



Ch 5: Các nhóm lệnh ứng dụng

Ngoài các lệnh cơ bản, PLC thông thường được thiết kế để có thêm các nhóm lệnh ứng dụng, thí dụ như có nhóm lệnh chuyên tác động điều khiển lưu trình, có nhóm lệnh chuyên thực hiện xử lý số học ...

Không phải tất cả các nhóm lệnh đều có thể sử dụng trên tất cả các họ PLC. Do đó cần phải lưu ý khi sử dụng nhóm lệnh trên 1 PLC nào đó.

Đối với PLC họ FX (Mitsubishi), có khoảng 200 lệnh được phân vào 16 nhóm lệnh.



5.1. Program flow

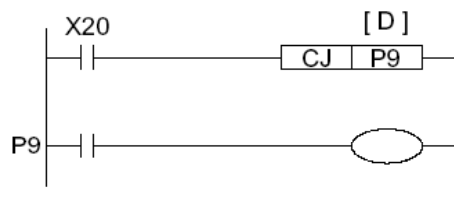
CJ	Conditional jump	FNC 00
CALL	Call Subroutine	FNC 01
SRET	Subroutine Return	FNC 02
IRET	Interrupt Return	FNC 03
EI	Enable Interrupt	FNC 04
DI	Disable Interrupt	FNC 05
FEND	First End	FNC 06
WDT	Watchdog Timer	FNC 07
FOR	Start of a For/Next Loop	FNC 08
NEXT	End a For/Next Loop	FNC 09



5.1. Program flow

- **Lệnh CJ:** nhảy tới vị trí con trỏ đích xác định.

Mnemonic	Function	Operands	Program steps
		D	
CJ FNC 00 (Conditional Jump)	Jumps to the identified pointer position	Valid pointers from the range 0 to 63	CJ, CJP: 3 steps Jump pointer PPP: 1 step



- Khi lệnh CJ được kích hoạt thì con trỏ lệnh nhảy đến vị trí xác định trong chương trình, bỏ qua một số bước chương trình (không xử lý) do đó làm tăng tốc độ quét chương trình.

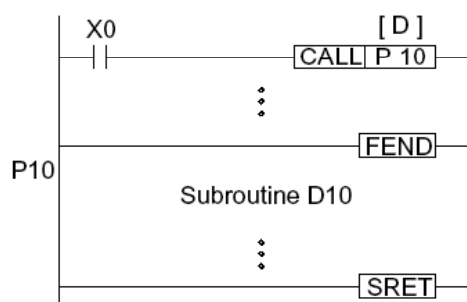
- Mỗi con trỏ đích phải có duy nhất một con số. Dùng con trỏ P63 tương đương với việc nhảy đến lệnh END.



5.1. Program flow

- **Lệnh CALL:** Gọi chương trình con.

Mnemonic	Function	Operands	Program steps
		D	
CALL FNC 01 (Call sub-routine)	Executes the subroutine program starting at the identified pointer position	Valid pointers from the range 0 to 62 Nest levels: 5 including the initial CALL	CALL, CALLP: 3 step Subroutine pointer PPP: 1 steps

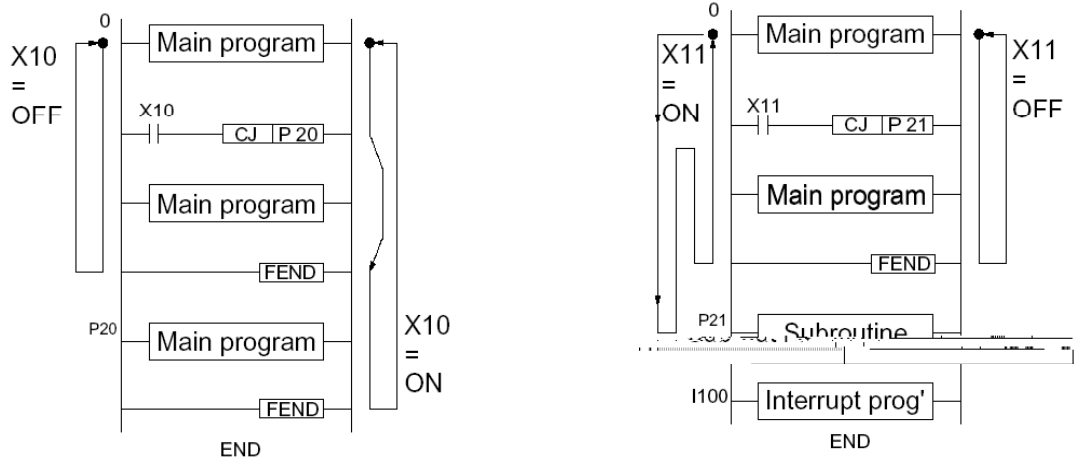


- Khi một chương trình con được gọi, điều khiển được chuyển từ chương trình chính vào chương trình con. Khi gặp lệnh SRET, chương trình con kết thúc và điều khiển được chuyển về lệnh kế tiếp sau lệnh gọi chương trình con trong chương trình chính.

- Con trỏ của lệnh CJ và lệnh CALL không được trùng nhau.

