Saarland University
Department 6.2 – Computer Science

Dr. M. A. Hillebrand
Prof. Dr. W. J. Paul

Computer Architecture I – WS 06/07

(due: 30.10.06)

Organizational Notes

1. Exercises:
   (a) Please register for an exercise group until **Friday, 27.10.06,** on the homepage.

   (b) Exercise sheets will be handed out on Mondays before the lecture. Your solutions will be collected one week after that (before the lecture).

   (c) You are allowed to solve the exercises **in groups** of up to **3 students**. Groups should not change over the semester. **Everybody** who has his name on a solution **must be able to present it** in tutorial.

2. Exam:
   (a) There will be **two regular exams**. The first on **Monday, 11.12.06,** within the lecture and a second one on **Monday, 19.02.06**. Additionally, there will be a **backup exam or oral test**, which you have to take if you do not pass both regular exams.

   (b) For the admission to the exam, you will need **at least 50%** of the combined points of all exercises handed out so far. Moreover, you must successfully present at least **one solution in a tutorial in each part of the lecture**, i.e., once before the first exam and another time before the second exam.

3. Grades for the lecture will be based on the grades for the exams; you have to pass two exams to pass the lecture.

Exercise 1: (decomposition lemma) (4 points)

Let $a \in \mathbb{B}^n$ and $\langle a \rangle$ its binary representation. Prove that for all $k \in \{0, \ldots, n-1\}$ the following equation holds:

$$\langle a[n-1:0] \rangle = \langle a[n-1:k] \rangle \cdot 2^k + \langle a[k-1:0] \rangle$$

Exercise 2: (basic properties of two's complement numbers) (10 points)

In the class, two's complement numbers and their properties where presented. In this exercise you need to show that these properties hold:

1. $[0, a] = \langle a \rangle$
2. $[a] \equiv \langle a[n-2:0] \rangle \mod 2^{n-1}$

3. $[a] \equiv \langle a \rangle \mod 2^n$
4. $[a[n-1], a] = [a]$ (sign extension)
5. $-[a] = [\neg a] + 1$

Exercise 3: (n-bit carry chain adder) (9 points)

We have seen the addition algorithm and the specification of adders. A *carry chain adder* (CCA) is an adder, which is implemented by chaining together full adders.

1. Construct an n-bit carry chain adder (CCA) based on the full adder circuit from the lecture and the inductive definition for the binary addition.
2. Compute the delay and the cost of your construction as a closed formula in n.
3. Prove the correctness of your construction.

Exercise 4: (n-bit decoder) (9 points)

An n-bit decoder is a circuit which takes an input $a[n-1:0]$ and computes an output $y[2^n-1:0]$ with the following property:

$$unary(y) \land \langle a \rangle = \langle y \rangle_{\mathrm{u}}$$

1. Construct an n-bit decoder.
2. Compute the delay and the cost of your construction as a closed formula in n.
3. Prove the correctness of your construction.

Note regarding exercises 3 and 4. The delay of AND and OR gates is 2, as opposed to what was presented in the lecture and is listed in the book by Mueller and Paul. This is a known bug, which is also listed in the book's errata list.