University of the Saarland
Department 6.2 - Informatik
Prof. Dr. W.J. Paul

Computer Architecture I - WS 02/03

(due: Monday 20.01.2003)

Excercise 1: (Fast Forwarding Circuit) **(7 + 8 + 5 points)**
In the lecture we build a forwarding circuit capable of forwarding data from 3 stages. The construction obviously generalizes to s-stage forwarding, with $s > 3$. The actual data selection is then performed by s cascaded multiplexers. Thus, the delay of this realization of an s-stage forwarding engine is $O(s)$.

However, these s multiplexers can also be arranged as a balanced binary tree of depth $\lceil \log(s) \rceil$. Signal *top.j* indicates that stage j provides the current data of the requested operand. These signals *top.j* can be used in order to govern the multiplexer tree.

- Construct a circuit TOP which generates the signals *top.j* using a parallel prefix circuit.
- Construct a s-stage forwarding engine based on the multiplexer tree and circuit TOP. Show that this realization has a delay of $O(\log(s))$.
- How can the delay of the forwarding engine be improved even further?

Excercise 2: (Deadlock) **(10 points)**
In case of a data hazard, the interlock engine stalls the stages `IF` and `ID`. The forwarding circuit `Forw` signals a hit of stage $j \in \{2, 3, 4\}$ by

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

These hit signals are used in order to generate the data hazard signal *dhaz*. The check whether stage j is full (i.e., $full.j = 1$) is essential for the correctness of the interlock mechanism.

Show that, when simplifying the hit signals to

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

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