5.1. Program flow

- **Lệnh FEND:** Khai báo phần kết thúc đoạn chương trình chính.



- Lệnh FEND dùng để báo kết thúc đoạn chương trình chính và bắt đầu đoạn chương trình con. Cấu trúc thường dùng: CJ-P-FEND, CALL-P-SRET.
- Nhiều lệnh FEND có thể được sử dụng đồng thời. Lệnh FEND không được phép đặt sau lệnh END.

5.1. Program flow

- **Lệnh FOR, NEXT:** Thực hiện vòng lặp với số lần xác định.

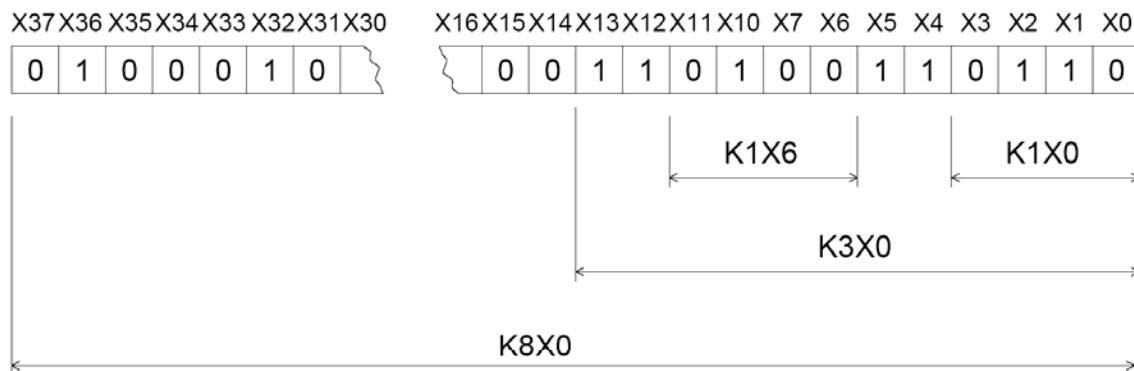
Mnemonic	Function	Operands	Program steps
		S	
FOR FNC 08 (Start of a FOR-NEXT loop)	Identifies the start position and the number of repeats for the loop	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	FOR: 3 step
NEXT FNC 09 (End of a FOR-NEXT loop)	Identifies the end position for the loop	N/A Note: The FOR-NEXT loop can be nested for 5 levels, i.e. 5 FOR-NEXT loops are programmed within each other.	NEXT: 1 step

- Giá trị toán hạng S phải nằm trong phạm vi từ 1 đến 32767. Những giá trị khác của S sẽ tự động làm S được gán bằng 1, nghĩa là vòng lặp FOR-NEXT được thực hiện 1 lần.



5.1. Program flow

- Lưu ý: các bit thường được nhóm lại thành từng đơn vị 4 bit

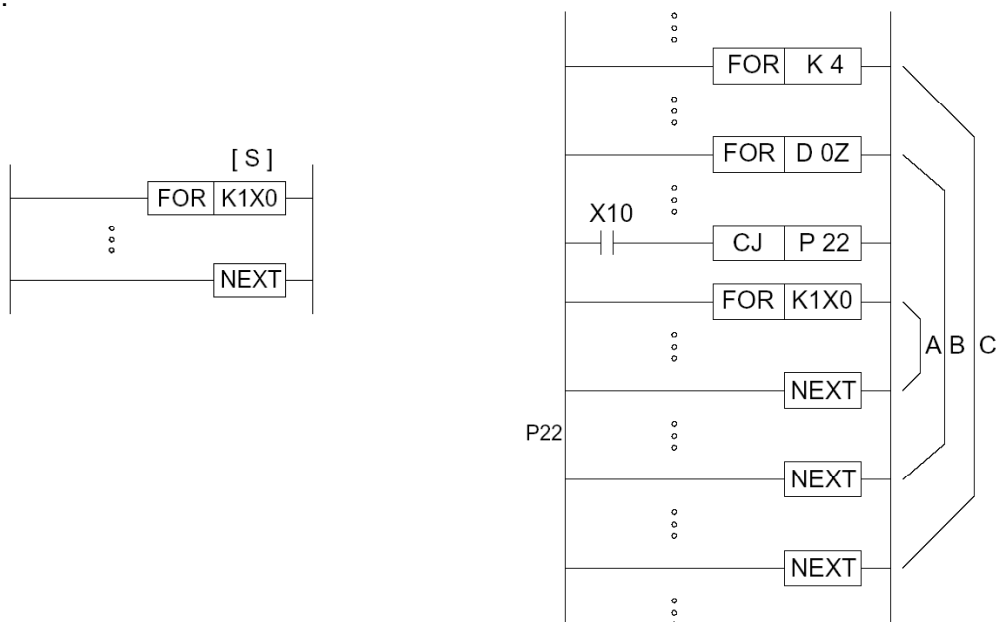


- K1X0 : từ X0 đến X3 (dữ liệu 4 bit bắt đầu từ X0)
- K1X6 : từ X6 đến X11 (dữ liệu 4 bit bắt đầu từ X6)
- K3X0 : từ X0 đến X13 (dữ liệu 12 bit bắt đầu từ X0)
- K8X0 : từ X0 đến X37 (dữ liệu 32 bit bắt đầu từ X0)



