accumulator by the Data Store 5 (F55) instruction. In this example the 6th bit (BCD 40) is on. When the Encode (F83) instruction executes the accumulator will contain the BCD value of 6. The lower four bits of the accumulator are copied to the lower four bits of register R404 by the Data Out 3 (F63) instruction.

DirectSOFT Display

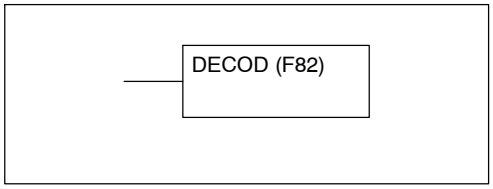


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	5	ENT	
R	5	ENT		
F	8	3	ENT	
F	6	3	ENT	
R	4	0	4	ENT

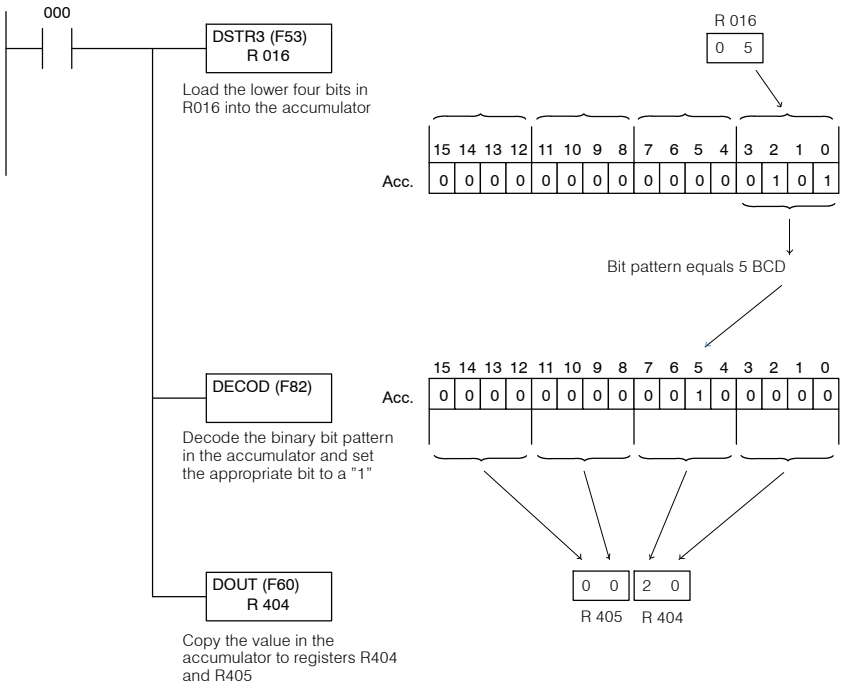
Decode DECOD (F82)

The Decode instruction decodes a four bit binary number (0–F) in the accumulator and sets the corresponding bit position to a one. If the accumulator contains a HEX F (decimal 15) the most significant bit (bit 15) will be set in the accumulator. If the accumulator contains a zero the least significant bit (bit 0) will be set to a one. All other bits in the accumulator will be set to a zero.



In the following example, when 000 is on the binary value of the lower four bits in R016 (5) will be loaded in the accumulator by the Data Store 3 (F53) instruction. The Decode instruction will then translate the value 5 to a 1 in the fifth bit position of the accumulator. The value 20 in the accumulator is copied to data registers R404 and R405 with the Data Out (F60) instruction.

DirectSOFT Display

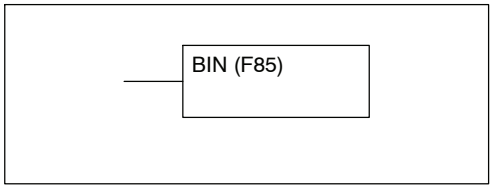


Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	5	3	ENT	
R	1	6	ENT	
F	8	2	ENT	
F	6	0	ENT	
R	4	0	4	ENT

