DATA TRANSFER GROUP

Move

MOV	A,A	7F
MOV	A,B	78
MOV	A,C	79
MOV	A,D	7A
MOV	A,E	7B
MOV	A,H	7C
MOV	A,L	7D
MOV	A,M	7E
MOV	B,A	47
MOV	B,B	40
MOV	B,C	41
MOV	B,D	42
MOV	B,E	43
MOV	B,H	44
MOV	B,L	45
MOV	B,M	46
MOV	C,A	4F
MOV	C,B	48
MOV	C,C	49
MOV	C,D	4A
MOV	C,E	4B
MOV	C,H	4C
MOV	C,L	4D
MOV	C,M	4E
MOV	D,A	57
MOV	D,B	50
MOV	D,C	51
MOV	D,D	52
MOV	D,E	53
MOV	D,H	54
MOV	D,L	55
MOV	D,M	56

Move (cont)

MOV	E,A	5F
MOV	E,B	58
MOV	E,C	59
MOV	E,D	5A
MOV	E,E	5B
MOV	E,H	5C
MOV	E,L	5D
MOV	E,M	5E
MOV	H,A	67
MOV	H,B	60
MOV	H,C	61
MOV	H,D	62
MOV	H,E	63
MOV	H,H	64
MOV	H,L	65
MOV	H,M	66
MOV	L,A	6F
MOV	L,B	68
MOV	L,C	69
MOV	L,D	6A
MOV	L,E	6B
MOV	L,H	6C
MOV	L,L	6D
MOV	L,M	6E
MOV	M,A	77
MOV	M,B	70
MOV	M,C	71
MOV	M,D	72
MOV	M,E	73
MOV	M,H	74
MOV	M,L	75

XCHG EB

Move Immediate

MVI	A, byte	3E
MVI	B, byte	06
MVI	C, byte	0E
MVI	D, byte	16
MVI	E, byte	1E
MVI	H, byte	26
MVI	L, byte	2E
MVI	M, byte	36

Load Immediate

LXI	B, dble	01
LXI	D, dble	11
LXI	H, dble	21
LXI	SP, dble	31

Load / Store

LDAX B	0A
LDAX D	1A
LHLD adr	2A
LDA adr	3A
STAX B	02
STAX D	12
SHLD adr	22
STA adr	32

byte = constant, or logical/arithmetic expression that evaluates to an 8-bit data quantity. (Second byte of 2-byte instructions)

dble = constant, or logical/arithmetic expression that evaluates to a 16-bit data quantity. (Second and Third bytes of 3-byte instructions)

adr = 16-bit address (Second and Third bytes of 3-byte instructions)

* = all flags (C, Z, S, P, AC) affected.

** = all flags except CARRY affected. (Exception : INX and DCX affect no flags)

† = only CARRY affected

ARITHMETIC AND LOGICAL GROUP

Add*

ADD	A	87
ADD	B	80
ADD	C	81
ADD	D	82
ADD	E	83
ADD	H	84
ADD	L	85
ADD	M	86
ADC	A	8F
ADC	B	88
ADC	C	89
ADC	D	8A
ADC	E	8B
ADC	H	8C
ADC	L	8D
ADC	M	8E
SUB	A	97
SUB	B	90
SUB	C	91
SUB	D	92
SUB	E	93
SUB	H	94
SUB	L	95
SUB	M	96
SBB	A	9F
SBB	B	98
SBB	C	99
SBB	D	9A
SBB	E	9B
SBB	H	9C
SBB	L	9D
SBB	M	9E

Increment**

INR	A	3C
INR	B	04
INR	C	0C
INR	D	14
INR	E	1C
INR	H	24
INR	L	2C
INR	M	34
INX	B	03
INX	D	13
INX	H	23
INX	SP	33

Decrement **

DCR	A	3D
DCR	B	05
DCR	C	0D
DCR	D	15
DCR	E	1D
DCR	H	25
DCR	L	2D
DCR	M	35
DCX	B	0B
DCX	D	1B
DCX	H	2B
DCX	SP	3B

Specials

DAA*	27
CMA	2F
STC †	37
CMC †	3F

Logical*

ANA	A	A7
ANA	B	A0
ANA	C	A1
ANA	D	A2
ANA	E	A3
ANA	H	A4
ANA	L	A5
ANA	M	A6
XRA	A	AF
XRA	B	A8
XRA	C	A9
XRA	D	AA
XRA	E	AB
XRA	H	AC
XRA	L	AD
XRA	M	AE
ORA	A	B7
ORA	B	B0
ORA	C	B1
ORA	D	B2
ORA	E	B3
ORA	H	B4
ORA	L	B5
ORA	M	B6
CMP	A	BF
CMP	B	B8
CMP	C	B9
CMP	D	BA
CMP	E	BB
CMP	H	BC
CMP	L	BD
CMP	M	BE

