

RLL^{PLUS}

Instruction Set

In This Chapter. . . .

- Introduction
- Stage Instructions
- Comparative Boolean Instructions
- Timer, Counter, and Shift Register Instructions

Introduction

This chapter provides information concerning the instructions used with RLL^{PLUS} CPUs. If you are not familiar with RLL^{PLUS} programming concepts, you should read Chapter 10 first. Chapter 10 will help you understand the basic concepts. 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 ISG is the mnemonic for Initial Stage. Each instruction definition will show in parentheses the keystrokes used to enter the instruction.

NOTE: Don't assume that the instructions in this chapter are the only ones you can use with your RLL^{PLUS} CPU. There are many others that are discussed in Chapter 11 that you can use as well. If you are using an RLL^{PLUS} CPU, such as the DL330P, then you should always consult this chapter before you use one of the instructions shown in Chapter 11. There may be differences in the way the instruction operates in an RLL^{PLUS} CPU.

This chapter provides a description of several instructions that are similar, but slightly different from their RLL CPU counterparts. For example, you'll notice that a Counter instruction has two input lines in an RLL CPU but only one input line in an RLL^{PLUS} CPU.

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

- If you know the instruction category (Stage, 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.

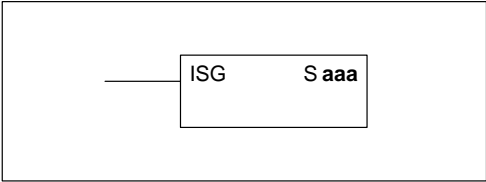
Instruction	Page
AND CNT	12-17
AND SG	12-9
AND TMR	12-16
ANDN CNT	12-17
ANDN SG	12-9
ANDN TMR	12-16
CNT	12-19
ISG	12-3
JMP	12-5
NJMP	12-5
OR CNT	12-15
OR SG	12-8
OR TMR	12-14
ORN CNT	12-15
ORN SG	12-8

Instruction	Page
ORN TMR	12-14
RST	12-10
RST (counter)	12-20
RST SG	12-11
SET	12-10
SET SG	12-11
SG	12-3
SR	12-21
STR CNT	12-13
STR TMR	12-12
STR SG	12-7
STRN CNT	12-13
STRN SG	12-7
STRN TMR	12-12
TMR	12-18

Stage Instructions

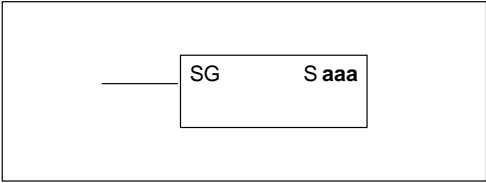
Initial Stage
(ISG)
DL330P Only

The Initial Stage instruction is normally used as the first segment of a RLL^{PLUS} program. Initial stages are activated when the CPU enters the run mode, this creates a starting point in the program. The Initial Stage can be made inactive by either jumping from it or resetting it. Multiple Initial Stages are allowed in a program.



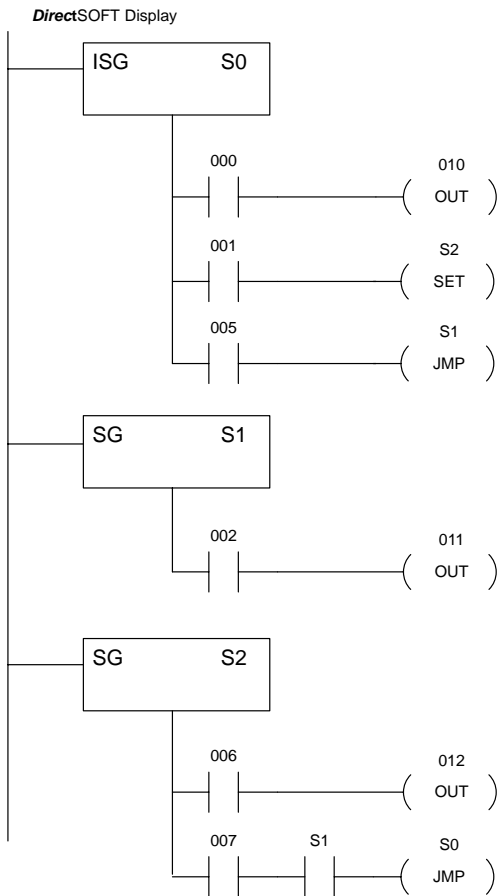
Stage
(SG)
DL330P Only

The Stage instruction creates segments of a RLL^{PLUS} program. Stages are activated by transitional logic, a jump or set stage executed from an active stage. Stages are de-activated one scan after transitional logic, a jump, or a reset stage instruction is executed.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Stages SG	—	—	0-177

The following example is a simple RLL^{PLUS} program. This program utilizes the Initial Stage, and Jump instructions to create a structured program.



