34. FUNCTION REFERENCE

The function references that follow are meant to be an aid for programming. There are some notes that should be observed, especially because this list discusses instructions for more than one type of PLC.

• The following function descriptions are for both the Micrologix and PLC-5 processor families. There are some differences between PLC models and families.
  - Floating point operations are not available on the Micrologix.
  - Some instruction names, definition and terminologies have been changed from older to newer models. I attempt to point these out, or provide a general description that is true for all.
  - Details for specific instructions can be found in the manuals available at (http://www.ab.com)

Many flags in status memory can be used with functions, including;
- S2:0/0 carry in math operation
- S2:0/1 overflow in math operation
- S2:0/2 zero in math operation
- S2:0/3 sign in math operation

34.1 FUNCTION DESCRIPTIONS

34.1.1 General Functions

AFI

Status Bits:
Putting this instruction in a line will force the line to be false. This is primarily designed for debugging programs.

Description:
none

Registers: none

Available on: Micrologix, PLC-5

AFI - Always False Instruction
Status Bits:
These functions update a few inputs and outputs during a program scan, instead of the beginning and end. In this example the IIN function will update the input values on 'I:001' if 'A' is true. If 'B' is true then the output values will be updated for 'O:002'.

Description:
none

Registers: none

Available on: Micrologix, PLC-5

IIN, IOT - Immediate INput, Immediate OuTput

---

Status Bits:
The OTL 'L' will latch on an output or memory bit, and the 'OTL 'U' will unlatch it. If a value has been changed with a latch its value will stay fixed even if the PLC has been restarted.

Description:
none

Registers: none

Available on: Micrologix, PLC-5

OTL, OTU - OutpuT Latch, OutpuT Unlatch
**34.1.2 Program Control**

**Status Bits:**

These are the three most basic and common instructions. The input 'A' is a normally open contact (XIC), the input 'B' is a normally closed contact (XIO). Both of the outputs are normally off (OTE).

**Description:** none

**Registers:** none

**Available on:** Micrologix, PLC-5

XIC, XIO, OTE - eXamine If Closed, eXamine If Open, OuTput Enable

A I:001/0
B I:001/1

**Status Bits:**

The JMP instruction will allow the PLC to bypass some ladder logic instructions. When 'A' is true in this example the JMP will go to label '2', after which the program scan will continue normally. If 'A' is false the JMP will be ignored and program execution will continue normally.

In either case, 'X' will be equal to 'B'.

**Description:** none

**Registers:** none

**Available on:** Micrologix, PLC-5

JMP, L BL - JuMP, LaBe L

X
JMP
JUMP
Label 2
B
LBL
2
Status Bits:

MCR instructions need to be used in pairs. If the first MCR line is true the instructions up to the next MCR will be examined normally. If the first MCR line is not true the outputs on the lines after will be FORCED OFF. Be careful when using normal outputs in these blocks.

Description:
none

Registers: none

Available on: Micrologix, PLC-5

MCR - Master Control Relay

---

Status Bits:

This instruction will allow a line to be true for only one scan. If 'A' becomes true then output of the 'ONS' instruction will turn on for only one scan. 'A' must be turned off for one scan before the 'ONS' can be triggered again. The bit is used to track the previous input state, it is similar to an enabled bit.

Description:
none

Registers: none

Available on: Micrologix, PLC-5

ONS - ONe Shot

---

X
A
ONS
B3/10
34.1.3 Timers and Counters

Counter memory instructions can share the same memory location, so some redundant bits are mentioned here.

Status Bits:

This instruction will convert a single positive edge and convert it to a bit that is on for only one scan. When 'A' goes from false to true a positive (or rising) edge occurs, and bit 'O:001/2' will be on for one scan. Bit 'B3:4/5' is used to track the state of the input to the function, and it can be considered equivalent to an enable bit.

The OSF function is similar to the OSR function, except it is triggered on a negative edge where the input falls from true to false.

Description:

none

Registers: none

Available on: Micrologix, PLC-5

OSR, OSF - One Shot Rising, One Shot Falling

OSR

ONE SHOT RISING

Storage Bit B3:4/5

Output Bit 2

Output Word O:001

Status Bits:

When 'A' is true this statement will cause the PLC to stop examining the ladder logic program, as if it has encountered the normal end-of-program statement.

Description:

none

Registers: none

Available on: Micrologix, PLC-5

TND - Temporary END

TND

A
plc function ref - 34.6

CTD
COUNT DOWN
Counter
Pres et
Acc um.
C5:0
50
0

A

Status Bits:
The counter accumulator will decrease once each time the input goes from false to true. If the accumulator value reaches the preset the done bit, DN, will be set. The accumulator value will still decrease even when the done bit is set.

