CHƯƠNG 4
CÁC LỆNH ỨNG DỤNG

I. Nhóm lệnh về điều khiển lưu trình _ Các hàm từ 00 đến 09:

Nội dung:

CJ	_	Conditional Jump	FNC00
CALL	_	Call Subroutine	FNC01
SRET	_	Subroutine Return	FNC02
IRET	_	Interrupt Return	FNC03
EI	_	Enable Interrupt	FNC04
DI	_	Disable Interrupt	FNC05
FEND	_	First End	FNC06
WDI	_	Watchdog Timer	FNC07
FOR	_	Start of a For/Next Loop	FNC08
NEXT	_	End of a For/Next Loop	FNC09

Các ký hiệu:

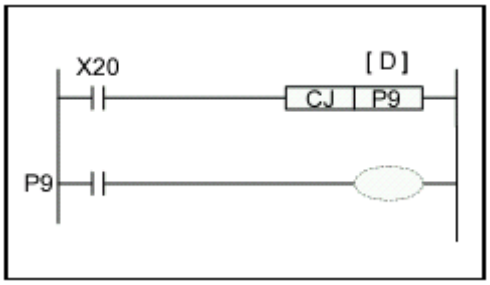
D	_	Toán hạng đích
S	_	Toán hạng nguồn
m, n	_	Số ký hiệu cho thiết bị, nhóm bit hay hằng số

Sự hiệu chỉnh lệnh:

☆☆☆	Lệnh hoạt động ở chế độ 16 bit, trong đó ☆☆☆ chỉ tên lệnh.
☆☆☆P	Lệnh đã hiệu chỉnh để dùng tác vụ xung 16 bit.
D☆☆☆	Lệnh đã hiệu chỉnh để hoạt động trong tác vụ 32 bit.
D☆☆☆P	Lệnh đã hiệu chỉnh để dùng tác vụ xung 32 bit.

1.1. Lệnh CJ (FNC 00)

Lệnh	Chức năng	Toán hạng	Số bước
		D	
CJ FNC 00 (Conditional Jump)	Nhảy tới một vị trí con trỏ đích đã định	Các con trỏ đích hợp lệ có giá trị từ 0 - 63	CJ, CJP: 3 bước Con trỏ đích P☆☆: 1 bước



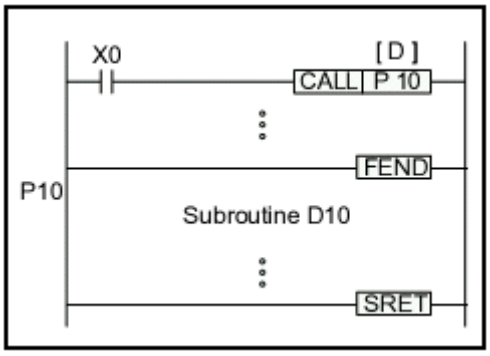
Hoạt động:

Khi lệnh CJ được thi hành, nó buộc chương trình nhảy tới vị trí xác định trong chương trình. Lệnh này sẽ bỏ qua một số bước chương trình nào đó. Có nghĩa là chúng không được xử lý trong chương trình. Điều này làm tăng tốc độ quét chương trình.

1.2. Lệnh CALL (FNC 01)

Lệnh	Chức năng	Toán hạng	Số bước
		D	
CALL FNC 01 (Call sub-routine)	Gọi chương trình con hoạt động	Con trỏ chương trình con từ 0 – 62. Số mức lồng: 5 kể cả lệnh CALL khởi tạo.	CALL, CALLP: 3 bước Con trỏ chương trình con P☆☆: 1 bước

Hoạt động:



Khi lệnh CALL được tác động, nó sẽ cho chạy chương trình con từ chỗ có con trỏ chương trình con được gọi (đoạn chương trình Subroutine P10). Lệnh này phải được dùng với lệnh FEND (FNC 06) và SRET (FNC 02).

Chương trình sẽ nhảy đến con trỏ chương trình con (sau lệnh FEND) và xử lý các lệnh trong chương trình con đó cho đến khi gặp lệnh SRET và trở về dòng chương trình ngay sau lệnh CALL.

1.3. Lệnh SRET (FNC 02)

Lệnh	Chức	Toán hạng	Số bước
		D	
SRET FNC 02 (Subroutine return)	Trở về từ chương trình con	Không có Tự động trở về bước ngay sau lệnh CALL kích hoạt chương trình con	SRET: 1 bước

Hoạt động:

SRET báo hiệu kết thúc chương trình con hiện hành và trở về bước ngay sau lệnh CALL đã kích hoạt chương trình con đó.

1.4. IRET, EI, DI (FNC 03, 04, 05)

Lệnh	Chức năng	Toán hạng D	Số bước
IRET FNC 03 (Interrupt return)	Trở về từ chương trình ngắt	Không có Tự động trở về bước chương trình chính khi đang xử lý ngắt	IRET: 1 bước
EI FNC 04 (Enable interrupts)	Cho phép các ngõ vào ngắt	Không có Bất kỳ ngõ vào ngắt bị kích hoạt sau lệnh EI và trước lệnh FEND hoặc DI thì sẽ được xử lý ngay trừ khi nó bị vô hiệu	EI: 1 bước
DI FNC 05 (Disable interrupts)	Vô hiệu hóa việc xử lý chương trình ngắt	Không có Bất kỳ ngõ vào ngắt bị kích hoạt sau lệnh DI và trước lệnh EI thì sẽ lưu lại cho đến khi có lệnh EI chờ	DI: 1 bước
I (Interrupt pointer)	Chỉ định điểm bắt đầu của 1 chương trình ngắt	Một mã 3 số cho biết loại ngắt và hoạt động của ngắt	I☆☆☆: 1 bước

1.5. FEND (FNC 06)

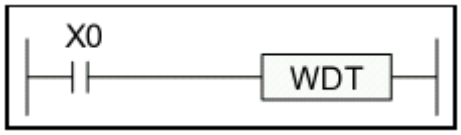
Lệnh	Chức năng	Toán hạng D	Số bước
FEND FNC 06 (First end)	Dùng để chỉ cuối khối chương trình chính	Không có Lưu ý: Có thể dùng với lệnh CJ (FNC 00), CALL (FNC 01) và chương trình ngắt	FEND: 1 bước

Hoạt động:

Lệnh FEND chỉ định điểm kết thúc của chương trình chính và điểm bắt đầu của đoạn chương trình con. Khi hoạt động bình thường lệnh FEND hoạt động giống như lệnh END, nghĩa là: việc xử lý ngõ ra, ngõ vào và làm tươi bộ định thì watchdog được thực hiện khi thi hành đến lệnh này.

