

ĐHBK Tp HCM-Khoa Đ-ĐT
 BMĐT
 GVPT: Hồ Trung Mỹ
 Môn học: Vi Xử Lý

Z80

Tập lệnh

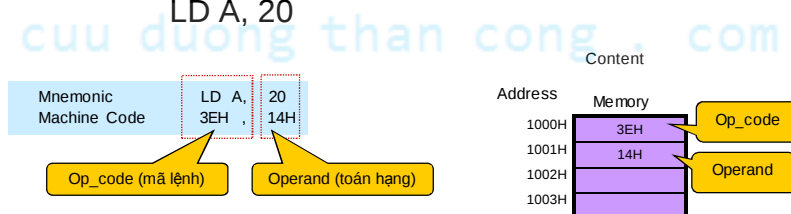
cuu duong than cong . com

Hợp ngữ và Mã máy

- Ngôn ngữ gợi nhớ (Mnemonic language) = hợp ngữ (assembly language)
- Ngôn ngữ máy (Machine language) = mã máy (machine code)

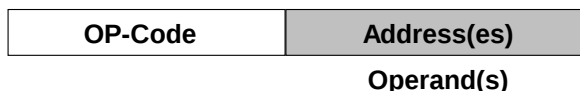
• TD: org 1000h

LD A, 20



Cấu trúc lệnh

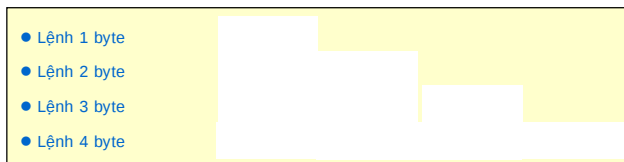
Mọi lệnh gồm có mã lệnh và nếu cần, 1 phần địa chỉ của (các) toán hạng.



OP-Code của Z80 thường thì 1 Byte (ngoại trừ trong các tác vụ dùng các thanh ghi chỉ số thì mã lệnh chiếm 2 byte)

Số địa chỉ:	0	R1 op R2	→	R3
	1	(A) op R	→	R
	2	(A1) op (A2)	→	R
	3	(A1) op (A2)	→	(A3)

Chú ý: (A): nd của A, op: tác vụ, →: chuyển, A: địa chỉ, R: thanh ghi



Các ký hiệu được dùng trong tập lệnh

- d = độ dời (số có dấu 8 bit = -128 → +127)
- n = hằng số 8 bit
- nn = hằng số 16 bit
- r,r' = thanh ghi 8 bit
- dd, qq, ss, rr = thanh ghi 16 bit

r, r'	Reg.	dd	Pair	qq	Pair
000	B	00	BC	00	BC
001	C	01	DE	01	DE
010	D	02	HL	10	HL
011	E	03	SP	11	AF
100	H				
101	L				
111	A				

Các nhóm lệnh

- Z80 có 158 lệnh và 10 cách định địa chỉ
- Có thể chia các lệnh thành các nhóm sau:
 1. Chuyển dữ liệu (data transfer Group)
 2. Số học và Logic (Arithmetic and Logical Group)
 3. Xoay và dịch (Rotate and Shift Group)
 4. Rẽ nhánh (Branch Group)
 5. Điều khiển CPU
 6. Truy cập cổng xuất/nhập
 7. Xử lý bit (Bit Manipulation Group)
 8. Hoán đổi, chuyển khối và tìm kiếm
(Exchange, Block Transfer, and Search Group)

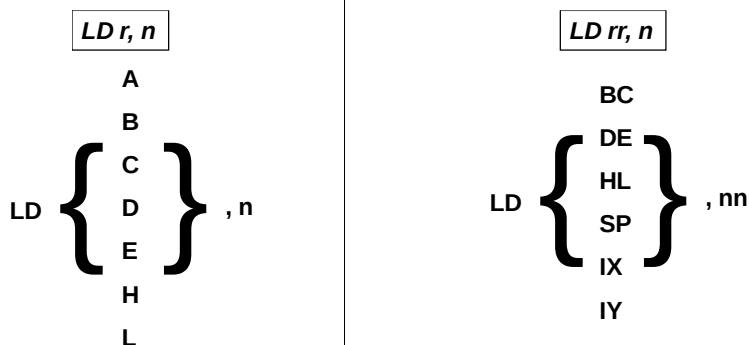
cuu duong than cong . com

1. Nhóm chuyển dữ liệu

- Dạng tổng quát: **LD op1, op2**
 - Sao chép nội dung của toán hạng 2 (op2) vào toán hạng 1 (op1)
 - 2 toán hạng phải cùng chiều dài (8 hay 16 bit)
 - Có nhiều cách định địa chỉ cho các toán hạng
 - toán hạng op1 không thể là hằng số
- Chỉ có 1 toán hạng: với stack
 - **PUSH rp**
 - > cất nội dung của thanh ghi 16 bit vào stack qua con trỏ SP
 - **POP rp**
 - > lấy nội dung của phần tử đỉnh stack (qua con trỏ SP) cất vào thanh ghi 16 bit

Nhóm chuyển dữ liệu – Nạp hằng số cho thanh ghi

- Nạp hằng số 8 bit cho thanh ghi 8 bit: **LD r, n**
- Nạp hằng số 16 bit cho thanh ghi 16 bit: **LD rr, n**



cuu duong than cong . com

TD: Nạp hằng số 8 bit cho thanh ghi

- TD:

LD A, -24; decimal	LD A, 'B'; ASCII
LD A, 15; decimal	LD A, 11001010b; binary
LD A, 0F7H; hex	LD A, 123Q ; Octal

- TD: Hãy viết mã máy cho lệnh LD A, -24
- ĐS: Tra tập lệnh ta thấy lệnh này có 2 byte và opcode như sau: (r = A = 111, và n = -24 = bù 2 của 24 = ECh)

00 r 110
← n →

Như vậy mã máy là: byte 1 = 00111110 = 3Eh
byte 2 = ECh

TD: Nạp hằng số 16 bit cho thanh ghi 16 bit (cặp thanh ghi 8 bit)

- TD:

LD HL,-24

LD DE, 11001010b

LD SP,3FFFh

LD B, 'A'

LD C, '8' ; BC = 'A8'

LD IX, 1234h

- TD: Hãy viết mã máy cho lệnh LD HL,-24

Mnemonic	Symbolic Operation	Flags S Z F5 H F3 P/V N C	Opcode 76 543 210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd ← nn		00 dd0 001		3	3	10	dd Pair
			← n →					00 BC
			← n →					01 DE
LD IX, nn	IX ← nn		11 011 101	DD	4	4	14	02 HL
			00 110 001	21				03 SP
			← n →					
			← n →					
LD IY, nn	IY ← nn		11 111 101	FD	4	4	14	
			00 110 001					

-24 biểu diễn sang số 16 bit bù 2 là FF E8h

Như vậy mã máy là: byte 1 = 0010 0001 = 21h

byte 2 = E8h

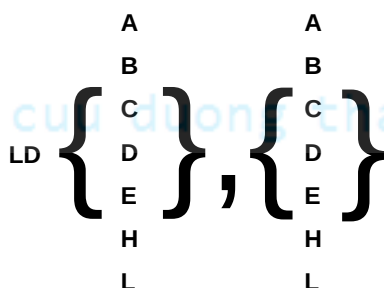
byte 3 = FFh

cuuduongthancong.com

Nhóm chuyển dữ liệu – Sao chép giữa các thanh ghi

- Sao chép nội dung thanh ghi 8 bit ($r \leftarrow r'$) : **LD r, r'**
- Sao chép nội dung thanh ghi 16 bit ($SP \leftarrow rr$): **LD SP, rr**

LD r, r'



TD: LD A, A

LD E, L

LD SP, rr



TD: LD SP, HL

LD SP, IX

Nhóm chuyển dữ liệu – Đọc/ghi bộ nhớ dữ liệu 8 bit

- Đọc bộ nhớ dữ liệu 8 bit cất vào thanh ghi 8 bit với con trỏ là thanh ghi 16 bit rp có thể là HL, IX, và IY:

LD r, (HL)
LD r, (IX+d)
LD r, (IY+d)

- Riêng với r = A có thêm:

LD A, (BC)
LD A, (DE)
LD A, (nn)

Chú ý:

- r là 1 trong các thanh ghi A, B, C, D, E, H và L
- (rp) : định địa chỉ gián tiếp qua thanh ghi HL, BC, hoặc DE
- (IX+d) hay (IY+d) : định địa chỉ chỉ số với d là hằng số có dấu 8 bit → tra bảng
- (nn) : định địa chỉ trực tiếp

- Ghi vào bộ nhớ nội dung thanh ghi 8 bit (r) hoặc dữ liệu 8 bit (n) với con trỏ là thanh ghi 16 bit rp có thể là HL, IX, và IY:

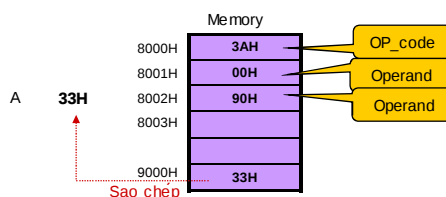
LD (HL), r LD (HL), n
LD (IX+d), r LD (IX+d), n
LD (IY+d), r LD (IY+d), n

- Riêng với r = A có thêm:

