D16/M Instruction Set

Addressing Modes

The D16/M processor has five addressing modes, including Implied Addressing (0 address instructions), as follows:

INSTR   --  **Implied Addressing.**
The instruction operates on the CPU, and there is no memory operand.

INSTR  m  **Immediate Addressing.**
The memory word following the instruction opcode is the operand for the instruction.

INSTR  (m)  **Direct Addressing.**
The memory word following the opcode is the address of the operand.

INSTR  [m]  **Indirect Addressing.**
The memory word following the opcode is the address of the address of the operand.

INSTR  [m++]  **Post-Indexed Indirect Addressing.**
As in Indirect Addressing, the memory word following the opcode is the address of the address of the operand. After the instruction executes, the address of the operand is incremented by one.

D16/M Zero-Address Instructions

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Opcode</th>
<th>Function</th>
<th>Cycles (Extended)</th>
</tr>
</thead>
<tbody>
<tr>
<td>INC</td>
<td>001A</td>
<td>Increment Accumulator</td>
<td>7 (1)</td>
</tr>
<tr>
<td>DEC</td>
<td>001B</td>
<td>Decrement Accumulator</td>
<td>7 (1)</td>
</tr>
<tr>
<td>COM</td>
<td>001C</td>
<td>Ones Complement Accumulator</td>
<td>6 (1)</td>
</tr>
<tr>
<td>NEG</td>
<td>001D</td>
<td>Twos Complement Accumulator</td>
<td>6 (1)</td>
</tr>
<tr>
<td>SET</td>
<td>001E</td>
<td>Set Accumulator (AC &lt; FFFF)</td>
<td>6 (1)</td>
</tr>
<tr>
<td>CLR</td>
<td>001F</td>
<td>Clear Accumulator (AC &lt; 0000)</td>
<td>6 (1)</td>
</tr>
<tr>
<td>SHL</td>
<td>0020</td>
<td>Shift Left Accumulator (CF &lt; AC15, AC0 &lt; 0)</td>
<td>6 (1)</td>
</tr>
<tr>
<td>LSR</td>
<td>0021</td>
<td>Logical Shift Right Accumulator (AC15 &lt; 0, CF &lt; AC0)</td>
<td>6 (1)</td>
</tr>
<tr>
<td>ASR</td>
<td>0022</td>
<td>Arithmetic Shift Right Accumulator (AC15 &lt; AC15, CF &lt; AC0)</td>
<td>6 (1)</td>
</tr>
<tr>
<td>ROL</td>
<td>0023</td>
<td>Rotate Left Accumulator (CF &lt; AC15, AC0 &lt; CF)</td>
<td>6 (1)</td>
</tr>
<tr>
<td>ROR</td>
<td>0024</td>
<td>Rotate Right Accumulator (AC15 &lt; CF, CF &lt; AC0)</td>
<td>6 (1)</td>
</tr>
<tr>
<td>LSW</td>
<td>0025</td>
<td>Load Accumulator with Switch Register</td>
<td>6 (1)</td>
</tr>
</tbody>
</table>
### SCF 0026 Set Carry Flag
(CF < 1)

