Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 1 (due: 12.08.13) - 27 points

Organizational notes:

- Please, register for the lecture on our website.
- 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 during the tutorial.
- You need to solve 50% of all exercises in order to be admitted to the exam.
- There will be two exercise groups: Group 1 (Mo/We 14-15:30) and Group 2 (Mo/We 15:30-17:00). You can choose your exercise group when submitting solutions to this exercise sheet (Aug 12 before the lecture). We want the students to be divided roughly equally between two groups. Hence, those who submit their exercise sheets later than others might have no choice of the group left.
- Note, that there are a lot of bonus questions. These questions are not graded and don't count towards the admission. Yet, they might be helpful to further your understanding and to prepare for the exam.
- For feedback on the difficulty of the sheet, please write down the amount of time spent on the exercise sheet (in hours, excluding bonus exercises). This number is irrelevant for your admission but helps us adjust the amount of exercises on each sheet.

Name, matr. nr., time spent: _____

Name, matr. nr., time spent: _____

Name, matr. nr., time spent: _____

Preferred exercise group (1 or 2): _____

Exercise 1: (8)

For each of the following statements, point out which ones are definitions and which ones are theorems[1]. Prove the theorems using only definitions and statements further up in the list.

(a) (1 point) $x = x + 0$

(b) (1 point) $x = 0 + x$

(c) (1 point) $x + (y + 1) = (y + x) + 1$

(d) (1 point) $(x + 1) + y = (y + x) + 1$

(e) (1 point) $(x + y) + z = x + (y + z)$

(f) (1 point) $x + y = y + x$

(g) (1 point) $x \cdot 0 = 0$

(h) (1 point) $0 \cdot x = 0$

[1] According to the language of the old man.

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 1 (due: 12.08.13) - 27 points

Exercise 2: **(1)**

Compute by hand the length $|\cdot|$, binary interpretation $\langle\cdot\rangle$, and two's complement implementation $[\cdot]$ of the following bitstrings:

(a) 1000101

(b) 00111

Exercise 3: **(3)**

Prove that $a \equiv b \mod k$ is an equivalence relation.

Exercise 4: **(4)**

For $n \in \mathbb{N}$ and $a \in B^n$, prove all of the following statements with at most two lines each:

(a) (1 point) $[a] \in T_n$

(b) (1 point) $\langle a \rangle \equiv [a] \bmod 2^n$

(c) (1 point) $[a] = [a_{n-1}a]$

(d) (1 point) $[\bar{a}] = -[a] - 1$

Exercise 5: **(3)**

Prove a decomposition lemma for two's complement interpretation: for all $n, m \in \mathbb{N}$ and $a \in \mathbb{B}^n$,

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

Exercise 6: **(3)**

Prove that for $\circ \in \{+, -, *\}$, $n \in \mathbb{N}$, and $u, a, b \in \mathbb{B}^n$, we have:

$$u = a \circ_n b \implies [u] = ([a] \circ [b] \operatorname{tmod} 2^n)$$

Exercise 7: **(1)**

Prove by induction that for $n \in \mathbb{N}$, the depth of the n-bit conditional-sum-adder is $D(n) = 3 + 3 \log n$.

Exercise 8: **(4)**

For $n \in \mathbb{N}$ and an associative binary function $\circ \in M \times M \to M$ the n-bit parallel prefix circuit is a circuit which takes one input $x \in M^n$ and computes the output $y \in M^n$, where

$$y_0 = x_0$$
$$y_{i+1} = x_{i+1} \circ y_i.$$

(a) Compute the outputs $y_0, \ldots, y_{n-1}$ of the following instances of the parallel prefix circuit:

i. (1 point) For $n = 4$, $M = \mathbb{B}$, and $a \circ b = a \oplus b$, with $x = 1101$.

ii. (1 point) For $n = 4$, $M = \mathbb{B}^2$, and $a \circ b = c$, where $c_0 = a_0 \wedge b_0$ and $c_1 = b_1 \vee a_1 \wedge b_0$, with $x = 10011101$.

(b) Construct a circuit computing $\circ$ and construct the parallel prefix circuit for the following instances:

i. (1 point) For $n = 4$, $M = \mathbb{B}$, and $a \circ b = a \oplus b$.

ii. (1 point) For $n = 4$, $M = \mathbb{B}^2$, and $a \circ b = c$, where $c_0 = a_0 \wedge b_0$ and $c_1 = b_1 \vee a_1 \wedge b_0$.

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 1 (due: 12.08.13) - 27 points

Bonus Exercise 9:

Consider the formal induction principle. For every set A,

$$0 \in A \wedge (\forall n \in \mathbb{N}.n \in A \Rightarrow (n+1) \in A) \Rightarrow \mathbb{N} \subseteq A$$

(a) Copy the induction principle. Then mark by underlining:
   i. The property to be shown
   ii. The induction base
   iii. The induction hypothesis
   iv. The induction step

(b) Consider the proof of the School Addition Correctness Lemma given in the lecture[2]. Write down the proof while clearly marking:
   i. The property to be shown
   ii. The induction base
   iii. The induction hypothesis
   iv. The induction step

(c) Apply the proof steps of the School Addition Correctness Lemma manually to prove $\langle 110\rangle + \langle 101\rangle + 0 = \langle 1011\rangle$, i.e., perform the transformations according to the proof. When you reach the place where the induction hypothesis is used, first prove it recursively, then use it like in the proof. Do not calculate $\langle 110\rangle$ etc. Do not collect \$200.