Description:
CU
CD
DN
OV
UN
Not used for this instruction
Will be true when the input is true
Will be set when ACC < PRE
Not used for this instruction
Will be set if the counter value has gone below -32,768

Registers:
ACC
PRE
The time that has passed since the input went true
The maximum time delay before the timer goes on

Available on: Micrologix, PLC-5
Status Bits:
The counter accumulator will increase once each time the input goes from false to true. If the accumulator value reaches the preset the done bit, DN, will be set. The accumulator value will still increase even when the done bit is set.

Description:
CU: Will be true when the input is true
CD: Not used for this instruction
DN: Will be set when ACC >= PRE
OV: Will be set if the counter value has gone above 32,767
UN: Not used for this instruction

Registers:
ACC: The total count
PRE: The maximum count before the counter goes on

Available on: Micrologix, PLC-5
Status Bits:
This timer will delay turning off (the done bit, DN, will turn on immediately). Once the input turns off the accumulated value (ACC) will start to increase from zero. When the preset (PRE) value is reached the DN bit is turned off and the accumulator will reset to zero. If the input turns on before the off delay is complete the accumulator will reset to zero.

Description:
- EN: This bit is true while the input to the timer is true
- TT: This bit is true while the accumulator value is increasing
- DN: This bit is true when the accumulator value is less than the preset value and the input is true, or the accumulator is changing

Registers:
- ACC: The time that has passed since the input went false
- PRE: The maximum time delay before the timer goes off

Available on: Micrologix, PLC-5
plc function ref - 34.9

TON - Timer ON DELAY

Timer
Time Base
Pres et
Acc um.

A

Status Bits:
This timer will delay turning on, but will turn off immediately. Once the input turns on the accumulated value (ACC) will start to increase from zero. When the preset (PRE) value is reached the DN bit is set. The done bit will turn off and the accumulator will reset to zero if the input goes false.

Description:
EN
TT
DN
This bit is true while the input to the timer is true
This bit is true while the accumulator value is increasing
This bit is true when the accumulator value is equal to the preset value

Registers:
ACC
PRE
The time that has passed since the input went true
The maximum time delay before the timer goes on

Available on: Micrologix, PLC-5
34.1.4 Compare RTO

RETENTIVE TIMER ON

Timer Time Base

T4:0 1.0 10 0

A

Status Bits:
This timer will delay turning on. When the input turns on the accumulated value (ACC) will start to increase from zero. When the preset (PRE) value is reached the DN bit is set. If the input goes false the accumulator value is not reset to zero. To reset the timer and turn off the timer the RES instruction should be used.

Description:
EN TT DN
This bit is true while the input to the timer is true
This bit is true while the accumulator value is increasing
This bit is true when the accumulator value is less than the preset value

Registers:
ACC PRE
The time that has passed since the input went true
The maximum time delay before the timer goes on

Available on: Micrologix, PLC-5

RTO - Retentive Timer On

CMP - Compare

Expression
"(N7:0 + 8) > N7:1"

A

Status Bits:
This function uses a free form expression to compare the two values. The comparison values that are allowed include =, >, >=, <>, <, <=. The expression must not be more than 80 characters long.

Description:
none

Registers: none

Available on: PLC-5

CMP - Compare
**plc function ref - 34.11**

**DT R**

**DATA TRANSITION**

**Source**

**Mask**

**Reference**

N7:0

00FF

N7:1

A

**Status Bits:**

This function will examine the source value and mask out bits using the mask. The value will be compared to the Reference value, and if the values agree, then the function will be true for one scan, after that it will be false.

**Description:**

none

**Registers:** none

**Available on:** Micrologix, PLC-5

---

**DTR - Data TRansition**

**EQU**

**EQUALS**

**Source A**

**Source B**

N7:0

N7:1

A

**Status Bits:**

The basic compare has six variations. Each of these will look at the values in source A and B and check for the comparison case. If the comparison case is true, the output will be true. The types are,

- EQU - Equals
- GEQ - Greater than or equals
- GRT - Greater than
- LEQ - Less than or equals
- LES - Less than
- NEQ - Not equal

**Description:**

none

**Registers:** none

**Available on:** Micrologix, PLC-5

---

**EQU, GEQ, GRT, LEQ, LES, NEQ - EQUals, Greater than or EQuals, GReater Than, Less than or EQuals, LESs than, Not EQuals**
This instruction will compare the bits in two files and store the positions of differences in a result file. In this example the files compared when 'A' goes true. Both files start at 'B3:0' and 'B9:0' and 10 bits are to be compared. When differences are found the bit numbers will be stored in a list starting at 'N10:0', the list can have up to three values (integer words). The 'Cmp Control' word is for the bits being compared. The 'Result Control' word is for the list of differences. The manual recommends clearing 'S:24' before running this instruction to avoid a possible processor fault.

The DDT instruction is the same as the FBC instruction, except that when a different bit is found the source bit overwrites the reference bit. It is useful for storing a reference pattern for later use by a FBC.