Handheld Programmer Keystrokes

ISG	SHF	0	ENT
STR	SHF	0	ENT
OUT	SHF	1	0 ENT
STR	SHF	1	ENT
SET	SG	SHF	2 ENT
STR	SHF	5	ENT
JMP	SG	1	ENT
SG	SHF	1	ENT
STR	SHF	2	ENT
OUT	SHF	1	1 ENT
SG	SHF	2	ENT
STR	SHF	6	ENT
OUT	SHF	1	2 ENT
STR	SHF	7	ENT
AND	SG	1	ENT
JMP	SG	0	ENT

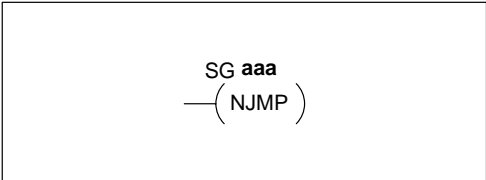
Jump
(JMP)
DL330P Only

The Jump instruction allows the program to transition from an active stage which contains the jump instruction to another which is specified in the instruction. The jump will occur when the input logic is true. The active stage that contains the Jump will be de-activated 1 scan after the Jump instruction is executed.



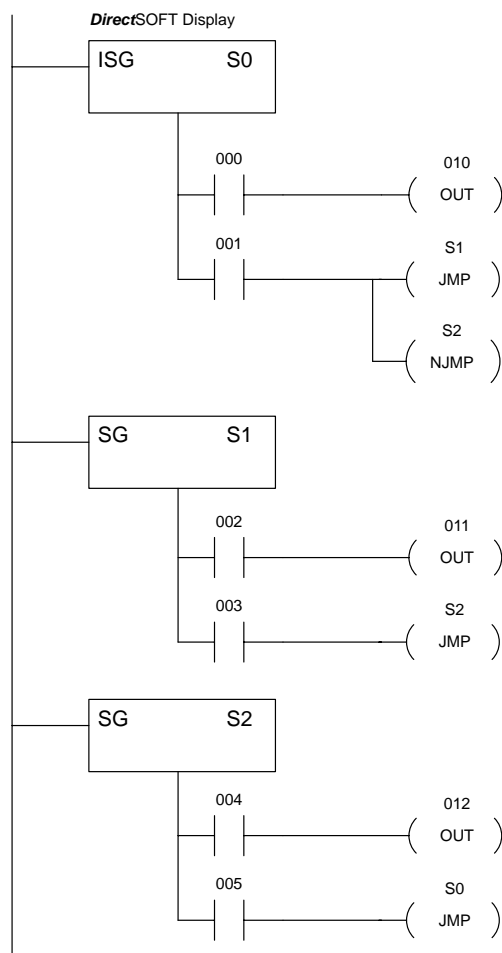
Not Jump
(NOT JMP)
DL330P Only

The Not Jump instruction allows the program to transition from an active stage which contains the jump instruction to another which is specified in the instruction. The jump will occur when the input logic is false. The active stage that contains the Not Jump will be de-activated 1 scan after the Not Jump instruction is executed.



Data Type		D3-330 Range	D3-340 Range	D3-330P Range
		aaaa	aaaa	aaaa
Stages	SG	—	—	0-177

The following example is a simple RLL^{PLUS} program. This program utilizes the Initial Stage, Stage, Jump, and Not Jump instructions to create a structured program.

