

# **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<sup>PLUS</sup> CPUs. If you are not familiar with RLL<sup>PLUS</sup> 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<sup>PLUS</sup> CPU. There are many others that are discussed in Chapter 11 that you can use as well. If you are using a RLL<sup>PLUS</sup> 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<sup>PLUS</sup> 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 a RLL CPU but only one input line in an RLL<sup>PLUS</sup> 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.

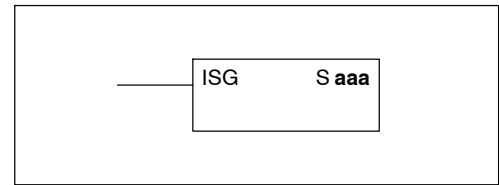
Instruction	Page
AND CNT	12-17
AND SG	12-9
AND TMR	12-16
ANDN CNT	12-17
ANDN SG	12-9
ANDN TMR	12-16
CNT	12-19
ISG	12-3
JMP	12-5
NJMP	12-5
OR CNT	12-15
OR SG	12-8
OR TMR	12-14
ORN CNT	12-15
ORN SG	12-8

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

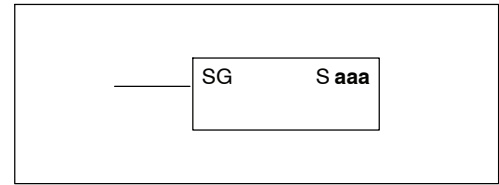
### Initial Stage (ISG) DL330P Only

The Initial Stage instruction is normally used as the first segment of a RLL<sup>PLUS</sup> 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.



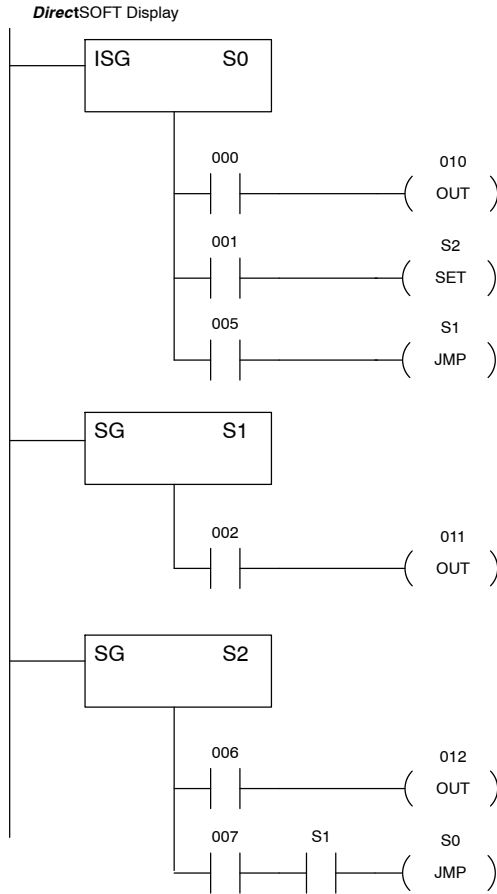
### Stage (SG) DL330P Only

The Stage instruction creates segments of a RLL<sup>PLUS</sup> 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<sup>PLUS</sup> program. This program utilizes the Initial Stage, 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	

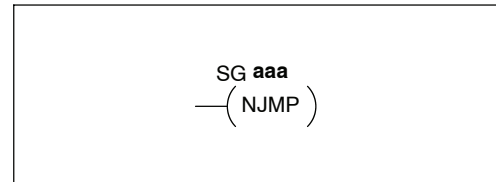
**Jump  
 (JMP)  
 DL330P Only**

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.



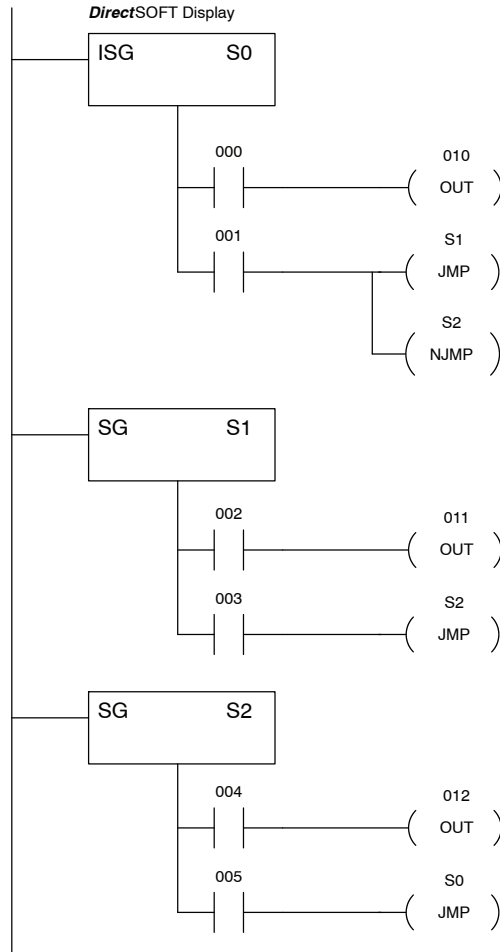
**Not Jump  
 (NOT JMP)  
 DL330P Only**

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<sup>PLUS</sup> program. This program utilizes the Initial Stage, Stage, Jump, and Not Jump instructions to create a structured program.