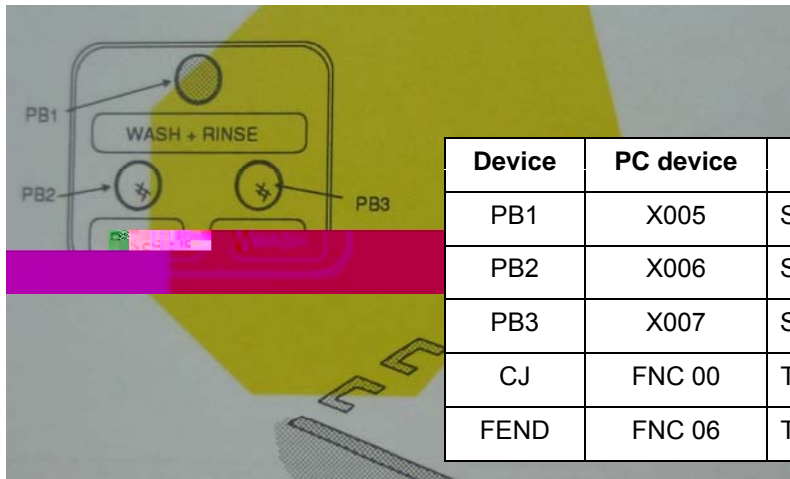
5.1. Program flow

- Cặp lệnh FOR và NEXT luôn đi đôi với nhau và được phép sử dụng lồng vào nhau đến 5 cấp.





Applications – Using conditional jumps



Device	PC device	Description
PB1	X005	Select wash and rinse
PB2	X006	Select rinse only
PB3	X007	Select wash only
CJ	FNC 00	The CJ applied instruction
FEND	FNC 06	The FEND applied instruction

- Tại trạm rửa xe áp lực, có 3 dịch vụ: rinse only / wash only / wash and rinse.
- Sử dụng 2 đoạn chương trình con tương ứng cho wash và rinse sẽ giúp cho kích thước chương trình giảm đi phân nửa.



5.2. Move and compare

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



5.2. Move and compare

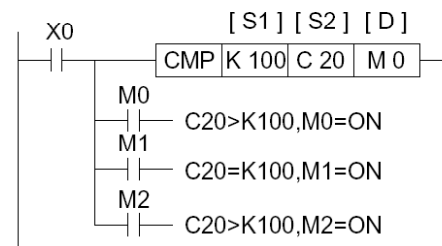
- **Lệnh CMP:** Thực hiện so sánh giữa hai giá trị.

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
CMP FNC 10 (Compare)	Compares two data values - results of <, = and > are given.	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		Y, M, S Note: 3 consecutive devices are used.	CMP, CMPP: 7 steps DCMP, DCMPP: 13 steps

- Dữ liệu [S₁] được so sánh với dữ liệu [S₂]. Kết quả được phản ánh trên 3 thiết bị logic (bắt đầu từ [D]).

- [S₂] < [S₁] ⇒ [D] = 1
- [S₂] = [S₁] ⇒ [D₊₁] = 1
- [S₂] > [S₁] ⇒ [D₊₂] = 1

- Dữ liệu [D] vẫn được duy trì ngay cả khi lệnh CMP không còn được thực hiện.



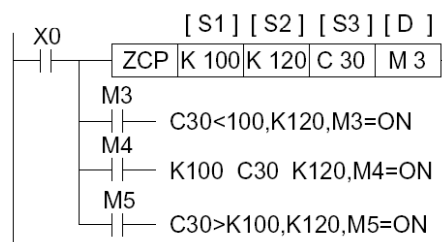
5.2. Move and compare

- **Lệnh ZCP:** Thực hiện so sánh giữa một giá trị với 1 dãy giá trị.

Mnemonic	Function	Operands			Program steps
		S1	S2	S3	
ZCP FNC 11 (Zone compare)	Compares a data value against a data range - results of <, = and > are given.	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		S3	Y, M, S Note: 3 consecutive devices are used.
		Note: S1 should be less than S2			ZCP, ZCPP: 9 steps DZCP, DZCPP: 17 steps

- Dữ liệu [S₃] được so sánh với dãy dữ liệu từ [S₁] đến [S₂]. Kết quả được phản ánh trên 3 thiết bị logic (bắt đầu từ [D]).

- [S₃] < [S₁] & [S₂] ⇒ [D] = 1
- [S₃] nằm từ [S₁] đến [S₂] ⇒ [D₊₁] = 1
- [S₃] > [S₁] & [S₂] ⇒ [D₊₂] = 1

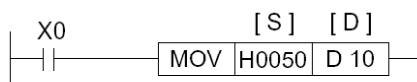




5.2. Move and compare

- **Lệnh MOV:** Gán giá trị từ vùng nhớ này đến vùng nhớ khác.

