มหาวิทยาลัยสงขลานครินทร์
คณะวิศวกรรมศาสตร์

สอบกลางภาค: ภาคการศึกษาที่ 1 **ปีการศึกษา:** 2545

วันที่สอบ: 30 กรกฎาคม 2545 **เวลาสอบ:** 13.30-16.30 น.

รหัสวิชา: 240-235 **ห้องสอบ:**

ชื่อวิชา: MICROPROCESSOR ARCHITECTURE AND SYSTEM DESIGN

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

เวลา: 3 ชั่วโมง (180 นาที)

รายละเอียดของข้อสอบ: ข้อสอบมีทั้งหมด 20 หน้า

- เป็นกระดาษคำถามจำนวน 11 หน้า
- เป็น Datasheet จำนวน 9 หน้า

อนุญาต: เครื่องเขียนต่าง ๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือและสมุดโน๊ตใด ๆ เข้าห้องสอบ

คำสั่ง:

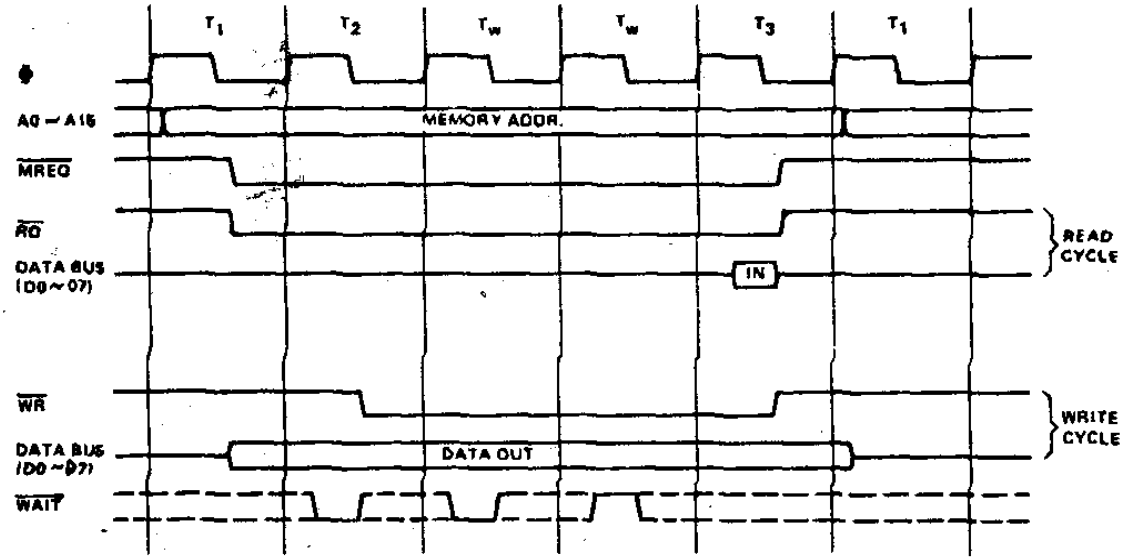
- ให้ทำทุกข้อ
- คำตอบทั้งหมดจะต้องเขียนลงในข้อสอบ
- เขียนชื่อและรหัสให้ชัดเจนในข้อสอบ**ทุกแผ่น**
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งเพิ่มเติมในแต่ละข้อให้ชัดเจน

1. จงอธิบายความหมายของคำต่างๆ ต่อไปนี้ (10 คะแนน)

- Register
- Double Precision Arithmetic
- Program
- Program Monitor
- Interrupt
- Interrupt Vector
- Daisy Chain
- DMA
- PIO
- Machine Cycle
- ALU
- Opcode
- Operand
- Stack
- Assembler
- Subroutine
- Mnemonic

2. จงอธิบายหน้าที่การใช้งานของแฟลกรีจิสเตอร์ของ Z-80 แต่ละบิตว่าใช้แสดงสถานะอะไรบ้าง พร้อมยกตัวอย่างที่ทำให้แต่ละแฟลกเซ็ทค่าหรือรีเซ็ทค่ามาอย่างละเอียด (5 คะแนน)

3. จากรูป ไดอะแกรมดังกล่าวแสดงถึงการทำงานอะไร ของ Z-80 จงอธิบายอย่างละเอียด (5 คะแนน)



4. จงวาดรูป Timing Diagram ของการเฟตช์คำสั่งแบบมีช่วงเวลาการรอคอยของ Z-80 พร้อมทั้งอธิบายการทำงานอย่างละเอียด (10 คะแนน)

5. จากโปรแกรมดังต่อไปนี้ จงแปลโปรแกรมดังกล่าวให้เป็นภาษาเครื่องโดยให้เขียนในรูปแบบของเลขฐานสิบหก โดยเปิดตารางคำสั่งและตารางรหัสแอสกี้ที่แนบมากับท้ายข้อสอบนี้ (10 คะแนน)

```
        ORG 1000H
        db 00H
        db 'A'
        db 'Start'

START:  LD BC,2500H     ;use pointer
        ADD IX,BC       ;add pointer
        CALL DLY        ;delay
        LD (IX+45),03   ;load data
        JP START

DLY:    EQU  1100H
        END
```