Double Add †

DAD	B	09
DAD	D	19
DAD	H	29
DAD	SP	39

Rotate †

RLC	07
RRC	0F
RAL	17
RAR	1F

Arith & Logical Immediate

ADI byte	C6
ACI byte	CE
SUI byte	D6
SBI byte	DE
ANI byte	E6
XRI byte	EE
ORI byte	F6
CPI byte	FE

ARITHMETIC AND LOGICAL GROUP

Branch Control Group

Jump

JMP adr	C3
JNZ adr	C2
JZ adr	CA
JNC adr	D2
JC adr	DA
JPO adr	E2
JPE adr	EA
JP adr	F2
JM adr	FA
PCHL	E9

Call

CALL adr	CD
CNZ adr	C4
CZ adr	CC
CNC adr	D4
CC adr	DC
CPO adr	E4
CPE adr	EC
CP adr	F4
CM adr	FC

Return

RET	C9
RNZ	C0
RZ	C8
RNC	D0
RC	D8
RPO	E0
RPE	E8
RP	F0
RM	F8

Restart

RST	0	C7
RST	1	CF
RST	2	D7
RST	3	DF
RST	4	E7
RST	5	EF
RST	6	F7
RST	7	FF

I/O and Machine Control

Stack Ops

PUSH	B	C5
PUSH	D	D5
PUSH	H	E5
PUSH	PSW	F5
POP	B	C1
POP	D	D1
POP	H	E1
POP	PSW*	F1

XTHL	E3
SPHL	F9

Input / Output

OUT byte	D3
IN byte	DB

Control

DI	F3
EI	FB
NOP	00
HLT	76

New Instructions (8085 Only)

RIM	20
SIM	30

ASSEMBLER REFERENCE

Operators

(,)
NUL
LOW, HIGH
*,/, MOD, SHL, SHR
+, –
NOT
AND
OR, XOR

Assembles reference (Cont.)

Pseudo Instruction

General

ORG
END
EQU
SET
DS
DB
DW

Macros :

MACRO
ENDM
LOCAL
REPT
IRP
IRPC
EXITM

Relocation

ASEG	NAME
DSEG	STKLN
CSEG	STACK
PUBLIC	MEMORY
EXTRN	

Conditional Assembly :

IF
ELSE
ENDIF

Constant Definition

0BDH, 1AH	Hex
105D, 105	Decimal
72O, 72Q	Octal
11011B, 00110B	Binary
'TEST', 'A' 'B'	ASCII

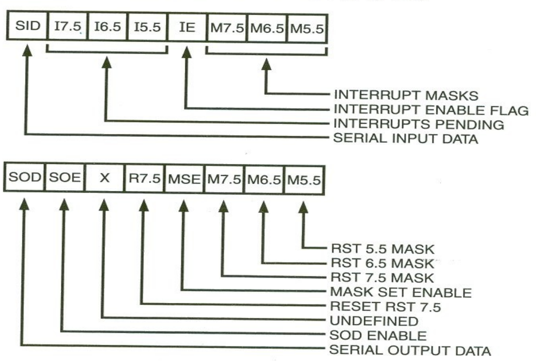
RESTART TABLE

Name	Code	Restart Address
RST 0	C7	0000_{16}
RST 1	CF	0008_{16}
RST 2	D7	0010_{16}
RST 3	DF	0018_{16}
RST 4	E7	0020_{16}
TRAP	Hardware* Function	0024_{16}
RST 5	EF	0028_{16}
RST 5.5	Hardware* Function	$002C_{16}$
RST 6	F7	0030_{16}
RST 6.5	Hardware* Function	0034_{16}
RST 7	FF	0038_{16}
RST 7.5	Hardware* Function	$003C_{16}$

*NOTE : The hardware functions refer to the on-chip interrupt feature of the 8085 only

USE OF THE A REGISTER BY RIM AND SIM INSTRUCTIONS (8085 ONLY)

A REGISTER AFTER EXECUTING RIM



8080 / 8085
INSTRUCTION SET REFERENCE TABLES

INTERNAL REGISTER ORGANIZATION

A Reg. (8)

B Reg. (8)	C Reg. (8)
D Reg. (8)	E Reg. (8)
H Reg. (8)	L Reg. (8)
Program Counter (16)	
Stack Pointer (16)	

S	Z	X	AC	X	P	X	CY

CARRY
PARITY
AUX. CARRY
ZERO
SIGN

UNDEFINED

REGISTER - PAIR ORGANIZATION

PSW

A (8)	FLAGS (8)

NOTE : Leftmost Byte is high-order byte for arithemetic operations and addressing. Left byte is pushed on stack first. Right byte is popped first.

