

**Operation:**(Applicable to all units)

The contents of the two source devices (S1, S2) are multiplied together and the result is stored at the destination device (D). Note the normal rules of algebra apply.

## Points to note:

a) When operating the MUL instruction in 16bit mode, two 16 bit data sources are multiplied together. They produce a 32 bit result. The device identified as the destination address is the lower of the two devices used to store the 32 bit result.

Using the above example with some test data:

$5(D0) \times 7(D2) = 35$  - The value 35 is stored in (D4, D5) as a single 32 bit word.

b) When operating the MUL instruction in 32 bit mode, two 32 bit data sources are multiplied together. They produce a 64 bit result. The device identified as the destination address is the lower of the four devices used to store the 64 bit result.

c) If the location of the destination device is smaller than the obtained result, then only the portion of the result which directly maps to the destination area will be written, i.e if a result of 72 (decimal) is to be stored at K1Y4 then only Y7 would be active. In binary terms this is equivalent to a decimal value of 8, a long way short of the real result of 72!



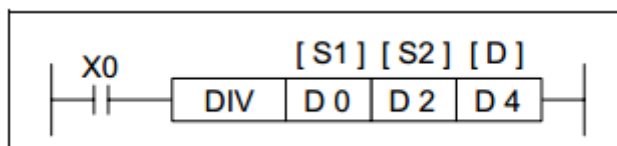
**Viewing 64 bit numbers**

- It is currently impossible to monitor the contents of a 64 bit result. However, the result can be monitored in two smaller, 32 bit, blocks, i.e. a 64 bit result is made up of the following parts: (upper 32 bits)  $\times 2^{32}$  + (lower 32 bits).

#### 5.3.4 DIV (FNC 23)

| Mnemonic                    | Function                                                                           | Operands                                                                                                                                                                                                              |    |                                                        | Program steps         |
|-----------------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|--------------------------------------------------------|-----------------------|
|                             |                                                                                    | S1                                                                                                                                                                                                                    | S2 | D                                                      |                       |
| DIV<br>FNC 23<br>(Division) | Divides one source value by another the result is stored in the destination device | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z                                                                                                                                                                               |    | KnY, KnM, KnS, T, C, D, Z(V)                           | DIV, DIVP: 7 steps    |
|                             |                                                                                    | See page 4-46 for more details regarding floating point format.<br><br>When using M8023 to subtract floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used. used to perform |    | Note: Z(V) may <b>NOT</b> be used for 32 bit operation | DDIV, DDIVP: 13 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|
|------------------|------------------|---------|



Operation: (Applicable to all units)

The primary source (S1) is

divided by the secondary source (S2). The result is stored in the destination (D). Note the normal rules of algebra apply.

#### Points to note:

- When operating the DIV instruction in 16bit mode, two 16 bit data sources are divided into each other. They produce two 16 bit results. The device identified as the

destination address is the lower of the two devices used to store the these results.

This storage device will actually contain a record of the number of whole times S2 will divide into S1 (the quotient).

The second, following destination register contains the remained left after the last whole division (the remainder). Using the previous example with some test data:

$$51 (D0) \div 10 (D2) = 5 (D4) 1 (D5)$$

This result is interpreted as 5 whole divisions with 1 left over ( $5 \times 10 + 1 = 51$ ).

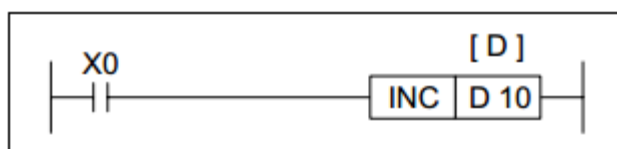
b) When operating the DIV instruction in 32 bit mode, two 32 bit data sources are divided into each other. They produce two 32 bit results. The device identified as the destination address is the lower of the two devices used to store the quotient and the following two devices are used to store the remainder, i.e. if D30 was selected as the destination of 32 bit division operation then D30, D31 would store the quotient and D32, D33 would store the remainder. If the location of the destination device is smaller than the obtained result, then only the portion of the result which directly maps to the destination area will be written. If bit devices are used as the destination area, no remainder value is calculated.

c) If the value of the source device S2 is 0 (zero) then an operation error is executed and the operation of the DIV instruction is cancelled.

### 5.3.5 INC (FNC 24)

| Mnemonic                          | Function                                                                        | Operands                                                                         | Program steps                                          |
|-----------------------------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------|--------------------------------------------------------|
|                                   |                                                                                 | D                                                                                |                                                        |
| INC<br>FNC 24<br>(Increment)<br>➔ | The designated device is incremented by 1 on every execution of the instruction | KnY, KnM, KnS,<br>T, C, D, V, Z<br>Standard V,Z rules apply for 32 bit operation | INC,INCP:<br>3 steps<br><br>DINC,<br>DINCP:<br>5 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

On every execution of the instruction the device specified as the destination D, has its current value incremented (increased) by a value of 1.

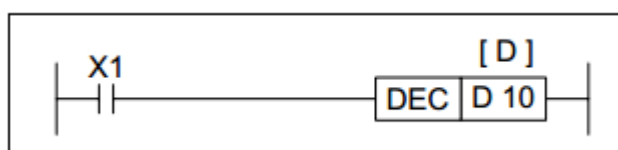
In 16 bit operation, when +32,767 is reached, the next increment will write a value of -32,768 to the destination device.

In 32 bit operation, when +2,147,483,647 is reached the next increment will write a value of -2,147,483,648 to the destination device. In both cases there is no additional flag to identify this change in the counted value.

### 5.3.6 DEC (FNC 24)

| Mnemonic                          | Function                                                                        | Operands                                                                         | Program steps                                          |
|-----------------------------------|---------------------------------------------------------------------------------|----------------------------------------------------------------------------------|--------------------------------------------------------|
|                                   |                                                                                 | D                                                                                |                                                        |
| DEC<br>FNC 25<br>(Decrement)<br>➔ | The designated device is decremented by 1 on every execution of the instruction | KnY, KnM, KnS,<br>T, C, D, V, Z<br>Standard V,Z rules apply for 32 bit operation | DEC,DECP:<br>3 steps<br><br>DDEC,<br>DDECP:<br>5 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

On every execution of the instruction the device specified as the destination D, has its current value decremented (decreased) by a value of 1.

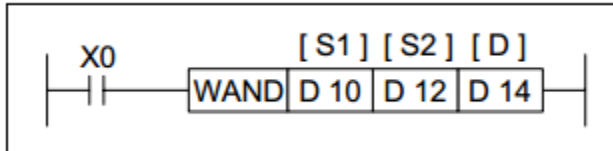
In 16 bit operation, when -32,768 is reached the next increment will write a value of +32,767 to the destination device.

In 32 bit operation, when -2,147,483,648 is reached the next increment will write a value of +2,147,483,647 to the destination device. In both cases there is no additional flag to identify this change in the counted value.

### 5.3.7 WAND (FNC 26)

| Mnemonic                             | Function                                                                        | Operands                                         |    |                                 | Program steps                                         |
|--------------------------------------|---------------------------------------------------------------------------------|--------------------------------------------------|----|---------------------------------|-------------------------------------------------------|
|                                      |                                                                                 | S1                                               | S2 | D                               |                                                       |
| WAND<br>FNC 26<br>(Logical word AND) | A logical AND is performed on the source devices - result stored at destination | K, H,<br>KnX, KnY,<br>KnM, KnS,<br>T, C, D, V, Z |    | KnY, KnM, KnS,<br>T, C, D, V, Z | WAND,WANDP:<br>7 steps<br>DAND,<br>DANDP:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The bit patterns of the two source devices are analyzed (the contents of S2 is compared against the contents of S1). The result of the logical AND analysis is stored in the destination device (D).

The following rules are used to determine the result of a logical AND operation. This takes place for every bit contained within the source devices: General rule: (S1) Bit n WAND (S2) Bit n = (D) Bit n

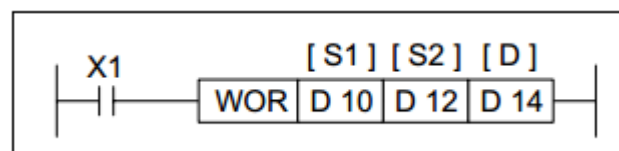
1 WAND 1 = 1 0 WAND 1 = 0

1 WAND 0 = 0 0 WAND 0 = 0

## 5.3.8 WOR (FNC 27)

| Mnemonic                           | Function                                                                       | Operands                                |    |                              | Program steps                             |
|------------------------------------|--------------------------------------------------------------------------------|-----------------------------------------|----|------------------------------|-------------------------------------------|
|                                    |                                                                                | S1                                      | S2 | D                            |                                           |
| WOR<br>FNC 27<br>(Logical word OR) | A logical OR is performed on the source devices - result stored at destination | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z |    | KnY, KnM, KnS, T, C, D, V, Z | WOR, WORP: 7 steps<br>DOR, DORP: 13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The bit patterns of the two source devices are analyzed (the contents of S2 is compared against the contents of S1). The result of the logical OR

analysis is stored in the destination device (D).

The following rules are used to determine the result of a logical OR operation. This takes place for every bit contained within the source devices: General rule:

$(S1)Bitn \text{ WOR } (S2)Bitn = (D)Bitn$

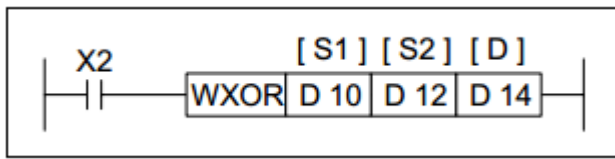
$1 \text{ WOR } 1 = 1$   $0 \text{ WOR } 1 = 1$

$1 \text{ WOR } 0 = 1$   $0 \text{ WOR } 0 = 0$

### 5.3.9 WXOR (FNC 28)

| Mnemonic                                       | Function                                                                        | Operands                                     |    |                                 | Program steps                                        |
|------------------------------------------------|---------------------------------------------------------------------------------|----------------------------------------------|----|---------------------------------|------------------------------------------------------|
|                                                |                                                                                 | S1                                           | S2 | D                               |                                                      |
| WXOR<br>FNC 28<br>(Logical<br>exclusive<br>OR) | A logical XOR is performed on the source devices - result stored at destination | K, H<br>KnX, KnY, KnM, KnS,<br>T, C, D, V, Z |    | KnY, KnM, KnS,<br>T, C, D, V, Z | WXOR,<br>WXORP:<br>7 steps<br>DXOR,DXORP<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



## Operation:

The bit patterns of the two source devices are analyzed (the

source devices are analyzed (the

contents of S2is compared against the contents of S1). The result of the logical XOR analysis is stored in the destination device (D).

The following rules are used to determine the result of a logical XOR operation. This takes place for every bit contained within the source devices: General rule: (S1)Bitn

nWXOR (S2)Bitn= (D)Bitn

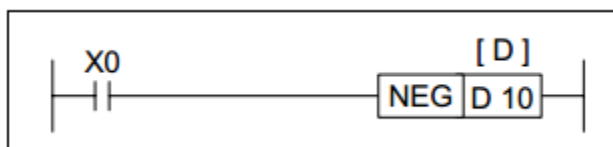
1WXOR1=0 0WXOR1=1

1WXOR0=1 0WXOR0=0

## 5.3.10 NEG (FNC 29)

| Mnemonic                         | Function                                                | Operands                        | Program steps                                      |
|----------------------------------|---------------------------------------------------------|---------------------------------|----------------------------------------------------|
|                                  |                                                         | D                               |                                                    |
| NEG<br>FNC 29<br>(Negation)<br>→ | Logically inverts the contents of the designated device | KnY, KnM, KnS,<br>T, C, D, V, Z | NEG,NEGP:<br>3 steps<br>DNEG,<br>DNEGP:<br>5 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



## Operation:

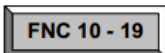
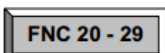
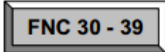
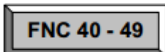
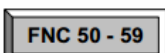
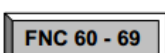
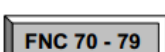
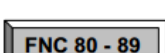
The bit pattern of the selected

device is inverted.

This means any occurrence of a ‘1’ becomes a ‘0’ and any occurrence of a ‘0’ will be written as a ‘1’ .

When this is complete, a further binary 1 is added to the bit pattern. The result is the total logical sign change of the selected devices contents, e.g. a positive number will become a negative number or a negative number will become a positive.

#### Applied Instructions:

|                                                                                   |     |                                                                                     |                                                              |       |
|-----------------------------------------------------------------------------------|-----|-------------------------------------------------------------------------------------|--------------------------------------------------------------|-------|
|                                                                                   | 1.  |    | Program Flow                                                 | 5-4   |
|                                                                                   | 2.  |    | Move And Compare                                             | 5-16  |
|                                                                                   | 3.  |    | Arithmetic And Logical Operations (+, -, $\times$ , $\div$ ) | 5-24  |
|  | 4.  |    | Rotation And Shift                                           | 5-34  |
|                                                                                   | 5.  |    | Data Operation                                               | 5-42  |
|                                                                                   | 6.  |  | High Speed Processing                                        | 5-52  |
|                                                                                   | 7.  |  | Handy Instructions                                           | 5-66  |
|                                                                                   | 8.  |  | External FX I/O Devices                                      | 5-80  |
|                                                                                   | 9.  |  | External FX Serial Devices                                   | 5-94  |
|                                                                                   | 10. |  | Floating Point 1 & 2                                         | 5-110 |
|                                                                                   | 11. |  | Trigonometry (Floating Point 3)                              | 5-118 |
|                                                                                   | 12. |  | Data Operations 2                                            | 5-122 |
|                                                                                   | 13. |  | Positioning Control                                          | 5-126 |
|                                                                                   | 14. |  | Real Time Clock Control                                      | 5-136 |
|                                                                                   | 15. |  | Gray Codes                                                   | 5-146 |
|                                                                                   | 16. |  | Additional Functions                                         | 5-146 |
|                                                                                   | 17. |  | In-line Comparisons                                          | 5-150 |



## Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D1,S3or for lists/abled devices D3+0,S+9etc. MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆- An instruction operating in 16 bit mode, where identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆- An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation



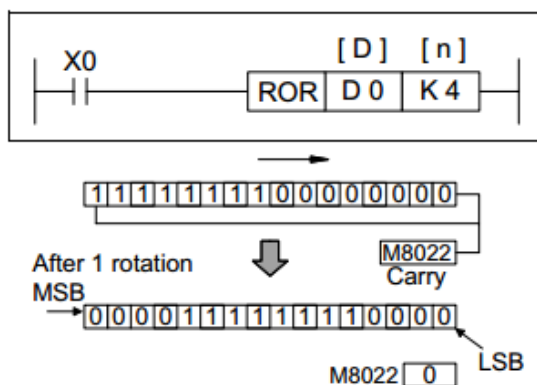
- A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

- ☒ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

#### 5.4.1 ROR (FNC 30)

| Mnemonic                               | Function                                                                                        | Operands                                                                                   |                                                                           | Program steps                                  |
|----------------------------------------|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|------------------------------------------------|
|                                        |                                                                                                 | D                                                                                          | n                                                                         |                                                |
| ROR<br>FNC 30<br>(Rotation right)<br>✈ | The bit pattern of the destination device is rotated 'n' places to the right on every execution | KnY, KnM, KnS, T, C, D, V, Z<br>Note:<br>16 bit operation Kn=K4,<br>32 bit operation Kn=K8 | K, H,<br>☒<br>Note:<br>16 bit operation n ≤ 16<br>32 bit operation n ≤ 32 | ROR, RORP: 5 steps<br><br>DROR, DRORP: 9 steps |

|                  |                  |         |       |             |
|------------------|------------------|---------|-------|-------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Carry M8022 |
|------------------|------------------|---------|-------|-------------|



#### Operation:

The bit pattern of the destination device (D) is rotated n bit places to the right on every operation of the instruction.

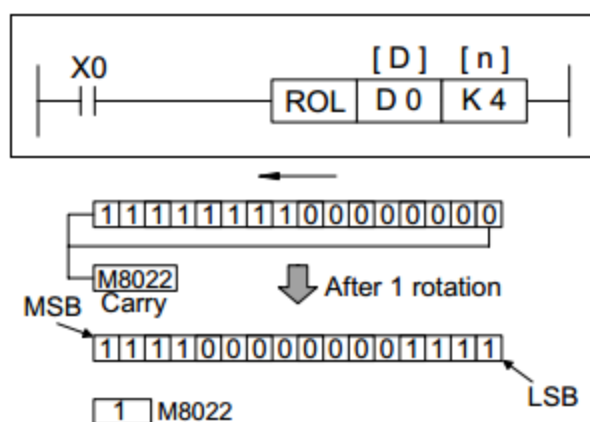
The status of the last bit rotated is copied to the carry flag M8022.

The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

## 5.4.2 ROL (FNC 31)

| Mnemonic                              | Function                                                                                       | Operands                                                                                     |                                                                                         | Program steps                                  |
|---------------------------------------|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|------------------------------------------------|
|                                       |                                                                                                | S                                                                                            | D                                                                                       |                                                |
| ROL<br>FNC 31<br>(Rotation left)<br>→ | The bit pattern of the destination device is rotated 'n' places to the left on every execution | KnY, KnM, KnS, T, C, D, V, Z<br>Note:<br>16 bit operation Kn= K4,<br>32 bit operation Kn= K8 | K, H,<br>☒<br><br>Note:<br>16 bit operation $n \leq 16$<br>32 bit operation $n \leq 32$ | ROL, ROLP: 5 steps<br><br>DROL, DROLP: 7 steps |

|                  |                  |         |       |             |
|------------------|------------------|---------|-------|-------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Carry M8022 |
|------------------|------------------|---------|-------|-------------|



Operation:

The bit pattern of the destination device (D) is rotated n bit places to the left on every operation of the instruction.

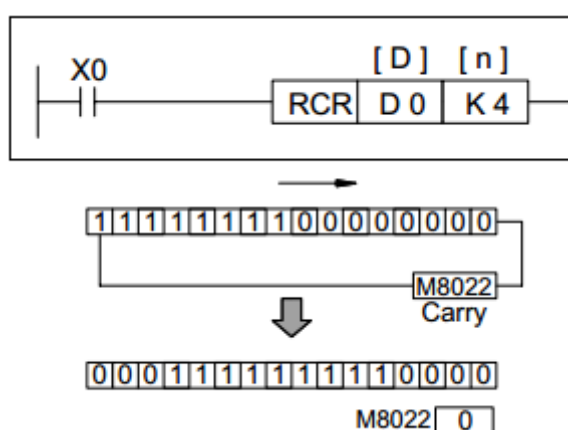
The status of the last bit rotated is copied to the carry flag M8022.

The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

## 5.4.3 RCR (FNC 32)

| Mnemonic                                          | Function                                                                                        | Operands                                                                                    |                                                                                         | Program steps                                  |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|------------------------------------------------|
|                                                   |                                                                                                 | D                                                                                           | n                                                                                       |                                                |
| RCR<br>FNC 32<br>(Rotation right with carry)<br>→ | The contents of the destination device are rotated right with 1 bit extracted to the carry flag | KnY, KnM, KnS, T, C, D, V, Z<br>Note:<br>16 bit operation Kn= K4,<br>32 bit operation Kn=K8 | K, H,<br>☒<br><br>Note:<br>16 bit operation $n \leq 16$<br>32 bit operation $n \leq 32$ | RCR, RCRP: 5 steps<br><br>DRCR, DRCRP: 7 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Carry M8022 |
|------------------|------------------|---------|-------|-------------|
|------------------|------------------|---------|-------|-------------|



Operation:

The bit pattern of the destination device (D) is rotated n bit places to the right on every operation of the instruction.

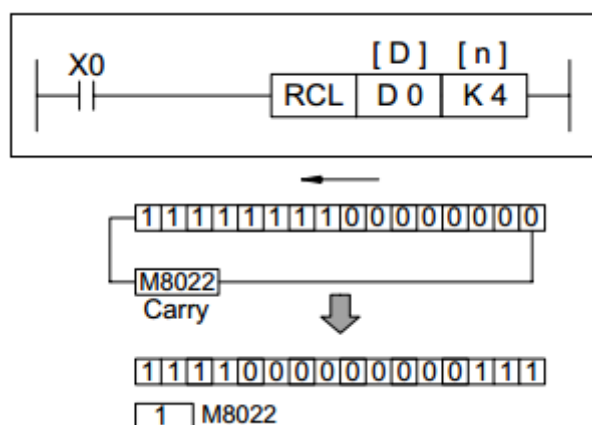
The status of the last bit rotated is moved into the carry flag M8022. On the following operation of the instruction M8022 is the first bit to be moved back into the destination device.

The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

#### 5.4.4 RCL (FNC 33)

| Mnemonic                                         | Function                                                                                       | Operands                                                                                            |                                                                                         | Program steps                                           |
|--------------------------------------------------|------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|---------------------------------------------------------|
|                                                  |                                                                                                | S                                                                                                   | D                                                                                       |                                                         |
| RCL<br>FNC 33<br>(Rotation left with carry)<br>→ | The contents of the destination device are rotated left with 1 bit extracted to the carry flag | KnY, KnM, KnS,<br>T, C, D, V, Z<br><br>Note:<br>16 bit operation Kn= K4,<br>32 bit operation Kn= K8 | K, H,<br>☒<br><br>Note:<br>16 bit operation $n \leq 16$<br>32 bit operation $n \leq 32$ | RCL, RCLP:<br>5 steps<br><br>DRCL,<br>DRCLP:<br>9 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Carry M8022 |
|------------------|------------------|---------|-------|-------------|
|------------------|------------------|---------|-------|-------------|



#### Operation:

The bit pattern of the destination device (D) is rotated  $n$  bit places to the left on every operation of the instruction.

The status of the last bit rotated is

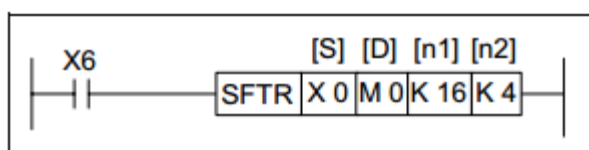
moved into the carry flag M8022. On the following operation of the instruction M8022 is the first bit to be moved back into the destination device.

The example shown left is based on the instruction noted above it, where the bit pattern represents the contents of D0.

## 5.4.5 SFTR (FNC 34)

| Mnemonic                                 | Function                                                                                                    | Operands   |         |                                                                                                                                  |    | Program steps           |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------|------------|---------|----------------------------------------------------------------------------------------------------------------------------------|----|-------------------------|
|                                          |                                                                                                             | S          | D       | n1                                                                                                                               | n2 |                         |
| SFTR<br>FNC 34<br>(Bit shift right)<br>➔ | The status of the source devices are copied to a controlled bit stack moving the existing data to the right | X, Y, M, S | Y, M, S | K, H,<br>☒<br>Note:<br>FX users:<br>$n2 \leq n1 \leq 1024$<br>FX <sub>0</sub> , FX <sub>0N</sub> users:<br>$n2 \leq n1 \leq 512$ |    | SFTR, SFTRP:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

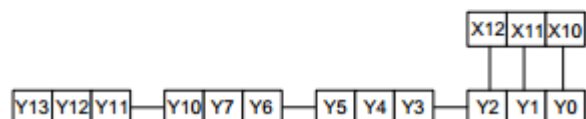
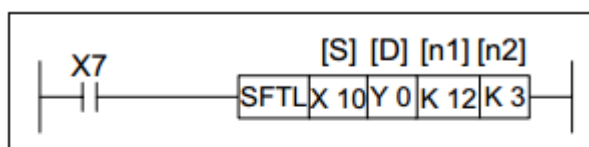


The instruction copies n2source devices to a bit stack of length n1.

For every new addition of n2bits, the existing data within the bit stack is shifted n2bits to the right. Any bit data moving to a position exceeding the n1limit is diverted to an overflow area. The bit shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

## 5.4.6 SFTL (FNC 35)

| Mnemonic                                | Function                                                                                                   | Operands   |         |                                                                                                                               |    | Program steps           |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------|------------|---------|-------------------------------------------------------------------------------------------------------------------------------|----|-------------------------|
|                                         |                                                                                                            | S          | D       | n1                                                                                                                            | n2 |                         |
| SFTL<br>FNC 35<br>(Bit shift left)<br>➔ | The status of the source devices are copied to a controlled bit stack moving the existing data to the left | X, Y, M, S | Y, M, S | K, H,<br>☒<br>Note:<br>FX users: $n2 \leq n1 \leq 1024$<br>FX <sub>0</sub> , FX <sub>0N</sub> users:<br>$n2 \leq n1 \leq 512$ |    | SFTL, SFTLP:<br>9 steps |



|                  |                  |  |
|------------------|------------------|--|
| 16 BIT OPERATION | 32 BIT OPERATION |  |
|------------------|------------------|--|

## Operation:

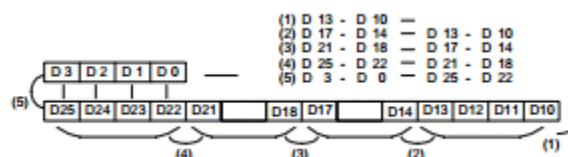
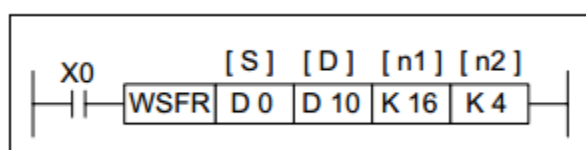
The instruction copies n2source devices to a bit stack of length n1. For every new addition of n2bits, the existing data within the bit stack is shifted n2bits to the left. Any bit data moving to a position exceeding the n1limit is diverted to an overflow area.

The bit shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

### 5.4.7 WSFR (FNC 36)

| Mnemonic                                  | Function                                                                                                    | Operands                    |                        |                                                     |    | Program steps        |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------|-----------------------------|------------------------|-----------------------------------------------------|----|----------------------|
|                                           |                                                                                                             | S                           | D                      | n1                                                  | n2 |                      |
| WSFR<br>FNC 36<br>(Word shift right)<br>➔ | The value of the source devices are copied to a controlled word stack moving the existing data to the right | KnX, KnY, KnM, KnS, T, C, D | KnY, KnM, KnS, T, C, D | K, H, ☒<br>Note:<br>FX users: $n2 \leq n1 \leq 512$ |    | WSFR, WSFRP: 9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



## Operation:

The instruction copies n2source devices to a word stack of length n1.

For each addition of n2words, the existing data within the word stack is shifted n2words

To the right. Any word data moving to a position exceeding the n1limit is diverted to an overflow area.

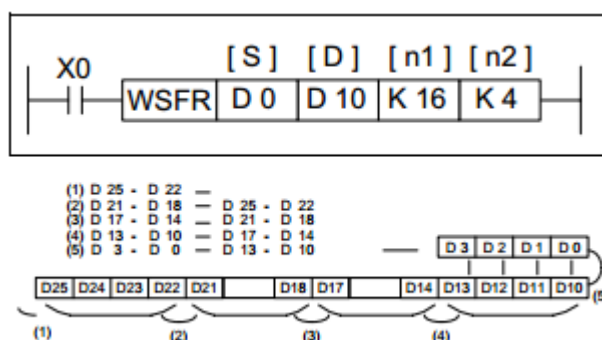
The word shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

**Note:** when using bit devices as source (S) and destination (D) the Kn value must be equal.

#### 5.4.8 WSFL (FNC 37)

| Mnemonic                                 | Function                                                                                                   | Operands                    |                        |                                                        |    | Program steps           |
|------------------------------------------|------------------------------------------------------------------------------------------------------------|-----------------------------|------------------------|--------------------------------------------------------|----|-------------------------|
|                                          |                                                                                                            | S                           | D                      | n1                                                     | n2 |                         |
| WSFL<br>FNC 37<br>(Word shift left)<br>→ | The value of the source devices are copied to a controlled word stack moving the existing data to the left | KnX, KnY, KnM, KnS, T, C, D | KnY, KnM, KnS, T, C, D | K, H, ☒<br>Note:<br>FX users:<br>$n2 \leq n1 \leq 512$ |    | WSFL, WSFLP:<br>9 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|
|------------------|------------------|---------|



**Operation:**

The instruction copies n2source devices to a word stack of length n1. For each addition of n2words, the existing data within the word

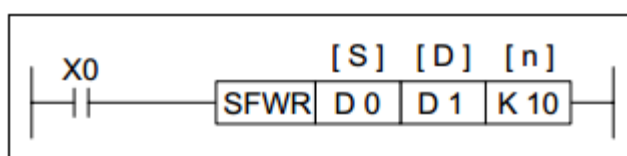
stack is shifted n2words to the left. Any word data moving to a position exceeding the n1limit is diverted to an overflow area.

The word shifting operation will occur every time the instruction is processed unless it is modified with either the pulse suffix or a controlled interlock.

Note: when using bit devices as source (S) and destination (D) the Kn value must be equal.

#### 5.4.9 SFWR (FNC 38)

| Mnemonic                                      | Function                                                                                       | Operands                               |                         |                                      | Program steps        |
|-----------------------------------------------|------------------------------------------------------------------------------------------------|----------------------------------------|-------------------------|--------------------------------------|----------------------|
|                                               |                                                                                                | S                                      | D                       | N                                    |                      |
| SFWR<br>FNC 38<br>(Shift register write)<br>→ | This instruction creates and builds a FIFO stack n devices long -must be used with SFRD FNC 39 | K, H, KnX, KnY, KnM,KnS, T, C, D, V, Z | KnY, KnM, KnS, T, C, D, | K, H, ☒<br>Note: $2 \leq n \leq 512$ | SFWR, SFWRP: 7 steps |
| 16 BIT OPERATION                              |                                                                                                | 32 BIT OPERATION                       | PULSE-P                 | Flags                                | Carry M8022          |



Operation:

The contents of the source device

(S) are written to the FIFO stack.

The position of insertion into the

stack is automatically calculated by the PLC.

The destination device (D) is the head address of the FIFO stack. The contents of D

identify where the next record will be stored (as an offset from D+1).

If the contents of D exceed the value “n-1” (n is the length of the FIFO stack) then insertion into the FIFO stack is stopped. The carry flag M8022 is turned ON to identify this situation.

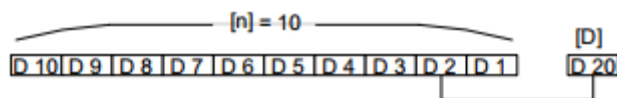
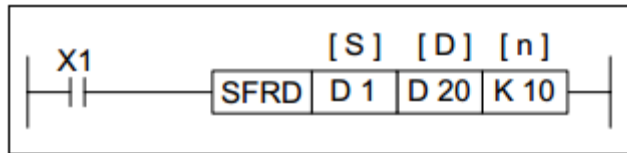
#### Points to note:

- a) FIFO is an abbreviation for ‘First-In/ First-OUT’ .
- b) Although n devices are assigned for the FIFO stack, only n-1 pieces of information may be written to that stack. This is because the head address device (D) takes the first available register to store the information regarding the next data insertion point into the FIFO stack.
- c) Before starting to use a FIFO stack ensure that the contents of the head address register (D) are equal to ‘0’ (zero).
- d) This instruction should be used in conjunction with SFRD FNC 39. The n parameter in both instructions should be equal.

#### 5.4.10 SFRD (FNC 39)

| Mnemonic                                     | Function                                                                     | Operands                                      |                                                     |                                 | Program steps              |
|----------------------------------------------|------------------------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------------|---------------------------------|----------------------------|
|                                              |                                                                              | S                                             | D                                                   | n                               |                            |
| SFRD<br>FNC 39<br>(Shift register read)<br>→ | This instruction reads and reduces FIFO stack- must be used with SFWR FNC 38 | KnY, KnM, KnS, T, C, D, KnY, KnM,KnS, T, C, D | KnY, KnM, KnS, T, C, D, KnY, KnM,KnS, T, C, D, V, Z | K,H,<br>☒<br>Note:<br>2≤ n≤ 512 | SFRD,<br>SFRDP:<br>7 steps |

|                  |                  |         |       |            |
|------------------|------------------|---------|-------|------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Zero M8020 |
|------------------|------------------|---------|-------|------------|



## Operation:

The source device (S) identifies the head address of the FIFO stack. Its contents reflect the last entry point of data on to the FIFO

stack, i.e. where the end of the FIFO is (current position).

This instruction reads the first piece of data from the FIFO stack (register S+1), moves all of the data within the stack ‘up’ one position to fill the read area and decrements the contents of the FIFO head address (S) by 1. The read data is written to the destination device (D).

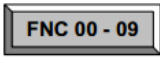
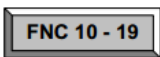
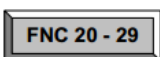
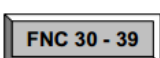
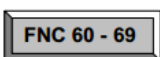
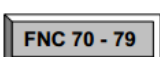
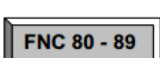
When the contents of the source device (S) are equal to ‘0’ (zero), i.e. the FIFO stack is empty, the flag M8020 is turned ON.

