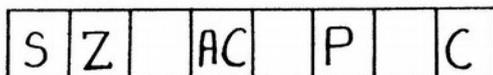


8080

Codes opération

Mnémoniques



REGISTRE D'ETAT DU 80 85

Indicateurs

Il existe 5 indicateurs liés à l'exécution d'instructions par le SAB 8085 : zéro, signe, parité, retenue et retenue auxiliaire. Chacun d'eux est représenté par un registre de 1 bit dans l'UC. Le positionnement d'un indicateur s'effectue en imposant la valeur 1 au bit et sa suppression en imposant la valeur 0.

Sauf spécification contraire, une instruction agissant sur un indicateur le fait comme suit.

- Z** Zéro : Cet indicateur est remis à 1 quand le résultat d'une instruction a une valeur nulle ; il est remis à 0 dans le cas contraire.
- S** Signe : Cet indicateur est remis à 1 quand le bit de plus fort poids du résultat de l'opération a la valeur 1, il est remis à 0 dans le cas contraire.
- P** Parité : Cet indicateur est remis à 1 quand la somme modulo 2 des bits du résultat de l'opération est nulle (c'est-à-dire quand le résultat est pair) ; il est remis à 0 dans le cas contraire (c'est-à-dire quand le résultat est impair)
- C** Retenue : Cet indicateur est remis à 1 quand l'instruction produit une retenue positive (par une addition) ou une retenue négative (par une soustraction ou une comparaison) à partir du bit de plus fort poids ; il est remis à 0 dans le cas contraire.
- AC** Retenue auxiliaire : Cet indicateur est remis à 1 quand l'exécution d'une instruction entraîne un report du bit 3 dans le bit 4 du résultat ; il est remis à 0 dans le cas contraire. Cet indicateur est affecté par les additions, soustractions, incréments, décréments, comparaisons et opérations logiques en simple précision, mais est essentiellement utilisé pour les additions et incréments précédant une instruction DAA (accumulateur de correction décimale) (decimal adjust accumulator)

a) Instructions de transfert de données

MNÉMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPÉRATION
LDA	3	13	$(A) \leftarrow (\langle B_3 \rangle \langle B_2 \rangle)$ transfère le contenu mémoire adressé par B_3 et B_2 dans l'accumulateur
LDAXB	1	7	$(A) \leftarrow ((B)(C))$ charge l'accumulateur par le contenu de la mémoire adressée par B et C
LDAXD	1	7	$(A) \leftarrow ((D)(E))$ charge l'accumulateur par le contenu de la mémoire adressée par D et E
LHLD $\langle B_2 \rangle$ $\langle B_3 \rangle$	3	16	$(L) \leftarrow (\langle B_1 \rangle \langle B_2 \rangle)$; $(H) \leftarrow (\langle B_1 \rangle \langle B_2 \rangle + 1)$ les contenus de la mémoire adressée par B_3 , B_2 et B_1 , B_2 incrémenté sont placés respectivement dans L et H
LXIB $\langle B_2 \rangle$ $\langle B_3 \rangle$	3	10	$(C) \leftarrow \langle B_2 \rangle$; $(B) \leftarrow \langle B_3 \rangle$ le deuxième octet est chargé dans C ; le troisième est chargé dans B
LXID $\langle B_2 \rangle$ $\langle B_3 \rangle$	3	10	$(E) \leftarrow \langle B_2 \rangle$; $(D) \leftarrow \langle B_3 \rangle$ le deuxième octet est chargé dans E ; le troisième est chargé dans D
LXIH $\langle B_2 \rangle$ $\langle B_3 \rangle$	3	10	$(L) \leftarrow \langle B_2 \rangle$; $(H) \leftarrow \langle B_3 \rangle$ le deuxième octet est chargé dans H ; le troisième octet est chargé dans L
LXI SP $\langle B_2 \rangle$ $\langle B_3 \rangle$	3	10	$(SP)_- \leftarrow \langle B_2 \rangle$; $(SP)_+ \leftarrow \langle B_3 \rangle$ B_3 est transféré dans les 8 bits les plus significatifs de SP ; B_2 dans les 8 bits les moins significatifs de SP
MOV r_1, r_2	1	5	$(r_1) \leftarrow (r_2)$ charge le registre r_1 avec le contenu de r_2 . Le contenu de r_2 est inchangé
MOV r, M	1	7	$(r) \leftarrow (M)$, charge registre r par la donnée mémoire adressée par H et L
MOV M, r	1	7	$(M) \leftarrow r$, mémorise le contenu de r en mémoire adressé par H et L
MVI r $\langle B_2 \rangle$	2	7	$(r) \leftarrow \langle B_2 \rangle$ charge le registre r par B_2

MNÉMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPERATION
MVI M < B ₂ >	2	10	(M) ← < B ₂ > place l'octet B ₂ en mémoire adressée par H et L
SHLD < B ₂ > < B ₃ >	3	16	(< B ₁ > < B ₂ >) ← (L), (< B ₃ > < B ₂ > + 1) ← (H) les contenus de H et L sont placés en mémoire adressée par B ₃ et B ₂ , et B ₃ et B ₂ + 1
STA < B ₂ > < B ₃ >	3	13	(< B ₃ > < B ₂ >) ← (A) transfère le contenu de l'accumulateur en mémoire adressée par B ₃ et B ₂
STAX B	1	7	((B) (C)) ← (A) place le contenu de l'accumulateur en mémoire adressée par B et C
STAX D	1	7	((D) (E)) ← (A) place le contenu de l'accumulateur en mémoire adressée par D et E
XCHG	1	4	(H) ↔ (D), (E) ↔ (L) échange le contenu de H, L avec D, E

b) Instructions d'opérations arithmétiques

ACI < B ₂ >	2	7	(A) ← (A) + < B ₂ > + (carry) ADDITION avec carry (report)
ADC r	1	4	(A) ← (A) + (r) + (carry) additionne r, l'accumulateur et carry, résultat dans l'accumulateur ; le résultat agit sur les indicateurs
ADC M	1	7	(A) ← (A) + (M) + (carry) ADD avec carry (report)
ADD r	1	4	(A) ← (A) + (r), additionne r et l'accumulateur avec résultat dans l'accumulateur. Le résultat agit sur les indicateurs
ADD M	1	7	(A) ← (A) + (M) addition
ADI < B ₂ >	2	7	(A) ← (A) + < B ₂ > addition

MNEMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPERATION
DAA	1	4	ajustement Décimal de l'accumulateur. Le contenu de l'accumulateur, résultant d'une opération sur des opérandes codés en B.C.D. est ajusté pour donner deux digits B.C.D., d'après les règles suivantes : $\begin{array}{cccc} 7 & 6 & 5 & 4 \\ \hline \boxed{a} & & & \end{array} \quad \begin{array}{cccc} 3 & 2 & 1 & 0 \\ \hline \boxed{b} & & & \end{array} : \text{accumulateur}$ Si $b \geq 10$ ou si l'indicateur du carry du bit 3 (CY_3) = 1 alors $b = b + 6$ avec carry transféré sur a. Si $a \geq 10$ ou si l'indicateur du carry du bit 7 (CY_7) = 1, ou encore si $b \geq 10$ avec $a = 9$ alors $a = a + 6$.
DAD SP	1	10	$(H)(L) \leftarrow (H)(L) + (SP)$ additionne le contenu de SP et de HL, place le résultat dans H, L ; report final (carry) est mis à « 1 »
DAD B	1	10	$(H)(L) \leftarrow (H)(L) + (B)(C)$
DAD H	1	10	$(H)(L) \leftarrow (H)(L) + (H)(L)$
DAD D	1	10	$(H)(L) \leftarrow (H)(L) + (D)(E)$
INR r	1	5	$(r) \leftarrow (r) + 1$ le registre r est incrémenté le résultat agit sur les indicateurs excepté carry
DCR r	1	5	$(r) \leftarrow (r) - 1$ le registre r est décrémenté ; le résultat agit sur les indicateurs excepté carry
INRM	1	10	$(M) \leftarrow (M) + 1$ le contenu de la mémoire adressé par H et L est incrémenté
DCRM	1	10	$(M) \leftarrow (M) - 1$ le contenu de la mémoire adressé par H et L est décrémenté
INXB	1	5	$(B)(C) \leftarrow (B)(C) + 1$ le contenu de la paire B, C est incrémenté. Les indicateurs ne sont pas affectés
INXH	1	5	$(H)(L) \leftarrow (H)(L) + 1$
INXD	1	5	$(D)(E) \leftarrow (D)(E) + 1$
INXSP	1	5	$(SP) \leftarrow (SP) + 1$

MNÉMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPÉRATION
DCXB	1	5	$(B)(C) \leftarrow (B)(C) - 1$
DCXH	1	5	$(H)(L) \leftarrow (H)(L) - 1$
DCXD	1	5	$(D)(E) \leftarrow (D)(E) - 1$
DCXSP	1	5	$(SP) \leftarrow (SP) - 1$
SUI < B ₂ >	2	7	$(A) \leftarrow (A) - \langle B_2 \rangle$ soustraction
SBI < B ₂ >	2	7	$(A) \leftarrow (A) - \langle B_2 \rangle -$ (carry) soustraction avec retenu
SUB r	1	4	$(A) \leftarrow (A) - (r)$ soustrait r à l'accumulateur ; résultat dans l'accumulateur. Le résultat a agit sur les indicateurs
SBB r	1	4	$(A) \leftarrow (A) - (r) -$ (carry) retranche r et carry à l'accumulateur ; résultat dans l'accumula- teur. Le résultat agit sur les indicateurs
SUB M	1	7	$(A) \leftarrow (A) - (M)$ soustraction
SBB M	1	7	$\bar{A} \leftarrow (A) - (M) -$ (carry) soustraction avec carry

c) Instructions d'opérations logiques

ANA r	1	4	$(A) \leftarrow (A) \wedge (r)$ place le produit logique de r et de l'accumulateur dans l'accumulateur ; remet à zéro carry et à un l'accumulateur Carry
ORA r	1	4	$(A) \leftarrow (A) \vee (r)$ place le « ou logique » entre r et l'accumulateur dans l'accumulateur ; remet à zéro carry et auxiliaire Carry
XRA r	1	4	$(A) \leftarrow (A) \vee (r)$ place le « ou exclusif » entre r et l'accumulateur dans l'accumulateur ; remet à zéro carry et auxiliaire carry
CMP r	1	4	$(A) - (r)$ compare r et l'accumulateur dont le contenu est inchangé si $A = r$ le drapeau « zéro » passe à 1 si $A < r$ la hascule de l'indicateur « carry » passe à « 1 »

MNÉMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPERATION
ANA M	1	7	$(A) \leftarrow (A) \wedge (M)$ « ET » logique
XRA M	1	7	$(A) \leftarrow (A) \vee (M)$ « OU » exclusif
ORA M	1	7	$(A) \leftarrow (A) \vee (M)$ « OU » logique
CMP M	1	7	$(A) - (M)$ comparaison
CMA	1	4	$(A) \leftarrow (\bar{A})$ le contenu de l'accumulateur est complétement (n'affecte par les indicateurs)
CMC	1	4	$(\text{carry}) \leftarrow \overline{(\text{carry})}$ complémente le carry
STC	1	4	$(\text{carry}) = \text{« 1 »}$ met à « 1 » le carry
ANI < B ₂ >	2	7	$(A) \leftarrow (A) \wedge B_2$ « ET » logique
XRI B ₂	2	7	$(A) \leftarrow (A) \vee B_2$ « OU » exclusif
ORI B ₂	2	7	$(A) \leftarrow (A) \vee B_2$ « OU » logique
CPI B ₂	2	7	$(A) - \text{« } B_2 \text{ »}$ comparaison
RLC	1	4	$A_{m+1} \leftarrow A_m$; $A_0 \leftarrow A_7$; $(\text{carry}) \leftarrow A_7$ décalage de l'accumulateur vers la gauche transfert A ₇ dans A ₀ et dans carry
RAR	1	4	$A_m \leftarrow A_{m+1}$; $A_7 \leftarrow (\text{carry})$; $(\text{carry}) \leftarrow A_0$ décalage de l'accumulateur vers la droite ; transfert carry dans A ₇ et A ₀ dans carry
RRC	1	4	$A_m \leftarrow A_{m+1}$; $A_7 \leftarrow A_0$; $(\text{carry}) \leftarrow A_0$ transfert de A ₀ dans A ₇ et dans carry ; décalage vers la droite de l'accumulateur
RAL	1	4	$A_{m+1} \leftarrow A_m$; $A_0 \leftarrow (\text{carry})$; $(\text{carry}) \leftarrow A_7$ décalage vers la gauche de l'accumulateur ; carry est transféré dans A ₀ et A ₇ dans carry

