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

b = byte data
d = 16b data
a = 16b address
flags:
* = affects all
† = carry only
‡ = all but carry

Instruction Timing (T States)

Data Transfer		CMP r	4	Control	
MOV rd,rs	4	CMP M	7	DI	4
MOV M,r	7			EI	4
MOV r,M	7	16 bit		NOP	4
XCHG	4	DAD x	10	HLT	5
MVI r	7	INX x	6		
MVI M	10	DCX x	6	Stack	
LXI x	10			PUSH	12
		Rotate		POP	10
Load/Store		RLC	4		
LDAX x	7	RRC	4	Branch	
LDA	13	RAL	4	RST x	12
STAX x	7	RAR	4	JMP	10
SHLD	16			Jx	7/10
STA	13	Immediate		CALL	18
		ADI	7	Cx	9/18
Arith & Logic		ACI	7	RET	10
ADD r	4	SUI	7	Rx	6/12
ADD M	7	SBI	7		
ADC r	4	ANI	7	T-State Time	
ADC M	7	XRI	7	CPU *f*	t_{CYC}
SUB r	4	ORI	7	8.0MHz	125nS
SUB M	7	CPI	7	6.0	166.7
SBB r	4			5.0	200
SBB M	7	Specials		4.0	250
INR R	4	DAA	4	3.58	279
INR M	10	CMA	4	3.125	320
DCR r	4	STC	4	3.072	325.5
DCR M	10	CMC	4	3.0	333.3
ANA r	4			2.5	400
ANA M	7	I/O		2.0	500
XRA r	4	OUT	10	1.25	800
XRA M	7	IN	10	1.2	833.3
ORA r	4	RIM	4	1.0	1000
ORA M	7	SIM	4	0.75	1333

Accumulator Operations

Instruction	Code	Function
XRA A	AF	Clear A and Clear Carry
CMA	2F	Complement Accumulator
ORA A	B7	Clear Carry
CMC	3F	Complement Carry
STC	37	Set Carry
RLC	7	Rotate Left, MSB=CY
RRC	0F	Rotate Right, LSB=>CY
RAL	17	Rotate Ledt Thru Carry
RAR	1F	Rotate Right Thru Carry

After RIM:

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5

SID = Serial In Data
Ix.5 = Interrupt Pending
IE = Interrupt Enable Flag
Mx.5 = Interrupt Masks for external lines

Before SIM:

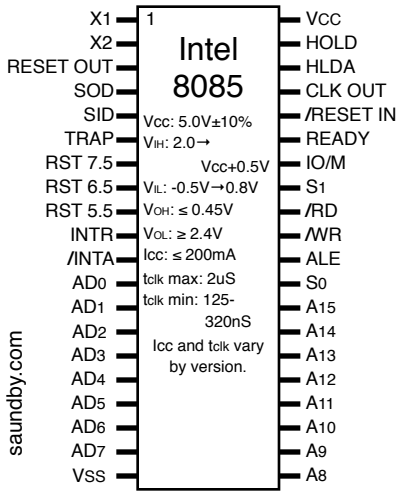
SOD	SOE	---	R7.5	MSE	M7.5	M6.5	M5.5

SOD = Serial Out Data
SOE = Serial Out Enable (enable SOD output)
R7.5= Reset (clear) RST 7.5 Interrupt
MSE = Mask Set Enable (1 to set new masks)
Mx.5= New RSTx.5 Mask Setting (1 to enable)

Note: RST5.5, RST6.5, RST7.5 are masked by default. Masks must be cleared before use.

Visit http://saundby.com/ for more 8085 info.

Intel 8085 Reference Card



saundby.com

Register Organization

A(8)	F(8) PSW(16)

Flags

S	Z	[U]	AC	-	P	[V]	C

Sign, Zero, [U], Aux Carry, Parity, [V], Carry

B(8)	C(8) B/C(16)
D(8)	E(8) D/E(16)
H(8)	L(8) H/L(16)
Program Counter	PC (16)
Stack Pointer	SP (16)

