Data Processing Instructions [op = 00]

Table B.1 Data-processing instructions

<table>
<thead>
<tr>
<th>cmd</th>
<th>Name</th>
<th>Description</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>AND Rd, Rn, Src2</td>
<td>Bitwise AND</td>
<td>Rd + Rn &amp; Src2</td>
</tr>
<tr>
<td>0001</td>
<td>EOR Rd, Rn, Src2</td>
<td>Bitwise XOR</td>
<td>Rd + Rn ^ Src2</td>
</tr>
<tr>
<td>0010</td>
<td>SUB Rd, Rn, Src2</td>
<td>Subtract</td>
<td>Rd + Rn - Src2</td>
</tr>
<tr>
<td>0011</td>
<td>RSB Rd, Rn, Src2</td>
<td>Reverse Subtract</td>
<td>Rd + Src2 - Rn</td>
</tr>
<tr>
<td>0100</td>
<td>ADD Rd, Rn, Src2</td>
<td>Add</td>
<td>Rd + Rn+Src2</td>
</tr>
<tr>
<td>0101</td>
<td>ADC Rd, Rn, Src2</td>
<td>Add with Carry</td>
<td>Rd + Rn+Src2+C</td>
</tr>
<tr>
<td>0110</td>
<td>SBC Rd, Rn, Src2</td>
<td>Subtract with Carry</td>
<td>Rd + Rn - Src2 - C</td>
</tr>
<tr>
<td>0111</td>
<td>RSC Rd, Rn, Src2</td>
<td>Reverse Sub w/ Carry</td>
<td>Rd + Src2 - Rn + C</td>
</tr>
<tr>
<td>1000 (S = 1)</td>
<td>TST X, Rn, Src2</td>
<td>Test</td>
<td>Set flags based on Rn &amp; Src2</td>
</tr>
<tr>
<td>1001 (S = 1)</td>
<td>TEQ X, Rn, Src2</td>
<td>Test Equivalence</td>
<td>Set flags based on Rn ^ Src2</td>
</tr>
<tr>
<td>1010 (S = 1)</td>
<td>CMP Rn, Src2</td>
<td>Compare</td>
<td>Set flags based on Rn - Src2</td>
</tr>
<tr>
<td>1011 (S = 1)</td>
<td>CMN Rn, Src2</td>
<td>Compare Negative</td>
<td>Set flags based on Rn+Src2</td>
</tr>
<tr>
<td>1100</td>
<td>DRR Rd, Rn, Src2</td>
<td>Bitwise OR</td>
<td>Rd + Rn</td>
</tr>
<tr>
<td>1101</td>
<td>Sh:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MOV Rd, Src2</td>
<td>Move</td>
<td>Rd + Src2</td>
</tr>
<tr>
<td></td>
<td>LSL Rd, Rm, Rs/shamt5</td>
<td>Logical Shift Left</td>
<td>Rd + Rm&lt;&lt; Src2</td>
</tr>
<tr>
<td></td>
<td>LSR Rd, Rm, Rs/shamt5</td>
<td>Logical Shift Right</td>
<td>Rd + Rm&gt;&gt; Src2</td>
</tr>
<tr>
<td></td>
<td>ASR Rd, Rm, Rs/shamt5</td>
<td>Arithmetic Shift Right</td>
<td>Rd + Rm&gt;&gt;&gt; Src2</td>
</tr>
<tr>
<td></td>
<td>RRX Rd, Rm, Rs/shamt5</td>
<td>Rotate Right Extend</td>
<td>(Rd, C) + (C, Rd)</td>
</tr>
<tr>
<td></td>
<td>ROR Rd, Rm, Rs/shamt5</td>
<td>Rotate Right</td>
<td>Rd + R for Src2</td>
</tr>
<tr>
<td></td>
<td>BIC Rd, Rn, Src2</td>
<td>Bitwise Clear</td>
<td>Rd + Rn &amp; ~Src2</td>
</tr>
<tr>
<td></td>
<td>MVN Rd, X, Src2</td>
<td>Bitwise NOT</td>
<td>Rd + ~Src2</td>
</tr>
</tbody>
</table>

NOP (no operation) is typically encoded as 0xE1A000, which is equivalent to MOV R0, R0.
Memory Instructions  \([\text{op} = 01]\)  
- ldr, str, ldrb, strb

<table>
<thead>
<tr>
<th>cond</th>
<th>Mnemonic</th>
<th>Name</th>
<th>CondEx</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>EQ</td>
<td>Equal</td>
<td>Z</td>
</tr>
<tr>
<td>0001</td>
<td>NE</td>
<td>Not equal</td>
<td>Z</td>
</tr>
<tr>
<td>0010</td>
<td>CS/HS</td>
<td>Carry set / unsigned higher or same</td>
<td>C</td>
</tr>
<tr>
<td>0011</td>
<td>CC/LO</td>
<td>Carry clear / unsigned lower</td>
<td>C</td>
</tr>
<tr>
<td>0100</td>
<td>MI</td>
<td>Minus / negative</td>
<td>N</td>
</tr>
<tr>
<td>0101</td>
<td>PL</td>
<td>Plus / positive or zero</td>
<td>(\overline{N})</td>
</tr>
<tr>
<td>0110</td>
<td>VS</td>
<td>Overflow / overflow set</td>
<td>V</td>
</tr>
<tr>
<td>0111</td>
<td>VC</td>
<td>No overflow / overflow clear</td>
<td>V</td>
</tr>
<tr>
<td>1000</td>
<td>HI</td>
<td>Unsigned higher</td>
<td>(\overline{ZC})</td>
</tr>
<tr>
<td>1001</td>
<td>LS</td>
<td>Unsigned lower or same</td>
<td>(\overline{Z OR \overline{C}})</td>
</tr>
<tr>
<td>1010</td>
<td>GE</td>
<td>Signed greater than or equal</td>
<td>(N@V)</td>
</tr>
<tr>
<td>1011</td>
<td>LT</td>
<td>Signed less than</td>
<td>(N@V)</td>
</tr>
<tr>
<td>1100</td>
<td>GT</td>
<td>Signed greater than</td>
<td>(\overline{Z(N@V)})</td>
</tr>
<tr>
<td>1101</td>
<td>LE</td>
<td>Signed less than or equal</td>
<td>(\overline{Z OR (N@V)})</td>
</tr>
<tr>
<td>1110</td>
<td>AL (or none)</td>
<td>Always / unconditional</td>
<td>Ignored</td>
</tr>
</tbody>
</table>

Branch Instructions  \([\text{op} = 10]\)

<table>
<thead>
<tr>
<th>cond</th>
<th>\text{op}</th>
<th>1L</th>
<th>imm24</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td></td>
<td>(\text{signed two's complement})</td>
</tr>
</tbody>
</table>

(signed two's complement \# of instructions away from PC [current instruction + 2])

When reading from PC (r15), PC = Current instruction executing + 8 bytes (+2 instructions away)