Mnemonic	Function	Operands		Program steps
		S	D	
MOV FNC 12 (Move)	Moves data from one storage area to a new storage area	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	MOV, MOVP: 5 steps DMOV, DMOVP: 9 steps



- Các hoạt động về sao chép vùng nhớ được dùng để tăng cường các chức năng sẵn có, ví dụ như cho phép thay đổi các giá trị xác lập cho bộ định thì hay bộ đếm. Các loại ứng dụng này rất phổ biến, cho phép người điều khiển nhập các giá trị tham số khác nhau trước khi hoặc trong lúc PLC hoạt động.

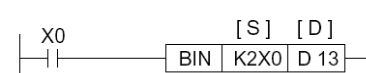
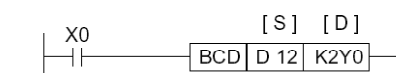


5.2. Move and compare

- **Lệnh BCD:** Chuyển đổi số nhị phân sang dạng BCD.
- **Lệnh BIN:** Chuyển đổi số BCD sang dạng nhị phân tương ứng.

Mnemonic	Function	Operands		Program steps
		S	D	
BCD FNC 18 (Binary coded decimal)	Converts binary numbers to BCD equivalents / Converts floating point data to scientific format	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BCD, BCDP: 5 steps DBCD, DBCDP: 9 steps

Mnemonic	Function	Operands		Program steps
		S	D	
BIN FNC 19 (Binary)	Converts BCD numbers to their binary equivalent / Converts scientific format data to floating point format	KnX, KnY, KnM, KnS, T, C, D, V, Z	KnY, KnM, KnS, T, C, D, V, Z	BIN, BINP: 5 steps DBIN, DBINP: 9 steps





5.3. Arithmetic and logical operations

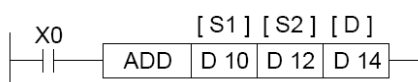
ADD	Addition	FNC 20
SUB	Subtraction	FNC 21
MUL	Multiplication	FNC 22
DIV	Division	FNC 23
INC	Increment	FNC 24
DEC	Decrement	FNC 25
WAND	Word AND	FNC 26
WOR	Word OR	FNC 27
WXOR	Word Exclusive OR	FNC 28
NEG	Negation	FNC 29



5.3. Arithmetic and logical operations

- **Lệnh ADD:** Cộng hai giá trị.

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
ADD FNC 20 (Addition)	The value of the two source devices is added and the result stored in the destination device	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z		KnY, KnM, KnS, T, C, D, V, Z	ADD, ADDP: 7 steps DADD, DADDP: 13 steps
		When using M8023 to add floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used. See page 4-46 for more details regarding floating point format.			



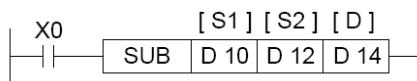
- Phép cộng được thực hiện cho các số có dấu.
- Nếu giá trị tính được vượt quá kích thước ở toán hạng đích, thì giá trị thực sẽ bị cắt bớt đi để phù hợp với kích thước toán hạng đích.
- Khi giá trị tính được bằng không thì M8020 = 1



5.3. Arithmetic and logical operations

- **Lệnh SUB:** Tính sai lệch giữa hai giá trị.

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
SUB FNC 21 (Subtract)	One source device is subtracted from the other - 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, V, Z	KnY, KnM, KnS, T, C, D, V, Z	SUB, SUBP: 7steps DSUB, DSUBP: 13 steps
		When using M8023 to subtract floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used. See page 4-46 for more details regarding floating point format.			



$$- [D] = [S_2] - [S_1]$$

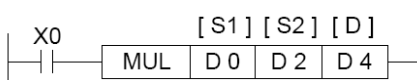
- Các điểm lưu ý cho phép trừ cũng giống như bên phép cộng.



5.3. Arithmetic and logical operations

- **Lệnh MUL:** Nhân hai giá trị.

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
MUL FNC 22 (Multiplication)	Multiplies the two source devices together 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)	KnY, KnM, KnS, T, C, D, Z(V)	MUL, MULP: 7steps DMUL, DMULP: 13 steps
		See page 4-46 for more details regarding floating point format.			
		When using M8023 to subtract floating point data, only double word (32 bit) data registers (D) or constants (K/H) may be used.			



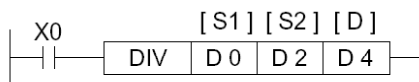
- Với thao tác 16 bit, hai giá trị 16 bit nhân với nhau sẽ cho kết quả là con số 32 bit và kết quả này được lưu vào $[D_{+1}] [D]$.

- Nếu giá trị tính được vượt quá kích thước ở toán hạng đích, thì giá trị thực cũng sẽ bị cắt bớt đi để phù hợp với kích thước toán hạng đích.

5.3. Arithmetic and logical operations

- **Lệnh DIV:** Thực hiện phép chia giữa hai giá trị.

