Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
M. Sc. Christian Müller

Computer Architecture I - WS 07/08

Exercise Sheet 1

Organizational stuff:

- The registration for this lecture is opened until **November 12th**!
- Every **monday** you will get an exercise sheet (today is an exception).
- Every tutorial will start with a minitest (15 minutes).
- In every minitest you have to solve 3 simple exercises, based on the previous exercise sheet.
- You need to solve 50% of all minitest exercises in order to be admitted to the exam.

Excercise 1: (properties of boolean operators)
For boolean values $a, b, c \in \mathbb{B}$ prove the following equations:

1. $a \oplus b = b \oplus a$ (commutativity)
2. $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ (associativity)
3. $a \wedge (b \vee c) = a \wedge b \vee a \wedge c$ (distributivity)
4. $\neg(\bigwedge_{i \in \{0,\ldots,n\}} a_i) = \bigvee_{i \in \{0,\ldots,n\}} \neg a_i$ (de Morgan)

Excercise 2: (associativity of addition)
Let $N : \mathbb{N} \to \mathbb{N}$ be the successor function defined in the lecture. We define the set of natural numbers $\mathbb{N}$ (including 0) using the Peano axioms:

A1: $0 \in \mathbb{N}$

A2: $\forall n \in \mathbb{N} : N(n) \in \mathbb{N}$

A3: $\nexists n \in \mathbb{N} : N(n) = 0$

A4: $\forall n, m \in \mathbb{N} : n = m \iff N(n) = N(m)$

A5: $\forall X \subseteq \mathbb{N} : 0 \in X \wedge (\forall n \in X : N(n) \in X) \Rightarrow X = \mathbb{N}$

On the set of natural numbers we define the addition as follows:

$$\begin{aligned} x + 0 &= x && (1) \\ x + N(y) &= N(x + y) && (2) \end{aligned}$$

Prove the following equations for $x, y, z \in \mathbb{N}$. Justify each your step.

a) $(x + N(0)) + N(0) = x + (N(0) + N(0))$

b) $(x + y) + z = x + (y + z)$

c) $x + 0 = 0 + x$

d) $x + N(0) = N(0) + x$

e) $x + y = y + x$

Saarland University

Department 6.2 - Computer Science

Prof. Dr. W. J. Paul

M. Sc. Christian Müller

Computer Architecture I - WS 07/08

Exercise Sheet 1

Excercise 3: (delay minimization)

Let $D(x)$ be the delay of the logical gate x. Assume, $D(AND) = D(OR) = 2$ and $D(NAND) = D(NOR) = D(INV) = 1$. Let $n = 2^i, i \in \mathbb{N}^+$. Consider the function

$$\begin{aligned} or &: \mathbb{B}^n \to \mathbb{B} \\ or(a_{n-1}, \ldots, a_0) &= \bigvee_{i \in \{0,\ldots,n-1\}} a_i \end{aligned}$$

1. Implement this function using only OR, AND or INV gates.

2. Minimize the delay of your construction using additional gates NAND and NOR.