### CCF 0027 Clear Carry Flag
(CF < 0)

### PSA 0028 Push Accumulator onto Stack
7 (2)

### POA 0029 Pop Accumulator off of Stack
6 (2)

### PSF 002A Push Flag Register onto Stack
7 (2)

### POF 002B Pop Flag Register off of Stack
6 (2)

### ENI 002C Enable Interrupts
5 (1)

### DII 002D Disable Interrupts
5 (1)

### RTN 002E Return from Subroutine
6 (2)

### RTI 002F Return from Interrupt, Enabling Interrupts
10 (4)

### HLT 00FF Processor Halt
5 (1)

### NOP 0030 No Operation
6 (1)

---

### D16/M Memory Reference (One Address) Instructions

<table>
<thead>
<tr>
<th>Mnemonic (Extended)</th>
<th>Opcode</th>
<th>Function</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDA m0301</td>
<td></td>
<td>Load Accumulator (Immediate)</td>
<td>8 (2)</td>
</tr>
<tr>
<td>LDA (m) 0101</td>
<td></td>
<td>Load Accumulator (Direct)</td>
<td>8 (3)</td>
</tr>
<tr>
<td>LDA [m] 0501</td>
<td></td>
<td>Load Accumulator (Indirect)</td>
<td>9 (4)</td>
</tr>
<tr>
<td>LDA [m++] 0D01</td>
<td></td>
<td>Load Accumulator (Indirect Post-Indexed)</td>
<td>12 (5)</td>
</tr>
<tr>
<td>LDS m0302</td>
<td></td>
<td>Load Stack Pointer</td>
<td>8 (2)</td>
</tr>
<tr>
<td>LDS (m) 0102</td>
<td></td>
<td></td>
<td>8 (3)</td>
</tr>
<tr>
<td>LDS [m] 0502</td>
<td></td>
<td></td>
<td>9 (4)</td>
</tr>
<tr>
<td>LDS [m++] 0D02</td>
<td></td>
<td></td>
<td>12 (5)</td>
</tr>
<tr>
<td>STA (m) 0103</td>
<td></td>
<td>Store Accumulator</td>
<td>7 (3)</td>
</tr>
</tbody>
</table>
STA [m] 0503 8 (4)
STA [m++] 0D03 11 (5)

STS (m) 0104 Store Stack Pointer 7 (3)
STS [m] 0504 8 (4)
STS [m++] 0D04 11 (5)

ADD m 0305 Add to Accumulator 7 (2)
ADD (m) 0105 8 (3)

ADD [m] 0505 9 (4)
ADD [m++] 0D05 12 (5)

ADC m0306 Add to Accumulator with Carry 7 (2)
ADC (m) 0106 8 (3)
ADC [m] 0506 9 (4)
ADC [m++] 0D06 12 (5)

SUB m0307 Subtract from Accumulator 7 (2)
SUB (m) 0107 8 (3)
SUB [m] 0507 9 (4)
SUB [m++] 0D07 12 (5)

SBC m0308 Subtract from Accumulator with Borrow 7 (2)
SBC (m) 0108 8 (3)
SBC [m] 0508 9 (4)
SBC [m++] 0D08 12 (5)

AND m 0309 Logical AND with Accumulator 7 (2)
AND (m) 0109 8 (3)
AND [m] 0509 9 (4)
AND [m++] 0D09 12 (5)

ORA m030A Logical OR with Accumulator 7 (2)
ORA (m) 010A 8 (3)
ORA [m] 050A 9 (4)
ORA [m++] 0D0A 12 (5)

XOR m030B Logical Exclusive-OR with Accumulator 7 (2)
XOR (m) 010B 8 (3)
XOR [m] 050B 9 (4)
XOR [m++] 0D0B 12 (5)

ISZ (m)010C Increment and Skip if Zero 9 (4)
ISZ [m]050C 10 (5)

DSZ (m) 010D Decrement and Skip if Zero 9 (4)
DSZ [m] 050D 10 (5)

Any Skipped instruction after ISZ or DSZ: 4 (1)

JMP m030E Unconditional Jump 7 (2)
JMP (m) 010E 8 (3)

JOZ m 030F Jump on Zero Accumulator 9 (2)
JOZ (m)010F 10 (3)

JNZ m 0310 Jump on Non-Zero Accumulator 9 (2)
### JNZ (m) 0110 10 (3)
### JPL m 0311 Jump on Plus Accumulator 9 (2)
### JPL (m) 0111 10 (3)
### JMI m 0312 Jump on Minus Accumulator 9 (2)
### JMI (m) 0112 10 (3)
### JOC m 0313 Jump on Carry (CF = 1) 9 (2)
### JOC (m) 0113 10 (3)
### JNC m 0314 Jump on No Carry (CF = 0) 9 (2)
### JNC (m) 0114 10 (3)
### JOV m 0315 Jump on Twos Complement Overflow (OVF = 1) 9 (2)
### JOV (m) 0115 10 (3)
### JNV m 0316 Jump on No Overflow (OVF = 0) 9 (2)
### JNV (m) 0116 10 (3)
### CSR m 0317 Call Subroutine 9 (3)

## D16/M Input/Output Instructions

<table>
<thead>
<tr>
<th>Mnemonic (Extended)</th>
<th>Opcode</th>
<th>Function</th>
<th>Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>INP (m)</td>
<td>0118</td>
<td>Input to Accumulator from I/O port at address m</td>
<td>7 (3)</td>
</tr>
<tr>
<td>OUT (m)</td>
<td>0119</td>
<td>Output from Accumulator to I/O port at address m</td>
<td>7 (3)</td>
</tr>
</tbody>
</table>