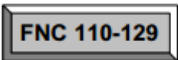
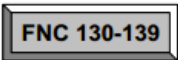
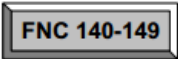
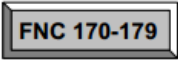
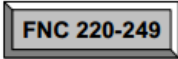
## Points to note:

- FIFO is an abbreviation for ‘First-In/ First-OUT’ .
- Only n-1pieces of data may be read from a FIFO stack. This is because the stack requires that the first register, the head address (S) is used to contain information about the current length of the FIFO stack.
- This instruction will always read the source data from the register S+1.

d) This instruction should be used in conjunction with SFWR FNC 38. The n parameter in both instructions should be equal.

### Applied Instructions:

|                                                                                        |                                                                                     |                                                |      |
|----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------------|------|
| 1.                                                                                     |    | Program Flow                                   | 5-4  |
| 2.                                                                                     |    | Move And Compare                               | 5-16 |
| 3.                                                                                     |   | Arithmetic And Logical Operations (+, -, ×, ÷) | 5-24 |
| 4.                                                                                     |  | Rotation And Shift                             | 5-34 |
|  5. |  | Data Operation                                 | 5-42 |
| 6.                                                                                     |  | High Speed Processing                          | 5-52 |
| 7.                                                                                     |  | Handy Instructions                             | 5-66 |
| 8.                                                                                     |  | External FX I/O Devices                        | 5-80 |
| 9.                                                                                     |  | External FX Serial Devices                     | 5-94 |

|     |                                                                                   |                                 |       |
|-----|-----------------------------------------------------------------------------------|---------------------------------|-------|
| 10. |  | Floating Point 1 & 2            | 5-110 |
| 11. |  | Trigonometry (Floating Point 3) | 5-118 |
| 12. |  | Data Operations 2               | 5-122 |
| 13. |  | Positioning Control             | 5-126 |
| 14. |  | Real Time Clock Control         | 5-136 |
| 15. |  | Gray Codes                      | 5-146 |
| 16. |  | Additional Functions            | 5-146 |
| 17. |  | In-line Comparisons             | 5-150 |



#### Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D1,S3or for lists/abled devices D3+0,S+9etc. MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆- An instruction operating in 16 bit mode, where identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆- An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

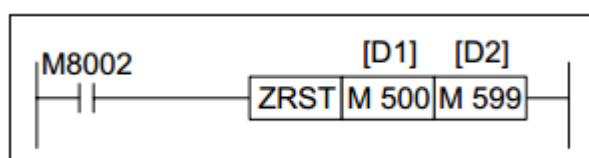
 - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

 - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

### 5.5.1 ZRST (FNC 40)

| Mnemonic                       | Function                                               | Operands                                                                                                                                              |   | Program steps              |
|--------------------------------|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---|----------------------------|
|                                |                                                        | S                                                                                                                                                     | D |                            |
| ZRST<br>FNC 40<br>(Zone Reset) | Used to reset a range of like devices in one operation | Y, M, S, T, C, D<br>Note:<br>D <sub>1</sub> must be less than or equal ( ≤ ) to D <sub>2</sub> .<br>Standard and High speed counters cannot be mixed. |   | ZRST,<br>ZRSTP:<br>5 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|
|------------------|------------------|---------|



Operation:

The range of devices, inclusive of those specified as the two destinations are reset, i.e. for data devices the current value

is set to 0 (zero) and for bit elements, the devices are turned OFF, i.e. also set to 0 (zero).

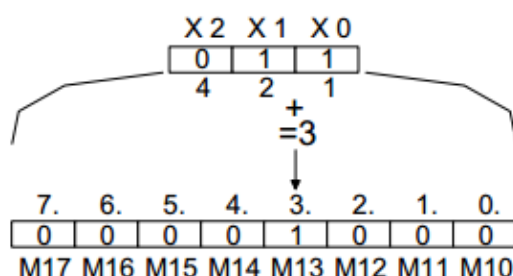
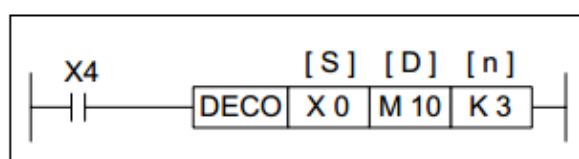
The specified device range cannot contain mixed device types, i.e. C000 specified as the first destination device (D1) cannot be paired with T199 as the second destination device (D2). When resetting counters, standard and high speed counters cannot be reset as part of the same range.

If D1 is greater than (>) D2 then only device D1 is reset.

### 5.5.2 DECO (FNC 41)

| Mnemonic                   | Function                                                                                     | Operands                        |                  |                                                                                                            | Program steps              |
|----------------------------|----------------------------------------------------------------------------------------------|---------------------------------|------------------|------------------------------------------------------------------------------------------------------------|----------------------------|
|                            |                                                                                              | S                               | D                | n                                                                                                          |                            |
| DECO<br>FNC 41<br>(Decode) | Source data value Q identifies the Qth bit of the destination device which will be turned ON | K, H, X, Y, M, S, T, C, D, V, Z | Y, M, S, T, C, D | K, H, ☒<br>Note:<br>D= Y,M,S then n range = 1-8<br>D= T,C,D then n range = 1-4<br>n= 0, then no processing | DECO,<br>DECOP:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

Source data is provided by a combination of operands S and n.

Where S specifies the head address of the data and n, the number of consecutive bits. The source data is

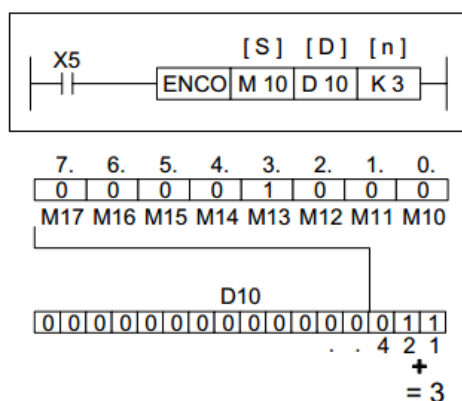
read as a single number (binary to decimal conversion) Q. The source number Q is the

location of a bit within the destination device (D) which will be turned ON (see example opposite). When the destination device is a data device n must be within the range 1 to 4 as there are only 16 available destination bits in a single data word. All unused data bits within the word are set to 0.

### 5.5.3 ENCO (FNC 42)

| Mnemonic                   | Function                                                                                        | Operands                        |                  |                                                                                                                           | Program steps              |
|----------------------------|-------------------------------------------------------------------------------------------------|---------------------------------|------------------|---------------------------------------------------------------------------------------------------------------------------|----------------------------|
|                            |                                                                                                 | S                               | D                | n                                                                                                                         |                            |
| ENCO<br>FNC 42<br>(Encode) | Then location of the highest active bit is stored as a numerical position from the head address | X, Y, M, S,<br>T, C, D, V,<br>Z | T, C, D, V,<br>Z | K, H,<br>☒<br>Note:<br>S=X, Y, M, S then<br>n range=1-8<br>S= T,C,D then<br>n range = 1-4<br>n = 0, then no<br>processing | ENCO,<br>ENCOP:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



#### Operation:

The highest active bit within the readable range has its location noted as a numbered offset from the source head address (S). This is

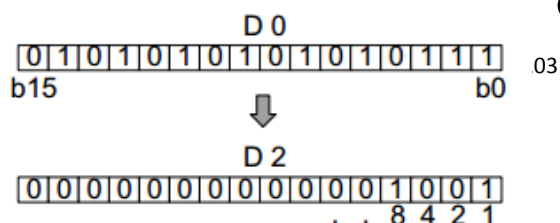
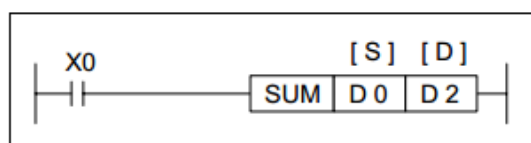
stored in the destination register (D).

#### Points to note:

- a) The readable range is defined by the largest number storable in a binary format within the number of destination storage bits specified by n, i.e. if n was equal to 4 bits a maximum number within the range 0 to 15 can be written to the destination device. Hence, if bit devices were being used as the source data, 16 bit devices would be used, i.e. the head bit device and 15 further, consecutive devices.
- b) If the stored destination number is 0 (zero) then the source head address bit is ON, i.e. the active bit has a 0 (zero) offset from the head address. However, if NO bits are ON within the source area, 0 (zero) is written to the destination device and an error is generated.
- c) When the source device is a data or word device n must be taken from the range 1to4as there are only 16 source bits available within a single data word.

#### 5.5.4 SUM (FNC 43)

| Mnemonic                              | Function                                                                                    | Operands                                |                              | Program steps                                          |
|---------------------------------------|---------------------------------------------------------------------------------------------|-----------------------------------------|------------------------------|--------------------------------------------------------|
|                                       |                                                                                             | S                                       | D                            |                                                        |
| SUM<br>FNC 43<br>(Sum of active bits) | The number (quantity) of active bits in the source data is stored in the destination device | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z | KnY, KnM, KnS, T, C, D, V, Z | SUM,SUMP:<br>7 steps<br><br>DSUM,<br>DSUMP:<br>9 steps |



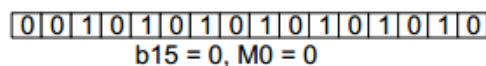
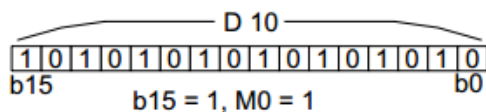
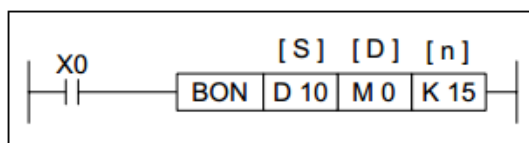
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE |
|------------------|------------------|-------|
|------------------|------------------|-------|

The number of active (ON) bits within the source device (S), i.e. bits which have a value of “1” are counted. The count is stored in the destination register (D). If a double word format is used, both the source and destination devices use 32 bit, double registers. The destination device will always have its upper 16 bits set to 0 (zero) as the counted value can never be more than 32. If no bits are ON then zero flag, M8020 is set.

## 5.5.5 BON (FNC 44)

| Mnemonic                                            | Function                                                                                         | Operands                                            |         |                                                                                      | Program steps                                       |
|-----------------------------------------------------|--------------------------------------------------------------------------------------------------|-----------------------------------------------------|---------|--------------------------------------------------------------------------------------|-----------------------------------------------------|
|                                                     |                                                                                                  | S                                                   | D       | n                                                                                    |                                                     |
| BON<br>FNC 44<br>(Check<br>specified<br>bit status) | The status of the<br>specified bit in the<br>source device is<br>indicated at the<br>destination | K, H,<br>KnX, KnY,<br>KnM, KnS,<br>T, C, D, V,<br>Z | Y, M, S | K,H,<br>☒<br>Note:<br>16 bit operation<br>n=0 to 15<br>32 bit operation<br>n=0 to 31 | BON, BONP:<br>7steps<br>DBONP,<br>DBON:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

A single bit position (n) is specified from within a source device/area (S). n could be regarded as a specified offset from the

source head address (S), i.e. 0 (zero) being the first device (a 0 offset) where as an offset of 15 would actually be the 16th device.

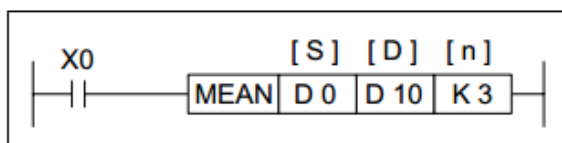
If the identified bit becomes active, i.e. ON, the destination device (D) is activated to “flag” the new status.

The destination device could be said to act as a mirror to the status of the selected bit source.

## 5.5.6 MEAN (FNC 45)

| Mnemonic                 | Function                                  | Operands                          |                                    |                                  | Program steps                                              |
|--------------------------|-------------------------------------------|-----------------------------------|------------------------------------|----------------------------------|------------------------------------------------------------|
|                          |                                           | S                                 | D                                  | n                                |                                                            |
| MEAN<br>FNC 45<br>(Mean) | Calculates the mean of a range of devices | KnX, KnY,<br>KnM, KnS,<br>T, C, D | KnY, KnM,<br>KnS,<br>T, C, D, V, Z | K,H,<br>☒<br>Note:<br>n= 1 to 64 | MEAN,<br>MEANP:<br>7 steps<br>DMEAN,<br>DMEANP:<br>13steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



General rule

$$D = \frac{\sum_{S_0}^{S_n} S}{n} = \frac{(S_0 + S_1 + \dots + S_n)}{n}$$

Example

$$D10 = \frac{(D0) + (D1) + (Dn)}{3}$$

Operation:

The range of source data is defined by operands S and n. S is the head address of the source data and n specifies the number of consecutive source devices used.

The value of all the devices within the source range is summed and then divided by the number of devices summed, i.e. n. This generates an integer mean value which is

stored in the destination device (D). The remainder of the calculated mean is ignored.

#### Points to note:

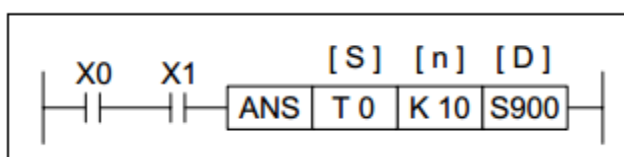
If the source area specified is actually smaller than the physically available area, then only the available devices are used. The actual value of n used to calculate the mean will reflect the used, available devices. However, the value for n which was entered into the instruction will still be displayed. This can cause confusion as the mean value calculated manually using this original n value will be different from that which is displayed.

If the value of n is specified outside of the stated range (1 to 64) an error is generated.

#### 5.5.7 ANS (FNC 46)

| Mnemonic                                 | Function                                                                                | Operands                                    |                                              |                                                              | Program steps   |
|------------------------------------------|-----------------------------------------------------------------------------------------|---------------------------------------------|----------------------------------------------|--------------------------------------------------------------|-----------------|
|                                          |                                                                                         | S                                           | D                                            | n                                                            |                 |
| ANS<br>FNC 46<br>(Timed annunciator set) | This instruction starts a timer. Once timed out the selected annunciator flag is set ON | T<br>Note:<br>available range<br>T0 to T199 | S<br>Note:<br>annunciator range S900 to S999 | K<br>☒<br>Note:<br>n range 1 to 32,767 - in units of 100msec | ANS:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



#### Operation:

This instruction, when energized, starts a timer (S) for

n,100 msec. When the timer completes its cycle the assigned annunciator (D) is set ON.

If the instruction is switched OFF during or after completion of the timing cycle the timer is automatically reset. However, the current status of the annunciator coil remains unchanged.

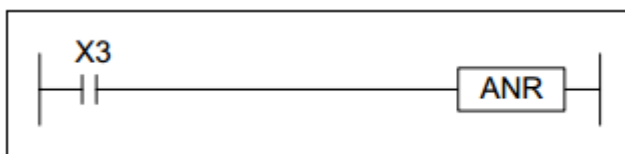


**Note:** This is only one method of driving annunciator coils, others such as direct setting can also be used.

#### 5.5.8 ANR (FNC 47)

| Mnemonic                                  | Function                                                                      | Operands | Program steps      |
|-------------------------------------------|-------------------------------------------------------------------------------|----------|--------------------|
|                                           |                                                                               | D        |                    |
| ANR<br>FNC 47<br>(Annunciator reset)<br>→ | The lowest active annunciator is reset on every operation of this instruction | N/A      | ANR,ANRP:<br>1step |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



**Operation:**

Annunciators which have been activated are sequentially reset

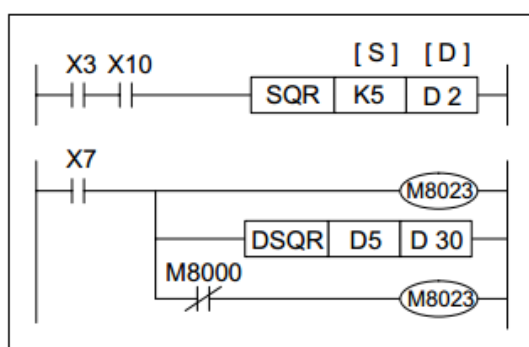
one-by-one, each time the ANR instruction is operated.

If the ANR instruction is driven continuously it will carry out its resetting operation on every program scan unless it is modified by the pulse, P prefix or by a user defined program interlock.

### 5.5.9 SQR (FNC 48)

| Mnemonic                       | Function                                                    | Operands                                                                                                                                               |   | Program steps                                       |
|--------------------------------|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|---|-----------------------------------------------------|
|                                |                                                             | S                                                                                                                                                      | D |                                                     |
| SQR<br>FNC 48<br>(Square root) | Performs a mathematical square root e.g.:<br>$D = \sqrt{S}$ | K,H,D<br><br>When using M8023 in float mode, only double word (32bit) data can be processed. See page 4- 46 for more details regarding floating point. | D | SQR, SQRP:<br>5 steps<br>DSQR,<br>DSQRP:<br>9 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Zero M8020<br>Borrow M8021 |
|------------------|------------------|---------|-------|----------------------------|
|------------------|------------------|---------|-------|----------------------------|



#### Operation 1:

This instruction performs a square root operation on source data (S) and stores the result at destination device (D). The operation is conducted entirely in

whole integers rendering the square root answer rounded to the lowest whole number. For example, if (S) = 154, then (D) is calculated as being 12. M8020 is set ON when the square root operation result is equal to zero. Answers with rounded values will activate M8021.

Operation 2: This function is equivalent to FNC 127 ESQR This operation is similar to Operation 1. However, it is only activated when the mode setting float flag, M8023 is used.

This then allows the SQR instruction to process answers in floating point format. The source data (S) must either be supplied in floating point format for data register use, or it can be supplied as a constant (K,H). When constants are used as a source, they

are automatically converted to floating point format. Operation 2 is only valid for double word (32 bit) operation,

hence both (S) and (D) will be 32 bit values and the SQR instruction will be entered as DSQR or DSQRP.

### General note:

Performing any square root operation (even on a calculator) on a negative number will result in an error. This will be identified by special M coil M8067 being activated:

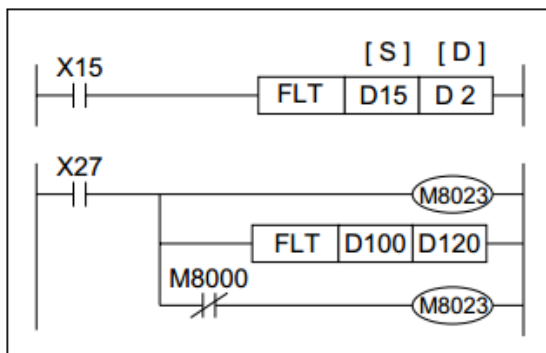
$\sqrt{-168}$  = Error and M8067 will be set ON This is true for both operating modes.

#### 5.5.10 FLT (FNC 49)

| Mnemonic                          | Function                                               | Operands                                                                  |   | Program steps                                       |
|-----------------------------------|--------------------------------------------------------|---------------------------------------------------------------------------|---|-----------------------------------------------------|
|                                   |                                                        | S                                                                         | D |                                                     |
| FLT<br>FNC 49<br>(Floating point) | Used to convert data to and from floating point format | D                                                                         | D | FLT, FLTP:<br>5 steps<br>DFLT,<br>DFLTP:<br>9 steps |
|                                   |                                                        | M8023 = OFF data is converted from decimal to floating point format       |   |                                                     |
|                                   |                                                        | M8023 = ON data is converted from floating point format to decimal format |   |                                                     |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



#### Operation:

When the float instruction is used without the float flag (M8023 = OFF) the source data (S) is converted in to an equivalent value stored in float format

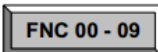
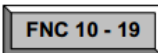
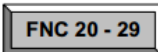
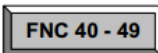
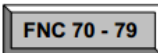
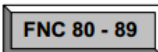
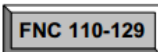
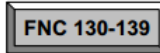
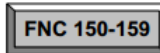
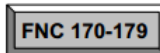
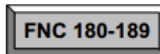
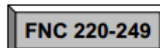
at the destination device (D).

Please note that two consecutive devices (D and

D+1) will be used to store the converted float number. This is true regardless of the size of the source data (S), i.e. whether (S) is a single device (16 bits) or a double device (32 bits) has no effect on the number of destination devices (D) used to store the floating point number. Examples:

| Decimal source data<br>(S) | Floating point<br>destination value (D) |
|----------------------------|-----------------------------------------|
| 1                          | 1                                       |
| -26700                     | $-2.67 \times 10^4$                     |
| 404                        | $4.04 \times 10^2$                      |

Applied Instructions:

|                                                                                   |                                                                                     |                                                                                    |                            |       |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|----------------------------|-------|
| 1.                                                                                |    | Program Flow                                                                       | 5-4                        |       |
| 2.                                                                                |    | Move And Compare                                                                   | 5-16                       |       |
| 3.                                                                                |    | Arithmetic And Logical Operations (+, -, ×, ÷)                                     | 5-24                       |       |
| 4.                                                                                |    | Rotation And Shift                                                                 | 5-34                       |       |
| 5.                                                                                |    | Data Operation                                                                     | 5-42                       |       |
|  | 6.                                                                                  |   | High Speed Processing      | 5-52  |
|                                                                                   | 7.                                                                                  |   | Handy Instructions         | 5-66  |
|                                                                                   | 8.                                                                                  |   | External FX I/O Devices    | 5-80  |
|                                                                                   | 9.                                                                                  |   | External FX Serial Devices | 5-94  |
|                                                                                   | 10.                                                                                 |  | Floating Point 1 & 2       | 5-110 |
| 11.                                                                               |  | Trigonometry (Floating Point 3)                                                    | 5-118                      |       |
| 12.                                                                               |  | Data Operations 2                                                                  | 5-122                      |       |
| 13.                                                                               |  | Positioning Control                                                                | 5-126                      |       |
| 14.                                                                               |  | Real Time Clock Control                                                            | 5-136                      |       |
| 15.                                                                               |  | Gray Codes                                                                         | 5-146                      |       |
| 16.                                                                               |  | Additional Functions                                                               | 5-146                      |       |
| 17.                                                                               |  | In-line Comparisons                                                                | 5-150                      |       |

## Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D1,S3or for lists/taled devices D3+0,S+9etc. MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆- An instruction operating in 16 bit mode, where identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆- An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

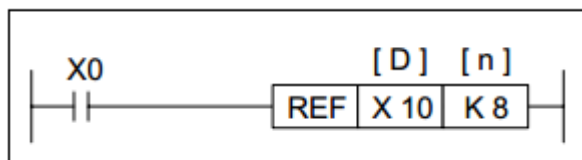
 - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

 - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

### 5.6.1 REF (FNC 50)

| Mnemonic                        | Function                                                     | Operands                                                                            |                                                                                   | Program steps         |
|---------------------------------|--------------------------------------------------------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------|
|                                 |                                                              | D                                                                                   | n                                                                                 |                       |
| REF<br>FNC 50<br>(Refresh)<br>➔ | Forces an immediate update of inputs or outputs as specified | X, Y<br>☒<br>Note:<br>D should always be a multiple of 10, i.e. 00, 10, 20, 30 etc. | K, H<br>☒<br>Note:<br>n should always be a multiple of 8, i.e. 8, 16, 24, 32 etc. | REF, REFP:<br>5 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

Standard PLC operation processes

output and input status between the END instruction of one program scan and step 0 of the following program scan. If an immediate update of the I/O device status is required the REF instruction is used. The REF instruction can only be used to update or refresh blocks of 8 (n) consecutive devices. The head address of the refreshed devices should always have its last digit as a 0 (zero), i.e. in units of 10.

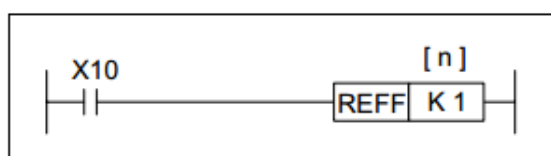


**Note:** A short delay will occur before the I/O device is physically updated, in the case of inputs a time equivalent to the filter setting, while outputs will delay for their set energized time.

## 5.6.2 REFF (FNC 51)

| Mnemonic                                      | Function                                                                              | Operands                                                                                                                                                     | Program steps              |
|-----------------------------------------------|---------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
|                                               |                                                                                       | n                                                                                                                                                            |                            |
| REFF<br>FNC 51<br>(Refresh and filter adjust) | Inputs are refreshed, and their input filters are reset to the newly designated value | K, H,<br>☒<br>Note: n= 0 to 60 msec (0 = 50μs)<br>X000 to X007 (X000 to X017 for FX <sub>2N</sub> ) are automatically designated when using this instruction | REFF,<br>REFFP:<br>3 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



### Operation:

PLC's are provided with input filters to overcome problems generated by

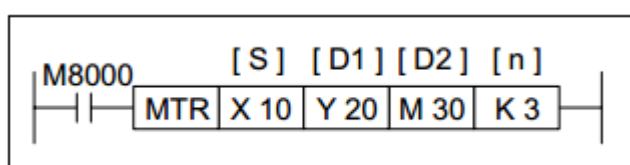
mechanical switch gear.

However, as this involves ensuring a steady input signal is received for a fixed time duration, the use of input filters slows down the PLC response times. For high speed applications, especially where solid state switching provides the input signal, input filter times may be reduced. The default setting for the input filters is approximately 10 msec. Using this instruction input filter times of 0 to 60 msec may be selected. The setting '0' (zero) is actually 50μsec. This is the minimum available setting. It is automatically selected when direct input, interrupts or high speed counting functions are used. The REFF instruction needs to be driven for each program scan if it is to be effective, otherwise, the standard 10 msec filter time is used.

### 5.6.3 MTR (FNC 52)

| Mnemonic                        | Function                                                                             | Operands                                                                            |        |              |                                 | Program steps |
|---------------------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------|--------------|---------------------------------|---------------|
|                                 |                                                                                      | S                                                                                   | D1     | D2           | n                               |               |
| MTR<br>FNC 52<br>(Input matrix) | Multiplexes a bank of inputs into a number of sets of devices. Can only be used ONCE | X<br>☒                                                                              | Y<br>☒ | Y, M, S<br>☒ | K, H,<br>☒<br>Note:<br>n=2 to 8 | MTR: 9 steps  |
|                                 |                                                                                      | Note:<br>These operands should always be a multiple of 10, i.e. 00, 10, 20, 30 etc. |        |              |                                 |               |

|                  |                  |         |       |                          |
|------------------|------------------|---------|-------|--------------------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Operation Complete M8029 |
|------------------|------------------|---------|-------|--------------------------|



#### Operation:

This instruction allows a selection of 8 consecutive input

devices (head address S) to be used multiple (n) times, i.e. each physical input has more than one, separate and quite different (D1) signal being processed. The result is stored in a matrix-table (head address D2).

#### Points to note:

- The MTR instruction involves high speed input/output switching. For this reason this instruction is only recommended for use with transistor output modules.
- For the MTR instruction to operate correctly, it must be driven continuously. It is recommended that special auxiliary relay M8000, the PLC RUN status flag, is used. After the completion of the first full reading of the matrix, operation complete flag M8029 is turned ON. This flag is automatically reset when the MTR instruction is turned OFF.

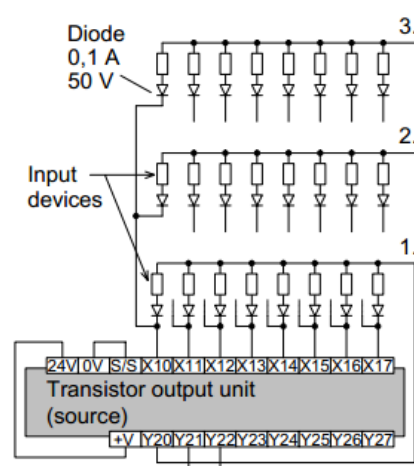
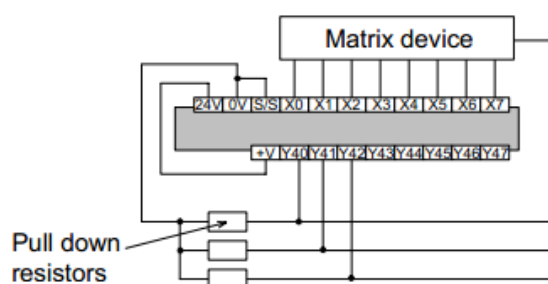
c) Each set of 8 input signals are grouped into a ‘bank’ (there are n number of banks).

d) Each bank is triggered/selected by a dedicated output (head address D1 ). This means the quantity of outputs from D1 , used to achieve the matrix are equal to the number of banks n. As there are now additional inputs entering the PLC these will each have a status which needs recording. This is stored in a matrix-table. The matrix-table starts at the head address D 2 . The matrix construction mimics the same 8 signal by n bank configuration.

Hence, when a certain input in a selected bank is read, its status is stored in an equivalent position within the result matrix-table.

e) The matrix instruction operates on an interrupt format, processing each bank of inputs every 20msec. This time is based on the selected input filters being set at 10msec. This would result in an 8 bank matrix, i.e. 64 inputs (8 inputs ´ 8 banks) being read in 160msec

**i** If high speed inputs (ex. X0) is specified for operand S, the reading time of each bank becomes only 10msec, i.e. a halving of the reading speed. However, additional pull down resistors are required on the drive outputs to ensure the high speed reading does not detect any residual currents from the last operation. These should be placed in parallel to the input bank and should be of a value of approximately .3k  $\Omega$ , 0.5W. For easier use, high speed inputs should not be specified at S.



f) Because this instruction uses a series of multiplexed signals it requires a certain amount of ‘ hard wiring ’ to operate. The example wiring diagram to the right

depicts the circuit used if the previous example instruction was programmed. As a general precaution to aid successful operation diodes should be placed after each input device (see diagram opposite). These should have a rating of 0.1A, 50V.

#### g) Example Operation

When output Y20 is ON only those inputs in the first bank are read. These results are then stored; in this example, auxiliary coils M30 to M37. The second step involves Y20 going OFF and Y21 coming ON. This time only inputs in the second bank are read. These results are stored in devices M40 to M47. The last step of this example has Y21 going OFF and Y22 coming ON. This then allows all of the inputs in the third bank to be read and stored in devices M50 to M57. The processing of this instruction example would take  $20 \times 3 = 60\text{msec}$ .



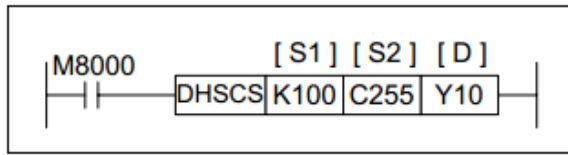
**Notice** how the resulting matrix-table does not use any of the ☆ 8 and ☆ 9 bit devices when state S or auxiliary M relays are used as the storage medium.

#### 5.6.4 HSCS (FNC 53)

| Mnemonic                                   | Function                                                                                   | Operands                                |                                                                |                                                            | Program steps      |
|--------------------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------|----------------------------------------------------------------|------------------------------------------------------------|--------------------|
|                                            |                                                                                            | S1                                      | S2                                                             | n                                                          |                    |
| HSCS<br>FNC 53<br>(High speed counter set) | Sets the selected output when the specified high speed counter value equals the test value | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z | C<br>Note:<br>C = 235 to 254, or available high speed counters | Y, M, S<br><br>Interrupt pointers I010 to I060 can be set. | DHSCS:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



#### Operation:

The HSCS set, compares the current value of the selected high speed counter (S 2 )against a selected value (S1 ). When the counters current value changes to a value equal to S1 the device specified as the destination (D)is set ON. The example above shows that Y10 would be set ON only when C255' s value stepped from 99-100 OR 101-100. If the counters current value was forced to equal 100, output Y10 would NOTbe set ON.

#### Points to note:

- a) It is recommended that the drive input used for the high speed counter functions; HSCS, HSCR, HSCZ is the special auxiliary RUN contact M8000.
- b) If more than one high speed counter function is used for a single counter the selected flag devices (D) should be kept within 1 group of 8 devices, i.e. Y0-7, M10-17.
- c) All high speed counter functions use an interrupt process, hence, all destination devices (D) are updated immediately.



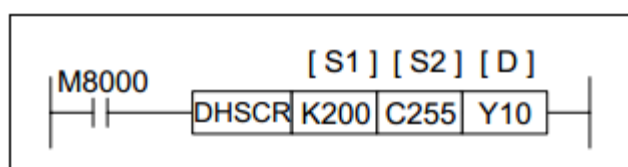
#### Note:

For all units Max. 6 simultaneously active HSCS/R and HSZ instructions. Please remember that the use of high speed counter functions has a direct impact on the maximum allowable counting speed!

### 5.6.5 HSCR (FNC 54)

| Mnemonic                                     | Function                                                                               | Operands                                |                                                                  |                                                       | Program steps      |
|----------------------------------------------|----------------------------------------------------------------------------------------|-----------------------------------------|------------------------------------------------------------------|-------------------------------------------------------|--------------------|
|                                              |                                                                                        | S1                                      | S2                                                               | D                                                     |                    |
| HSCR<br>FNC 54<br>(High speed counter reset) | Resets the selected output when the specified high speed counter equals the test value | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z | C<br>Note:<br>C = C235 to C255, or available high speed counters | Y, M, S<br>C<br>Note:<br>If C, use same counter as S2 | DHSCR:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



#### Operation:

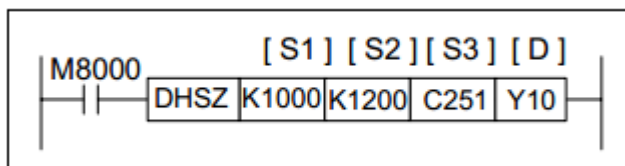
The HSCR, compares the current value of the selected

high speed counter (S2) against a selected value (S1). When the counter's current value changes to a value equal to S1, the device specified as the destination (D) is reset. In the example above, Y10 would be reset only when C255's value stepped from 199 to 200 or from 201 to 200. If the current value of C255 was forced to equal 200 by test techniques, output Y10 would NOT reset. For further, general points, about using high speed counter functions, please see the subsection 'Points to note' under the HSCS (FNC 53). Relevant points are; a, b, and c. Please also reference the note about the number of high speed instructions allowable.