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
DIV FNC 23 (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 DDIV, DDIVP: 13 steps
		See page 4-46 for more details regarding floating point format. 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 NOT be used for 32 bit operation	



- $[S_1] \div [S_2]$

- Với thao tác 16 bit, hai giá trị 16 bit chia cho nhau thì sẽ cho kết quả là con số 16 bit. Sau phép chia, số nguyên được lưu ở [D] và số dư được lưu ở [D+1]. Thí dụ: $51 (D0) \div 10 (D2) = 5 (D4) 1 (D5)$

5.3. Arithmetic and logical operations

- **Lệnh INC:** Tăng nội dung toán hạng 1 đơn vị
- **Lệnh DEC:** Giảm nội dung toán hạng 1 đơn vị

Mnemonic	Function	Operands	Program steps
		D	
INC FNC 24 (Increment) →	The designated device is incremented by 1 on every execution of the instruction	KnY, KnM, KnS, T, C, D, V, Z Standard V,Z rules apply for 32 bit operation <div><div>X0</div><div>[D]</div><div>INC D 10</div></div>	INC,INCP: 3 steps DINC, DINCP: 5 steps

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



5.4. Rotate and shift

ROR	Rotation Right	FNC 30
ROL	Rotation Left	FNC 31
RCR	Rotation Right with Carry	FNC 32
RCL	Rotation Left with Carry	FNC 33
SFTR	(Bit) Shift Right	FNC 34
SFTL	(Bit) Shift Left	FNC 35
WSFR	Word Shift Right	FNC 36
WSFL	Word Shift Left	FNC 37
SFWR	Shift Register Write	FNC 38
SFRD	Shift Register Read	FNC 39

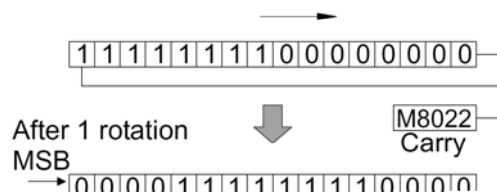


5.4. Rotate and shift

- **Lệnh ROR:** Thực hiện quay chuỗi bit sang phải 'n' vị trí

Mnemonic	Function	Operands		Program steps
		D	n	
ROR FNC 30 (Rotation)	The bit pattern of the destination device is rotated	KnY, KnM, KnS, T, C, D, V, Z Note: 16 bit operation n ≤ 16 32 bit operation n ≤ 32	K, H, <input checked="" type="checkbox"/>	ROR, RORP: 5 steps Note: 16 bit operation n ≤ 16 32 bit operation n ≤ 32

- Khi lệnh này hoạt động, chuỗi bit của toán hạng được quay sang phải n bit. LSB của chuỗi lần lượt trở thành MSB của chuỗi. Trạng thái của bit cuối cùng được quay sẽ được copy vào cờ nhớ M8022 (carry flag).



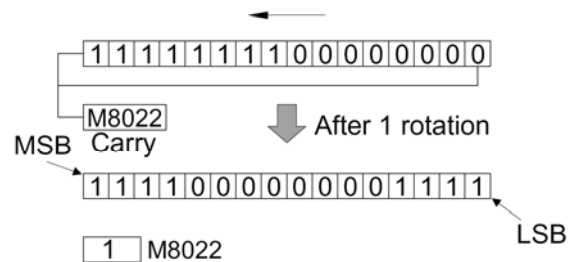
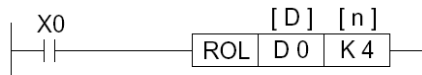


5.4. Rotate and shift

- **Lệnh ROL:** Thực hiện quay chuỗi bit sang trái 'n' vị trí

Mnemonic	Function	Operands		Program steps
		S	D	
ROL FNC 31 (Rotation left) →	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 Note: 16 bit operation Kn= K4, 32 bit operation Kn= K8	K, H, ☒ Note: 16 bit operation $n \leq 16$ 32 bit operation $n \leq 32$	ROL, ROLP: 5 steps DROL, DROLP: 7 steps

- Khi lệnh này hoạt động, chuỗi bit của toán hạng được quay sang trái n bit. MSB của chuỗi lần lượt trở thành LSB của chuỗi. Trạng thái của bit cuối cùng được quay sẽ được copy vào cờ nhớ M8022 (carry flag).

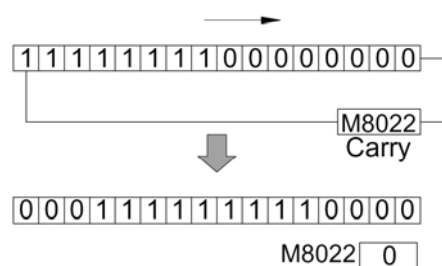
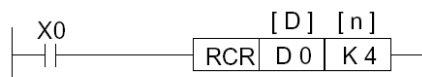


5.4. Rotate and shift

- **Lệnh RCR:** Thực hiện quay chuỗi bit sang phải 'n' vị trí với carry bit

Mnemonic	Function	Operands		Program steps
		D	n	
RCR FNC 32 (Rotation right with carry) →	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 Note: 16 bit operation Kn= K4, 32 bit operation Kn=K8	K, H, ☒ Note: 16 bit operation $n \leq 16$ 32 bit operation $n \leq 32$	RCR, RCRP: 5 steps DRCR, DRCRP: 7 steps

- Khi lệnh này hoạt động, chuỗi bit của toán hạng được quay sang phải n bit. LSB của chuỗi chuyển vào cờ nhớ M8022 và trạng thái trước đó của cờ nhớ M8022 được chuyển vào MSB.