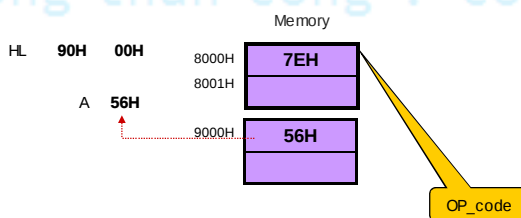
LD (BC), A
LD (DE), A
LD (nn), A

TD: Đọc/ghi bộ nhớ dữ liệu 8 bit

- TD: LD A, (9000H); mã máy là 3AH, 00H, 90H

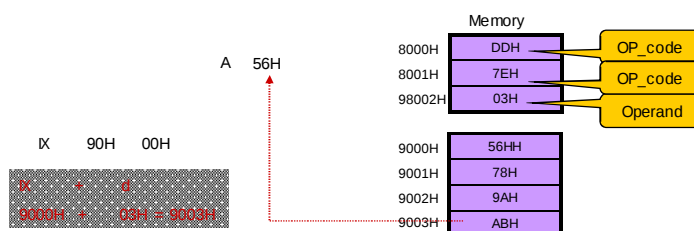


- TD: LD A, (HL); mã máy là 7EH



TD: Đọc/ghi bộ nhớ dữ liệu 8 bit – Tra bảng

- TD: LD A,(IX + 3) có mã máy là DDH,7EH,03H



- Với cùng sơ đồ trên, hãy cho biết nội dung của thanh ghi A sau khi thực thi các lệnh sau:

```
LD IX,9003H
LD A,(IX-2)
```

cuu duong than cong . com

Nhóm chuyển dữ liệu – Đọc/ghi bộ nhớ dữ liệu 16 bit

- Đọc bộ nhớ dữ liệu 16 bit cất vào thanh ghi 16 bit với địa chỉ 16 bit:

LD rp, (nn)

nghĩa là

$rp_L \leftarrow (nn)$ và $rp_H \leftarrow (nn+1)$

- Định địa chỉ hiểu ngầm qua SP:

POP rp

Nghĩa là

$rp_L \leftarrow (SP), SP \leftarrow SP + 1$

$rp_H \leftarrow (SP), SP \leftarrow SP + 1$

Chú ý:

- rp_L là byte thấp của rp và rp_H là byte cao của rp
- rp là 1 trong các thanh ghi 16 bit: BC, DE, HL, SP, IX, và IY
- (nn) : định địa chỉ [trực tiếp] mở rộng (vì dữ liệu 16 bit)

- Ghi vào bộ nhớ dữ liệu 16 bit (ở thanh ghi 16 bit rp) với địa chỉ 16 bit:

LD (nn),rp

nghĩa là

$(nn) \leftarrow rp_L$ và $(nn+1) \leftarrow rp_H$

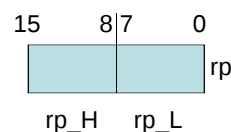
- Định địa chỉ hiểu ngầm qua SP:

PUSH rp

Nghĩa là

$SP \leftarrow SP - 1, (SP) \leftarrow rp_H$

$SP \leftarrow SP - 1, (SP) \leftarrow rp_L$



TD: Đọc/ghi bộ nhớ dữ liệu 16 bit

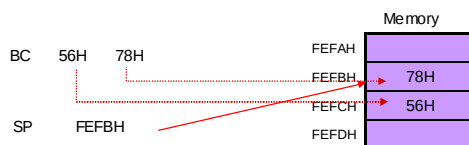
- LD BC, (9000H) có mã máy là EDH, 4BH, 00H, 90H



- LD (9000H), BC có mã máy là EDH, 43H, 00H, 90H



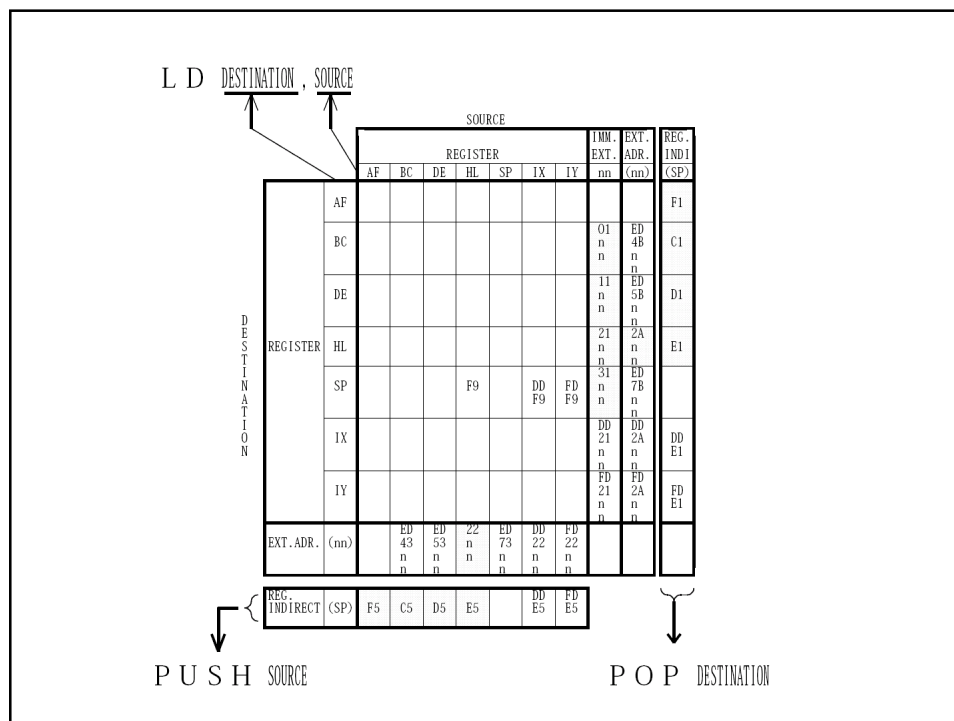
- PUSH BC có mã máy là C5H (giả sử trước khi thực thi: SP = FEFDH)



L D DESTINATION, SOURCE

D
E
S
T
I
N
A
T
I
O
N

		SOURCE																INDEXED	EXT. ADR.	IMM n
		IMPLIED		REGISTER								REG INDIRECT			INDEXED					
		I	R	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(IX +d)	(IY +d)	(nn)				
REGISTER	A	ED 57	ED 5F	7F	78	79	7A	7B	7C	7D	7E	0A	1A	DD 7E d	FD 7E d	3A n n	3E n	06 n		
	B			47	40	41	42	43	44	45	46			DD 46 d	FD 46 d	0A n	0E n	0E n		
	C			4F	48	49	4A	4B	4C	4D	4E			DD 4E d	FD 4E d	0E n	0E n	0E n		
	D			57	50	51	52	53	54	55	56			DD 56 d	FD 56 d	16 n	16 n	16 n		
	E			5F	58	59	5A	5B	5C	5D	5E			DD 5E d	FD 5E d	1E n	1E n	1E n		
	H			67	60	61	62	63	64	65	66			DD 66 d	FD 66 d	26 n	26 n	26 n		
	L			6F	68	69	6A	6B	6C	6D	6E			DD 6E d	FD 6E d	2E n	2E n	2E n		
REG INDIRECT	(HL)			77	70	71	72	73	74	75							36 n	36 n		
	(BC)			02																
	(DE)			12																
INDEXED	(IX +d)			DD 77 d	DD 70 d	DD 71 d	DD 72 d	DD 73 d	DD 74 d	DD 75 d							DD 36 d n	DD 36 d n		
	(IY +d)			FD 77 d	FD 70 d	FD 71 d	FD 72 d	FD 73 d	FD 74 d	FD 75 d							FD 36 d n	FD 36 d n		
EXT. ADR.	(nn)			32 n n																
IMPLIED	I			ED 47																
	R			ED 4F																



2. Nhóm số học và logic

Đa số các lệnh thuộc nhóm này ảnh hưởng đến các cờ (mã điều kiện) trong thanh ghi F (Flag)

- S (Sign) : sau phép toán thì nó S=MSB của kết quả=1(âm)/0(dương)
- Z (Zero) : nếu kết quả là 0 thì Z=1, ngược lại Z=0
- H (Half carry) : số nhớ tại vị trí bit thứ 3 trong phép toán +/-
- P/V (Parity/Overflow) (kiểm tra chẵn lẻ/trên trên)
 - P (với phép toán Logic): 1 (nếu số bit 1 trong A là chẵn = Even parity), ngược lại thì P/V= 0(Odd)
 - V (với phép toán số học): = 1 khi có tràn với phép toán +/-
- N (Negation) -1(sub)/0(add)
- C (Carry) – số nhớ tại vị trí bit thứ 7 trong phép toán +/-

2.1 Nhóm số học 8 bit

Phép toán	Dạng lệnh	Ý nghĩa
Cộng	ADD A,source	$A \leftarrow A + \text{source}$
Cộng có nhớ	ADC A,source	$A \leftarrow A + \text{source} + C$
Trừ	SUB source	$A \leftarrow A - \text{source}$ (A hiểu ngầm)
Trừ có mượn	SBC A,source	$A \leftarrow A - \text{source} - C$
So sánh	CP source	Thực hiện $A - \text{source}$ để đặt các cờ
Tăng thêm 1	INC source	$\text{source} \leftarrow \text{source} + 1$
Giảm bớt 1	DEC source	$\text{source} \leftarrow \text{source} - 1$

Chú ý:

- source có thể là hằng số, thanh ghi 8 bit (A,B,C,D,E,H,L), (HL), (IX+d), (IY+d)
- Các lệnh INC và DEC: source không được là hằng số và Cờ C không bị ảnh hưởng
- Cờ N = 1 sau các lệnh SUB, SBC, và DEC

cuu duong than cong . com

TD: Cộng/trừ 8 bit

; Định nghĩa các hằng số

NUM1 EQU 25 ; NUM1 = 19H

NUM2 EQU -17 ; NUM2 = EFH (bù 2 của 17)

; Thử lệnh ADD với toán hạng 8 bit

LD A,NUM1

LD B,NUM2

ADD A,B ; $A \leftarrow A + B = 25 + (-17) = 8$

cuu duong than cong . com

; Thử lệnh SUB với toán hạng 8 bit

LD A,NUM1

LD B,NUM2

SUB B ; $A \leftarrow A - B = 25 - (-17) = 42$

- *Thí dụ:* Điền các số 0 vào các ô nhớ có địa chỉ từ 8100H đến 817FH.

```

ORG 8000H
LD HL, 8100H
LD A, 00H ; hằng số cần nạp
LD C, 80H ; 817Fh – 8100h +1 =80h
LOOP: LD (HL), A
      INC HL
      DEC C
      JP NZ, LOOP; nếu C ≠ 0 quay lại LOOP
      HALT
      END