**Description:**
- **Enable (Cmp)**: enabled when the instruction input is active
- **Done (Cmp)**: enabled when the operation is complete
- **Error (Cmp)**: set if an error occurred during the operation
- **Inhibit (Cmp)**: set when mismatch found, must be cleared to continue the comparison
- **Found (Cmp)**: set when a mismatch is found
- **Done (Result)**: set when the result list is full
- **Error (Result)**: set if an error occurred with the results list

**Registers:**
- **Length (Cmp)**: the number of bits to be compared
- **Position (Cmp)**: the position of the current bit being compared
- **Length (Result)**: the number of result positions allowed
- **Position (Result)**: the location of the last result added

Available on: Micrologix, PLC-5

**FBC, DDT - File Bit Compare, Diagnostic Detection**
**LIM**

**LIMIT TEST (CIRC)**

**Low limit Test**

**High Limit**

N7:0

N7:1

N7:2

**Status Bits:**

This function will check to see if a value is between two limits. If the high limit is larger than the low limit and the test value is \( \geq \) low limit or \( \leq \) high limit, then the output is true. If the low limit is higher than the high limit, then a value not between the low and high limits will be true.

**Description:** none

**Registers:** none

**Available on:** Micrologix, PLC-5

**MEQ - Masked EQUAL**

**Status Bits:**

The Source and Mask values are ANDed together. This will screen out bits not on in the mask. The value is then compared to the 'Compare' value. If the values are equal, the output is true.

**Description:** none

**Registers:** none

**Available on:** Micrologix, PLC-5
34.1.5 Calculation and Conversion

**ACS**

**ARCCOSINE**

**Source**

De st N7:0 N7:1

**A**

**Status Bits:**

- **C** Carry - set if a carry is generated
- **V** Overflow - only set if value exceeds maximum for number type
- **Z** Zero - sets if the result is zero.
- **S** Sign - set if result is negative

**Registers:** none

**Available on:** Micrologix, PLC-5

ACS, ASN, ATN, COS, LN, LOG, NEG, SIN, SQR, TAN - ArcCosine, ArcSiNe, ArcTaNgent, COSine, Logarythm Natural, LOGarythm, NEGative, SINe, SQuare Root, T ANgent
ADD Source A
Source B
N7:0
N7:1

Status Bits:
These are binary math functions that will load two values from sources A and B, do the calculation indicated, and store the results in the destination. Functions possible include
ADD - Add two numbers
DIV - Divide source A by source B
MUL - Multiply A and B
SUB - Subtract B from A
XPY - Raise X to the power of Y

Description:
C
V
Z
S
Carry - sets if a carry is generated
Overflow - only set if value exceeds maximum for number type
Zero - sets if the result is zero
Sign - sets if the result is negative

Registers: none
Available on: Micrologix, PLC-5

ADD, DIV, MUL, SUB, XPY - ADDition, DIVision, MULtiplication,
De st N7:2
SUBtraction, X to the Power of Y
Status Bits:
These functions do the basic statistical calculations, average (AVE) and standard deviation (STD). When the input goes from false to true the calculation is begun. The values to be used for the calculation are taken from the memory starting at the start of the file location, for the length indicated. The final result is stored in the Dest. The control file is used for the calculation to keep track of position, and indicate when the calculation is done (it may take more than one PLC scan).

Description:
- Carry - always 0
- Overflow - only set if value exceeds maximum for number type
- Zero - sets if the result is zero.
- Sign - sets if the result is negative
- Enable - on when the instruction input is on
- Done - set when the calculation is complete
- Error - set if an error was encountered during calculation

Registers: none
Available on: Micrologix, PLC-5

AVE, STD - AVERAGE, STANDARD DEVIATION

Control R6:0
Length
Position
PLC Function Ref - 34.17

CLR - CLeaR

CPT - ComPuTe

De st N7:0

A

Status Bits:
This value will clear a memory location by putting a zero in it when the input to the function is true.

Description:
none

Registers: none

Available on: Micrologix, PLC-5

Expression "N7:1 - N7:3"

Status Bits:
This expression allows free-form entry of equations. A maximum of 80 characters is permitted. Operations allowed include +, -, | (divide), *, FRD, BCD, SQR, AND, OR, NOT, XOR, ** (x^y = x to power y), RAD, DEG, LOG, LN, SIN, COS, TAN, ASN, ACS, ATN

Description:
none

Registers: none

Available on: PLC-5

PLC function ref - 34.18

FRD
FROM BCD
Source
De st
N7:0
N7:1
A
Status Bits:
This function will convert the value in the source location and store the result in the Dest location. The functions possible include,
FRD - From BCD to a 2s compliment integer number
TOD - From 2s compliment integer number to BCD
DEG - Convert from radians to degrees
RAD - Convert from degrees to radians

Description:

C
V
Z
S
Carry - always 0
Overflow - sets if an overflow as generated during conversion
Zero - sets if the result is zero.
Sign - sets if the MSB of the result is set

Registers: none

Available on: Micrologix, PLC-5

FRD, TOD, DEG, RAD - From BCD to integer, TO BCD from integer, DEGREES from radians, RADIIANS from degrees
This function sorts the values in memory from lowest value in the first location to the highest value. When the input goes from false to true the calculation is begun. The values to be used for the calculation are sorted in the memory starting at the start of the file location, for the length indicated. The control file is used for the calculation to keep track of position, and indicate when the calculation is done (it may take more than one PLC scan).

