Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl.-Ing. Christoph Baumann

Exercise Sheet 1
Computer Architecture II

(Due: Oct 29th, 2013)

Organisational Information:

- Each week on Tuesday one exercise sheet will be released. The solutions should be handed in before or after the next Tuesday lecture if not stated otherwise. For the admission to the exam you will need at least 50% of the points of the exercises.
- The tutorial takes place on Thursdays, 12:00 to 13:30 in E1.3, SR 3.28 starting from Oct 24th.
- You are allowed to solve the exercise sheets in groups of two people. Everybody who has his name on the solution must be able to present it in the tutorials. Everybody must present the solution of at least two exercises.
- Please, register for the lecture at the lecture's webpage until Nov 5th, 2013!
  `http://www-wjp.cs.uni-saarland.de/lehre/vorlesung/rechnerarchitektur2/ws1314/`
  Also do not forget to register for the exam in the HISPOS system!
- The oral exam will take place in February. An exact date will be decided upon in class.

Exercise 1: (Decomposition Lemma) **(5 Points)**

Prove the following Lemma for an n-digit number representation $a \in \{0, \ldots, B-1\}^n$ with $B \in \mathbb{N}^+$, $m, n \in \mathbb{N}^+$, $n > 1$ and $m < n$!

$$\langle a[n-1:0] \rangle_B \quad = \quad \langle a[n-1:m] \rangle_B \cdot B^m + \langle a[m-1:0] \rangle_B$$

Note: With $\langle a \rangle_B$ we denote the value of the number representation a regarding the base B.

$$\langle a[n-1:0] \rangle_B \equiv \sum_{i=0}^{B-1} a_i \cdot B^i$$

Also we agree on the convention $\langle a[n-1:0] \rangle \equiv \langle a[n-1:0] \rangle_2$.

Exercise 2: (Two's Complement Numbers) **(2+1+2 Points)**

In class, two's complement numbers and their definition were presented. In this exercise you have to show that the following properties hold for $a \in \mathbb{B}^n$.

a) $[a] < 0 \iff a_{n-1} = 1$

b) $[0a] \equiv \langle a \rangle$

c) $[a] \equiv \langle a \rangle \mod 2^n$

Gate	NOT	NAND / NOR	AND / OR	XOR / XNOR	n-MUX
Cost	1	2	2	4	$3n$
Delay	1	1	2	2	2

Table 1: Cost and Delay Model

Exercise 3: (Half Adders and Full Adders) **(3+3+4 Points)**

A Half Adder (HA) is a circuit if inputs $a, b \in \mathbb{B}$ and outputs $c', s \in \mathbb{B}$ fulfilling the formula:

$$\langle c', s \rangle = a + b$$

A Full Adder (FA) is a similar circuit with additional input $c \in \mathbb{B}$ and the following specification.

$$\langle c', s \rangle = a + b + c$$

a) Implement a Half Adder using the gates from Table 1!

b) Construct a Full Adder from two Half Adders and an OR gate and compute its cost and delay according to Table 1!

c) Find a faster and cheaper implementation of a Full Adder!

Hint: The optimal solution has Cost 11 and Delay 4.

Exercise 4: (Conditional Sum Adder) **(5+5 Points)**

Prove the following formulas for the cost and delay of a Conditional Sum Adder by induction on input bitwidth $n = 2^k$.

$$C(n) = \left(C(FA) + \frac{9}{2}\right) n^{\log 3} - 3n - \frac{3}{2} \qquad D(n) = 2 \log n + D(FA)$$

Exercise 5: (Parallel Prefix and Carry Lookahead Adder) **(2+4+4 Points)**

We have introduced a circuit with linear cost and logarithmic delay for Parallel Prefix computations with some associative binary operation $\circ : M \times M \to M$.[1] The circuit has inputs $x \in M^n$ and outputs $y \in M^n$ such that:

$$y_0 = x_0 \qquad \forall i > 0.\ y_i = x_i \circ x_{i-1}$$

In order to construct a fast and cheap adder we then introduced the operation $\circ_{GP} : \mathbb{B}^2 \times \mathbb{B}^2 \to \mathbb{B}^2$ on pairs of *generate*- and *propagate*-bits $(g1, p1), (g2, p2)$.

$$(g2, p2) \circ_{GP} (g1, p1) = (g2 \vee g1 \wedge p2, p1 \wedge p2)$$

a) In the lecture, the base case of the parallel prefix construction was erroneously shown for $n = 1$. Give a correct construction for the base case $n = 2$!

b) Prove for this construction the correctness of the following cost and delay estimates for $n = 2^k$.

$$C(n) \leq 2 \cdot C(\circ) \cdot n \qquad D(n) = D(\circ) \cdot (2 \log n - 1)$$

c) Prove that $\circ_{GP}$ is in deed associative, i.e.:

$$((g3, p3) \circ_{GP} (g2, p2)) \circ_{GP} (g1, p1) = (g3, p3) \circ_{GP} ((g2, p2) \circ_{GP} (g1, p1))$$

[1] see S. Müller and W. Paul, Computer Architecture, Springer 2000, pages 26-30