Note: [ ]=Undocumented/80C85B only

Data Transfer Instruction Group

```
MOV  A, A  7F    MOV  E, A  5F    MVI A, b  3E
     A, B  78         E, B  58        B, b  06
     A, C  79         E, C  59        C, b  0E
     A, D  7A         E, D  5A        D, b  16
     A, E  7B         E, E  5B        E, b  1E
     A, H  7C         E, H  5C        H, b  26
     A, L  7D         E, L  5D        L, b  2E
     A, M  7E         E, M  5E        M, b  36

MOV  B, A  47    MOV  H, A  67    XCHG      EB
     B, B  40         H, B  60
     B, C  41         H, C  61    LXI B, d  01
     B, D  42         H, D  62        D, d  11
     B, E  43         H, E  63        H, d  21
     B, H  44         H, H  64        SP,d  31
     B, L  45         H, L  65
     B, M  46         H, M  66    Load/Store
                                  LDAX B    0A
MOV  C, A  4F    MOV  L, A  6F    LDAX D    1A
     C, B  48         L, B  68    LHLD a    2A
     C, C  49         L, C  69    LDA  a    3A
     C, D  4A         L, D  6A    [LHLX]   [ED]
     C, E  4B         L, E  6B
     C, H  4C         L, H  6C    STAX B    02
     C, L  4D         L, L  6D    STAX D    12
     C, M  4E         L, M  6E    SHLD a    22
                                  STA  a    32
                                  [SHLX]   [D9]
MOV  D, A  57    MOV  M, A  77
     D, B  50         M, B  70    arguments:
     D, C  51         M, C  71    b = byte data
     D, D  52         M, D  72    d = 16b data
     D, E  53         M, E  73    a = 16b address
     D, H  54         M, H  73    flags:
     D, L  55         M, L  75    * = affects all
     D, M  56                     † = carry only
                                  ‡ = all but carry
```

Arithmetic & Logical Instruction Group

```
ADD* A  87    INR‡ A  3C    ORA* A  B7
     B  80         B  04         B  B0
     C  81         C  0C         C  B1
     D  82         D  14         D  B2
     E  83         E  1C         E  B3
     H  84         H  24         H  B4
     L  85         L  2C         L  B5
     M  86         M  34         M  B6

ADC* A  8F    DCR‡ A  3D    CMP* A  BF
     B  88         B  05         B  B8
     C  89         C  0D         C  B9
     D  8A         D  15         D  BA
     E  8B         E  1D         E  BB
     H  8C         H  25         H  BC
     L  8D         L  2D         L  BD
     M  8E         M  35         M  BE

SUB* A  97    ANA* A  A7    16 Bit
     B  90         B  A0    DAD† B  09
     C  91         C  A1         D  19
     D  92         D  A2         H  29
     E  93         E  A3         SP 39
     H  94         H  A4    INX  B  03
     L  95         L  A5         D  13
     M  96         M  A6         H  23
                                 SP 33
SBB* A  9F    XRA* A  AF    DCX  B  0B
     B  98         B  A8         D  1B
     C  99         C  A9         H  2B
     D  9A         D  AA         SP 3B
     E  9B         E  AB
     H  9C         H  AC    [DSUB]  [08]
     L  9D         L  AD
     M  9E         M  AE
```

Arith & Logic continued -->

Arith & Logic

```
Rotate
RLC†      07
RRC†      0F
RAL†      17
RAR†      1F
[AHRL]   [10]

Immediate
ADI  b    C6
ACI  b    CE
SUI  b    D6
SBI  b    DE
ANI  b    E6
XRI  b    EE
ORI  b    F6
CPI  b    FE
[LDHI b][28]
[LDSI b][38]

Specials
DAA*      27
CMA       2F
STC†      37
CMC†      3F
```

I/O & Control

```
OUT  b    D3
IN   b    DB

RIM       20
SIM       30

Control
DI        F3
EI        FB
NOP       00
HLT       76
```

Stack

```
PUSH B    C5
     D    D5
     H    E5
     PSW  F5
POP  B    C1
     D    D1
     H    E1
     PSW* F1
XTHL      E3
SPHL      F9
```

Branch

```
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
[RST V]  [CB]

Restart Vector
RST  0   0000H
RST  1   0008H
RST  2   0010H
RST  3   0018H
RST  4   0020H
TRAP     0024H
RST  5   0028H
RST 5.5  002CH
RST  6   0030H
RST 6.5  0034H
RST  7   0038H
RST 7.5  003CH
```