**Description:**
- **Enable** - on when the instruction input is on
- **Done** - set when the calculation is complete
- **Error** - set if an error was encountered during calculation

**Registers:** none

**Available on:** Micrologix, PLC-5
34.1.6 Logical

**AND**

Source A

Source B

N7:0

N7:1

**Description:**

- **C (Carry):** always 0
- **V (Overflow):** always 0
- **Z (Zero):** sets if the result is zero.
- **S (Sign):** sets if the MSB of the result is set.

**Status Bits:**

- **C (Carry):** always 0
- **V (Overflow):** always 0
- **Z (Zero):** sets if the result is zero.
- **S (Sign):** sets if the MSB of the result is set.

**Available on:** Micrologix, PLC-5

**Description:**

This function will invert all of the bits in a word in memory whenever the input is true.

**Status Bits:**

- **C (Carry):** always 0
- **V (Overflow):** always 0
- **Z (Zero):** sets if the result is zero.
- **S (Sign):** sets if the MSB of the result is set.

**Available on:** Micrologix, PLC-5
34.1.7 Move

**BIT FIELD DISTRIB**

**Source**

**Source bit**

N7:0

A

**Status Bits:**

- This function will copy the bits starting at N7:0/0 to N7:1/4 for a length of 5 bits.

**Description:**

- none

**Registers:**

- none

**Available on:** Micrologix, PLC-5

**BTD - BiT Distribute**

<table>
<thead>
<tr>
<th>Dest N7:1</th>
<th>Dest bit</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>N7:0</td>
<td>N7:1</td>
<td>4</td>
</tr>
</tbody>
</table>

**MOV**

**MOVE**

**Source**

**Dest**

N7:0

N7:1

A

**Status Bits:**

- This instruction will move values from one location to another, and if necessary change value types, such as integer to a floating point.

**Description:**

- Carry - always 0
- Overflow - Sets if an overflow occurred during conversion
- Zero - sets if the result is zero.
- Sign - sets if the MSB of the result is set

**Registers:**

- none

**Available on:** Micrologix, PLC-5

**MOV - MOYe**
Most file instructions will contain Mode options. The user may choose these with the implications listed below.

- **All** - All of the operations will be completed in a single scan when the input to the function is edge triggered. Care must be used not to create an operation so long it causes a watchdog fault in the PLC.
- **Incremental** - Each time there is a positive input edge the function will advance the file operation by one.
- **'number'** - when a number is supplied the function will perform that many iterations while the input rung is true.

**MVM - MoVe Masked**

**Source**

<table>
<thead>
<tr>
<th>Source</th>
<th>Mask</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>N7:0</td>
<td></td>
<td>N7:2</td>
</tr>
<tr>
<td>N7:1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Status Bits:**

- **C** - Carry - always 0
- **V** - Overflow - always 0
- **Z** - Zero - sets if the result is zero.
- **S** - Sign - sets if the MSB of the result is set.

**Description:**

This function will retrieve the values from the source and mask memory and AND them together. Only the bits that are true in the mask will be copied to the new location.

**Registers:** none

**Available on:** Micrologix, PLC-5
Status Bits: This instruction copies from one list to another. When 'A' is true the instruction will copy the entire source list to the destination location in a single scan. In this example this would mean N7:20=N7:50, N7:21=N7:51, N7:22=N7:52 and N7:23=N7:53. The source values are not changed. This instruction will not convert data types.

Description: none

Registers: none

Available on: Micrologix, PLC-5

COP - file COPy
This function will evaluate the expression over a range of values. The length specifies the number of positions in the expression and destination files. The position value will be updated to indicate the current position in the calculation. See earlier in this section for a description of the Mode variable. This example would perform all of the calculations in a single scan. These calculations would be N7:10=N7:0-N7:21, N7:11=N7:1-N7:21, ......N7:19=N7:9-N7:21. More complex mathematical expressions can be used with the following operators; +, -, *, | - basic math
BCD/FRD - BCD conversion
SQR - square root
AND, OR, NOT, XOR - Boolean operators
Note: advanced math operators are also available

Status Bits:
- enable - this will be on while the function is active
- done - this will be on when a calculation has completed
- error - this will be set if there was an error during calculation

Registers:
- position - tracks the current position in the list
- length - the length of the file

Available on: Micrologix, PLC-5
Status Bits:
The contents of a single memory location are copied into a list. In this example the value in 'F8:0' is copied into locations 'F8:30' to 'F8:39' each scan when 'A' is true. The source value is not changed. This instruction will not convert data types.