Bonus Exercise 10:

Consider the set $\{1, 2, 3, \ldots, 8, 9, 10\}$ and the following equivalence classes: $\{1, 4, 6\}$, $\{2, 8\}$, $\{3\}$, $\{5, 7\}$, $\{9\}$, $\{10\}$.

(a) Give a system of representatives.

(b) Give the equivalence classes of 1, 2, 3, and 4.

Bonus Exercise 11:

Informally explain the difference between mod and tmod, then give an example where they differ.

Bonus Exercise 12:

Informally explain the difference between $a = (b \mod k)$ and $a \equiv b \bmod k$. Give an example (i.e., an a, b, and k) where they differ, e.g., $a \neq (b \mod k)$ but $a \equiv b \bmod k$.

Bonus Exercise 13:

Recall the binary subtraction algorithm that uses only addition.

(a) Refute that for all $n \in \mathbb{N}$ and $a, b \in \mathbb{B}^n$, $\langle a\rangle - \langle b\rangle = \langle a\rangle + \langle \bar{b}\rangle + 1$

(b) Refute that for all $n \in \mathbb{N}$ and $a, b \in \mathbb{B}^n$, $\langle a\rangle - \langle b\rangle = (\langle a\rangle + \langle \bar{b}\rangle + 1 \bmod 2^n)$

(c) Prove that for all $n \in \mathbb{N}$ and $a, b \in \mathbb{B}^n$, $\langle a\rangle - \langle b\rangle \equiv \langle a\rangle + \langle \bar{b}\rangle + 1 \bmod 2^n$

[2] If you don't have a copy of the proof, prove it yourself: $\langle a[n-1:0]\rangle + \langle b[n-1:0]\rangle + c_0 = \langle c_n s\rangle$, where c_n, s is the output of the school addition algorithm.

Saarland University
Department 6.2 - Computer Science
Prof. Dr. W. J. Paul
Dr. Mikhail Kovalev, Jonas Oberhauser

System Architecture (block course) - SS13

Exercise Sheet 1 (due: 12.08.13) - 27 points

Bonus Exercise 14:

Prove that for $n \in \mathbb{N}$, $\mathrm{bin}_n(\cdot)$ and $\mathrm{twoc}_n(\cdot)$ are the inverse functions of $\langle\cdot\rangle$ and $[\cdot]$ by proving:

(a) For $x \in B_n$, $\langle \mathrm{bin}_n(x)\rangle = x$

(b) For $x \in T_n$, $[\mathrm{twoc}_n(x)] = x$

(c) For $a \in \mathbb{B}^n$, $\mathrm{bin}_n(\langle a\rangle) = a$

(d) For $a \in \mathbb{B}^n$, $\mathrm{twoc}_n([a]) = a$

Bonus Exercise 15:

Prove the Cheap Lemma. That is, prove that for $a, b, k \in \mathbb{Z}$, $a \equiv b \bmod k$ implies both:

(a) $a \in B_k \implies a = (b \bmod k)$

(b) $a \in T_k \implies a = (b \operatorname{tmod} k)$

(c) What does this mean for $a \in B_k \cap T_k$?

Bonus Exercise 16:

For $n \in \mathbb{N}$, define circuits that compute n-bit bitwise and, or, negation, and xor.

Bonus Exercise 17:

For $n \in \mathbb{N}$, define an n-bit multiplexer, i.e., a circuit with three inputs $a, b \in \mathbb{B}^n$ and $s \in \mathbb{B}$ computing the following output $x \in \mathbb{B}^n$:

$$x = \begin{cases} a & s = 1 \\ b & \text{otherwise} \end{cases}$$

Bonus Exercise 18:

Recall the full-adder and consider this informal definition of the sum bit: "The sum bit s is 1 if and only if an odd number of input bits are 1." Give a similar description of the carry bit.

Bonus Exercise 19:

Recall, for $n \in \mathbb{N}$, the specification of an n-adder, i.e., a circuit with three inputs $a, b \in \mathbb{B}^n$ and $c_0 \in \mathbb{B}$ such that the outputs $c_n \in \mathbb{B}$ and $s \in \mathbb{B}^n$ have the following property:

$$\langle a\rangle + \langle b\rangle + c_0 = \langle c_n s\rangle$$

(a) Instantiate the carry-chain-adder for $n = 4$ by following the recursive definition. When you reach the place where the recursive construction is used, first construct it recursively, then use it like in the construction (i.e., as a building block). Give the depth and cost of this circuit.

(b) Instantiate the conditional-sum-adder for $n = 4$ by following the recursive definition. When you reach the place where the recursive construction is used, first construct it recursively, then use it like in the construction (i.e., as a building block). Give the depth and cost of this circuit, and compare it to the cost and depth of the carry-chain-adder with $n = 4$.

(c) Give a proof of correctness for the conditional-sum-adder, i.e., show that the outputs c_n and s have the property mentioned above.