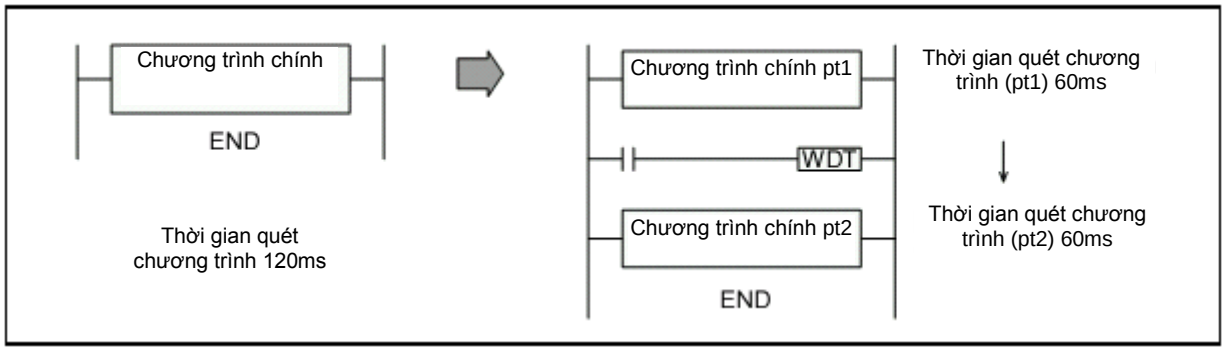
1.6. WDT (FNC 07)

Lệnh	Chức năng	Toán hạng D	Số bước
WDT FNC 07 (Watch dog timer refresh)	Dùng để làm tươi bộ định thì watchdog trong suốt thời gian quét chương trình	Không có Có thể được kích hoạt bất cứ lúc nào trong chương trình chính	WDT, WDTP: 1 bước



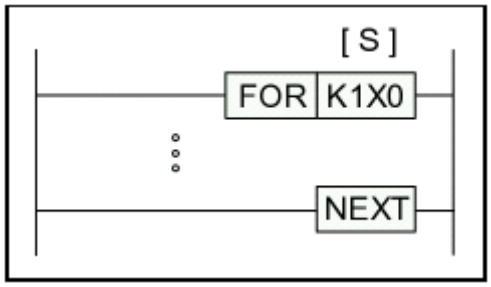
Hoạt động:

Lệnh WDT làm tươi bộ định thì watchdog của bộ điều khiển. Bộ định thì watchdog kiểm tra xem thời gian quét chương trình có vượt quá giới hạn thời gian tùy chọn đã định. Giả sử rằng nếu giới hạn thời gian này bị vi phạm thì chương trình sẽ báo lỗi. kế đến PC sẽ dừng hoạt động để ngăn các lỗi khác xảy ra. Bằng cách làm tươi bộ định thì watchdog (dùng lệnh WDT) thời gian quét được tăng lên.



1.7. FOR, NEXT (FNC 08, 09)

Lệnh	Chức năng	Toán hạng	Số bước
		S	
FOR FNC 08 (Start of a FOR-NEXT loop)	Xác định vị trí bắt đầu và số lần lặp của vòng lặp	K, H KnX, KnY, KnM, KnS, T, C, D, V, Z	FOR: 3 bước
NEXT FNC 09 (End of a FOR-NEXT loop)	Xác định vị trí cuối của vòng lặp	Không có Lưu ý: Vòng FOR-NEXT có thể được lồng 5 mức nhĩa là lập trình d0ược 5 vòng lặp FOR-NEXT lồng nhau	NEXT: 1 bước



Hoạt động:

Các lệnh FOR và NEXT cho phép một đoạn chương trình được lặp lại một số S lần.

II. Nhóm lệnh về di chuyển vùng nhớ và so sánh_ Các hàm từ 10 đến 19:

Nội dung:

CMP	_	Compare	FNC 10
ZCP	_	Zone Compare	FNC 11
MOV	_	Move	FNC 12
SMOV	_	Shift Move	FNC 13
CML	_	Compliment	FNC 14
BMOV	_	Block Move	FNC 15
FMOV	_	Fill Move	FNC 16
XCH	_	Exchange	FNC 17
BCD	_	Binary Coded Decimal	FNC 18
BIN	_	Binary	FNC 19

Các ký hiệu:

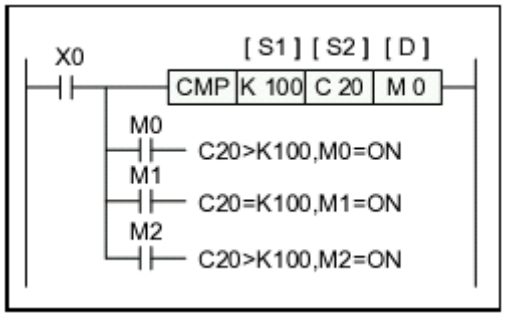
D	_	Toán hạng đích
S	_	Toán hạng nguồn
m, n	_	Số ký hiệu cho thiết bị, nhóm bit hay hằng số

Sự hiệu chỉnh lệnh:

☆☆☆	Lệnh hoạt động ở chế độ 16 bit, trong đó ☆☆☆ chỉ tên lệnh.
☆☆☆P	Lệnh đã hiệu chỉnh để dùng tác vụ xung 16 bit.
D☆☆☆	Lệnh đã hiệu chỉnh để hoạt động trong tác vụ 32 bit.
D☆☆☆P	Lệnh đã hiệu chỉnh để dùng tác vụ xung 32 bit.

2.1. Lệnh CMP (FNC 10)

Lệnh	Chức năng	Toán hạng			Số bước
		S_1	S_2	D	
CMP FNC 10 (Compare)	So sánh 2 giá trị dữ liệu cho kết quả <, =, >	K, H KnX, KnY, KnM, KnS, T, C, D, V, Z		Y, M, S Lưu ý: Dùng 3 thiết bị liên tiếp nhau	CMP, CMPP: 7 bước DCMP, DCMPP: 13 bước



Hoạt động:

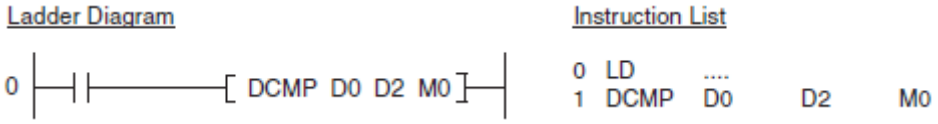
S1 được so sánh với dữ liệu S2. Kết quả so sánh được thể hiện bằng 3 bit có địa chỉ đầu chứa trong D cho biết:

S2 < S1 _ bit D là ON

S2 = S1 _ bit D+1 là ON

S2 > S1 _ bit D+2 là ON

Để so sánh dữ liệu 32 bit ta dùng lệnh DCMP thay cho CMP.

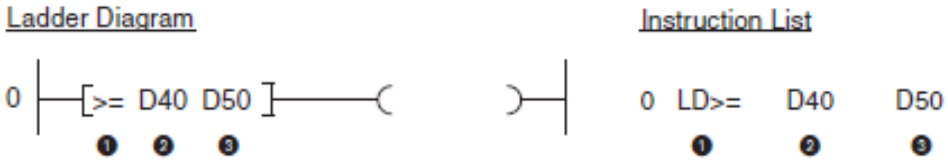


Ở ví dụ trên thanh ghi D0 và D1 so sánh với thanh ghi D2 và D3. Kết quả so sánh được thể hiện bằng 3 bit, tương tự như so sánh dữ liệu 16 bit.