## 5.6.6 HSZ (FNC 55)

| Mnemonic                                   | Function                                                                                                   | Operands                                |                                             |                               |                                                                           | Program steps     |
|--------------------------------------------|------------------------------------------------------------------------------------------------------------|-----------------------------------------|---------------------------------------------|-------------------------------|---------------------------------------------------------------------------|-------------------|
|                                            |                                                                                                            | S1                                      | S2                                          | S3                            | D                                                                         |                   |
| HSZ<br>FNC 55<br>(High speed zone compare) | Operation 1:<br>The current value of a high speed counter is checked against a specified range             | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z |                                             | C<br>Note:<br>C = 235 to 255, | Y, M, S<br>Note:<br>3 consecutive devices are used                        | DHSZ:<br>17 steps |
|                                            | Operation 2:<br>The designated range is held in a data table driving 'Y' outputs directly                  | D                                       | K,H<br>Using values from 1 to 128 (decimal) |                               | M8130 (only)<br>This flag can only be used with one DHSZ instr' at a time |                   |
|                                            | Operation 3:<br>The designated range is held in a data table driving PLSY frequencies directly using D8132 |                                         |                                             |                               | M8132 (only)<br>This flag can only be used with one                       |                   |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation 1 -

Standard:(Applicable to all

units) This instruction works in exactly the same way as the standard ZCP (FNC11). The only difference is that the device being compared is a high speed counter (specified as S3 ).Also, all of the outputs (D) are updated immediately due to the interrupt operation of the DHSZ. It should be remembered that when a device is specified in operand D it is in fact a head address for 3 consecutive devices. Each one is used to represent the status of the current comparison, i.e. using the above example as a basis,

Y10 (D) C251 is less than S1, K1000 (S3 <S1)

Y11 (D+1) C251 is greater than S1, K1000 but less than S 2, K1200 ( $S3 > S1$ ,  $S3 < S2$ )

Y12 (D+2) C251 is greater than S2, K1200 ( $S3 > S2$ )



For further, general points, about using high speed counter functions please see the subsection ‘Points to note’ under the HSCS (FNC 52). Relevant points are; a, b, and c. Please also reference the note about the number of high speed instructions allowable.

The following points should be read while studying the example on the right of the page. Please note, all normal rules associated with high speed counters still apply.

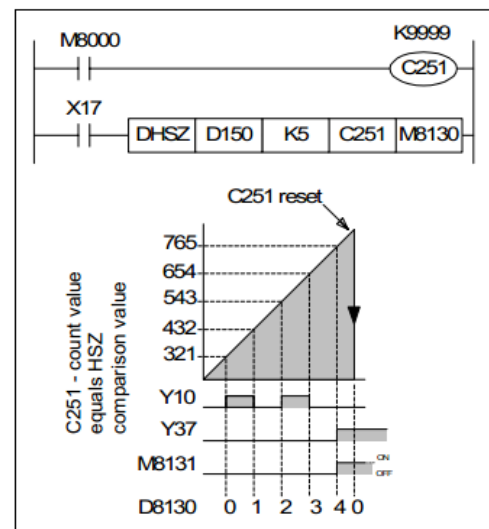
The data table is processed one ‘record number’ at a time, i.e only 1 record is ever active as the comparison data. The currently

| Record number<br>[D8130] | Comparison value<br>(lower/upper register)<br>[D, D+1] | Selected 'Y' Output Device<br>[D+2] | SET/RESET 'Y' Device<br>(1=SET, 0=RESET)<br>[D+3] |
|--------------------------|--------------------------------------------------------|-------------------------------------|---------------------------------------------------|
| 0                        | [D150, D151]<br>K321                                   | [D152]<br>H10 (Y10)                 | [D153]<br>K1                                      |
| 1                        | [D154, D155]<br>K432                                   | [D156]<br>H10 (Y10)                 | [D157]<br>K0                                      |
| 2                        | [D158, D159]<br>K543                                   | [D160]<br>H10 (Y10)                 | [D161]<br>K1                                      |
| 3                        | [D162, D163]<br>K765                                   | [D164]<br>H10 (Y10)                 | [D165]<br>K0                                      |
| 4                        | [D166, D167]<br>K765                                   | [D168]<br>H37 (Y37)                 | [D169]<br>K1                                      |

active record number is stored in data register D8130. As the comparison value for the

active record is ‘reached’, the assigned ‘Y’ device is SET or RESET and the active ‘Record number’ is incremented by 1. Once all records in a data table have been processed, the current record pointer (D8130) is reset to 0 (the table is then ready to process again) and the operation complete flag M8131 is set ON.

If the high speed counter is reset (by program or hardware input), when it resumes counting and reaches the first record’s comparison value, the M8131 flag will be reset. Both the status of M8131 and contents of D8130 are not editable by the user. If the DHSZ instruction is turned OFF then all associated flags are reset. Care should be exercised when resetting the high speed



counter or turning OFF the DHSZ instruction as all associated ‘Y’ output devices will remain in their last state, i.e. if an output was ON it will remain ON until independently reset by the user. The data within in active records can be changed during operation allowing data tables to be updated. Any change made is processed at the end of the current program scan. The HSZ instruction will continue to process only the active data record, i.e. it will not reset due to the updating of an inactive data record.

When the DHSZ instruction is initially activated it will not process a comparison until the

following program scan as the CPU requires a slight time delay to initialize the

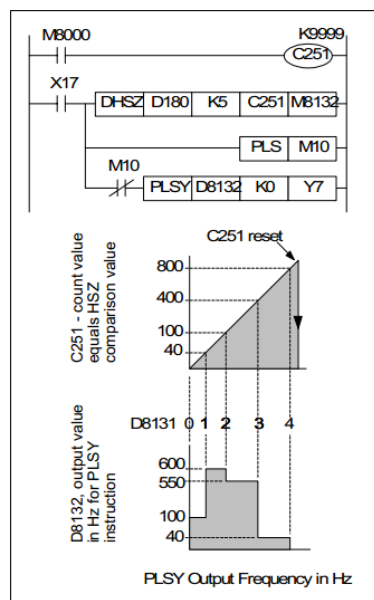
comparison

table.

As with Operation 2 only one record in the data table is active at anyone time. The current ‘Record number’ being processed is stored in data register D8131. To observe the current comparative value, data registers D8134 and D8135 should be monitored as a double word (32 bit) device.

Once the final entry in the data table has been processed, the operation complete flag M8133 is set ON and the

record counter (D8131) cycles back to the first record. It is recommended that if the high speed counter and PLSY operations form a closed loop that the last record entry in the data table is set to K0 for the comparison value and K0 for the PLSY output frequency. This will bring the controlled system to a stop and the ‘Record number’ counter will not be able to cycle back to the start of the data table until the associated high speed counter is reset by either pro-gram or hardware methods. This situation



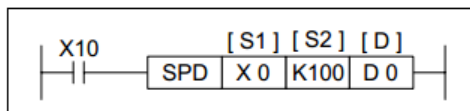
can be easily monitored by checking the paired data registers D8134 and D8135 for the '0' value.

It is recommended that the operation of the PLSY instruction is delayed for 1 scan to allow the DHSZ data table to be constructed on initial operation. A suggested program using a pulsed flag is shown in the example on this page.

### 5.6.7 SPD (FNC 56)

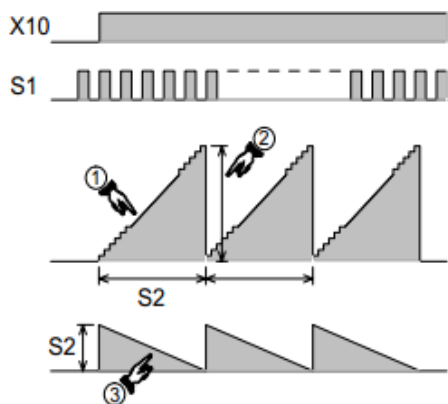
| Mnemonic                           | Function                                                                                             | Operands |                                         |                                                                                                               | Program steps   |
|------------------------------------|------------------------------------------------------------------------------------------------------|----------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------|-----------------|
|                                    |                                                                                                      | S1       | S2                                      | D                                                                                                             |                 |
| SPD<br>FNC 56<br>(Speed detection) | Detects the number of 'encoder' pulses in a given time frame. Results can be used to calculate speed | X0 to X5 | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z | T, C, D, Z (V)<br>Note:<br>3 consecutive devices are used. In the case of D= Z monitor D8028, D8029 and D8030 | SPD:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The number of pulses received at S1 are counted and stored in D+1; this is the current



count value. The counting takes place over a set time frame specified by S2 in msec. The time remaining on the current 'timed count', is displayed in device D +2. The number of counted pulses (of S1) from the last timed count are stored in D. The timing chart opposite shows the SPD operation in a graphical sense.

Note: ①: Current count value, device D +1

② : Accumulated/ last count value, device D

③: Current time remaining in msec, device D+2

**Points to note:**

- a) When the timed count frame is completed the data stored in D +1 is immediately written to D. D +1 is then reset and a new time frame is started.
- b) Because this is both a high speed and an interrupt process only inputs X0 to X5 may be used as the source device S 1 . However, the specified device for S1 must NOT coincide with any other high speed function which is operating, i.e. a high speed counter using the same input. The SPD instruction is considered to act as a single phase counter.
- c) Multiple SPD instructions may be used, but the identified source devices S1 restrict this to a maximum of 6 times.
- d) Once values for timed counts have been collected, appropriate speeds can be calculated using simple mathematics. These speeds could be radial speeds in rpm, linear speeds in M/ min it is entirely down to the mathematical manipulation placed on the SPD results. The following interpretations could be used;

$$\text{Linear speed } N \text{ (km/h)} = \frac{3600 \times (D)}{n \times S_2} \times 10^3$$

where n = the number of linear encoder divisions per kilometer.

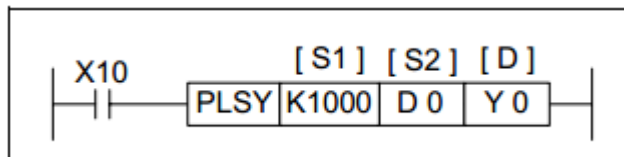
$$\text{Radial speed } N \text{ (rpm)} = \frac{60 \times (D)}{n \times S_2} \times 10^3$$

where n = the number of encoder pulses per revolution of the encoder disk.

## 5.6.8 PLSY (FNC 57)

| Mnemonic                           | Function                                                | Operands                                |    |                                    | Program steps                         |
|------------------------------------|---------------------------------------------------------|-----------------------------------------|----|------------------------------------|---------------------------------------|
|                                    |                                                         | S1                                      | S2 | D                                  |                                       |
| PLSY<br>FNC 57<br>(Pulse Y output) | Outputs a specified number of pulses at a set frequency | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z |    | Y<br>Note:<br>Y000 or Y001 only ☒. | PLSY:<br>7 steps<br>DPLSY:<br>13steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Operation Complete M8029 |
|------------------|------------------|---------|-------|--------------------------|
|------------------|------------------|---------|-------|--------------------------|



Operation:

A specified quantity of pulses S2

is output through device D at a

specified frequency S1. This instruction is used in situations where the quantity of outputs is of primary concern.

Points to note:

a) HCA1P/ HCA2P/HCA2C users may use frequencies of 1 to 132,767Hz (16-bit operation) and 1 to 100kHz (32-bit operation).

b) The maximum number of pulses: 16 bit operation: 1 to 32,767 pulses, 32 bit operation: 1 to 2,147,483,647 pulses.

Note: special auxiliary coil M8029 is turned ON when the specified number of pulses

has been completed. The pulse count and completion flag (M8029) are reset when the PLSY instruction is de-energized. If “0” (zero) is specified the PLSY instruction will continue generating pulses for as long as the instruction is energized.

c) A single pulse is described as having a 50% duty cycle. This means it is ON for 50% of the pulse and consequently OFF for the remaining 50% of the pulse. The actual output is controlled by interrupt handling, i.e. the output cycle is NOT affected by the scan time of the program.

d) The data in operands S 1 and S 2 may be changed during execution. However, the new data in S 2 will not become effective until the current operation has been completed, i.e. the instruction has been reset by removal of the drive contact.

e) Two FNC 57 (PLSY) can be used at the same time in a program to output pulses to Y000 and Y001 respectively. Or, only one FNC 57 PLSY and one FNC 59 PLSR can be used together in the active program at once, again outputting independent pulses to Y000 and Y001

.



It is possible to use subroutines or other such programming techniques to isolate different instances of this instructions. In this case, the current instruction must be deactivated before changing to the new instance.

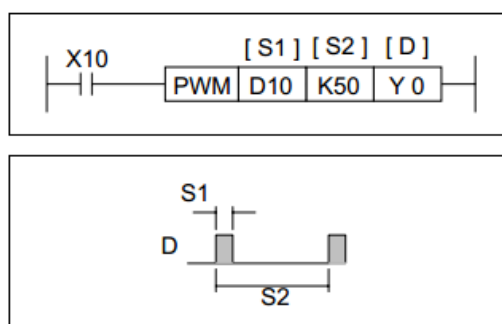
f) Because of the nature of the high speed output, transistor output units should be used with this instruction. Relay outputs will suffer from a greatly reduced life and will cause false outputs to occur due to the mechanical ‘bounce’ of the contacts. The

load current should be 10 - 100mA with the HCA1P/HCA2P/HCA2C Series. It may be found that ‘pull up’ resistors will be required.

### 5.6.9 PWM (FNC 58)

| Mnemonic                                  | Function                                                   | Operands                                                     |    |                                              | Program steps   |
|-------------------------------------------|------------------------------------------------------------|--------------------------------------------------------------|----|----------------------------------------------|-----------------|
|                                           |                                                            | S1                                                           | S2 | D                                            |                 |
| PWM<br>FNC 58<br>(Pulse width modulation) | Generates a pulse train with defined pulse characteristics | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z<br>Note:<br>S1<br>S2 |    | Y<br>Note:<br>All units: Y000 or Y001 only ☒ | PWM:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



#### Operation:

A continuous pulse train is output through device D when this instruction is driven.

The characteristics of the pulse are defined as: The distance, in time (msec), between

two identical parts of consecutive pulses (S 2 ). And how long, also in time (msec), a single pulse will be active for (S1 )

#### Points to note:

- Because this is a 16 bit instruction, the available time ranges for S 1 and S 2 are 1 to 32,767.
- A calculation of the duty cycle is easily made by dividing S1 by S2 . Hence S 1 cannot

have a value greater than S 2 as this would mean the pulse is on for longer than the distance between two pulses, i.e. a second pulse would start before the first had finished. If this is programmed an error will occur. This instruction is used where the length of the pulse is the primary concern.

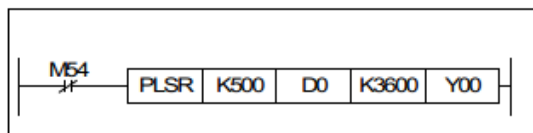
c) The PWM instruction may only be used once in a users program.

d) Because of the nature of the high speed output, transistor output units should be used with this instruction. Relay outputs will suffer from a greatly reduced life and will cause false outputs to occur due to the mechanical ‘bounce’ of the contacts. The load current should be 10 - 100mA with the HCA1P/ HCA2P/HCA2C Series. It may be found that ‘pull up’ resistors will be required.

#### 5.6.10 PLSR (FNC 59)

|                  |                  |         |       |                          |
|------------------|------------------|---------|-------|--------------------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Operation Complete M8029 |
|------------------|------------------|---------|-------|--------------------------|

| Mnemonic                       | Function                                                                                  | Operands                                                     |    |    |                                           | Program steps                          |
|--------------------------------|-------------------------------------------------------------------------------------------|--------------------------------------------------------------|----|----|-------------------------------------------|----------------------------------------|
|                                |                                                                                           | S1                                                           | S2 | S3 | D                                         |                                        |
| PLSR<br>FNC 59<br>(Pulse ramp) | Outputs a specified number of pulses, ramping up to a set frequency and back down to stop | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z<br>Note:<br>S1<br>S2 |    |    | Y<br><br>HCA5users:<br>Y000 or Y001 only. | PLSR:<br>9 steps<br>DPLSR:<br>17 steps |



#### Operation:

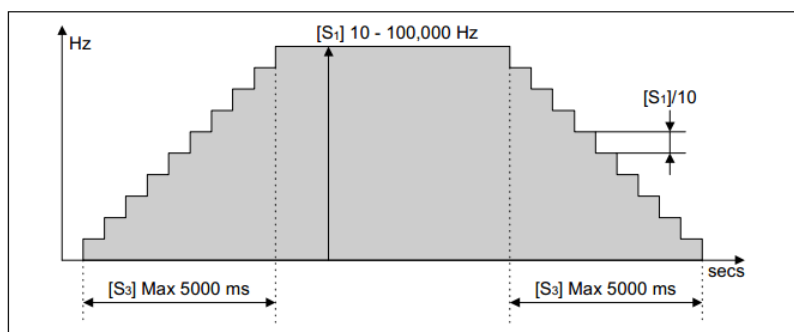
A specified quantity of pulses S 2 is

output through device D. The output

frequency is first ramped up in 10 steps to the maximum frequency S1 in acceleration

time S 3 ms, then ramped down to stop also in S3 ms. This instruction is used to

generate simple acc/dec curves where the quantity of outputs is of primary concern.



#### Points to Note:

- a) HCA1P/ HCA2P/HCA2C users may use frequencies of 10 to 100,000Hz. The frequency should be set to a multiple of 10. If not it will be rounded up to the next multiple of 10. The acceleration and deceleration steps are set to 1/10 of the maximum frequency. Take this in to consideration to prevent slipping, when using stepping motors.
- b) All HCA1P/ HCA2P /HCA2C units, maximum number of pulses: 16 bit operation: 110 to 32,767 pulses, 32 bit operation: 110 to 2,147,483,647 pulses. Correct pulse output can not be guaranteed for a setting of 110 or less.
- c) The acceleration time must conform to the limitations described below.
- d) The output device is limited to Y000 or Y001 only and should be transistor type.
- e) Two FNC 59 (PLSR) can be used at the same time in a program to output pulses to Y000 and Y001 respectively. Or, only one FNC 57 PLSY and one FNC 59 PLSR can be used together in the active program at once, again outputting independent pulses to Y000 and Y001.

⚠ It is possible to use subroutines or other such programming techniques to isolate different instances of this instructions. In this case, the current instruction must be deactivated before changing to the new instance.

f) If the number of pulses is not enough to reach the maximum frequency then the frequency is automatically cut

g) Special auxiliary coil M8029 turns ON when the specified number of pulses has been completed. The pulse count and completion flag (M8029) are reset when the PLSR instruction is de-energized.

### Acceleration time limitations

The acceleration time S3 has a maximum limit of 5000 ms. However, the actual limits of S3 are determined by other parameters of the system according to the following 4 points.

1) Set S3 to be more than 10 times the maximum program scan time (D8012). If set to less than this, then the timing of the acceleration steps becomes uneven.

$$S3 \geq \frac{90000}{S1} \times 5$$

2) The following formula gives the minimum value for S3 .

3) The following formula gives the maximum value for S3.

$$S3 \leq \frac{S2}{S1} \times 818$$

4) The pulse output always increments in 10 step up to the maximum frequency as shown on the previous page.

If the parameters do not meet the above conditions, reduce the size of S1 .



- Possible output frequency is limited to 10 to 100,000Hz for HCA1P/HCA2P/HCA2C. If either the maximum frequency or the acceleration step size are outside this limit then they are automatically adjusted to bring the value back to the limit.

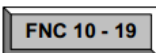
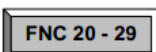
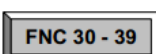
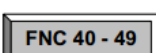
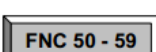
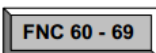
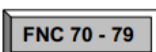
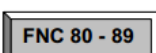
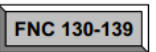
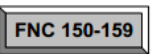
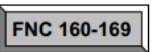
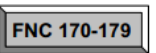
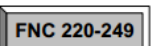
- If the drive signal is switch off, all output stops. When driven ON again, the process starts from the beginning.
- Even if the operands are changed during operation, the output profile does not change. The new values take effect from the next operation.



— 禾川股份 — HCFA CORPORATION

---

## Applied Instructions:

|                                                                                    |                                                                                   |                                                                                     |                                 |       |
|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------|-------|
| 1.                                                                                 |  | Program Flow                                                                        | 5-4                             |       |
| 2.                                                                                 |  | Move And Compare                                                                    | 5-16                            |       |
| 3.                                                                                 |  | Arithmetic And Logical Operations (+, -, ×, ÷)                                      | 5-24                            |       |
| 4.                                                                                 |  | Rotation And Shift                                                                  | 5-34                            |       |
| 5.                                                                                 |  | Data Operation                                                                      | 5-42                            |       |
| 6.                                                                                 |  | High Speed Processing                                                               | 5-52                            |       |
|  | 7.                                                                                |   | Handy Instructions              | 5-66  |
|                                                                                    | 8.                                                                                |  | External FX I/O Devices         | 5-80  |
|                                                                                    | 9.                                                                                |  | External FX Serial Devices      | 5-94  |
|                                                                                    | 10.                                                                               |  | Floating Point 1 & 2            | 5-110 |
|                                                                                    | 11.                                                                               |  | Trigonometry (Floating Point 3) | 5-118 |
|                                                                                    | 12.                                                                               |  | Data Operations 2               | 5-122 |
|                                                                                    | 13.                                                                               |  | Positioning Control             | 5-126 |
|                                                                                    | 14.                                                                               |  | Real Time Clock Control         | 5-136 |
|                                                                                    | 15.                                                                               |  | Gray Codes                      | 5-146 |
|                                                                                    | 16.                                                                               |  | Additional Functions            | 5-146 |
|                                                                                    | 17.                                                                               |  | In-line Comparisons             | 5-150 |



### Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D 1 ,S3 or for lists/tabled devices D3+0, S +9etc.

MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D ☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation



A repetitive instruction which will change the destination value on every scan

unless modified by the pulse function.

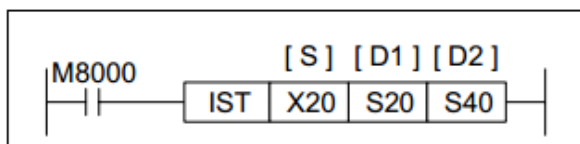
✖ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will

have no effect to the value of the operand.

### 5.7.1 IST (FNC 60)

| Mnemonic                         | Function                                                | Operands                                           |    |    | Program steps   |
|----------------------------------|---------------------------------------------------------|----------------------------------------------------|----|----|-----------------|
|                                  |                                                         | S1                                                 | S2 | S3 |                 |
| IST<br>FNC 60<br>(Initial state) | Automatically sets up a multi-mode STL operating system | X, Y, M, S,<br>Note:<br>uses 8 consecutive devices | S, |    | IST:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction automatically sets up a multi-mode STL operating system.

This consists of variations of ‘manual’ and ‘automatic’ operation modes.

Points to note:

- The IST instruction automatically assigns and uses many bit flags and word devices; these are listed in the boxed column on the right of this page.
- The IST instruction may only be used ONCE . It should be programmed close to the beginning of the program, before the controlled STL circuits.
- The required operation mode is selected by driving the devices associated with

operands S+0 through to S+4 (5 inputs). None of the devices within this range should be ON at the same time. It is recommended that these ‘ inputs ’ are selected through use of a rotary switch. If the currently selected operating mode is changed before the ‘ zero return complete ’ flag (M8043) is set, all outputs will be turned OFF.

d) The ‘ zero position ’ is a term used to identify a datum position from where the controlled device, starts from and returns too after it has completed its task. Hence, the operating mode ‘ zero return ’ , causes the controlled system to return to this datum.

#### **Assigned devices**

Indirect user selected devices:

S +0Manual operation

S +1Zero return

S +2Step operation

S +3One cycle operation

S +4Cyclic operation

S +5Zero return start

S +6Automatic operation start

S +7Stop

#### **Initial states:**

S0 initiates ‘ manual ’ operation

S1 initiates ‘ zero return ’ operation

S2 initiates ‘ automatic ’ operation

**General states:**

S10toS19 ‘zero return’ sequence

D 1 to D 2 ‘automatic return’ sequence

**Special bit flags:**

M8040 = ON STL state transfer is inhibited

M8041 = ON initial states are enabled

M8042 = Start pulse given by start input

M8043 = ON zero return completed

M8044 = ON machine zero detected

M8047 = ON STL monitor enabled



The ‘zero’ position is sometimes also referred to as a home position, safe position, neutral position or a datum position.

e) The available operating modes are split into two main groups, manual and automatic. There are sub-modes to these groups. Their operation is defined as:

**Manual**

Manual (selected by device S+0)- Power supply to individual loads is turned ON and OFF by using a separately provided means, often additional push buttons. Zero Return (selected by device S+1) -Actuators are returned to their initial positions when the Zero input (S+5) is given.

**Automatic**

One Step (selected by device S+2)- The controlled sequence operates automatically but will only proceed to each new step when the start input (S+6) is given.

One Cycle (selected by device S+3) - The controlled actuators are operated for one operation cycle. After the cycle has been completed, the actuators stop at their ‘zero’ positions. The cycle is started after a ‘start’ input (S+6) has been given.

A cycle which is currently being processed can be stopped at any time by activating the ‘stop’ input (S+7). To restart the sequence from the currently ‘paused’ position the start input must be given once more.

Automatic (selected by device S+4)-Fully automatic operation is possible in this mode. The programmed cycle is executed repeatedly when the ‘start’ input (S+6) is given. The currently operating cycle will not stop immediately when the ‘stop’ input (S+7) is given. The current operation will proceed to then end of the current cycle and then stop its operation.



**Note:** Start, stop and zero inputs are often given by additional, manually operated push buttons.

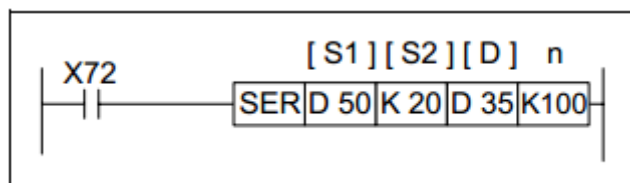
Please note that the ‘stop’ input is only a program stop signal. It cannot be used as a replacement for an ‘Emergency stop’ push button. All safety, ‘Emergency stop’ devices should be hardwired systems which will effectively isolate the machine from operation and external power supplies. Please refer to local and national standards for

applicable safety practices.

### 5.7.2 SER (FNC 61)

| Mnemonic                               | Function                                                                               | Operands                    |                                             |                                                                  |                                                                                         | Program steps                                   |
|----------------------------------------|----------------------------------------------------------------------------------------|-----------------------------|---------------------------------------------|------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------|
|                                        |                                                                                        | S1                          | S2                                          | D                                                                | n                                                                                       |                                                 |
| SER<br>FNC 61<br>(Search a Data Stack) | Generates a list of statistics about a single data value located/found in a data stack | KnX, KnY, KnM, KnS, T, C, D | KnX, KnY, KnM, KnS, T, C, D<br>V, Z<br>K, H | KnY, KnM, KnS<br>T, C, D<br>Note: 5 consecutive devices are used | K, H, D<br>☒<br>Note:<br>n= 1~256 for 16 bit operation<br>n= 1~128 for 32 bit operation | SER, SERP: 9 steps<br><br>DSER, DSERP: 17 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The SER instruction searches a defined data stack from head

address S1, with a stack length n. The data searched for is specified in parameter S2 and the results of the search are stored at destination device D for 5 consecutive devices.

| Destination device | Device description                                                                                                                                                            |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| D                  | Total number of occurrences of the searched value S2 (0 if no occurrences are found)                                                                                          |
| D+1                | The position (within the searched data stack) of the first occurrence of the searched value S2                                                                                |
| D+2                | The position (within the searched data stack) of the last occurrence of the searched value S2                                                                                 |
| D+3                | The position (within the searched data stack) of the smallest value found in the data stack (last occurrence is returned if there are multiple occurrences of the same value) |
| D+4                | The position (within the searched data stack) of the largest value found in the data stack (last occurrence is returned if there are multiple occurrences of the same value)  |

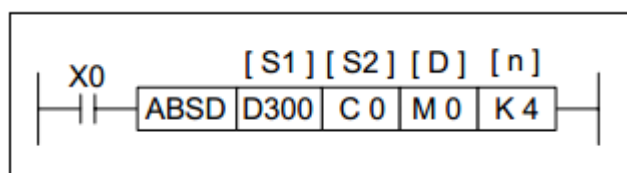
### Points to note:

- a) Normal rules of algebra are used to determine the largest and smallest values, i.e. -30 is smaller than 6 etc.
- b) If no occurrence of the searched data can be found then destination devices D, D+1 and D+2 will equal 0 (zero).
- c) When using data register s as the destination device D please remember that 16 bit operation will occupy 5 consecutive, data registers but 32 bit operation will occupy 10 data registers in pairs forming 5 double words.
- d) When multiple bit devices are used to store the result (regardless of 16 or 32 bit operation), only the specified size of group is written to for 5 consecutive occurrences, i.e. K1Y0 would occupy 20 bit devices from Y0 (K1 = 4 bit devices and there will be 5 groups for the 5 results). As the maximum data stack is 256 (0 to 255) entries long, the optimum group of bit devices required is K2, i.e. 8 bit devices.

### 5.7.3 ABSD (FNC 62)

| Mnemonic                                    | Function                                                       | Operands                                                                                         |    |                                                   |                           | Program steps                                 |
|---------------------------------------------|----------------------------------------------------------------|--------------------------------------------------------------------------------------------------|----|---------------------------------------------------|---------------------------|-----------------------------------------------|
|                                             |                                                                | S1                                                                                               | S2 | D                                                 | n                         |                                               |
| ABSD<br>FNC 62<br>(Absolute drum sequencer) | Generates multiple output patterns in response to counter data | KnX, KnY, KnM, KnS, (in groups of 8)<br>T, C, D<br><br>Note: High speed counters are not allowed | C  | Y,M,S<br><br>Note: n consecutive devices are used | K,H<br><br>☒ Note: n ≤ 64 | ABSD: 9 steps<br><br>DABSD: 17 steps. see f). |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

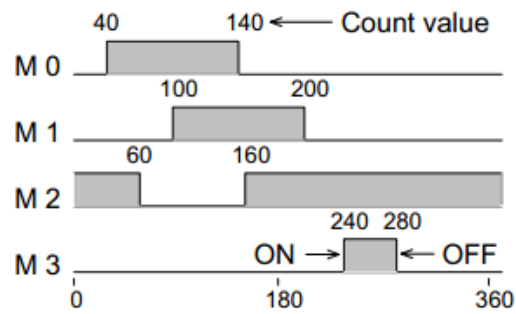
This instruction generates a variety of output patterns (there are n number of addressed outputs) in response to the current value of a selected counter, S2.

**Points to note:**

- a) The current value of the selected counter (S2) is compared against a user defined data table. This data table has a head address identified by operand S1. S1 should always have an even device number.
- b) For each destination bit (D) there are two consecutive values stored in the data table. The first allocated value represents the event number when the destination device (D) will be turned ON. The second identifies the reset event. The data table values are allocated as a consecutive pair for each sequential element between D and D+n.
- c) The data table has a length equal to  $2 \times n$  data entries. Depending on the format of the data table, a single entry can be one data word such as D300 or a group of 16 bit devices e.g. K4X000.
- d) Values from 0 to 32,767 may be used in the data table.
- e) The ABSD instruction may only be used **ONCE**.

From the example instruction and the data table below, the following timing diagram for elements M0 to M3 can be constructed.

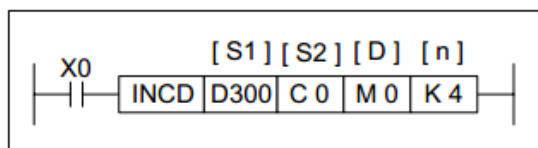
| When counter S2 equals the value below, the destination device D is |            | Assigned destination device D |
|---------------------------------------------------------------------|------------|-------------------------------|
| turned ON                                                           | turned OFF |                               |
| D300 - 40                                                           | D301 - 140 | M0                            |
| D302 - 100                                                          | D303 - 200 | M1                            |
| D304 - 160                                                          | D305 - 60  | M2                            |
| D306 - 240                                                          | D307 - 280 | M3                            |



## 5.7.4 INCD (FNC 63)

| Mnemonic                                       | Function                                                       | Operands                                        |                                  |                                                     |                            | Program steps    |
|------------------------------------------------|----------------------------------------------------------------|-------------------------------------------------|----------------------------------|-----------------------------------------------------|----------------------------|------------------|
|                                                |                                                                | S1                                              | S2                               | D                                                   | n                          |                  |
| INCD<br>FNC 63<br>(Incremental drum sequencer) | Generates a single output sequence in response to counter data | KnX, KnY, KnM, KnS, (in groups of 8)<br>T, C, D | C<br>Uses 2 consecutive counters | Y, M, S<br><br>Note: n consecutive devices are used | K, H<br><br>☒ Note: n ≤ 64 | INCD:<br>9 steps |

|                  |                  |         |       |                          |
|------------------|------------------|---------|-------|--------------------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Operation Complete M8029 |
|------------------|------------------|---------|-------|--------------------------|



Operation:

This instruction generates a sequence of sequential output patterns (there are n number of addressed outputs) in response to the current value of a pair of selected counters (S2, S2+1).

