Saarland University
Department 6.2 – Computer Science
Dr. M. A. Hillebrand
Prof. Dr. W. J. Paul

Computer Architecture I – WS 06/07

Exercise Sheet 2 **(due: 06.11.06)**

Exercise 1: (parallel-prefix computation) **(3+5+3+2+2=15 points)**

Let $u2hu : \mathbb{B}^n \to \mathbb{B}^n$ be a function that assigns each input bit vector a with $unary(a)$ its half-unary representation. Hence,

$$u2hu(a) = b \text{ such that } unary(a) \wedge \langle a \rangle_u = \langle b \rangle_{hu}$$

1. Construct a simple circuit with cost and delay in $O(n)$ that computes the function $u2hu$. Prove the correctness of your construction.
2. Construct a parallel-prefix circuit that computes $u2hu$ and prove its correctness. Furthermore, compute the delay and the cost of your construction as a closed formula.
3. Draw the PP-circuit for $n = 8$.
4. In the real world, gates may only drive a limited small number of other gates. The *fan-out of a gate* is defined as the number of gates that are driven by the output of this gate. The *fan-out of a circuit* is the maximum fan-out of any gate in it.

   Characterize and compare the fan-outs of the circuits that you have designed in 1. and 2. above.
5. Is there an additional purpose of the function $u2hu$, i.e., what happens in case $unary(a)$ does not hold for the input?

Exercise 2: (parallel-prefix computation; carry-lookahead adder) **(5 points)**

In the construction of the carry-lookahead adder, we have computed the prefix over the following function $\circ : M \times M \to M$ for $M = \mathbb{B}^2$ in a parallel prefix circuit:

$$(g_1, p_1) \circ (g_2, p_2) = (g_2 \vee g_1 \wedge p_2, p_1 \wedge p_2)$$

Show that $\circ$ is associative, i.e., $(x \circ (y \circ z)) = ((x \circ y) \circ z)$ for all x, y, $z \in M$.

Exercise 3: (conditional sum incrementer) **(9 points)**

In class, an incrementer was defined as a circuit that computes

$$s = a +_n 0^{n-1} c_{\text{in}}$$

Let $m = \lceil n/2 \rceil$ for any $n > 1$.

The high order sum bits ($s[n-1:m]$) of an n-bit incrementer with inputs $a[n-1:0]$ and c_{in} can be expressed as

$$\begin{aligned}\langle s[n-1:m]\rangle &= \langle a[n-1:m]\rangle + c_{m-1} \\ &= \begin{cases}\langle a[n-1:m]\rangle & \text{for } c_{m-1}=0 \\ \langle a[n-1:m]\rangle + 1 & \text{for } c_{m-1}=1\end{cases}\end{aligned}$$

where c_{m-1} denotes the carry bit from position $m-1$ to m.

1. Use this to construct an n-bit conditional sum incrementer.
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: (shifter) **(11 points)**

1. Prove the correctness of the n-bit cyclic left shifter presented in class.
2. Prove the following shift properties:
   - $\langle lls(a,i)\rangle \equiv \langle a\rangle \cdot 2^i \mod 2^n$
   - $\langle lrs(a,i)\rangle \equiv \lfloor \langle a\rangle / 2^i \rfloor$
   - $[ars(a,i)] \equiv \lfloor [a]/2^i \rfloor$