ARMv8 Condition Flags

Condition Flags

Bits in the \texttt{pstate} register

\texttt{CMP Xs|SP, Xm}

CPU performs the subtraction $Xs|SP - Xm$

More precisely, CPU performs the addition $Xs|SP + \text{twoscomp}(Xm)$ and sets the condition flags depending upon the sum:

\begin{center}
\begin{tabular}{ | l | l |}
\hline
\textbf{Condition Code} & \textbf{Description} \\
\hline
Z (zero flag) & CPU sets Z to 1 iff all bits of the sum are 0. \\
N (negative flag) & CPU sets N to 1 iff the most significant bit of the sum is 1. \\
C (carry flag) & CPU sets C to 1 iff the addition caused a carry. \\
V (overflow flag) & CPU sets V to 1 iff both addends are $\geq 0$ and the sum is $< 0$, or both addends are $< 0$ and the sum is $\geq 0$. \\
\hline
\end{tabular}
\end{center}

Conditional Branch Instructions
(Used After Comparing Unsigned Numbers)

\begin{center}
\begin{tabular}{ | l | l |}
\hline
\textbf{Instruction} & \textbf{Branch if and only if:} \\
\hline
\texttt{beq (branch iff equal)} & Z==1 \\
\texttt{bne (branch iff not equal)} & Z==0 \\
\texttt{blo (branch iff lower)} & C==0 \\
\texttt{bhs (branch iff higher or same)} & C==1 \\
\texttt{bls (branch iff lower or same)} & C==0 || Z==1 \\
\texttt{bhi (branch iff higher)} & C==1 && Z==0 \\
\hline
\end{tabular}
\end{center}

Why does \texttt{blo} branch iff $C==0$? Examples (assuming a 4-bit computer):

1. \texttt{5 - 3 = 0101} - \texttt{0011 = 0101 + 1101 = 0010}, $C==1$ -> don't branch
2. \texttt{3 - 5 = 0011} - \texttt{0101 = 0011 + 1011 = 1110}, $C==0$ -> branch

So branch if and only if $C==0$. 

Page 1 of 2
Conditional Branch Instructions
(Used After Comparing Signed Numbers)

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Branch if and only if:</th>
</tr>
</thead>
<tbody>
<tr>
<td>beq (branch iff equal)</td>
<td>Z==1</td>
</tr>
<tr>
<td>bne (branch iff not equal)</td>
<td>Z==0</td>
</tr>
<tr>
<td>blt (branch iff less than)</td>
<td>N!=V</td>
</tr>
<tr>
<td>bge (branch iff greater than or equal)</td>
<td>N==V</td>
</tr>
<tr>
<td>ble (branch iff less than or equal)</td>
<td>N!=V</td>
</tr>
<tr>
<td>bgt (branch iff greater than)</td>
<td>N==V &amp;&amp; Z==0</td>
</tr>
</tbody>
</table>

Why does $\text{blt}$ branch iff if $N!=V$? Examples (assuming a 4 bit computer):

(1) $5 - 3 = 0101_B - 0011_B = 010_2 + 1101_2 - 0010_2$, $N==0$, $V==0 \rightarrow N==V \rightarrow$ don't branch

(2) $3 - 5 = 0011_B - 0101_B = 0011_2 + 1011_2 - 1110_2$, $N==1$, $V==0 \rightarrow N!=V \rightarrow$ branch

(3) $-5 - -3 = 1011_B - 1101_B = 1110_2 - 0011_2$, $N==1$, $V==0 \rightarrow N!=V \rightarrow$ branch

(4) $-3 - -5 = 1101_B - 1011_B = 1101_2 + 0101_2 - 0010_2$, $N==0$, $V==0 \rightarrow N==V \rightarrow$ don't branch

(5) $3 - -2 = 0011_B - 1110_B = 0011_2 + 1101_2 - 0101_2$, $N==0$, $V==0 \rightarrow N==V \rightarrow$ don't branch

(6) $3 - -6 = 0011_B - 1010_B = 0011_2 + 0110_2 - 1001_2$, $N==1$, $V==1 \rightarrow N==V \rightarrow$ don't branch

(7) $-3 - 2 = 1101_B - 0010_B = 1110_2 - 0101_2$, $N==0$, $V==0 \rightarrow N!=V \rightarrow$ branch

(8) $-3 - 6 = 1101_B - 0110_B = 1101_2 + 1010_2 - 0111_2$, $N==0$, $V==1 \rightarrow N!=V \rightarrow$ branch

So branch if and only if $N!=V$.  

Copyright © 2019 by Robert M. Dondero, Jr.