Points to note:

a) This instruction uses a 'data table' which contains a single list of values which are to be selected and compared by two consecutive counters (S2 and S2+1). The data

table is identified as having a head address S1 and consists of n data elements.

b) Counter S2 is programmed in a conventional way. The set value for counter S2 MUST be greater than any of the values entered into the data table. Counter S2 counts a user event and compares this to the value of the currently selected data element from the data table.

When the counter and data value are equal, S2 increments the count of counter S2+1 and resets its own current value to '0' (zero). This new value of counter S2+1 selects the new data element from the data table and counter S2 now compares against the new data elements value.

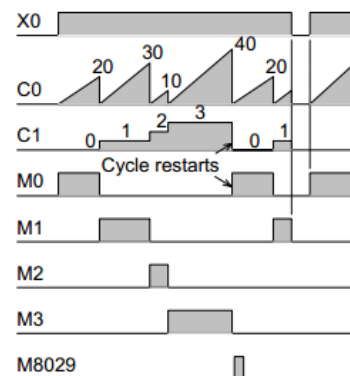
c) The counter S2+1 may have values from 0 to n. Once the nth data element has been processed, the operation complete flag M8029 is turned ON. This then automatically resets counter S2+1 hence, the cycle starts again with data element S1+0.

d) Values from 0 to 32,767 may be used in the data table.

e) The INCD instruction may only be used ONCE in a program.

From the example instruction and the data table identified left, the following timing diagram for elements M0 to M3 can be constructed.

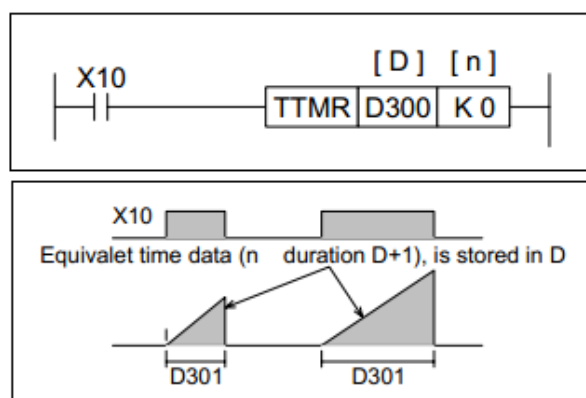
| Data table   |                                         | Value of counter S2+1 |
|--------------|-----------------------------------------|-----------------------|
| Data element | Data value / count value for counter S2 |                       |
| D300         | 20                                      | 0                     |
| D301         | 30                                      | 1                     |
| D302         | 10                                      | 2                     |
| D303         | 40                                      | 3                     |



### 5.7.5 TTMR (FNC 64)

| Mnemonic                           | Function                                                                         | Operands                                                    |                                                                                                  | Program steps    |
|------------------------------------|----------------------------------------------------------------------------------|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------|------------------|
|                                    |                                                                                  | D                                                           | n                                                                                                |                  |
| TTMR<br>FNC 64<br>(Teaching timer) | Monitors the duration of a signal and places the timed data into a data register | D<br><br>Note:<br>2 devices 16 bit words are used D and D+1 | K, H<br>☒<br>Note:<br>n= 0: (D) = (D+1) × 1<br>n= 1: (D) = (D+1) × 10<br>n= 2: (D) = (D+1) × 100 | TTMR:<br>5 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



#### Operation:

The duration of time that the TTMR instruction is energized, is measured and stored in device D +1 (as a count of 100ms periods).

The data value of D+1(in secs),

multiplied by the factor selected by the operand n, is moved in to register D. The contents of D could be used as the source data for an indirect timer setting or even as raw data for manipulation.

When the TTMR instruction is de-energized D+1 is automatically reset (D is unchanged)

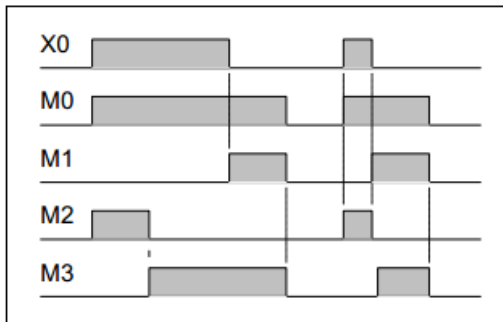
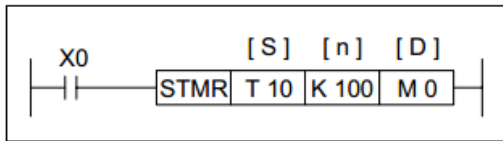
### 5.7.6 STMR (FNC 65)

| Mnemonic                          | Function                                                | Operands                                        |                                      |                                                      | Program steps    |
|-----------------------------------|---------------------------------------------------------|-------------------------------------------------|--------------------------------------|------------------------------------------------------|------------------|
|                                   |                                                         | S                                               | n                                    | D                                                    |                  |
| STMR<br>FNC 65<br>(Special timer) | Provides dedicated off-delay, one shot and flash timers | T<br>Note:<br>Timers 0 to 199 (100msec devices) | K, H<br>☒<br>Note:<br>n= 1 to 32,767 | Y, M, S<br>Note:uses 4 consecutive devices D+0to D+3 | STMR:<br>7 steps |

## 16 BIT OPERATION

## 32 BIT OPERATION

## PULSE-P



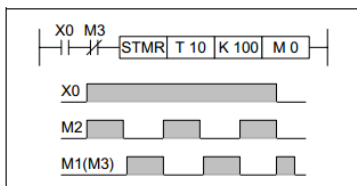
### Operation:

The designated timer S will operate for the duration n with the operational effect

being flagged by devices D+0 to D+3.

Device D+0 is an off-delay timer, D+1 is a one shot timer. When D+3 is used in the

configuration below, D+1 and D+2 act in an alternate flashing sequence.



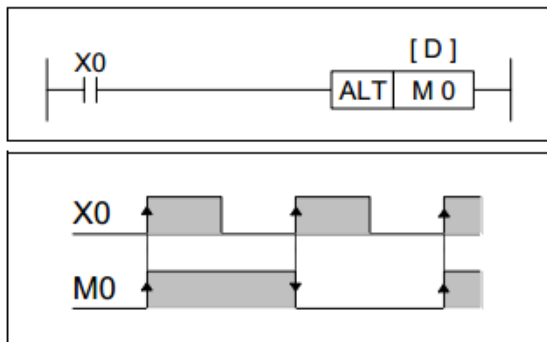
### 5.7.7 ALT (FNC 66)

| Mnemonic                                | Function                                                                            | Operands | Program steps         |
|-----------------------------------------|-------------------------------------------------------------------------------------|----------|-----------------------|
|                                         |                                                                                     | D        |                       |
| ALT<br>FNC 66<br>(Alternate state)<br>→ | The status of the assigned device is inverted on every operation of the instruction | Y, M, S  | ALT, ALTP:<br>3 steps |

## 16 BIT OPERATION

## 32 BIT OPERATION

## PULSE-P



## Operation:

The status of the destination device (D) is alternated on every operation of the ALT instruction.

This means the status of each bit device

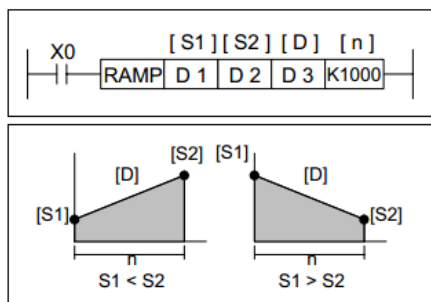
will flip-flop between ON and OFF. This will occur on every program scan unless a pulse modifier or a program interlock is used.

The ALT instruction is ideal for switching between two modes of operation e.g. start and stop, on and off etc.

## 5.7.8 RAMP (FNC 67)

| Mnemonic                                | Function                                                                  | Operands                                                                                                   |    |   |                                      | Program steps    |
|-----------------------------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|----|---|--------------------------------------|------------------|
|                                         |                                                                           | S1                                                                                                         | S2 | D | n                                    |                  |
| RAMP<br>FNC 67<br>(Ramp variable value) | Ramps a device from one value to another in the specified number of steps | D<br>Note:<br>Device D uses two consecutive registers identified as D and D+1 these are read only devices. |    |   | K, H<br>☒<br>Note:<br>n= 1 to 32,767 | RAMP:<br>9 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Operation Complete M8029 |
|------------------|------------------|---------|-------|--------------------------|
|------------------|------------------|---------|-------|--------------------------|



## Operation:

The RAMP instruction varies a current value (D) between the data limits set by the user (S1 and S2). The ‘journey’ between these extreme limits takes n program scans. The current scan

number is stored in device D+1. Once the current value of D equals the set value of

S2the execution complete flag

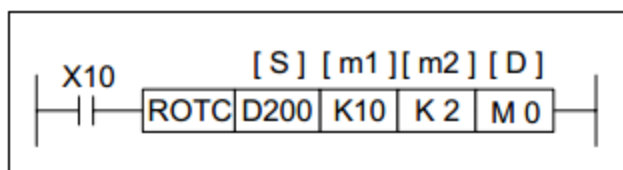
M8029 is set ON. The RAMP instruction can vary both increasing and decreasing

differences between S1and S2.

### 5.7.9 ROTC (FNC 68)

| Mnemonic                                 | Function                                                                           | Operands                                                  |                                         |                                         |                                                | Program steps    |
|------------------------------------------|------------------------------------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------|-----------------------------------------|------------------------------------------------|------------------|
|                                          |                                                                                    | S                                                         | m1                                      | m2                                      | D                                              |                  |
| ROTC<br>FNC 68<br>(Rotary table control) | Controls a rotary tables movement is response to a requested destination/ position | D<br>Note:<br>uses 3 consecutive devices<br>$S+1 \leq m1$ | K, H<br>☒<br>Note:<br>$m1= 2$ to 32,767 | K, H<br>☒<br>Note:<br>$m2= 0$ to 32,767 | Y, M, S<br>Note:<br>uses 8 consecutive devices | ROTC:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



**Operation:**

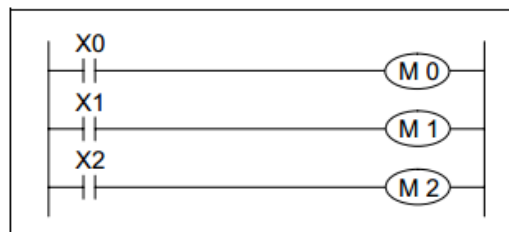
The ROTC instruction is used to aid the tracking and positional

movement of the rotary table as it moves to a specified destination.

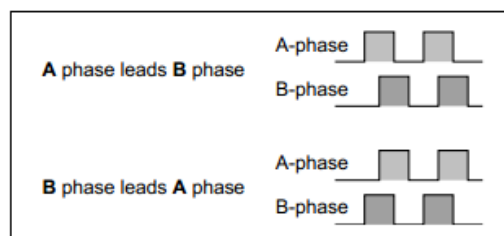
**Points to note:**

a) This instruction has many automatically de-fined devices. These are listed below.

- b) The ROTC instruction may only be used **ONCE** .
- c) The ROTC instruction uses a built in 2-phase counter to detect both movement direction and distance travelled. Devices D+0 and D+1 are used to input the phase pulses, while device D+2 is used to input the ‘zero position’ on the rotary table. These devices should be programmed as shown in the example below (where the physical termination takes place at the associated X inputs).



The movement direction is found by checking the relationship of the two phases of the 2 phase counter, e.g.



## Assigned devices

### Indirect user selected devices:

D +0 A-phase counter signal - input

D +1 B-phase counter signal - input

D +2 Zero point detection - input

D +3 High speed forward - output

D +4 Low speed forward - output

D +5 Stop - output

D +6 Low speed reverse - output

D +7 High speed reverse – output

### **Rotary table constants:**

m 1 Number of encoder pulses per table revolution

m 2 Distance to be travelled at low speed (in encoder pulses)

### **Operation variables:**

S +0 Current position at the ‘zero point’ READ ONLY

S +1 Destination position (selected station to be moved to) relative to the ‘zero point’ - User defined

S +2 Start position (selected station to be moved) relative to the ‘zero point’ -User defined

d) When the ‘zero point’ input (D+2) is received the contents of device S+0 is reset to ‘0’

(zero). Before starting any new operation it is advisable to ensure the rotary table is initialized by moving the ‘zero point’ drive dog or marker around to the ‘zero point’ sensor.

This could be considered as a calibration technique. The re-calibration of the rotary

table should be carried out periodically to ensure a consistent/accurate operation.

e) Devices D+3 to D+7 are automatically set by the ROTC instruction during its operation.

These are used as flags to indicate the operation which should be carried out next.

f) All positions are entered in the form of the required encoder pulses. This can be seen in the following example:

**- Example:**

A rotary table has an encoder which outputs 400 (m1) pulses per revolution. There are 8 stations (0 to 7) on the rotary table. This means that when the rotary table moves from one station to its immediately following station, 50 encoder pulses are counted. The ‘zero position’ is station ‘0’ (zero). To move the item located at station 7 to station 3 the following values must be written to the ROTC instruction:

$S+1=3 \times 50 = 150$  (station 3’ s position in encoder pulses from the zero point)

$S+2=7 \times 50 = 350$  (station 7’ s position in encoder pulses from the zero point)

$m1= 400$  (total number of encoder pulses per rev)

The rotary table is required approach the destination station at a slow speed starting from

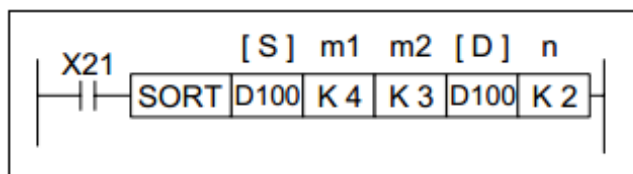
1.5 stations before the destination. Therefore;

$m2= 1.5 \times 50 = 75$  slow speed distance either side of the destination station (in encoder pulses)

## 5.7.10 SORT (FNC 69)

| Mnemonic                                      | Function                                                                                  | Operands |                                                 |    |        |                                        | Program steps     |
|-----------------------------------------------|-------------------------------------------------------------------------------------------|----------|-------------------------------------------------|----|--------|----------------------------------------|-------------------|
|                                               |                                                                                           | S        | m1                                              | m2 | D      | n                                      |                   |
| SORT<br>FNC 69<br>(SORT<br>Tabulated<br>Data) | Data in a defined table can be sorted on selected fields while retaining record integrity | D<br>☒   | K, H<br>☒<br>Note:<br>m1= 1 to 32<br>m2= 1 to 6 |    | D<br>☒ | K, H<br>D<br>☒<br>Note:<br>n = 1 to m2 | SORT:<br>11 steps |

|                  |                  |         |       |                          |
|------------------|------------------|---------|-------|--------------------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Operation Complete M8029 |
|------------------|------------------|---------|-------|--------------------------|



Operation:

This instruction constructs a

data table of m1 records with m

2 fields having a start or head address of S. Then the data in field n is sorted in to

numerical order while retaining each individual records integrity. The resulting (new)

data table is stored from destination device D.

### Points to note:

a) When a sort occurs each record is sorted in to ascending order based on the data in the selected sort field n.

b) The source (S) and destination (D) areas can be the same BUT if the areas are chosen to be different, there should be no overlap between the areas occupied by the tables.

c) Once the SORT operation has been completed the ‘Operation Complete Flag’ M8029 is turned ON. For the complete sort of a data table the SORT instruction will be processed m1times.

d) During a SORT operation, the data in the SORT table must not be changed. If the data is changed, this may result in an incorrectly sorted table.

e) The SORT instruction may only be used **ONCE** in a program.

From the example instruction and the ‘data table’ below left, the following data manipulation will occur when ‘n’ is set to the identified field

Original

Table1st table sort when n= 2

2nd table sort when n=1

|                |   | FIELD (m2)    |               |               |
|----------------|---|---------------|---------------|---------------|
|                |   | 1             | 2             | 3             |
| RECORD<br>(m1) | 1 | (D100)<br>32  | (D104)<br>162 | (D108)<br>4   |
|                | 2 | (D101)<br>74  | (D105)<br>6   | (D109)<br>200 |
|                | 3 | (D102)<br>100 | (D106)<br>80  | (D110)<br>62  |
|                | 4 | (D103)<br>7   | (D107)<br>34  | (D111)<br>6   |

|                |   | FIELD (m2)    |               |               |
|----------------|---|---------------|---------------|---------------|
|                |   | 1             | 2             | 3             |
| RECORD<br>(m1) | 1 | (D100)<br>74  | (D104)<br>6   | (D108)<br>200 |
|                | 2 | (D101)<br>7   | (D105)<br>34  | (D109)<br>6   |
|                | 3 | (D102)<br>100 | (D106)<br>80  | (D110)<br>62  |
|                | 4 | (D103)<br>32  | (D107)<br>162 | (D111)<br>4   |

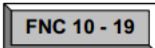
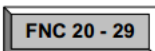
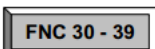
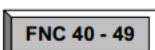
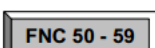
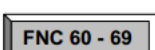
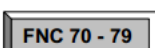
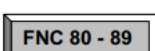
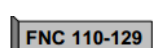
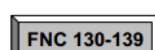
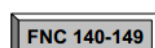
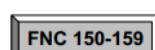
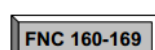
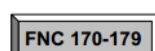
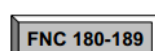
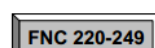
|                |   | FIELD (m2)    |               |               |
|----------------|---|---------------|---------------|---------------|
|                |   | 1             | 2             | 3             |
| RECORD<br>(m1) | 1 | (D100)<br>7   | (D104)<br>34  | (D108)<br>6   |
|                | 2 | (D101)<br>32  | (D105)<br>162 | (D109)<br>4   |
|                | 3 | (D102)<br>74  | (D106)<br>6   | (D110)<br>200 |
|                | 4 | (D103)<br>100 | (D107)<br>80  | (D111)<br>62  |



— 禾川股份 — HCFA CORPORATION

---

## Applied Instructions:

|                                                                                     |                                                                                    |                                                                                                                        |       |
|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|-------|
| 1.                                                                                  |   | Program Flow                                                                                                           | 5-4   |
| 2.                                                                                  |   | Move And Compare                                                                                                       | 5-16  |
| 3.                                                                                  |   | Arithmetic And Logical Operations (+, -, ×, ÷)                                                                         | 5-24  |
| 4.                                                                                  |   | Rotation And Shift                                                                                                     | 5-34  |
| 5.                                                                                  |   | Data Operation                                                                                                         | 5-42  |
| 6.                                                                                  |   | High Speed Processing                                                                                                  | 5-52  |
| 7.                                                                                  |  | Handy Instructions                                                                                                     | 5-66  |
|  | 8.                                                                                 | <br>External FX I/O Devices         | 5-80  |
|                                                                                     | 9.                                                                                 | <br>External FX Serial Devices      | 5-94  |
|                                                                                     | 10.                                                                                | <br>Floating Point 1 & 2            | 5-110 |
|                                                                                     | 11.                                                                                | <br>Trigonometry (Floating Point 3) | 5-118 |
|                                                                                     | 12.                                                                                | <br>Data Operations 2               | 5-122 |
|                                                                                     | 13.                                                                                | <br>Positioning Control             | 5-126 |
|                                                                                     | 14.                                                                                | <br>Real Time Clock Control         | 5-136 |
|                                                                                     | 15.                                                                                | <br>Gray Codes                      | 5-146 |
|                                                                                     | 16.                                                                                | <br>Additional Functions            | 5-146 |
|                                                                                     | 17.                                                                                | <br>In-line Comparisons             | 5-150 |

### Symbols list:

D - Destination device.

S - Source device. m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D 1 ,S3 or for lists/abled devices D3+0, S +9etc.

MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ -An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D ☆☆☆ -An instruction modified to operate in 32 bit operation.

D☆☆☆P -A 32 bit mode instruction modified to use pulse (single) operation

 - A repetitive instruction which will change the destination value on every scan

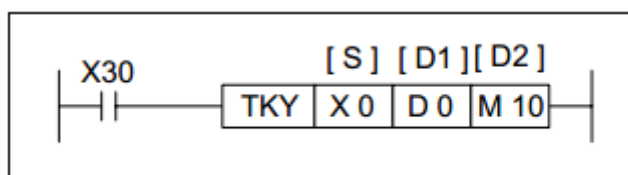
unless modified by the pulse function.

✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

### 5.8.1 TKY (FNC 70)

| Mnemonic                         | Function                                                             | Operands                                                                   |                                                                                       |                                                                         | Program steps                            |
|----------------------------------|----------------------------------------------------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------|------------------------------------------|
|                                  |                                                                      | S                                                                          | D1                                                                                    | D2                                                                      |                                          |
| TKY<br>FNC 70<br>(Ten key input) | Reads 10 devices with associated decimal values into a single number | X, Y, M, S<br>Note: uses 10 consecutive devices (identified as S+0 to S+9) | KnY, KnM, KnS, T, C, D, V, Z<br>Note: uses 2 consecutive devices for 32 bit operation | Y, M, S<br>Note: uses 11 consecutive devices (identified D2+0 to D2+10) | TKY:<br>7 steps<br><br>DTKY:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction can read from 10 consecutive devices(S +0 to

S +9) and will store an entered numeric string in device D 1 .

#### Points to note:

a) When a source device becomes active its associated destination (bit) device D2 also becomes active. This destination device will remain active until another source device is operated. Each source device maps directly to its own D2 device, i.e. S +0maps to

D2+0 ,S+7 maps to D2+7 etc. These in turn, map directly to decimal values which are then stored in the destination data devices specified by D1 .

b) One source device may be active at any one time. The destination device D2+10 is used to signify that a key (one of the 10 source devices) has been pressed. D 2+10 will remain active for as long as the key is held down. When the TKY instruction is active, every press of a key adds that digit to the stored number in D1 . When the TKY is OFF, all of the D 2 devices are reset, but the data value in D 1 remains intact.

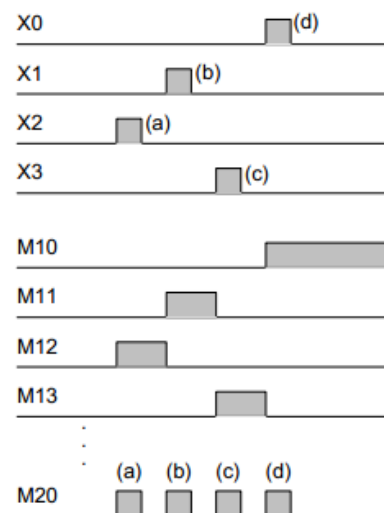
c) When the TKY instruction is used with 16 bit operation, D 1 can store numbers from 0 to 9,999 i.e. max. 4 digits. When the DTKY instruction is used (32 bit operation) values of 0 to 99,999,999 (max. 8 digits) can be accommodated in two consecutive devices D 1 and D 1+1 . In both cases if the number to be stored exceeds the allowable ranges, the highest digits will overflow until an allowable number is reached. The overflowed digits are lost and can no longer be accessed by the user. Leading zero’ s are not accommodated, i.e. 0127 will actually be stored as

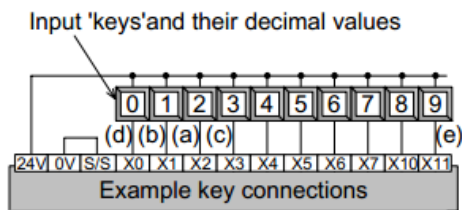
127 only.

d) The TKY instruction may only be used ONCE .

e) Using the above instruction as a brief example:

If the ‘ keys’ identified (a) to (d) are pressed in that order the number 2,130 will be entered into D1 .If the key identified as (e) is then pressed the value in D 1 will become 1,309. The initial ‘2’ has been lost.

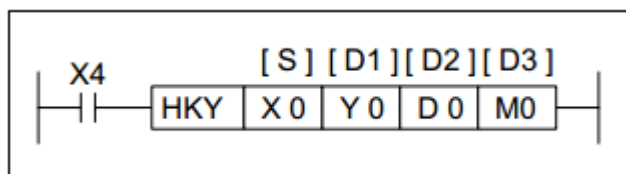




## 5.8.2 HKY (FNC 71)

| Mnemonic                                 | Function                                                                         | Operands                               |                                        |                                                                        |                                             | Program steps                            |
|------------------------------------------|----------------------------------------------------------------------------------|----------------------------------------|----------------------------------------|------------------------------------------------------------------------|---------------------------------------------|------------------------------------------|
|                                          |                                                                                  | S                                      | D1                                     | D2                                                                     | D3                                          |                                          |
| HKY<br>FNC 71<br>(Hexadecimal key input) | Multiplexes inputs and outputs to create a numeric keyboard with 6 function keys | X,<br>Note: uses 4 consecutive devices | Y,<br>Note: uses 4 consecutive devices | T, C, D, V, Z<br>Note: uses 2 consecutive devices for 32 bit operation | Y, M, S<br>Note: uses 8 consecutive devices | HKY:<br>9 steps<br><br>DHKY:<br>17 steps |

|                  |                  |         |       |                          |
|------------------|------------------|---------|-------|--------------------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Operation Complete M8029 |
|------------------|------------------|---------|-------|--------------------------|



Operation 1 - Standard:

This instruction creates a multiplex of 4 outputs (D1) and

4 inputs (S) to read in 16 different devices. Decimal values of 0 to 9 can be stored while 6 further function flags may be set.

### Points to note:

a) Each of the first 10 multiplexed source devices (identified as 0 to 9) map directly to decimal values 0 to 9. When entered, i.e. a source device is activated, then its associated decimal value is added to the data string currently stored in D2 . Activation of any of these keys causes bit device D 3+7 to turn ON for the duration of that key

press.

b) The last 6 multiplexed source devices (identified as function keys A to F) are used to set bit devices D3+0 to D 3+5 respectively. These bit flags, once set ON, remain ON until the next function key has been activated. Activation of any of these keys causes bit device D3+6 to turn ON for the duration of that key press.

c) In all key entry cases, when two or more keys are pressed, only the key activated first is effective. When the pressing of a key is sensed the M8029 (execution complete flag) is turned ON. When the HKY instruction is OFF, all D 3 devices are reset but data value D 2 remains intact.

d) When the HKY instruction is used with 16bit operation, D 2 can store numbers from 0 to 9,999 i.e. max. 4 digits.

When the DHKY instruction is used (32 bit operation) values of 0 to 99,999,999 (max. 8 digits) can be

accommodated in two consecutive devices D2 and D

2+1 .In both cases if the number to be stored exceeds the allowable ranges, the

highest digits will overflow until an allowable number is reached. The over-flowed

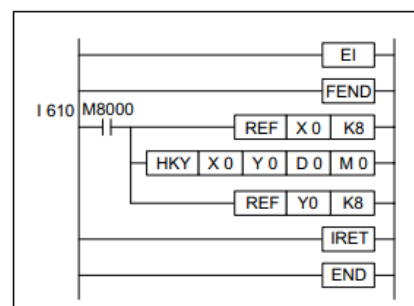
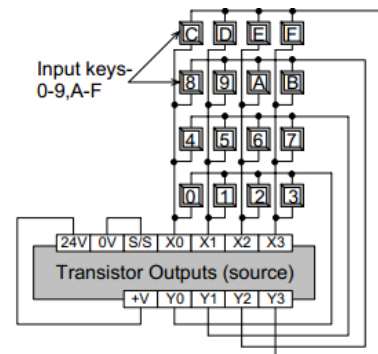
digits are lost and can no longer be accessed by the user. Leading zero ' s are not

accommodated, i.e. 0127 will actually be stored as 127 only. This operation is similar

to that of the TKY instruction.

e) The HKY instruction may only be used ONCE .

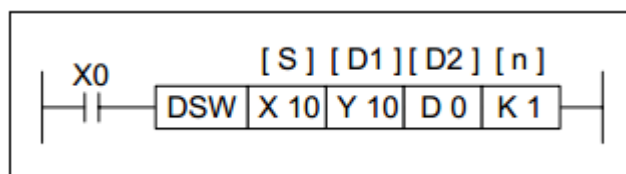
f) Normal operation requires 8 scans to read the



key inputs. To achieve a steady and repeatable performance, constant scan mode should be used, i.e. M8039 is set ON and a user defined scan time is written to register D8039. However, for a faster response the HKY instruction should be programmed in a timer interrupt routine as shown in the example opposite.

### 5.8.3 DSW (FNC 72)

| Mnemonic                          | Function                                                   | Operands                                    |                                          |                                                      |                                 | Program steps            |
|-----------------------------------|------------------------------------------------------------|---------------------------------------------|------------------------------------------|------------------------------------------------------|---------------------------------|--------------------------|
|                                   |                                                            | S                                           | D1                                       | D2                                                   | n                               |                          |
| DSW<br>FNC 72<br>(Digital switch) | Multiplexed reading of n sets of digital (BCD) thumbwheels | X<br>Note:<br>If n=2 then 8 devices else 4. | Y<br>Note:<br>uses 4 consecutive devices | T, C, D, V, Z<br>Note: If n=2 then 2 devices else 1. | K, H<br>☒<br>Note:<br>n= 1 or 2 | DSW:<br>9 steps          |
| 16 BIT OPERATION                  |                                                            | 32 BIT OPERATION                            |                                          | PULSE-P                                              | Flags                           | Operation Complete M8029 |



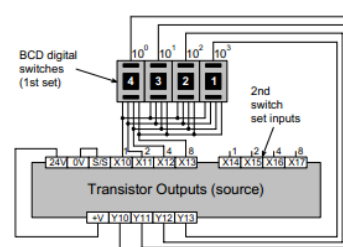
Operation:

This instruction multiplexes 4 outputs (D1) through 1 or 2(n)

sets of switches. Each set of switches consists of 4 thumbwheels providing a single digit input.

Points to note:

a) When n = 1 only one set of switches are read. The multiplex is completed by wiring the thumbwheels



in parallel back to 4 consecutive inputs from the head address specified in operand S.

The (4 digit) data read is stored in data device D 2 .

Continued on next page...

b) When  $n=2$ , two sets of switches are read. This configuration requires 8 consecutive inputs taken from the head address specified in

operand S. The data from the first set of switches,

i.e. those using the first 4 inputs, is read into data

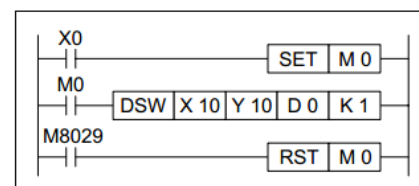
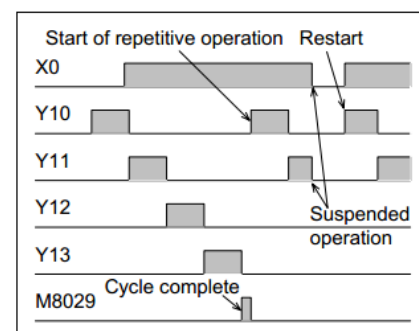
device D 2 . The data from the second set of

switches (again 4 digits) is read into data device

D2+1 .

c) The outputs used for multiplexing (D 1 )are

cycled for as long as the DSW instruction is driven.



After the completion of one reading, the execution complete flag M8029 is set. The

number of outputs used does not depend on the number of switches  $n$ .

d) If the DSW instruction is suspended during mid-operation, when it is restarted it will start from the beginning of its cycle and not from its last status achieved.

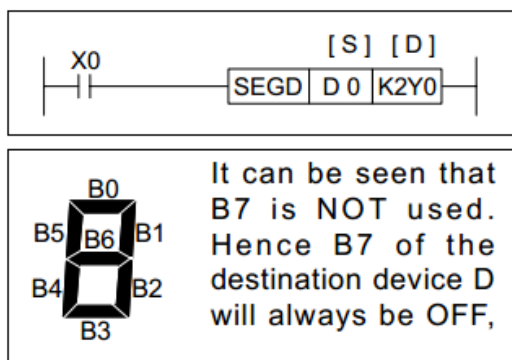
e) It is recommended that transistor output units are used with this instruction.

However, if the program technique at the right is used, relay output units can be successfully operated as the outputs will not be continually active.

f). HCA1P/ HCA2P/HCA2C units can operate an Unlimited number of DSW instructions.

