Computer Architecture 1 - Übungsblatt 8

Dirk Heine, 2030941
Jan Hendrik Dithmar, 2031259
Steffen Heil, 2019358

Aufgabe 2

The signal $hit[j]$ is supposed to indicate that the register accessed by the instruction in stage 1 is modified by the instruction in stage j. Except for the first four clock cycles all pipeline stages are full. During the initial cycles, an empty stage is prevented from signaling a hit by its full flag. The signal

$$top.j = hit[j] \wedge \bigwedge_{x=2}^{j-1} \neg hit[x]$$

indicates, that there occurs no hit in stages above j.
If we assume that $hit[j]$ is defined like this

$$hit[j] = GPRw.j \wedge (ad \neq 0) \wedge (ad = Cad.j)$$

instead of

$$hit[j] = (full.j \wedge GPRw.j) \wedge (ad \neq 0) \wedge (ad = Cad.j)$$

the signal $top.j$ is not generated correctly.
The data hazard signal $dhaz$ is defined like this

$$\begin{aligned} dhaz &= dhazA \vee dhazB \\ dhazA &= topA.2 \wedge \neg v[2].2 \vee topA.3 \wedge \neg v[3].3 \\ dhazB &= topB.2 \wedge \neg v[2].2 \vee topB.3 \wedge \neg v[3].3 \end{aligned}$$

If the signal $top.j$ is not generated correctly, the $dhaz$ signal won't be correct, too.
We track true instructions and dummy instructions in stage j by a single bit $full.j$, where $full.j = 1$ signals a true instruction and $full.j = 0$ signals a dummy instruction.

- **We assume that we have a dummy instruction and we use the signal** $full.j$.
  The signal $hit[j]$ will be equal to 0 because of $full.j = 0$. If we look at the signal $top.j$ we see that it is also equal to 0 because of $hit[j] = 0$.
  Now let's have a look at the signals $dhazA$ and $dhazB$. Because of $top.j = 0$, $dhazA = 0$ and $dhazB = 0$. For this reason, the signal $dhaz$ also equals 0.
  A dummy instruction cannot activate the signal $dhaz$.

- **Now we assume that we have a dummy instruction and we do not use the signal** $full.j$.
  The signal $hit[j]$ could now be different to the $hit[j]$ signal from our assumption above. Because of the fact, that the $hit[j]$ signal is used to compute

$top.j$ this signal might be different, too. If we have a 'wrong' signal $top.j$, the signal $dhaz$ could be activated by $dhazA$ or $dhazB$ or both. Now we see that the signal $dhaz$ could be activated by a dummy instruction if we do not use $full.j$ to generate $hit[j]$.

Aufgabe 3

$$
\begin{aligned}
movi2s(c) &= rtype(c) \wedge opc(c) = 0^6 \wedge fu(c) = 10001 \\
movs2i(c) &= rtype(c) \wedge opc(c) = 0^6 \wedge fu(c) = 10000
\end{aligned}
$$

$$
\begin{aligned}
movi2s(c) &\Leftrightarrow \begin{cases} c'.SPR(SA(c)) = c.GPR(RS1(c)) & : MODE = 0_{32} \\ c'.SPR(x) = c.SPR(x) & : MODE = 1_{32} \end{cases} \\
movs2i(c) &\Leftrightarrow \begin{cases} c'.GPR(RD(c)) = c.SPR(SA(c)) & : MODE = 0_{32} \\ c'.GPR(x) = c.GPR(x) & : MODE = 1_{32} \end{cases}
\end{aligned}
$$

Aufgabe 4

- Instructions in the pipeline do not overtake each other.

$$
\left.\begin{array}{l} I_i(k,t) \\ I_j(k-1,t) \end{array}\right\} \xrightarrow{\delta} \begin{cases} I_i(k+1,t+1) \\ I_j(k,t+1) \end{cases} \quad \forall i,j\ \forall k\ \forall t
$$

- We have never two instructions in the same stage at the same time.

$$
I_i(k,t) \Rightarrow \neg I_j(k,t) \quad \forall i,j\ \forall k\ \forall t
$$