M O N R O B O T X I

A D V A N C E D P R O G R A M M I N G I N F O R M A T I O N

MO-96ADV Rev. 2-69

Special forms of some operation codes are of value to the experienced programmer in order to obtain maximum efficiency in a complex program. In some instances they may permit a completely different approach to the problem at hand. These special forms have not been included in the program manual for reasons of clarity. This write-up describes these special operation codes and gives some examples of their use.

SPECIAL FORMS OF OPERATION CODES

Detract

14(FA)

When the detract command has a fast access address (000-006) the order code should be 14 plus the fast access address. This will save one word time per execution of the instruction.

1406

This command serves as a double function: T004 followed by U500. The programmer is cautioned that FA-6 must not be clear before execution of the command and that a one will be added to FA-5.

1ADR

If the contents of the specified address are greater than the accumulator and the accumulator is not clear, the contents of the specified address are transferred to FA-4 without affecting the accumulator. The contents of FA-5 will not be altered. This is the only means by which information can be transfered from one register to another without affecting the accumulator.

Overflow During Detraction

If FA-5 overflows during a detract command, a one will be subtracted from the accumulator. For example, if a 1406 instruction is used when FA-5 = -1, the contents of the accumulator will be transfered to FA-4, FA-5 becomes zero, and the accumulator will become equal to -1.

Input

2201, 2401, 2801

The Input operation code with a one added to the low order will force a parity error condition when the carriage return code is read. The contents of the accumulator will be 8000 0081 (rather than 0000 0080). All other standard characters are unchanged.

The carriage return when used as an end of input code can be rapidly detected as follows:

```
2401        Input forcing parity error on carriage return code
7ADR        Test for parity error

Where ADR is:
W           Subtract 8000 0081 for carriage return code
6           Jump if zero to exit

2200        Halt, true parity error
```

Similar results will be obtained with special codes which contain a T-1 8's bit, no bit at T-0 1's bit position, and fulfills the requirement of odd parity.

Jump Mark

3401, 3406

When an instruction is created or modified in the accumulator, it is not necessary to store it in proper sequence in general storage prior to execution. If the modified instruction must be saved, T001 3401 will store and execute the modified instruction.

If the instruction need not be saved, it can be executed from the accumulator using the operation code 3406. In this case the contents of register 007 must be 3002 0000 or (Instruction) 3002.

The use of either of these techniques can save a store command (in the case of 3406), the register in which the instruction would have been stored, possibly a jump command, and the execution time since fast access rather than general storage is involved.

3402

The contents of FA-2 are transfered to the instruction register at the same time that the contents of the instruction register are transfered to FA-2. The result, then, is an interchange between FA-2 and the instruction register. The next instructions executed will be the previous contents of FA-2. The progrøm will then sequence to FA-3 unless FA-2 had contained (Instruction) 3002.

Multiply

54(FA)

When the multiply command has a fast access address (000-006) the order code should be 54 plus the fast access address. This will save one word time per execution of the instruction.

Register Unpack By Multiplication

Multiplication of the contents of the accumulator by the constant 0001 0000 is in effect a binary left end around shift of 48 bits. After operation, the previous low order 16 bits of FA-6 (B link) will appear as the high order 16 bits of FA-5 (A link). The previous high order 16 bits of FA-6 (A link) will appear as the low order 16 bits of FA-6 (B link). If a high order bit was present in FA-6 prior to multiplication, the A link of FA-6 will contain 16 1's bits after multiplication, otherwise 16 zero bits will be present.

The constant 0001 0000 can be used to unpack an instruction register, or by moving this single bit right or left to other positions, many variations of register unpacking or shifting may be obtained.

Multiplication of Negative Numbers in FA-6

The number -2^{31} (8000 0000) when in the accumulator will be treated as 2^{31} by the multiply command. When -2^{31} is the contents of any other register address it will be treated as -2^{31} by the multiply command.

End Around Shift

This form of the binary left end around shift will place a bit in the high order position of FA-5 before shifting takes place. It is particularly valuable for 4 bit (sexadecimal) or 6 bit (alpha-numeric) register unpacking. After clearing FA-5, the first shift of 4 or 6 bits (8W08 or 8W20) will create a flag bit in the T0-8 or T1-2 bit position. Subsequent shifts are in the normal form (8U08 or 8U20) followed by a check for a zero accumulator indicating unpacking is complete.