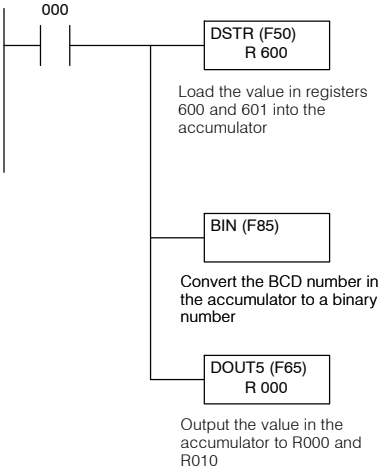
Binary BIN (F85)

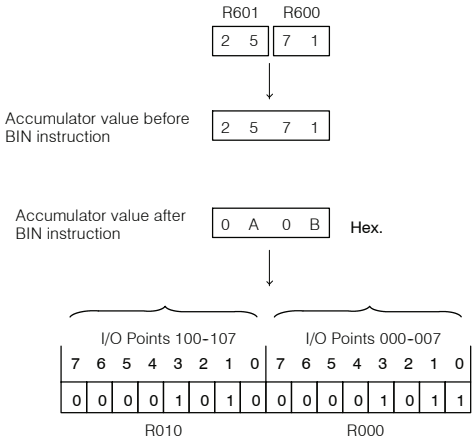
The Binary (F85) instruction converts a BCD value in the accumulator to the binary/HEX equivalent value. The result of the conversion resides in the accumulator.



In the following example, when input 000 is on the value (2571 BCD) in R600 is loaded into the accumulator using the Data Store (F50) instruction. The value in the accumulator is converted to a binary number (HEX 0A0B) using the Binary (F85) instruction with the result residing in the accumulator. The value in the accumulator is copied to I/O registers R000 and R010 (which corresponds to I/O points 0–7 and 100–107) with the Data Out 5 (F65) instruction.

DirectSOFT Display



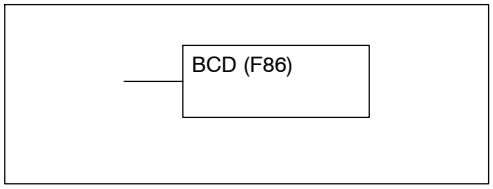


Handheld Programmer Keystrokes

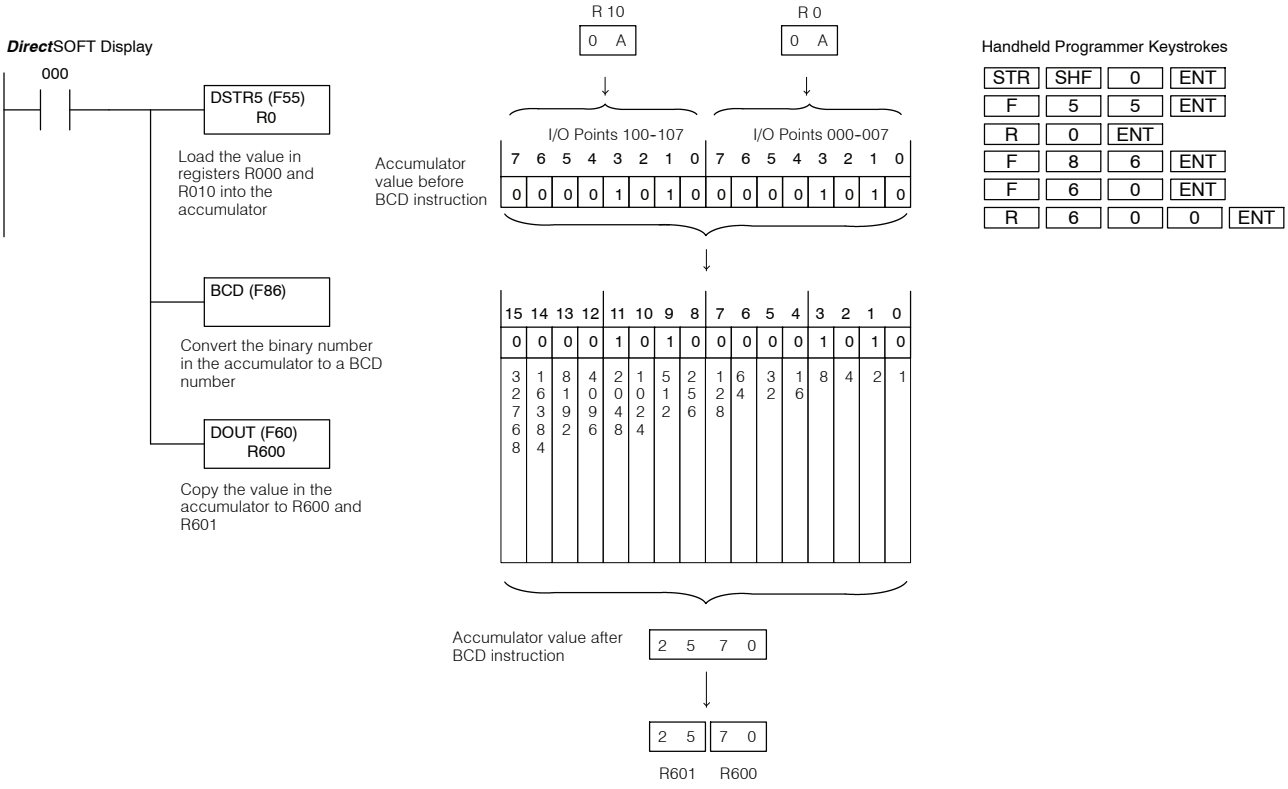
STR	SHF	0	ENT	
F	5	0	ENT	
R	6	0	0	ENT
F	8	5	ENT	
F	6	5	ENT	
R	0	ENT		