d) Instructions de branchement

MNEMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPÉRATION
CALL < B ₂ > < B ₁ >	3	17	((SP - 1)) ((SP - 2)) ← (PC), (SP) ← (SP) - 2 (PC) ← < B ₁ > < B ₂ > ; transfert du contenu de PC dans la pile mémoire adressée par SP ; SP est décrémenté deux fois, puis saut inconditionnel à l'adresse < B ₁ > < B ₂ >
CC < B ₂ > < B ₁ >	3	11/17	si (carry) = « 1 » ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2 ; (PC) ← < B ₁ > < B ₂ > sinon (PC) ← (PC) + 3
CNC < B ₂ > < B ₁ >	3	11/17	si (carry) = « 0 » ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2 ; (PC) ← < B ₁ > < B ₂ > sinon (PC) ← (PC) + 3
CZ < B ₂ > < B ₁ >	3	11/17	si (zéro) = « 1 » ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2 ; (PC) ← < B ₁ > < B ₂ > sinon (PC) ← (PC) + 3
CNZ < B ₂ > < B ₁ >	3	11/17	si (zéro) = « 0 » ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2 ; (PC) ← < B ₁ > < B ₂ > sinon (PC) ← (PC) + 3
CP < B ₂ > < B ₁ >	3	11/17	si (signe) = « 0 » ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2, (PC) ← < B ₁ > < B ₂ > sinon (PC) ← (PC) + 3
CM < B ₂ > < B ₁ >	3	11/17	si (signe) = 1 ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2, (PC) ← < B ₁ > < B ₂ > sinon (PC) ← (PC) + 3
CPE < B ₂ > < B ₁ >	3	11/17	si (parité) = 1 ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2, (PC) ← < B ₁ > < B ₂ > sinon (PC) ← (PC) + 3

MNEMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPERATION
CPO < B ₂ > < B ₃ >	3	11/17	si (parité) = 0 ; ((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2 ; (PC) ← < B ₃ > < B ₂ > sinon (PC) ← (PC) + 3
RET	1	10	(PC) ← ((SP)) ((SP + 1)), (SP) ← (SP) + 2 retour à l'instruction mémoire adressée par la dernière valeur introduite dans la pile adressée par SP. Le contenu de SP est incrémenté 2 fois
RC	1	5/11	si (carry) = « 1 »
RNC	1	5/11	si (carry) = « 0 »
RZ	1	5/11	si (zéro) = « 1 »
RNZ	1	5/11	si (zéro) = « 0 »
RP	1	5/11	si (signe) = « 1 »
RM	1	5/11	si (signe) = « 0 »
RPE	1	5/11	si (parité) = « 1 »
RPO	1	5/11	si (parité) = « 0 »
RST	1	11	((SP - 1)) ((SP - 2)) ← (PC) (SP) ← (SP) - 2 (PC) ← (0 × 000 × 0000 × 00AAA(00)) le code de RST étant 11AAA111
JMP < B ₂ > < B ₃ >	3	10	(PC) ← < B ₃ > < B ₂ >, saut inconditionnel à l'instruction adressée par B ₂ et B ₃
JC < B ₂ > < B ₃ >	3	7/10	si (carry) = « 1 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) ← (PC) + 3
JNC < B ₂ > < B ₃ >	3	7/10	si (carry) = « 0 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) = (PC) + 3
JZ < B ₂ > < B ₃ >	3	7/10	si (zéro) = « 1 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) = (PC) + 3