#### 5.8.4 SEGD (FNC 73)

| Mnemonic                                  | Function                                                               | Operands                                                                         |                                                                            | Program steps              |
|-------------------------------------------|------------------------------------------------------------------------|----------------------------------------------------------------------------------|----------------------------------------------------------------------------|----------------------------|
|                                           |                                                                        | S                                                                                | D                                                                          |                            |
| SEGD<br>FNC 73<br>(Seven segment decoder) | Hex data is decoded into a format used to drive seven segment displays | K, H<br>KnX, KnY, KnM, KnS,<br>T, C, D, V, Z<br>Note: Uses only the lower 4 bits | KnY, KnM, KnS,<br>T, C, D, V, Z<br>Note: The upper 8 bits remain unchanged | SEGD,<br>SEGDP:<br>5 steps |
| 16 BIT OPERATION                          |                                                                        | 32 BIT OPERATION                                                                 | PULSE-P                                                                    | Flags Zero M8020           |



#### Operation:

A single hexadecimal digit (0 to 9, A to F)

occupying the lower 4 bits of source device

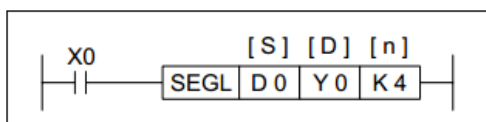
S is decoded into a data format used to

drive a seven segment display. A

representation of the hex digit is then displayed. The decoded data is stored in the lower 8 bits of destination device D. The upper 8 bits of the same device are not written to. The diagram opposite shows the bit control of the seven segment display. The active bits correspond to those set to 1 in the lower 8 bits of the destination device D.

### 5.8.5 SEGL (FNC 74)

| Mnemonic                                           | Function                                                                                     | Operands                                       |                                                                                             |                                                                                                 | Program steps               |
|----------------------------------------------------|----------------------------------------------------------------------------------------------|------------------------------------------------|---------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|-----------------------------|
|                                                    |                                                                                              | S                                              | D                                                                                           | n                                                                                               |                             |
| SEGL<br>FNC 74<br>(Seven<br>segment<br>with latch) | Writes data to<br>multiplexed single<br>digit displays - 4<br>digits per set,<br>max. 2 sets | K, H<br>KnX, KnY,<br>KnM, KnS<br>T, C, D, V, Z | Y<br>Note:<br>n = 0 to 3, 8<br>outputs are<br>used<br>n = 4 to 7, 12<br>outputs are<br>used | K, H, ☒<br>Note:<br>n= 0 to 3, 1 set<br>of 7 Seg active<br>n= 4 to 7, 2 sets<br>of 7 Seg active | SEGL:<br>7 steps            |
| 16 BIT OPERATION                                   |                                                                                              | 32 BIT OPERATION                               |                                                                                             | PULSE-P                                                                                         | Flags                       |
|                                                    |                                                                                              |                                                |                                                                                             |                                                                                                 | Operation<br>Complete M8029 |



#### Operation:

This instruction takes a source decimal value (S) and writes it to a set of 4

multiplexed, outputs (D). Because the logic used with latched seven segment displays varies between display manufactures, this instruction can be modified to suit most logic requirements. Configurations are selected depending on the value of n, see the following page.

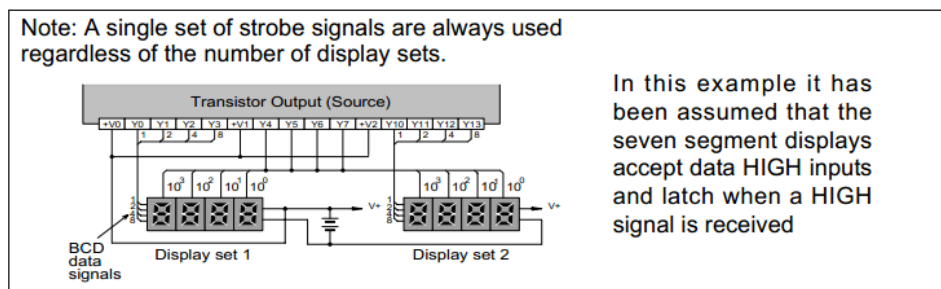
#### Points to note:

a) Data is written to a set of multiplexed outputs (D+0to D+7, 8 outputs) and hence seven segment displays. A set of displays consists of 4 single digit seven segment units. A maximum of two sets of displays can be driven with this instruction. When two sets are used the displays share the same strobe outputs (D +4 to D+7are the strobe outputs). An additional set of 4 output devices is required to supply the new data for the second set of displays (D +10 to D +13 , this is an octal addition). The strobe

outputs cause the written data to be latched at the seven segment display.

b) Source data within the range of 0 to 9,999 (decimal) is written to the multiplexed outputs. When one set of displays are used this data is taken from the device specified as operand S. When two sets of displays are active the source device S+1 supplies the data for the second set of displays. This data must again be within the range 0 to 9,999. When using two sets of displays the data is treated as two separate numbers and is not combined to provide a single output of 0 to 99,999,999.

c) The SEGL instruction takes 12 program scans to complete one output cycle regardless of the number of display sets used. On completion, the execution complete flag M8029 is set.



d) If the SEGL instruction is suspended during mid-operation, when it is restarted it will start from the beginning of its cycle and not from its last status achieved.

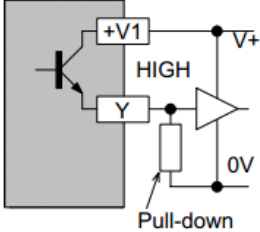
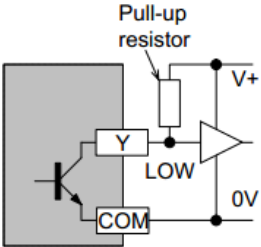
e) HCA1P & HCA2P & HCA2C units can operate an Unlimited number of SEGL instructions.

## Selecting the correct value for operand n

The selection of parameter n depends on 4 factors;

1) The logic type used for the PLC output

- 2) The logic type used for the seven segment data lines
- 3) The logic type used for the seven segment strobe signal

| Device considered           |                     | Positive logic                                                                                     | Negative logic                                                                                     |
|-----------------------------|---------------------|----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| PLC Logic                   |                     | Source output<br> | Sink output<br> |
|                             |                     | With a source output, when the output is HIGH the internal logic is '1'                            | With a sink output, when the output is LOW the internal logic is '1'                               |
| Seven segment Display logic | Strobe signal logic | Data is latched and held when this signal is HIGH, i.e. its logic is '1'                           | Data is latched and held when this signal is LOW, i.e. its logic is '1'                            |
|                             | Data signal logic   | Active data lines are held HIGH, i.e. they have a logic value of '1'                               | Active data lines are held LOW, i.e. they have a logic value of '1'                                |

There are two types of logic system available, positive logic and negative logic.

Depending on the type of system, i.e. which elements have positive or negative logic

the value of n can be selected from the table below with the final reference to the

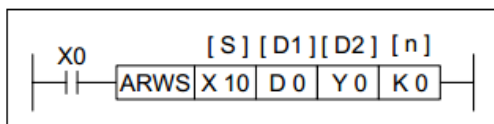
number of sets of seven segment displays being used.

| PLC Logic         | Seven segment display logic |                 | n             |                |
|-------------------|-----------------------------|-----------------|---------------|----------------|
|                   | Data Logic                  | Strobe logic    | 1 display set | 2 display sets |
| Positive (Source) | Positive (High)             | Positive (High) | 0             | 4              |
| Negative (Sink)   | Negative (Low)              | Negative (Low)  |               |                |
| Positive (Source) | Positive (High)             | Negative (Low)  | 1             | 5              |
| Negative (Sink)   | Negative (Low)              | Positive (High) |               |                |
| Positive (Source) | Positive (High)             | Negative (Low)  | 2             | 6              |
| Negative (Sink)   | Negative (Low)              | Positive (High) |               |                |
| Positive (Source) | Positive (High)             | Positive (High) | 3             | 7              |
| Negative (Sink)   | Negative (Low)              | Negative (Low)  |               |                |

### 5.8.6 ARWS (FNC 75)

| Mnemonic                         | Function                                                 | Operands                                       |                                                           |                                       |                                                                 | Program steps    |
|----------------------------------|----------------------------------------------------------|------------------------------------------------|-----------------------------------------------------------|---------------------------------------|-----------------------------------------------------------------|------------------|
|                                  |                                                          | S                                              | D1                                                        | D2                                    | n                                                               |                  |
| ARWS<br>FNC 75<br>(Arrow switch) | Creates a user defined, (4 key) numeric data entry panel | X, Y, M, S<br>Note: uses 4 consecutive devices | T, C, D, V, Z<br>Note: data is stored in a decimal format | Y<br>Note: uses 8 consecutive devices | K, H<br><input checked="" type="checkbox"/><br>Note: n= 0 to 3, | ARWS:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



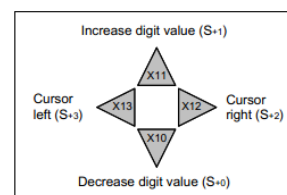
#### Operation:

This instruction displays the contents of a

single data device D1 on a set of 4 digit, seven segment displays. The data within D1 is actually in a standard decimal format but is automatically converted to BCD for display on the seven segment units. Each digit of the displayed number can be selected and edited. The editing procedure directly changes the value of the device specified as D1 .

Points to note:

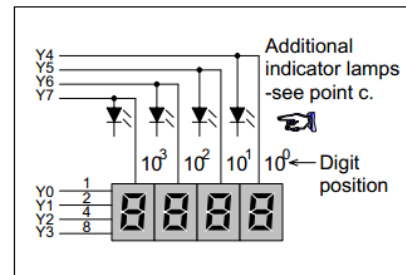
a) The data stored in destination device D1 can have a value from the range 0 to 9,999 (decimal), i.e. 4 digit data. Each digits data value, can be incremented (S+1) or decremented



(S+0) by pressing the associated control keys. The edited numbers automatically

‘wrap-around’ from 9 - 0 - 1 and 1 - 0 - 9. The digit data is displayed by the lower 4 devices from D 2 ,i.e.D 2+0 to D 2+3 .

b) On initial activation of the ARWS instruction, the digit in the numeric position  $10^3$  is currently selected. Each digit position can be sequentially ‘cursored through’ by moving to the left (S +2)



or to the right (S +3). When the last digit is reached, the ARWS instruction automatically wraps the cursor position around, i.e. after position  $10^3$ , position  $10^0$  is selected and vice-versa. Each digit is physically selected by a different ‘strobe’ output.

c) To aid the user of an operation panel controlled with the ARWS instruction, additional lamps could be wired in parallel with the strobe outputs for each digit. This would indicate which digit was currently selected for editing.

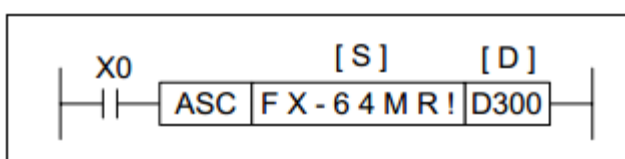
d) The parameter n has the same function as parameter n of the SEGL instruction, ‘Selecting the correct value for operand n’. Note: as the ARWS instruction only controls one set of displays only values of 0 to 3 are valid for n.

e) The ARWS instruction can be used ONCE. This instruction should only be used on transistor output PLC’s.

### 5.8.7 ASC (FNC 76)

| Mnemonic                                 | Function                                                           | Operands                                                                                                               |                                                | Program steps    |
|------------------------------------------|--------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------|------------------|
|                                          |                                                                    | S                                                                                                                      | D1                                             |                  |
| ASC<br>FNC 76<br>(ASCII code conversion) | An entered alphanumeric string can be converted to its ASCII codes | Alphanumeric data e.g. 0-9, A - Z and a - z etc.<br>Note: Only one, 8 character string may be entered at any one time. | T, C, D<br>Note:<br>uses 4 consecutive devices | ASC<br>: 7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The source data string S consists of up to 8 characters taken from the

printable ASCII character (Char) set. If less than 8 Char are used,

The difference is made up with null Char(ASCII 00).

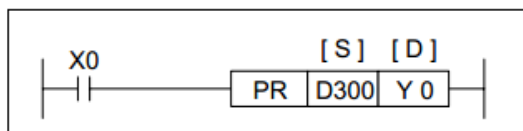
The source data is converted to its associated ASCII codes. The codes are then stored in the destination devices D, see example shown below.

| D    | Byte   |        |
|------|--------|--------|
|      | High   | Low    |
| D300 | 58 (X) | 46 (F) |
| D301 | 36 (6) | 2D (-) |
| D302 | 4D (M) | 34 (4) |
| D303 | 21 (!) | 52 (R) |

**Note:** ASCII Char cannot be entered from a hand held programmer.

### 5.8.8 PR (FNC 77)

| Mnemonic                | Function                                          | Operands                                                                                                                   |                                         | Program steps                     |
|-------------------------|---------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|-----------------------------------|
|                         |                                                   | S                                                                                                                          | D1                                      |                                   |
| PR<br>FNC 77<br>(Print) | Outputs ASCII data to items such as display units | T, C, D<br>Note: 8 byte mode (M8027=OFF) uses 4 consecutive devices<br>16 byte mode (M8027= ON) uses 8 consecutive devices | Y<br>Note: uses 10 consecutive devices. | PR:<br>5 steps                    |
| 16 BIT OPERATION        |                                                   | 32 BIT OPERATION                                                                                                           | PULSE-P                                 | Flags<br>Operation Complete M8029 |



#### Operation:

Source data (stored as ASCII values) is

read byte by byte from the source data

devices. Each byte is mapped directly to the first 8 consecutive destination devices D +0to D +7). The final two destination bits provide a strobe signal (D +10 , numbered in octal) and an execution/busy flag (D +11 ,in octal).

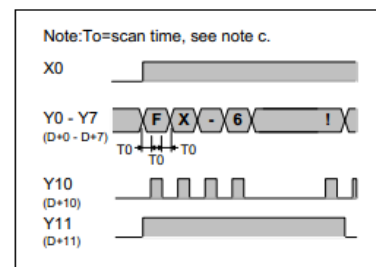
#### Points to note:

- The source byte-data maps the lowest bit to the first destination device D+0. Consequently the highest bit of the byte is sent to destination device D +7.
- The PR instruction may only be used TWICE in a sequence program. This instruction should only be used on transistor output PLC ' s. The PR instruction will not automatically repeat its operation unless the drive input has been turned OFF and ON again.

c) The operation of the PR instruction is program scan dependent. Under standard circumstances it takes 3 program scans to send 1 byte. However, for a faster operation the PR instruction could be written into a timer interrupt routine similar to the one demonstrated for HKY.

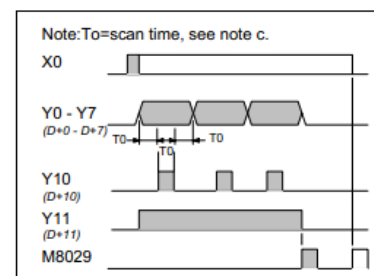
d) 8 byte operation has the following timing diagram.

It should be noted that when the drive input (in the example X0) is switched OFF the PR instruction will cease operation. When it is restarted the PR



instruction will start from the beginning of the message string. Once all 8 bytes have been sent the execution/busy flag is dropped and the PR instruction suspends operation.

e) 16 byte operation requires the special auxiliary flag M8027 to be driven ON (it is recommended that M8000 is used as a drive input). In this operation mode the drive input (in the example X0) does not



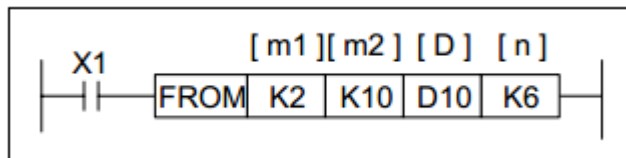
have to be active all of the time. Once the PR instruction is activated it will operate continuously until all 16 bytes of data have been sent or the value 00H (null) has been

sent. Once the operation is complete the execution/busy flag (D +11 ,octal)is turned OFF and M8029 the execution complete flag is set.

## 5.8.9 FROM (FNC 78)

| Mnemonic                 | Function                                                               | Operands                      |                                                                                       |                                    |                                                                         | Program steps                                               |
|--------------------------|------------------------------------------------------------------------|-------------------------------|---------------------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------------|-------------------------------------------------------------|
|                          |                                                                        | m1                            | m2                                                                                    | D                                  | n                                                                       |                                                             |
| FROM<br>FNC 78<br>(FROM) | Read data from the buffer memories of attached special function blocks | K, H ☒<br>Note:<br>m1= 0 to 7 | K, H ☒<br>Note:<br>m2 = FX <sub>(2C)</sub><br>0 to 31,<br>FX <sub>2N</sub> 0 to 32767 | KnY, KnM,<br>KnS, T, C,<br>D, V, Z | K, H ☒<br>Note:<br>16 bit op:<br>n= 1 to 32<br>32 bit op:<br>n= 1 to 16 | FROM,<br>FROMP:<br>9 steps<br>DFROM,<br>DFROMP:<br>17 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The FROM instruction reads n words of data starting from the

buffer memory address m2 of the special function block with the logical block position specified as m 1, The read data is stored in the PLC at head address D for n word devices.

Points to note:

a) All special function blocks which are addressable with the FROM/TO instructions

are connected to the extension bus on the right hand side of the PLC. Each special function block can be inserted at any point within the chain of extended units (as long as the system configuration rules are not broken). Each special function block is consecutively addressed from 0 to 7 beginning with the one closest to the base unit

b) Each special function unit has different buffer memory registers. These often have a dedicated use for each individual unit. Before any reading or writing of data is undertaken ensure that the correct buffer memory allocations for the unit used are known. m 2 : This defines the head address of the (special function blocks) buffer memories being accessed. m 2 may have a value from the range 0 to 31. n: This identifies the number of words which are to be transferred between the special function block and the PLC base unit. n may have a value of 1 to 31 for 16 bit operation but a range of 1 to 16 is available for 32 bit operation.

c) The destination head address for the data read FROM the special function block is specified under the D operand; and will occupy n further devices.

d) This instruction will only operate when the drive input is energized.

e) Users of all PLC models have the option of allowing interrupts to occur immediately, i.e. during the operation of the FROM/TO instructions or to wait until the completion of the current FROM/ TO instruction. This is achieved by controlling the special auxiliary flag M8028. The following table identifies certain points associated with this control and operation.

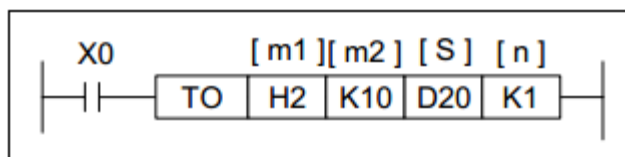
| Interruptation Disabled                                                                                              | Interruptation Enabled                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| M8028 = OFF                                                                                                          | M8028 = ON                                                                                                                                                           |
| Jumps called by interrupt operation are delayed until the completion of the data transfer of the FROM/TO instruction | Jumps called by interrupt operation occur immediately                                                                                                                |
| A small delay of (800m +200) μsec can be expected in the worst case. Note: m = the number of 32 bit words            | Data transfer will resume upon return from the interrupt program. This may not be desirable if a FROM/TO instruction has been programmed within the called interrupt |
| Ensures that FROM/TO instructions included in an interrupt program will not interact with others elsewhere           | M8028 should only be used when a very short delay is required in applications where timing and accuracy's are important                                              |

Users of HCA1P have no option for interruption of the FROM/TO instructions and hence always operate in a mode equivalent to having M8028 switched OFF.

## 5.8.10 TO (FNC 79)

| Mnemonic             | Function                                                               | Operands                      |                                    |                                        |                                                                         | Program steps                                     |
|----------------------|------------------------------------------------------------------------|-------------------------------|------------------------------------|----------------------------------------|-------------------------------------------------------------------------|---------------------------------------------------|
|                      |                                                                        | m1                            | m2                                 | S                                      | n                                                                       |                                                   |
| TO<br>FNC 79<br>(TO) | Writes data to the buffer memories of attached special function blocks | K, H ☒<br>Note:<br>m1= 0 to 7 | K, H ☒<br>Note:<br>m2 = 0 to 32767 | K,H, KnX, KnY, KnM, KnS, T, C, D, V, Z | K, H ☒<br>Note:<br>16 bit op:<br>n= 1 to 32<br>32 bit op:<br>n= 1 to 16 | TO, TOP:<br>9 steps<br><br>DTO, DTOP:<br>17 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The TO instruction writes n words of data to the head buffer

memory address m2 of the special function block with the logical block position specified in m1 . The written data is taken from the PLC ' s head address S for n word devices.

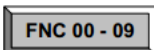
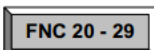
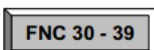
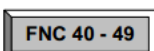
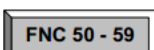
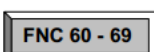
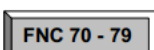
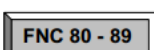
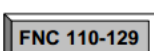
Points to note:

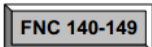
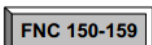
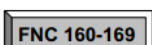
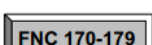
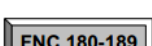
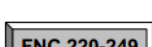
All points are the same as the FROM instruction (see previous page) except point c)

which is replaced by the following:

a) The source head address for the data written TO the special function block is specified under the S operand.

#### Applied Instructions:

|                                                                                     |                                                                                     |                                                                                     |                            |      |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------|------|
| 1.                                                                                  |   | Program Flow                                                                        | 5-4                        |      |
| 2.                                                                                  |  | Move And Compare                                                                    | 5-16                       |      |
| 3.                                                                                  |  | Arithmetic And Logical Operations (+, -, ×, ÷)                                      | 5-24                       |      |
| 4.                                                                                  |  | Rotation And Shift                                                                  | 5-34                       |      |
| 5.                                                                                  |  | Data Operation                                                                      | 5-42                       |      |
| 6.                                                                                  |  | High Speed Processing                                                               | 5-52                       |      |
| 7.                                                                                  |  | Handy Instructions                                                                  | 5-66                       |      |
| 8.                                                                                  |  | External FX I/O Devices                                                             | 5-80                       |      |
|  | 9.                                                                                  |  | External FX Serial Devices | 5-94 |
| 10.                                                                                 |  | Floating Point 1 & 2                                                                | 5-110                      |      |

|     |                                                                                   |                                 |       |
|-----|-----------------------------------------------------------------------------------|---------------------------------|-------|
| 11. |  | Trigonometry (Floating Point 3) | 5-118 |
| 12. |  | Data Operations 2               | 5-122 |
| 13. |  | Positioning Control             | 5-126 |
| 14. |  | Real Time Clock Control         | 5-136 |
| 15. |  | Gray Codes                      | 5-146 |
| 16. |  | Additional Functions            | 5-146 |
| 17. |  | In-line Comparisons             | 5-150 |



### Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D 1 ,S3 or for lists/abled devices D3+0, S +9etc.

MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a

number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D ☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

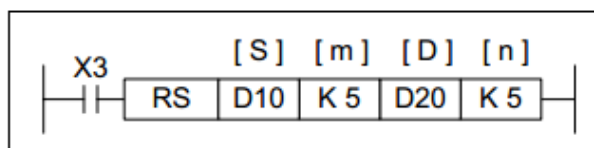
✈ - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

✉ - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

### 5.9.1 RS (FNC 80)

| Mnemonic                                            | Function                                                                  | Operands                        |                                               |   |                                              | Program steps |
|-----------------------------------------------------|---------------------------------------------------------------------------|---------------------------------|-----------------------------------------------|---|----------------------------------------------|---------------|
|                                                     |                                                                           | S                               | m                                             | D | n                                            |               |
| RS<br>FNC 80<br>(Serial Communications instruction) | Used to control serial communications from/to the programmable controller | D<br>(including file registers) | K, H, D<br>✉<br>m = 1 to 256, FX2N 1 to 4096. | D | K, H, D<br>✉<br>m = 1 to 256, FX2N 1 to 4096 | RS: 9 steps   |

|                  |                  |         |       |                                           |
|------------------|------------------|---------|-------|-------------------------------------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Zero M8020<br>Borrow M8021<br>Carry M8022 |
|------------------|------------------|---------|-------|-------------------------------------------|



Operation:

This instruction performs the direct control of communications over

communication adapters which connect to the left hand port of the Main Processing Unit.

**Points to note:**

- a) This instruction has many automatically defined devices. These are listed in the boxed column to the right of this page.
- b) The RS instruction has two parts, send (or transmission) and receive. The first elements of the RS instruction specify the transmission data buffer (S) as a head address, which contains m number of elements in a sequential stack. The specification of the receive data area is contained in the last two parameters of the RS instruction. The destination (D) for received messages has a buffer or stack length of n data elements. The size of the send and receive buffers dictates how large a single message can be. Buffer sizes may be updated at the following times:
  - 1) Transmit buffer - before transmission occurs, i.e. before M8122 is set ON
  - 2) Receive buffer - after a message has been received and before M8123 is reset.
- c) Data cannot be sent while a message is being received, the transmission will be delayed - see M8121.
- d) More than one RS instruction can be programmed but only one may be active at any one time.

e) Refer to the HC Communications Manual when using this function

### **Assigned devices**

#### **Data devices:**

D8120 - Contains the configuration parameters for communication, i.e. Baud rate, Stop bits etc. Full details over the page

D8122 - Contains the current count of the number of remaining bytes to be sent in the currently transmitting message.

D8123 - Contains the current count of the number of received bytes in the ‘incoming’ message.

D8124 - Contains the ASCII code of the character used to signify a message header - default is ‘STX’ ,02 HEX.

D8125 - Contains the ASCII code of the character used to signify a message terminator -default is ‘ ETX ’ ,03HEX.

#### **Operational flags:**

M8121 - This flag is ON to indicate a transmission is being delayed until the current receive operation is completed.

M8122 - This flag is used to trigger the transmission of data when it is set ON.

M8123 - This flag is used to identify (when ON) that a complete message has been received.

M8124 - Carrier detect flag. This flag is for use with HCFA Main Processing Units. It is typically useful in modem communications

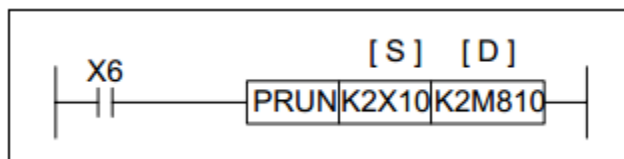
M8161 - 8 or 16 bit operation mode ON = 8 bit mode where only the lower 8 bits in

each source or destination device are used, i.e. only one ASCII character is stored in one data register OFF = 16bit mode where all of the available source/ destination register is used, i.e. two ASCII characters are stored in each data register

### 5.9.2 RUN (FNC 81)

| Mnemonic                         | Function                                                   | Operands                                                                                                                                     |          | Program steps                                              |
|----------------------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|----------|------------------------------------------------------------|
|                                  |                                                            | S                                                                                                                                            | D        |                                                            |
| PRUN<br>FNC 81<br>(Parallel run) | Used to control the FX parallel link adapters: FX2-40AW/AP | KnX, KnM<br><br>Note:<br>n = 1 to 8<br>For ease and convenience, the head address bit should be a multiple of '10', e.g. X10, M100, Y30 etc. | KnY, KnY | PRUN,<br>PRUNP:<br>5 steps<br>DPRUN,<br>DPRUNP:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



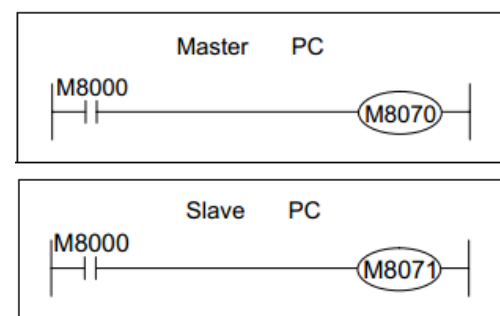
Operation:

This instruction is used with the parallel link adapters. It allows

source data to be moved into the bit transmission area. The actual control of the parallel link communication is by special M flags.

Points to note:

a) Parallel link communications automatically take place when both systems are ‘linked’ and the Master station (M8070), Slave station flags (M8071) have been set ON (there is no need to have a PRUN instruction for communications).



There can only be one of each type of station as this system connects only two PLC's.

The programs shown opposite should be inserted into the appropriate PLC's programs.

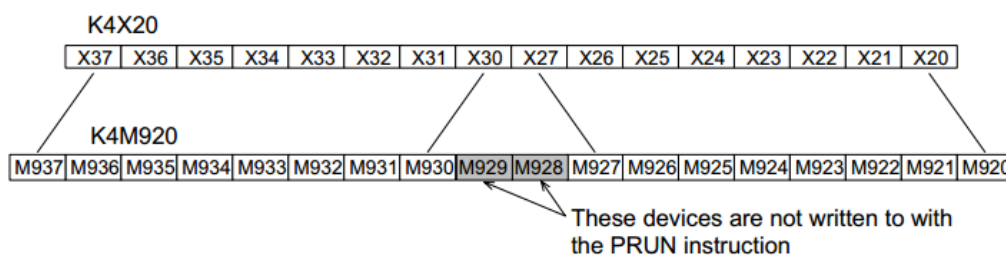


Once the station flags have been set, they can only be cleared by either forcibly resetting them when the PLC is in STOP mode or turning the power OFF and ON again.

b) During automatic communications the following data is 'swapped' between the Master and Slave PLC's.

| Master station |                           | Communication direction | Slave Station             |            |
|----------------|---------------------------|-------------------------|---------------------------|------------|
| M8070 = ON     | Bit Data                  |                         | Bit Data                  | M8071 = ON |
|                | M800 to M899 (100 points) | →                       | M800 to M899 (100 points) |            |
|                | M900 to M999 (100 points) | ←                       | M900 to M999 (100 points) |            |
|                | Data words                |                         | Data words                |            |
|                | D490 to D499 (10 points)  | →                       | D490 to D499 (10 points)  |            |
|                | D500 to D509 (10 points)  | ←                       | D500 to D509 (10 points)  |            |

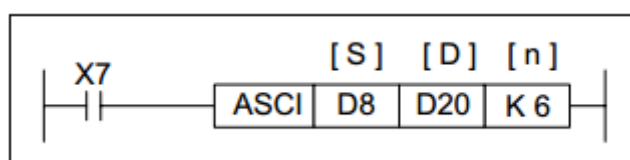
c) The PRUN instruction enables data to be moved into the bit transmission area or out of the (bit) data received area. The PRUN instruction differs from the move statement in that it operates in octal. This means if K4X20 was moved using the PRUN instruction to K4M920, data would not be written to M928 and M929 as these devices fall outside of the octal counting system. This can be seen in the diagram below.



### 5.9.3 ASCII (FNC 82)

| Mnemonic                                         | Function                                                 | Operands                                        |                             |                                        | Program steps               |
|--------------------------------------------------|----------------------------------------------------------|-------------------------------------------------|-----------------------------|----------------------------------------|-----------------------------|
|                                                  |                                                          | S                                               | D                           | n                                      |                             |
| ASCII<br>FNC 82<br>(Converts<br>HEX to<br>ASCII) | Converts a data<br>value from<br>hexadecimal to<br>ASCII | K, H, KnX,<br>KnY,<br>KnM, KnS<br>T, C, D, V, Z | KnY, KnM,<br>KnS<br>T, C, D | K, H<br><br>Note:<br>n = 1 to 256<br>☒ | ASCII,<br>ASCIP:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction reads n

hexadecimal data characters

from head source address (S) and converts them in to the equivalent ASCII code.

This is then stored at the destination (D) for n number of bytes.

#### Points to note:

Please note that data is converted ‘as read’ , i.e. using the example above with the following data in (D9,D8) ABCDH,EF26H. Taking the first n hexadecimal characters (digits) from the right (in this case n= 6) and converting them to ASCII will store values in 6 consecutive bytes from D20, i.e. D20 = (67, 68), D21 = (69, 70) and D22 = (50, 54) respectively. In true characters symbols that would be read asCDEF26.

This can be shown graphically as in the table to the right. Please take special note that the source data (S) read from the most significant device to the least significant. While the destination data (D) is read in the opposite direction.

| Source (S) | Data   |                 |       |            |     |        |
|------------|--------|-----------------|-------|------------|-----|--------|
| D9         | b12-15 | Destination (D) |       | ASCII Code |     | Symbol |
|            | b8-11  |                 |       | HEX        | DEC |        |
|            | b4-7   | D20             | b8-15 | 43         | 67  | 'C'    |
|            | b0-3   |                 | b0-7  | 44         | 68  | 'D'    |
| D8         | b12-15 | D21             | b8-15 | 45         | 69  | 'E'    |
|            | b8-11  |                 | b0-7  | 46         | 70  | 'F'    |
|            | b4-7   | D22             | b8-15 | 32         | 50  | 'Z'    |
|            | b0-3   |                 | b0-7  | 36         | 54  | '6'    |

The ASCII instruction can be used with the M8161, 8 bit/16bit mode flag. The example to the right shows the effect when M8161 is OFF.

If M8161 was set ON, then only the lower destination byte (b0-7) would be used to store data and hence 6 data registers would be required (D20 through D25).