Binary Coded Decimal BCD (F86)

The Binary Coded Decimal (F86) instruction converts a binary/HEX value in the accumulator to the BCD equivalent. The result of the conversion resides in the accumulator.

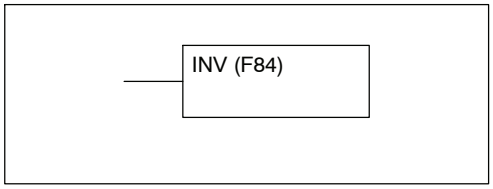


In the following example, when input 000 is on the value (HEX 0A0A) in R000 and R010 is loaded into the accumulator with the Data Store 5 (F55) instruction. The value in the accumulator is converted to a BCD number (BCD 2570) using the BCD (F86) instruction with the result residing in the accumulator. The value in the accumulator is output to register R600 using the Data Out (F60) instruction.

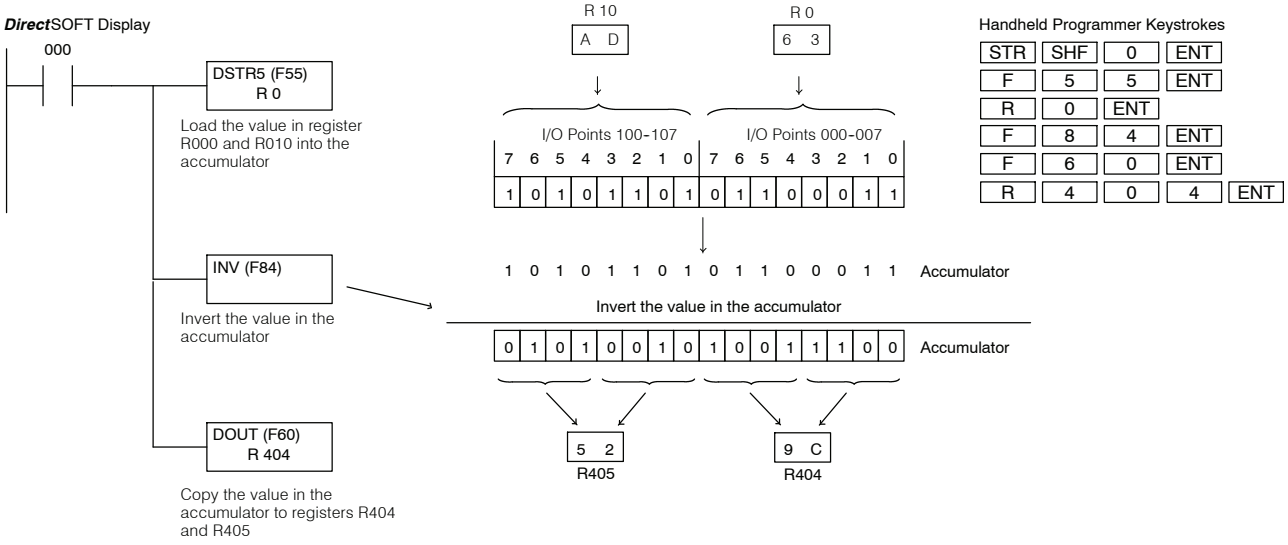


Invert INV (F84)

The Invert instruction generates the one's complement of the number in the accumulator. The result is stored in the accumulator.