Dans tous ces cas si
la condition est
vérifiée alors :

(PC) ← ((SP)) ((SP + 1))
(SP) ← (SP) + 2
sinon (PC) ← (PC) + 1

MNÉMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPÉRATION
JNZ < B ₂ > < B ₁ >	3	7/10	si (zéro) = « 0 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) = (PC) + 3
JP < B ₂ > < B ₃ >	3	7/10	si (signe) = « 0 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) = (PC) + 3
JM < B ₂ > < B ₃ >	3	7/10	si (signe) = « 1 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) = (PC) + 3
JPE < B ₂ > < B ₃ >	3	7/10	si (parity) = « 1 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) = (PC) + 3
JPO < B ₂ > < B ₃ >	3	7/10	si (parity) = « 0 » ; (PC) ← < B ₃ > < B ₂ > sinon (PC) = (PC) + 3
PCHL	1	5	(PC) ← (H)(L) saut indirect
<i>e) Instructions machines, entrées sorties de pile (stack)</i>			
EI	1	4	autorisation d'interruption
DI	1	4	interdiction d'interruption
HLT	1	7	après exécution de cette instruction le processeur est arrêté ; tous les registres et compteurs restent dans l'état antérieur
NOP	1	4	pas d'opération
IN < B ₂ >	2	10	(A) ← (entrée) la donnée présente sur le bus de donnée du circuit E/S (adressé par < B ₂ >) est chargée dans l'accumulateur
OUT < B ₂ >	2	10	(sortie) ← (A) ; le contenu de l'accumulateur est délivré sur le bus de donnée du circuit d'E/S (adressé par < B ₂ >)

MNEMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPERATION
PUSH PSW	1	10	$((SP) - 1) \leftarrow (A)$; $((SP) - 2) \leftarrow (F)$; $(SP) \leftarrow (SP) - 2$ sauvegarde le contenu de l'accumulateur et de F (indicateurs) dans le stack adressé par SP. Le contenu de SP est décrémenté deux fois. Le mot (F) apparaît comme suit : $D_0 = CY_2$ (carry) ; $D_1 = "1"$; $D_2 = \text{parité}$; $D_3 = "0"$; $D_4 = CY_1$; $D_5 = "0"$; $D_6 = \text{zéro}$; $D_7 = \text{signe}$
PUSH B	1	10	$((SP) - 1) \leftarrow (B)$; $((SP) - 2) \leftarrow (C)$; $(SP) \leftarrow (SP) - 2$
PUSH D	1	10	$((SP) - 1) \leftarrow (D)$; $((SP) - 2) \leftarrow (E)$; $(SP) \leftarrow (SP) - 2$
PUSH H	1	10	$((SP) - 1) \leftarrow (H)$; $((SP) - 2) \leftarrow (L)$; $(SP) \leftarrow (SP) - 2$
POP PSW	1	10	$(F) \leftarrow ((SP))$; $(A) \leftarrow ((SP) + 1)$; $(SP) \leftarrow (SP) + 2$ restitue les dernières valeurs, introduites dans la pile adressée par SP, dans A et F. SP est incrémenté deux fois
POP B	1	10	$(C) \leftarrow ((SP))$; $(B) \leftarrow ((SP) + 1)$; $(SP) \leftarrow (SP) + 2$
POP D	1	10	$(E) \leftarrow ((SP))$; $(D) \leftarrow ((SP) + 1)$; $(SP) \leftarrow (SP) + 2$
POP H	1	10	$(L) \leftarrow ((SP))$; $(H) \leftarrow ((SP) + 1)$; $(SP) \leftarrow (SP) + 2$
SPHL	1	5	$(SP) \leftarrow (H)(L)$ transfère le contenu de H et L dans SP
XTHL	1	18	$(L) \leftrightarrow ((SP))$, $(H) \leftrightarrow ((SP) + 1)$ échange les contenus de H et L avec les 2 dernières valeurs mémorisées dans la pile adressée par SP qui reste inchangé