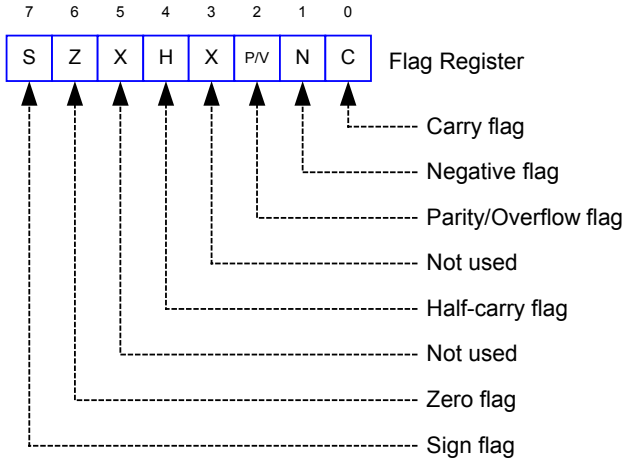
6. จากโปรแกรมดังกล่าวจงหาว่าแต่ละคำสั่งทำงานจำนวนกี่ครั้ง พร้อมทั้งคำนวณเวลาที่ใช้ในการทำโปรแกรมทั้งหมด และบันทึกผลลงในตารางที่ 1 กำหนดให้ใช้ Z-80 ความถี่ 10 MHz (10 คะแนน)

```
        org 100h
START:  LD  HL,2800H     ;Address data
        LD  B,10         ;load count
        XOR A            ;clear sum
LOOP:   ADD A,(HL)       ;add data
        INC     HL       ;address next data
        DJNZ    LOOP     ;loop B times
        END
```

คำสั่ง	จำนวนครั้งในการทำคำสั่ง	เวลาที่ใช้
LD HL,2800H		
LD B,10		
XOR A		
ADD A,(HL)		
INC HL		
DJNZ LOOP		
	รวมเวลาที่ใช้ในการรันโปรแกรม	

ตารางที่ 1

7. จากโปรแกรมดังกล่าว จงหาว่าหลังจากการทำงานของแต่ละคำสั่งเสร็จสิ้นแล้ว ค่ารีจิสเตอร์อะไรเปลี่ยนแปลงบ้าง และจงหาค่าหลังจากการเปลี่ยนแปลงของแต่ละรีจิสเตอร์ (20 คะแนน)



ก่อนรันโปรแกรม

A	75h
Flag Register	C7h
B	A8h
C	C7h
D	10h
E	09h
H	10h
L	05h

100Ah	C5h
1009h	E8h
1008h	04h
1007h	07h
1006h	03h
1005h	A4h
1004h	A3h
1003h	CDh
1002h	71h
1001h	29h
1000h	0Ah

โปรแกรม

```
LD HL,1001H
LD  A,(1009H)
ADD A, B
LD (HL),A
```

หลังรันโปรแกรม

A	
Flag Register	
B	
C	
D	
E	
H	
L	

100Ah	
1009h	
1008h	
1007h	
1006h	
1005h	
1004h	
1003h	
1002h	
1001h	
1000h	

ก่อนรันโปรแกรม

A	75h
Flag Register	C7h
B	A8h
C	C7h
D	10h
E	09h
H	10h
L	05h

100Ah	C5h
1009h	E8h
1008h	04h
1007h	07h
1006h	03h
1005h	A4h
1004h	A3h
1003h	CDh
1002h	71h
1001h	29h
1000h	0Ah

โปรแกรม

```
LD HL,1001H
LD DE,1006H
LD BC,0004H
LDIR
```

หลังรันโปรแกรม

A	
Flag Register	
B	
C	
D	
E	
H	
L	

100Ah	
1009h	
1008h	
1007h	
1006h	
1005h	
1004h	
1003h	
1002h	
1001h	
1000h	

ก่อนรันโปรแกรม

A	75h
Flag Register	C7h
B	A8h
C	C7h
D	10h
E	09h
H	10h
L	05h

100Ah	C5h
1009h	E8h
1008h	04h
1007h	07h
1006h	03h
1005h	A4h
1004h	A3h
1003h	CDh
1002h	71h
1001h	29h
1000h	0Ah

โปรแกรม

```
    CP B
    JR C,PP1
    XOR A
PP1: LD C,A
    SRA L
```

หลังรันโปรแกรม

A	
Flag Register	
B	
C	
D	
E	
H	
L	

100Ah	
1009h	
1008h	
1007h	
1006h	
1005h	
1004h	
1003h	
1002h	
1001h	
1000h	

ก่อนรันโปรแกรม

A	75h
Flag Register	C7h
B	A8h
C	C7h
D	10h
E	09h
H	10h
L	05h

100Ah	C5h
1009h	E8h
1008h	04h
1007h	07h
1006h	03h
1005h	A4h
1004h	A3h
1003h	CDh
1002h	71h
1001h	29h
1000h	0Ah