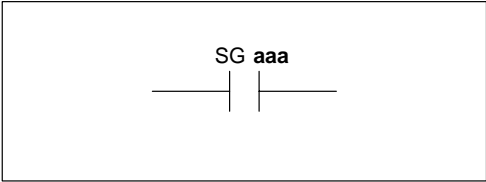


Handheld Programmer Keystrokes

ISG	SG	SHF	0	ENT
STR	SHF	0	ENT	
OUT	SHF	1	0	ENT
STR	SHF	1	ENT	
JMP	SG	SHF	1	ENT
JMP	NOT	SG	SHF	2
SG	SHF	1	ENT	
STR	SHF	002	ENT	
OUT	SHF	1	1	ENT
STR	SHF	3	ENT	
JMP	SG	SHF	2	ENT
SG	SHF	2	ENT	
STR	SHF	4	ENT	
OUT	SHF	1	2	ENT
STR	SHF	5	ENT	
JMP	SG	SHF	0	ENT

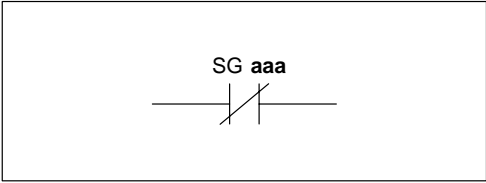
Store Stage
(STR SG)
DL330P Only

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



Store Not Stage
(STR NOT SG)
DL330P Only

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



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Stages SG	—	—	0-177

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

DirectSOFT Display



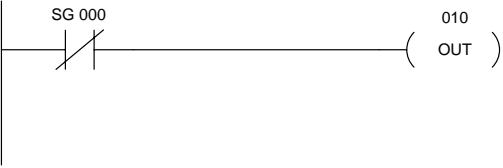
Handheld Programmer Keystrokes

STRSGSHF0ENT

OUTSHF10ENT

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

DirectSOFT Display



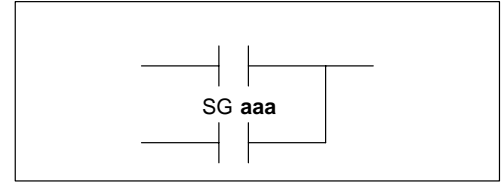
Handheld Programmer Keystrokes

STRNOTSGSHF0ENT

OUTSHF10ENT

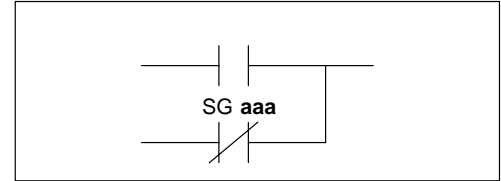
Or Stage (OR SG) DL330P Only

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



Or Not Stage (OR NOT SG) DL330P Only

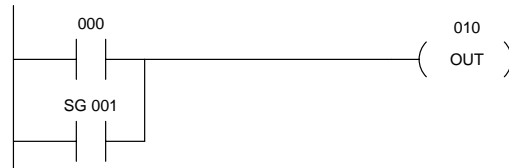
The Or Not instruction logically ors a normally closed stage contact in parallel with another contact in a rung. The status of the contact will be opposite the state of the associated stage memory location.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Stages SG	—	—	0-177

