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

System Architecture (block course) - SS13

Exercise Sheet 2 (due: 14.08.13) - 20 points

Organizational notes:

- 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: _____

Exercise 1: (4)

Consider, for $n \in \mathbb{N}$, $j \in [0 : n-1]$, $i \in [0 : j]$, $a, b \in \mathbb{B}^n$ and $c_0 \in \mathbb{B}$ the definition of $g_{i,j}$ and $p_{i,j}$ given below:

$$g_{i,j} := \langle 10^{j-i+1} \rangle \leq \begin{cases} \langle a[j:i] \rangle + \langle b[j:i] \rangle & i > 0 \\ \langle a[j:i] \rangle + \langle b[j:i] \rangle + c_0 & i = 0 \end{cases}$$

$$p_{i,j} := 2^{j-i+1} - 1 = \langle a[j:i] \rangle + \langle b[j:i] \rangle$$

(a) (1 point) Prove that when $i > 0$, $g_{i,j} = 1 \iff \forall c \in \mathbb{B}.\langle a[j:i] \rangle + \langle b[j:i] \rangle + c = \langle 1*^{j-i+1} \rangle$, i.e., that a carry bit is always *g*enerated, independent of the carry-in.

(b) (1 point) Prove that $p_{i,j} = 1 \iff \forall c \in \mathbb{B}.\langle a[j:i] \rangle + \langle b[j:i] \rangle + c = \langle c*^{j-i} \rangle$, i.e., that the carry bit is always *p*ropagated.

(c) (2 extra points) Prove that $g_{i,j} = g_{k+1,j} \vee (g_{i,k} \wedge p_{k+1,j})$

(d) (2 extra points) Prove that $g_{0,j} = c_{j+1}$

(e) (2 points) Instantiate the construction of the n-bit carry-look-ahead adder for $n = 4$

Exercise 2: (2)

For sets M, $x \in \mathbb{N}$, and associative function $\circ : M \times M \to M$, let the circuit $\circ$-T_x take inputs $a^1, \ldots, a^x \in M$ and compute output $a' = a^1 \circ \ldots \circ a^x$ with logarithmic depth. Construct $\vee$-T_4 recursively for $M = \mathbb{B}^2$, where $a \vee b = (a_1 \vee b_1, a_0 \vee b_0)$.

Exercise 3: (2)

Recall the $n - 3/2 - Adder$ from the lecture that adds three numbers to two numbers with constant depth.

(a) (1 point) Give a **recursive** construction of the $n - 3/2 - Adder$. Instantiate it for $n = 4$.

(b) (1 point) Give a compositional construction of the $3/2 - Adder$, i.e., by giving circuits computing s_i and t_i.

Exercise 4: (7)

Recall the arithmetic unit, which for $n \in \mathbb{N}$ takes parameters $a, b \in \mathbb{B}^n$, and $u, sub \in \mathbb{B}$ and returns outputs $s \in \mathbb{B}^n$, and $ovf, neg \in \mathbb{B}$

(a) (1 point) What is the purpose of the AU? How is the output of the AU controlled?

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

System Architecture (block course) - SS13

Exercise Sheet 2 (due: 14.08.13) - 20 points

(b) (2 points) Recall the exact result $S \in \mathbb{Z}$ and the following specification:

$$ovf = \begin{cases} S \notin T_n & u = 0 \\ 0 & u = 1 \end{cases}$$
$$neg = S < 0$$

Prove the correctness of the circuit computing ovf and neg, i.e., prove:

i.
$$neg = 1 \iff (\neg u \wedge (a_{n-1} \oplus d_{n-1} \oplus c_n)) \vee (u \wedge sub \wedge \neg c_n)$$

ii.
$$ovf = \neg u \wedge (c_n \oplus c_{n-1})$$

(c) (1 point) Construct a 2-bit AU using only basic gates, multiplexers, and n-adders.

(d) (1 point) What is the purpose of the ALU? How is the output of the ALU controlled?

(e) (2 points) Extend your construction of an AU to an ALU (you can use more advanced circuits like equality testers, etc.)

Exercise 5: **(2)**

Recall, for $n \in \mathbb{N}$, the n-bit decoder, which is a circuit with an input $a \in \mathbb{B}^n$ and has an output $a' \in \mathbb{B}^{2^n}$ such that $a'_i = 1 \iff i = \langle a \rangle$.

(a) (bonus) Explain informally what an n-bit decoder does and give an example of what it could be used for.

(b) (1 point) For $n = 3$, give a table with all a and corresponding a'. Your table should be sorted by $\langle a \rangle$.

(c) (1 point) Instantiate the n-bit decoder for $n = 3$

Exercise 6: **(3)**

Recall the definition of computation. For $K = \mathbb{B}^2$, $I = \mathbb{B}$, and $\delta(c, i) = c +_2 \bar{i}i$, consider $c^0 = 01$ and $i^0 = 1$, $i^1 = 1$, $i^2 = 0$, $i^3 = 1$.

(a) (1 point) Compute $c^1, \ldots, c^4$.

(b) (2 points) Design a circuit computing δ. Then add registers to store the configuration. Don't forget to initialize these registers for c^0 by using the *reset* signal.

(c) (bonus) Recall the example circuit from the lecture. Prove by induction on n that it computes $R^n = (n \mod 2)$

Bonus Exercise 7:

Construct a left shifter and prove it's correctness. I.e., for $n, k \in \mathbb{N}$, $a \in \mathbb{B}^n$ and $b \in \mathbb{B}^k$ construct a circuit computing $a' = \mathrm{sll}(a, \langle b \rangle)$ where:

$$\mathrm{sll}(a, i) = a[n - 1 - i : 0]0^i$$

Bonus Exercise 8:

Compare a $(1, 5)$-$\vee$-tree with a 5-$\vee$-tree.

Bonus Exercise 9:

Instantiate a 4-zero circuit, then extend it to a 4-eq circuit.