```

cuu duong than cong . com

				SOURCE								REG. INDI (HL)	INDEXED (IX +d) (IY +d)		IMM n	
				REGISTER												
DESTINATION				A	B	C	D	E	H	L						
A D D A, SOURCE				REGISTER	A	87	80	81	82	83	84	85	86	DD 86 d	FD 86 d	CG n
A D C A, SOURCE ADD w CARRY				"	A	8F	88	89	8A	8B	8C	8D	8E	DD 8E d	FD 8E d	CE n
S U B SOURCE SUBTRACT				INPLIED (A Reg.)		97	90	91	92	93	94	95	96	DD 96 d	FD 96 d	D6 n
S B C A, SOURCE SUB w CARRY				REGISTER	A	9F	98	99	9A	9B	9C	9D	9E	DD 9E d	FD 9E d	DE n
A N D SOURCE				INPLIED (A Reg.)		A7	A0	A1	A2	A3	A4	A5	A6	DD A6 d	FD A6 d	E6 n
X O R SOURCE				"		AF	A8	A9	AA	AB	AC	AD	AE	DD AE d	FD AE d	EE n
O R SOURCE				"		B7	B0	B1	B2	B3	B4	B5	B6	DD B6 d	FD B6 d	FG n
C P SOURCE COMPARE				"		BF	B8	B9	BA	BB	BC	BD	BE	DD BE d	FD BE d	FE n
I N C SOURCE INCREMENT						3C	04	0C	14	1C	24	2C	34	DD 34 d	FD 34 d	
D E C SOURCE DECREMENT						3D	05	0D	15	1D	2C	2D	35	DD 35 d	FD 35 d	

2.2 Nhóm số học 16 bit

Phép toán	Dạng lệnh	Ý nghĩa
Cộng	ADD HL,ss	$HL \leftarrow HL + ss$
	ADD IX,pp	$IX \leftarrow IX + pp$
	ADD IY,rr	$IY \leftarrow IY + rr$
Cộng có nhớ	ADC HL,ss	$HL \leftarrow HL + ss + C$
Trừ có mượn	SBC HL,ss	$HL \leftarrow HL + ss - C$
Tăng thêm 1	INC source	$source \leftarrow source + 1$
Giảm bớt 1	DEC source	$source \leftarrow source - 1$

Chú ý:

- source có thể là 1 trong các thanh ghi 16 bit (BC,DE,HL, SP, IX và IY)
- ss có thể là 1 trong các thanh ghi 16 bit (BC,DE,H,L, SP)
- pp có thể là 1 trong các thanh ghi 16 bit (BC,DE,IX, SP)
- rr có thể là 1 trong các thanh ghi 16 bit (BC,DE,IY, SP)
- Cờ C không bị ảnh hưởng bởi các lệnh INC và DEC

- *Thí dụ:* Viết chương trình con MULT10 để nhân 10 lần giá trị được cất trong cặp thanh ghi HL và kết quả cất lại vào HL (giả sử kết quả tính không bị tràn).

```

MULT10:    ADD  HL, HL ; HL ← 2 x HL
           LD   D, H      ; (HL)=>(DE)
           LD   E, L
           ADD  HL, HL ; 4 x HL
           ADD  HL, HL ; 8 x HL
           ADD  HL, DE ; (8+2) x HL = 10 x HL
           RET

```

TD: Cộng/trừ 16 bit với hằng số

```

; Định nghĩa hằng số
BIN16_1 EQU 2578 ; = 0A12H
BIN16_2 EQU 1289 ; = 0509H
; Thu lệnh ADD với toán hạng 16 bit dung cặp thanh ghi HL, DE
LD DE,BIN16_1
LD HL,BIN16_2
ADD HL,DE ; HL <- HL + DE = 1289 + 2578 = 3867 = 0F1BH

; Thu lệnh SBC với toán hạng 16 bit dung cặp thanh ghi HL, DE
SCF
CCF ; Xóa cờ Carry (CF=0) hoặc có thể dùng 1 lệnh AND A
LD DE,BIN16_1
LD HL,BIN16_2
SBC HL,DE ; HL <- HL - DE = 1289 - 2578 = -1289 = FAF7H

```

cuu duong than cong . com

TD: Cộng/trừ 16 bit với biến số (1)

<pre> ; Thu lệnh ADD với toán hạng 16 bit dung thanh ghi 8 bit LD IX,BIN16_1 ; Lay byte thap truooc LD A,(IX) LD IY,BIN16_2 LD B,(IY) ADD A,B LD DE,RESULT LD (DE),A INC IX ; Pointer chỉ đến byte cao INC IY INC DE LD A,(IX) LD B,(IY) ADC A,B LD (DE),A HALT </pre>	<pre> ; định nghĩa các biến BIN16_1: DEFW 1289 ; 0509h BIN16_2: DEFW 2578 ; 0A12h RESULT: DEFS 2 END </pre> <u>Chú ý:</u> • 1 số assembler dùng DW thay cho DEFW
--	---

cuu duong than cong . com

TD: Cộng/trừ 16 bit với biến số (2)

```

; Thu lệnh ADD với toán hạng 16 bit dùng cặp thanh ghi HL, DE
LD DE,(BIN16_2) ; DE = 2578 = 0A12h
LD HL,(BIN16_1) ; HL = 1289 = 0509h
ADD HL,DE ; HL <- HL + DE = 1289 + 2578 = 3867 = 0F1BH
LD (RESULT_1),HL

; Thu lệnh SBC với toán hạng 16 bit dùng cặp thanh ghi HL, DE
SCF
CCF ; Xóa cờ Carry (CF=0) hoặc có thể dùng 1 lệnh AND A
LD DE,(BIN16_2)
LD HL,(BIN16_1)
SBC HL,DE ; HL <- HL - DE = 1289 - 2578 = -1289 = FAF7H
LD (RESULT_2),HL
HALT

; định nghĩa các biến
BIN16_1:   DEFW 1289 ; 0509h
BIN16_2:   DEFW 2578 ; 0A12h
RESULT_1:  DEFS 2
RESULT_2:  DEFS 2
END

```

cuu duong than cong . com

		SOURCE						
		DESTINATION	REGISTER					
			BC	DE	HL	SP	IX	IY
HL ADD IX, SOURCE IY	REGISTER	HL	09	19	29	39		
	"	IX	DD 09	DD 19		DD 39	DD 29	
	"	IY	FD 09	FD 19		FD 39		FD 29
ADC HL, SOURCE ADD with CARRY and set flags		"	ED 4A	ED 5A	ED 6A	ED 7A		
SBC HL, SOURCE SUB with CARRY and set flags		"	ED 42	ED 52	ED 62	ED 72		
INC SOURCE INCREMENT			03	13	23	33	DD 23	FD 23
DEC SOURCE DECREMENT			0B	1B	2B	3B	DD 2B	FD 2B

D A A Decimal Adjust Acc	INPLIED (A Reg.)	27
C P L Complement Acc	"	2F
N E G Negate Acc(2's complement)	"	ED 44
C C F Complement Carry Flag	INPLIED (F Reg.)	3F
S C F Set Carry Flag	"	37

cuu duong than cong . com

2.3 Nhóm số học đa dụng

Phép toán	Dạng lệnh	Ý nghĩa
Điều chỉnh thập phân (BCD) sau phép cộng/trừ	DAA	Chuyển đổi nội dung của A sang BCD nén sau phép toán cộng/trừ với các toán hạng BCD
Lấy bù 1	CPL	$A \leftarrow \text{NOT}(A)$
Lấy bù 2	NEG	$A \leftarrow 0 - A = -A$
Lấy bù cờ nhớ	CCF	$C \leftarrow \text{NOT}(C)$
Đặt cờ nhớ lên 1	SCF	$C \leftarrow 1$

TD: Cộng 2 số BCD 4 digit

```

; Cong 2 so BCD 4 digits
LD IX,BIN16_1 ; Lay byte thap truoc
LD A,(IX)
LD IY,BIN16_2
LD B,(IY)
ADD A,B
DAA
LD DE,RESULT
LD (DE),A ; Cat 2 ky so BCD thap truoc cua ket qua
INC IX ; Pointer chi den byte cao
INC IY
INC DE
LD A,(IX)
LD B,(IY)
ADC A,B
DAA
LD (DE),A ; Cat 2 ky so BCD cao truoc cua ket qua
HALT
BIN16_1: DEFW 1289H; Bieu dien BCD cua 1289
BIN16_2: DEFW 2578H; Bieu dien BCD cua 2578
RESULT: DEFS 2
END

```

TD: Tính bù 2 của 1 số 16 bit trong bộ nhớ

```

org 100H
; Lay bu 2 cua 1 so 16 bit NUM1 trong bo nho o dia chi 200H
; va cat lai vao bien 16 bit NUM2 trong bo nho o dia chi 202H
LD HL,(NUM1) ; Neu (NUM1) = 56D3H
LD A,L
CPL ; tinh bu 1 cua byte thap
LD L,A
LD A,H
CPL ; tinh bu 1 cua byte cao
LD H,A
INC HL
LD (NUM2),HL ; thi bu 2 cua no o (NUM2) = A92DH
HALT
org 200H
NUM1: defw 56D3H
NUM2: defs 2 ; bu 2 của 56D3H là A92DH
NUM3: defs 2
END

```

2.4 Nhóm Logic 8 bit

Phép toán	Dạng lệnh	Ý nghĩa
AND	AND source	$A \leftarrow A \text{ AND source}$
OR	OR source	$A \leftarrow A \text{ OR source}$
XOR	XOR source	$A \leftarrow A \text{ XOR source}$