Example:

S07X	Clear FA-5
8W08	Left end around shift, generate a flag (8's bit)
S47X	Output character
8U08	Left end around shift
6	To exit, word unpacked
3	To output character

Output

S4XX, S8XX

Output of even parity or mixed parity characters from FA-5 to the tape punch is possible by the use of the SDXX command instead of SD7X. Prior to output, an odd parity character must be shifted one bit position left and an even parity character must be shifted two bit positions left and a low order bit added:

Odd Parity . . . . 9001, T005, SDXX

Even Parity. . . . . 9001, 9401, T005, SDXX

The value of this special output instruction lies in the ability to reproduce a mixed or even parity punched tape.

Note:

9001	BSL 1 bit
9401	BSL bit and generate a 1's bit on even parity character

No Operation Instructions

The program manual mentions S100 as a no operation instruction (no-op).

The entire list of no operation instructions is:

0200	0700	S100
0300	6400	V006
0600	7400	W400

However, if a B link instruction is to be a no operation command, it should be written as an unconditional jump to the next register (pre-write the automatic jump instruction).

When an instruction is to perform as a no operation command has a variable address caused by some address modification function, the no operation command should be of the 64-- form. Any address added to 64-- will still produce a no operation instruction.

In five channel input or whenever the parity bit is to be ignored because of table look-up, the input character is added to a virgin instruction which has 6400 in the A link, VADR of table character in the B link. If the character produced a parity error, the A-link will become W400, once again a no operation command.

Program for Even or Mixed Parity Tape Reproduction

S5403006 for tape feed
S REG: 000   E REG: 003   Starting address: 002

```
000  9001     BSL 1 bit
     9401     BSL 1 bit and generate a 1's bit on even parity

001  T005     Store character for output
     S4XX     Output Character (Device #2)

002  2400     Read character from tape (Device #2)
     7000     Jump to shift even parity character

003  9001     BSL 1 bit on odd parity character
     3001     Jump to output character
```

Shifting Negative Numbers

Binary or decimal shifts of negative numbers can produce incorrect results in some instances. The rules for a shifting negative numbers are as follows:

80--	Decimal Shift Left: Correct result if shift is one place (8001) and normal conditions for positive numbers are met.
88--	Decimal Shift Right: Incorrect results are always obtained. The value must be complemented to positive form and the results re-complemented.
90--	Binary Shift Left: Correct results until a significant bit (a zero) is shifted into the high-order position.
9U--	Binary Shift Right, Maintain Sign: Correct results are always obtained. Remeber that Remember that "rounding up" takes place with negative: $-3/2 = -2$; $-\frac{1}{2} = -1$

Decimal Extract

Two programmed approaches are available to obtain the low order digits of a large decimal number. If the number of digits in the high order of the number is large, it is best to isolate the high order digits by shift commands, then subtract to obtain the low order digits.

```
Example:  V(A)     The number
          88       Divide off digits not required

          80       Shift back zeros
          T(B)     Save result

          V(A)     Re-load the number
          W(B)     Low order value
```

If the original number was 43, 756 and the 56 is to be extracted, the data would appear as follows:

```
          V         43756
          8802        437

          8002      43700
          T         43700

          V         43756
          W         43700

          FA-6  =      56
```

If the number of digits in the high order of the number is even, it is possible to isolate the low order digits by detracting by an appropriate power of ten.

Example:	FA-6	=	The number
	1		Detract by 10^n to remove high order digits
	FA-6	=	Low order digits

Generation of High Order Bits

9S01, 02, 04, 08, 10, 20, 40 80

High order bits may be generated into a clear accumulator by any of the above commands. The command 9S01 will produce 8000 0000 while 9S80 will produce XX00 0000. This can be a convenient way to produce a flag or count registers.

Reactions of 6400

When 64DR is in the A link, multiplication by 0001 0000 changes 6401 into 3002; adds a bit in T-4, subtracts 4 from T-6, BSR 1 bit in T-7

Changes Occuring in No-Op Codes, If Negative, & If Zero Codes

6400, 7400, 6ADR, 7ADR

When the code 6400 goes thru the instruction register, it becomes 3001; the same result for 7400.

The general ,pattern is that in 6ADR, a 4 is subtracted from the A, D remains the same, and a one is added to the R.

Extra-Special Codes (RAC addition)

0300 3006, 4XXX 3006