❖ **Lệnh so sánh bên trong hoạt động logic.**

Lệnh CMP được mô tả là kết quả cuối cùng của lệnh so sánh được lưu trữ vào 3 thiết bị bit. Tuy nhiên, khi ta cần kích hoạt 1 ngõ ra hay hoạt động logic trên cơ sở kết quả của sự so sánh, và không cần sử dụng 3 thiết bị bit. Chúng ta có thể thực hiện lệnh này với lệnh "load compare"

So sánh bắt đầu bằng hoạt động logic.



❶ Điều kiện so sánh

❷ Giá trị so sánh thứ nhất (S1)

❸ Giá trị so sánh thứ 2 (S2)

Nếu điều kiện so sánh đúng, trạng thái sau khi so sánh được bật lên "1". Trạng thái sau khi so sánh được bật "0" khi điều kiện so sánh sai.

- So sánh bằng = (S1 = S2)

  Lệnh out được set 1 khi 2 giá trị bằng nhau bằng nhau.

- So sánh lớn hơn > (S1 > S2)

  Lệnh out được set 1 khi giá trị đầu lớn hơn giá trị thứ hai

- So sánh nhỏ hơn < (S1 < S2)

Lệnh out được set 1 khi giá trị đầu nhỏ hơn giá trị thứ hai

- So sánh không bằng <> (S1 <> S2)

Lệnh out được set 1 khi giá trị đầu khác giá trị thứ hai

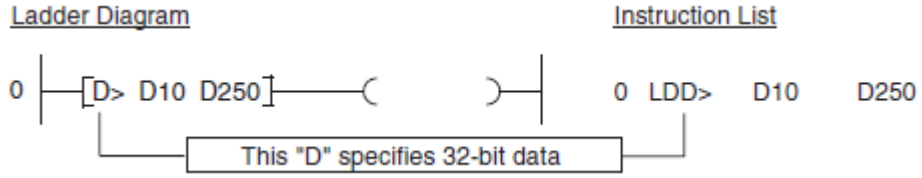
- So sánh nhỏ hơn bằng ≤ (S1 ≤ S2)

Lệnh out được set 1 khi giá trị đầu nhỏ hơn hay bằng giá trị thứ hai

- So sánh lớn hơn bằng ≥ (S1 ≥ S2).

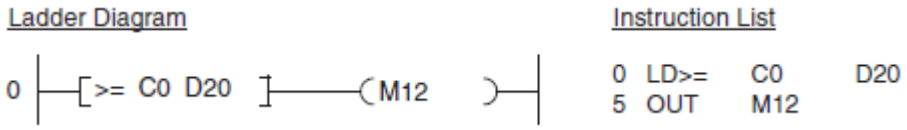
Lệnh out được set 1 khi giá trị đầu lớn hơn hay bằng giá trị thứ hai

Để so sánh dữ liệu 32 bit, ta thêm D trước lệnh so sánh

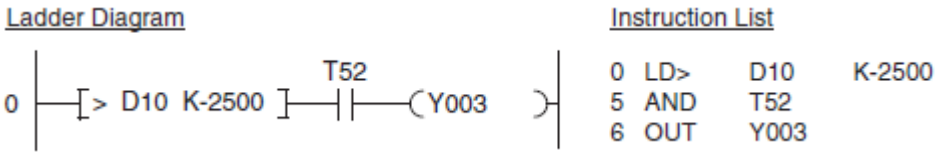


Ví dụ trên kiểm tra dữ liệu ở thanh ghi D10 và D11 lớn hơn dữ liệu thanh ghi D250 và D251.

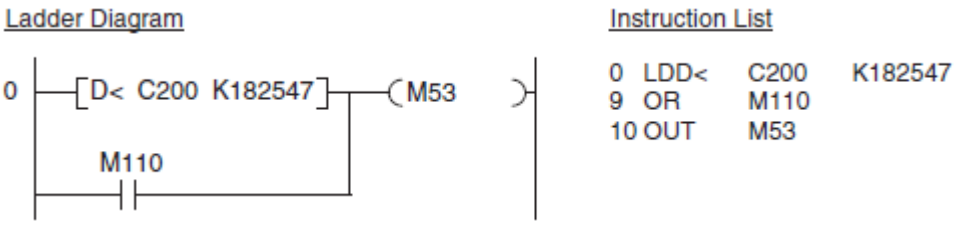
Ví dụ:



Relay M12 được kích hoạt khi giá trị của counter C0 lớn hơn hay bằng nội dung thanh ghi D20.



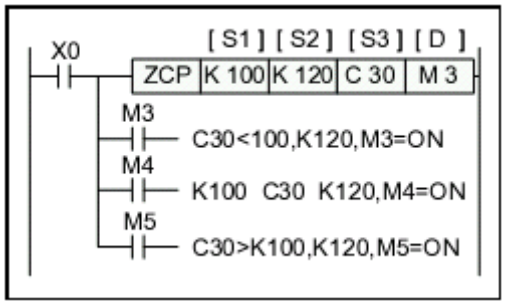
Ngõ ra Y003 sẽ được kích hoạt khi nội dung của D10 lớn hơn -2500 và relay thời gian T52 đã hoạt động xong.



Relay M53 được kích hoạt nếu giá trị counter C200 nhỏ hơn 182547 hay relay M110 được kích hoạt.

2.2. Lệnh ZCP (FNC 11)

Lệnh	Chức năng	Toán hạng				Số bước
		S_1	S_2	S_3	D	
ZCP FNC 11 (Zone compare)	So sánh 1 dãy dữ liệu với một giá trị dữ liệu cho kết quả <, =, >	K, H KnX, KnY, KnM, KnS, T, C, D, V, Z Lưu ý: S1 nên nhỏ hơn S2			Y, M, S Lưu ý: Dùng 3 thiết bị liên tiếp nhau	ZCP, ZCPP: 9 bước DZCP, DZCPP: 17 bước



Hoạt động:

Hoạt động giống như lệnh CMP chỉ khác là một trị dữ liệu đơn S3 được so sánh với 1 dãy dữ liệu S1-S2

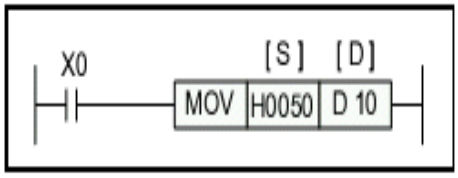
S3 < (S1-S2) – bit D là ON

(S1 < S3 < S2) – bit D+1 là ON

(S1-S2) < S3 – bit D+2 là ON

2.3. Lệnh MOV (FNC 12)

Lệnh	Chức năng	Toán hạng		Số bước
		S	D	
MOV FNC 12 (Move)	Di chuyển dữ liệu từ vùng nhớ này đến vùng nhớ khác	K, H KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	MOV, MOVP: 5 bước DMOV, DMOVP: 9 bước