Chú ý:

- Thanh ghi A là toán hạng hiểu ngầm
- source có thể là hằng số, thanh ghi 8 bit (A,B,C,D,E,H,L), (HL), (IX+d), (IY+d)
- Thực hiện phép toán logic cho từng cặp bit tương ứng của 2 toán hạng
- Cờ PV cho biết trị parity chẵn (=1 nếu số bit 1 trong kết quả là số chẵn)
- Cờ C luôn luôn = 0 sau lệnh logic! => dùng để xóa cờ C

cuu duong than cong . com

TD: Các lệnh Logic

; Hiệu ứng cập nhật cờ Parity sau lệnh LOGIC

LD A,1

AND 0FFH ;**P/V=0** (do số bit 1 của kết quả AND là lẻ = 00000001b)

LD A,3

AND 0FFH ;**P/V=1** (do số bit 1 của kết quả AND là chẵn=00000011b)

; Tính bù 1 của 1 số nhị phân trong thanh ghi A

LD A,24 ; A = 18h

XOR 0FFh ; A = 18h XOR FFh = E7h

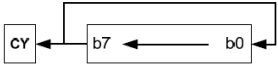
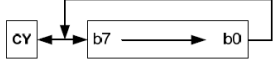
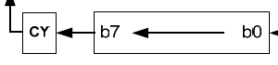
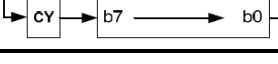
; Đặt 1 bit lên 1 bằng cách OR với 1 tại vị trí đó

LD A,25h ; A = 00100101b

OR 01001000b ; A = A OR 48h = 01101101b = 6Dh

3. Nhóm xoay và dịch bit

3.1 Nhóm xoay bit

Phép toán	Dạng lệnh	Ý nghĩa
Xoay tròn qua trái	RLCA RLC X	
Xoay tròn qua phải	RRCA RRC X	
Xoay trái	RLA RL X	
Xoay phải	RRA RR X	

Chú ý:

- X có thể là A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)
- Với xoay trái cờ C = MSB cũ của toán hạng
- Với xoay phải cờ C = LSB cũ của toán hạng

TD: Đổi số BCD 2 ký số (nén) sang biểu diễn ASCII tương ứng

; TD: A = 25H (BCD của 25) thì chuyển sang ASCII là 32H ('2') và 35H ('5')

; trong D và E (D = 32H và E = 35H)

BCD_ASCII:

LD C,A ; C = A = xyH với x,y = 0 → 9

AND 0F0H ; A = x0H

RRA

RRA

RRA

RRA ; lay nua byte cao: A = 0xH

OR 30H ; Doi sang ASCII tuong duong: A = 3xH

LD D,A ; D = 3xH

LD A,C ; A = xyH

AND 0FH ; lay nua byte thap: A = 0yH

OR 30H ; Doi sang ASCII tuong duong: A = 3yH

LD E,A ; E = 3yH

RET

Chú ý:

- Nội dung của thanh ghi A bị thay đổi trong chương trình con này!
- Lệnh OR 30H có thể được thay bằng lệnh ADD A,30H có cùng kết quả!

TD: Đổi sang biểu diễn ASCII của 2 ký số BCD sang số BCD 2 ký số (nén)

; TD: Biểu diễn ASCII của 25 trong D và E (D = 32H và E = 35H) được

; chuyển thành BCD nén 2 ký số ở A = 25H (BCD của 25)

ASCII_HEX:

```
LD A,D      ; A = D = 3xH
AND 0FH    ; A = 0xH
RLA
RLA
RLA
RLA
LD C,A      ; C = A = x0H
LD A,E      ; A = 3yH
AND 0FH    ; A = 0yH
ADD A,C     ; A = x0H + 0yH = xyH
RET
```

Chú ý:

- Chương trình này là 2m thay đổi nội dung các thanh ghi A và C.

cuu duong than cong . com

3.2 Nhóm xoay digit

Phép toán	Dạng lệnh	Ý nghĩa
Xoay trái digit	RLD	
Xoay phải digit	RRD	

Chú ý:

- Tác động thanh ghi A và nội dung bộ nhớ có địa chỉ trong HL
- Có ý nghĩa khi xoay trái/phải số BCD nên có 3 ký số (digit)

3.3 Nhóm dịch bit

Phép toán	Dạng lệnh	Ý nghĩa
Dịch trái số học	SLA X	
Dịch phải số học	SRA X	
Dịch phải logic	SRL X	

Chú ý:

- X có thể là A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)
- Không có lệnh dịch trái logic SLL vì hiệu ứng giống như SLA

TD: Hiệu ứng của các lệnh dịch bit

LD A, 11 ; A = 0000 1011b = 11
 SLA A ; A = 0001 0110b = 22 = 11 x 2
 SLA A ; A = 0010 1100b = 44 = 22 x 2
 SRA A ; A = 0001 0110b = 22 = 44 / 2
 SRA A ; A = 0000 1011b = 11 = 22 / 2
 SRA A ; A = 0000 0101b = 05 = 11 / 2
 LD A, -22 ; A = 1110 1010b = -22
 SRA A ; A = 1111 0101b = -11 = -22 / 2
 SRA A ; A = 1111 1010b = -6 = -11 / 2
 SLA A ; A = 1111 0100b = -12 = -6 x 2
 SRL A ; A = 0111 1010b = 123
 SRL A ; A = 0011 1101b = 61 = 123 / 2

NX:

- Có thể chia nguyên cho 2: dùng dịch trái để nhân đôi hoặc dịch phải để chia đôi.
- Với có dấu thì phải dùng SLA và SRA
- Có thể vận dụng phép toán dịch và cộng để thực hiện phép toán nhân/chia!

Nhân số không dấu X cho với 1 số có dạng $2^n + 1$

; $X * (2^n + 1) = X * 2^n + X = \text{dịch } X \text{ sang trái } n \text{ bit} + X$

; Giả sử kết quả của phép nhân sẽ đặt trong 1 số 16 bit.

; Chương trình con MUL_2NA1 nhân 1 số ở thanh ghi A với 1 số có dạng $2^n + 1$ ở B

; và kết quả 16 bit được đặt trong cặp thanh ghi DE

```

ORG 100H
LD A,27
LD C,17
CALL MUL_2NA1
HALT

MUL_2NA1:
LD D,0
LD E,A
DEC C
RET Z
CALL SHIFT_N
ADD A,E
LD E,A
LD A,0
ADC A,D
RET

```

; Tính $DE = A \times 2^n$ bằng cách dịch DE

; sang trái n bit (đặt trong C)

SHIFT_N:

```

SRL C
RET C
SLA E ; Dịch trái DE
RL D
JR SHIFT_N

```

Chú ý:

- SV hãy thêm các lệnh để bảo toàn AF và BC khi gọi MUL_2NA1

cuu duong than cong . com

Source and Destination													
REGISTER								REG. IND1	INDEXED		IMPLI ED A		
A	B	C	D	E	H	L	(HL)	(IX +d)	(IY +d)				
R L C —	CB 07	CB 00	CB 01	CB 02	CB 03	CB 04	CB 05	CB 06	DD CB d 06	FD CB d 06	R L C A 07		
Rotate Left Circular													
R R C —	CB 0F	CB 08	CB 09	CB 0A	CB 0B	CB 0C	CB 0D	CB 0E	DD CB d 0E	FD CB d 0E	R R C A 0F		
Rotate Right Circular													
R L —	CB 17	CB 10	CB 11	CB 12	CB 13	CB 14	CB 15	CB 16	DD CB d 16	FD CB d 16	R L A 17		
Rotate Left													
R R —	CB 1F	CB 18	CB 19	CB 1A	CB 1B	CB 1C	CB 1D	CB 1E	DD CB d 1E	FD CB d 1E	R R A 1F		
Rotate Right													
S L A —	CB 27	CB 20	CB 21	CB 22	CB 23	CB 24	CB 25	CB 26	DD CB d 26	FD CB d 26	S L A 27		
Shift Left Arithmetic													
S R A —	CB 2F	CB 28	CB 29	CB 2A	CB 2B	CB 2C	CB 2D	CB 2E	DD CB d 2E	FD CB d 2E	S R A 2F		
Shift Right Arithmetic													
S R L —	CB 3F	CB 38	CB 39	CB 3A	CB 3B	CB 3C	CB 3D	CB 3E	DD CB d 3E	FD CB d 3E	S R L 3F		
Shift Right Logical													
R L D —								ED 6F			R L D 6F		
Rotate Digit Left													
R R D —								ED 67			R R D 67		
Rotate Digit Right													

4. Nhóm rẽ nhánh chương trình

4.1. Các lệnh nhảy (jump) tuyệt đối

Tác vụ	Dạng lệnh	Ý nghĩa
Nhảy tuyệt đối không điều kiện	JP target	$PC \leftarrow \text{target}$
Nhảy tuyệt đối có điều kiện	JP cc, nn	Nếu cc đúng thì $PC \leftarrow nn$ ngược lại thì $PC \leftarrow$ địa chỉ lệnh kế

Chú ý:

- target là giá trị địa chỉ 16 bit có thể là hằng số, (HL), (IX), (IY)
- cc (condition code= mã điều kiện) có thể là Z, N, Z, C, C, P, M, PE, PO,

TD:

LOOP: JP NEXT ; có mã máy là C3h, byte thấp của NEXT, và byte cao của NEXT
 JP (HL) ; $PC \leftarrow HL$
 JP (IX) ; $PC \leftarrow IX$
 JP (IY) ; $PC \leftarrow IY$
 NEXT: JP PO, LOOP
 JP NC, CONTINUE