MNÉMONIQUE	OCTETS	CYCLES	DESCRIPTION DE L'OPÉRATION								
RIM	1	4	<p>L'accumulateur est chargé par les huit bits suivants</p> <table border="1"> <tr> <td>SID</td> <td>I7.5</td> <td>I6.5</td> <td>I5.5</td> <td>IE</td> <td>M7.5</td> <td>M6.5</td> <td>M5.5</td> </tr> </table> <p> SID = bit présent sur l'entrée série In.5 = 1 : interruption n.5 en cours de traitement IE = 1 : les interruptions sont autorisées Mn.5 = 1 : l'interruption n.5 est temporairement masquée c'est-à-dire interdite </p>	SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5
SID	I7.5	I6.5	I5.5	IE	M7.5	M6.5	M5.5				
SIM	1	4	<p>masquage des interruptions d'après le contenu de l'accumulateur</p> <table border="1"> <tr> <td>SOD</td> <td>SOE</td> <td>X</td> <td>R7.5</td> <td>MSE</td> <td>M7.5</td> <td>M6.5</td> <td>M5.5</td> </tr> </table> <p> SOD = bit à présenter sur la sortie série SOE = « 1 » : la sortie série est autorisée R7.5 = « 1 » : remise à « 0 » de la bascule qui mémorise les impulsions sur l'entrée RST 7.5 MSE = « 1 » : les masquages sont autorisés Mn.5 = « 1 » : l'interruption n.5 est temporairement masquée </p>	SOD	SOE	X	R7.5	MSE	M7.5	M6.5	M5.5
SOD	SOE	X	R7.5	MSE	M7.5	M6.5	M5.5				

Appendix A

INSTRUCTION SUMMARIES

This Appendix summarizes all of the instruction opcodes for convenient reference. Instructions are listed (I) by functional categories, (II) in ascending order of hex opcode value, and (III) alphabetically by standard mnemonic. The 8080A/8085A Instruction Description in Chapter 3 of this Handbook provides detailed descriptions of the operation of each processor instruction.

There are 256 possible combinations of the eight bits of the instruction opcode. Twelve are unassigned and should not be used. They are:

HEX	BINARY
08	00001000
10	00010000
18	00011000
20	00100000
28	00101000
30	00110000
38	00111000
CB	11001011
D9	11011001
DD	11011101
ED	11101101
FD	11111101

8080A/8085A INSTRUCTION SUMMARY BY FUNCTIONAL GROUPS

Data Transfer							
HEX	MNEMONIC	HEX	MNEMONIC	HEX	MNEMONIC	HEX	MNEMONIC
40	MOV B,B	58	MOV E,B	70	MOV M,B	1A	LDAX D
41	MOV B,C	59	MOV E,C	71	MOV M,C	2A	LHLD
42	MOV B,D	5A	MOV E,D	72	MOV M,D	3A	LDA
43	MOV B,E	5B	MOV E,E	73	MOV M,E	02	STAX B
44	MOV B,H	5C	MOV E,H	74	MOV M,H	12	STAX D
45	MOV B,L	5D	MOV E,L	75	MOV M,L	22	SHLD
46	MOV B,M	5E	MOV E,M	77	MOV M,A	32	STA
47	MOV B,A	5F	MOV E,A	78	MOV A,B	01	LXI B
48	MOV C,B	60	MOV H,B	79	MOV A,C	11	LXI D
49	MOV C,C	61	MOV H,C	7A	MOV A,D	21	LXI H
4A	MOV C,D	62	MOV H,D	7B	MOV A,E	31	LXI SP
4B	MOV C,E	63	MOV H,E	7C	MOV A,H	F9	SPHL
4C	MOV C,H	64	MOV H,H	7D	MOV A,L	E3	XTHL
4D	MOV C,L	65	MOV H,L	7E	MOV A,M	EB	XCHG
4E	MOV C,M	66	MOV H,M	7F	MOV A,A	D3	OUT
4F	MOV C,A	67	MOV H,A	06	MVI B	DB	IN
50	MOV D,B	68	MOV L,B	0E	MVI C	C5	PUSH B
51	MOV D,C	69	MOV L,C	16	MVI D	D5	PUSH D
52	MOV D,D	6A	MOV L,D	1E	MVI E	E5	PUSH H
53	MOV D,E	6B	MOV L,E	26	MVI H	F5	PUSH PSW
54	MOV D,H	6C	MOV L,H	2E	MVI L	C1	POP B
55	MOV D,L	6D	MOV L,L	36	MVI M	D1	POP D
56	MOV D,M	6E	MOV L,M	3E	MVI A	E1	POP H
57	MOV D,A	6F	MOV L,A	0A	LDAX B	F1	POP PSW