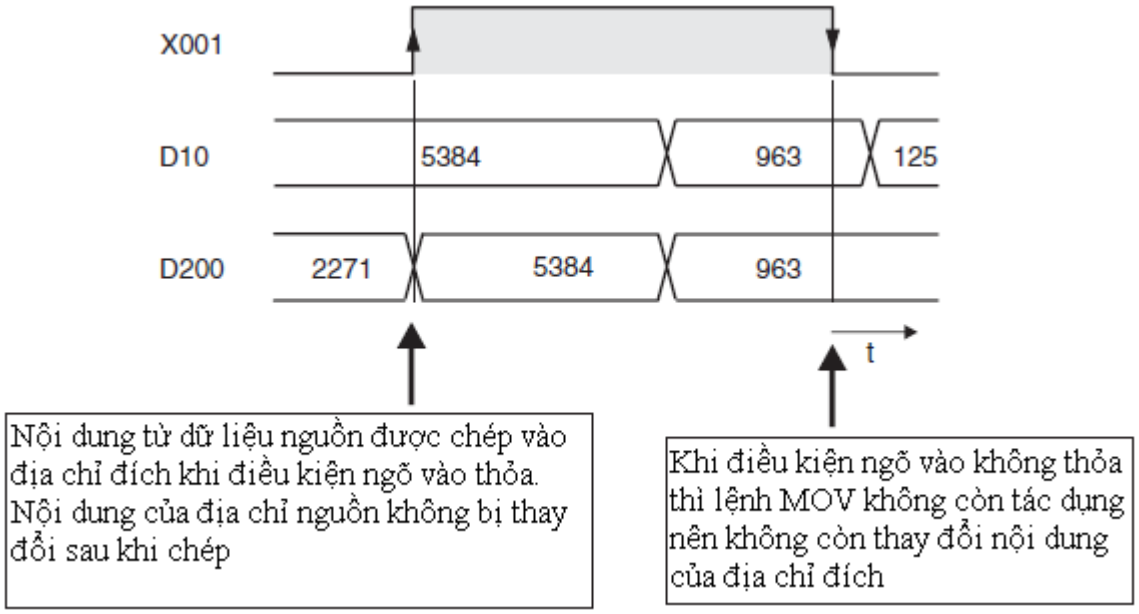


Hoạt động:

Nội dung các thiết bị nguồn (S) được sao chép vào thiết bị đích (D) khi thỏa điều kiện ngõ vào. Nếu không tác động lệnh MOV thi không có gì xảy ra.

Ví dụ:

Ở ví dụ trên, dữ liệu trong thanh ghi D10 được chép vào trong thanh ghi D200 khi ngõ vào X001 được kích hoạt. Kết quả được biểu diễn ở giản đồ sau:

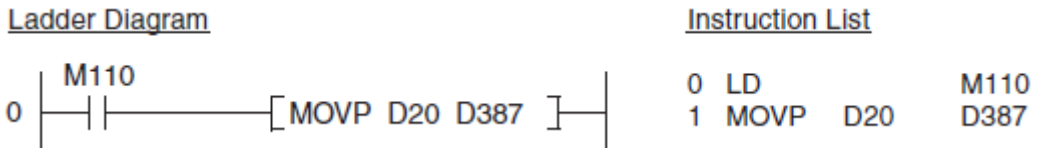


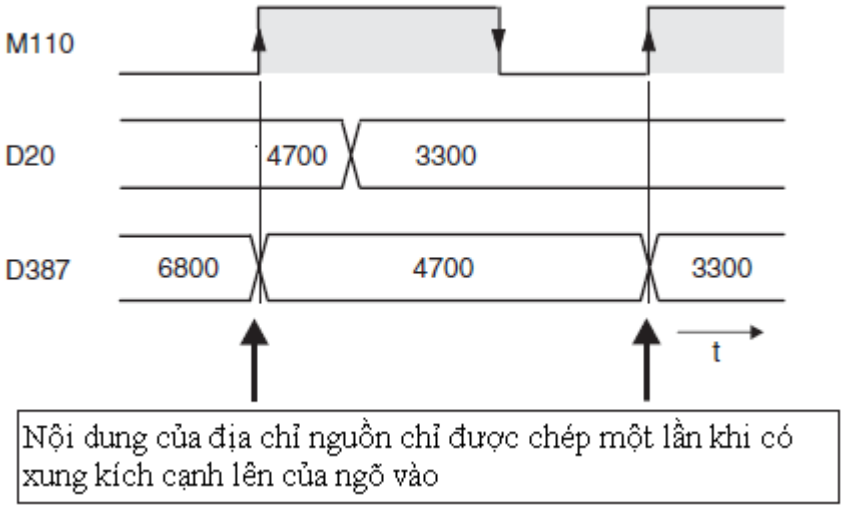
❖ **Thực hiện xung kích của lệnh MOV.**

Trong một vài ứng dụng sẽ tốt hơn nếu dữ liệu được chép đến địa chỉ đích trong chương trình chỉ trong một chu kỳ. Ví dụ, lệnh này sẽ được thực hiện nếu các lệnh khác trong chương trình cũng chép đến cùng địa chỉ hay lệnh phải được hoạt động trong khoảng thời gian được xác định.

Nếu thêm "P" vào lệnh MOV (MOVP) thì lệnh chỉ được thực hiện một lần .

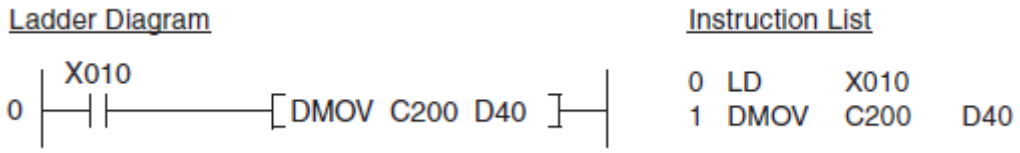
Trong ví dụ dưới nội dung của thanh ghi D20 được chép vào thanh ghi D387, khi trạng thái của M110 chuyển từ "0" lên "1".



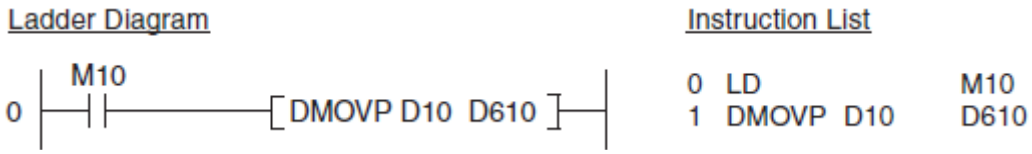


❖ **MOV dữ liệu 32 bit.**

Để dịch chuyển dữ liệu 32 bit chỉ cần thêm "D"vào lệnh MOV



Khi ngõ vào X010 ON, dữ liệu từ counter C200 được chép vào thanh ghi D40 và D41. Thanh ghi D40 chứa các bit có địa chỉ thấp.



Khi relay M10 được kích hoạt, dữ liệu thanh ghi D10 và D11 được chép vào thanh ghi D610 và D611.