โปรแกรม

```
LD SP,1002H
PUSH DE
LD SP,HL
POP AF
```

หลังรันโปรแกรม

A	
Flag Register	
B	
C	
D	
E	
H	
L	

100Ah	
1009h	
1008h	
1007h	
1006h	
1005h	
1004h	
1003h	
1002h	
1001h	
1000h	

ก่อนรันโปรแกรม

A	75h
Flag Register	C7h
B	A8h
C	C7h
D	10h
E	09h
H	10h
L	05h

100Ah	C5h
1009h	E8h
1008h	04h
1007h	07h
1006h	03h
1005h	A4h
1004h	A3h
1003h	CDh
1002h	71h
1001h	29h
1000h	0Ah

โปรแกรม

```
EX DE,HL
LDD
ADC A,C
DAA
```

หลังรันโปรแกรม

A	
Flag Register	
B	
C	
D	
E	
H	
L	

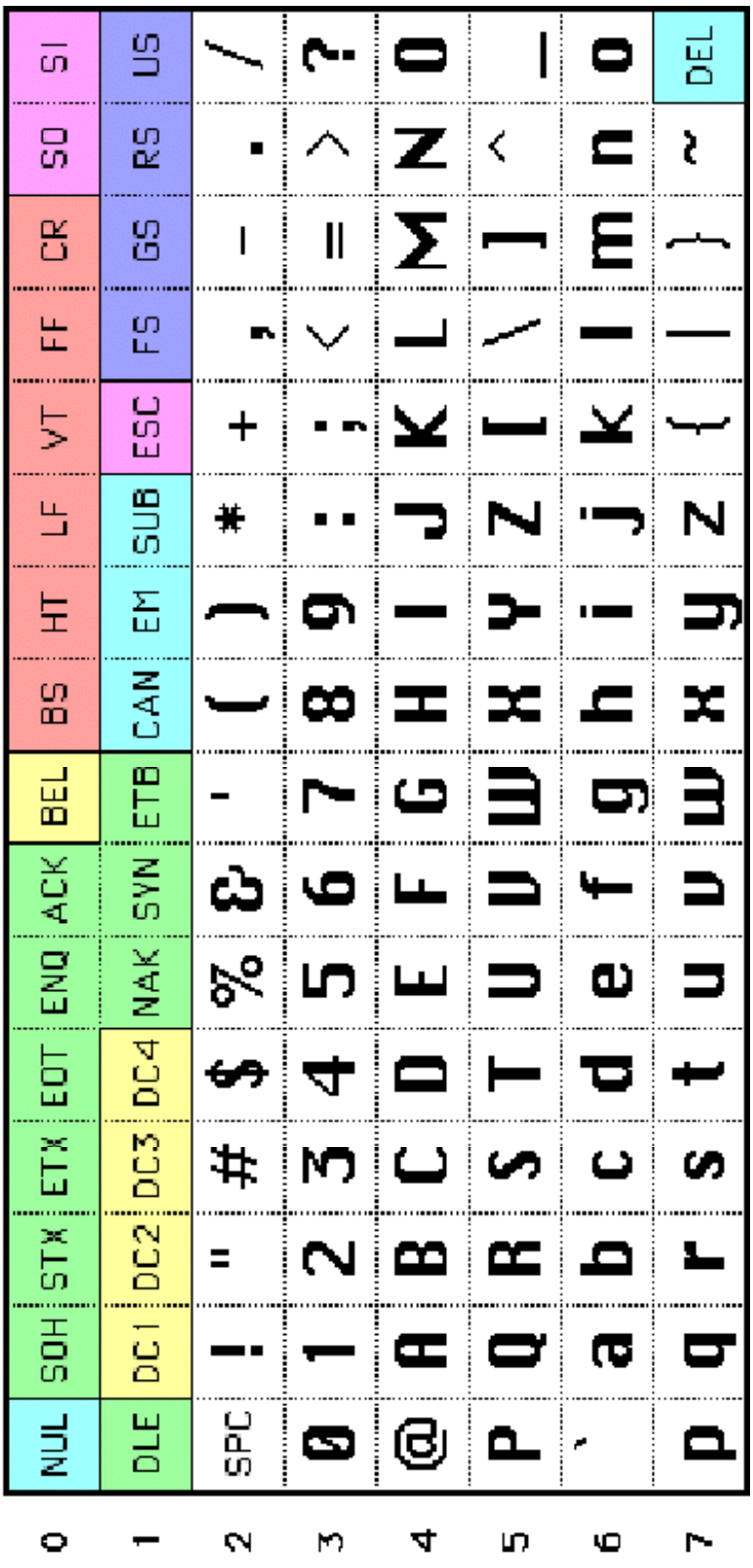
100Ah	
1009h	
1008h	
1007h	
1006h	
1005h	
1004h	
1003h	
1002h	
1001h	
1000h	

