Saarland University
Department 6.2 – Computer Science
Dr. M. A. Hillebrand
Prof. Dr. W. J. Paul

Computer Architecture I – WS 06/07

Exercise Sheet 8 **(due: 08.01.06)**

Exercise 1: (fast forwarding circuit) **(3+7+5+5=20 points)**

In class, a forwarding circuit capable of forwarding data from 3 stages has been presented. The construction obviously generalizes to s-stage forwarding, with $s > 3$, where the data selection is then performed by s cascaded multiplexers. Thus, the delay of this realization of an s-stage forwarding engine is in $(O)(s)$.

Moreover, signals $top.j$ are used to indicate that stage j provides the current data of the requested operand. These signals $top.j$ are used in order to govern the multiplexers.

1. Show how the construction of the multiplexer circuit can be modified such that the data selection can be done in $\mathcal{O}(log(s))$. (Assume that the select signals can be computed fast enough (cf. part 2))
2. Construct a circuit TOP which generates the signals $top.j$ in time $\mathcal{O}(log(s))$. (Hint: You might consider using a parallel prefix circuit)
3. Construct a s-stage forwarding engine based on your results from 1) and 2). Show that this realization has delay of $\mathcal{O}(log(s))$.
4. How can the delay of the forwarding engine be improved even further?

Exercise 2: (forwarding and hardware interlock: deadlock) **(10 points)**

Assume, we have an interlock engine that stalls the stages IF and ID in case of a data hazard, i. e. in case the forwarding engine cannot deliver the right data.

The forwarding circuit then signals a hit in a stage $j \in \{2, 3, 4\}$ by

$$hit[j] = full.j \wedge gprw.j \wedge (Cadr.j = adr) \wedge (adr \neq 0^5)$$

These hit signals are used to generate the *top* signals. Assuming for simplicity that we only consider one operand instead of two, these are then used to compute the data hazard signal *dhaz* as

$$dhaz = top.2 \wedge \neg v[2].2 \vee top.3 \wedge \neg v[3].3$$

(Recall that the $v[j]$ signals where the valid signals indicating that the data is already available in stage j.)

In this exercise, you will show that the check whether stage j is full (i. e. $full.j = 1$) is essential for the correctness of the interlock mechanism. Thus, show that when simplifying the hit signals to

$$hit[j] = gprw.j \wedge (Cadr.j = adr) \wedge (adr \neq 0^5)$$

dummy instructions could also activate the hazard flag, and that the interlock engine could run into a deadlock.