❖ **Di chuyển một nhóm bit.**

Những chuỗi liên tiếp của các relay hay các thiết bị bit khác cũng có thể được dùng để cất giữ những giá trị, và có thể sao chép các giá trị này như một nhóm bằng các lệnh ứng dụng, để làm điều này cần phải thêm "K" vào trước thiết bị đầu tiên cần chép, và chỉ rõ số lượng thiết bị muốn sao chép.

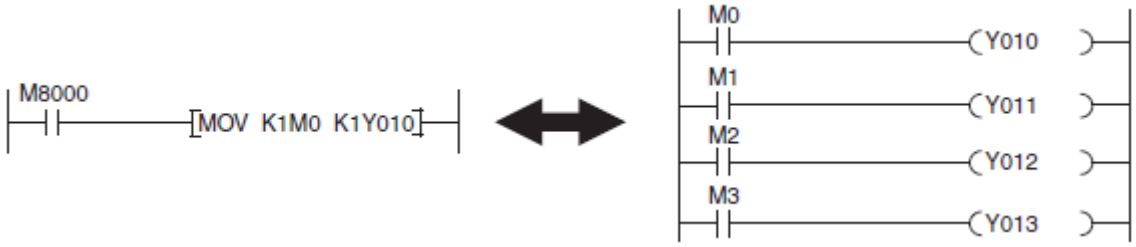
Cách biểu diễn dạng trên có dạng Kn ☆ ở đó ☆ biểu diễn đại chỉ đầu của nhóm bit đang xét. Số Kn xác định số bit "n" có thể là một số từ 0 đến 8. Mỗi đơn vị của "n" biểu diễn 4 bit, nghĩa là K1 = 4 bit, K8 = 32 bit. Do đó nhóm bit phải chia hết cho 4. K1 đến K4 hợp lệ với dữ liệu 16 bit, K1 đến K8 hợp lệ với đối với dữ liệu 32 bit.

Ví dụ: K2M0 xác định 2 nhóm 4 bit M0 đến M3 và M4 đến M7, tổng cộng có 8 bit hay 1 byte.

- K1X0: 4 ngõ vào, bắt đầu tại X0.
- K2X4: 8 ngõ vào, bắt đầu tại X4.
- K4M16: 16 relay, bắt đầu tại M16.

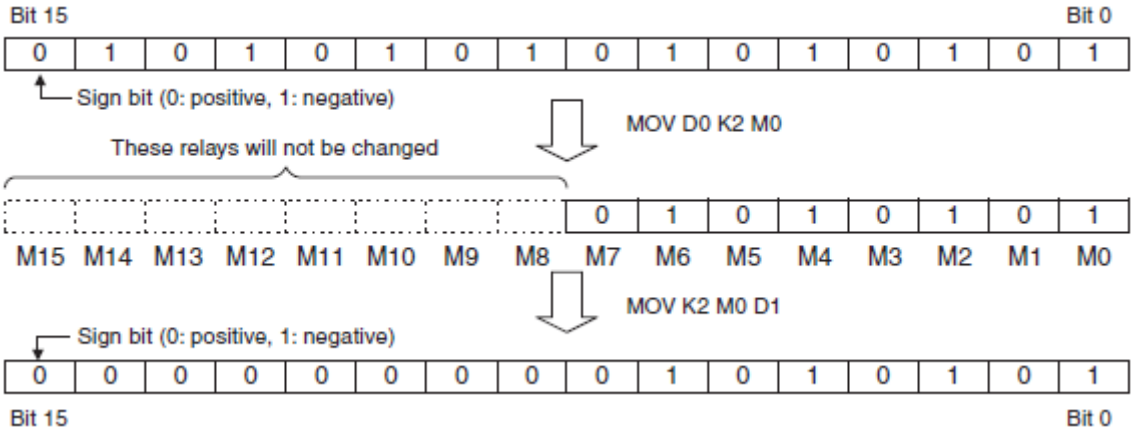
Sự kích hoạt nhiều thiết bị bằng một lệnh làm lập trình nhanh hơn và chương trình gọn hơn.

Ví dụ:



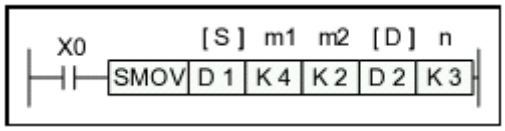
Nếu phạm đến nhỏ hơn phạm vi nguồn thì những biết dư được bỏ qua. Nếu phạm vi đích lớn hơn phạm vi nguồn thì "0" được viết vào các thiết bị dư. Chú ý, khi điều này xảy ra thì kết quả luôn dương.

Ví dụ:



2.4. Lệnh SMOV (FNC 13)

Lệnh	Chức năng	Toán hạng					Số bước
		m1	m2	n	S	D	
SMOV FNC 13 (Shift move)	Lấy các phần tử của số thập phân 4 chữ số và chèn vào vị trí mới có 4 chữ số	K, H Lưu ý: Có thể dùng số từ 1 đến 4			K, H KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	SMOV, SMOVP: 11 bước
					Thập phân: 0 – 9,999 BCD: 0 – 9,999 khi dùng M8168		



Hoạt động 1:

Lệnh này sao chép các con số của số thập phân 1 số của nguồn S vào đích D (cũng là số thập phân 4 số).

Dữ liệu của số thập phân đích bị ghi chồng. Sư tính toán số tậhp phân dùng được trên tất cả các bộ điều khiển FX và FX2C. Trong đó:

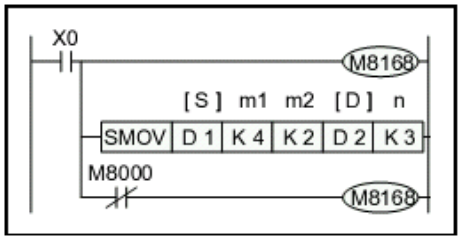
m1 - vị trí của con số thứ nhất trên thiết bị nguồn

m2 - số con số trên toán hạng nguồn

n - vị trí đích cho con số đầu tiên

Hoạt động 2:

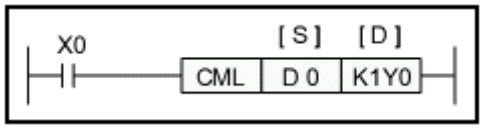
Việc bổ sung lệnh SMOV cho phép thực hiện các số BCD hoàn toàn giống như các số tậhp phân nghĩa là lệnh này sao chép các con số từ nguồn S là số BCD 4 số vào đích D (cũng là số BCD 4 số).



Để chọn chế độ BCD thì lệnh SMOV được dùng kết hợp với cờ chuyên dùng M8168 là ON. Từ đó tất cả lệnh SMOV sẽ hoạt động ở dạng BCD cho đến khi chế độ này được reset nghĩa là M8168 là OFF.