8. จงเขียนโปรแกรมเพื่อบวกค่าตัวเลขขนาด 8 บิทจำนวน 25 ตัวที่เก็บอยู่ในหน่วยความจำตำแหน่งเริ่มต้นที่แอดเดรส 2000H โดยให้ผลลัพธ์จากการบวกมีขนาด 16 บิท หลังจากบวกเสร็จให้เก็บผลลัพธ์การบวกไว้ในหน่วยความจำตำแหน่ง 1000H และ 1001H โดยเก็บไบต์ต่ำและไบต์สูงตามลำดับ (10 คะแนน)

9. อินเตอร์รัพต์ของไมโครโพรเซสเซอร์ Z-80 มีกี่แบบ อะไรบ้าง จงอธิบายรายละเอียดกลไกการอินเตอร์รัพต์แต่ละแบบพร้อมทั้งยกตัวอย่างประกอบคำอธิบายอย่างละเอียด (10 คะแนน)

อ. ปัญญยศ ไชยกาฬ ผู้ออกข้อสอบ

ASCII table


0 1 2 3 4 5 6 7 8 9 A B C D E F
0 NUL SOH STX ETX EOT ENQ ACK BEL BS HT LF VT FF CR SO SI
1 DLE DC1 DC2 DC3 DC4 NAK SYN ETB CAN EM SUB ESC FS GS RS US
2 SPC ! = # $ % & - ( ) * + , - . /
3 0 1 2 3 4 5 6 7 8 9 : ; < = > ?
4 @ A B C D E F G H I J K L M N O
5 P Q R S T U V W X Y Z [\ ] _
6 ` a b c d e f g h i j k l m n o
7 p q r s t u v w x y z { | } ~ DEL

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T Cycles	Comments
LD r,r'	r ← r'	•	•	•	•	•	•	01 r r'	1	1	4	r, r' Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
LD r,n	r ← n	•	•	•	•	•	•	00 r 110 ← n →	2	2	7	
LD r,(HL)	r ← (HL)	•	•	•	•	•	•	01 r 110	1	2	7	
LD r,(IX+d)	r ← (IX + d)	•	•	•	•	•	•	11 011 101 01 r 110 ← d →	3	5	19	
LD r,(IY+d)	r ← (IY + d)	•	•	•	•	•	•	11 111 101 01 r 110 ← d →	3	5	19	
LD (HL),r	(HL) ← r	•	•	•	•	•	•	01 110 r	1	2	7	
LD (IX+d),r	(IX + d) ← r	•	•	•	•	•	•	11 011 101 01 110 r ← d →	3	5	19	
LD (IY+d),r	(IY + d) ← r	•	•	•	•	•	•	11 111 101 01 110 r ← d →	3	5	19	
LD (HL),n	(HL) ← n	•	•	•	•	•	•	00 110 110 ← n →	2	3	10	
LD (IX+d),n	(IX + d) ← n	•	•	•	•	•	•	11 011 101 00 110 110 ← d → ← n →	4	5	19	
LD (IY+d),n	(IY + d) ← n	•	•	•	•	•	•	11 111 101 00 110 110 ← d → ← n →	4	5	19	
LD A,(BC)	A ← (BC)	•	•	•	•	•	•	00 001 010	1	2	7	
LD A,(DE)	A ← (DE)	•	•	•	•	•	•	00 011 010	1	2	7	
LD A,(nn)	A ← (nn)	•	•	•	•	•	•	00 111 010 ← n → ← n →	3	4	13	
LD (BC),A	(BC) ← A	•	•	•	•	•	•	00 000 010	1	2	7	
LD (DE),A	(DE) ← A	•	•	•	•	•	•	00 010 010	1	2	7	
LD (nn),A	(nn) ← A	•	•	•	•	•	•	00 110 010 ← n → ← n →	3	4	13	
LD, A,I	A ← I	•	↕	IFF	↕	0	0	11 101 101 01 010 111	2	2	9	
LD A,R	A ← R	•	↕	IFF	↕	0	0	11 101 101 01 011 111	2	2	9	
LD, I,A	I ← A	•	•	•	•	•	•	11 101 101 01 000 111	2	2	9	
LD R,A	R ← A	•	•	•	•	•	•	11 101 101 01 001 111	2	2	9	

Notes:
r, r' means any of the registers A, B, C, D, E, H, L
IFF the content of the interrupt enable flip-flop (IFF) is copied into the P/V flag.
Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
RETI	Return from interrupt	•	•	•	•	•	•	11 101 101 01 001 101	2	4	14	
RETN	Return from nonmaskable interrupt	•	•	•	•	•	•	11 101 101 01 000 101	2	4	14	
RST p	(SP—1) ← PC_H (SP — 2) ← PC_L PC_H ← 0 PC_L ← P	•	•	•	•	•	•	11 t 111	1	3	11	

cc	Condition
000	NZ nonzero
001	Z zero
010	NC noncarry
011	C carry
100	PO parity odd
101	PE parity even
110	P sign positive
111	M sign negative

