Multiplying 32-bit binary data

D*(P)(_U) instruction and DMUL(P)(_U) instruction can be used for multiplication of 32-bit binary data.

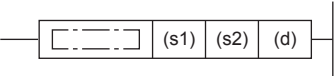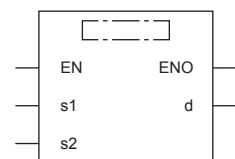## D*(P)(_U)

| FX5S | FX5UJ | FX5U | FX5UC |

These instructions multiply the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text[1]	
▢─▢▢ (s1) (s2) (d)	ENO:=DMULTI(EN,s1,s2,n,d); ENO:=DMULTIP(EN,s1,s2,n,d);	ENO:=DMULTI_U(EN,s1,s2,n,d); ENO:=DMULTIP_U(EN,s1,s2,n,d);

FBD/LD
▢─▢▢ EN   ENO s1   d s2

("DMULTI", "DMULTIP", "DMULTI_U", "DMULTIP_U" enters ▢.)

[1]   Supported by engineering tool version "1.035M" and later.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	D*(P)	Multiplicand data or the head device where the data to be multiplied by another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D*(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	D*(P)	Multiplier data or the head device where the data by which another is to be multiplied is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	D*(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	D*(P)	Head device for storing the operation result	—	64-bit signed binary	ANY32_S_ARRAY (Number of elements: 2)
	D*(P)_U			64-bit unsigned binary	ANY32_U_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word				Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U▢\G▢	Z	LC	LZ			K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—	
(s2)	○	○	○	○	○	○	○	○	—	—	—	
(d)	○	○	—	—	○	—	○	—	—	—	—	

Processing details

- These instructions multiply the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the multiplication result in the device specified by (d).

- When (d) is a bit device, only the lower 32 bits of the multiplication result are stored and the upper 32 bits cannot be specified. If the upper 32 bits data of the multiplication operation result are required, temporarily store the result in a word device, and transfer the data stored in word device ((d)+2) and ((d)+3) to the specified bit devices.

Ex.

Multiplication result when (d) is a bit device

- K1 ... Lower 4 bits (b0 to b3)
- K4 ... Lower 16 bits (b0 to b15)
- K8 ... Lower 32 bits (b0 to b31)

Operation error

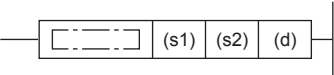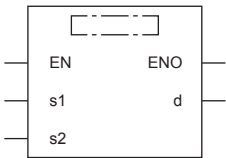| Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.

7

DMUL(P)(_U)

| FX5S | FX5UJ | FX5U | FX5UC |

These instructions multiply the 32-bit binary data in the device specified by (s1) and the 32-bit binary data in the device specified by (s2), and store the result in the device specified by (d).

Ladder diagram	Structured text	
⟦ ⟧ (s1) (s2) (d)	ENO:=DMUL(EN,s1,s2,d); ENO:=DMULP(EN,s1,s2,d);	ENO:=DMUL_U(EN,s1,s2,d); ENO:=DMULP_U(EN,s1,s2,d);

FBD/LD
EN   ENO s1   d s2

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	DMUL(P)	Multiplicand data or the head device where the data to be multiplied by another is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DMUL(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(s2)	DMUL(P)	Multiplier data or the head device where the data by which another is to be multiplied is stored	-2147483648 to +2147483647	32-bit signed binary	ANY32_S
	DMUL(P)_U		0 to 4294967295	32-bit unsigned binary	ANY32_U
(d)	DMUL(P)	Head device for storing the operation result	—	64-bit signed binary	ANY32_S_ARRAY (Number of elements: 2)
	DMUL(P)_U			64-bit unsigned binary	ANY32_U_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	○	○	○	○	—	—	—
(s2)	○	○	○	○	○	○	○	○	—	—	—
(d)	○	○	○	—	○	—	○	—	—	—	—

Processing details

- These instructions multiply the 32-bit binary data in the device specified by (s1) by the 32-bit binary data in the device specified by (s2), and store the multiplication result in the device specified by (d).

(s1)+1	(s1)		(s2)+1	(s2)		(d)+3	(d)+2	(d)+1	(d)
b31 ⋯ b16	b15 ⋯ b0		b31 ⋯ b16	b15 ⋯ b0		b63 ⋯ b48	b47 ⋯ b32	b31 ⋯ b16	b15 ⋯ b0
567890 (BIN)		×	123456 (BIN)		⇒	70109427840 (BIN)			

- When nibble is specified ranging from K1 to K8 for (d), the result is obtained only for the lower-order 32 bits, and is not obtained for the higher-order 32 bits. Transfer the data to word devices once, then execute the operation.
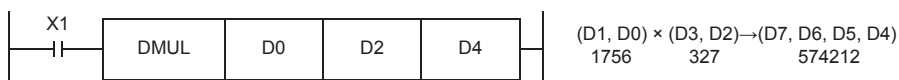
```
      X0        (s1)    (s2)    (d)
 ┤├────┬── DMUL   D50    K150   D100    (D51,D50)  (D103,D102,D101,D100)
      │                                 K100 × K150 → K15000
      │
      ├──────── DMOV   D100   K8Y0     D100 → Y17 to Y0
      │                                 D101 → Output to Y37 to Y20
      │
      └──────── DMOV   D102   K8Y40    D102 → Y57 to Y40
                                        D103 → Output to Y77 to Y60
```
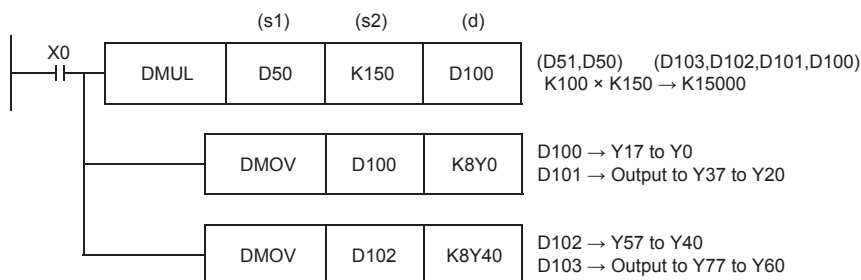
■Related flag

Device	Name	Description
SM8304	Zero	When the operation result is 0, the zero flag is turned ON.

Precautions

- Even if word devices are used, the operation result (64 bits binary data) cannot be monitored at one time. In such a case, a floating point operation is recommended.

Program example

```
      X1
 ┤├──────── DMUL   D0     D2     D4     (D1, D0) × (D3, D2)→(D7, D6, D5, D4)
                                        1756      327        574212
```

Operation error

Error code (SD0/SD8067)	Description
2820H	The device range specified by (d) exceeds the corresponding device range.