2.5. Lệnh CML (FNC 14)

Lệnh	Chức năng	Toán hạng		Số bước
		S	D	
CML FNC 14 (Compli-ment)	Sao chép và nghịch đảo chuỗi bit nguồn sang đích	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	CML, CMLP: 5 bước DCML. DCMLP: 9 bước



Hoạt động:

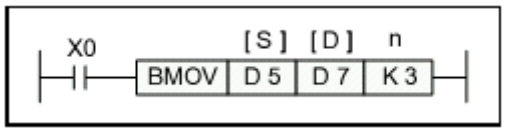
Mỗi bit dữ liệu trong thiết bị nguồn S sẽ được nghịch đảo và chuyển sang thiết bị đích D.

Điều này có nghĩa là các bit trong dữ liệu nguồn trở thành bit 0 trong dữ liệu đích và mỗi bit 0 trong dữ liệu nguồn se thành 1 trong dữ liệu đích. Nếu vùng dữ liệu đích nhỏ hơn dữ liệu nguồn thì chỉ các bit vừa với thiết bị đích mới được xử lý.

2.6. Lệnh BMOV (FNC 15)

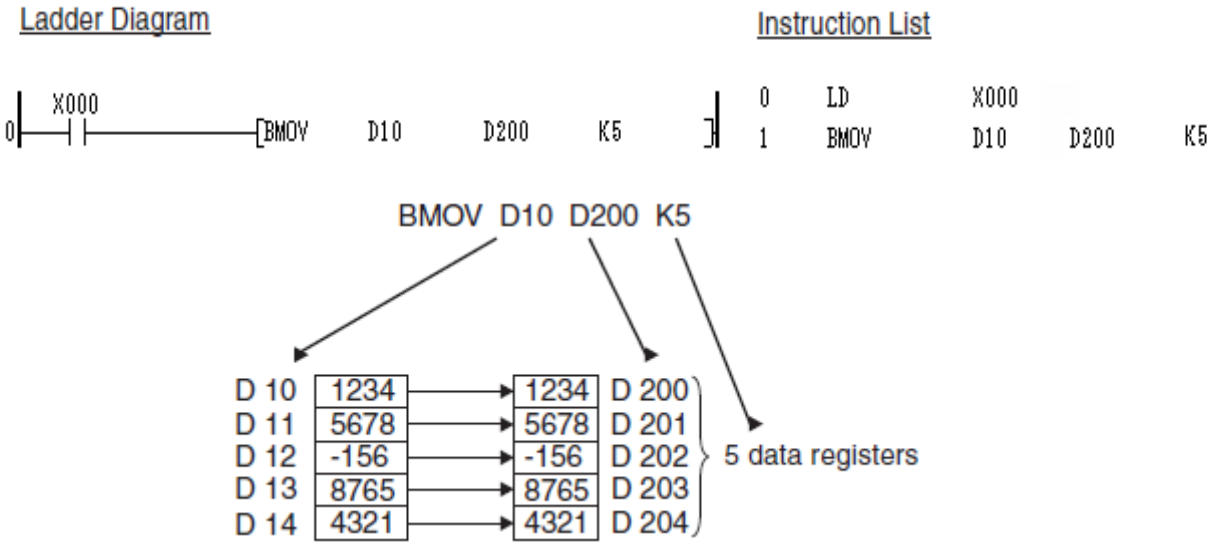
Lệnh	Chức năng	Toán hạng			Số bước
		S	D	n	
BMOV FNC 15 (Block move)	Nhảy tới một vị trí con trỏ đích đã định	KnX, KnY, KnM, KnS, T, C, D, V, Z Thanh ghi tập tin (RAM)	KnY, KnM, KnS, T, C, D, V, Z Thanh ghi tập tin (RAM)	K, H Lưu ý: n ≤ 512	BMOV, BMOVP: 7 bước

Hoạt động:



Sao chép 1 dữ liệu nhiều phần tử liên tiếp từ thiết bị nguồn vào thiết bị đích mới. Dữ liệu nguồn được xác định theo địa chỉ đầu S và số lượng các phần tử của dữ liệu liên tiếp n. Các n phần tử này được chuyển đến thiết bị đích D.

Ví dụ:



2.7. Lệnh FMOV (FNC 16)

Lệnh	Chức năng	Toán hạng			Số bước
		S	D	n	
FMOV FNC 16 (Fill move)	Sao chép 1 dữ liệu đơn đến dãy đích mới	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	K, H Lưu ý: n ≤ 512	FMOV, FMOVP: 7 bước DFMOV, DFMOVP: 13 bước

Hoạt động:

Dữ liệu trong thiết bị nguồn S sẽ được sao chép vào từng thiết bị nằm trong dãy đích.

Dãy thiết bị đích được xác định bằng địa chỉ đầu D và số các phần tử liên tiếp n. nếu số thiết bị đích n vượt quá số vị trí trống của thiết bị đích thì chỉ các thiết bị đích vừa với số vị trí trống trên mới được ghi vào. Lưu ý rằng lệnh này hoạt động 32 bit chỉ áp dụng trên các bộ điều khiển FX có CPU phiên bản 3.07 trở lên và FX_{2C}

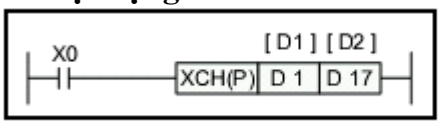
2.8. Lệnh XCH (FNC 17)

Lệnh	Chức năng	Toán hạng		Số bước
		D1	D2	
XCH FNC 17 (Exchange)	Hoán đổi dữ liệu trong thiết bị xác định	KnY, KnM, KnS, T, C, D, V, Z Lưu ý: Khi dùng XCH cho byte (tức M8160 là ON) D1 và D2 phải ở cùng 1 thiết bị nếu không sẽ có lỗi và N8067 sẽ là ON		XCH, XCHP: 5 bước DXCH, DXCHP: 9 bước

Hoạt động 1:

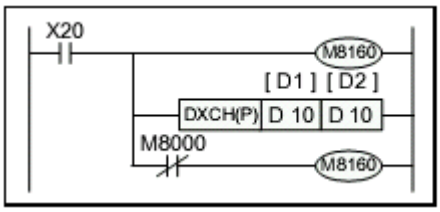
Nội dung của 2 thiết bị đích D1 và D2 sẽ hoán đổi lẫn nhau tức là toàn bộ word được hoán đổi.

Hoạt động 2:



Data register	Before XCH	After XCH
D1	20	530
D17	530	20

Các byte trong mỗi word của thiết bị đích D1 được hoán đổi khi M8160 = ON. Lưu ý chế độ hoạt động byte sẽ duy trì đến khi nó được reset, tức là M8160 là OFF.Ví dụ:

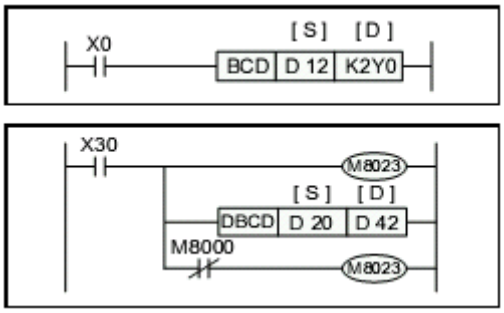


Values are in Hex for clarity		Before DXCH	After DXCH
D10	Byte 1	1FH	8BH
	Byte 1	8BH	1FH
D11	Byte 1	C4H	35H
	Byte 1	35H	C4H

2.9. Lệnh BCD (FNC 18)

Lệnh	Chức năng	Toán hạng		Số bước
		S	D	
BCD FNC 18 (Binary coded decimal)	Chuyển đổi số nhị phân sang BCD hay chuyển đổi dữ liệu dấu chấm động sang dạng khoa học	KnX,KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BCD, BCDP: 5 bước DBCD, DBCDP: 9 bước
		Khi dùng M8023 để chuyển đổi dữ liệu sang dạng khoa học thì chỉ dùng được các thanh ghi 32 bit D.		