B	(B/C)	(16)
D	(D/E)	(16)
H	(H/L)	(16)
Prog. Ctr.		(16)
Stack Ptr.		(16)

BRANCH CONTROL INSTRUCTIONS

Flag Condition	Jump		Call		Return	
Zero = True	JZ	CA	CZ	CC	RZ	C8
Zero = False	JNZ	C2	CNZ	C4	RNZ	C0
Carry = True	JC	DA	CC	DC	RC	D8
Carry = False	JNC	D2	CNC	D4	RNC	D0
Sign = Positive	JP	F2	CP	F4	RP	F0
Sign = Negative	JM	FA	CM	FC	RM	F8
Parity = Even	JPE	EA	CPE	EC	RPE	E8
Parity = Odd	JPO	E2	CPO	E4	RPO	E0
Unconditional	JMP	C3	CALL	CD	RET	C9

ACCUMULATOR OPERATIONS

	Code	Function
XRA A	AF	Clear A and Clear Carry
ORA A	B7	Clear Carry
CMC	3F	Complement Carry
CMA	2F	Complement Accumulator
STC	37	Set Carry
RLC	07	Rotate Left
RRC	0F	Rotate Right
RAL	17	Rotate Left Thru Carry
RAR	1F	Rotate Right Thru Carry
DAA	27	Decimal Adjust Accum.

REGISTER PAIR AND STACK OPERATIONS

		Register Pair					
	PSW (A/F)	B (B/C)	D (D/E)	H (H/L)	SP	PC	Function
INX		03	13	23	33		Increment Register Pair
DCX		0B	1B	2B	3B		Decrement Register Pair
LDAX		0A	1A	7E(1)			Load A Indirect (Reg. Pair holds Addrs)
STAX		02	12	77(2)			Store A Indirect (Reg. Pair holds Addrs)
LHLD				2A			Load H/L Direct (Bytes 2 and 3 hold Addrs)
SHLD				22			Store H/L Direct (Bytes 2 and 3 hold Addrs)
LXI		01	11	21	31	C3(3)	Load Reg. Pair Immediate (Bytes 2 and 3 hold immediate data)
PCHL						E9	Load PC with H/L (Branch to Adrs in H/L)
XCHG			EB				Exchange Reg. Pairs D/E and H/L
DAD		09	19	29	39		Add Reg. Pair to H/L
PUSH	F5	C5	D5	E5			Push Reg. Pair on Stack
POP	F1	C1	D1	E1			Pop Reg. Pair off Stack
XTHL				E3			Exchange H/L with Top of Stack
SPHL					F9		Load SP with H/L

Notes : 1. This is MOV A,M. 2. This is MOV M,A. 3. This is JMP.