t	P
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

Notes:
e represents the extension in the relative addressing mode.
e is a signed two's complement number in the range <—126, 129>
e — 2 in the op-code provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.
Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76 543 210				
LD dd,nn	dd ← nn	•	•	•	•	•	•	00 dd0 001 ← n → ← n →	3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX,nn	IX ← nn	•	•	•	•	•	•	11 011 101 00 100 001 ← n → ← n →	4	4	14	
LD IY,nn	IY ← nn	•	•	•	•	•	•	11 111 101 00 100 001 ← n → ← n →	4	4	14	
LD HL,(nn)	H ← (nn + 1) L ← (nn)	•	•	•	•	•	•	00 101 010 ← n → ← n →	3	5	16	
LD, dd,(nn)	dd_H ← (nn + 1) dd_L ← (nn)	•	•	•	•	•	•	11 101 101 01 dd1 011 ← n → ← n →	4	6	20	
LD IX,(nn)	IX_H ← (nn + 1) IX_L ← (nn)	•	•	•	•	•	•	11 011 101 00 101 010 ← n → ← n →	4	6	20	
LD IY,(nn)	IY_H ← (nn + 1) IY_L ← (nn)	•	•	•	•	•	•	11 111 101 00 101 010 ← n → ← n →	4	6	20	
LD (nn),HL	(nn + 1) ← H (nn) ← L	•	•	•	•	•	•	00 100 010 ← n → ← n →	3	5	16	
LD (nn),dd	(nn + 1) ← dd_H (nn) ← dd_L	•	•	•	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	4	6	20	
LD (nn),IX	(nn + 1) ← IX_H (nn) ← IX_L	•	•	•	•	•	•	11 011 101 00 100 010 ← n → ← n →	4	6	20	
LD (nn),IY	(nn + 1) ← IY_H (nn) ← IY_L	•	•	•	•	•	•	11 111 101 00 100 010 ← n → ← n →	4	6	20	
LD SP,HL	SP ← HL	•	•	•	•	•	•	11 111 001	1	1	6	
LD SP,IX	SP ← IX	•	•	•	•	•	•	11 011 101 11 111 001	2	2	10	
LD SP,IY	SP ← IY	•	•	•	•	•	•	11 111 101 11 111 001	2	2	10	
PUSH qq	(SP — 2) ← qq_L (SP — 1) ← qq_H	•	•	•	•	•	•	11 qq0 101	1	3	11	qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP — 2) ← IX_L (SP — 1) ← IX_H	•	•	•	•	•	•	11 011 101 11 100 101	2	4	15	
PUSH IY	(SP — 2) ← IY_L (SP — 1) ← IY_H	•	•	•	•	•	•	11 111 101 11 100 101	2	4	15	
POP qq	qq_H ← (SP + 1) qq_L ← (SP)	•	•	•	•	•	•	11 qq0 001	1	3	10	
POP IX	IX_H ← (SP + 1) IX_L ← (SP)	•	•	•	•	•	•	11 011 101 11 100 001	2	4	14	
POP IY	IY_H ← (SP + 1) IY_L ← (SP)	•	•	•	•	•	•	11 111 101 11 100 001	2	4	14	

Notes:
dd is any of the register pairs BC, DE, HL, SP
qq is any of the register pairs AF, BC, DE, HL

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76	543	210	No. of Bytes	No. of M Cycles	No. of T States	Comments
EX DE,HL	DE ↔ HL	•	•	•	•	•	•	11	101	011	1	1	4	
EX AF,AF'	AF ↔ AF'	•	•	•	•	•	•	00	001	000	1	1	4	
EXX	(BC ↔ BC', DE ↔ DE', HL ↔ HL')	•	•	•	•	•	•	11	011	001	1	1	4	Register bank and auxiliary register bank exchange
EX (SP),HL	H ↔ (SP + 1) L ↔ (SP)	•	•	•	•	•	•	11	100	011	1	5	19	
EX (SP),IX	IX_H ↔ (SP + 1) IX_L ↔ (SP)	•	•	•	•	•	•	11 11	011 100	101 011	2	6	23	
EX (SP),IY	IY_H ↔ (SP + 1) IY_L ↔ (SP)	•	•	•	•	•	•	11 11	111 100	101 011	2	6	23	
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1	•	•	① ↕	•	0	0	11 10	101 100	101 000	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1 Repeat until BC = 0	•	•	0	•	0	0	11 10	101 110	101 000	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
LDD	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1	•	•	① ↕	•	0	0	11 10	101 101	101 000	2	4	16	
LDDR	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1 Repeat until BC = 0	•	•	0	•	0	0	11 10	101 111	101 000	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A − (HL) HL ← HL + 1 BC ← BC − 1	•	② ↕	① ↕	↕	1	↕	11 10	101 100	101 001	2	4	16	
CPIR	A − (HL) HL ← HL + 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0	•	② ↕	① ↕	↕	1	↕	11 10	101 110	101 001	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A − (HL) HL ← HL − 1 BC ← BC − 1	•	② ↕	① ↕	↕	1	↕	11 10	101 101	101 001	2	4	16	
CPDR	A − (HL) HL ← HL − 1 BC ← BC − 1 Repeat until A = (HL) or BC = 0	•	② ↕	① ↕	↕	1	↕	11 10	101 111	101 001	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)

Notes:
① P/V flag is 0 if the result of BC − 1 = 0, otherwise P/V = 1
② Z flag is 1 if A = (HL), otherwise Z = 0.

