Unspecified 8085 op codes enhance programming

by Wolfgang Dehnhardt and Villy M. Sorensen
GSI, Darmstadt, and Sorensen Software, Seeheim, West Germany

Ten operating codes and two flag bits previously unknown to most users of the 8085 microprocessor will enable programmers to write more efficient routines. The new members of the instruction set, which were stumbled upon during the testing of an assembler-disassembler module, include seven op codes that involve the processing of register pairs, two that involve jump operations with one new flag bit, and one that performs a conditional restart on the overflow indication of the other flag bit.

The seven register pair instructions (all with 16-bit operands) consist of a double subtraction, a rotate, a shift, indirect loading and storing of a word, and two offset operations. Either BC, DE, HL, or SP are the designated register pairs used in these op codes.

The mnemonic names of the instructions have been selected to be compatible with the 8085's existing mnemonics. In the double subtraction (DSUB), register pair BC is subtracted from HL. This instruction thus performs the opposite task of DAD B, a well-known instruction. The instruction RDEL rotates register pair DE left 1 bit through the carry. ARHL is an arithmetic shift to the right of HL. It serves to divide HL by 2, except in cases where HL is −1.

All 16 bits of register pair HL can be stored indirectly at the address contained in the DE pair by specifying instruction SHLX. To load HL, LHLX must be employed.

As an example of how this instruction can be used to cut instruction steps, consider the common sequence used for a routine table jump shown in part (a) of the figure. By assigning the register DE for HL and using the LHLX instruction, this sequence can be replaced by the much simpler arrangement shown at the bottom of part (a).

As for adding the contents of register pairs with an additional byte (offset), DE can be loaded with HL plus the byte by selecting the instruction LDHI, which simplifies array addressing. Usually, the architecture of 8080-type systems dictate the addressing of arrays in what are called pages of 256 bytes. This restriction means that the starting address of an array must be placed near the beginning of a page. A typical call is as shown in part (b) of the figure.

The page limitation is bypassed using the LDHI instruction code and constant indexes. The starting address of the array can now be placed anywhere, and addressing occurs as shown at the bottom of part b.

Any additional byte can be combined with register-pair SP in DE if instruction LDSI is specified. This instruction is designed for operating system routes that transfer arguments on the stack. An example sequence, shown in (c), stores HL into the 16-bit word located as the second item below the top of the stack.

The jump and restart instructions work in conjunction with the two discovered flag bits, X5 and V. Op codes JX5 and JNX5 jump depending on the state of the X5 flag. Op code RSTV makes a restart call to hexadecimal address 40 if the V flag is set; otherwise it functions as a no-operation instruction.

Flag bit V indicates a 2's complement overflow condition for 8- and 16-bit arithmetical operations. Flag bit X5 has been named for its position in the condition code byte and not for its function. It does not resemble any normal flag bit. The only use for this bit found thus far are as an unsigned overflow indicator resulting from a data change of FFFF to 0000 on executing the instruction of INX and as an unsigned underflow indicator from a data change of 0000 to FFFF on executing DCX.

The new 8085 instructions are outlined in the table. □

(a)

Source statement		Comments
MOV	E, M	;ROUTINE ADR LOW BYTE
INX	H	;HL = TABLE ADR
MOV	D, M	;ROUTINE ADR HIGH BYTE
XCHG		;DE = ROUTINE ADR
PCHL		;GO TO ROUTINE ADR

⬇

LHLX		;DE = TABLE ADR
PCHL		;HL = ROUTINE ADR

(b)

Source statement		Comments
LXI	H, ARRAY	;ARRAY BASE ADR
MVI	L, INDEX	;8-BIT INDEX, HL = ARRAY ADR

⬇

LXI	H, ARRAY	;ARRAY BASE ADR
LDHI	INDEX	;8-BIT INDEX, DE = ARRAY ADR

(c)

Source statement		Comments
LDSI	2	;DE = SP + 2
SHLX		;REPLACE 2. ITEM ON STACK

Options. Newly discovered operating codes for 8085 shown in table enables the writing of more efficient programs. Program for table jump (a, top) may be reduced significantly when new instruction (a, bottom) are implemented. Array routines (b, top) can be rewritten (b, bottom) so that arrays can be addressed across page boundaries. Data words can be entered at any point in a stack register (c).