5.4. Rotate and shift

- **Lệnh RCL:** Thực hiện quay chuỗi bit sang trái 'n' vị trí với carry bit

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

- Khi lệnh này hoạt động, chuỗi bit của toán hạng được quay sang trái n bit. MSB của chuỗi chuyển vào cờ nhớ M8022 và trạng thái trước đó của cờ nhớ M8022 được chuyển vào LSB.

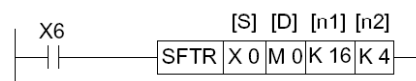


5.4. Rotate and shift

- **Lệnh SFTR:** Thực hiện copy và dịch chuyển chuỗi bit sang phải

Mnemonic	Function	Operands				Program steps
		S	D	n1	n2	
SFTR FNC 34 (Bit shift right) →	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, ☒ Note: FX users: n2 ≤ n1 ≤ 1024 FX0, FX0N users: n2 ≤ n1 ≤ 512		SFTR, SFTRP: 9 steps

- Lệnh này copy [n₂] bit [S] vào [n₁] bit [D]. Mỗi lần thêm vào [n₂] bit [S], dữ liệu hiện có của [D] sẽ được dịch sang phải [n₂] bit.



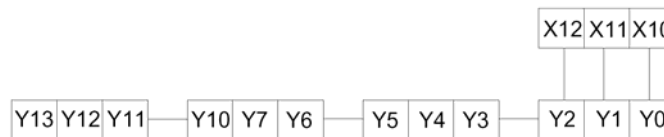
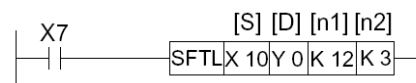


5.4. Rotate and shift

- **Lệnh SFTL:** Thực hiện copy và dịch chuyển chuỗi bit sang trái

Mnemonic	Function	Operands				Program steps
		S	D	n1	n2	
SFTL FNC 35 (Bit shift left) →	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, ☒ Note: FX users: $n2 \leq n1 \leq 1024$ FX ₀ , FX _{0N} users: $n2 \leq n1 \leq 512$		SFTL, SFTLP: 9steps

- Lệnh này copy $[n_2]$ bit $[S]$ vào $[n_1]$ bit $[D]$.
Mỗi lần thêm vào $[n_2]$ bit $[S]$, dữ liệu hiện có của $[D]$ sẽ được dịch sang trái $[n_2]$ bit.



5.5. Data operation

ZRST	Zone Reset	FNC 40
DECO	Decode	FNC 41
ENCO	Encode	FNC 42
SUM	The Sum Of Active Bits	FNC 43
BON	Check Specified Bit Status	FNC 44
MEAN	Mean	FNC 45
ANS	(Timed) Annunciator Set	FNC 46
ANR	Annunciator Reset	FNC 47
SQR	Square Root	FNC 48
FLT	Float, (Floating Point)	FNC 49



5.5. Data operation

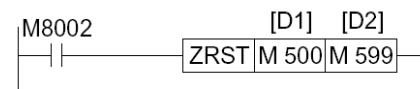
- **Lệnh ZRST:** Thực hiện reset một dãy các thiết bị

Mnemonic	Function	Operands		Program steps
		D1	D2	
ZRST FNC 40 (Zone Reset)	Used to reset a range of like devices in one operation	Y, M, S, T, C, D Note: D1 must be less than or equal (\leq) to D2. Standard and High speed counters cannot be mixed.		ZRST, ZRSTP: 5 steps

- Lệnh này reset một dãy thiết bị từ $[D_1]$ đến $[D_2]$. Đối với thiết bị dữ liệu, thì giá trị được đặt bằng 0; còn với thiết bị logic thì trạng thái của nó là FALSE (hay OFF, LOW, 0 ...).

- Phân toán hạng phải chứa cùng loại thiết bị.

- Nếu $[D_1] > [D_2]$ thì chỉ có $[D_1]$ được reset.



5.6. High speed processing

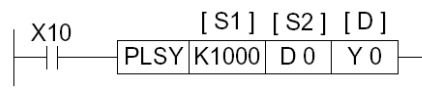
REF	Refresh	FNC 50
REFF	Refresh and filter adjust	FNC 51
MTR	Input matrix	FNC 52
HSCS	High speed counter set	FNC 53
HSCR	High speed counter reset	FNC 54
HSZ	High speed counter zone compare	FNC 55
SPD	Speed detect	FNC 56
PLSY	Pulse Y output	FNC 57
PWM	Pulse width modulation	FNC 58
PLSR	Ramp Pulse output	FNC 59

5.6. High speed processing

- **Lệnh PLSY:** Tạo ra một dãy xung với số xung và tần số xác định

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
PLSY FNC 57 (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 Note: FX _{0(S)} /FX _{0N} users: Y000 only ☒. any YPPP. FX _{2N(C)} users: Y000 or Y001 only ☒.	PLSY: 7 steps DPLSY: 13 steps

- Số xung [S₂] được tạo ở thiết bị [D] với tần số [S₁] (Hz, 50% ON và 50% OFF).
- Cờ M8029 sẽ ON khi dãy xung được tạo xong và tự động OFF khi lệnh không được kích hoạt.
- Nếu K0 được sử dụng trong PLSY, thì xung sẽ được tạo ra liên tục cho đến khi ngừng kích hoạt lệnh.