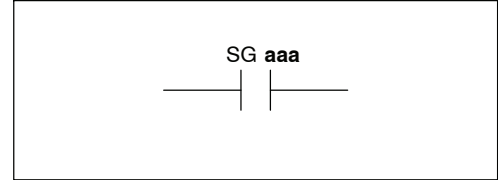
Handheld Programmer Keystrokes

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	ENT
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	

RLL<sup>PLUS</sup> Instructions

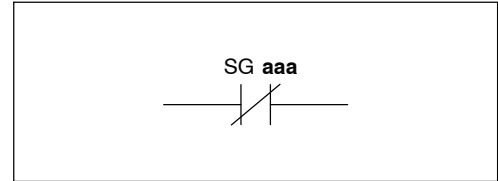
**Store Stage  
(STR SG)  
DL330P Only**

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.



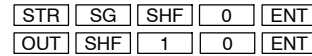
Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Stages	SG	--	0-177

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

DirectSOFT Display



Handheld Programmer Keystrokes

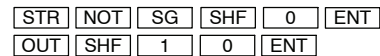


In the following Store Not example, when stage contact 000 is off output 010 will energize.

DirectSOFT Display

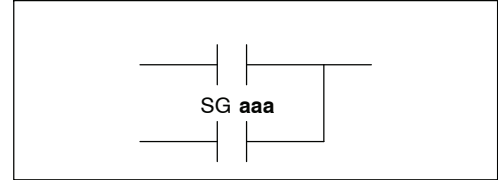


Handheld Programmer Keystrokes



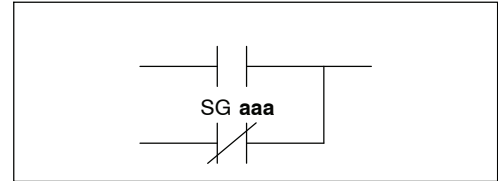
**Or Stage  
(OR SG)  
DL330P Only**

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**

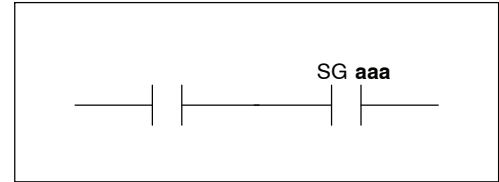
**Handheld Programmer Keystrokes**

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



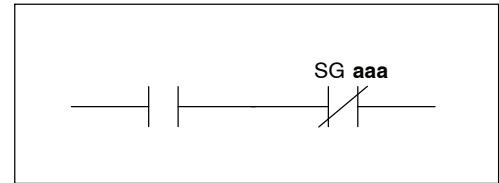
**And Stage  
(AND Stage)  
DL330P Only**

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**

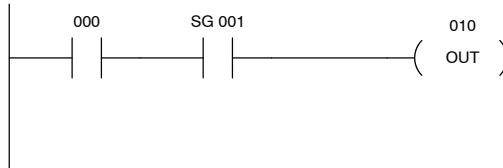
The And Not instruction logically ands a normally closed stage contact in series with another contact in a rung. The status of the contact will be opposite the state of the associated stage memory location.



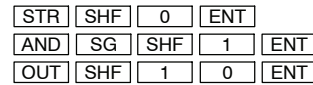
Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Stages SG	-	-	0-177

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

DirectSOFT Display

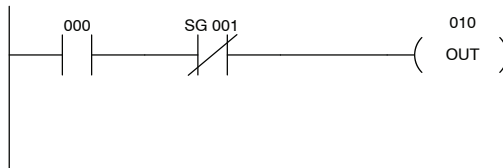


Handheld Programmer Keystrokes

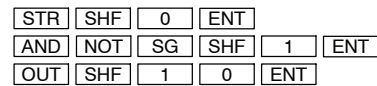


In the following And Not example, when input 000 is on and stage contact 001 is off output 010 will energize.

