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

Computer Architecture I - WS 02/03

(due: 16.12.2002)

Excercise 1: (Cyclic Shifter) **(3 points)**

In the lecture we have build a cyclic left shifter of variable shift distance $b[k-1:0]$ by using k many cyclic left shifters of constant bit width. These constant shifters have a select signal s which selects if they shift or not. The select signal of the i-th[1] constant shifter (which shifts by 2^i positions) is connected to bit $b[i]$. The output of the i-th constant shifter is called r^i.

Prove the following lemma from the lecture: $\forall i : r^i = \mathrm{cls}(a, \langle b[i:0] \rangle)$.

Excercise 2: (Half Decoder) **(3 + 3 points)**

In the lecture we defined the function of a half decoder.

1. Give a recursive construction for a n bit half decoder.
2. Prove the correctness of your construction

Excercise 3: (Decoder) **(3 + 3 points)**

A n-decoder is a circuit with inputs $x[n-1:0]$ and outputs $Y[2^n-1:0]$ such that for all i

$$Y_i = 1 \quad \leftrightarrow \quad \langle x \rangle = i.$$

1. Give a recursive construction for a n-decoder. Your construction should have delay logarithmic in n.
2. Prove the correctness of your construction.

Excercise 4: (Incrementer Elimination) **(4 + 3 points)**

In the circuit *dist* which computes the correct shift distance for the shifter environment we have an five bit incrementer. This incrementer is used when doing cyclic right shifts with the cyclic left shifter.

The incrementer can be removed from the shift distance computation. Without the incrementer the circuit dist computes sometimes a wrong shift distance. This error can be corrected by an additional multiplexer in the datapaths of the shifter environment.

1. Which changes in the shifter environment are neccessary to get correct shifts without the five bit incrementer?
2. Proof that your modified shifter environment does the same as the old one.

---

[1] $i \in \{0, \ldots, k-1\}$

Excercise 5: (Aligned Memory Accesses) **(4 points)**

Prove that *legal* memory accesses in the DLX processor are aligned. For this you have to show that:

$$M(r(x,d)) \quad = \quad bytes_{br(x,d)} word(M,x)$$

In this equation x denotes the effective address of the memory access and M, r, d, *bytes*, br and *word* are defined as in the lecture.

Excercise 6: (Arithmetic Unit) **(4 points + 6 bonus points)**

In the lecture we have introduced a conditional sum adder and a carry look-ahead adder, and extended them to an arithmetic unit AU (including *overflow* and *negative* signals). Let $D_{AU}(n)$ denote the maximal delay of the n-bit AU, and let $D_{AU}(s[1:0];n)$ denote the delay of the two least significant sum bits.

1. Show that for the conditional sum AU and for any $n \geq 2$ the delay of the two least significant sum bits can be estimated as

$$D_{AU}(s[1:0];n) \quad \leq \quad D_{AU}(2).$$

2. Prove the same for the carry look-ahead AU.