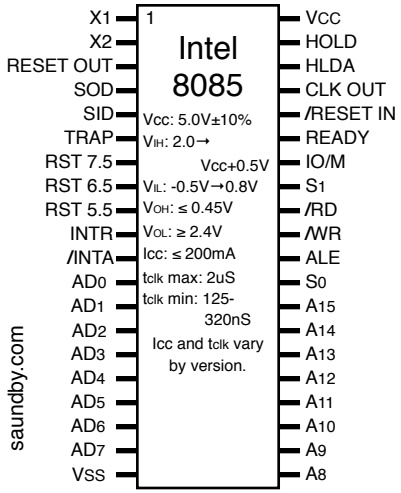
Branch

```
Jump
JMP  a    C3
JNZ  a    C2
JZ   a    CA
JNC  a    D2
JC   a    DA
JPO  a    E2
JPE  a    EA
JP   a    F2
JM   a    FA
PCHL      E9
[JNUI]   [DD]
[JUI]    [FD]

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

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

Intel 8085 Reference Card



saundby.com

Register Organization

A(8)	F(8) PSW(16)

Flags

S	Z	[U]	AC	-	P	[V]	C

Sign, Zero, [U], Aux Carry, Parity, [V], Carry

B(8)	C(8) B/C(16)
D(8)	E(8) D/E(16)
H(8)	L(8) H/L(16)
Program Counter	PC (16)
Stack Pointer	SP (16)

Note: [ ]=Undocumented/80C85B only

Data Transfer Instruction Group

MOV	A, A	7F	MOV	E, A	5F	MVI	A, b	3E
	A, B	78		E, B	58		B, b	06
	A, C	79		E, C	59		C, b	0E
	A, D	7A		E, D	5A		D, b	16
	A, E	7B		E, E	5B		E, b	1E
	A, H	7C		E, H	5C		H, b	26
	A, L	7D		E, L	5D		L, b	2E
	A, M	7E		E, M	5E		M, b	36
MOV	B, A	47	MOV	H, A	67	XCHG		EB
	B, B	40		H, B	60	LXI	B, d	01
	B, C	41		H, C	61		D, d	11
	B, D	42		H, D	62		H, d	21
	B, E	43		H, E	63		SP,d	31
	B, H	44		H, H	64			
	B, L	45		H, L	65			
	B, M	46		H, M	66			
MOV	C, A	4F	MOV	L, A	6F			
	C, B	48		L, B	68			
	C, C	49		L, C	69			
	C, D	4A		L, D	6A			
	C, E	4B		L, E	6B			
	C, H	4C		L, H	6C			
	C, L	4D		L, L	6D			
	C, M	4E		L, M	6E			
MOV	D, A	57	MOV	M, A	77			
	D, B	50		M, B	70			
	D, C	51		M, C	71			
	D, D	52		M, D	72			
	D, E	53		M, E	73			
	D, H	54		M, H	73			
	D, L	55		M, L	75			
	D, M	56						

Load/Store

LDAX	B	0A
LDAX	D	1A
LHLD	a	2A
LDA	a	3A
[LHLX]		[ED]
STAX	B	02
STAX	D	12
SHLD	a	22
STA	a	32
[SHLX]		[D9]

arguments:
b = byte data
d = 16b data
a = 16b address

flags:
* = affects all
† = carry only
ǂ = all but carry

Arithmetic & Logical Instruction Group

ADD*	A	87	INR†	A	3C	ORA*	A	B7
	B	80		B	04		B	B0
	C	81		C	0C		C	B1
	D	82		D	14		D	B2
	E	83		E	1C		E	B3
	H	84		H	24		H	B4
	L	85		L	2C		L	B5
	M	86		M	34		M	B6
ADC*	A	8F	DCR†	A	3D	CMP*	A	BF
	B	88		B	05		B	B8
	C	89		C	0D		C	B9
	D	8A		D	15		D	BA
	E	8B		E	1D		E	BB
	H	8C		H	25		H	BC
	L	8D		L	2D		L	BD
	M	8E		M	35		M	BE
SUB*	A	97	ANA*	A	A7			
	B	90		B	A0			
	C	91		C	A1			
	D	92		D	A2			
	E	93		E	A3			
	H	94		H	A4			
	L	95		L	A5			
	M	96		M	A6			
SBB*	A	9F	XRA*	A	AF			
	B	98		B	A8			
	C	99		C	A9			
	D	9A		D	AA			
	E	9B		E	AB			
	H	9C		H	AC			
	L	9D		L	AD			
	M	9E		M	AE			

16 Bit

DAD†	B	09
	D	19
	H	29
	SP	39
INX	B	03
	D	13
	H	23
	SP	33
DCX	B	0B
	D	1B
	H	2B
	SP	3B
[DSUB]		[08]

Arith & Logic continued -->

Arith & Logic

Rotate

RLC†		07
RRC†		0F
RAL†		17
RAR†		1F
[AHRL]		[10]

Immediate

ADI	b	C6
ACI	b	CE
SUI	b	D6
SBI	b	DE
ANI	b	E6
XRI	b	EE
ORI	b	F6
CPI	b	FE