DirectSOFT Display

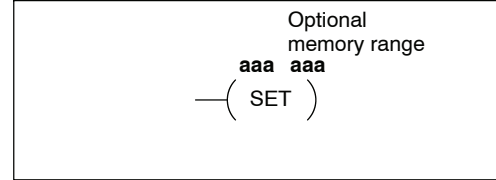


Handheld Programmer Keystrokes



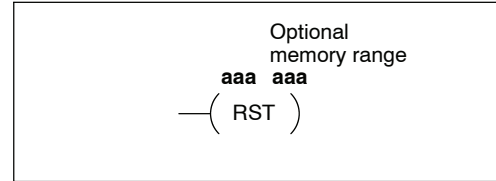
**Set (SET)**  
DL330P Only

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.



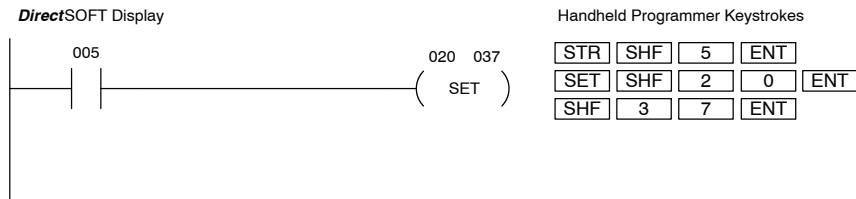
**Reset (RST)**  
DL330P Only

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.

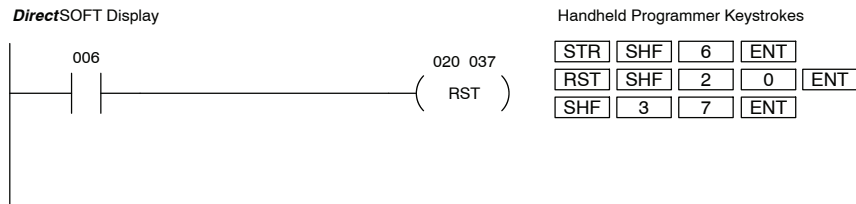


Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaaa	aaaa	aaaa
Outputs	--	--	000-177 700-767
Control Relays	--	--	160 - 167 170 - 174 200 - 277

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.



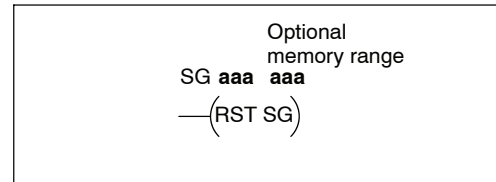
**Set Stage  
(SET SG)  
DL330P Only**

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.



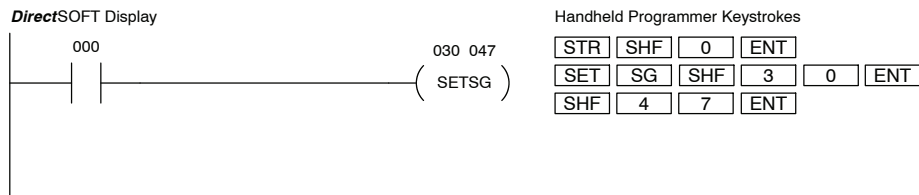
**Reset Stage  
(RST SG)  
DL330P Only**

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.

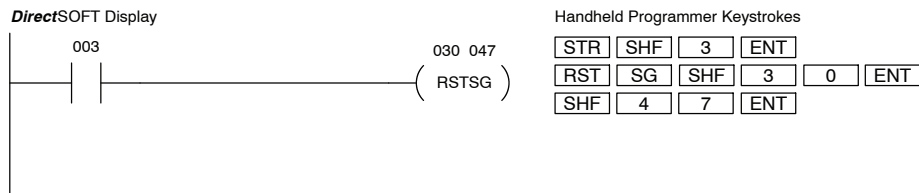


Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	aaa	aaa
Stage	--	--	000-177

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



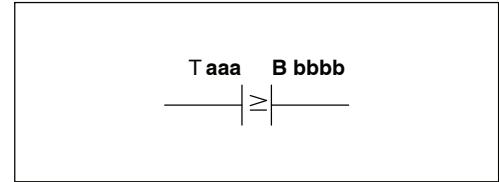
In the following Reset Stage example, when input 003 is on, stages 30-47 will be reset off.



## Comparative Boolean Instructions

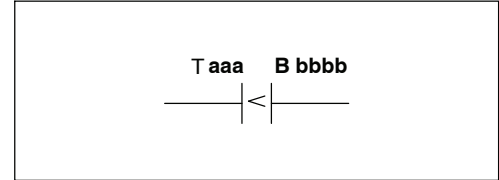
### Store If Greater Than or Equal To Timer (STR TMR) DL330P Only

The Store If Greater Than or Equal To instruction begins a new rung or additional branch in a rung with a normally open comparative timer contact. The contact will be on if the specified timer T aaa  $\geq$  B bbbb.



