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

Computer Architecture I - WS 03/04

(due: 28.01.2004)

Exercise 1: (Fast Forwarding) **(7+8 point)**

In the lecture we considered a forwarding circuit from 3 stages (MP00 Fig. 4.18). The construction obviously can be generalized to s-stage forwarding. The actual data selection is then performed by s cascaded multiplexors. Thus, the delay of this realization of an s-stage forwarding engine is $O(s)$.

However, these s multiplexors 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 multiplexor 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 multiplexor tree and circuit TOP. Show that this realization has a delay of $O(\log(s))$.

Exercise 2: (Shift Unit) **(5 points)**

Construct Shift Unit (only a circuit, not a layout) capable to perform all the shift operations for DLX. It should take data to be shifted $a[31:0]$, shift amount $b[4:0]$ and the code of the operation to be performed $sh[1:0]$ (logic left/right shift, arithmetic shift). As an idea you can use MP00 Fig.3.13, but you need give a more detailed construction. This circuit won't be used for $load/store$ instructions, but only for $shift$-instructions. Blocks, except for these which we already have (CLS, incrementor) should be constructed on the gate level. In the case that top level abstraction will be used, we will give 0 points.

Exercise 3: (ue_i computation for sequential DLX) **(5 points)**

Construct a circuit (not a layout) for computation of ue_i signals ($0 \leq i \leq 4$) for the sequential DLX, i.e. in each time cycle you need to update only one stage in consecutive order.