NEW 8085 INSTRUCTIONS

NEW CONDITION CODES:

V = bit 1 2's complement overflow

X5 = bit 5 Underflow (DCX) or overflow (INX)

X5 = 01·02 + 01·R + 02·R, where
01 = sign of operand 1, 02 = sign of operand 2,
R = sign of result. For subtraction and comparisons, replace 02 with $\overline{02}$.

Condition code format							
S	Z	X5	AC	0	P	V	C

DSUB (double subtraction)

(H) (L) = (H) (L) − (B) (C)

The contents of register pair B and C are subtracted from the contents of register pair H and L. The result is placed in register pair H and L. All condition flags are affected.

0 0 0 0 1 0 0 0

(08)

cycles:	3
states:	10
addressing:	register
flags:	Z, S, P, CY, AC, X5, V

ARHL (arithmetic shift of H and L to the right)

(H7=H7); (Hn−1) = (Hn)
(L7=Ho); (Ln−1) = (Ln); (CY) = (Lo)

The contents of register pair H and L are shifted right one bit. The uppermost bit is duplicated and the lowest bit is shifted into the carry bit. The result is placed in register pair H and L. Note: only the CY flag is affected.

0 0 0 1 0 0 0 0

(10)

cycles:	2
states:	7
addressing:	register
flags:	CY

RDEL (rotate D and E left through carry)

(Dn+1) = (Dn); (Do) = (E7)
(CY) = (D7); (En+1) = (En); (Eo) = (CY)

The contents of register pair D and E are rotated left one position through the carry flag. The low-order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high-order bit. Only the CY and the V flags are affected.

0 0 0 1 1 0 0 0

(18)

cycles:	3
states:	10
addressing:	register
flags:	CY, V

LDHI (load D and E with H and L plus immediate byte)

(D) (E) = (H) (L) + (byte 2)

The contents of register pair H and L are added to the immediate byte. The result is placed in register pair D and E. Note: no condition flags are affected.

0 0 1 0 1 0 0 0
data

(28)

cycles.	3
states:	10
addressing:	immediate register
flags:	none

LDSI (load D and E with SP plus immediate byte)

(D) (E) = (SPH)(SPL) + (byte 2)

The contents of register pair SP are added to the immediate byte. The result is placed in register pair D and E. Note. no condition flags are affected.

0 0 1 1 1 0 0 0
data

(38)

cycles:	3
states:	10
addressing:	immediate register
flags:	none

RSTV (restart on overflow)

If (V):
((SP)−1) = (PCH)
((SP)−2) = (PCL)
(SP) = (SP)−2
(PC) = 40 hex

If the overflow flag V is set, the actions specified above are performed; otherwise control continues sequentially.

1 1 0 0 1 0 1 1

(C8)

cycles:	1 or 3
states:	6 or 12
addressing:	register indirect
flags:	none

SHLX (store H and L indirect through D and E)

((D)(E)) = (L)
((D)(E)+1) = (H)

The contents of register L are moved to the memory location whose address is in register pair D and E. The contents of register H are moved to the succeeding memory location.

1 1 0 1 1 0 0 1

(D9)

cycles:	3
states:	10
addressing:	register indirect
flags:	none

JNX5 (jump on not X5)

If (not X5):
(PC) = (byte 3) (byte 2)

If the X5 flag is reset, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise control continues sequentially.

1 1 0 1 1 1 0 1
low-order address
high-order address

(DD)

cycles:	2 or 3
states:	7 or 10
addressing:	immediate
flags:	none

LHLX (load H and L indirect through D and E)

(L) = ((D)(E))
(H) = ((D)(E)+1)

The content of the memory location whose address is in D and E, are moved to register L. The contents of the succeeding memory location are moved to register H.

1 1 1 0 1 1 0 1

(ED)

cycles:	3
states:	10
addressing:	register indirect
flags:	none

JX5 (jump on X5)

If (X5):
(PC) = (byte 3) (byte 2)

If the X5 flag is reset, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise control continues sequentially.

1 1 1 1 1 1 0 1
low-order address
high-order address

(FD)

cycles:	2 or 3
states:	7 or 10
addressing:	immediate
flags:	none