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76	543	210	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD r	A ← A + r	↕	↕	V	↕	0	↕	10	000	r	1	1	4	r Reg.
ADD n	A ← A + n	↕	↕	V	↕	0	↕	11	000	110	2	2	7	000 B 001 C 010 D
								←	n	→				011 E
ADD (HL)	A ← A + (HL)	↕	↕	V	↕	0	↕	10	000	110	1	2	7	100 H
ADD (IX+d)	A ← A + (IX + d)	↕	↕	V	↕	0	↕	11	011	101	3	5	19	101 L
								10	000	110				111 A
								←	d	→				
ADD (IY+d)	A ← A + (IY + d)	↕	↕	V	↕	0	↕	11	111	101	3	5	19	
								10	000	110				
								←	d	→				
ADC s	A ← A + s + CY	↕	↕	V	↕	0	↕		001					s is any of r, n,
SUB s	A ← A − s	↕	↕	V	↕	1	↕		010					(HL), (IX + d)
SBC s	A ← A − s − CY	↕	↕	V	↕	1	↕		011					(IY + d) as shown for
AND s	A ← A ∧ s	0	↕	P	↕	0	1		100					ADD instruction.
OR s	A ← A ∨ s	0	↕	P	↕	0	0		110					The indicated bits replace
XOR s	A ← A ⊕ s	0	↕	P	↕	0	0		101					the 000 in the ADD set
CP s	A − s	↕	↕	V	↕	1	↕		111					above.
INC r	r ← r + 1	•	↕	V	↕	0	↕	00	r	100	1	1	4	
INC (HL)	(HL) ← (HL) + 1	•	↕	V	↕	0	↕	00	110	100	1	3	11	
INC (IX+d)	(IX + d) ← (IX + d) + 1	•	↕	V	↕	0	↕	11	011	101	3	6	23	
								00	110	100				
								←	d	→				
INC (IY+d)	(IY + d) ← (IY + d) + 1	•	↕	V	↕	0	↕	11	111	101	3	6	23	
								00	110	100				
								←	d	→				
DEC d	d ← d − 1	•	↕	V	↕	1	↕			101				d is any of r, (HL), (IX + d), (IY + d) as shown for INC. Same format and states as INC. Replace 100 with 101 in OP code.

Notes:
The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity. V = 1 means overflow, V = 0 means not overflow, P = 1 means parity of the result is even, P = 0 means parity of the result is odd.

Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
DAA	Converts acc. content into packed bcd following add or subtract with packed bcd operands	↕	↕	P	↕	•	↕	00 100 111	1	1	4	Decimal adjust accumulator
CPL	A ← $\overline{A}$	•	•	•	•	1	1	00 101 111	1	1	4	Complement accumulator (one's complement)
NEG	A ← 0 − A	↕	↕	V	↕	1	↕	11 101 101 01 000 100	2	2	8	Negate acc. (two's complement)
CCF	CY ← $\overline{CY}$	↕	•	•	•	0	X	00 111 111	1	1	4	Complement carry flag
SCF	CY ← 1	1	•	•	•	0	0	00 110 111	1	1	4	Set carry flag
NOP	No operation	•	•	•	•	•	•	00 000 000	1	1	4	
HALT	CPU halted	•	•	•	•	•	•	01 110 110	1	1	4	
DI	IFF ← 0	•	•	•	•	•	•	11 110 011	1	1	4	
EI	IFF ← 1	•	•	•	•	•	•	11 111 011	1	1	4	
IM 0	Set interrupt mode 0	•	•	•	•	•	•	11 101 101 01 000 110	2	2	8	
IM 1	Set interrupt mode 1	•	•	•	•	•	•	11 101 101 01 010 110	2	2	8	
IM 2	Set interrupt mode 2	•	•	•	•	•	•	11 101 101 01 011 110	2	2	8	

Notes:
IFF indicates the interrupt enable flip-flop
CY indicates the carry flip-flop.

Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
ADD HL,ss	HL ← HL + ss	↕	•	•	•	0	X	00 ss1 001	1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL,ss	HL ← HL + ss + CY	↕	↕	V	↕	0	X	11 101 101 01 ss1 010	2	4	15	
SBC HL,ss	HL ← HL − ss − CY	↕	↕	V	↕	1	X	11 101 101 01 ss0 010	2	4	15	
ADD IX,pp	IX ← IX + pp	↕	•	•	•	0	X	11 011 101 00 pp1 001	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY,rr	IY ← IY + rr	↕	•	•	•	0	X	11 111 101 00 rr1 001	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INC ss	ss ← ss + 1	•	•	•	•	•	•	00 ss0 011	1	1	6	
INC IX	IX ← IX + 1	•	•	•	•	•	•	11 011 101 00 100 011	2	2	10	
INC IY	IY ← IY + 1	•	•	•	•	•	•	11 111 101 00 100 011	2	2	10	
DEC ss	ss ← ss − 1	•	•	•	•	•	•	00 ss1 011	1	1	6	
DEC IX	IX ← IX − 1	•	•	•	•	•	•	11 011 101 00 101 011	2	2	10	
DEC IY	IY ← IY − 1	•	•	•	•	•	•	11 111 101 00 101 011	2	2	10	