## ASCII Character Codes

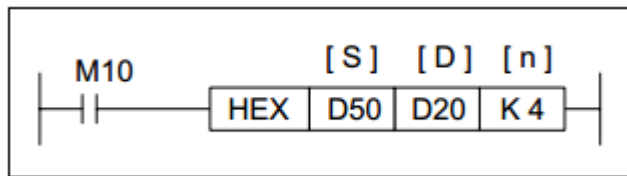
The table below identifies the usable hexadecimal digits and their associated ASCII codes.

| HEX Character    |     | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | A   | B   | C   | D   | E   | F   |
|------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ASCII Code       | HEX | 30  | 31  | 32  | 33  | 34  | 35  | 36  | 37  | 38  | 39  | 41  | 42  | 43  | 44  | 45  | 46  |
|                  | DEC | 48  | 49  | 50  | 51  | 52  | 53  | 54  | 55  | 56  | 57  | 65  | 66  | 67  | 68  | 69  | 70  |
| Character Symbol |     | '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' | 'A' | 'B' | 'C' | 'D' | 'E' | 'F' |

### 5.9.4 HEX (FNC 83)

| Mnemonic                                 | Function                                                        | Operands                            |                                |                                        | Program steps            |
|------------------------------------------|-----------------------------------------------------------------|-------------------------------------|--------------------------------|----------------------------------------|--------------------------|
|                                          |                                                                 | S                                   | D                              | n                                      |                          |
| HEX<br>FNC 83<br>(Converts ASCII to HEX) | Converts a data value from ASCII in to a hexadecimal equivalent | K, H, KnX, KnY, KnM, KnS<br>T, C, D | KnY, KnM, KnS<br>T, C, D, V, Z | K, H<br><br>Note:<br>n = 1 to 256<br>☒ | HEX,<br>HEXP:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction reads n ASCII data bytes from head source

address (S) and converts them in to the equivalent Hexadecimal character. This is then stored at the destination (D) for n number of bytes.

## Points to note:

Please note that this instruction ‘works in reverse’ to the ASCI instruction, i.e. ASCII data stored in bytes is converted into associated hexadecimal characters. The HEX instruction can be used with the M8161

8bit/16bit flag. In this case the source data (S) is read from either the lower byte (8bits)

| Source (S)           | ASCII Code |     | Symbol | Destination (D) | Data |
|----------------------|------------|-----|--------|-----------------|------|
|                      | HEX        | DEC |        |                 |      |
| D51<br>b8-15<br>b0-7 | 43         | 67  | 'C'    | b12-15          | -    |
|                      | 41         | 65  | 'A'    | b8-11           | -    |
| D50<br>b8-15<br>b0-7 | 42         | 66  | 'B'    | b4-7            | A    |
|                      | 31         | 49  | '1'    | b0-3            | 1    |

when M8161 is ON, or the whole word when M8161 is OFF i.e. using the example above with the following data in devices D50 and D51 respectively (43H,41H)(42H,31H) and assuming M8161 is ON. The ASCII data is converted to its hexadecimal equivalent and stored sequentially digit by digit from the destination head address. If M8161 had been OFF, then the contents of D20 would read CAB1H.



For further details regarding the use of the HEX instruction and about the

available ASCII data ranges, please see the following information point ‘ASCII Character Codes’ under the ASCII instruction on the previous page.

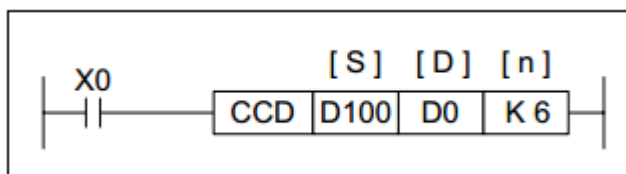


If an attempt is made to access an ASCII Code (HEX or Decimal) which falls outside of the ranges specified in the table on previous page, the instruction is not executed. Error 8067 is flagged in data register D8004 and error 6706 is identified in D8067. Care should be taken when using the M8161 flag, and additional in the specification of the number of element ‘n’ which are to be processed as these are the most likely places where this error will be caused.

#### 5.9.5 CCD (FNC 84)

| Mnemonic                      | Function                                     | Operands                      |                          |                                         | Program steps            |
|-------------------------------|----------------------------------------------|-------------------------------|--------------------------|-----------------------------------------|--------------------------|
|                               |                                              | S                             | D                        | n                                       |                          |
| CCD<br>FNC 84<br>(Check Code) | Checks the 'vertical' parity of a data stack | KnX, KnY, KnM, KnS<br>T, C, D | KnY, KnM, KnS<br>T, C, D | K, H<br>D<br>Note:<br>n = 1 to 256<br>☒ | CCD,<br>CCDP:<br>7 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction looks at a byte (8 bit) stack of data from head

address (S) for n bytes and checks the vertical bit pattern for parity and sums the total data stack. These two pieces of data are then stored at the destination (D).

Points to note:

- a) The SUM of the data stack is stored at destination D while the Parity for the data stack is stored at D+1.
- b) During the Parity check an even result is indicated by the use of a 0 (zero) while an odd parity is indicated by a 1 (one).
- c) This instruction can be used with the 8 bit/ 16 bit mode flag M8161. The following results will occur under these circumstances.

| M8161=OFF                |   |             |   |   |   |   |   |   |   |   |
|--------------------------|---|-------------|---|---|---|---|---|---|---|---|
| Source (S)               |   | Bit pattern |   |   |   |   |   |   |   |   |
| D100                     | H | FF          | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|                          | L | FF          | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D101                     | H | FF          | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|                          | L | 00          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D102                     | H | F0          | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
|                          | L | 0F          | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Vertical<br>parity<br>D1 |   |             | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SUM D0                   |   | 3FC         |   |   |   |   |   |   |   |   |

| M8161=ON                 |   |             |   |   |   |   |   |   |   |   |
|--------------------------|---|-------------|---|---|---|---|---|---|---|---|
| Source (S)               |   | Bit pattern |   |   |   |   |   |   |   |   |
| D100                     | L | FF          | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|                          | L | 00          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| D102                     | L | 0F          | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
|                          | L | F0          | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| D104                     | L | F0          | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
|                          | L | 0F          | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| Vertical<br>parity<br>D1 |   |             | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| SUM D0                   |   | 2FD         |   |   |   |   |   |   |   |   |

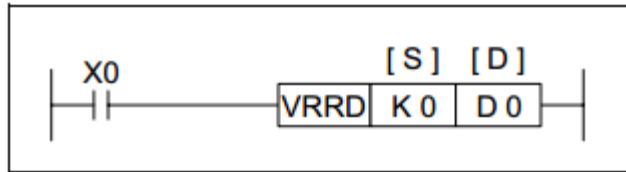
It should be noted that when M8161 is OFF ‘n’ represents the number of consecutive bytes checked by the CCD instruction. When M8161 is ON only the lower bytes of ‘n’ consecutive words are used.

The ‘SUM’ is quite simply a summation of the total quantity of data in the data stack. The Parity is checked vertically through the data stack as shown by the shaded areas.

#### 5.9.6 VRRD (FNC 85)

| Mnemonic                        | Function                                                      | Operands                                                                             |                                | Program steps              |
|---------------------------------|---------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------|----------------------------|
|                                 |                                                               | S                                                                                    | D                              |                            |
| VRRD<br>FNC 85<br>(Volume read) | Reads an analog value from 1 of 8 volume inputs on the FX-8AV | K, H<br>Note:<br>S= 0 to 7<br>corresponding to the 8 available volumes on the FX-8AV | KnY, KnM, KnS<br>T, C, D, V, Z | VRRD,<br>VRRDP:<br>5 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|
|------------------|------------------|---------|



Operation:

The identified volume (S) on the 8AV is read as an analog input.

The analog data is in an 8 bit format, i.e. values from 0 to 255 are readable. The read data is stored at the destination device identified under operand D.



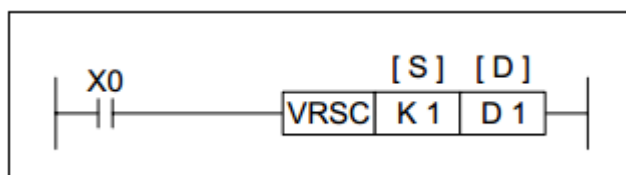
Note:

The 8AV volume ‘inputs’ are able to be read in two formats, a) as an analog value and b) as an 11 (0 to 10) position rotary switch. The second use is described in the VRSC instruction (FNC 86).

## 5.9.7 VRSD (FNC 86)

| Mnemonic                         | Function                                                                | Operands                                                                             |                                | Program steps              |
|----------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------|----------------------------|
|                                  |                                                                         | S                                                                                    | D                              |                            |
| VRSC<br>FNC 86<br>(Volume scale) | Reads the set position value, 0 to 10, from volume inputs on the FX-8AV | K, H<br>Note:<br>S= 0 to 7<br>corresponding to the 8 available volumes on the FX-8AV | KnY, KnM, KnS<br>T, C, D, V, Z | VRSC,<br>VRSCP:<br>5 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|
|------------------|------------------|---------|



Operation:

The identified volume (S) on the 8AV is read as a rotary switch

with 11 set positions (0 to 10). The position data is stored at device D as an integer

from the range 0 to 10.



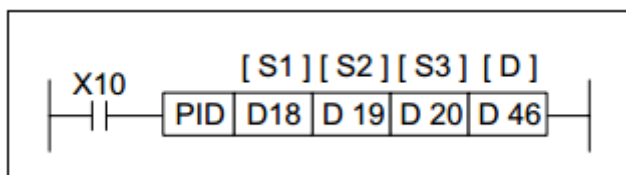
**Note:**

The 8AV volume ‘inputs’ are able to be read in two formats, a) as a 11(0 to 10) position rotary switch and b) as an analog value. The second use is described in the VRRD instruction (FNC 85).

### 5.9.8 PID (FNC 88)

| Mnemonic                                             | Function                                                                                           | Operands                                                        |    |                                                                  |                                                          | Program steps   |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|----|------------------------------------------------------------------|----------------------------------------------------------|-----------------|
|                                                      |                                                                                                    | S1                                                              | S2 | S3                                                               | D                                                        |                 |
| PID<br>FNC 88<br>(PID control loop)<br>register each | Receives a data input and calculates a corrective action to a specified level based on PID control | D $\boxtimes$<br><br>Note: S1 and S2 use a single data register |    | D $\boxtimes$<br><br>Note: S3 uses 25 consecutive data registers | D $\boxtimes$<br><br>Note: D uses a single data register | PID:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



**Operation:**

This instruction takes a current value (S2) and compares it to a

predefined set value (S1). The difference or error between the two values is then processed through a PID loop to produce a correction factor which also takes into account previous iterations and trends of the calculated error. The PID process calculates a correction factor which is applied to the current output value and stored as a corrected output value in destination device (D). The setup parameters for the PID control loop are stored in 25 consecutive data registers S3+0 through S3+24.

**Points to note:**

- a) Every PID application is different. There will be a certain amount of “trial and error” necessary to set the variables at optimal levels.
- b) On HCA2P/HCA2C MPUs a Pre-tuning feature is available that can quickly provide initial values for the PID process.
- c) The HCA1P does not have analog capabilities, it is therefore necessary to use RS232 communications to achieve basic PID operation.
- d) As 25 data register are required for the setup parameters for the PID loop, the head address of this data stack cannot be greater than D975. The contents of this data stack are explained later in this section. Multiple PID instructions can be programmed, however each PID loop must not have conflicting data registers.
- e) There are control limits in the PLC intended to help the PID controlled machines operate in a safe manner. If it becomes necessary to reset the Set Point Value (S1 ) during operation, it is recommended to turn the PID command Off and restore the command after entering the new Set Point Value. This will prevent the safety control limits from stopping the operation of the PID instruction prematurely.
- f) The PID instruction has a special set of error codes associated with it. Errors are identified in the normal manner. The error codes associated with the PID loop will be flagged by M8067 with the appropriate error code being stored in D8067. These error devices are not exclusive to the PID instruction so care should be taken to investigate errors properly. Please see chapter 6, ‘Diagnostic Devices’ for more information.

g) A full PID iteration does not have to be performed. By manipulation of the setup parameters P (proportional), I (Integral) or D (derivative) loops may be accessed individually or in a user defined/selected group. This is detailed later in this section.

## PID Equations

### Forward

$$PV_{nf} > SV$$

$$\Delta MV = K_P \left\{ (EV_n - EV_{(n-1)}) + \frac{T_S}{T_I} EV_n + D_n \right\}$$

$$EV_n = PV_{nf} - SV$$

$$D_n = \frac{T_D}{T_S + K_D \cdot T_D} (-2PV_{nf-1} + PV_{nf} + PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$$



$$MV_n = \sum \Delta MV$$

$$SV > PV_{nf}$$

$$\Delta MV = K_P \left\{ (EV_n - EV_{n-1}) + \frac{T_S}{T_I} EV_n + D_n \right\}$$

$$EV_n = SV - PV_{nf}$$

$$D_n = \frac{T_D}{T_S + K_D \cdot T_D} (2PV_{nf-1} - PV_{nf} - PV_{nf-2}) + \frac{K_D \cdot T_D}{T_S + K_D \cdot T_D} \cdot D_{n-1}$$

$$MV_n = \sum \Delta MV \Delta$$

$$PV_{nf} = PV_n + \alpha(PV_{nf-1} - PV_n)$$

$EV_n$  = the current Error Value

$EV_{n-1}$  = the previous Error Value

$SV$  = the Set Point Value ( $S_1$ )

$PV_n$  = the current Process Value ( $S_2$ )

$PV_{nf}$  = the calculated Process Value

$PV_{nf-1}$  = the previous Process Value

$PV_{nf-2}$  = the second previous Process Value

$\Delta MV$  = the change in the Output

Manipulation Values

$MV_n$  = the current Output Manipulation Value (D)

$D_n$  = the Derivative Value

$D_{n-1}$  = the previous Derivative Value

$K_P$  = the Proportion Constant

$\alpha$  = the Input Filter

$T_S$  = the Sampling Time

$T_I$  = the Integral Time Constant

$T_D$  = the Time Derivative Constant

$K_D$  = the Derivative Filter Constant

Please see the Parameter setup section for a more detailed description of the variable parameters and in which memory register they must be set.

Forward and Reverse operation (S3 +1, b0)

The Forward operation is the condition where the Process Value,  $PV_{nf}$ , is greater than the Set Point,  $SV$ . An example is a building that requires air conditioning. Without air conditioning, the temperature of the room will be higher than the Set Point so work is required to lower  $PV_{nf}$ . The Reverse operation is the condition where the Set Point is higher than the Process Value.

An example of this is an oven. The temperature of the oven will be too low unless some work is done to raise it, i.e. - the heating element is turned On.

The assumption is made with PID control that some work will need to be performed to bring the system into balance. Therefore,  $\Delta MV$  will always have a value. Ideally, a system that is stable will require a constant amount of work to keep the Set Point and Process Value equal.

PID setup parameters; S3

The PID setup parameters are contained in a 25 register data stack. Some of these devices require data input from the user, some are reserved for the internal operation and some return output data from the PID operation. Parameters S3+0 through S3 +6 must be set by the user.

| Parameter<br>S3 +P | Parameter<br>name/function                                 | Description                                                                                              |                                                   | Setting range      |
|--------------------|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|---------------------------------------------------|--------------------|
| S3+0               | Sampling time<br>TS                                        | The time interval set between the reading the current Process Value of the system (PVnf )                |                                                   | 1 to 32767<br>msec |
| S3+1               | Action -<br>reaction<br><br>direction and<br>alarm control | b0                                                                                                       | Forward operation(0), Reverse operation (1)       | Not<br>applicable  |
|                    |                                                            | b1                                                                                                       | Process Value (PVnf ) alarm enable, OFF(0)/ ON(1) |                    |
|                    |                                                            | b2                                                                                                       | Output Value (MV) alarm enable, OFF(0)/ON(1)      |                    |
|                    |                                                            | b3 -<br>15                                                                                               | Reserved                                          |                    |
| S3+2               | Input filter $\alpha$                                      | Alters the effect of the input filter.                                                                   |                                                   | 0 to 99%           |
| S3+3               | Proportional<br>gain KP                                    | This is a factor used to align the proportional output in a known magnitude to the change in the Process |                                                   | 1to<br>32767%      |

|               |                                              |                                                                                                                                                                                                                                                                 |                                                                                                                                           |                                |
|---------------|----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
|               |                                              | Value (PV nf). This is the P part of the PID loop                                                                                                                                                                                                               |                                                                                                                                           |                                |
| S3+4          | Integral time constant TI                    | This is the I part of the PID loop.<br><br>This is the time taken for the corrective integral value to reach a magnitude equal to that applied by the proportional or P part of the loop. Selecting 0 (zero) for this parameter disables the I effect.          |                                                                                                                                           | (0 to 32767)<br><br>x 100 msec |
| S3+5          | Derivative gain KD                           | This is the D part of the PID loop.<br><br>This is the time taken for the corrective derivative value to reach a magnitude equal to that applied by the proportional or P part of the loop.<br><br>Selecting 0 (zero) for this parameter disables the D effect. |                                                                                                                                           | (0 to 32767)<br><br>x10msec    |
| S3+7 to S3+19 | Reserved for use for the internal processing |                                                                                                                                                                                                                                                                 |                                                                                                                                           |                                |
| S3+20         | Process Value, maximum positive change       | Active when S3+1, b1 is set ON.                                                                                                                                                                                                                                 | This is a user defined maximum limit for the Process Value (PVnf). If the Process Value (PVnf) exceeds the limit, S3+24, bit b0 is set On | 0 to 32767                     |

|       |                                                |                                             |                                                                                                                                                                                         |                |
|-------|------------------------------------------------|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| S3+21 | Process Value,<br>minimum value                |                                             | This is a user defined lower limit for the Process Value. If the Process Value (PVnf) falls below the limit, S3+24, bit b1 is set On                                                    |                |
| S3+22 | Output Value,<br>maximum<br>positive change    | Active<br>when<br>S3+1, b2<br>is set<br>ON. | This is a user defined maximum limit for the quantity of positive change which can occur in one PID scan. If the Output Value (MV) exceeds this, S3+24, bit b2 is set On                |                |
| S3+23 | Output Value,<br>maximum<br>negative<br>change |                                             | This is a user defined maximum limit for the quantity of negative change which can occur in one PID scan. If the Output Value (MV) falls below the lower limit, S3+24, bit b3 is set On |                |
| S3+24 | Alarm flags<br>(Read Only)                     | b0                                          | High limit exceeded in Process Value (PVnf)                                                                                                                                             | Not applicable |

|  |  |         |                                                |  |
|--|--|---------|------------------------------------------------|--|
|  |  | b1      | Below low limit for the Process Value (PVnf)   |  |
|  |  | b2      | Excessive positive change in Output Value (MV) |  |
|  |  | b3      | Excessive negative change in Output Value (MV) |  |
|  |  | b4 - 15 | Reserved                                       |  |

## Configuring the PID loop

The PID loop can be configured to offer variations on PID control. These are as follows:

| Control method | Selection via setup registers        |                                      |                                      | Description                        |
|----------------|--------------------------------------|--------------------------------------|--------------------------------------|------------------------------------|
|                | S <sub>3</sub> + 3 (K <sub>P</sub> ) | S <sub>3</sub> + 4 (T <sub>I</sub> ) | S <sub>3</sub> + 6 (T <sub>D</sub> ) |                                    |
| P              | User value                           | Set to 0 (zero)                      | Set to 0 (zero)                      | Proportional effect only           |
| PI             | User value                           | User value                           | Set to 0 (zero)                      | Proportional and integral effect   |
| PD             | User value                           | Set to 0 (zero)                      | User value                           | Proportional and derivative effect |
| PID            | User value                           | User value                           | User value                           | Full PID                           |

It should be noted that in all situations there must be a proportional or ‘P’ element to the loop.

### P - proportional change

When a proportional factor is applied, it calculates the difference between the Current Error Value, EV<sub>n</sub>, and the Previous Error Value, EV<sub>n-1</sub>. The Proportional Change is based upon how fast the Process Value is moving closer to (or further away from) the

Set Point Value NOT upon the actual difference between the PVnf and SV.

Note: Other PID systems might operate using an equation that calculates the

Proportional change based upon the size of the Current Error Value only.

### **I - integral change**

Once a proportional change has been applied to an error situation, ‘fine tuning’ the correction can be performed with the I or integral element. Initially only a small

change is applied but as time increases and the error is not corrected the integral

effect is increased. It is important to note how T I actually effects how fast the total

integral correction is applied. The smaller T I is, the bigger effect the integral will have.

Note: The TI value is set in data register S3+4. Setting zero for this variable disables the Integral effect.

### **The Derivative Change**

The derivative function supplements the effects caused by the proportional response.

The derivative effect is the result of a calculation involving elements TD,TS, and the

calculated error. This causes the derivative to initially output a large corrective action

which dissipates rapidly over time. The speed of this dissipation can be controlled by

the value TD: If the value of TD is small then the effect of applying derivative control is increased.

Because the initial effect of the derivative can be quite severe there is a ‘softening’

effect which can be applied through the use of KD, the derivative gain. The action of

KD could be considered as a filter allowing the derivative response to be scaled

between 0 and 100%.

The phenomenon of chasing, or overcorrecting both too high and too low, is most often associated with the Derivative portion of the equation because of the large initial correction factor.

Note: The TD value is set in Data register S3+6. Setting zero for this variable disables the Derivative effect.

### **Effective use of the input filter $\alpha$ S3+2**

To prevent the PID instruction from reacting immediately and wildly to any errors on the Current Value, there is a filtering mechanism which allows the PID instruction to observe and account for any significant fluctuations over three samples.

The quantitative effect of the input filter is to calculate a filtered Input Value to the PID instruction taken from a defined percentage of the Current Value and the previous two filtered Input Values.

This type of filtering is often called first-order lag filter. It is particularly useful for removing the effects of high frequency noise which may appear on input signals received from sensors. The greater the filter percentage is set the longer the lag time. When the input filter is set to zero, this effectively removes all filtering and allows the Current Value to be used directly as the Input Value.

### **Initial values for PID loops**

The PID instruction has many parameters which can be set and configured to the

user's needs. The difficulty is to find a good point from which to start the fine tuning of the PID loop to the system requirements. The following suggestions will not be ideal for all situations and applications but will at least give users of the PID instruction a reasonable points from which to start.

A value should be given to all the variables listed below before turning the PID instruction ON. Values should be chosen so that the Output Manipulated Value does not exceed  $\pm 32767$ .

**Recommended initial settings:**

**TS=** Should be equal to the total program scan time or a multiple of that scan time, i.e. 2 times, 5times,etc.

**$\alpha$ =50%**

**KP=** This should be adjusted to a value dependent upon the maximum corrective action to reach the set point - values should be experimented with from an arbitrary 75%

**TI=** This should ideally be 4 to 10 times greater than the TD time

**KD= 50%**

**TD=** This is set dependent upon the total system response, i.e. not only how fast the programmable controller reacts but also any valves, pumps or motors.

For a fast system reaction TD will be set to a quick or small time, this should however never be less than TS. A slower reacting system will require the TD duration to be longer. A beginning value can be TD twice the value of TS.

Care should be taken when adjusting PID variables to ensure the safety of the operator and avoid damage to the equipment.

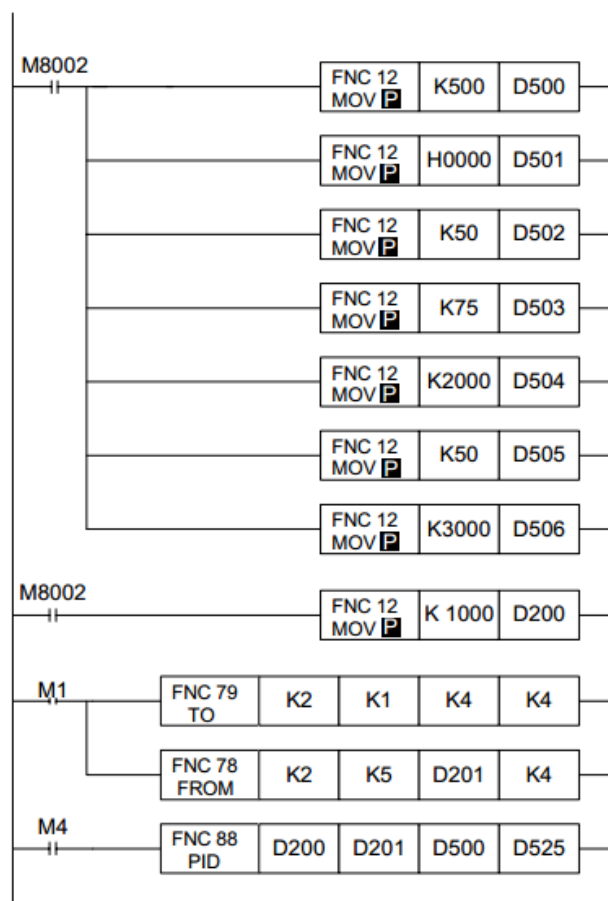
With ALL PID values there is a degree of experimentation required to tune the PID loop to the exact local conditions. A sensible approach to this is to adjust one parameter at a time by fixed percentages, i.e. say increasing (or decreasing) the KP value in steps of 10%. Selecting PID parameters without due consideration will result in a badly configured system which does not perform as required and will cause the user to become frustrated. Please remember the PID process is a purely mathematical calculation and as such has no regard for the ‘quality’ of the variable data supplied by the user/system - the PID will always process its PID mathematical function with the data available.

#### Example PID Settings

From the PID instruction at the bottom of the ladder, S1= D200; S2= D201; S3= D500; and D or MV = D525.

D500: Ts = 500 ms

D501: Forward Operation,



Alarms Not Enabled

D502: Input Filter = 50%

D503: KP=75%

D504: TI =4000ms

D505: KD= 50%

D506: TD = 1000 ms

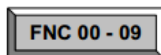
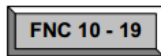
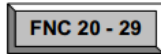
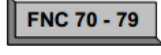
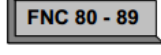
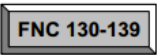
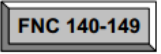
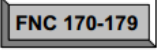
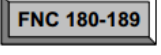
D200: Set Point = 1000

D201:  $PV_{nf}$  (an analog input value)

Begin the PID instruction

D525: PID Output Value

Applied Instructions:

|                                                                                    |                                                                                   |                                                                                     |                                 |       |
|------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------|-------|
| 1.                                                                                 |  | Program Flow                                                                        | 5-4                             |       |
| 2.                                                                                 |  | Move And Compare                                                                    | 5-16                            |       |
| 3.                                                                                 |  | Arithmetic And Logical Operations (+, -, ×, ÷)                                      | 5-24                            |       |
| 4.                                                                                 |  | Rotation And Shift                                                                  | 5-34                            |       |
| 5.                                                                                 |  | Data Operation                                                                      | 5-42                            |       |
| 6.                                                                                 |  | High Speed Processing                                                               | 5-52                            |       |
| 7.                                                                                 |  | Handy Instructions                                                                  | 5-66                            |       |
| 8.                                                                                 |  | External FX I/O Devices                                                             | 5-80                            |       |
| 9.                                                                                 |  | External FX Serial Devices                                                          | 5-94                            |       |
|  | 10.                                                                               |   | Floating Point 1 & 2            | 5-110 |
|                                                                                    | 11.                                                                               |  | Trigonometry (Floating Point 3) | 5-118 |
|                                                                                    | 12.                                                                               |  | Data Operations 2               | 5-122 |
|                                                                                    | 13.                                                                               |  | Positioning Control             | 5-126 |
|                                                                                    | 14.                                                                               |  | Real Time Clock Control         | 5-136 |
|                                                                                    | 15.                                                                               |  | Gray Codes                      | 5-146 |
|                                                                                    | 16.                                                                               |  | Additional Functions            | 5-146 |
|                                                                                    | 17.                                                                               |  | In-line Comparisons             | 5-150 |



### Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D1,S3or for lists/abled devices D3+0,S+9etc.

MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e.

positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D ☆☆☆ P - A 32 bit mode instruction modified to use pulse (single) operation.



- A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

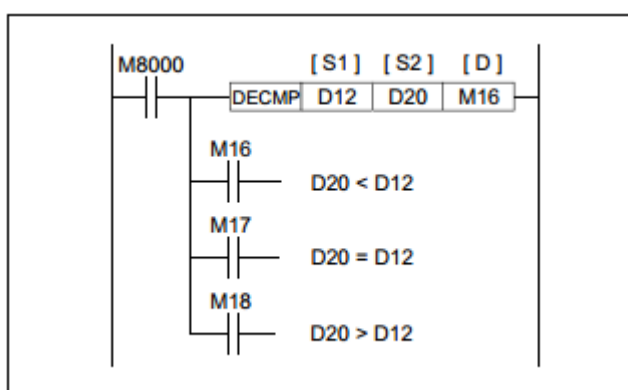


- An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

## 5.10.1 ECMP (FNC 110)

| Mnemonic                                    | Function                                                             | Operands                                                                                                             |    |                                                         | Program steps                 |
|---------------------------------------------|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----|---------------------------------------------------------|-------------------------------|
|                                             |                                                                      | S1                                                                                                                   | S2 | D                                                       |                               |
| ECMP<br>FNC 110<br>(Floating Point Compare) | Compares two floating point values - results of <, = and > are given | K, H - integer value automatically converted to floating point<br><br>D - must be in floating point format (32bits). |    | Y, M, S<br><br>Note:<br>3 consecutive devices are used. | DECMP,<br>DECMPP:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



### Operation:

The data of S1 is compared to the data of S2. The result is indicated by 3 bit devices specified with the

head address entered as D. The bit devices indicate:

S2 is less than < S1- bit device D is ON

S2 is equal to = S1-bit device D+1 is ON

S2 is greater than > S1- bit device D+2 is ON



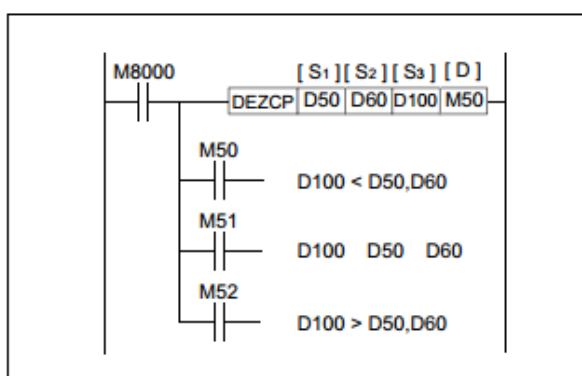
### Points to note:

The status of the destination devices will be kept even if the ECMP instruction is deactivated. Full algebraic comparisons are used: i.e.  $-1.79 \times 10^{27}$  is smaller than  $9.43 \times 10^{-15}$ .

### 5.10.2 EZCP (FNC 111)

| Mnemonic                                         | Function                                                                    | Operands                                                                                                                                                      |    |    |                                                      | Program steps           |
|--------------------------------------------------|-----------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----|------------------------------------------------------|-------------------------|
|                                                  |                                                                             | S1                                                                                                                                                            | S2 | S3 | D                                                    |                         |
| EZCP<br>FNC 111<br>(Floating Point Zone Compare) | Compares a float range with a float value - results of <, = and > are given | K, H - integer value automatically converted to floating point<br><br>D - must be in floating point format (32 bits).<br><b>Note:</b> S1 must be less than S2 |    |    | Y, M, S<br><br>Note: 3 consecutive devices are used. | DEZCP, DEZCPP: 13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The operation is the same as the

ECMP instruction except that a

single data value (S3) is compared to

a data range (S1-S2).

S3 is less than S1 and S2 - bit device D is ON

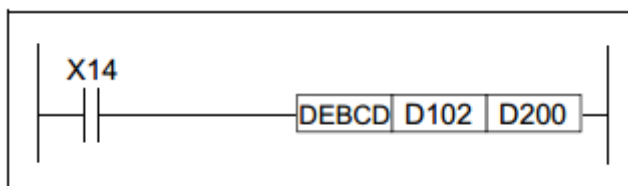
S3 is between S1 and S2 - bit device D+1 is ON

S3 is greater than S2 - bit device D+2 is ON

### 5.10.3 EBCD (FNC 118)

| Mnemonic                                            | Function                                                          | Operands                                        |                                                                           | Program steps          |
|-----------------------------------------------------|-------------------------------------------------------------------|-------------------------------------------------|---------------------------------------------------------------------------|------------------------|
|                                                     |                                                                   | S                                               | D                                                                         |                        |
| EBCD<br>FNC 118<br>(Float to Scientific conversion) | Converts floating point number format to scientific number format | D - must be in floating point format (32 bits). | D - 2 consecutive devices are used<br><br>D - mantissa<br>D+1 - exponent. | DEBCD, DEBCDP: 9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



**Operation:**

Converts a floating point value at S into separate mantissa and

exponent parts at D and D+1 (scientific format).

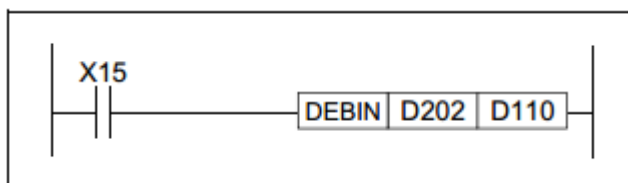
**Points to note:**

- a) The instruction must be double word format. The destinations D and D+1 represent the mantissa and exponent of the floating point number respectively.
- b) To provide maximum accuracy in the conversion the mantissa D will be in the range 1000 to 9999 (or 0) and the exponent D+1 corrected to an appropriate value.
- c) E.g.  $S = 3.4567 \times 10^{-5}$  will become  $D = 3456$ ,  $D+1 = -8$ .

**5.10.4 EBIN (FNC 119)**

| Mnemonic                                            | Function                                                          | Operands                                                                 |                                       | Program steps          |
|-----------------------------------------------------|-------------------------------------------------------------------|--------------------------------------------------------------------------|---------------------------------------|------------------------|
|                                                     |                                                                   | S                                                                        | D                                     |                        |
| EBIN<br>FNC 119<br>(Scientific to Float conversion) | Converts scientific number format to floating point number format | D - 2 consecutive devices are used<br><br>S- mantissa<br>S+1 - exponent. | D - a floating point value (32 bits). | DEBIN, DEBINP: 9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



**Operation:**

Generates a floating point

number at D from scientific format data at source S.