Ý nghĩa của các [mã] điều kiện

Điều kiện	Ý nghĩa
Z	Đúng khi Z=1 (Zero)
NZ	Đúng khi Z=0 (Non Zero)
C	Đúng khi C=1 (Carry)
NC	Đúng khi C=0 (No Carry)
P	Đúng khi S=0 (P ositive=dương)
M	Đúng khi S=1 (M inus = âm)
PE	Đúng khi PV=1 (P arity E ven = parity chẵn hay có tràn)
PO	Đúng khi PV=0 (P arity O dd= parity lẻ hay không có tràn)

4.2. Các lệnh nhảy tương đối

Tác vụ	Dạng lệnh	Ý nghĩa
Nhảy tương đối không điều kiện	JR target_N	$PC \leftarrow PC + \text{offset}$
Nhảy tương đối có điều kiện	JR cond, target_N	Nếu cond đúng thì $PC \leftarrow \text{nn}$ ngược lại thì $PC \leftarrow$ địa chỉ lệnh kế
Nhảy tương đối có điều kiện với B	DJNZ target_N	$B \leftarrow B - 1$ Nếu $B \neq 0$ thì $PC \leftarrow \text{offset}$ ngược lại thì $PC \leftarrow$ địa chỉ lệnh kế

Chú ý:

- target_N là nhân mà cách xa tương đối với địa chỉ lệnh kế trong dải (-128 → +127)
- offset là giá trị có dấu 8 bit (-128 → +127) được tính theo công thức sau
 $\text{offset} = \text{target_N} - \text{địa chỉ lệnh kế}$
- cond (condition = điều kiện) có thể là Z, NZ, C, NC.
- Kết hợp DJNZ với các lệnh nhảy khác ta có tạo nên các cấu trúc điều khiển:
repeat-until, while-do

cuu duong than cong . com

TD: xóa 1 phần bộ nhớ

Ta muốn xóa nội dung bộ nhớ từ địa chỉ BASE đến địa chỉ BASE + LENGTH - 1 với LENGTH (có giá trị từ 0 đến 255).

TD: Với BASE và LENGTH cụ thể như sau

```
ORG 100H
BASE EQU 400h
LENGTH EQU 32
LD B, LENGTH ; Nạp vào B chiều dài LENGTH
LD A, 0 ; xóa ô nhớ với giá trị trong A
LD HL, BASE ; chỉ đến BASE
```

CLEAR:

```
LD (HL), A ; xóa ô nhớ có địa chỉ chứa trong HL
INC HL ; HL = HL + 1 để chỉ đến ô nhớ kế
; B = B - 1 giảm bộ đếm số ô nhớ
```



HALT

Chú ý:

- Có thể thay 2 lệnh DEC B và JR .. bằng 1 lệnh:
- Khi B = 0 thì số byte cần xóa là bao nhiêu?



Phân tích mã máy của lệnh JR			
Địa chỉ	Mã máy	Dòng	Hợp ngữ
0100		1	ORG 100H
		2	; Xóa 1 phần bộ nhớ
	00 04	3	BASE EQU 400h
	20 00	4	LENGTH EQU 32
0100		5	program
0100	06 20	6	LD B, LENGTH
0102	3E FF	7	LD A, 0FFh
0104	21 00 04	8	LD HL, BASE
0107	77	10	LD (HL), A
0108	23	11	INC HL
0109	05	12	DEC B
	76	14	HALT
	00 01	15	end program

JR NZ, CLEAR có mã máy có 2 byte là 20h, offset (8bit)
 Offset là byte thấp của phép trừ:
 địa chỉ tại CLEAR – địa chỉ lệnh kế = 0107h – 010Ch = FF

Chú ý: Nếu thay JR.. bằng JP NZ, CLEAR thì tại đó ta có mã máy là 3 byte:
 C2 07 01 (vì địa chỉ tại CLEAR là 0107h)

cuu duong than cong . com

TD: Tính tổng của 1 mảng dữ liệu trong bộ nhớ

```

org 100H
; Chương trình tính tổng 1 mảng các số 8 bit không dấu với số phần tử
; trong B và với địa chỉ bắt đầu trong HL
; Kết quả tổng (Sum) được cất vào trong DE
LD A,(N)
LD B,A ; Lấy số phần tử đặt vào B
LD HL, DATA
LD A, 0
LD DE,0 ; Sum <- 0
LOOP: ADD A,(HL)
LD E,A ; Cập nhật byte thấp của Sum
JR NC, NEXT
INC D ; Cập nhật byte cao của Sum
NEXT: INC HL
DJNZ LOOP
HALT

```

DATA: defb 10
 defb 20
 defb 30
 defb 40
 defb 50
 defb 60
 defb 70
 defb 80
 N: defb \$-DATA
 ; \$ là địa chỉ hiện tại
 ; do đó \$ - DATA = 8

4.3. Các lệnh liên quan chương trình con

Tác vụ	Dạng lệnh	Ý nghĩa
Gọi chương trình con	CALL nn CALL cc, nn	$SP \leftarrow SP - 1, (SP) \leftarrow PC_H$ $SP \leftarrow SP - 1, (SP) \leftarrow PC_L$ $PC \leftarrow nn$
Quay về từ chương trình con	RET RET cc	$PC_L \leftarrow (SP), SP \leftarrow SP + 1$ $PC_H \leftarrow (SP), SP \leftarrow SP + 1$

Chú ý:

- cc (condition code= mã điều kiện) có thể là Z, N Z, C, NC, P, M, PE, PO
- Với CALL và RET có điều kiện thì lệnh chỉ được thực thi nếu **cc** đúng, ngược lại thì tiếp tục lệnh kế

TD: Điền các khối bộ nhớ với giá trị ở thanh ghi A

- HL= Địa chỉ bắt đầu
 - B = chiều dài của khối cần điền
- ```

org 100h
LD HL, 400H
LD B, 32
LD A, 0FFH
CALL FILL_PROC ; Điền FFh vào 32 byte từ địa chỉ 0400h
LD HL, 440H
LD B, 16
LD A, 055H
CALL FILL_PROC ; Điền 55h vào 16 byte từ địa chỉ 0440h
HALT
FILL_PROC:
LD (HL), A
INC HL; tăng con trỏ
DJNZ FILL_PROC
RET

```

**TD: Kiểm tra 1 ký tự có thuộc 1 dải trị số không**

Viết chương trình con kiểm tra xem 1 ký tự ASCII ở ô nhớ LOC có phải là số trong dải số từ 0 đến 9, nếu đúng thì cho cờ Z=1.

**Bài giải.**

IS\_NUMBER:

```
LD A, (LOC); lấy ký tự vào A
CP 30H ; so sánh với ASCII của số 0
JR C, EXIT ; nếu A < 30H thì thoát
CP 3AH ; so sánh với '9' + 1
JR NC, EXIT ; nếu A > 39H thì thoát
CP A ; làm cho cờ Z=1
```

EXIT: RET

**Cách 2: Dùng RET cc**

IS\_NUMBER:

```
LD A, (LOC); lấy ký tự vào A
CP 30H ; so sánh với ASCII của số 0
RET C ; nếu A < 30H thì thoát
CP 3AH ; so sánh với '9' + 1
RET NC ; nếu A > 39H thì thoát
CP A ; làm cho cờ Z=1
RET
```

cuu duong than cong . com

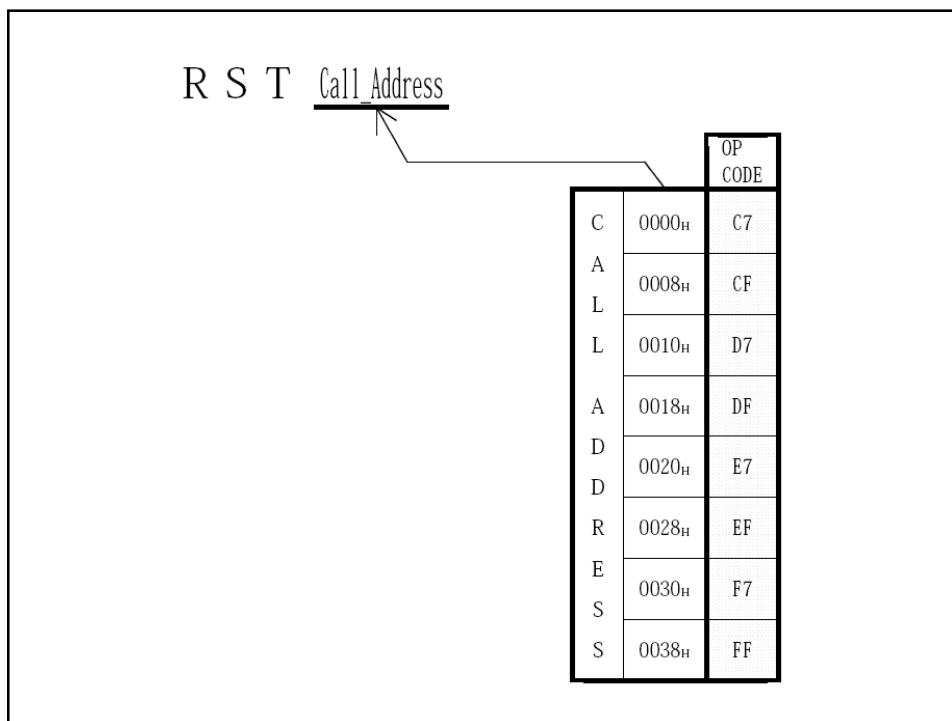
| J P                                                    |                                  | Condition, Address |              | CONDITION    |              |              |              |              |              |              |              |  |  |  |
|--------------------------------------------------------|----------------------------------|--------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--|--|--|
|                                                        |                                  | UN-COND            | CARRY        | NON CARRY    | ZERO         | NON ZERO     | PARITY EVEN  | PARITY ODD   | SIGN NEG     | SIGN POS     | REG B≠0      |  |  |  |
|                                                        |                                  |                    | C            | N C          | Z            | N Z          | P E          | P O          | M            | P            |              |  |  |  |
| J P<br>JUMP                                            | IMMED. EXTENDED                  | nn                 | C3<br>n<br>n | DA<br>n<br>n | D2<br>n<br>n | CA<br>n<br>n | C2<br>n<br>n | EA<br>n<br>n | E2<br>n<br>n | FA<br>n<br>n | F2<br>n<br>n |  |  |  |
|                                                        | (HL)                             | E9                 |              |              |              |              |              |              |              |              |              |  |  |  |
| J P<br>JUMP                                            | REGISTER INDIRECT                | (IX)               | DD<br>E9     |              |              |              |              |              |              |              |              |  |  |  |
|                                                        | (IY)                             | FD<br>E9           |              |              |              |              |              |              |              |              |              |  |  |  |
| J R<br>JUMP RELATIVE                                   | RELATIVE                         | PC+e               | 18<br>e-2    | 38<br>e-2    | 30<br>e-2    | 28<br>e-2    | 20<br>e-2    |              |              |              |              |  |  |  |
| D J N Z<br>DECREMENT B,<br>JUMP IF NON-ZERO<br>C A L L | "                                | "                  |              |              |              |              |              |              |              |              | 10<br>e-2    |  |  |  |
| R E T<br>RETURN                                        | IMMED. EXTENDED                  | nn                 | CD<br>n<br>n | DC<br>n<br>n | D4<br>n<br>n | CC<br>n<br>n | C4<br>n<br>n | EC<br>n<br>n | E4<br>n<br>n | FC<br>n<br>n | F4<br>n<br>n |  |  |  |
|                                                        | REGISTER INDIRECT (SP)<br>(SP+1) | C9                 | D8           | D0           | C8           | C0           | E8           | E0           | F8           | F0           |              |  |  |  |
| R E T I<br>RETURN FROM INT                             | "                                | "                  | ED<br>4D     |              |              |              |              |              |              |              |              |  |  |  |
| R E T N<br>RETURN FROM<br>NON MASKABLE INT             | "                                | "                  | ED<br>45     |              |              |              |              |              |              |              |              |  |  |  |