Description:
none

Registers: none
Available on: Micrologix, PLC-5
lists of numbers can be compared using the FSC command. When 'A' becomes true the function will start to compare values as determined by the 'Mode' (see the beginning of this section for details on the mode). The expression will be evaluated from the initial locations in the expression. The end of the list is determined by the Length. In this example 3 values will be evaluated for each scan. The comparison in the first scan will be F8:5>F8:0, F8:6>F8:0 and F8:7>F8:0. This instruction will continue until all 14 values have been compared, and all are true, at which time X will turn on and stay on while A is on. If any values are false the compare will stop, and the output will stay off.

**Description:**
- **EN**
- **DN**
- **ER**
- **IN**
- **FD**

- **enable** - will be on while the instruction input is on
- **done** - will be on when the length is reached, or a false compare occurred
- **error** - will occur if there is an error in the expression or range
- **inhibit** - if a false statement is found the inhibit bit will be set. if this is turned off (i.e., R6:36/IN=0) the search will continue
- **found** - this bit will be set when a false condition is found

**Registers:**
- **LEN**
- **POS**

**length** - the number of the comparison list

**position** - the current position in the comparison list

Available on: Micrologix, PLC-5
34.1.9 List

**BSL**

**File**

**Control**

**Bit Address**

**Length**

#B3:0

R6:0

I:0.0/0

6

**Description:**

These functions will shift bits through left or right through a string of bits starting at #B3:0 with a length of 6 in the example above. As the bits shift the bit shifted out will be put in the UL bit. A new bit will be shifted into the vacant spot from the Bit Address. When the bits are shifted they are moved in the memory locations starting at file #B3:0.

The two options available are:

- **BSR** - Bit Shift Right
- **BSL** - Bit Shift Left

**Enable** - is on when the input to the function is on

**Done** - is on when the shift operation is complete

**Error** - indicates when an error has occurred

**Unload** - the unloaded value is stored in this bit

**Registers:** none

**Available on:** Micrologix, PLC-5
Status Bits:
Stack instructions will take integer words and store them, and then allow later retrieval. The load instructions will store a value on the stack on a false to true input change. The Unload instructions will remove a value from that stack and store it in the Dest location. A Last On First Off stack will return the last value pushed on. A First On First Off stack will give the oldest value on the stack. If an attempt to load more than the stack length, the values will be ignored. The instructions available are:

- **FFL** - FIFO stack load
- **FFU** - FIFO stack unload
- **LFL** - LIFO stack load
- **LFU** - LIFO stack unload

**Description:**
- **EN** - Enable - is on when the input to the function is on
- **DN** - Done - is on when the shift operation is complete
- **ER** - Error - indicates when an error has occurred
- **UL** - Unload - the unloaded value is stored in this bit

**Registers:** none

**Available on:** Micrologix, PLC-5
### Function Description

This function is used to compare a source value to a set of values in a sequencer table. In this example, the 8 most significant bits of 'N7:0' will be loaded each time 'A' goes from false to true. The sequencer will load words from 'N7:10' to 'N7:17'.

### Status Bits

- **EN**: enable - true when the function is enabled
- **DN**: done - set when the sequencer is full
- **ER**: error - set if an error has occurred

### Available on:

Micrologix, PLC-5

### SQL - SeQuencer Input

<table>
<thead>
<tr>
<th>Length</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

### SQL - SeQuencer Load

<table>
<thead>
<tr>
<th>Length</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0</td>
</tr>
</tbody>
</table>

When the input goes from false to true, the value at the source will be loaded into the sequencer. After the position has reached the length, the following values will be ignored, and the done bit will be set.

### Description

- **EN**: enable - will be true when the input to the function is true
- **DN**: done - will be set when the sequencer is fully loaded
- **ER**: error - will be set when there has been an error

### Registers

None

### Available on:

Micrologix, PLC-5
### Program Control

**SQO**

**SEQUENCER OUTPUT**

**File #N7:10**

**A**

**Status Bits:**

- When the input goes from false to true the sequencer will output a value from a new position in the sequencer table. After the position has reached the length the sequencer will reset to position 1. Note that the first entry in the sequencer table will only be output the first time the function is used, or if reset has been used.

**Description:**

- **Enable** - will be true when the input to the function is true
- **Done** - will be set when the sequencer is fully loaded
- **Error** - will be set when there has been an error

**Registers:**

- none

**Available on:** Micrologix, PLC-5

---

**SQO - SeQuencer Output**

**Ma*sk**

**De*st**

**Control**

**Length**

**Posi*tion**

**FF00**

**N7:0**

**R6:0**

**6**

**0**

---

**EO T**

**A**

**Status Bits:**

This function will cause a transition in an SFC. This will be in a program file for an SFC step. When 'A' becomes true the transition will end and the SFC will move to the next step and transitions.

**Description:**

- none

**Registers:**

- none

**Available on:** PLC-5

---

**EOT - End Of Transition**
plc function ref - 34.31

FOR
FOR
Label Number
Index
Initial Value
Terminal Value

0
N7:0
0
10

A

Status Bits:
This instruction will create a loop like traditional programming languages with a start and end value with a step size for each loop. Instructions between the FOR and NXT will be repeated. If the line with the BRK statement becomes true, the NXT command will be ignored.