Hoạt động:



Dữ liệu nhị phân nguồn S được chuyển đổi thành số BCD tương ứng và lưu ở thiết bị đích D. Nếu số BCD vượt quá dãy cho phép 0 đến 9,999 (hoạt động 16 bit) và 0 đến 99,999,999 (hoạt động 32 bit) sẽ gây ra lỗi. Lệnh này có thể dùng để xuất dữ liệu trực tiếp cho đèn 7 đoạn.

2.10. Lệnh XCH (FNC 17)

Lệnh	Chức năng	Toán hạng		Số bước
		S	D	
BIN FNC 19 (Binary)	Chuyển đổi các số BCD sang nhị phân tương ứng hay chuyển đổi dữ liệu dạng khoa học sang dạng thập phân	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BIN, BINP: 5 bước DBIN, DBINP: 9 bước
		Khi dùng M8023 để chuyển đổi dữ liệu sang dạng khoa học thì chỉ dùng được các thanh ghi 32 bit D.		

Hoạt động:

Dữ liệu nguồn BCD chuyển thành dạng nhị phân tương ứng và lưu ở thiết bị đích D. Nếu dữ liệu nguồn không thuộc dạng BCD sẽ gây ra lỗi. Lệnh này có thể dùng để đọc trực tiếp dữ liệu từ bộ chọn nhấn.

III. Nhóm lệnh về xử lý số học và logic _ Các hàm từ 20 đến 29:

Nội dung:

ADD	-	Addition	FNC 20
SUB	-	Subtraction	FNC 21
MUL	-	Multiplication	FNC 22
DIV	-	Division	FNC 23
INC	-	Increment	FNC 24
DEC	-	Decrement	FNC 25
WAND	-	Word AND	FNC 26
WOR	-	Word OR	FNC 27
WXOR	-	Word Exclusive OR	FNC 28
NEG	-	Negation	FNC 29

Các ký hiệu:

D	_	Toán hạng đích
S	_	Toán hạng nguồn
m, n	_	Số ký hiệu cho thiết bị, nhóm bit hay hằng số

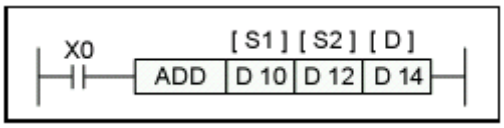
Sự hiệu chỉnh lệnh:

☆☆☆	Lệnh hoạt động ở chế độ 16 bit, trong đó ☆☆☆ chỉ tên lệnh.
☆☆☆P	Lệnh đã hiệu chỉnh để dùng tác vụ xung 16 bit.
D☆☆☆	Lệnh đã hiệu chỉnh để hoạt động trong tác vụ 32 bit.
D☆☆☆P	Lệnh đã hiệu chỉnh để dùng tác vụ xung 32 bit.

3.1. Lệnh ADD (FNC 20)

<table>
<tr><th rowspan="2">Lệnh</th><th rowspan="2">Chức năng</th><th colspan="3">Toán hạng</th><th rowspan="2">Số bước</th></tr>
<tr><th>S1</th><th>S2</th><th>D</th></tr>
<tr><td rowspan="2">ADD
FNC 20
(Addition)</td><td rowspan="2">Cộng 2 dữ liệu nguồn, kết quả lưu ở thiết bị đích</td><td colspan="2">K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z</td><td>KnY, KnM, KnS, T, C, D, V, Z</td><td rowspan="2">ADD, ADDP: 7 bước
DADD, DADDP: 13 bước</td></tr>
<tr><td colspan="3">Khi dùng M8023 để cộng dữ liệu dạng dấu chấm động thì chỉ các thanh ghi dữ liệu 32 bit D hay hằng số K, H mới dùng được.</td></tr>
</table>

Hoạt động:

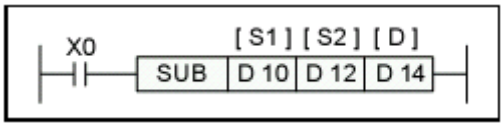


Dữ liệu chứa trong thiết bị nguồn (S1, S2) được cộng lại và tổng của nó lưu ở thiết bị đích D.

3.2. Lệnh SUB (FNC 21)

Lệnh	Chức năng	Toán hạng			Số bước
		S1	S2	D	
SUB FNC 21 (Subtract)	Trừ 2 dữ liệu nguồn, kết quả lưu ở thiết bị đích	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	SUB, SUBP: 7 bước DSUB, DSUBP: 13 bước
		Khi dùng M8023 để trừ dữ liệu dạng dấu chấm động thì chỉ các thanh ghi dữ liệu 32 bit D hay hằng số K, H mới dùng được.			

Hoạt động:

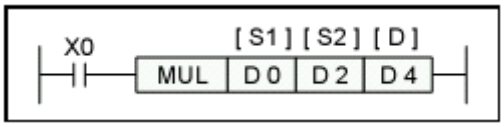


Dữ liệu nguồn S2 được trừ đi giá trị của nguồn S1. Kết quả được lưu trong thiết bị đích D.

3.3. Lệnh MUL (FNC 22)

Lệnh	Chức năng	Toán hạng			Số bước
		S1	S2	D	
MUL FNC 22 (Multiplica -tion)	Nhân 2 dữ liệu nguồn, kết quả lưu ở thiết bị đích	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z(V) Lưu ý: Z(V) không dùng cho hoạt động 32 bit	MUL, MULP: 7 bước DMUL, DMULP: 13 bước
		Khi dùng M8023 để nhân dữ liệu dạng dấu chấm động thì chỉ các thanh ghi dữ liệu 32 bit D hay hằng số K, H mới dùng được.			

Hoạt động:

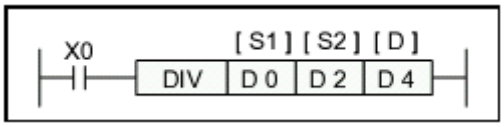


Nội dung của 2 thiết bị nguồn (S1, S2) được nhân với nhau và kết quả được lưu vào thiết bị đích D.