In the following example, when input 000 is on the value (AD63) in R000 and R010 is loaded into the accumulator using the Data Store (F55) instruction. The value in the accumulator is inverted with the result residing in the accumulator. The value (HEX 529C) is copied to registers R404 and R405 using the Data Out (F60) instruction.

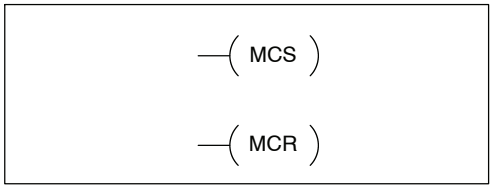


Program Control Instructions

Master Control Set (MCS) DL330/DL340 Only

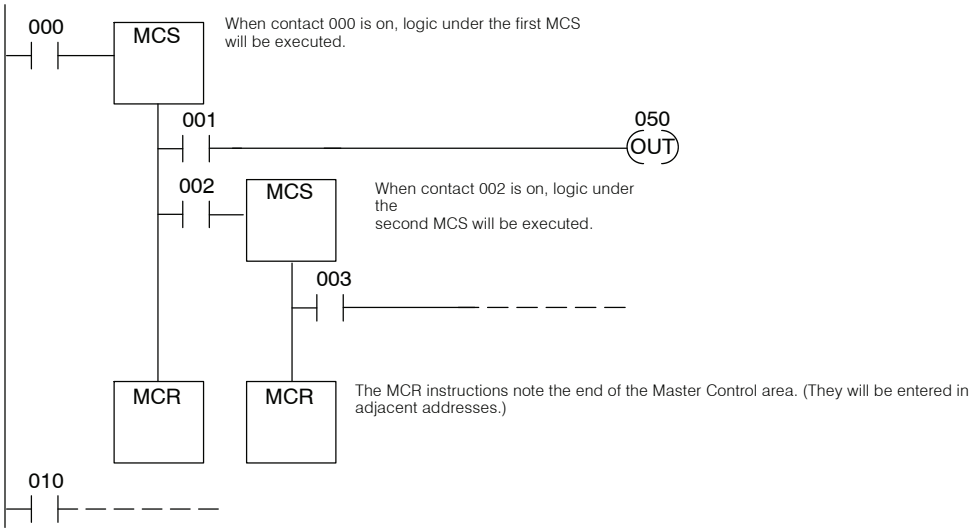
Master Control Reset (MCR) DL330/DL340 only

The Master Control Set and Master Control Reset instructions are used to provide an additional left power rail which is controllable by a input contact. This is sometimes known as a sub power rail. Any number of rungs of ladder logic can be disabled using these instructions.