Description:
none

Registers: none

Available on: Micrologix, PLC-5

FOR, NXT, BRK - For, Next, Break
The JSR will jump to another program file and pass a list of arguments that can be a variable length. The first statement in the subroutine program file should be SBR to retrieve the arguments passed. The subroutine will end with the RET command that will go back to where the JSR function was encountered. The RET function can return a variable number of arguments.

Status Bits:

Description:

none

Registers: none

Available on: Micrologix, PLC-5
Status Bits:
This function will reset a SFC. In this example when 'A' goes true the SFC main program stored in program file 3 will be examined. All subprograms will be examined, and then the SFC will be reset to the initial position.

Description:
none

Registers:
none

Available on:
PLC-5

UID, UIE - User Interrupt Disable, User Interrupt Enable
34.1.11 Advanced Input/Output

BTR

BLOCK TRANSFER READ

Rack

Group

Module

Control Block

2

3

0

BT10:2

Status Bits:

These instructions communicate with complex input-output cards in a PLC rack. The instruction is needed when a card requires more than one word of input and/or output data. The rack and group indicate the location of the card as 'O:023'. The module number is needed when using two slot addressing for larger racks (this is not needed for racks with less than 8 cards). The control memory is 'BT', although integer memory could also be used. The data file indicates the location of the data to be sent, in this case it is from 'N9:10' to 'N9:22'. The length and contents of the data file are dependant upon the card type. If the instruction is continuous, it will send out the data as soon as the last transmission is complete. If it is not continuous 'A ' must go from false to true to trigger a transmission.

Description:

EN

ST

DN

ER

CO

EW

NR

TO

RW

enable -

start -

done -

error -

continuous -

enable waiting -

no response -

time out -

read write -

Registers:

RLEN

DLEN

FILE

ELEM

RGS

requested data length -

transmitted data length -

file number -

element number -

rack, group, slot - card address

Available on: Micrologix, PLC-5

BTR, BTW - Block Transfer Read, Block Transfer Write
This is a multipurpose instruction that deals with communications in general. The instruction is controlled by the contents of the control block, which is normally set up using the programming software. The instruction can send and receive data across most interfaces including DH, DH+, Ethernet, RS-232, RS-422 and RS-485. The message blocks 'MG' are preferred for storing the configuration, but integer memory may also be used. The messages are segments of PLC memory. These can be read from, or written to a remote destination.

**Description:**
- **Enable** - indicates when the instruction is active
- **Start**
- **Done** - indicates when the instruction is complete
- **Error** - an error occurred
- **Continuous** - when set the instruction doesn’t need a true input
- **Waiting**
- **No response** - the remote destination was not detected
- **Time out** - the remote destination did not respond in time

**Registers:** many refer to manuals

**Available on:** Micrologix, PLC-5

**MSG** - Message
This function calculates a value for a control output based on a feedback value. When 'A' is true the instruction will do a PID calculation. In this example the PID calculation is based on the parameters stored in 'PD9:0'. It will use the setpoint 'PD9:0.SP', and the feedback value 'N10:0' to calculate a new control output 'N10:30'. The control variables are normally set using the programming software, although it is possible to set up this instruction using MOV instructions.

**Description:**

- **EN**: enable - indicates when the input is active
- **DN**: done - this indicates when the instruction is done (not available when using the 'PD' control block.

**Registers:**

- **KC**: controller gain - the overall gain for the controller
- **TI**: reset time - this gives a relative time for integration
- **TD**: rate time - this gives a relative time for the derivative
- **MAXS**: maximum setpoint
- **MINS**: minimum setpoint
- **SP**: setpoint - the setpoint for the process

*Note: This is only a partial list, see the manuals for additional status bits and registers.*

**Available on:** PLC-5

**PID**: Proportional Integral Derivative controller
34.1.12 String ABL

ASCII TE ST FOR LINE Channel Control

The ABL instruction checks for available characters in the input buffer. In this example, when 'A' goes true the function will check the input buffer for channel '0' and put characters in 'R6:0.POS'. The count will include end of line characters such as 'CR' and 'LF'.

The ACB instruction is the same, except that it does not include the end of line characters.

Description: none

Registers: POS the number of characters waiting in the buffer.

Available on: Micrologix, PLC-5

ABL, ACB - Ascii available Line, Ascii Characters in Buffer
STRING TO INTEGER CONVERSION

Status Bits:
The ACI instruction will convert a string to an integer value. In this example it retrieves the string in 'ST10:2', converts it to an integer and stores it in 'N9:5'. When converting to an integer it is possible to have an overflow error.

AIC function will convert an integer to a string.

Description:

Carry - sets if a carry is generated
Overflow - only set if value exceeds maximum for number type
Zero - sets if the result is zero.
Sign - sets if the result is negative

Registers: POS the number of characters waiting in the buffer.