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

DirectSOFT Display



Handheld Programmer Keystrokes

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

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

DirectSOFT Display

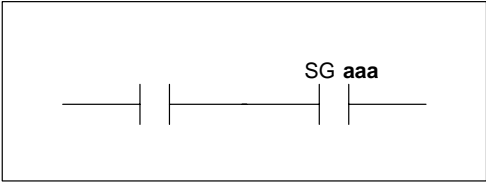


Handheld Programmer Keystrokes

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

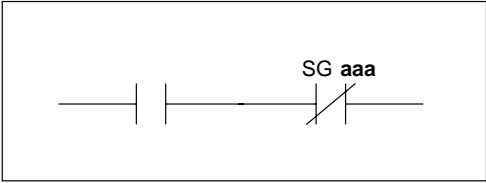
And Stage
(AND Stage)
DL330P Only

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



And Not Stage
(AND NOT SG)
DL330P Only

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



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Stages SG	-	-	0-177

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

DirectSOFT Display

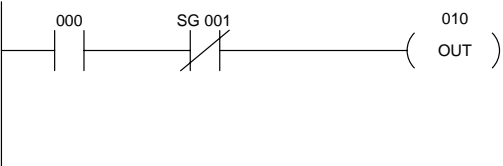


Handheld Programmer Keystrokes

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

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

DirectSOFT Display

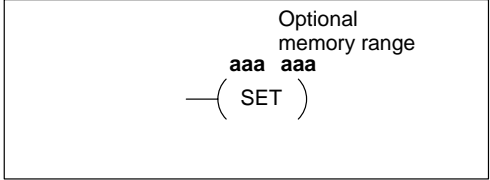


Handheld Programmer Keystrokes

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

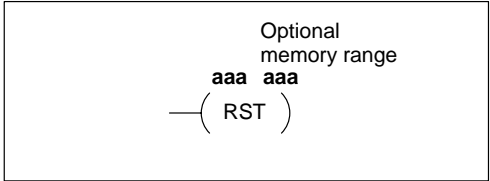
Set (SET) DL330P Only

The Set instruction sets or turns on a output or a consecutive range of outputs. 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.



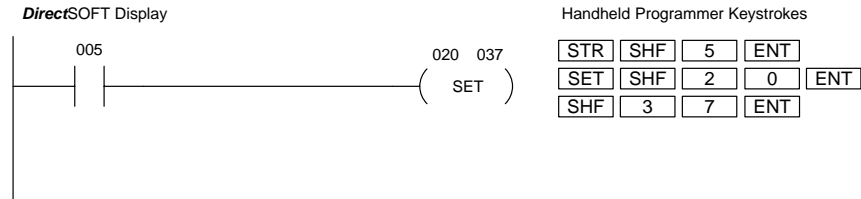
Reset (RST) DL330P Only

The Reset instruction resets or turns off an output or a consecutive range of outputs. 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.

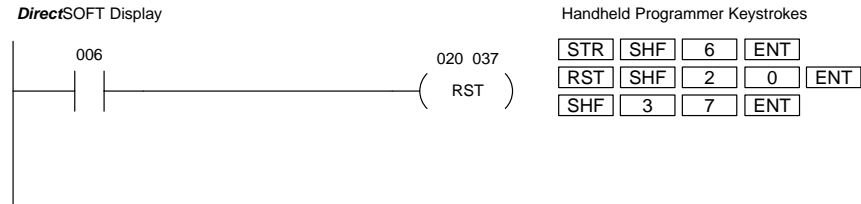


Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Outputs	—	—	000-177 700-767
Control Relays	—	—	160 - 167 170 - 174 200 - 277

In the following Set example, when input location 005 is on, outputs 20-37 will be set on.



In the following Reset example, when input location 006 is on, outputs 020-37 will be reset to the off state.



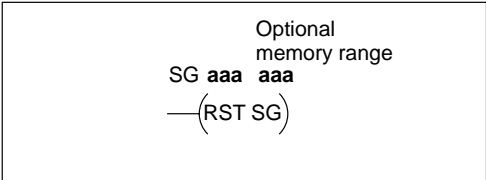
Set Stage
(SET SG)
DL330P Only

The Set Stage instruction sets or turns on a stage or a consecutive range of stages. Once the stage is set it will remain on until a transition is made to another stage or the stage is reset using the Reset Stage instruction. It is not necessary for the input controlling the Set Stage instruction to remain on.



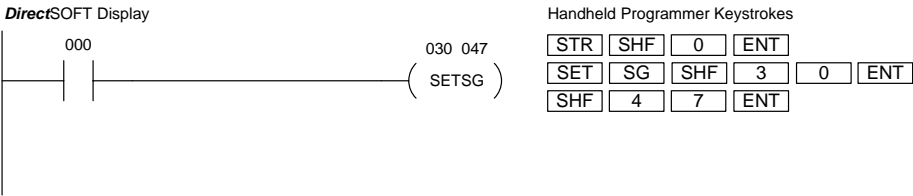
Reset Stage
(RST SG)
DL330P Only

The Reset instruction resets or turns off a stage or a consecutive range of stages. Once the stage(s) is reset it is not necessary for the input to remain on.



Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	aaa	aaa
Stage	—	—	000-177

In the following Set Stage example, when input 000 is on, stages 30-47 will be set on.



In the following Reset Stage example, when input 003 is on, stages 30-47 will be reset off.