3.4. Lệnh DIV (FNC 23)

Lệnh	Chức năng	Toán hạng			Số bước
		S1	S2	D	
DIV FNC 23 (Division)	Chia 2 dữ liệu nguồn, kết quả lưu ở thiết bị đích	K, H, KnX, KnY, KnM, KnS,T, C, D, V, Z		KnY, KnM, KnS, T, C, D, Z(V)	DIV, DIVP: 7 bước DDIV, DDIVP: 13 bước
				Lưu ý: Z(V) không dùng cho hoạt động 32 bit	
		Khi dùng M8023 để chia dữ liệu dạng dấu chấm động thì chỉ các thanh ghi dữ liệu 32 bit D hay hằng số K, H mới dùng được.			

Hoạt động:



Nguồn S1 được chia cho nguồn S2. Kết quả được lưu vào thiết bị đích D. Các qui tắc về đại số được áp dụng cho trường hợp này.

3.5. Lệnh INC (FNC 24)

Lệnh	Chức năng	Toán hạng	Số bước
		D	
INC FNC 24 (Increment)	Thiết bị đích được tăng lên 1 khi dùng lệnh này	KnY, KnM, KnS, T, C, D, V, Z Qui tắc V, Z áp dụng cho hoạt động 32 bit	INC, INCP: 3 bước DINC, DINCP: 5 bước

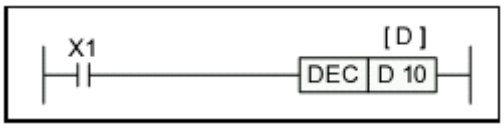
Hoạt động:

Mỗi khi thực hiện lệnh này thì giá trị hiện hành của thiết bị đích sẽ tăng lên 1. trường hợp hoạt động 16 bit khi đạt đến +32,767 thì lần tăng tiếp theo sẽ ghi -32,768 vào thiết bị đích. Trường hợp hoạt động 32 bit khi đạt đến +2,147,483,647 thì lần tăng tiếp theo sẽ ghi -2,147,483,648 vào thiết bị đích. Trong cả 2 trường hợp không có cờ báo hiệu có sự thay đổi trên.

3.6. Lệnh DEC (FNC 25)

Lệnh	Chức năng	Toán hạng	Số bước
		D	
DEC FNC 25 (Decrement)	Thiết bị đích được giảm xuống 1 khi dùng lệnh này	KnY, KnM, KnS, T, C, D, V, Z Qui tắc V, Z chuẩn áp dụng cho hoạt động 32 bit	DEC, DECP: 3 bước DDEC, DDECP: 5 bước

Hoạt động:



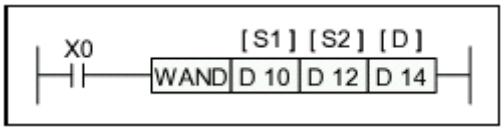
Mỗi khi thực hiện lệnh này thì giá trị hiện hành của thiết bị đích sẽ giảm xuống 1. trường hợp hoạt động 16 bit khi đạt đến -32,768 thì lần giảm tiếp theo sẽ ghi +32,767 vào thiết bị đích. Trường hợp hoạt động 32 bit khi đạt đến -2,147,483,648 thì lần giảm tiếp theo sẽ ghi +2,147,483,647 vào thiết bị đích.

Trong cả 2 trường hợp không có cờ báo hiệu có sự thay đổi trên.

3.7. Lệnh WAND (FNC 26)

Lệnh	Chức năng	Toán hạng			Số bước
		S1	S2	D	
WAND FNC 26 (Logical word AND)	Thực hiện logic AND trên 2 thiết bị nguồn – kết quả lưu trong thiết bị đích	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WAND, WANDP: 7 bước DAND, DANDP: 13 bước

Hoạt động:



Áp dụng logic AND chuỗi bit của 2 thiết bị nguồn S1 và S2 (nội dung của S2 được XOR với nội dụng của S1), kết quả được lưu vào thiết bị đích D. Qui tắc sau được dùng để xác định hoạt động logic AND. Thực hiện AND từng bit trong các thiết bị nguồn:

(S1) bit n WAND (S2) bit n = (D) bit n

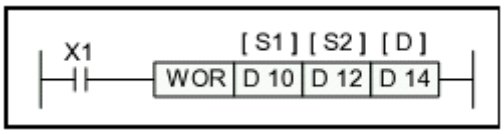
1 WAND 1 = 1 0 WAND 1 = 0

1 WAND 0 = 0 0 WAND 0 = 0

3.8. Lệnh WOR (FNC 27)

Lệnh	Chức năng	Toán hạng			Số bước
		S1	S2	D	
WOR FNC 27 (Logical word OR)	Thực hiện logic OR trên 2 thiết bị nguồn – kết quả lưu trong thiết bị đích	K,H, KnX,KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	WOR, WORP: 7 bước DOR, DORP: 13 bước

Hoạt động:



Áp dụng logic OR chuỗi bit của 2 thiết bị nguồn S1 và S2 (nội dung của S2 được OR với nội dụng của S1), kết quả được lưu vào thiết bị đích D. Qui tắc sau được dùng để xác định hoạt động logic OR. Thực hiện OR từng bit trong các thiết bị nguồn:

(S1) bit n WOR (S2) bit n = (D) bit n

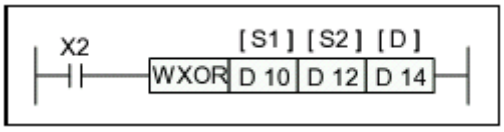
1 WOR 1 = 1 0 WOR 1 = 1

1 WOR 0 = 1 0 WOR 0 = 0

3.9. Lệnh WXOR (FNC 28)

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
WXOR FNC 28 (Logical exclusive OR)	A logical XOR 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	WXOR, WXORP: 7 steps DXOR,DXORP 13 steps

Hoạt động:



Áp dụng logic XOR chuỗi bit của 2 thiết bị nguồn S1 và S2 (nội dung của S2 được XOR với nội dụng của S1), kết quả được lưu vào thiết bị đích D. Qui tắc sau được dùng để xác định hoạt động logic XOR. Thực hiện XOR từng bit trong các thiết bị nguồn:

(S1) bit n WXOR (S2) bit n = (D) bit n

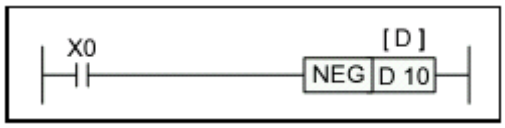
1 WXOR 1 = 0 0 WXOR 1 = 1

1 WXOR 0 = 1 0 WXOR 0 = 0

3.10. Lệnh NEG (FNC 29)

Lệnh	Chức năng	Toán hạng	Số bước
		D	
NEG FNC 29 (Negation)	Thực hiện đổi dấu nội dung thiết bị đích	KnY, KnM, KnS, T, C, D, V, Z	NEG, NEGP: 3 bước DNEG, DNEGP: 5 bước

Hoạt động:



Chuỗi bit của thiết bị đích bị nghịch đảo. Nghĩa là bit "1" trở thành "0" và ngược lại. khi hoàn tất, chuỗi bit đó sẽ được cộng thêm vào số nhị phân 1, nói cách khác là đổi dấu nội dung của thiết bị đích, thí dụ số dương sẽ trở thành số âm và ngược lại.