Available on: Micrologix, PLC-5

ACN - Ascii string Concatenate

Status Bits:
This will concatenate two strings together into one combined string. In this example while 'A' is true the strings in 'ST10:0' and 'ST10:1' will be added together and stored in 'ST10:2'.

Description:

none

Registers: none

Available on: Micrologix, PLC-5
AE X - ASCII String Extract

Status Bits:
This function will remove part of a string. In this example the characters in the 12th, 13th and 14th positions ('3' characters starting at the 11th position), are copied to the location ST9:0. The original string is not changed.

Description:
none

Registers: none
Available on: Micrologix, PLC-5

AHL - ASCII Handshake Line

Status Bits:
This instruction will check the serial interface using the DTR and RTS send bits. Bit 0 is DTR and bit 1 is the RTS. If a bit is set in the AND mask the bits will be turned off, otherwise they will be left alone. If a bit is set in the OR mask a bit will be turned on, otherwise they will be left alone. In this example the DTR and RTS bits will be turned on for channel 1.

Description:
EN
DN
ER
enable - this is set when the instruction is active
done - when the bits have been reset this bit is on
error - this bit is set if an error has occurred

Registers: none
Available on: Micrologix, PLC-5
The ARD instruction will read characters and write them to a string. In this example the characters are read from channel 0 and written to 'ST10:0'. All of the characters in the buffer, up to 15 in total, will be removed and written to the string memory. The number of characters will be stored in 'R6:10.POS'.

The ARL function is similar to the ARD function, except that the end-of-line values 'CR' or 'LF' will mark the end of a line. With the parameters above the string will be copied until 15 characters are reached, or there are fewer than 15 characters, or an end-of-line character is found.

**Description:**

- **EN**
  - enable - will be set while the instruction is enabled
  - done - will be set when the string has been read
  - error - will be set if an error has occurred
  - unload -
  - empty - will be set if no characters were found
  - queue -

- **DN**

- **ER**

- **UL**

- **EM**

- **EU**

**Registers:**

- **POS** the number of characters copied

**Available on:** Micrologix, PLC-5
ASC - ASCII String Search for Character

Description:
S2:17/8 ascii minor fault bit - this bit will be set if there was no match.

Registers: none
Available on: Micrologix, PLC-5

Result
ST9:1

Status Bits:
This function will search a string for a character. In this example the character will look for the character in string 'ST9:0' in position 20 (21st) in string 'ST9:1'. If a match is NOT found the bit 'S2:17/8' will be turned on.

ASR - ASCII String Compare

Description:
S2:17/8 ascii minor fault bit - this bit will be set if the string lengths don't match.

Registers: none
Available on: Micrologix, PLC-5

Status Bits:
This instruction will compare two strings. In this example, if 'A' is true then the strings 'ST10:10' and 'ST10:11' will be compared. If they are equal then 'X' will be true, otherwise it will be false. If the strings are different lengths then the bit 'S2:17/8' will be set.
The following table describes the arguments and return values for functions. Some notes are:

- 'immediate' values are numerical, not memory addresses.
- 'returns' indicates that the function returns that data value.
- Numbers between '[' and ']' indicate a range of values.
- Values such as 'yes' and 'no' are typed in literally.

### Table: ASCII WRITE

<table>
<thead>
<tr>
<th>Channel</th>
<th>Source</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ST11:9</td>
<td>A</td>
</tr>
</tbody>
</table>

**Status Bits:**

- enable: set while the instruction is active
- done: set after the string has been sent
- error bit: set when an error has occurred
- unload:
- empty: set if no string was found
- queue:

**Registers:**

- POS the number of characters sent instructions

**Available on:** Micrologix, PLC-5

**Description:**

- AWT, AWA - Ascii Write, Ascii Write Append

<table>
<thead>
<tr>
<th>String Length</th>
<th>Characters Sent</th>
</tr>
</thead>
<tbody>
<tr>
<td>R6:3</td>
<td>14</td>
</tr>
</tbody>
</table>
### Table 1: Instruction Data Types

<table>
<thead>
<tr>
<th>Function</th>
<th>Argument</th>
<th>Data Types</th>
<th>Return</th>
<th>Trigger</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABL channel</td>
<td>control characters</td>
<td>immediate int [0-4]</td>
<td>R</td>
<td>yes</td>
</tr>
<tr>
<td>ACB channel</td>
<td>control characters</td>
<td>immediate int [0-4]</td>
<td>R</td>
<td>yes</td>
</tr>
<tr>
<td>ACI source</td>
<td>destination</td>
<td>ST</td>
<td>N</td>
<td>no</td>
</tr>
<tr>
<td>ACS source</td>
<td>destination</td>
<td>N,F,immediate</td>
<td>N,F</td>
<td>no</td>
</tr>
<tr>
<td>ADD source A</td>
<td>source B</td>
<td>destination</td>
<td>N,F,immediate</td>
<td>N,F</td>
</tr>
<tr>
<td>AEX source</td>
<td>index</td>
<td>number</td>
<td>destination</td>
<td>ST</td>
</tr>
<tr>
<td>AFI no</td>
<td>AHL channel</td>
<td>AND mask</td>
<td>OR mask</td>
<td>control</td>
</tr>
<tr>
<td>AIC source</td>
<td>destination</td>
<td>N, immediate int</td>
<td>ST</td>
<td>no</td>
</tr>
<tr>
<td>ARD channel</td>
<td>destination</td>
<td>control</td>
<td>string length</td>
<td>characters read</td>
</tr>
</tbody>
</table>
Table 1: Instruction Data Types