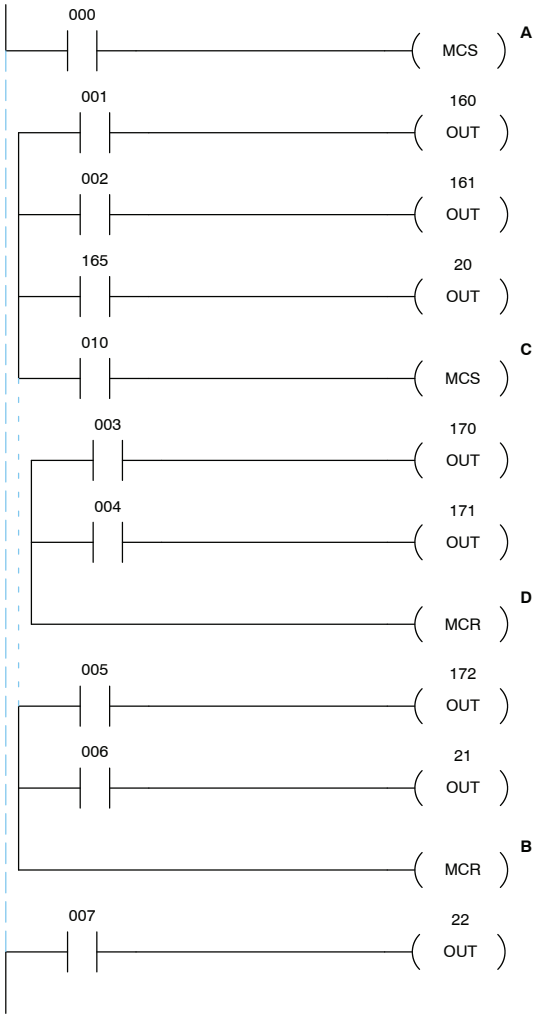
Understanding Master Control Relays

The Master Control Set (MCS) and Master Control Reset (MCR) instructions allow you to quickly enable (or disable) sections of the RLL program. This provides program control flexibility. The following example shows how the MCS and MCR instructions operate by creating a sub power rail for control logic.



MCS/MCR Example In the following MCS/MCR example logic between the first MCS (A) and the last MCR (B) will function only if input 000 is on. The logic between the second MCS (C) and the next to last MCR (D) will function only if input 010 is on. The last rung is not controlled by either of the MCS coils.

DirectSOFT Display



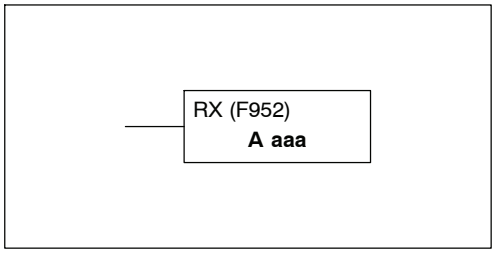
Handheld Programmer Keystrokes

STR	SHF	0	ENT		
MCS	ENT				
STR	SHF	1	ENT		
OUT	SHF	1	6	0	ENT
STR	SHF	2	ENT		
OUT	SHF	1	6	1	ENT
STR	SHF	1	6	5	ENT
OUT	SHF	2	0	ENT	
STR	SHF	1	0	ENT	
MCS	ENT				
STR	SHF	3	ENT		
OUT	SHF	1	7	0	ENT
STR	SHF	4	ENT		
OUT	SHF	1	7	1	ENT
MCR	ENT				
STR	SHF	5	ENT		
OUT	SHF	1	7	2	ENT
STR	SHF	6	ENT		
OUT	SHF	2	1	ENT	
MCR	ENT				
STR	SHF	7	ENT		
OUT	SHF	2	2	ENT	

NOTE: When programming the MCS instruction, do not put any parallel coils in the rung with the MCS.

Network Instructions

Read from Network RX (F952) DL340 Only



The Read from Network instruction is used by the master device on a ***Direct***NET network to read a block of data from another CPU or ***Direct***NET interface module. The function parameters are loaded into the accumulator and the first and second level of the accumulator stack by three additional instructions. Listed below are the steps necessary to program the Read from Network function.

Step 1: — Load the slave address (1-90 BCD) into the accumulator with the DSTR instruction. This must always be preceded by 00, so address 20 would be 0020. (Remember, D4-DCM slave device addresses are set with switches that use a hexadecimal format. Make sure you convert this address to the decimal equivalent for use with this instruction.)

Step 2: — Load the number of bytes (1 - 128 BCD) to be transferred from the network slave station.

Step 3: — Load the octal address for the data register that will be used to store the data obtained from the slave station.