**Points to note:**

- a) The instruction must be double word format. The source data S and S+1 represent the mantissa and exponent of the floating point number to be generated.
- b) To provide maximum accuracy in the conversion the mantissa S must be in the range 1000 to 9999 (or 0) and the exponent S+1 corrected to an appropriate value.
- c) E.g. S= 5432, S+1= 12 will become D=  $5.432 \times 10^{-9}$

**5.10.5 EADD (FNC 120)**

| Mnemonic                                     | Function                                 | Operands                                                                                                              |    |                                       | Program steps           |
|----------------------------------------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|----|---------------------------------------|-------------------------|
|                                              |                                          | S1                                                                                                                    | S2 | D                                     |                         |
| EADD<br>FNC 120<br>(Floating Point Addition) | Adds two floating point numbers together | K, H - integer value automatically converted to floating point<br><br>D - must be in floating point format (32 bits). |    | D - a floating point value (32 bits). | DEADD, DEADDP: 13 steps |



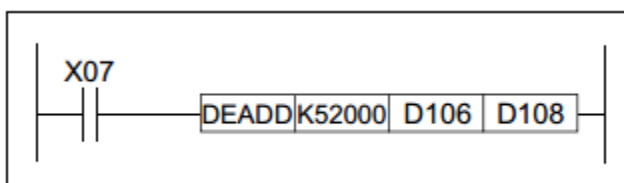
16 BIT OPERATION

32 BIT OPERATION

PULSE-P

Flags

Zero M8020  
Borrow M8021  
Carry M8022



**Operation:**

The floating point values stored

in the source devices S1 and S2 are algebraically added and the result stored in the destination device D.

**Points to note:**

- a) The instruction must use the double word format; i.e., DEADD or DEADDP. All source

data and destination data will be double word; i.e. uses two consecutive data registers to store the data (32 bits).

Except for K or H, all source data will be regarded as being in floating point format and the result stored in the destination will also be in floating point format.

b) If a constant K or H is used as source data, the value is converted to floating point before the addition operation.

c) The addition is mathematically correct: i.e.,  $2.3456 \times 10^2 + (-5.6 \times 10^{-1}) = 2.34 \times 10^2$

d) The same device may be used as a source and as the destination. If this is the case then, on continuous operation of the DEADD instruction, the result of the previous operation will be used as a new source value and a new result calculated.

This will happen every program scan unless the pulse modifier or an interlock program is used.

e) If the result of the calculation is zero “0” then the zero flag, M8020 is set ON.

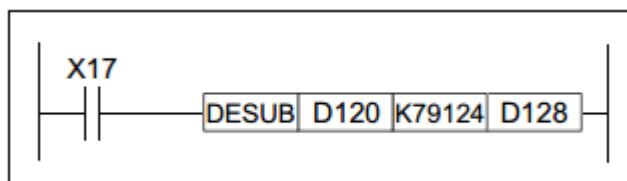
If the result of the calculation is larger than the largest floating point number then the carry flag, M8021 is set ON and the result is set to the largest value.

If the result of the calculation is smaller than the smallest floating point number then the borrow flag, M8022 is set ON and the result is set to the smallest value.

#### 5.10.6 EAUB (FNC 121)

| Mnemonic                                         | Function                                         | Operands                                                                                                                 |    |                                       | Program steps                 |
|--------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|----|---------------------------------------|-------------------------------|
|                                                  |                                                  | S1                                                                                                                       | S2 | D                                     |                               |
| ESUB<br>FNC 121<br>(Floating Point Sub-traction) | Subtracts one floating point number from another | K, H - integer value automatically converted to floating point<br>D - must be in floating point number format (32 bits). |    | D - a floating point value (32 bits). | DESUB,<br>DESUBP:<br>13 steps |

|                  |                  |         |       |                                           |
|------------------|------------------|---------|-------|-------------------------------------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Zero M8020<br>Borrow M8021<br>Carry M8022 |
|------------------|------------------|---------|-------|-------------------------------------------|



Operation:

The floating point value of S2 is subtracted from the floating

point value of S1 and the result stored in destination device D.

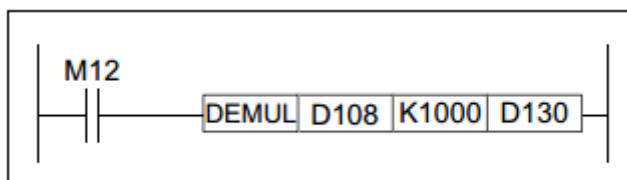
Points to note:

All points of the EADD instruction apply, except that a subtraction is performed.

#### 5.10.7 EMUL (FNC 122)

| Mnemonic                                           | Function                                       | Operands                                                                                                              |    |                                       | Program steps              |
|----------------------------------------------------|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|----|---------------------------------------|----------------------------|
|                                                    |                                                | S1                                                                                                                    | S2 | D                                     |                            |
| EMUL<br>FNC 122<br>(Floating Point Multiplication) | Multiplies two floating point numbers together | K, H - integer value automatically converted to floating point<br><br>D - must be in floating point format (32 bits). |    | D - a floating point value (32 bits). | DEMUL, DEMULP:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

The floating point value of S1 is multiplied with the floating

point value of S2. The result of the multiplication is stored at D as a floating point value.

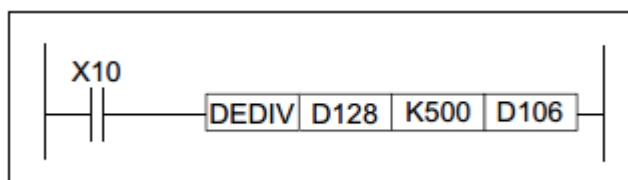
### Points to note:

Point a, b, c and d of the EADD instruction apply, except that a multiplication is performed.

### 5.10.8 EDIV (FNC 123)

| Mnemonic                                     | Function                                      | Operands                                                                                                              |    |                                       | Program steps                 |
|----------------------------------------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|----|---------------------------------------|-------------------------------|
|                                              |                                               | S1                                                                                                                    | S2 | D                                     |                               |
| EDIV<br>FNC 123<br>(Floating Point Division) | Divides one floating point number by another. | K, H - integer value automatically converted to floating point<br><br>D - must be in floating point format (32 bits). |    | D - a floating point value (32 bits). | DEDIV,<br>DEDIVP:<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



### Operation:

The floating point value of S1 is divided by the floating point

value of S2. The result of the division is stored in D as a floating point value. No remainder is calculated.

### Points to note:

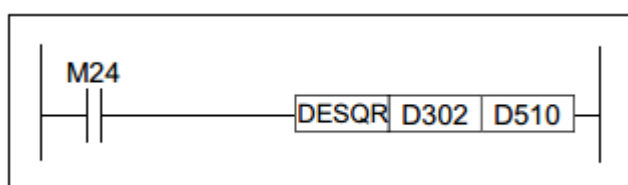
Points a, b, c, d of the EADD instruction apply, except that a division is performed.

- If S2 is 0 (zero) then a divide by zero error occurs and the operation fails.

### 5.10.9 ESQR (FNC 127)

| Mnemonic                                        | Function                                              | Operands                                                                                                                     |                                       | Program steps                |
|-------------------------------------------------|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|------------------------------|
|                                                 |                                                       | S                                                                                                                            | D                                     |                              |
| ESQR<br>FNC 127<br>(Floating Point Square Root) | Calculates the square root of a floating point value. | K, H - integer value automatically converted to floating point<br><br>D - must be in floating point number format (32 bits). | D - a floating point value (32 bits). | DESQR,<br>DESQRP:<br>9 steps |

|                  |                  |         |       |            |
|------------------|------------------|---------|-------|------------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Zero M8020 |
|------------------|------------------|---------|-------|------------|



Operation:

A square root is performed on the floating point value of S and the

result is stored in D.

#### Points to note:

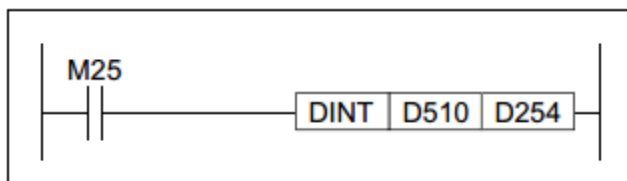
Points a, b, c, d of the EADD instruction apply, except that a square root is performed.

- If S is negative then an error occurs and error flag M8067 is set ON.

### 5.10.10 INT (FNC 129)

| Mnemonic                             | Function                                                       | Operands                                                      |                                                                                    | Program steps                                        |
|--------------------------------------|----------------------------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------|
|                                      |                                                                | S                                                             | D                                                                                  |                                                      |
| INT<br>FNC 129<br>(Float to Integer) | Converts a number from floating point format to decimal format | D - must be in floating point number format (always 32 bits). | D - decimal format<br><br>for INT, INTP - 16 bits<br><br>for DINT, DINTP - 32 bits | INT, INTP:<br>5 steps<br><br>DINT, DINTP:<br>9 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P | Flags | Zero M8020<br>Borrow M8021<br>Carry M8022 |
|------------------|------------------|---------|-------|-------------------------------------------|
|------------------|------------------|---------|-------|-------------------------------------------|



**Operation:**

The floating point value of S is rounded down to the nearest

integer value and stored in normal binary format in D.

#### Points to note:

a) The source data is always a double (32 bit) word; a floating point value.

For single word (16 bit) operation the destination is a 16 bit value.

For double word (32 bit) operation the destination is a 32 bit value.

b) This instruction is the inverse of the FLT instruction.

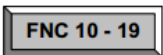
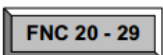
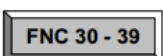
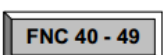
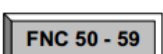
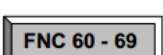
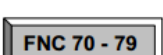
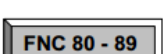
c) If the result is 0 then the zero flag M8020 is set ON.

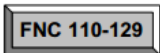
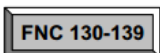
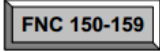
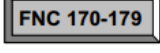
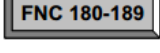
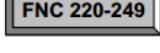
If the source data is not a whole number it must be rounded down. In this case the borrow flag M8021 is set ON to indicate a rounded value.

If the resulting integer value is outside the valid range for the destination device then an overflow occurs. In this case the carry flag M8022 is set on to indicate overflow.

Note: If overflow occurs, the value in the destination device will not be valid.

Applied Instructions:

|    |                                                                                     |                                                |      |
|----|-------------------------------------------------------------------------------------|------------------------------------------------|------|
| 1. |    | Program Flow                                   | 5-4  |
| 2. |    | Move And Compare                               | 5-16 |
| 3. |  | Arithmetic And Logical Operations (+, -, ×, ÷) | 5-24 |
| 4. |  | Rotation And Shift                             | 5-34 |
| 5. |  | Data Operation                                 | 5-42 |
| 6. |  | High Speed Processing                          | 5-52 |
| 7. |  | Handy Instructions                             | 5-66 |
| 8. |  | External FX I/O Devices                        | 5-80 |
| 9. |  | External FX Serial Devices                     | 5-94 |

|                                                                                   |     |                                                                                   |                                 |       |
|-----------------------------------------------------------------------------------|-----|-----------------------------------------------------------------------------------|---------------------------------|-------|
|                                                                                   | 10. |  | Floating Point 1 & 2            | 5-110 |
|  | 11. |  | Trigonometry (Floating Point 3) | 5-118 |
|                                                                                   | 12. |  | Data Operations 2               | 5-122 |
|                                                                                   | 13. |  | Positioning Control             | 5-126 |
|                                                                                   | 14. |  | Real Time Clock Control         | 5-136 |
|                                                                                   | 15. |  | Gray Codes                      | 5-146 |
|                                                                                   | 16. |  | Additional Functions            | 5-146 |
|                                                                                   | 17. |  | In-line Comparisons             | 5-150 |

## Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D1,S3or for lists/abled devices D3+0,S+9etc.

MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e.

positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆- An instruction operating in 16 bit mode, where☆☆☆ identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.

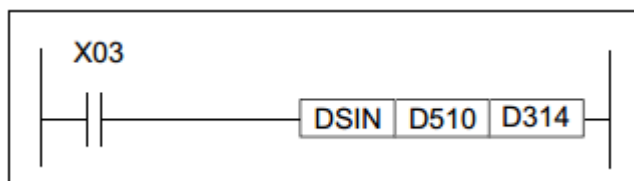
 - A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

 - An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

### 5.11.1 SIN (FNC 130)

| Mnemonic                 | Function                                      | Operands                                                        |                                       | Program steps              |
|--------------------------|-----------------------------------------------|-----------------------------------------------------------------|---------------------------------------|----------------------------|
|                          |                                               | S                                                               | D                                     |                            |
| SIN<br>FNC 130<br>(Sine) | Calculates the sine of a floating point value | D - must be in floating point number format (32 bits).(radians) | D - a floating point value (32 bits). | DSIN,<br>DSINP:<br>9 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|
|------------------|------------------|---------|



Contents:

This instruction performs the mathematical SIN operation on

the floating point value in S. The result is stored in D.

### Points to note:

a) The instruction must use the double word format: i.e., DSIN or DSINP. All source and destination data will be double word; i.e., uses two consecutive data registers to store the data (32 bits).

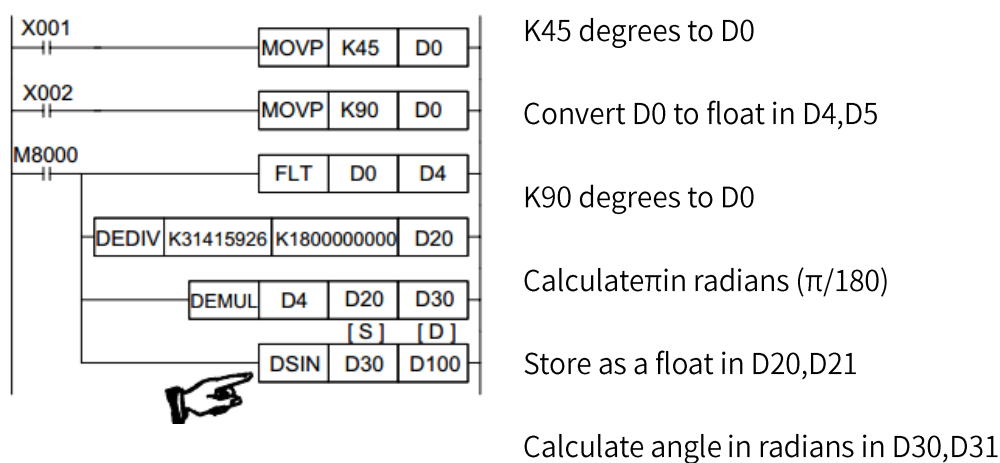
The source data is regarded as being in floating point format and the destination is also in floating point format.

b) The source value must be an angle between 0 to 360 degrees in radians; i.e.,

$$0^{\circ} \leq S < 360^{\circ}$$

Radian Angles

Below is an program example of how to calculate angles in radians using floating point.



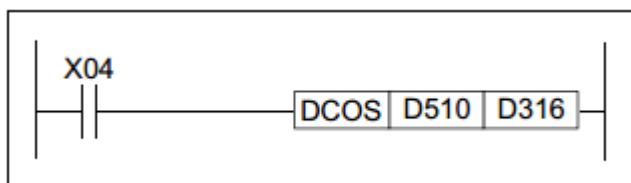
$$(\text{deg}^{\circ} \times \pi/180 = \text{rads})$$

Calculate SIN of angle in D100

### 5.11.2 COS (FNC 131)

| Mnemonic                   | Function                                        | Operands                                               |                                       | Program steps              |
|----------------------------|-------------------------------------------------|--------------------------------------------------------|---------------------------------------|----------------------------|
|                            |                                                 | S                                                      | D                                     |                            |
| COS<br>FNC 131<br>(Cosine) | Calculates the cosine of a floating point value | D - must be in floating point number format (32 bits). | D - a floating point value (32 bits). | DCOS,<br>DCOSP:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Contents:

This instruction performs the mathematical COS operation on

the floating point value in S. The result is stored in D.

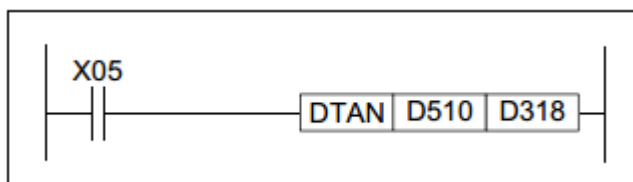
Points to note:

All the points for the SIN instruction apply, except that COS is calculated.

### 5.11.3 TAN (FNC 132)

| Mnemonic                   | Function                                         | Operands                                               |                                       | Program steps              |
|----------------------------|--------------------------------------------------|--------------------------------------------------------|---------------------------------------|----------------------------|
|                            |                                                  | S                                                      | D                                     |                            |
| TAN<br>FNC132<br>(Tangent) | Calculates the tangent of a floating point value | D - must be in floating point number format (32 bits). | D - a floating point value (32 bits). | DTAN,<br>DTANP:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



**Contents:**

This instruction performs the mathematical TAN operation on

the floating point value in S. The result is stored in D.

#### Points to note:

All the points for the SIN instruction apply, except that COS is calculated.

Applied Instructions:

|                                                                                         |                    |                                                |       |
|-----------------------------------------------------------------------------------------|--------------------|------------------------------------------------|-------|
| 1.                                                                                      | <b>FNC 00 - 09</b> | Program Flow                                   | 5-4   |
| 2.                                                                                      | <b>FNC 10 - 19</b> | Move And Compare                               | 5-16  |
| 3.                                                                                      | <b>FNC 20 - 29</b> | Arithmetic And Logical Operations (+, -, ×, ÷) | 5-24  |
| 4.                                                                                      | <b>FNC 30 - 39</b> | Rotation And Shift                             | 5-34  |
| 5.                                                                                      | <b>FNC 40 - 49</b> | Data Operation                                 | 5-42  |
| 6.                                                                                      | <b>FNC 50 - 59</b> | High Speed Processing                          | 5-52  |
| 7.                                                                                      | <b>FNC 60 - 69</b> | Handy Instructions                             | 5-66  |
| 8.                                                                                      | <b>FNC 70 - 79</b> | External FX I/O Devices                        | 5-80  |
| 9.                                                                                      | <b>FNC 80 - 89</b> | External FX Serial Devices                     | 5-94  |
| 10.                                                                                     | <b>FNC 110-129</b> | Floating Point 1 & 2                           | 5-110 |
| 11.                                                                                     | <b>FNC 130-139</b> | Trigonometry (Floating Point 3)                | 5-118 |
|  12. | <b>FNC 140-149</b> | Data Operations 2                              | 5-122 |
| 13.                                                                                     | <b>FNC 150-159</b> | Positioning Control                            | 5-126 |
| 14.                                                                                     | <b>FNC 160-169</b> | Real Time Clock Control                        | 5-136 |
| 15.                                                                                     | <b>FNC 170-179</b> | Gray Codes                                     | 5-146 |
| 16.                                                                                     | <b>FNC 180-189</b> | Additional Functions                           | 5-146 |
| 17.                                                                                     | <b>FNC 220-249</b> | In-line Comparisons                            | 5-150 |



### Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with

the same function e.g. D1,S3or for lists/abled devices D3+0,S+9etc. MSB - Most

Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e.

positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆- An instruction operating in 16 bit mode, where ☆☆☆identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆- An instruction modified to operate in 32 bit operation.

D☆☆☆ P - A 32 bit mode instruction modified to use pulse (single) operation



- A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

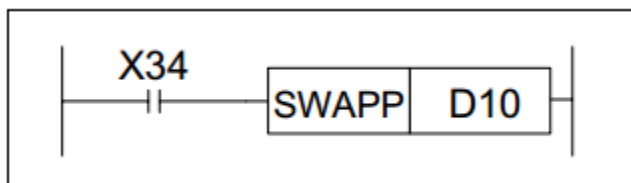


- An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

## 5.12.1 SWAP (FNC 147)

| Mnemonic                            | Function                                                      | Operands                     | Program steps                                   |
|-------------------------------------|---------------------------------------------------------------|------------------------------|-------------------------------------------------|
|                                     |                                                               | S                            |                                                 |
| SWAP<br>FNC 147<br>(Byte Swap)<br>→ | The high and low byte of the designated devices are exchanged | KnY, KnM, KnS, T, C, D, V, Z | SWAP, SWAPP : 5 steps<br>DSWAP, DSWAPP: 9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Contents:

The upper byte and the lower byte of the source device are swapped.

This instruction is equivalent to operation 2 of FNC 17 XCH

.

### Points to note:

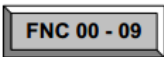
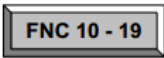
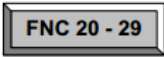
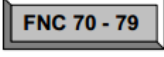
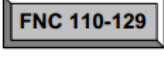
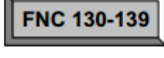
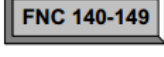
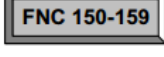
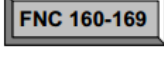
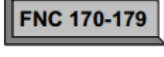
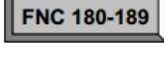
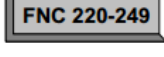
- In single word (16 bit) operation the upper and lower byte of the source device are exchanged.
- In double word (32 bit) operation the upper and lower byte of each or the two 16 bit devices are exchanged.

Result of DSWAP(P) D10:

| Values are in Hex for clarity |        | Before DSWAP | After DSWAP |
|-------------------------------|--------|--------------|-------------|
| D10                           | Byte 1 | 1FH          | 8BH         |
|                               | Byte 2 | 8BH          | 1FH         |
| D11                           | Byte 1 | C4H          | 35H         |
|                               | Byte 2 | 35H          | C4H         |

c) If the operation of this instruction is allowed to execute each scan, then the value of the source device will swap back to its original value every other scan. The use of the pulse modifier or an interlock program is recommended.

Applied Instructions:

|                                                                                     |                                                                                     |                                                |       |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------------|-------|
| 1.                                                                                  |    | Program Flow                                   | 5-4   |
| 2.                                                                                  |    | Move And Compare                               | 5-16  |
| 3.                                                                                  |    | Arithmetic And Logical Operations (+, -, ×, ÷) | 5-24  |
| 4.                                                                                  |    | Rotation And Shift                             | 5-34  |
| 5.                                                                                  |    | Data Operation                                 | 5-42  |
| 6.                                                                                  |    | High Speed Processing                          | 5-52  |
| 7.                                                                                  |    | Handy Instructions                             | 5-66  |
| 8.                                                                                  |    | External FX I/O Devices                        | 5-80  |
| 9.                                                                                  |    | External FX Serial Devices                     | 5-94  |
| 10.                                                                                 |  | Floating Point 1 & 2                           | 5-110 |
| 11.                                                                                 |  | Trigonometry (Floating Point 3)                | 5-118 |
| 12.                                                                                 |  | Data Operations 2                              | 5-122 |
|  |  | Positioning Control                            | 5-126 |
| 14.                                                                                 |  | Real Time Clock Control                        | 5-136 |
| 15.                                                                                 |  | Gray Codes                                     | 5-146 |
| 16.                                                                                 |  | Additional Functions                           | 5-146 |
| 17.                                                                                 |  | In-line Comparisons                            | 5-150 |



### Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D1,S3or for lists/abled devices D3+0,S+9etc.

MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆identifies the instruction mnemonic.

☆☆☆P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆- An instruction modified to operate in 32 bit operation.

D☆☆☆ P - A 32 bit mode instruction modified to use pulse (single) operation



- A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.



- . An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

### 5.13.1 Cautions when using Positioning Instructions

The following positioning instructions are application instructions that can be used many times in a program.

When designing a program, make sure to follow the cautions outlined below with regard to instruction drive timing.

**FNC 156 (ZRN)**

**FNC 157 (PLSV)**

**FNC 158 (DRVI)**

**FNC 159 (DRVA)**

- Do not drive positioning instructions which use the same output relay (Y000 or Y001) at the same time. If such instructions are driven at the same time, they will be treated as double coils, and not function correctly.
- Before setting a drive contact ON after it has been set to OFF be sure that the following condition is satisfied;

One or more operation cycles of the ‘pulse output monitor (Y000:M8147, Y001:M8148)’ must occur after the positioning instruction is turned OFF, before it can be used again.

This condition must be met, as one or more OFF operations are required for the re-driving of a positioning instruction.

If it is not met, and ‘operation error’ will occur during the instruction execution.

- Use the Step Ladder Program to correctly set up positioning instructions in conformance to the cautions above.

Caution when using Positioning instructions with FNC 57 (PLSY) & FNC 59 (PLSR)

- Pulse output instructions FNC 57 (PLSY) & FNC 59 (PLSR) use output points Y000 and Y001 in the same way as the positioning instructions described above.
- If a positioning and a pulse output instruction are used in the same operation, the conflicting instructions will be treated as double coils and not function correctly.
- It is recommended to use a FNC 158 (DRVI) instruction in place of either a FNC 57 (PLSY) or FNC 59 (PLSR) instruction to avoid incorrect operation when pulse outputs are required while positioning instructions are being used.

Output terminals Y000 and Y001 are high speed response type

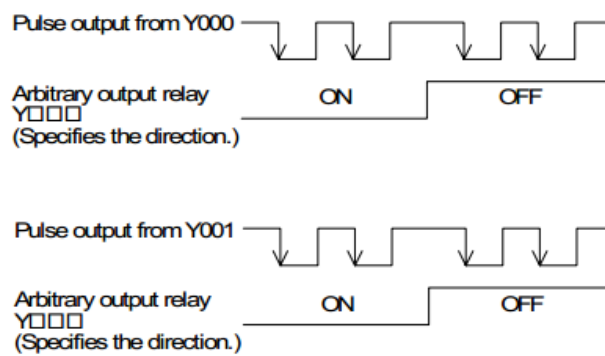
Voltage range : 5 to 24V DC

Current range : 10 to 100mA

Output frequency : 100kHz or less

### 5.13.2 Pulse train settings

When a positioning operation is executed from the PLC, the pulse output signal has the ‘Pulse train + Sign’ format during control, as shown in the figure below.



Make sure to set the pulse train input mode on the servo amplifier or stepper motor as follows;

Pulse train input mode: Pulse train + Sign

Pulse train logic: negative logic

### 5.13.3 Devices related to positioning

| Device No. |       | Data size | Initial value | Description                                                                   |
|------------|-------|-----------|---------------|-------------------------------------------------------------------------------|
| D8140      | Lower | 32 bit    | 0             | Operates as current value registers of positioning instruction output to Y000 |
| D8141      | Upper |           |               |                                                                               |

|       |       |        |   |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------|-------|--------|---|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |       |        |   | <p>For FNC 157 (PLSV), FNC 158 (DRVI), FNC 159 (DRVA) instructions, current value increases or decreases in accordance with direction of rotation.</p> <p>Although FNC 57 (PLSY) and FNC 59 (PLSR) instructions use the same current value registers, the current value represents the accumulating total number of output pulses during instruction execution.</p>                                                                                      |
| D8142 | Lower | 32 bit | 0 | <p>Operates as current value registers of positioning instruction output to Y001</p> <p>For FNC 157 (PLSV), FNC 158 (DRVI), FNC 159 (DRVA) instructions, current value increases or decreases in accordance with direction of rotation.</p> <p>Although FNC 57 (PLSY) and FNC 59 (PLSR) instructions use the same current value registers, the current value represents the accumulating total number of output pulses during instruction execution.</p> |
| D8143 | Upper |        |   |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| D8145 |       | 16 bit | 0 | <p>Bias speed when FNC 156 (ZRN), FNC 158 (DRVI), FNC 159 (DRVA) instructions are executed</p> <p>Set range: 1/10 or less of maximum speed (D8146 &amp; D8147) If the current value exceeds this range, it is automatically set to</p>                                                                                                                                                                                                                   |

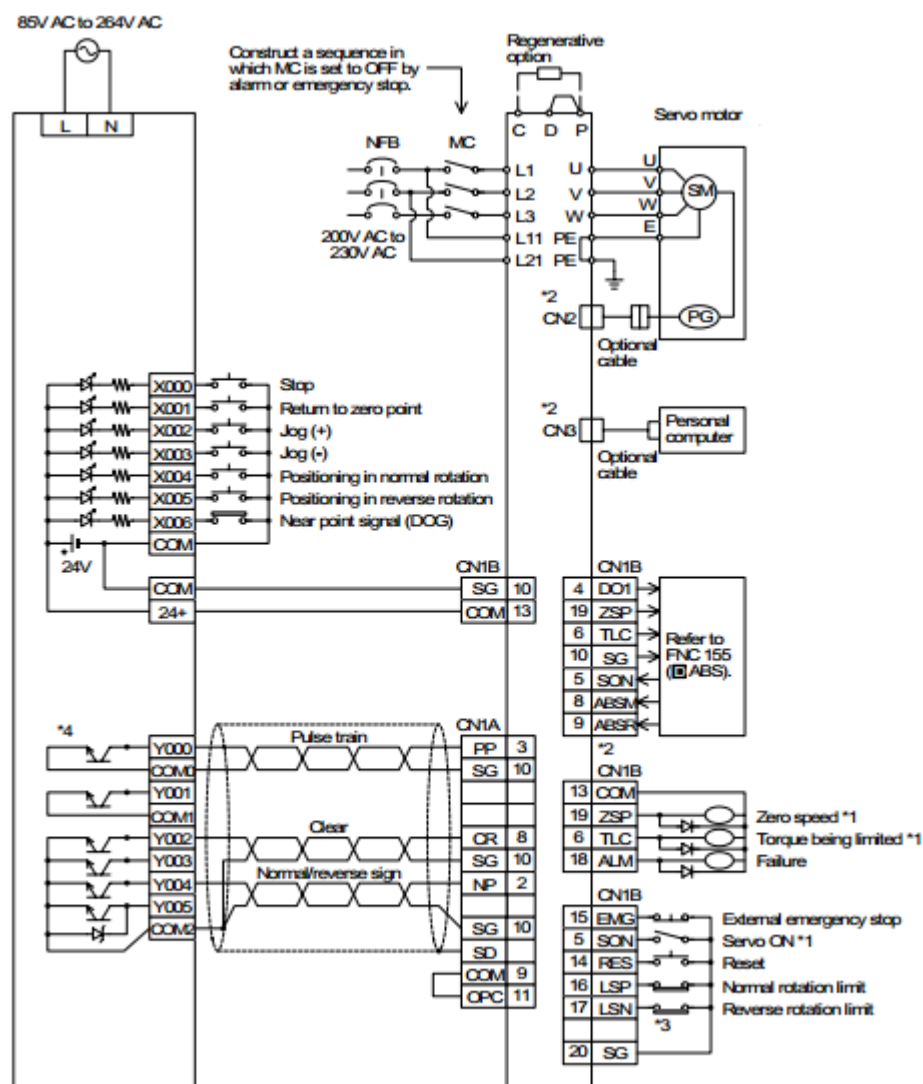
|       |       |        |         |                                                                                                                                                                                                                           |
|-------|-------|--------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |       |        |         | 1/10 of the maximum speed during operation.                                                                                                                                                                               |
| D8146 | Lower | 32 bit | 100,000 | Maximum speed when FNC 156 (ZRN), FNC 158 (DRVI), FNC 159 (DRVA) instructions are executed.<br><br>Set range 10 to 100,000 (Hz).                                                                                          |
| D8147 | Upper |        |         |                                                                                                                                                                                                                           |
| D8148 |       | 16 bit | 100     | Acceleration/Deceleration time in which maximum speed (D8146 & D8147) is achieved from bias speed (D8145) when FNC 156 (ZRN), FNC 158 (DRVI), FNC 159 (DRVA) instructions are executed.<br><br>Set range 50 to 5,000 (ms) |

| Device No. | Attribute      | Description                                     |
|------------|----------------|-------------------------------------------------|
| M8145      | Drive enable   | Y000 pulse output stop command (immediate stop) |
| M8146      | Drive enable   | Y001 pulse output stop command (immediate stop) |
| M8147      | Read only flag | Y000 pulse output monitor (BUSY/READY)          |
| M8148      | Read only flag | Y001 pulse output monitor (BUSY/READY)          |

#### 5.13.4 Servo Wiring Example

Example of connection to a HCFA servo.

**Note.** The PLC required for this connection is a SINK Transistor output type.



\*1 Connect to programmable controller when absolute position detection is required.

\*2 Ports CN1A, CN1B, CN2 & CN3 are the same shape. Do not confuse them.

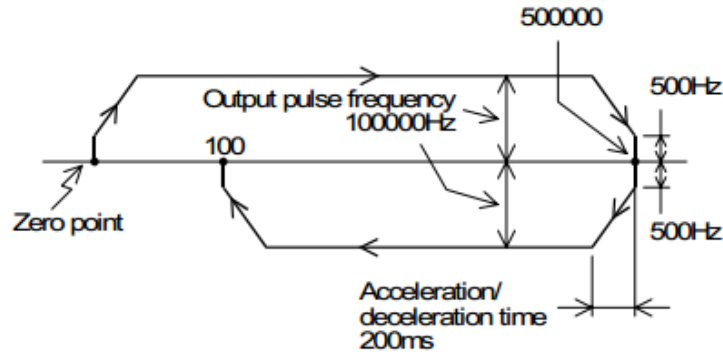
\*3 Connect a limit switch to the servo amplifier.