© C.B. Pham

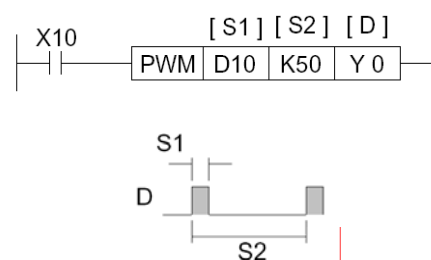
5-31

5.6. High speed processing

- **Lệnh PMW:** Tạo ra một dãy xung được điều chế độ rộng

Mnemonic	Function	Operands			Program steps
		S1	S2	D	
PWM FNC 58 (Pulse width modulation)	Generates a pulse train with defined pulse characteristics	K, H, KnX, KnY, KnM, KnS, T, C, D, V, Z Note: S1 S2		Y Note: FX _{0(S)} /FX _{0N} users: Y001 only ☒. FX users: any YPPP. FX _{2N(C)} users: Y000 or Y001 only ☒	PWM: 7 steps

- Một dãy xung liên tục được tạo ở thiết bị [D] khi lệnh này làm việc.
- Thời gian một chu kỳ được xác định bởi [S₂] (msec).
- Thời gian để tín hiệu ở mức cao được xác định bởi [S₁] (msec).



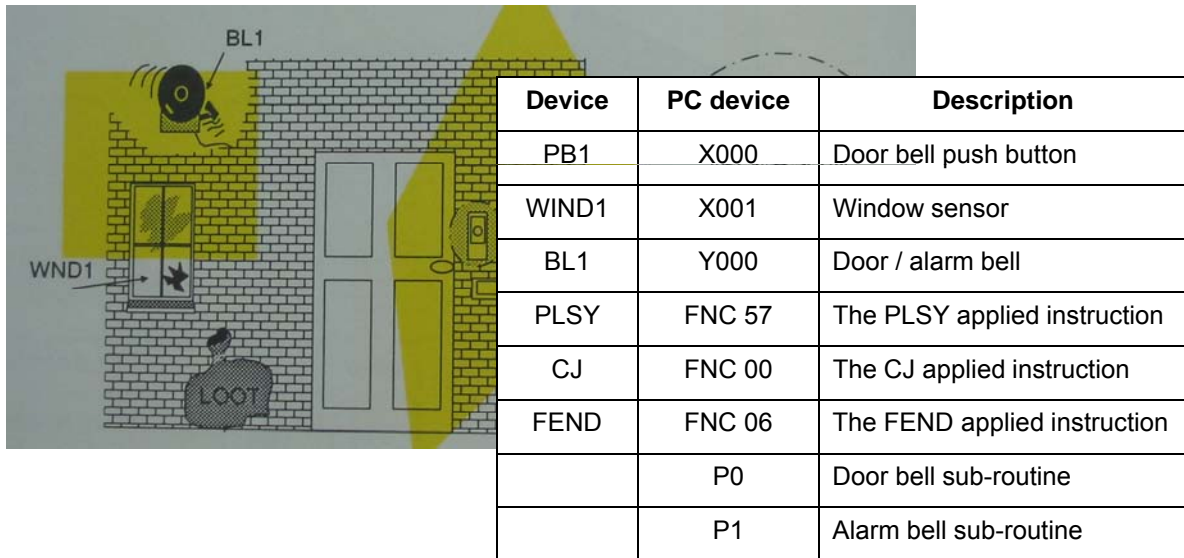
© C.B. Pham

Bộ điều khiển lập trình

5-32



Applications – Using multiple PLSY instructions



- Chuông BL1 có 2 hiệu ứng tương ứng với tín hiệu từ PB1 và WIND1.
- Khi có khách nhấn nút, chuông kêu 2 tiếng ngắn. Khi báo động, chuông kêu liên tục.



5.7. Handy instruction

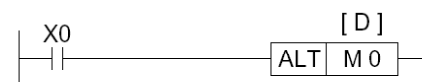
IST	Initial State	FNC 60
SER	Search	FNC 61
ABSD	Absolute Drum	FNC 62
INCD	Incremental Drum	FNC 63
TTMR	Teaching Timer	FNC 64
STMR	Special Timer Definable	FNC 65
ALT	Alternate State	FNC 66
RAMP	Ramp Variable Value	FNC 67
ROTC	Rotary Table Control	FNC 68
SORT	Sort Data	FNC 69