J P Condition, Address

CONDITION

J P  
JUMPJ P  
JUMPJ R  
JUMP RELATIVED J N Z  
DECREMENT B,  
JUMP IF NON-ZERO  
C A L LR E T  
RETURNR E T I  
RETURN FROM INTR E T N  
RETURN FROM NON MASKABLE INT

cuu duong than cong . com



cuu duong than cong . com

## 5. Nhóm điều khiển CPU

| Tác vụ       | Dạng lệnh | Ý nghĩa                                                                                                                         |
|--------------|-----------|---------------------------------------------------------------------------------------------------------------------------------|
| Không làm gì | NOP       | CPU không làm gì cả.<br>Lệnh này thường dùng để làm trễ<br>bảng phần mềm.                                                       |
| Dừng CPU     | HALT      | Đưa CPU vào trạng thái chờ và<br>khi đó nó không làm gì cả. Trạng<br>thái này sẽ bị phá vỡ khi có ngắt<br>(INT, NMI) hoặc reset |

Chú ý: Ở đây ta không khảo sát các lệnh liên quan về ngắt.

|                                      |          |
|--------------------------------------|----------|
| <b>N O P</b><br>NO OPERATION         | 00       |
| <b>H A L T</b>                       | 76       |
| <b>D I</b><br>Disable Interrupt      | F3       |
| <b>E I</b><br>Enable Interrupt       | FB       |
| <b>I M 0</b><br>Set Interrupt Mode 0 | ED<br>46 |
| <b>I M 1</b><br>Set Interrupt Mode 1 | ED<br>56 |
| <b>I M 2</b><br>Set Interrupt Mode 2 | ED<br>5E |

8080A Mode

Restart to Location 0038<sub>H</sub>

Indirect Call using Register I and 8 bits from Interrupting Device as a Pointer.

cuu duong than cong . com

## 6. Nhóm truy cập cổng xuất/nhập

| Tác vụ                                     | Dạng lệnh  | Ý nghĩa            |
|--------------------------------------------|------------|--------------------|
| Đọc cổng nhập với địa chỉ trực tiếp        | IN A, (n)  | $A \leftarrow (n)$ |
| Đọc cổng nhập với địa chỉ ở thanh ghi C    | IN r, (C)  | $r \leftarrow (C)$ |
| Ghi ra cổng xuất với địa chỉ trực tiếp     | OUT (n), A | $(n) \leftarrow A$ |
| Ghi ra cổng xuất với địa chỉ ở thanh ghi C | OUT (C), r | $(C) \leftarrow r$ |

Chú ý:

- Z80 dùng 8 bit địa chỉ thấp (A7..A0) cho các cổng I/O => có tối đa 256 cổng nhập và 256 cổng xuất
- r có thể là A, B, C, D, E, H, L

### TD: LED sáng lần lượt từ phải sang trái và lặp lại

```

org 100h
PATTERN EQU 01H
LD C,PATTERN
LOOP: LD A,C
 OUT (03H),A
 RLC C ; Dung RRC C se chay sang phai!
 CALL DELAY
 JR LOOP
; Chương trình con DELAY làm trễ
DELAY:
 LD B,0 ; vòng lặp với B là 256 lần!
L1: LD D,0 ; vòng lặp với D là 256 lần!
L2: DEC D
 JR NZ,L2
 DJNZ L1
 RET

```

#### Chú ý:

Trong thí dụ này, cổng xuất với địa chỉ 03H có gắn 8 LED đơn và khi xuất ra 1 tại bit đó thì LED sẽ sáng

### TD: Điều khiển LED sáng chạy sang trái/phải theo trị số 0/1 ở cổng tắc 0 (LSB của cổng nhập 02H)

```

PATTERN EQU 01H
ORG 100h ; Chương trình con DELAY làm trễ
LD C,PATTERN
LOOP: IN A,(02H)
 BIT 0,A
 JR NZ,RIGHT
 RLC C ; chạy sang trái
 JR NEXT
RIGHT: RRC C ; chạy sang phải
NEXT: CALL DELAY
 LD A,C
 OUT (03H),A
 JR LOOP
DELAY: LD B,0 ; vòng lặp với B là 256 lần!
L1: LD D,0 ; vòng lặp với D là 256 lần!
L2: DEC D
 JR NZ,L2
 DJNZ L1
 RET

```

- Khi Z = 0 = NOT (LSB của A) hay cổng tắc 0 có trị là 1 thì LED sáng chạy sang phải

### TD: Hiển thị liên tục các số từ 0 → 9 trên LED 7 đoạn

Giả sử các LED 7 đoạn trong hiển thị 4 ký số lần lượt có các địa chỉ là 0AH, 0BH, 0CH, và 0DH cho các ký số tương ứng từ vị trí cao xuống thấp. Trong TD này ta xuất LED 7 đoạn ở vị trí tận cùng bên phải, nên dùng địa chỉ 0DH!

```

 ORG 100h
 LD L,0 ; chứa số sẽ hiện ra
LOOP: LD A,L
 CALL DEC2LED ; chương trình con đổi BCD sang mã 7 đoạn
 OUT (0DH),A ; Xuất ra LED 7 đoạn tại digit đơn vị!
 CALL DELAY
 INC L
 LD A,9
 CP L
 JR NC,LOOP
 LD L,0
 JR LOOP

```

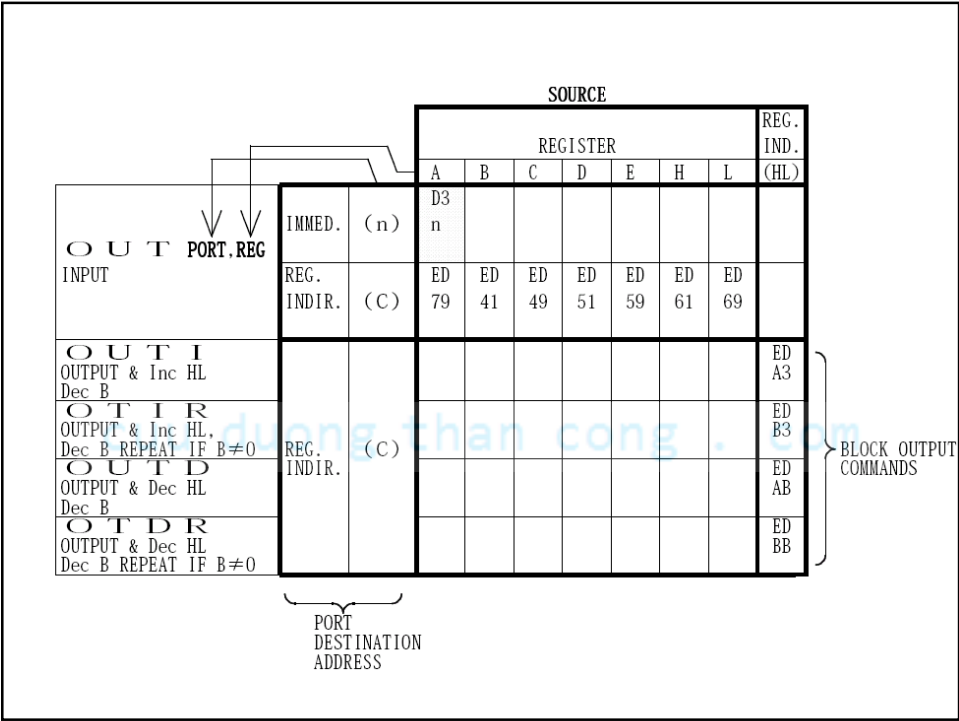
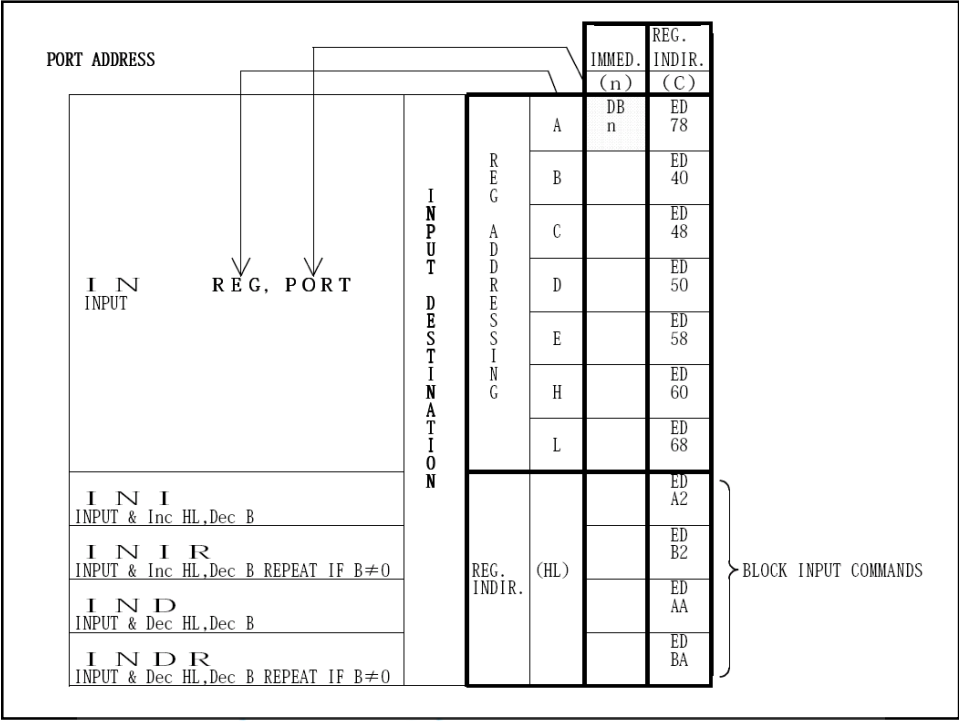
; Chương trình con DELAY làm trễ  
 DELAY: LD B,0 ; vòng lặp với B là 256 lần!  
 L1: LD C,100; vòng lặp với C là 100 lần!  
 L2: DEC C  
 JR NZ,L2  
 DJNZ L1  
 RET

