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

Exercise Sheet 5
Computer Architecture II
(Due: Nov 26th, 2013)

Exercise 1: (Fast $\vee$-Tree) **(4+4+4 Points)**

For the Implementation of the sticky bit computation we need an $\vee$-tree which is a balanced binary tree of OR gates. The exact delay of the circuit can be reduced by using the inverted gates NOR ($\overline{\vee}$) and NAND ($\overline{\wedge}$). Their semantics is given by the following formulas.

$$x_1 \,\overline{\vee}\, x_2 \;=\; \overline{x_1 \vee x_2} \qquad\qquad x_1 \,\overline{\wedge}\, x_2 \;=\; \overline{x_1 \wedge x_2}$$

The delay of NAND and NOR gates is considered to be 1 in our delay model, compared to 2 for regular AND and OR gates. The cost for each of these gates is 2.

a) Construct an $\vee$-tree with inputs $a \in \mathbb{B}^n$ for $n = 2^k$, output $b \in \mathbb{B}$ and the specification:

$$b \;=\; \bigvee_{i=0}^{n-1} a[i]$$

The circuit should have the following delay: $$D(n) = \begin{cases} \log n & : \quad k \text{ even} \\ \log n + 1 & : \quad k \text{ odd} \end{cases}$$

b) Prove the correctness of your implementation!

c) Show that your construction has the correct delay!

Note: We are using the following notation for a disjunction of $n \in \mathbb{N}$ boolean terms x_i:

$$\bigvee_{i=0}^{n-1} x_i \;\equiv\; x_{n-1} \vee \ldots \vee x_0$$

Exercise 2: (Logical Right Shifter for Arbitrary Numbers) **(5+5+5+5 Points)**

In the unpacker we need a logical right shifter for numbers which are not a power of two. Let $n \in \mathbb{N}$ be such a number, i.e. $\forall k.\ n \neq 2^k$ and $m = \lceil \log n \rceil$. A circuit $n{-}LRSA$ has inputs $x \in \mathbb{B}^n$, $b \in \mathbb{B}^m$ such that $0 \leq \langle b[m-1:0] \rangle \leq n$ and outputs $y \in \mathbb{B}^n$ specified by the following formula:

$$y[n-1:0] \;=\; 0^{\langle b \rangle} x[n-1:\langle b \rangle]$$

a) Give a recursive construction of an $n{-}LRSA$!

b) Prove the correctness of your implementation!

c) Compute cost and delay of your circuit!

d) Show that your results are correct!

Exercise 3: (Table Size vs. Number of Iterations) **(8 Points)**

In the lecture we designed the multiply/divide unit from the book which performs a division based on the Newton-Raphson method. The iteration starts out with an initial approximation x_0 which is obtained from a $2^\gamma \times \gamma$ lookup table. The intermediate results are truncated after $\sigma = 57$ bits. The number i of iterations necessary to reach $p + 2$ bits of precision (i.e. $\delta_i < 2^{-(p+2)}$) is then bounded by

$$i = \begin{cases} 1 & : \quad (p = 24) \wedge (\gamma = 16) \\ 2 & : \quad (p = 24) \wedge (\gamma = 8) \vee (p = 53) \wedge (\gamma = 16) \\ 3 & : \quad (p = 24) \wedge (\gamma = 5) \vee (p = 53) \wedge (\gamma = 8) \end{cases}$$

Prove that the number of iterations suffices to achieve the desired precision!