00	NOP		2B	DCX	H	56	MOV	D,M	81	ADD	C	AC	XRA	H	D7	RST	2
01	LXI	B,dble	2C	INR	L	57	MOV	D,A	82	ADD	D	AD	XRA	L	D8	RC	
02	STAX	B	2D	DCR	L	58	MOV	E,B	83	ADD	E	AE	XRA	M	D9	---	
03	INX	B	2E	MVI	L,byte	59	MOV	E,C	84	ADD	H	AF	XRA	A	DA	JC	adr
04	INR	B	2F	CMA		5A	MOV	E,D	85	ADD	L	B0	ORA	B	DB	IN	byte
05	DCR	B	30	SIM*		5B	MOV	E,E	86	ADD	M	B1	ORA	C	DC	CC	adr
06	MVI	B,byte	31	LXI	SP,dble	5C	MOV	E,H	87	ADD	A	B2	ORA	D	DD	---	
07	RLC		32	STA	adr	5D	MOV	E,L	88	ADC	B	B3	ORA	E	DE	SBI	byte
08	---		33	INX	SP	5E	MOV	E,M	89	ADC	C	B4	ORA	H	DF	RST	3
09	DAD	B	34	INR	M	5F	MOV	E,A	8A	ADC	D	B5	ORA	L	E0	RPO	
0A	LDAX	B	35	DCR	M	60	MOV	H,B	8B	ADC	E	B6	ORA	M	E1	POP	H
0B	DCX	B	36	MVI	M,byte	61	MOV	H,C	8C	ADC	H	B7	ORA	A	E2	JPO	adr
0C	INR	C	37	STC		62	MOV	H,D	8D	ADC	L	B8	CMP	B	E3	XTHL	
0D	DCR	C	38	---		63	MOV	H,E	8E	ADC	M	B9	CMP	C	E4	CPO	adr
0E	MVI	C,byte	39	DAD	SP	64	MOV	H,H	8F	ADC	A	BA	CMP	D	E5	PUSH	H
0F	RRC		3A	LDA	adr	65	MOV	H,L	90	SUB	B	BB	CMP	E	E6	ANI	byte
10	---		3B	DCX	SP	66	MOV	H,M	91	SUB	C	BC	CMP	H	E7	RST	4
11	LXI	D,dble	3C	INR	A	67	MOV	H,A	92	SUB	D	BD	CMP	L	E8	RPE	
12	STAX	D	3D	DCR	A	68	MOV	L,B	93	SUB	E	BE	CMP	M	E9	PCHL	
13	INX	D	3E	MVI	A,byte	69	MOV	L,C	94	SUB	H	BF	CMP	A	EA	JPE	adr
14	INR	D	3F	CMC		6A	MOV	L,D	95	SUB	L	C0	RNZ		EB	XCHG	
15	DCR	D	40	MOV	B,B	6B	MOV	L,E	96	SUB	M	C1	POP	B	EC	CPE	adr
16	MVI	D,byte	41	MOV	B,C	6C	MOV	L,H	97	SUB	A	C2	JNZ	adr	ED	---	
17	RAL		42	MOV	B,D	6D	MOV	L,L	98	SBB	B	C3	JMP	adr	EE	XRI	byte
18	---		43	MOV	B,E	6E	MOV	L,M	99	SBB	C	C4	CNZ	adr	EF	RST	5
19	DAD	D	44	MOV	B,H	6F	MOV	L,A	9A	SBB	D	C5	PUSH	B	F0	RP	
1A	LDAX	D	45	MOV	B,L	70	MOV	M,B	9B	SBB	E	C6	ADI	byte	F1	POP	PSW
1B	DCX	D	46	MOV	B,M	71	MOV	M,C	9C	SBB	H	C7	RST	0	F2	JP	adr
1C	INR	E	47	MOV	B,A	72	MOV	M,D	9D	SBB	L	C8	RZ		F3	DI	
1D	DCR	E	48	MOV	C,B	73	MOV	M,E	9E	SBB	M	C9	RET		F4	CP	adr
1E	MVI	E,byte	49	MOV	C,C	74	MOV	M,H	9F	SBB	A	CA	JZ	adr	F5	PUSH	PSW
1F	RAR		4A	MOV	C,D	75	MOV	M,L	A0	ANA	B	CB	---		F6	ORI	byte
20	RIM*		4B	MOV	C,E	76	HLT		A1	ANA	C	CC	CZ	adr	F7	RST	6
21	LXI	H,dble	4C	MOV	C,H	77	MOV	M,A	A2	ANA	D	CD	CALL	adr	F8	RM	
22	SHLD	adr	4D	MOV	C,L	78	MOV	A,B	A3	ANA	E	CE	ACI	byte	F9	SPHL	
23	INX	H	4E	MOV	C,M	79	MOV	A,C	A4	ANA	H	CF	RST	1	FA	JM	adr
24	INR	H	4F	MOV	C,A	7A	MOV	A,D	A5	ANA	L	D0	RNC		FB	EI	
25	DCR	H	50	MOV	D,B	7B	MOV	A,E	A6	ANA	M	D1	POP	D	FC	CM	adr
26	MVI	H,byte	51	MOV	D,C	7C	MOV	A,H	A7	ANA	A	D2	JNC	adr	FD	---	
27	DAA		52	MOV	D,D	7D	MOV	A,L	A8	XRA	B	D3	OUT	byte	FE	CPI	byte
28	---		53	MOV	D,E	7E	MOV	A,M	A9	XRA	C	D4	CNC	adr	FF	RST	7
29	DAD	H	54	MOV	D,H	7F	MOV	A,A	AA	XRA	D	D5	PUSH	D			
2A	LHLD	adr	55	MOV	D,L	80	ADD	B	AB	XRA	E	D6	SUI	byte			

*8085 Only