Step 4: — Insert the RX instruction which specifies the starting address in the slave station. If you are getting the data from a DL305 station, just enter the Data Register number. If you are getting the data from a DL205 or DL405 station, enter a constant that corresponds to the memory address. For example, to get the current count for Timer 600 from a DL305 CPU, you would use R600 with the RX instruction. If you wanted to get Counter 0 from a DL405 CPU, you would use a constant of 1000 with the RX instruction. (V1000 stores the current count for Counter 0 in a DL405 CPU.)

NOTE: The ***Direct***NET manual provides further information on the use of the RX and WX network instructions.

Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
*Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

* A constant is used to obtain data from a DL205 or DL405 system.

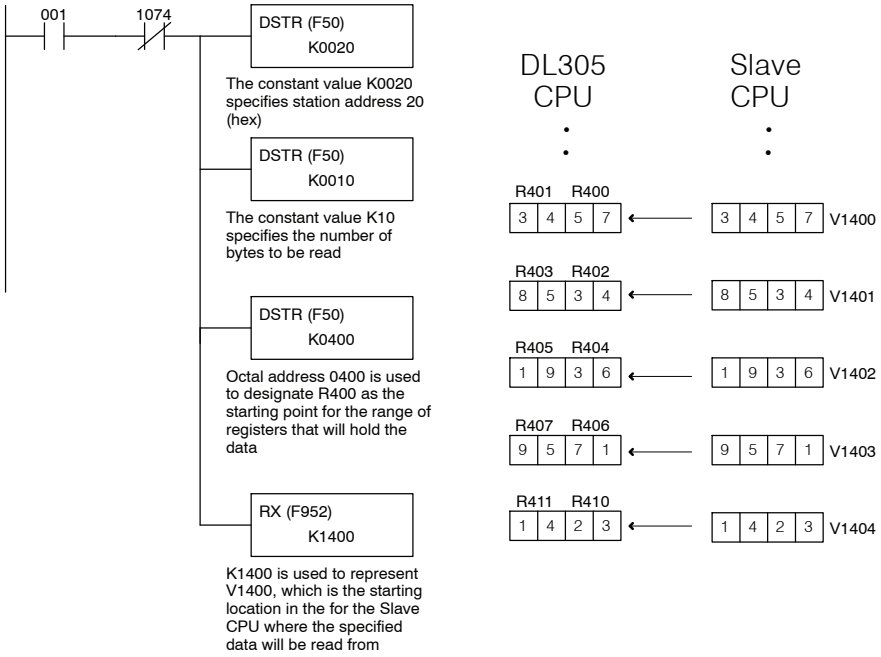
Discrete Bit Flags	Description
777	Parameters are not properly set. Check the slave address, data length, or data address reference.
1074	Communication port busy.
1075	Communication error. Data was not transmitted.

NOTE: See the DL205 or DL405 User's Manuals for a listing of V-memory addresses available with these CPUs. Since the DL305 only supports a 4-digit constant, you will not be able to access the entire V-memory ranges of the DL205 and DL405 CPUs. For example, you could not directly access V40400 stored in a DL405 CPU. If you require data from a range outside the area available with a 4-digit constant (from V0 – V9999) then add a routine to the slave station program that moves this data down into one of the accessible areas.

In the following example, when input 001 is on and the CPU busy relay 1074 (see special relays, p. 8–32) is not on, the RX instruction will access a DL405 CPU that has been assigned station address 20. (Note, the D4–DCM slave station addresses are set with switches that indicate a hexadecimal number. Make sure you determine the decimal equivalent to be used with the first DSTR instruction in the sequence.)

Ten consecutive bytes of data (V1400 – V1404) will be read from the slave station and stored in registers R400 – R411. (Remember, the DL205 and DL405 V-memory locations are 16 bits. The DL305 registers are only 8 bits, so you have to use two data registers for each V-memory location.)

