MCS-4™ Instruction Set

Those instructions preceded by an asterisk (*) are 2 word instructions that occupy 2 successive locations in ROM

MACHINE INSTRUCTIONS

<table>
<thead>
<tr>
<th>MNEMONIC</th>
<th>OPR D3 D2 D1 D0</th>
<th>OPA D3 D2 D1 D0</th>
<th>DESCRIPTION OF OPERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td>0 0 0 0</td>
<td>0 0 0 0</td>
<td>No operation</td>
</tr>
<tr>
<td>*JCN</td>
<td>0 0 0 1</td>
<td>C1 C2 C3 C4</td>
<td>Jump to ROM address A2 A1 A1 A1 (within the same ROM that contains this JCN instruction) if condition C1 C2 C3 C4 is true, otherwise skip (go to the next instruction in sequence).</td>
</tr>
<tr>
<td>*FIM</td>
<td>0 0 1 0</td>
<td>R R R 0</td>
<td>Fetch immediate (direct) from ROM Data D2 D1 D0 to index register pair location RRR (2)</td>
</tr>
<tr>
<td>SRC</td>
<td>0 0 1 0</td>
<td>R R R 1</td>
<td>Send register control. Send the address (contents of index register pair RRR) to ROM and RAM at X2 and X3 time in the Instruction Cycle.</td>
</tr>
<tr>
<td>FIN</td>
<td>0 0 1 1</td>
<td>R R R 0</td>
<td>Fetch indirect from ROM. Send contents of index register pair location 0 out as an address. Data fetched is placed into register pair location RRR.</td>
</tr>
<tr>
<td>JIN</td>
<td>0 0 1 1</td>
<td>R R R 1</td>
<td>Jump indirect. Send contents of register pair RRR out as an address at A1 and A0 time in the Instruction Cycle.</td>
</tr>
<tr>
<td>*JUN</td>
<td>0 1 0 0</td>
<td>A2 A2 A2 A1</td>
<td>Jump unconditional to ROM address A3 A2 A2 A1.</td>
</tr>
<tr>
<td>*JMS</td>
<td>0 1 0 1</td>
<td>A2 A2 A2 A1</td>
<td>Jump to subroutine ROM address A3 A2 A2 A1. Save old address. (Up 1 level in stack.)</td>
</tr>
<tr>
<td>INC</td>
<td>0 1 1 0</td>
<td>R R R R</td>
<td>Increment contents of register RRRR. (3)</td>
</tr>
<tr>
<td>*ISZ</td>
<td>0 1 1 1</td>
<td>R R R R</td>
<td>Increment contents of register RRRR. If ROM address A2 A1 (within the same ROM that contains this ISZ instruction) is not equal to 0, otherwise skip (go to the next instruction in sequence).</td>
</tr>
<tr>
<td>ADD</td>
<td>1 0 0 0</td>
<td>R R R R</td>
<td>Add contents of register RRRR to accumulator with carry.</td>
</tr>
<tr>
<td>SUB</td>
<td>1 0 0 1</td>
<td>R R R R</td>
<td>Subtract contents of register RRRR to accumulator with borrow.</td>
</tr>
<tr>
<td>LD</td>
<td>1 0 1 0</td>
<td>R R R R</td>
<td>Load contents of register RRRR accumulator.</td>
</tr>
<tr>
<td>XCH</td>
<td>1 0 1 1</td>
<td>R R R R</td>
<td>Exchange contents of index register RRRR and accumulator.</td>
</tr>
<tr>
<td>BBL</td>
<td>1 1 0 0</td>
<td>D D D D</td>
<td>Branch back (down 1 level in stack) and load data DDDD to accumulator.</td>
</tr>
<tr>
<td>LDM</td>
<td>1 1 0 1</td>
<td>D D D D</td>
<td>Load data DDDD to accumulator.</td>
</tr>
</tbody>
</table>

INPUT/OUTPUT AND RAM INSTRUCTIONS

(The RAM's and ROM's operated on in the ID and RAM instructions have been previously selected by the last SRC instruction executed.)

WRM 1 1 1 0 0 0 0 0 Write the contents of the accumulator into the previously selected RAM main memory character.

WMP 1 1 1 0 0 0 0 1 Write the contents of the accumulator into the previously selected RAM output port. (Output Lines)

WR 1 1 1 0 0 0 1 0 Write the contents of the accumulator into the previously selected RAM output port. (Output Lines)

WRM 1 1 1 0 0 0 1 1 Write the contents of the accumulator into the previously selected RAM main memory character.

WR 1 1 1 0 0 1 0 0 Write the contents of the accumulator into the previously selected RAM main memory character.

WR 1 1 1 0 0 1 0 1 Write the contents of the accumulator into the previously selected RAM main memory character.

WR 1 1 1 0 0 1 1 0 Write the contents of the accumulator into the previously selected RAM main memory character.

WR 1 1 1 0 0 1 1 1 Write the contents of the accumulator into the previously selected RAM main memory character.

RBM 1 1 1 0 1 0 0 0 Subtract the previously selected RAM main memory character from accumulator with borrow.

RDR 1 1 1 0 1 0 0 1 Read the contents of the previously selected RAM memory character into the accumulator.

ADM 1 1 1 0 1 0 1 1 Add the previously selected RAM main memory character to accumulator with carry.

RDG 1 1 1 0 1 1 0 0 Read the previously selected RAM status character 0 into accumulator.

RD 1 1 1 0 1 1 1 0 Read the previously selected RAM status character 1 into accumulator.

RD 1 1 1 0 1 1 1 1 Read the previously selected RAM status character 2 into accumulator.

RD 1 1 1 0 1 1 1 1 Read the previously selected RAM status character 3 into accumulator.

ACCUMULATOR GROUP INSTRUCTIONS

CLB 1 1 1 1 0 0 0 0 Clear both, (Accumulator and carry)

CLC 1 1 1 1 0 0 0 1 Clear carry.

IAC 1 1 1 1 0 0 1 0 Increment accumulator.

CMC 1 1 1 1 0 0 1 1 Complement carry.

CMA 1 1 1 1 0 1 0 0 Complement accumulator.

RAL 1 1 1 1 0 1 0 1 Rotate left, (Accumulator and carry)

RAR 1 1 1 1 0 1 1 0 Rotate right, (Accumulator and carry)

TCC 1 1 1 1 0 1 1 1 Transmit carry to accumulator and clear carry.

DAC 1 1 1 1 1 0 0 0 Decrement accumulator.

TCS 1 1 1 1 1 0 1 0 Transfer carry subtract and clear carry.

STC 1 1 1 1 1 0 1 1 Set carry.

DAA 1 1 1 1 1 1 0 0 Decimal adjust accumulator.

KBP 1 1 1 1 1 1 0 0 Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code.

DCL 1 1 1 1 1 1 1 0 Designate command line.

NOTES:

(1) The condition code is assigned as follows:

C1 = 1 Invert jump condition 
C2 = 1 Jump if accumulator is zero 
C4 = 1 Jump if test signal is a 0 
C3 = 0 Not invert jump condition 
C5 = 1 Jump if carry/link is a 1

(2) RRR is the address of 1 of 8 index register pairs in the CPU.

(3) The RAM's and ROM's operated on in the ID and RAM instructions are RRRR and RRRR respectively.

Each RAM chip has 4 registers, each with twenty 4-bit characters subdivided into 16 main memory characters and 4 status characters. Chip number, RAM register and main memory character are addressed by an SRC instruction. For the selected chip and register, however, status character locations are selected by the instruction code (IOPA).