\*4 ONLY use a transistor output type PLC.

## 5.13.5 Example Program

The following example program for forward/reverse operation uses the I/O assignment shown in section 5.13.4 Servo Wiring Example.

During operation positioning is performed using the absolute position method shown below.



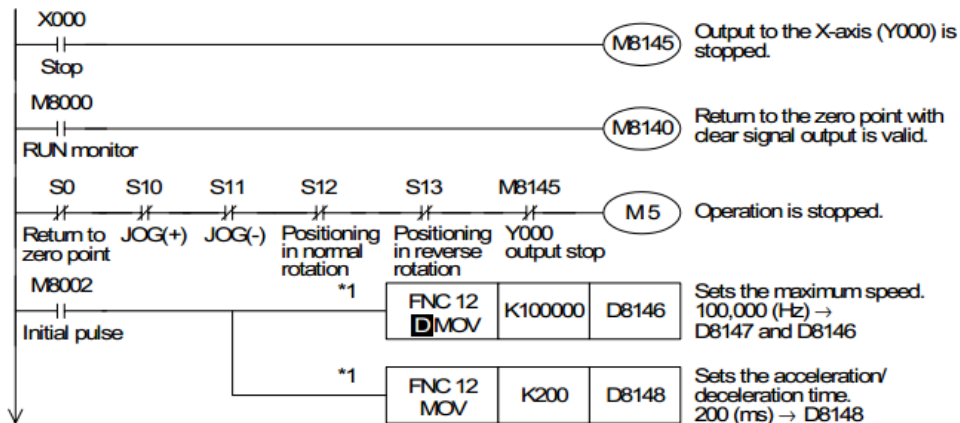
In this example the actual output frequency for the first step, acceleration, and the last step, deceleration, can be obtained using the following expression.

$$\sqrt{A \div (2 \times (C \div 1000))} = \text{Output}$$

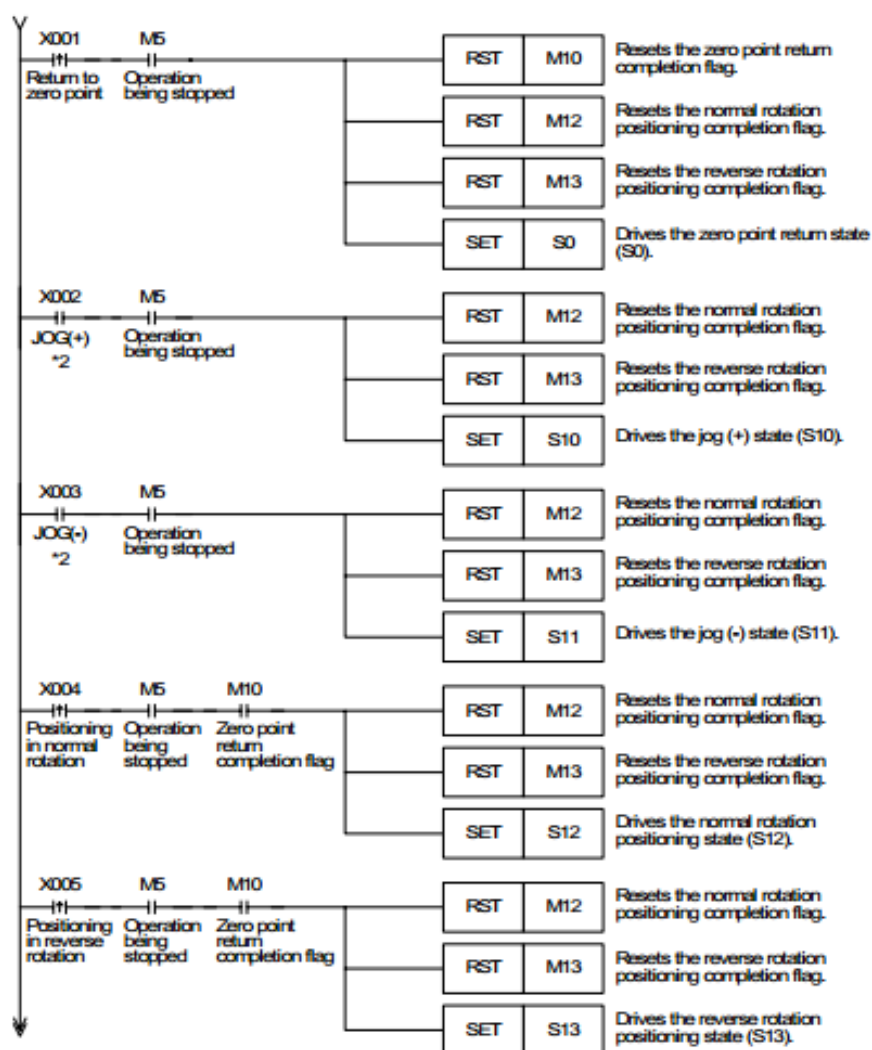
A = Maximum speed (D8146, D8147)  
B = Acceleration/Deceleration time

$$\sqrt{100000 \div (2 \times (200 \div 1000))} = 500\text{Hz}$$

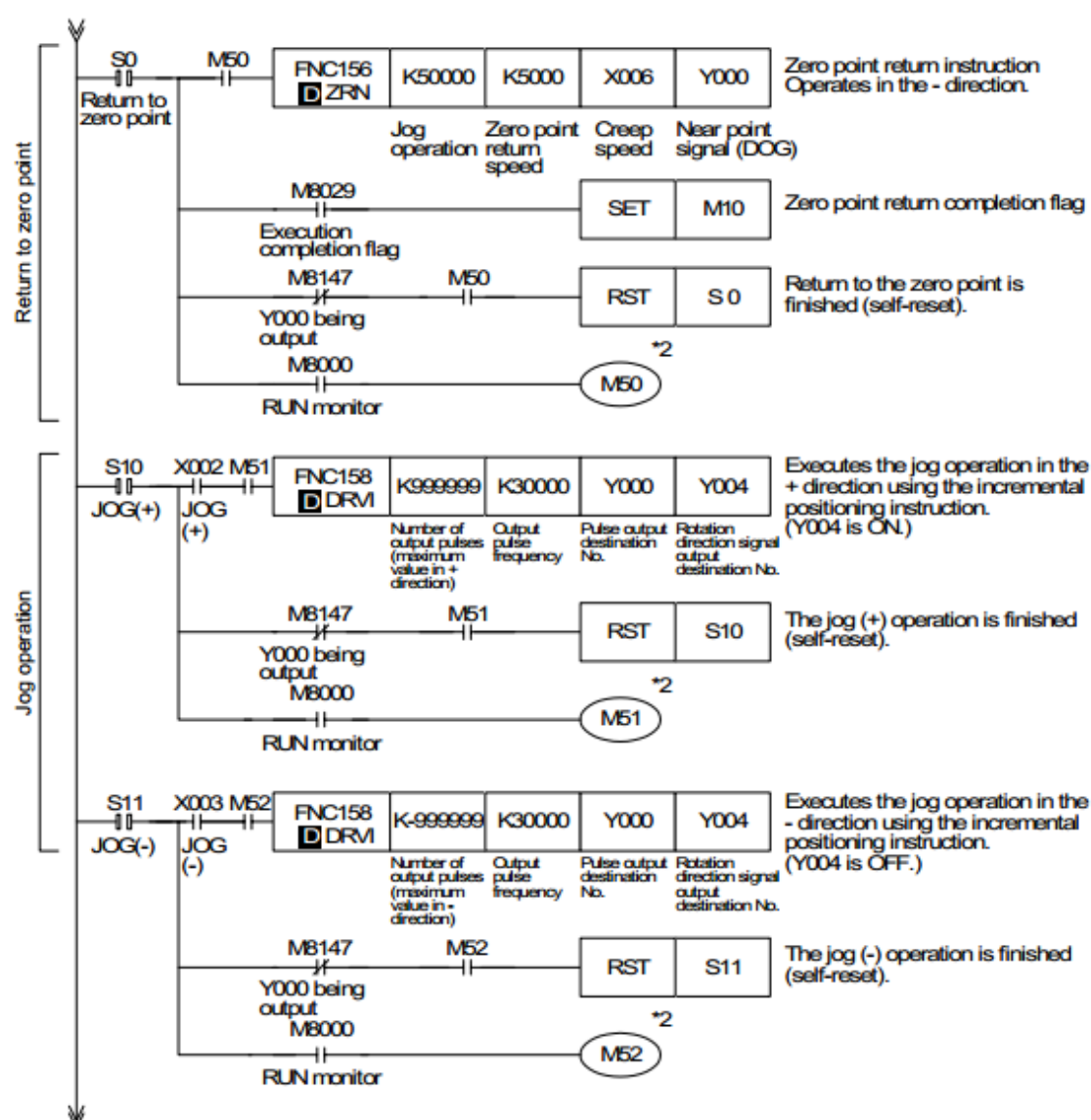
Step Ladder program.

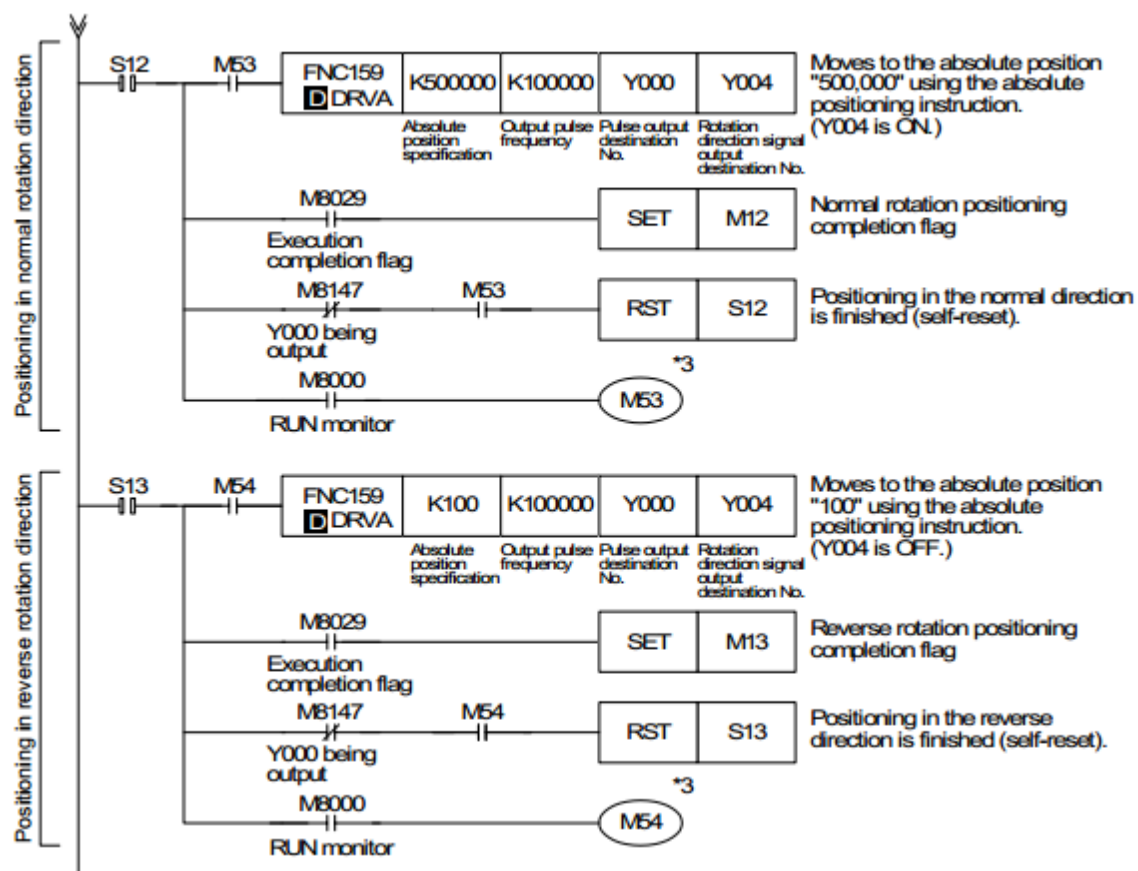


\*1 When the maximum speed or Acceleration/deceleration do not have to be changed from their initial values, programming is not required.



\*2 The maximum size of a JOG command is 999,999 pulses, as this is the maximum number of output pulses for a FNC 158 (DRVI) instruction. If a greater distance is required execute more than one JOG command.



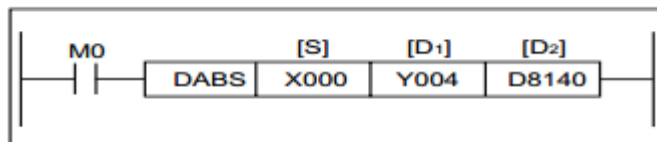


\*3 The instruction drive timing is delayed by one operation cycle to prevent simultaneous driving of positioning instructions.

## 5.13.6 ABS (FNC 155)

| Mnemonic                                      | Function                                       | Operands |                |                | Program steps    |
|-----------------------------------------------|------------------------------------------------|----------|----------------|----------------|------------------|
|                                               |                                                | S        | D <sub>1</sub> | D <sub>2</sub> |                  |
| ABS<br>FNC 155<br>Absolute current value read | Reads the absolute position from a servo motor | X,Y,M,S  | Y,M,S          | T,C,D,V,Z      | DABS<br>13 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction reads the

absolute position data when a HCFA servo motor, equipped with absolute positioning function is connected. [S] is the first of three inputs used for communication flags (see drawing below), [D1] is the first of three communication outputs and [D 2 ] is the data destination register.

**Points to note:**

- a) This instruction is 32-Bit. Be sure to input as “DABS”
- b) Read starts when the instruction drive contact turns ON. When the read is complete, the execution complete flag M8029 is energized.  
  
If the instruction drive contact is turned OFF during read, read is aborted.
- c) When designing a system, set the servo amplifier to be ON earlier than the power of the PLC, or so that they are both powered ON at the same time.
- d) The device [D2 ] to which the absolute value is read, can be set within a word device range.

However, the absolute value should be transferred at some point to the correct registers (D8141 & D8140)

- e) The DABS instruction drive contact uses an input which is always ON, even after the absolute value is read.

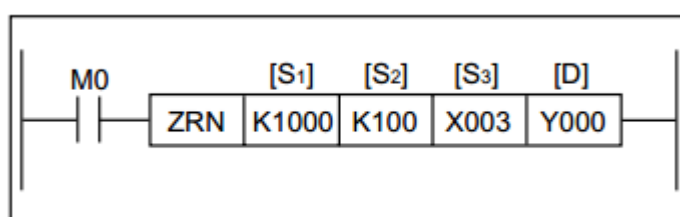
If the instruction drive contact turns OFF after the read is complete, the servo ON (SON) signal will turn OFF and the operation disabled.

- f) Even if the servo motor is equipped with an absolute position detection function, it is good practice to execute a zero return operation during initial system set up.

### 5.13.7 ZRN (FNC 156)

| Mnemonic                      | Function                                                       | Operands                             |                |                |                                      | Program steps                        |
|-------------------------------|----------------------------------------------------------------|--------------------------------------|----------------|----------------|--------------------------------------|--------------------------------------|
|                               |                                                                | S <sub>1</sub>                       | S <sub>2</sub> | S <sub>3</sub> | D                                    |                                      |
| ZRN<br>FNC 156<br>Zero return | Return to zero home point after machine ON or initial setting. | K,H,KnX,KnY,<br>KnM,KnS<br>T,C,D,V,Z |                | X,Y,<br>M,S    | Y<br>Note: ☒<br>Y000 or<br>Y001 only | ZRN:<br>9 steps<br>DZRN:<br>17 steps |

| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|
|------------------|------------------|---------|



Contents:

When executing incremental or absolute positioning, the PLC stores the current

position values which increase or decrease during operation.

Using these values, the PLC always knows the machine position. However when the power to the PLC is turned off, this data is lost. To cope with this the machine should return to the zero point when the power is turned ON, or during initial set up, to teach the zero position. [S 1] is the Zero Return Speed, [S 2] is the Creep Speed, [S 3] is the Near Point Signal, and [D] is the Pulse Output Designation.

**Points to note:**

a)Users may specify zero return speed [S1 ] as, 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

b)Users may specify the creep speed [S2 } of 10 to 32,767Hz

c)If any device other than an input relay (X) is specified for the Near point signal [S3] it will be affected by the operation cycle of the PLC and the dispersion of the zero point may be large.

d)Only Y000 or Y001 can be used for the pulse output [D].

Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.

e)If M8140 is set to ON, the clear signal is sent to the servo motor when the return to zero point is complete.

f)Related device numbers.

D8141 (upper digit) & D8140 (lower digit) : Current value register of Y000 (32-bit)

D8143 (upper digit) & D8142 (lower digit) : Current value register of Y001 (32-bit)

D8147 (upper digit) & D8146 (lower digit) : Maximum speed when FNC156, FNC158 or FNC159 are executed 100~100,000Hz.

D8148 : Acceleration/Deceleration time adopted when FNC156, FNC158 or FNC159 are executed.

M8145 : Y000 pulse output stop (immediate)

M8146 : Y001 pulse output stop (immediate)

M8147 : Y000 pulse output monitor (BUS/READY)

M8148 : Y001 pulse output monitor (BUS/READY)

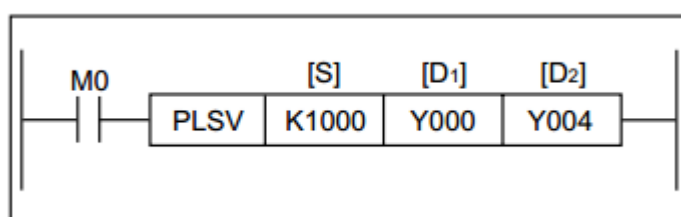
g)When a HCFA SV-X3 servo amplifier equipped with absolute position detection function is used, the current position of the servo can be read by FNC 155 (ABS).

- Dog search function is not supported. Start zero return from the front side of the Near point signal.
- Attention should be paid to the instruction drive timing.

#### 5.13.8 PLSV(FNC157)

| Mnemonic                   | Function                    | Operands                                 |                                      |       | Program steps                        |
|----------------------------|-----------------------------|------------------------------------------|--------------------------------------|-------|--------------------------------------|
|                            |                             | S                                        | D1                                   | D2    |                                      |
| PLSV<br>FNC 157<br>Pulse V | Variable speed pulse output | K,H,<br>KnX,KnY,<br>KnM,KnS<br>T,C,D,V,Z | Y<br>Note: ☒<br>Y000 or<br>Y001 only | Y,M,S | PLSV<br>9 steps<br>DPLSV<br>17 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This is a variable speed output pulse instruction, with a rotation direction

output.

[S] is the Pulse Frequency, [D 1 ] is the Pulse Output Designation, and [D2] is the Rotation Direction Signal.

**Points to note:**

a) Users may use output pulse frequencies [S1] of, 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

b) Only Y000 or Y001 can be used for the pulse output [D1 ].

Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.

c) Rotation direction signal output [D2] operated as follows: if [D 2 ] = OFF, rotation = negative, if [D 2 ] = ON, rotation = positive.

d) The pulse frequency [S] can be changed even when pulses are being output.

e) Acceleration/deceleration are not performed at start/stop. If cushion start/stop is required, increase or decrease the output pulse frequency [S] using the FNC67 RAMP instruction.

f) If the instruction drive contact turns off while pulses are output, the machine stops without deceleration

g) Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000 : [M8147] Y001 : [M8148]) is ON.

h) The normal or reverse direction is specified by the positive or negative sign of the output pulse frequency [S]

i) Related device numbers.

D8141 (upper digit) & D8140 (lower digit) : Current value register of Y000 (32-bit)

D8143 (upper digit) & D8142 (lower digit) : Current value register of Y001 (32-bit)

M8145 : Y000 pulse output stop (immediate)

M8146 : Y001 pulse output stop (immediate)

M8147 : Y000 pulse output monitor (BUS/READY)

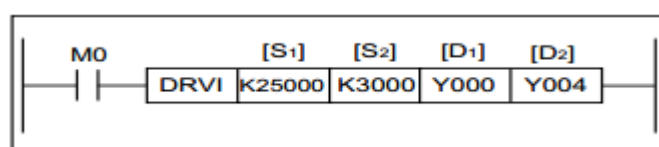
M8148 : Y001 pulse output monitor (BUS/READY)

- Attention should be paid to the instruction drive timing.

### 5.13.9 DRVI (FNC 158)

| Mnemonic                                 | Function                 | Operands                                 |                |                                          |                | Program steps                        |
|------------------------------------------|--------------------------|------------------------------------------|----------------|------------------------------------------|----------------|--------------------------------------|
|                                          |                          | S <sub>1</sub>                           | S <sub>2</sub> | D <sub>1</sub>                           | D <sub>2</sub> |                                      |
| DRVI<br>FNC 158<br>Drive to<br>increment | Increment<br>positioning | K,H,<br>KnX,KnY,<br>KnM,KnS<br>T,C,D,V,Z |                | Y<br><br>Note: ☒<br>Y000 or<br>Y001 only | Y,M,S          | DRVI<br>9 steps<br>DDRVI<br>17 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction is for single speed positioning in the form

of incremental movements. [S 1] is the Number of Pulses, [S 2] is the Pulse Output Frequency, [D1] is the Pulse Output Designation, and [DN 2] is the Rotation Direction Signal.

Points to note:

- The maximum number of pulses [S1 ] available are: 16-bit -32,768 to 32,767 pulses or 32-bit

-999,999 to 999,999 pulses.

b)Users may use output pulse frequencies [S2 ], 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

c)Only Y000 or Y001 can be used for the pulse output [D 1 ].

Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.

d)Rotation direction signal output [D2] operated as follows: if [D 2 ] = OFF, rotation = negative, if [D 2 ] = ON, rotation = positive.

e)If the contents of an operand are changed while the instruction is executed, it is not reflected on the operation. The new contents become effective when the instruction is next driven.

f)If the instruction drive contact turns off while the instruction is being executed, the machine decelerates and stops. At this time the execution complete flag M8029 does not turn ON.

g)Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000 : [M8147], Y001 : [M8148]) is ON.

h)For operation in the incremental drive method, the travel distance from the current position is specified with either a positive or a negative symbol.

i)The minimum value of output pulse frequency which can be actually used is determined by the following equation.

$$\sqrt{\text{MaxSpeed}[\text{D8147},\text{D8146}]\text{Hz} \div (2 \times (\text{Acceleration}\backslash\text{Deceleration}[\text{D8148}]\text{ms} \div 1000))}$$

f) Related device numbers.

D8145 : Bias speed adopted when either FNC158, DRVI or FNC159, DRVA are executed

D8147 (upper digit) & D8146 (lower digit) : Maximum speed when FNC156, FNC158 or FNC159 are executed 100~100,000Hz.

D8148 : Acceleration/Deceleration time adopted when FNC156, FNC158 or FNC159 are executed.

M8145 : Y000 pulse output stop (immediate)

M8146 : Y001 pulse output stop (immediate)

M8147 : Y000 pulse output monitor (BUS/READY)

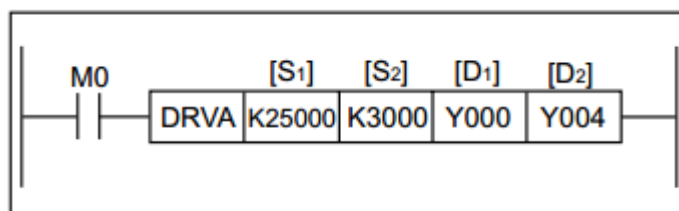
M8148 : Y001 pulse output monitor (BUS/READY)

- Attention should be paid to the instruction drive timing.

## 5.13.10 DRVA(FNC 159)

| Mnemonic                             | Function             | Operands                                 |                |                                      |                | Program steps                        |
|--------------------------------------|----------------------|------------------------------------------|----------------|--------------------------------------|----------------|--------------------------------------|
|                                      |                      | S <sub>1</sub>                           | S <sub>2</sub> | D <sub>1</sub>                       | D <sub>2</sub> |                                      |
| DRVA<br>FNC 159<br>Drive to absolute | Absolute positioning | K,H,<br>KnX,KnY,<br>KnM,KnS<br>T,C,D,V,Z |                | Y<br>Note: ☒<br>Y000 or<br>Y001 only | Y,M,S          | DRVA<br>9 steps<br>DDRVA<br>17 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Operation:

This instruction is for single speed positioning using a zero home point and absolute measurements. [S 1] is the Number of Pulses, [S 2 ] is the Output Frequency, [D1] is the Pulse Output Designations, and [D 2 ] is the Rotation Direction Signal.

**Points to note:**

a)The target position for absolute positioning [S1 ] can be: 16-bit -32,768 to 32,767 pulses or 32-bit -999,999 to 999,999 pulses.

b)Users may use output pulse frequencies [S2 ], 16-bit 10 to 32,767Hz or 32-bit 10 to 100kHz.

c)Only Y000 or Y001 can be used for the pulse output [D 1 ].

Because of the nature of the high speed output, transistor type output units should be used with this instruction. Relay type outputs will suffer a greatly reduced life, and will cause false outputs to occur.

d)Rotation direction signal output [D2] operated as follows: if [D 2 ] = OFF, rotation = negative, if [D 2 ] = ON, rotation = positive.

e)If the contents of an operand are changed while the instruction is executed, it is not reflected on the operation. The new contents become effective when the instruction is next driven.

f)If the instruction drive contact turns off while the instruction is being executed, the machine

decelerates and stops. At this time the execution complete flag M8029 does not turn

ON.

g) Once the instruction drive contact is off, re-drive of the instruction is not possible while the pulse output flag (Y000 : [M8147], Y001 : [M8148]) is ON.

h) For operation in the absolute drive method, the travel distance from the zero point is specified.

i) The minimum value of output pulse frequency which can be actually used is determined by the following equation

$$\sqrt{\text{MaxSpeed}[\text{D8147}, \text{D8146}] \text{Hz} \div (2 \times (\text{Acceleration/Deceleration}[\text{D8148}] \text{ms} \div 1000))}$$

f) Related device numbers.

D8145 : Bias speed adopted when either FNC158, DRVI or FNC159, DRVA are executed

D8147 (upper digit) & D8146 (lower digit) : Maximum speed when FNC156, FNC158 or FNC159 are executed 100~100,000Hz.

D8148 : Acceleration/Deceleration time adopted when FNC156, FNC158 or FNC159 are executed.

M8145 : Y000 pulse output stop (immediate)

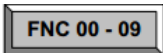
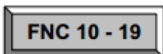
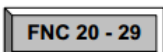
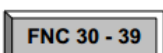
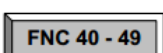
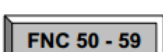
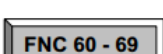
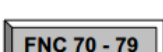
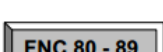
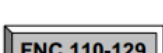
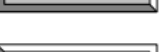
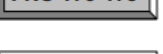
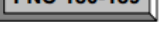
M8146 : Y001 pulse output stop (immediate)

M8147 : Y000 pulse output monitor (BUS/READY)

M8148 : Y001 pulse output monitor (BUS/READY)

- Attention should be paid to the instruction drive timing.

Applied Instructions:

|                                                                                     |                                                                                     |                                                |       |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|------------------------------------------------|-------|
| 1.                                                                                  |    | Program Flow                                   | 5-4   |
| 2.                                                                                  |    | Move And Compare                               | 5-16  |
| 3.                                                                                  |    | Arithmetic And Logical Operations (+, -, ×, ÷) | 5-24  |
| 4.                                                                                  |    | Rotation And Shift                             | 5-34  |
| 5.                                                                                  |    | Data Operation                                 | 5-42  |
| 6.                                                                                  |    | High Speed Processing                          | 5-52  |
| 7.                                                                                  |    | Handy Instructions                             | 5-66  |
| 8.                                                                                  |    | External FX I/O Devices                        | 5-80  |
| 9.                                                                                  |    | External FX Serial Devices                     | 5-94  |
| 10.                                                                                 |   | Floating Point 1 & 2                           | 5-110 |
| 11.                                                                                 |  | Trigonometry (Floating Point 3)                | 5-118 |
| 12.                                                                                 |  | Data Operations 2                              | 5-122 |
| 13.                                                                                 |  | Positioning Control                            | 5-126 |
|  |  | Real Time Clock Control                        | 5-136 |
| 15.                                                                                 |  | Gray Codes                                     | 5-146 |
| 16.                                                                                 |  | Additional Functions                           | 5-146 |
| 17.                                                                                 |  | In-line Comparisons                            | 5-150 |



### Symbols list:

D - Destination device.

S - Source device.

m, n- Number of active devices, bits or an operational constant.

Additional numeric suffixes will be attached if there are more than one operand with the same function e.g. D1,S3or for lists/abled devices D3+0,S+9etc.

MSB - Most Significant Bit, sometimes used to indicate the mathematical sign of a number, i.e. positive = 0, and negative = 1.

LSB - Least Significant Bit.

Instruction modifications:

☆☆☆ - An instruction operating in 16 bit mode, where ☆☆☆ identifies the instruction mnemonic.

☆☆☆ P - A 16 bit mode instruction modified to use pulse (single) operation.

D☆☆☆ - An instruction modified to operate in 32 bit operation.

D☆☆☆P - A 32 bit mode instruction modified to use pulse (single) operation.



A repetitive instruction which will change the destination value on every scan unless modified by the pulse function.

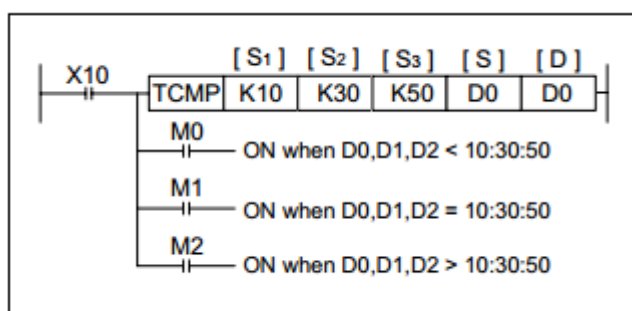


An operand which cannot be indexed, i.e. The addition of V or Z is either invalid or will have no effect to the value of the operand.

### 5.14.1 TCMP (FNC 160)

| Mnemonic                          | Function                                             | Operands                                |    |    |                                                     |         | Program steps        |
|-----------------------------------|------------------------------------------------------|-----------------------------------------|----|----|-----------------------------------------------------|---------|----------------------|
|                                   |                                                      | S1                                      | S2 | S3 | S                                                   | D       |                      |
| TCMP<br>FNC 160<br>(Time Compare) | Compares two times - results of <, = and > are given | K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z |    |    | T, C, D<br>Note:<br>3 consecutive devices are used. | Y, M, S | TCMP, TCMP: 11 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Contents:

S1,S2and S3represent hours, minutes and seconds respectively.

This time is compared to the time value in the 3 data devices

specified by the head address S. The result is indicated in the 3 bit devices specified by the head address D.

The bit devices in D indicate the following:

D+0issetON,whenthe timeinSislessthanthetimeinS1,S2and S3.

D+1is set ON, when the time in S is equal to the time in S1,S2and S3.

D+2is set ON, when the time in S is greater than the time in S1,S2and S3.

Points to note:

a) The status of the destination devices is kept, even if the TCMP instruction is deactivated.

b) The comparison is based on the time value specified in the source devices.

- The valid range of values for S1 and S+0 is 0 to 23 (Hours).

- The valid range of values for S2 and S+1 is 0 to 59 (Minutes).

- The valid range of values for S3 and S+2 is 0 to 59 (Seconds).

c) The current time of the real time clock can be compared by specifying D8015

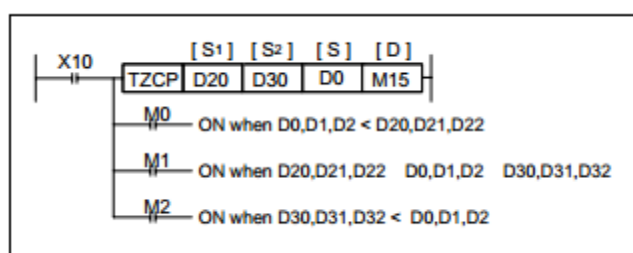
(Hours),

D8014 (Minutes) and D8013 (Seconds) as the devices for S1, S2 and S3 respectively.

#### 5.14.2 TZCP (FNC 161)

| Mnemonic                                     | Function                                                                                | Operands                                                                                        |    |   |         | Program steps              |
|----------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|----|---|---------|----------------------------|
|                                              |                                                                                         | S1                                                                                              | S2 | S | D       |                            |
| TZCP<br>FNC 161<br>(Time<br>Zone<br>Compare) | Compares a time<br>to a specified time<br>range - results of<br><, = and > are<br>given | T, C, D<br>S1 must be less than or equal to S2.<br>Note: 3 consecutive devices are used for all |    |   | Y, M, S | TZCP,<br>TZCPP:<br>9 steps |

|                  |                  |         |
|------------------|------------------|---------|
| 16 BIT OPERATION | 32 BIT OPERATION | PULSE-P |
|------------------|------------------|---------|



Contents:

S1, S2 and S represent time values. Each specifying the head address of 3 data devices. S is

compared to the time period defined by S1 and S2. The result is indicated in the 3 bit

devices specified by the head address D.

The bit devices in D indicate the following:

D+0 is set ON, when the time in S is less than the times in S1 and S2.