### Store Not IF Greater Than Timer (STR NOT TMR) DL330P Only

The Store Not If Greater Than instruction begins a new rung or additional branch in a rung with a normally closed comparative timer contact. The contact will be on if the specified timer T aaa  $<$  B bbbb.



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

In the following Store If Greater Than or Equal To example, when T602  $\geq$  the value 1538 the contact will turn on and output 014 will energize.

DirectSOFT Display



Handheld Programmer Keystrokes

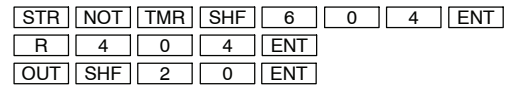


In the following Store Not If Greater Than example, when T602  $<$  the value in R404 the contact will turn on and output 020 will energize.

DirectSOFT Display

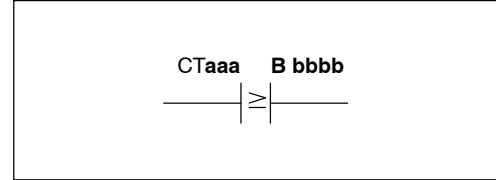


Handheld Programmer Keystrokes



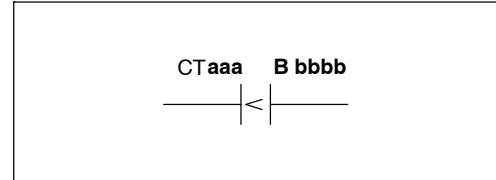
**Store If Greater Than or Equal To Counter (STR CNT)  
DL330P Only**

The Store If Greater Than or Equal To 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  $\geq$  B bbbb.



**Store Not If Greater Than Counter (STR NOT CNT)  
DL330P Only**

The Store Not If Greater Than 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	B	D3-330 Range		D3-340 Range		D3-330P Range	
		aaa	bbbb	aaa	bbbb	aaa	bbbb
Counters	CT	--	--	--	--	600-677	--
Data registers	R	--	--	--	--	--	400-577
Constant	K	--	--	--	--	--	0-9999

In the following Store If Greater Than or Equal To example, when CT602  $\geq$  the value in R404 the contact will turn on and output 014 will energize.

DirectSOFT Display



Handheld Programmer Keystrokes



In the following Store Not If Greater Than example, when CT602  $<$  the constant value 4620 the contact will turn on and output 020 will energize.

DirectSOFT Display

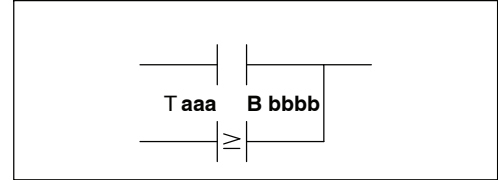


Handheld Programmer Keystrokes



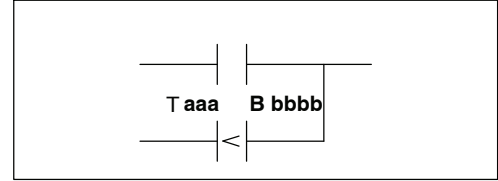
**Or If Greater Than or Equal To Timer (OR TMR)**  
DL330P Only

The Or If Greater Than or Equal To instruction connects a normally open comparative timer contact in parallel with another contact. The contact will be on if the specified timer  $T\ aaa \geq B\ bbbb$ .



**Or Not If Greater Than Timer (OR NOT TMR)**  
DL330P Only

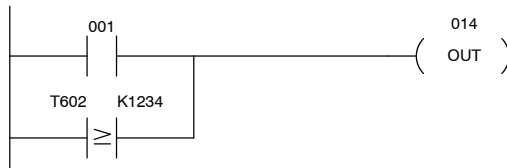
The Or Not If Greater Than instruction connects a normally closed comparative timer contact in parallel with another contact. The contact will be on if the specified timer  $T\ aaa < B\ bbbb$ .



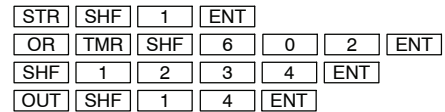
Operand Data Type	B	D3-330 Range		D3-340 Range		D3-330P Range	
		aaa	bbbb	aaa	bbbb	aaa	bbbb
Timers	T	--	--	--	--	600-677	--
Data registers	R	--	--	--	--	--	400-577
Constant	K	--	--	--	--	--	0-9999

In the following Or If Greater Than or Equal To example, when input contact 001 is on or  $T602 \geq$  the value 1234 the contact will turn on and output 014 will energize.