cuu duong than cong . com

### Chương trình con đổi BCD sang mã 7 đoạn

|                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> DEC2LED:     PUSH DE     PUSH IX     AND 0FH     LD E,A     LD D,0     LD IX,XLAT_TAB     ADD IX,DE     LD A,(IX+0)     POP IX     POP DE     RET </pre> | <pre> XLAT_TAB: ; LED 7 đoạn trong mô phỏng này có thu ; tu các đoạn như sau ; ( 0=LED tắt, 1=LED sáng ) ; -hgfedcba     DEFB 00111111B; 0     DEFB 00000110B; 1     DEFB 01011011B; 2     DEFB 01001111B; 3     DEFB 01100110B; 4     DEFB 01101101B; 5     DEFB 01111101B; 6     DEFB 00000111B; 7     DEFB 01111111B; 8     DEFB 01101111B; 9 </pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Chương trình con này bảo toàn nội dung của DE và IX
- IX = địa chỉ đầu bảng tra + chỉ số
- DE = chỉ số (được truyền từ chương trình gọi nó đặt trong A)
- Trị trả về là mã 7 đoạn tương ứng trong thanh ghi A



## 7. Nhóm xử lý bit

| Phép toán                             | Dạng lệnh | Ý nghĩa                                                              |
|---------------------------------------|-----------|----------------------------------------------------------------------|
| Lấy đảo 1 bit trong toán hạng 8 bit   | BIT b, m  | $Z \leftarrow \text{NOT}(m_b)$<br>Z = phủ định của bit thứ b trong m |
| Xóa 1 bit trong toán hạng 8 bit về 0  | RES b, m  | $m_b \leftarrow 0$<br>Bit thứ b trong m bị xóa ( <b>RE</b> Set)      |
| Đặt 1 bit trong toán hạng 8 bit lên 1 | SET b, m  | $m_b \leftarrow 1$<br>Bit thứ b trong m được gán 1 ( <b>SE</b> T)    |

**Chú ý:**

- b có trị là vị trí của bit đang xét, thuộc dải  $0 \rightarrow 7$  (7 = ở MSB và 0 = ở LSB)
- m có thể là A, B, C, D, E, H, L, (HL), (IX+d), (IY+d)

### TD: Mạch so sánh bằng nhau

Mô phỏng mạch số so sánh 2 toán hạng 8 bit (trong A và B), nếu chúng bằng nhau thì đặt LSB của D lên 1, ngược lại thì xóa nó về 0

```
org 100H
LD A,12
LD B,11
CALL COMP ; kết quả = 0
LD A,11
CALL COMP ; kết quả = 1
HALT
```

```
COMP:
 CP B
 JR Z,A_EQ_B
 RES 0,D
 RET
A_EQ_B:
 SET 0,D
 RET
```

**B I T**  
BIT TEST

|   | REGISTER |          |          |          |          |          |          | REG.         | INDEXED             |                     |
|---|----------|----------|----------|----------|----------|----------|----------|--------------|---------------------|---------------------|
|   | A        | B        | C        | D        | E        | H        | L        | INDI<br>(HL) | (IX<br>+d)          | (IY<br>+d)          |
| 0 | CB<br>47 | CB<br>40 | CB<br>41 | CB<br>42 | CB<br>43 | CB<br>44 | CB<br>45 | CB<br>46     | DD<br>CB<br>d<br>46 | FD<br>CB<br>d<br>46 |
| 1 | CB<br>4F | CB<br>48 | CB<br>49 | CB<br>4A | CB<br>4B | CB<br>4C | CB<br>4D | CB<br>4E     | DD<br>CB<br>d<br>4E | FD<br>CB<br>d<br>4E |
| 2 | CB<br>57 | CB<br>50 | CB<br>51 | CB<br>52 | CB<br>53 | CB<br>54 | CB<br>55 | CB<br>56     | DD<br>CB<br>d<br>56 | FD<br>CB<br>d<br>56 |
| 3 | CB<br>5F | CB<br>58 | CB<br>59 | CB<br>5A | CB<br>5B | CB<br>5C | CB<br>5D | CB<br>5E     | DD<br>CB<br>d<br>5E | FD<br>CB<br>d<br>5E |
| 4 | CB<br>67 | CB<br>60 | CB<br>61 | CB<br>62 | CB<br>63 | CB<br>64 | CB<br>65 | CB<br>66     | DD<br>CB<br>d<br>66 | FD<br>CB<br>d<br>66 |
| 5 | CB<br>6F | CB<br>68 | CB<br>69 | CB<br>6A | CB<br>6B | CB<br>6C | CB<br>6D | CB<br>6E     | DD<br>CB<br>d<br>6E | FD<br>CB<br>d<br>6E |
| 6 | CB<br>77 | CB<br>70 | CB<br>71 | CB<br>72 | CB<br>73 | CB<br>74 | CB<br>75 | CB<br>76     | DD<br>CB<br>d<br>76 | FD<br>CB<br>d<br>76 |
| 7 | CB<br>7F | CB<br>78 | CB<br>79 | CB<br>7A | CB<br>7B | CB<br>7C | CB<br>7D | CB<br>7E     | DD<br>CB<br>d<br>7E | FD<br>CB<br>d<br>7E |

cuu duong than cong . com

**R E S**  
BIT RESET

|   | REGISTER |          |          |          |          |          |          | REG.         | INDEXED             |                     |
|---|----------|----------|----------|----------|----------|----------|----------|--------------|---------------------|---------------------|
|   | A        | B        | C        | D        | E        | H        | L        | INDI<br>(HL) | (IX<br>+d)          | (IY<br>+d)          |
| 0 | CB<br>87 | CB<br>80 | CB<br>81 | CB<br>82 | CB<br>83 | CB<br>84 | CB<br>85 | CB<br>86     | DD<br>CB<br>d<br>86 | FD<br>CB<br>d<br>86 |
| 1 | CB<br>8F | CB<br>88 | CB<br>89 | CB<br>8A | CB<br>8B | CB<br>8C | CB<br>8D | CB<br>8E     | DD<br>CB<br>d<br>8E | FD<br>CB<br>d<br>8E |
| 2 | CB<br>97 | CB<br>90 | CB<br>91 | CB<br>92 | CB<br>93 | CB<br>94 | CB<br>95 | CB<br>96     | DD<br>CB<br>d<br>96 | FD<br>CB<br>d<br>96 |
| 3 | CB<br>9F | CB<br>98 | CB<br>99 | CB<br>9A | CB<br>9B | CB<br>9C | CB<br>9D | CB<br>9E     | DD<br>CB<br>d<br>9E | FD<br>CB<br>d<br>9E |
| 4 | CB<br>A7 | CB<br>A0 | CB<br>A1 | CB<br>A2 | CB<br>A3 | CB<br>A4 | CB<br>A5 | CB<br>A6     | DD<br>CB<br>d<br>A6 | FD<br>CB<br>d<br>A6 |
| 5 | CB<br>AF | CB<br>A8 | CB<br>A9 | CB<br>AA | CB<br>AB | CB<br>AC | CB<br>AD | CB<br>AE     | DD<br>CB<br>d<br>AE | FD<br>CB<br>d<br>AE |
| 6 | CB<br>B7 | CB<br>B0 | CB<br>B1 | CB<br>B2 | CB<br>B3 | CB<br>B4 | CB<br>B5 | CB<br>B6     | DD<br>CB<br>d<br>B6 | FD<br>CB<br>d<br>B6 |
| 7 | CB<br>BF | CB<br>B8 | CB<br>B9 | CB<br>BA | CB<br>BB | CB<br>BC | CB<br>BD | CB<br>BE     | DD<br>CB<br>d<br>BE | FD<br>CB<br>d<br>BE |