<table>
<thead>
<tr>
<th>Function</th>
<th>Argument Data Types</th>
<th>Edge Triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td>plc function</td>
<td>ref</td>
<td></td>
</tr>
<tr>
<td>ARL channel</td>
<td></td>
<td></td>
</tr>
<tr>
<td>destination</td>
<td>control</td>
<td></td>
</tr>
<tr>
<td>string length</td>
<td>characters read</td>
<td></td>
</tr>
<tr>
<td>immediate int</td>
<td>[0-4]</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>ASC source</td>
<td>index</td>
<td></td>
</tr>
<tr>
<td>search</td>
<td>result</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>N</td>
<td>immediate</td>
</tr>
<tr>
<td>ST</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>ASN source</td>
<td>A</td>
<td>F, N</td>
</tr>
<tr>
<td>source</td>
<td>destination</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>ST</td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASR source A</td>
<td>source B</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>ST</td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A TN source</td>
<td>destination</td>
<td></td>
</tr>
<tr>
<td>N,F</td>
<td>immediate</td>
<td></td>
</tr>
<tr>
<td>N,F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A VE file</td>
<td>destination</td>
<td>control</td>
</tr>
<tr>
<td>length</td>
<td>position</td>
<td></td>
</tr>
<tr>
<td>#F</td>
<td>#N</td>
<td>F, N</td>
</tr>
<tr>
<td>N</td>
<td>immediate int</td>
<td></td>
</tr>
<tr>
<td>returns N</td>
<td></td>
<td>yes</td>
</tr>
<tr>
<td>A WA channel</td>
<td>source</td>
<td>control</td>
</tr>
<tr>
<td>string length</td>
<td>characters sent</td>
<td></td>
</tr>
<tr>
<td>immediate int</td>
<td>[0-4]</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>A WT channel</td>
<td>source</td>
<td>control</td>
</tr>
<tr>
<td>length</td>
<td>characters sent</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>immediate int</td>
<td></td>
</tr>
<tr>
<td>ST</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>BSL file</td>
<td>control</td>
<td>bit address</td>
</tr>
<tr>
<td>bit address</td>
<td>length</td>
<td></td>
</tr>
<tr>
<td>#B</td>
<td>#N</td>
<td>R</td>
</tr>
<tr>
<td>any bit</td>
<td>immediate int</td>
<td></td>
</tr>
<tr>
<td>returns N</td>
<td></td>
<td>yes</td>
</tr>
</tbody>
</table>
plc function ref - 34.45

BSR file
control
bit address

length

#B,#N

R

any bit

immediate int [0-16000]
yes

BTD source

source bit

destination

destination bit

length

N,B,immediate

N,immediate int [0-15]

N

immediate int [0-15]
immediate int [0-15]

immediate int [0-15]

no

BTR rack

group

module

control block

data file

length

continuous

immediate octal [000-277]
immediate octal [0-7]
immediate octal [0-1]

BT ,N

N

immediate int [0-64]

'yes','no'
yes

BTW rack

group

module

control block

data file

length

continuous

immediate octal [000-277]
immediate octal [0-7]
immediate octal [0-1]

BT ,N

N

immediate int [0-64]

'yes','no'
yes

CLR destination N,F no

CMP expression expression no

COP source

destination

length

#any

#any

immediate int [0-1000]

no

COS source

destination

F ,immediate

F

no

CPT destination

expression

N,F

expression

no

CTD counter

preset

accumulated

C

returns N

returns N

yes

<table>
<thead>
<tr>
<th>Function</th>
<th>Argument Data Types</th>
<th>Edge Triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Instruction Data Types
Table 1: Instruction Data Types

<table>
<thead>
<tr>
<th>Function</th>
<th>Argument Data Types</th>
<th>Edge Triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTU</td>
<td>Preset, Accumulated</td>
<td>Yes</td>
</tr>
<tr>
<td>DDT</td>
<td>Source, Reference</td>
<td>Yes</td>
</tr>
<tr>
<td>Result</td>
<td>Compare, Control, Length, Position</td>
<td>Yes</td>
</tr>
<tr>
<td>Table</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Function Argument Data Types</th>
<th>Edge Triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 1: Instruction Data Types

<table>
<thead>
<tr>
<th>Function Argument Data Types</th>
<th>Edge Triggered</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>