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

Computer Architecture I - WS 02/03

(due: Monday 27.01.2003)

Excercise 1: (Forwarding from stage 3) **(10 points + 5 bonus points)**

Forwarding from stage 3 in case of a load instruction is not possible with the forwarding engine from the lecture. The reason for this is that the hit_i signals are on if we have an instruction in the corresponding stage($full_i = 1$) and the instruction writes the correct register. If we are in stage 3 and have a load instruction this construction doesn't work anymore because in order to have the correkt result of a load instruction we need to do the shift4load. So we have the correct result in stage 4 and not in stage 3.

- Change the construction of the DLX in a way that your machine doesn't need the Hypothesis (if I_i reads $GPR(r)$ then I_{i-1} and I_{i-2} are no loads to $GPR(r)$) for the correctness proof. Hint: for this you have to change the definition of the hazard signals.
- Where do you have to change the correctness proof for your construction?

Excercise 2: (Formalism) **(2 + 2 points)**

Use the scheduling function $I_\pi(k, T)$ to formalize the following claims:

- Instructions in the pipeline do not overtake each other.
- We have never two instructions in the same stage at the same time.

Excercise 3: (Fast stall engine) **(6 points)**

The stall engine presented in the lecture has a delay of $O(n)$ where n is the number of stages. The reason for this is the following definition of the stall signal for stage k: $stall_k = full_k \wedge (hazard_k \vee stall_{k+1})$. So $stall_0$ depends on all other stall signals and has for the obvious construction delay $O(n)$.

Construct a stall engine which computes the signals $stall_i$ with delay of $O(\log n)$.