DirectSOFT Display

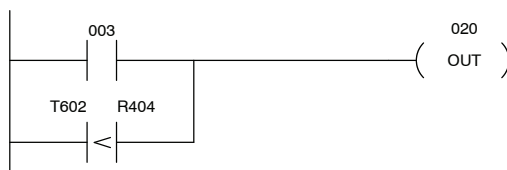


Handheld Programmer Keystrokes

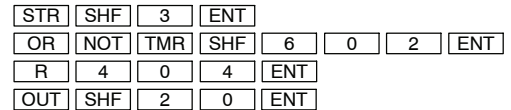


In the following Or Not If Greater Than example, when input contact 003 is on or  $T602 <$  the value in R404 the contact will turn on and output 020 will energize.

DirectSOFT Display

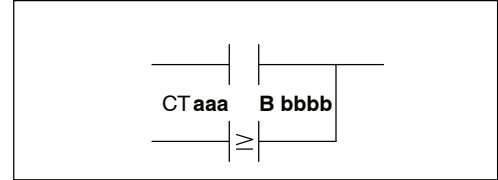


Handheld Programmer Keystrokes



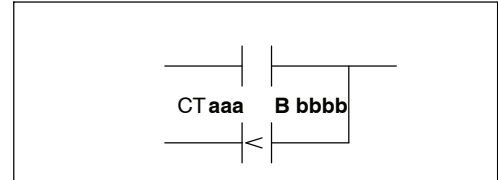
**Or If Greater Than or Equal To Counter (OR CNT) DL330P Only**

The Or If Greater Than or Equal To instruction connects a normally open comparative counter contact in parallel with another contact. The contact will be on if the specified counter CT aaa  $\geq$  B bbbb.



**Or Not If Greater Than Counter (OR NOT CNT) DL330P Only**

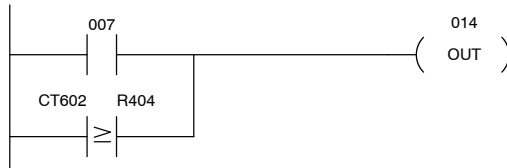
The Or Not If Greater Than instruction connects a normally closed comparative counter contact in parallel with another contact. The contact will be on if the specified counter CT aaa  $<$  B bbbb.



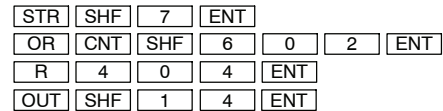
Operand Data Type	B	D3-330 Range		D3-340 Range		D3-330P Range	
		aaa	bbbb	aaa	bbbb	aaa	bbbb
Counters	CT	--	--	--	--	600-677	--
Data registers	R	--	--	--	--	--	400-577
Constant	K	--	--	--	--	--	0-9999

In the following Or If Greater Than or Equal To example, when input contact 007 is on or CT602  $\geq$  the value in R404 the contact will turn on and output 014 will energize.

DirectSOFT Display

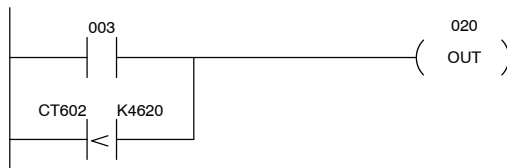


Handheld Programmer Keystrokes

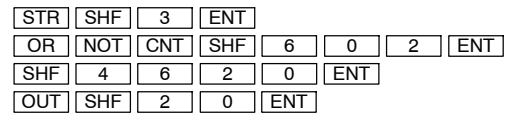


In the following Or Not If Greater Than example, when input contact 003 is on or CT602  $<$  the constant value 4620 the contact will turn on and output 020 will energize.

DirectSOFT Display

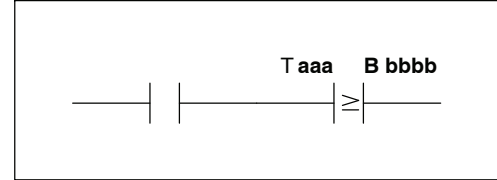


Handheld Programmer Keystrokes



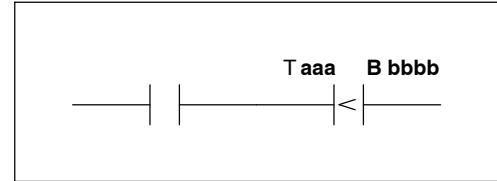
**And If Greater Than or Equal To Timer (AND TMR) DL330P Only**

The And If Greater Than or Equal To instruction connects a normally open comparative timer contact in series with another contact. The contact will be on if the specified timer T aaa  $\geq$  B bbbb.