cuu duong than cong . com

# S E T

BIT SET

|   | REGISTER |          |          |          |          |          |          | REG.     | INDEXED             |                     |
|---|----------|----------|----------|----------|----------|----------|----------|----------|---------------------|---------------------|
|   | A        | B        | C        | D        | E        | H        | L        | INDI     | (IX                 | (IY                 |
|   | (HL)     |          |          |          |          |          |          |          | +d)                 | +d)                 |
| 0 | CB<br>C7 | CB<br>C0 | CB<br>C1 | CB<br>C2 | CB<br>C3 | CB<br>C4 | CB<br>C5 | CB<br>C6 | DD<br>CB<br>d<br>C6 | FD<br>CB<br>d<br>C6 |
| 1 | CB<br>CF | CB<br>C8 | CB<br>C9 | CB<br>CA | CB<br>CB | CB<br>CC | CB<br>CD | CB<br>CE | DD<br>CB<br>d<br>CE | FD<br>CB<br>d<br>CE |
| 2 | CB<br>D7 | CB<br>D0 | CB<br>D1 | CB<br>D2 | CB<br>D3 | CB<br>D4 | CB<br>D5 | CB<br>D6 | DD<br>CB<br>d<br>D6 | FD<br>CB<br>d<br>D6 |
| 3 | CB<br>DF | CB<br>D8 | CB<br>D9 | CB<br>DA | CB<br>DB | CB<br>DC | CB<br>DD | CB<br>DE | DD<br>CB<br>d<br>DE | FD<br>CB<br>d<br>DE |
| 4 | CB<br>E7 | CB<br>E0 | CB<br>E1 | CB<br>E2 | CB<br>E3 | CB<br>E4 | CB<br>E5 | CB<br>E6 | DD<br>CB<br>d<br>E6 | FD<br>CB<br>d<br>E6 |
| 5 | CB<br>EF | CB<br>E8 | CB<br>E9 | CB<br>EA | CB<br>EB | CB<br>EC | CB<br>ED | CB<br>EE | DD<br>CB<br>d<br>EE | FD<br>CB<br>d<br>EE |
| 6 | CB<br>F7 | CB<br>F0 | CB<br>F1 | CB<br>F2 | CB<br>F3 | CB<br>F4 | CB<br>F5 | CB<br>F6 | DD<br>CB<br>d<br>F6 | FD<br>CB<br>d<br>F6 |
| 7 | CB<br>FF | CB<br>F8 | CB<br>F9 | CB<br>FA | CB<br>FB | CB<br>FC | CB<br>FD | CB<br>FE | DD<br>CB<br>d<br>FE | FD<br>CB<br>d<br>FE |

cuu duong than cong . com

### 8. Nhóm hoán đổi, chuyển khối và tìm kiếm

| OP Code<br>Hex | Mnemonic | Tác vụ                                                                                               | Mô tả                                                                                  |
|----------------|----------|------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|
| D9             | EXX      | BC $\leftrightarrow$ B'C'<br>DE $\leftrightarrow$ D'E'<br>HL $\leftrightarrow$ H'L'                  | Hoán đổi các cặp thanh ghi                                                             |
| ED B8          | LDDR     | (DE) $\leftarrow$ (HL)<br>DE $\leftarrow$ DE - 1<br>HL $\leftarrow$ HL - 1<br>BC $\leftarrow$ BC - 1 | Nạp ô nhớ ở (DE) với ô nhớ ở (HL), giảm DE, HL, và BC đi 1, lặp lại cho tới khi BC = 0 |
| ED A1          | CPI      | A - (HL)<br>HL $\leftarrow$ HL + 1<br>BC $\leftarrow$ BC - 1                                         | So sánh ô nhớ (HL) với A, tăng HL thêm 1, giảm BC đi 1, đặt các cờ, A giữ nguyên       |

|                                                                   |        | SOURCE |        |
|-------------------------------------------------------------------|--------|--------|--------|
|                                                                   |        | REG.   | INDIR. |
|                                                                   |        | (HL)   |        |
| <b>L D I</b>                                                      |        |        | ED     |
| Load(DE) $\leftarrow$ (HL),Inc HL & DE,Dec BC                     |        |        | A0     |
| <b>L D I R</b>                                                    |        |        | ED     |
| Load(DE) $\leftarrow$ (HL),Inc HL & DE,Dec BC,Repeat until BC = 0 | REG.   | (DE)   | B0     |
| <b>L D D</b>                                                      | INDIR. |        | ED     |
| Load(DE) $\leftarrow$ (HL) Dec HL & DE,Dec BC                     |        |        | A8     |
| <b>L D D R</b>                                                    |        |        | ED     |
| Load(DE) $\leftarrow$ (HL) Dec HL & DE,Dec BC,Repeat until BC = 0 |        |        | B8     |

Reg HL points to source  
 Reg DE points to destination  
 Reg BC is byte counter

cuu duong than cong . com

|                                                 |        | SERCH LOCATION |        |
|-------------------------------------------------|--------|----------------|--------|
|                                                 |        | REG.           | INDIR. |
|                                                 |        | (HL)           |        |
| <b>C P I</b>                                    |        |                | ED     |
| Inc HL,Dec BC                                   |        |                | A1     |
| <b>C P I R</b>                                  |        |                | ED     |
| Inc HL,Dec BC,Repeat until BC = 0 or find match | REG.   | (DE)           | B1     |
| <b>C P D</b>                                    | INDIR. |                | ED     |
| Dec HL,Dec BC                                   |        |                | A9     |
| <b>C P D R</b>                                  |        |                | ED     |
| Dec HL,Dec BC,Repeat until BC = 0 or find match |        |                | B9     |

HL points to location in memory  
 to be compared with accumulator  
 BC is byte counter

## Tóm tắt tập lệnh Z80 với các cờ

| Instruction              | C | Z | PV  | S | N | H | Comment                                                                         |
|--------------------------|---|---|-----|---|---|---|---------------------------------------------------------------------------------|
| ADD s; ADC s             | f | f | V   | f | 0 | f | 8-bit add or add with carry                                                     |
| SUB s; SBC s; CP s; NEG  | f | f | V   | f | 1 | f | 8-bit subtract, subtract with carry, compare and negate accumulator             |
| AND s                    | 0 | f | P   | f | 0 | 1 | } Logical operations                                                            |
| OR s; XOR s              | 0 | f | P   | f | 0 | 0 |                                                                                 |
| INC s                    | • | f | V   | f | 0 | f | 8-bit increment                                                                 |
| DEC s                    | • | f | V   | f | 1 | f | 8-bit decrement                                                                 |
| ADD ii,ss                | f | • | •   | • | 0 | × | 16-bit add                                                                      |
| ADC HL,ss                | f | f | V   | f | 0 | × | 16-bit add with carry                                                           |
| SBC HL,ss                | f | f | V   | f | 1 | × | 16-bit subtract with carry                                                      |
| RLA; RLCA; RRA; RRCA     | f | • | •   | • | 0 | 0 | Rotate accumulator                                                              |
| RL s; RLC s; RR s; RRC s | f | f | P   | f | 0 | 0 | Rotate and Shift location s                                                     |
| SLA s; SRA s; SRL s      | • | f | P   | f | 0 | 0 | Rotate digit left and right                                                     |
| RLD; RRD                 | f | f | P   | f | • | f | Decimal adjust accumulator                                                      |
| DAA                      | • | • | •   | • | 1 | 1 | Complement accumulator                                                          |
| CPL                      | 1 | • | •   | • | 0 | 0 | Set carry                                                                       |
| SCF                      | f | • | •   | • | 0 | × | Complement carry                                                                |
| CCF                      | • | f | P   | f | 0 | 0 | Input register indirect                                                         |
| IN r,(C)                 | • | f | ×   | × | 1 | × | } Block input and output                                                        |
| INI; IND; OUTI; OUTD     | • | 1 | ×   | × | 1 | × |                                                                                 |
| INIR; INDR; OTIR; OTDR   | • | × | f   | × | 0 | 0 | } Block transfer instructions                                                   |
| LDI; LDD                 | • | × | 0   | × | 0 | 0 |                                                                                 |
| LDIR; LDDR               | • | f | f   | × | 1 | × | } Block search instructions                                                     |
| CPI; CPIR; CPD; CPDR     | • | f | f   | × | 1 | × |                                                                                 |
| LD A,I; LD A,R           | • | f | IFF | f | 0 | 0 | The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag |
| BIT b,s                  | • | f | ×   | × | 0 | 1 | The state of bit b of location s is copied into the Z flag                      |

cuu duong than cong . com

The following notation is used in this table:

| Symbol | Operation                                                                               |
|--------|-----------------------------------------------------------------------------------------|
| C      | Carry flag.                                                                             |
| Z      | Zero flag.                                                                              |
| S      | Sign flag.                                                                              |
| P/V    | Parity or overflow flag. Parity(P) and overflow(V) share the same flag.                 |
| H      | Half-carry flag.                                                                        |
| N      | Add/Subtract flag.                                                                      |
| f      | The flag is affected according to the result of the operation.                          |
| •      | The flag is unchanged by the operation.                                                 |
| 0      | The flag is reset by the operation.                                                     |
| 1      | The flag is set by the operation.                                                       |
| ×      | The flag is a "don't care."                                                             |
| V      | P/V flag affected according to the overflow result of the operation.                    |
| P      | P/V flag affected according to the parity result of the operation.                      |
| r      | Any one of the CPU registers A,B,C,D,E,H,L.                                             |
| s      | Any 8-bit location for all the addressing modes allowed for the particular instruction. |
| ss     | Any 16-bit location for all the addressing modes allowed for that instruction.          |
| ii     | Any one of the registers HL,IX,IY.                                                      |
| R      | Refresh counter.                                                                        |
| n      | 8-bit value in range <0,255>                                                            |
| nn     | 16-bit value in range <0,65535>                                                         |