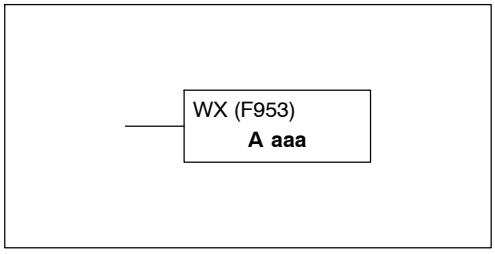
DirectSOFT Display



Handheld Programmer Keystrokes

STR	SHF	1	ENT				
AND	NOT	SHF	1	0	7	4	ENT
F	5	0	ENT				
SHF	0	0	2	0	ENT		
F	5	0	ENT				
SHF	1	0	ENT				
F	5	0	ENT				
SHF	4	0	0	ENT			
F	9	5	2	ENT			
SHF	1	4	0	0	ENT		

Write to Network WX (F953) DL340 Only



The Write to Network instruction is used by the master device on a ***Direct***NET network to write a block of data to another station. The function parameters are loaded into the accumulator and the first and second level of the accumulator stack by three additional instructions. Listed below are the steps necessary to program the Write to Network function.

Step 1: — Load the slave address (1-90 BCD) into the accumulator with the DSTR instruction. This must always be preceded by 00, so address 20 would be 0020. (Remember, the D4-DCM slave device addresses are set with switches that use a hexadecimal format. Make sure you convert this address to the decimal equivalent for use with this instruction.)

Step 2: — Load the number of bytes (1 - 128 BCD) to be transferred to the network slave station.

Step 3: — Load the octal address for the data register that will be used to obtain the data that will be sent to the slave station.

Step 4: — Insert the WX instruction which specifies the starting address in the slave station. If you are sending data to a DL305 station, just enter the Data Register number. If you are sending data to a DL205 or DL405 station, enter a constant that corresponds to the memory address. For example, to send data to Register 500 in a DL305 CPU, you would use R500 with the RX instruction. If you wanted to send data to V1400 in a DL405 CPU, you would use a constant of 1400 with the WX instruction.

NOTE: The ***Direct***NET manual provides further information on the use of the RX and WX network instructions.

Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-027
Shift Registers	R	040-056	040-056	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
*Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

* A constant is used to send data to a DL205 or DL405 system.

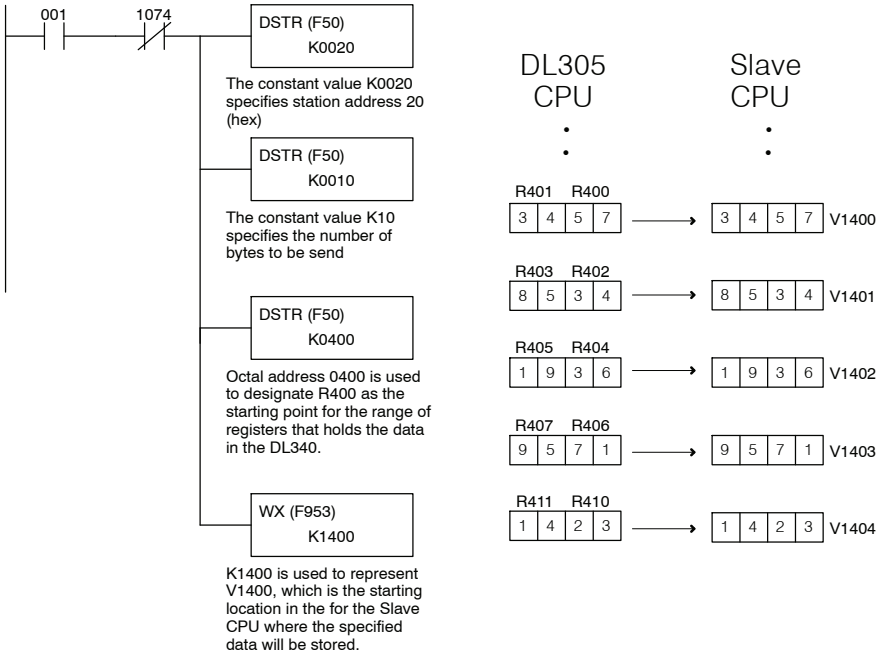
Discrete Bit Flags	Description
777	Parameters are not properly set. Check the slave address, data length, or data address reference.
1074	Communication port busy.
1075	Communication error. Data was not transmitted.