[LDHI b]		[28]
[LDSI b]		[38]

Specials

DAA*		27
CMA		2F
STC†		37
CMC†		3F

I/O & Control

OUT	b	D3
IN	b	DB
RIM		20
SIM		30

Control

DI		F3
EI		FB
NOP		00
HLT		76

Stack

PUSH	B	C5
	D	D5
	H	E5
	PSW	F5
POP	B	C1
	D	D1
	H	E1
	PSW*	F1
XTHL		E3
SPHL		F9

Branch

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
[RST V]		[CB]

Restart Vector

RST	0	0000H
RST	1	0008H
RST	2	0010H
RST	3	0018H
RST	4	0020H
TRAP		0024H
RST	5	0028H
RST	5.5	002CH
RST	6	0030H
RST	6.5	0034H
RST	7	0038H
RST	7.5	003CH

Branch

Jump

JMP	a	C3
JNZ	a	C2
JZ	a	CA
JNC	a	D2
JC	a	DA
JPO	a	E2
JPE	a	EA
JP	a	F2
JM	a	FA
PCHL		E9
[JNUI]		[DD]
[JUI]		[FD]

Call

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

Return

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

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

b = byte data
d = 16b data
a = 16b address

flags:
* = affects all
† = carry only
ǂ = all but carry

Instruction Timing (T States)

Data Transfer

MOV rd,rs	4
MOV M,r	7
MOV r,M	7
XCHG	4
MVI r	7
MVI M	10
LXI x	10

Load/Store

LDAX x	7
LDA	13
STAX x	7
SHLD	16
STA	13

Arith & Logic

ADD r	4
ADD M	7
ADC r	4
ADC M	7
SUB r	4
SUB M	7
SBB r	4
SBB M	7
INR R	4
INR M	10
DCR r	4
DCR M	10
ANA r	4
ANA M	7
XRA r	4
XRA M	7
ORA r	4
ORA M	7
CMP r	4
CMP M	7

16 bit

DAD x	10
INX x	6
DCX x	6

Rotate

RLC	4
RRC	4
RAL	4
RAR	4

Immediate

ADI	7
ACI	7
SUI	7
SBI	7
ANI	7
XRI	7
ORI	7
CPI	7

Specials

DAA	4
CMA	4
STC	4
CMC	4

I/O

OUT	10
IN	10
RIM	4
SIM	4

Control

DI	4
EI	4
NOP	4
HLT	5

Stack

PUSH	12
POP	10

Branch

RST x	12
JMP	10
Jx	7/10
CALL	18
Cx	9/18
RET	10
Rx	6/12

T-State Time

CPU *f*	tCYC
8.0MHz	125nS
6.0	166.7
5.0	200
4.0	250
3.58	279
3.125	320
3.072	325.5
3.0	333.3
2.5	400
2.0	500
1.25	800
1.2	833.3
1.0	1000
0.75	1333

Accumulator Operations

Instruction	Code	Function
XRA A	AF	Clear A and Clear Carry
CMA	2F	Complement Accumulator
ORA A	B7	Clear Carry
CMC	3F	Complement Carry
STC	37	Set Carry
RLC	7	Rotate Left, MSB=CY
RRC	0F	Rotate Right, LSB=>CY
RAL	17	Rotate Ledt Thru Carry
RAR	1F	Rotate Right Thru Carry

After RIM:

SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5

SID = Serial In Data
Ix.5 = Interrupt Pending
IE = Interrupt Enable Flag
Mx.5 = Interrupt Masks for external lines

Before SIM:

SOD	SOE	---	R7.5	MSE	M7.5	M6.5	M5.5

SOD = Serial Out Data
SOE = Serial Out Enable (enable SOD output)
R7.5= Reset (clear) RST 7.5 Interrupt
MSE = Mask Set Enable (1 to set new masks)
Mx.5= New RSTx.5 Mask Setting (1 to enable)

Note: RST5.5, RST6.5, RST7.5 are masked by default. Masks must be cleared before use.

Visit http://saundby.com/ for more 8085 info.