**And Not If Greater Than Timer (AND NOT TMR) DL330P Only**

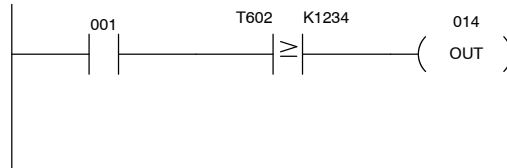
The And Not If Greater Than instruction connects a normally closed comparative timer contact in series with another contact. The contact will be on if the specified timer T aaa  $<$  B bbbb.



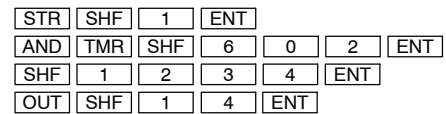
Operand Data Type	D3-330 Range		D3-340 Range		D3-330P Range		
	B	aaa	bbb	aaa	bbb	aaa	bbb
Timers	T	--	--	--	--	600-677	--
Data registers	R	--	--	--	--	--	400-577
Constant	K	--	--	--	--	--	0-9999

In the following And If Greater Than or Equal To example, when input contact 001 is on and T602  $\geq$  the value 1234 the contact will turn on and output 014 will energize.

DirectSOFT Display

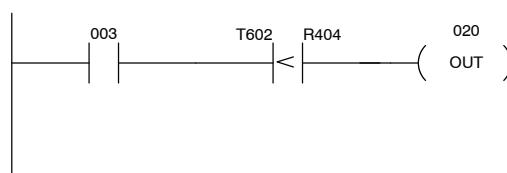


Handheld Programmer Keystrokes

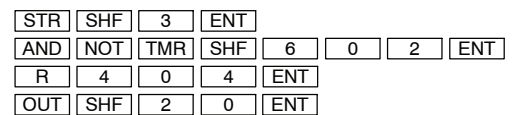


In the following Store Not If Greater Than example, when input contact 003 is on and T602  $<$  the value in R404 the contact will turn on and output 020 will energize.

DirectSOFT Display



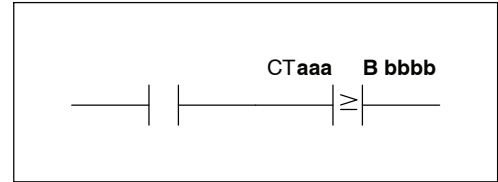
Handheld Programmer Keystrokes





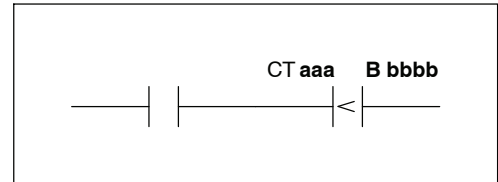
**And If Greater Than or Equal To Counter (AND CNT)  
DL330P Only**

The And If Greater Than or Equal To instruction connects a normally open comparative counter contact in series with another contact. The contact will be on if the specified counter CT aaa  $\geq$  B bbbb.



**And Not If Greater Than Counter (AND NOT CNT)  
DL330P Only**

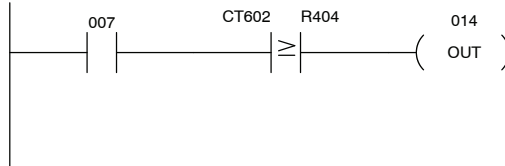
The And Not If Greater Than instruction connects a normally closed comparative counter contact in series with another contact. The contact will be on if the specified counter CT aaa  $<$  B bbbb.



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

In the following Or If Greater Than or Equal To example, when input contact 007 is on and CT602  $\geq$  the value in R404 the contact will turn on and output 014 will energize.

DirectSOFT Display

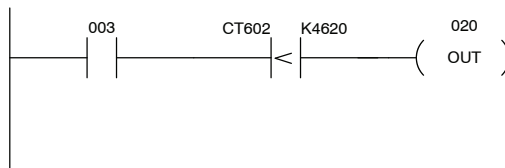


Handheld Programmer Keystrokes

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

In the following Or Not If Greater Than example, when input contact 003 is on and CT602  $<$  the constant value 4620 the contact will turn on and output 020 energize.