5.7. Handy instruction

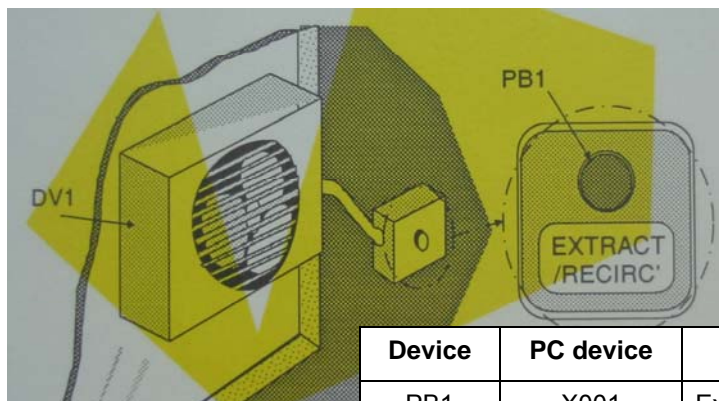
- **Lệnh ALT:** Thực hiện đảo trạng thái hiện tại

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

- Thiết bị [D] bị đảo trạng thái mỗi lần lệnh ALT được thực hiện. Điều này xảy ra ở mỗi lần quét chương trình trừ khi lệnh xung được sử dụng.
- Lệnh ALT thường được sử dụng để chuyển đổi giữa hai chế độ hoạt động (e.g. start and stop)



Applications – Selection between 2 modes using ALT



Device	PC device	Description
PB1	X001	Extraction / circulation push button
DV1	Y002	Move valve to circulation setting
	Y003	Move valve to extraction setting
ALT	FNC 66	The ALT applied instruction

- Quạt làm việc ở hai chế độ: hút / thổi.
- Mỗi lần nhấn nút PB1, quạt sẽ chuyển từ chế độ hiện tại sang chế độ còn lại.



5.8. External FX I/O devices

TKY	Ten Key Input	FNC 70
HKY	Hexadecimal Input	FNC 71
DSW	Digital Switch (Thumbwheel input)	FNC 72
SEGD	Seven Segment Decoder	FNC 73
SEGL	Seven Segment With Latch	FNC 74
ARWS	Arrow Switch	FNC 75
ASC	ASCII Code	FNC 76
PR	'Print' To A Display	FNC 77
FROM	Read From A Special Func. Block	FNC 78
TO	Write To A Special Function Block	FNC 79



5.9. External FX serial devices

RS	RS Communications	FNC 80
PRUN	FX2-40AP Parallel Run	FNC 81
ASCI	Hexadecimal to ASCII	FNC 82
HEX	ASCII to Hexadecimal	FNC 83
CCD	Check Code	FNC 84
VRRD	FX-8AV Volume Read	FNC 85
VRSC	FX-8AV Volume Scale	FNC 86
☆☆☆	Not Available	FNC 87
PID	PID Control Loop	FNC 88
☆☆☆	Not Available	FNC 89



5.10. External F2 units

MNET	F-16NP, Melsec Net Mini	FNC 90
ANRD	F2-6A, Analog Read	FNC 91
ANWR	F2-6A, Analog Write	FNC 92
RMST	F2-32RM, RM Start	FNC 93
RMWR	F2-32RM, RM Write	FNC 94
RMRD	F2-32RM, RM Read	FNC 95
RMMN	F2-32RM, RM Monitor	FNC 96
BLK	F2-30GM, Block	FNC 97
MCDE	F2-30GM, Machine Code	FNC 98
☆☆☆	Not Available	FNC 99



5.11. Floating point 1 & 2

Floating Point 1

ECMP	Float Compare	FNC 110
EZCP	Float Zone Compare	FNC 111
☆☆☆	Not Available	FNC 112 to 117
EBCD	Float to Scientific	FNC 118
EBIN	Scientific to Float	FNC 119



5.11. Floating point 1 & 2

Floating Point 2

EADD	Float Add	FNC 120
ESUB	Float Subtract	FNC 121
EMUL	Float Multiplication	FNC 122
EDIV	Float Division	FNC 123
☆☆☆	Not Available	FNC 124 to 126
ESQR	Float Square Root	FNC 127
PPP	Not Available	FNC 128
INT	Float to Integer	FNC 129



5.12. Trigonometry

Floating Point 3

SIN	Sine	FNC 130
COS	Cosine	FNC 131
TAN	Tangent	FNC 132
☆☆☆	Not Available	FNC 133 to 139



5.13. Data operations 2

☆☆☆	Not Available	FNC 140 to 146
SWAP	Float to Scientific	FNC 147
☆☆☆	Not Available	FNC 148 to 149



5.14. Real time clock control

TCMP	Time Compare	FNC 160
TZCP	Time Zone Compare	FNC 161
TADD	Time Add	FNC 162
TSUB	Time Subtract	FNC 163
☆☆☆	Not Available	FNC 164 to 165
TRD	Read RTC data	FNC 166
TWR	Set RTC data	FNC 167
☆☆☆	Not Available	FNC 168 to 169



5.15. Gray codes

GRY	Decimal to Gray Code	FNC 170
GBIN	Gray Code to Decimal	FNC 171
☆☆☆	Not Available	FNC 172 to 177



5.16. In-line comparisons

LD□	LoaD compare	FNC 224 to 230
AND□	AND compare	FNC 232 to 238
OR□	OR compare	FNC 240 to 246