NOTE: See the DL205 or DL405 User's Manuals for a listing of V-memory addresses available with these CPUs. Since the DL305 only supports a 4-digit constant, you will not be able to access the entire V-memory ranges of the DL205 and DL405 CPUs. For example, you could not directly access V40400 stored in a DL405 CPU. If you want to send data to a range outside the area available with a 4-digit constant (from V0 – V9999) then add a routine to the slave station program that moves the data from one of the accessible areas into the unavailable locations.

In the following example, when input 001 is on and the CPU busy relay 1074 (see special relays, p. 8–32) is not on, the WX instruction will access a DL405 CPU that has been assigned station address 20. (Note, the D4–DCM slave station addresses are set with switches that indicate a hexadecimal number. Make sure you determine the decimal equivalent to be used with the first DSTR instruction in the sequence.)

Ten consecutive bytes of data (R400 – R411) will be sent from the DL340 and stored in V-memory locations V1400 – V1404. (Remember, the DL205 and DL405 V-memory locations are 16 bits. The DL305 registers are only 8 bits, so you have to use two data registers for each V-memory location.)

DirectSOFT Display



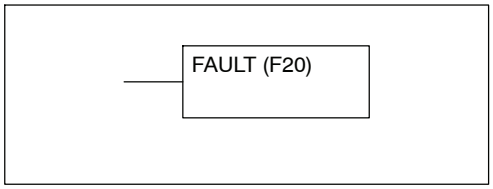
Handheld Programmer Keystrokes

STR	SHF	1	ENT				
AND	NOT	SHF	1	0	7	4	ENT
F	5	0	ENT				
SHF	0	0	2	0	ENT		
F	5	0	ENT				
SHF	1	0	ENT				
F	5	0	ENT				
SHF	4	0	0	ENT			
F	9	5	3	ENT			
SHF	1	4	0	0	ENT		

Message Instructions

Fault
FAULT (F20)

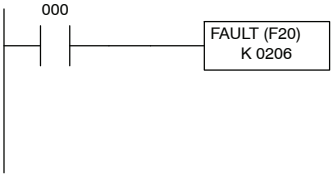
The Fault (F20) instruction is used to display a 4-digit BCD constant, 16-bit register, or two consecutive 8-bit data registers on the handheld programmer or in ***Direct***SOFT. When the fault instruction is executed the number being displayed will also be copied into the registers R574 and R575 and the discrete bit flag 771 will be on.



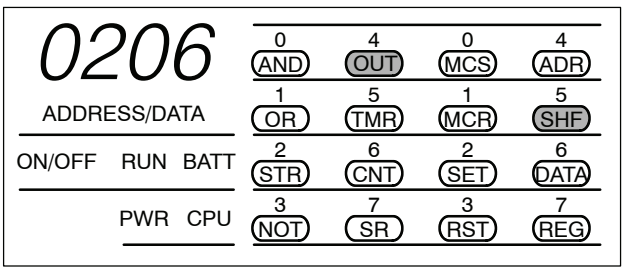
In the following example, when input 000 is on the number 0206 will be displayed on the programming device. This would typically be a user defined error code.

Data Type		D3-330 Range	D3-340 Range	D3-330P Range
	A	**aaaa**	**aaaa**	**aaaa**
Inputs / Outputs	R	000-014 070-075	000-014 070-075	000-014 070-075
Control Relays	R	016-036	016-036 100-105	016, 020-27
Shift Registers	R	040-056	040-56	--
Stages	R	--	--	100-116
Timer /Counters (16 bit)	R	600-677	600-677	600-677
Data Registers	R	400-577	400-577 700-777	400-577
Constant (4-digit BCD)	K	0000-9999	0000-9999	0000-9999

DirectSOFT Display



Handheld Programmer Display



Handheld Programmer Keystrokes

STR	SHF	0	ENT	
F	2	0	ENT	
SHF	2	0	6	ENT