Arithmetic							
HEX	MNEMONIC						
80	ADD B	C6	ADI	9E	SBB M	3C	INR A
81	ADD C	CE	ACI	9F	SBB A	03	INX B
82	ADD D	90	SUB B	D6	SUI	13	INX D
83	ADD E	91	SUB C	DE	SBI	23	INX H
84	ADD H	92	SUB D	09	DAD B	33	INX SP
85	ADD L	93	SUB E	19	DAD D	05	DCR B
86	ADD M	94	SUB H	29	DAD H	0D	DCR C
87	ADD A	95	SUB L	39	DAD SP	15	DCR D
88	ADC B	96	SUB M	27	DAA	1D	DCR E
89	ADC C	97	SUB A	04	INR B	25	DCR H
8A	ADC D	98	SBB B	0C	INR C	2D	DCR L
8B	ADC E	99	SBB C	14	INR D	35	DCR M
8C	ADC H	9A	SBB D	1C	INR E	3D	DCR A
8D	ADC L	9B	SBB E	24	INR H	0B	DCX B
8E	ADC M	9C	SBB H	2C	INR L	1B	DCX D
8F	ADC A	9D	SBB L	34	INR M	2B	DCX H
						3B	DCX SP

Logical							
HEX	MNEMONIC						
A0	ANA B	A9	XRA C	B2	ORA D	B5	CMP E
A1	ANA C	AA	XRA D	B3	ORA E	BC	CMP H
A2	ANA D	AB	XRA E	B4	ORA H	ED	CMP L
A3	ANA E	AC	XRA H	B5	ORA L	BE	CMP M
A4	ANA H	AD	XRA L	B6	ORA M	BF	CMP A
A5	ANA L	AE	XRA M	B7	ORA A	FE	CPI
A6	ANA M	AF	XRA A	F6	ORI	07	RLC
A7	ANA A	FE	XRI	B8	CMP B	0F	RRC
E6	ANI	B0	ORA B	B9	CMP C	17	RAL
A8	XRA B	B1	ORA C	BA	CMP D	1F	RAR
						2F	CMA

Branching							
HEX	MNEMONIC						
C3	JMP	D7	RST 2	EC	CPE		
C2	JNZ	DF	RST 3	F4	CP		
CA	JZ	E7	RST 4	FC	CM		
D2	JNC	EF	RST 5	C9	RET		
DA	JC	F7	RST 6	CD	RNZ		
E2	JPO	FF	RST 7	C8	RZ		
EA	JPE	CD	CALL	D0	RNC		
F2	JP	C4	CNZ	D8	RC		
FA	JM	CC	CZ	E0	RPO		
E9	PCHL	D4	CNC	E8	RPE		
C7	RST 0	DC	CC	F0	RP		
CF	RST 1	E4	CPO	FB	RM		

Control	
HEX	MNEMONIC
00	NOP
76	HLT
F3	DI
FB	EI
37	STC
3F	CMC

II 8080A/8080A INSTRUCTION SUMMARY
HEXADECIMAL ORDER

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

III 8080A/9080A INSTRUCTION SUMMARY
ALPHABETICAL ORDER

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