DirectSOFT Display



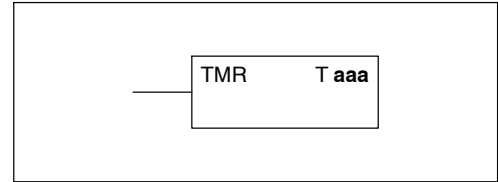
Handheld Programmer Keystrokes

STR SHF 3 ENT  
AND NOT CNT SHF 6 0 2 ENT  
SHF 4 6 2 0 ENT  
OUT SHF 2 0 ENT

# Timer, Counter, and Shift Register Instructions

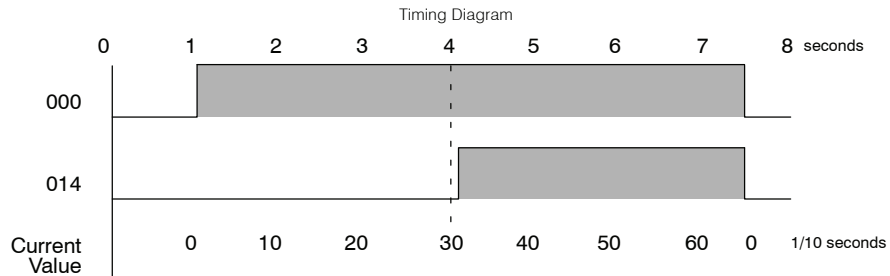
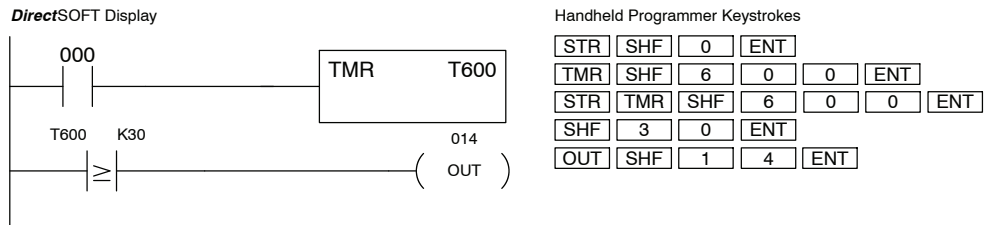
## Timer (TMR) DL330P Only

The Timer instruction used in the DL330P CPU 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 with relay 770 on. The timer will time up to the maximum value (999.9 or 99.99) as long as the input logic remains on, once the input logic turns off the timer will reset to 0. There is no timer bit associated with this timer. Comparative boolean instructions must be used to monitor the current value of this timer.



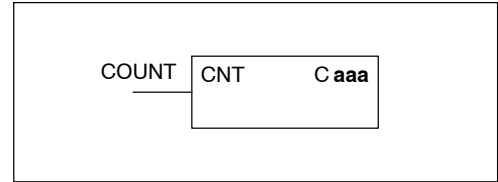
Operand Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	aaa	aaa
Time	--	--	600-677

In the following Timer example when input contact 000 is on timer 600 will time up. When input contact 000 goes off the timer will reset to zero. The comparative instruction will monitor the current value of the timer and energize when the current value of the timer is greater than or equal to the constant K30.



**Counter (CNT)  
DL330P Only**

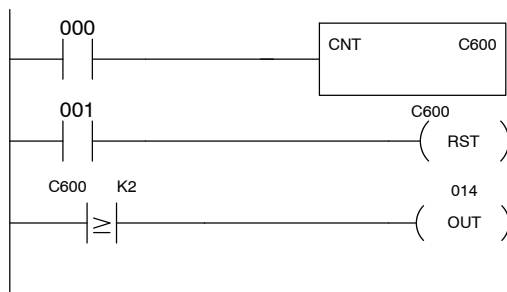
The Counter instruction used in the DL330P CPU provides a single input counter with a counting range of 0–9999. The counter will count up to 9999 and stop. The Reset Counter instruction must be used to reset this counter. There is no counter bit associated with this counter, so comparative boolean instructions must be used to monitor the current value of this counter.



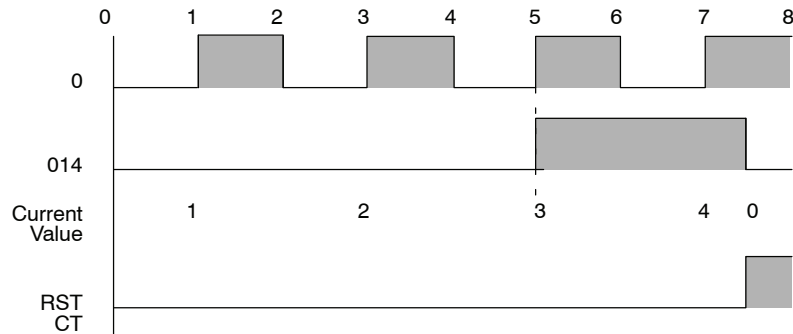
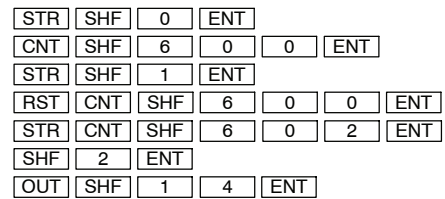
Operand Data Type	D3-330 Range	D3-340 Range	D3-330P Range
	aaa	aaa	aaa
Counter	--	--	600-677

In the following Counter example when input contact 000 transitions from off to on counter 600 will increment by one. When input contact 001 is on the Reset Counter instruction will reset the counter to 0. The comparative instruction will monitor the current value of the counter and energize when the current value of the counter  $\geq$  the constant K2.