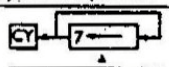
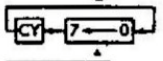
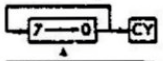
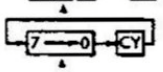
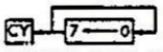
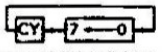
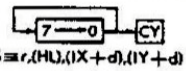
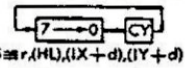
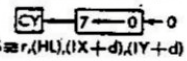
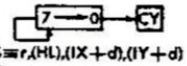
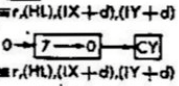
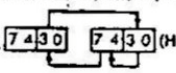
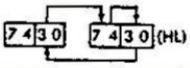
Notes:
ss is any of the register pairs BC, DE, HL, SP
pp is any of the register pairs BC, DE, IX, SP
rr is any of the register pairs BC, DE, IY, SP

Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
BIT b,r	Z ← $\overline{r_b}$	•	↕	X	X	0	1	11 001 011 01 b r	2	2	8	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
BIT b,(HL)	Z ← $\overline{(HL)_b}$	•	↕	X	X	0	1	11 001 011 01 b 110	2	3	12	
BIT b,(IX+d)	← $\overline{(IX + d)_b}$	•	↕	X	X	0	1	11 011 101 11 001 011 ← d → 01 b 110	4	5	20	
BIT b,(IY+d)	Z ← $\overline{(IY + d)_b}$	•	↕	X	X	0	1	11 111 101 11 001 011 ← d → 01 b 110	4	5	20	b Bit Tested 000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
SET b,r	r_b ← 1	•	•	•	•	•	•	11 001 011 11 b r	2	2	8	
SET b,(HL)	$(HL)_b$ ← 1	•	•	•	•	•	•	11 001 011 11 b 110	2	4	15	
SET b,(IX+d)	$(IX + d)_b$ ← 1	•	•	•	•	•	•	11 011 101 11 001 011 ← d → 11 b 110	4	6	23	
SET b,(IY+d)	$(IY + d)_b$ ← 1	•	•	•	•	•	•	11 111 101 11 001 011 ← d → 11 b 110	4	6	23	
RES b,s	s_b ← 0 s ≡ r,(HL), (IX + d), (IY + d)							10				To form new OP-code replace 11 of SET b,s with 10. Flags and time states for SET instruction

Notes:
The notation s_b indicates bit b (0 to 7) of location s.

Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
RLCA		↕	•	•	•	0	0	00 000 111	1	1	4	Rotate left circular accumulator
RLA		↕	•	•	•	0	0	00 010 111	1	1	4	Rotate left accumulator
RRCA		↕	•	•	•	0	0	00 001 111	1	1	4	Rotate right circular accumulator
RRA		↕	•	•	•	0	0	00 011 111	1	1	4	Rotate right accumulator
RLC r		↕	↕	P	↕	0	0	11 001 011 00 000 r	2	2	8	Rotate left circular register r
RLC (HL)		↕	↕	P	↕	0	0	11 001 011 00 000 110	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	 r,(HL),(IX+d),(IY+d)							11 011 101 11 001 011 ← d → 00 000 110	4	6	23	
RLC (IY+d)		↕	↕	P	↕	0	0	11 111 101 11 001 011 ← d → 00 000 110	4	6	23	Instruction format and states are as shown for RLC,s. To form new
RL s	 S≡r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	010				
RRC s	 S≡r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	001				OP-code replace 000 of RLC,s with shown code
RR s	 S≡r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	011				
SLA s	 S≡r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	100				
SRA s	 S≡r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	101				
SRL s	 S≡r,(HL),(IX+d),(IY+d)	↕	↕	P	↕	0	0	111				
RLD	 (HL)	•	↕	P	↕	0	0	11 101 101 01 101 111	2	5	18	Rotate digit left and right between the accumulator and location (HL) The content of the upper half of the accumulator is unaffected
RRD	 (HL)	•	↕	P	↕	0	0	11 101 101 01 100 111	2	5	18	

Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
JP nn	PC ← nn	•	•	•	•	•	•	11	000	011	3	3	10	
								←	n	→				
								←	n	→				
JP cc,nn	If condition cc is true PC ← nn, otherwise continue	•	•	•	•	•	•	11	cc	010	3	3	10	
								←	n	→				
								←	n	→				
JR e	PC ← PC + e	•	•	•	•	•	•	00	011	000	2	3	12	
								←	e − 2	→				
JR C,e	If C = 0, continue	•	•	•	•	•	•	00	111	000	2	2	7	If condition not met
								←	e − 2	→				
	If C = 1, PC ← PC + e										2	3	12	If condition is met
JR NC,e	If C = 1, continue	•	•	•	•	•	•	00	110	000	2	2	7	If condition not met
								←	e − 2	→				
	If C = 0, PC ← PC + e										2	3	12	If condition is met
JR Z,e	If Z = 0 continue	•	•	•	•	•	•	00	101	000	2	2	7	If condition not met
								←	e − 2	→				
	If Z = 1, PC ← PC + e										2	3	12	If condition is met
JR NZ,e	If Z = 1, continue	•	•	•	•	•	•	00	100	000	2	2	7	If condition not met
								←	e − 2	→				
	If Z = 0, PC ← PC + e										2	3	12	If condition met
JP (HL)	PC ← HL	•	•	•	•	•	•	11	101	001	1	1	4	
JP (IX)	PC ← IX	•	•	•	•	•	•	11	011	101	2	2	8	
								11	101	001				
JP (IY)	PC ← IY	•	•	•	•	•	•	11	111	101	2	2	8	
								11	101	001				
DJNZ, e	B ← B−1 If B = 0, continue	•	•	•	•	•	•	00	010	000	2	2	8	If B = 0
								←	e − 2	→				
	If B ≠ 0, PC ← PC + e										2	3	13	If B ≠ 0
CALL nn	(SP−1) ← PC_H (SP−2) ← PC_L PC ← nn	•	•	•	•	•	•	11	001	101	3	5	17	
								←	n	→				
								←	n	→				
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	•	•	•	•	11	cc	100	3	3	10	If cc is false
								←	n	→				
								←	n	→	3	5	17	If cc is true
RET	PC_L ← (SP) PC_H ← (SP + 1)	•	•	•	•	•	•	11	001	001	1	3	10	
RET cc	If condition cc is false continue, otherwise same as RET	•	•	•	•	•	•	11	cc	000	1	1	5	If cc is false
											1	3	11	If cc is true

cc	Condition
000	NZ nonzero
001	Z zero
010	NC noncarry
011	C carry
100	PO parity odd
101	PE parity even
110	P sign positive
111	M sign negative

cc	Condition
000	NZ nonzero
001	Z zero
010	NC noncarry
011	C carry
100	PO parity odd
101	PE parity even
110	P sign positive
111	M sign negative

Mnemonic	Symbolic Operation	Flags C	Z	P/V	S	N	H	Op-Code 76 543 210	No. of Bytes	No. of M Cycles	No. of T States	Comments
IN A,(n)	A ← (n)	•	•	•	•	•	•	11 011 011 ← n →	2	3	11	n to A_0–A_7 Acc to A_8–A_{15}
IN r,(C)	r ← (C) if r = 110 only the flags will be affected	•	↕	P	↕	0	↕	11 101 101 01 r 000	2	3	11	C to A_0–A_7 B to A_8–A_{15}
INI	(HL) ← (C) B ← B − 1 HL ← HL + 1	•	① ↕	X	X	1	X	11 101 101 10 100 010	2	4	16	C to A_0–A_7 B to A_8–A_{15}
INIR	(HL) ← (C) B ← B − 1 HL ← HL + 1 Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 110 010	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A_0–A_7 B to A_8–A_{15}
IND	(HL) ← (C) B ← B − 1 HL ← HL − 1	•	① ↕	X	X	1	X	11 101 101 10 101 010	2	4	16	C to A_0–A_7 B to A_8–A_{15}
INDR	(HL) ← (C) B ← B − 1 HL ← HL − 1 Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 111 010	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A_0–A_7 B to A_8–A_{15}
OUT (n),A	(n) ← A	•	•	•	•	•	•	11 010 011 ← n →	2	3	11	n to A_0–A_7 Acc to A_8–A_{15}
OUT (C),r	(C) ← r	•	•	•	•	•	•	11 101 101 01 r 001	2	3	12	C to A_0–A_7 B to A_8–A_{15}
OUTI	(C) ← (HL) B ← B − 1 HL ← HL + 1	•	① ↕	X	X	1	X	11 101 101 10 100 011	2	4	16	C to A_0–A_7 B to A_8–A_{15}
OTIR	(C) ← (HL) B ← B − 1 HL ← HL + 1 Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 110 011	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A_0–A_7 B to A_8–A_{15}
OUTD	(C) ← (HL) B ← B − 1 HL ← HL − 1	•	① ↕	X	X	1	X	11 101 101 10 101 011	2	4	16	C to A_0–A_7 B to A_8–A_{15}
OTDR	(C) ← (HL) B ← B − 1 HL ← HL − 1 Repeat until B = 0	•	1	X	X	1	X	11 101 101 10 111 011	2 2	5 (If B ≠ 0) 4 (If B = 0)	21 16	C to A_0–A_7 B to A_8–A_{15}

Notes:
① If the result of B − 1 is zero the Z flag is set, otherwise it is reset.

Flag Notation:
• = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
↕ = flag is affected according to the result of the operation.