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

Exercise Sheet 6
Computer Architecture II
(Due: Dec 2th, 2013)

Exercise 1: (Optimized FDselect Circuit) **(5 Points)**

Figure 1 shows the FDSELECT circuit of the Multiply/Divide Unit in our FPU. In the lecture we discovered that $E = \lfloor f_a \cdot x \rfloor_{p+1}$ can actually be represented with $p+2$ bits. This influences the length of E_b, too. Additionally D_a and D_b just store pipelined copies of the unpacked and normalized significands f_a and f_b. Consequently FDSELECT contains superfluous data lines and gates. Give a short summary which circuitry can be omitted and construct a corresponding optimized version of FDSELECT!

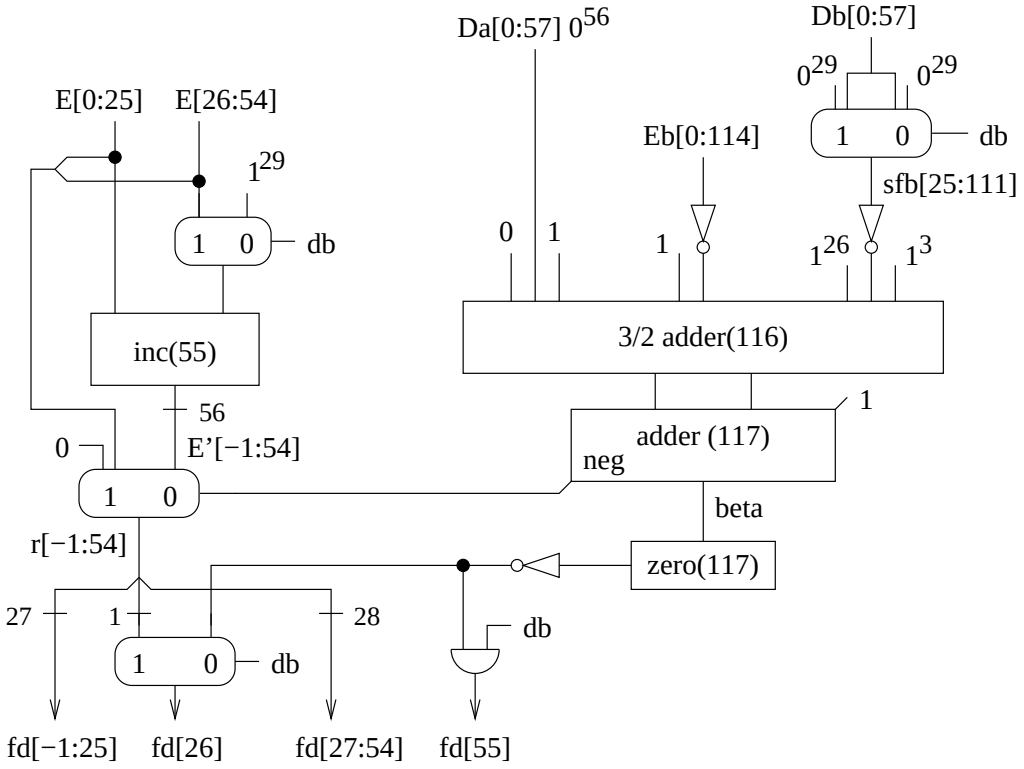

Figure 1: The FDSELECT circuit selecting the correct significand for the quotient in the Multiply/Divide unit (taken from *Computer Architecture*, S. M. Müller, W. J. Paul, Springer 2000)

Exercise 2: (Division Pipeline Optimization) **(5+5 Points)**

In the lecture it was noted that the multiplier in the Multiply/Divide unit actually only needs width 58 for the computation of E. All other multiplications in the division algorithm can be implemented with less bits. In the following we want to use this insight to optimize the delay of the division significand computation. Let $p = 53$ and $\gamma = 8$.

a) In the current algorithm we always use the same σ for all computations. Instead we could have individual σ_i for each iteration $i \geq 0$. Then we have $\sigma_2 = 57$. Find one pair of values for (σ_0, σ_1), such that $\sigma_0 + \sigma_1 \leq 50$ and $\delta_3 < 2^{-(p+2)}$! Prove your claim!

b) At the moment the significand of the quotient is computed by iterating a loop several times. Imagine that we would unroll this loop and compute every multiplication in a dedicated pipeline stage. Then we could optimize the width of the individual multiplication trees and adders. Using your results from a) calculate the minimal bit widths of these circuits for the computation of z_i, x_{i+1}, E and E_b with $i \in [0:2]$! Also list for each of the stages which signals should be connected to inputs a and b of the multiplier!

Exercise 3: (Compound Adder) **(5+5+5 Points)**

In the implementation of the floating-point rounder we need to produce two exponents e_n and e_{n+1}. An n-compound adder is a circuit with inputs $a[n-1:0]$, $b[n-1:0]$ and outputs $s[n:0]$, $t[n:0]$ satisfying:

$$\begin{aligned} \langle s \rangle &= \langle a \rangle + \langle b \rangle \\ \langle t \rangle &= \langle a \rangle + \langle b \rangle + 1 \end{aligned}$$

a) Provide a recursive implementation of an n-compound adder for $n = 2^k$!
Hint: The structure of the circuit is similar to that of a Conditional Sum Adder.

b) Show the correctness of your construction!

c) Deduce the exact cost of the adder and prove your result!

Exercise 4: (Exponent Lemma) **(5 Points)**

For the construction of the normalization shifter we had to compute $\hat{e}$ for the exact result x in case of $TINY(x)$. There we used the following lemma for $p \in \mathbb{N}$, $e, e' \in \mathbb{Z}$, $f, f' \in [1:2)$.

$$2^e \cdot f =_{p-e} 2^{e'} \cdot f' \implies e = e'$$

Give a proof for this lemma!

Exercise 5: (TINY Computation) **(5 Points)**

The FLAGS circuit in the floating-point rounder computes among others the signal $TINY$. To this end the number of leading zeroes lz of the result's significand $f_r > 0$ is substracted from the constant $bias = \langle 1^{n-1} \rangle$. However bit 13 of the difference is omitted in the circuit. Show that this is justified, i.e. prove the following statement:

$$bias - lz \in T_{n+2}$$

Note: The set $T_n = \{-2^{n-1}, \ldots, 2^{n-1} - 1\}$ contains the range of numbers which are representable as two's complement numbers of length n.