DirectSOFT Display

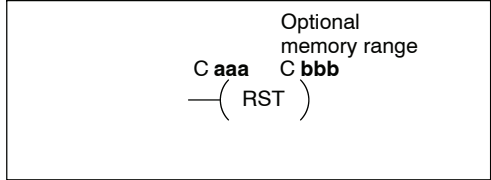


Handheld Programmer Keystrokes



**Reset Counter (RST)**  
**DL330P Only**

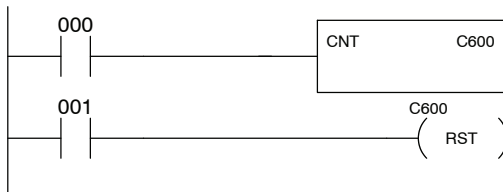
The Reset Counter instruction used in the DL330P CPU provides a reset for the counter instruction. One counter or a range of counters can be reset.



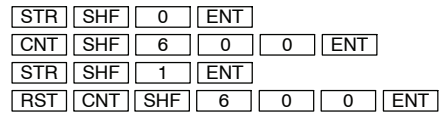
Operand Data Type	D3-330 Range		D3-340 Range		D3-330P Range	
	aaa	bbbb	aaa	bbbb	aaa	bbbb
Counters	--	--	--	--	600-677	600-677

In the following Reset Counter example when input contact 001 is on the Reset Counter instruction will reset counter 600.

DirectSOFT Display



Handheld Programmer Keystrokes

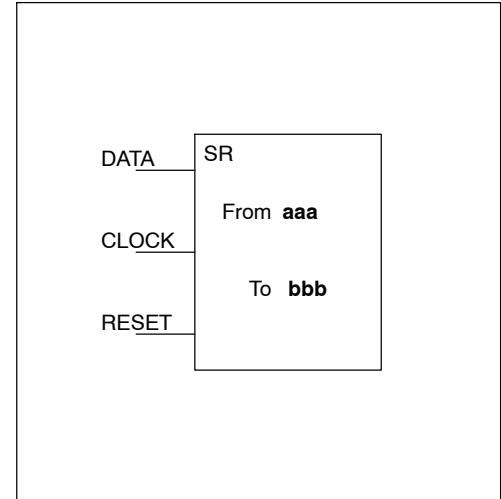


### Shift Register (SR) DL330P Only

The Shift Register instruction shifts data through a predefined number of control relays. There are 77 control relays which can be used for internal control relays or shift register bits. 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 77.

The Shift Register has three input contacts.

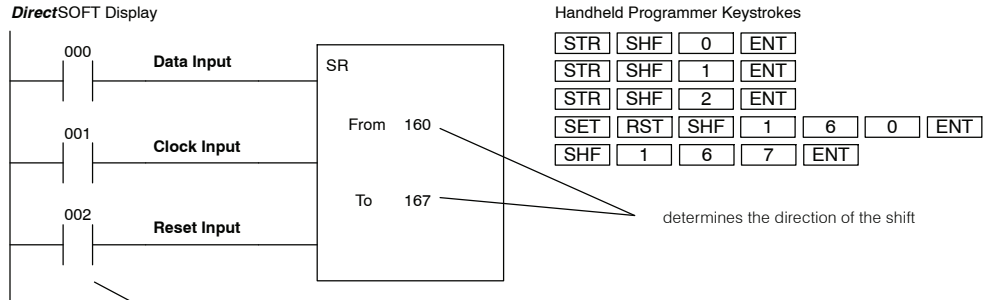
- Data — determines the value (1 or 0) that will enter the register
- Clock — shifts the bits one position on each low to high transition
- Reset — resets the Shift Register to all zeros.



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 placed into the starting bit position of the shift register. The direction of the shift depends on the entry in the From and To fields. From 160 to 167 would define a shift right block of eight bits to be shifted from bit left to right. From 167 to 160 would define a shift left block of eight bits, but would shift from right to left. The maximum size of the shift register block is limited to 77 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	--	--	--	--	160-174 200-277	160-174 200-277

In the following example, when the clock input transitions from low to high the value in the Data input is placed in the first bit position of the shift register and the successive successive bits are shifted to the right. When the Reset input transitions from low to high the entire shift register is set to zeros.



Inputs on Successive Scans			Shift Register Bits																		
Data	Clock	Reset	160															167			
1	1	0	█																		
0	1	0	█																		
0	1	0		█																	
1	1	0	█	█																	
0	1	0	█	█	█																
0	0	1																			

RLL<sup>PLUS</sup> Instructions