University of the Saarland
Department 6.2 - Informatik
Prof. Dr. W.J. Paul

Computer Architecture I - WS 02/03

(due: 25.11.2002)

Excercise 1: (Binary Representation of E_{2j}) **(2 points)**

In the lecture we proved that (if $j > 0$) the binary representation e_{2j} of E_{2j} could be computed by:

$$\begin{aligned} \langle e_{2j} &= \langle 1\overline{s_{2j}}, d_{2j} \oplus s_{2j} \rangle + s_{2j} \\ \langle e_0 &= \langle \overline{s_0} s_0 s_0, d_0 \oplus s_0 \rangle + s_0. \end{aligned}$$

Prove the claim for $j = 0$.

Excercise 2: (Encoder) **(4 + 4 + 2 + 2 points)**

A 2^n bit encoder is a circuit with input $x[2^n - 1 : 0]$ and output $y[n-1:0]$. If there is only one 1 in the input x the encoder computes the position of this 1:

$$x = 0^{n-i-1} 1 0^i \Longrightarrow y = \langle i \rangle$$

If there is no 1 or more than one 1 in the input then the output of the circuit is not defined.

1. Construct a 2^n bit encoder.
2. Prove that your circuit is a correct implementation of the above definition.
3. What are the cost and the delay of your construction. Give it as a recursion formula and in O notation. You don't have to give correctness proofs.
4. What is the output of your construction if the input has not exactly one 1.

Excercise 3: (Multiplication of Two's Complement Numbers) (3 + 3 points + 8 bonus points)

The multipliers which we build in the lecture multiply only positive numbers.

1. Build a multiplier which multiplies two's complement numbers. In order to do this use the rule that

$$a * b = \begin{cases} +|a| * |b| \text{ if } (a > 0 \wedge b > 0) \vee (a < 0 \wedge b < 0) \\ -|a| * |b| \text{ if } (a > 0 \wedge b < 0) \vee (a < 0 \wedge b > 0) \end{cases}$$

   and that one can compute $-a$ by inverting all bits of a and increment the result.

2. Prove that for $a[n-1:0]$ and $b[n-1:0]$ one can compute $[a] * [b]$ by:

$$[a] * [b] = < a > * < b > -2^n * a_{n-1} * b[n-2:0] - 2^n * b_{n-1} * a[n-2:0]$$

   .

3. (bonus) Can you (with the formula from part 2) build a two's complement multiplier without the additional delay of the incrementer?

   Tip: You can incorporate the incrementer which is needed for the subtraction in the above formula into the addition tree without really using an incrementer.

Excercise 4: (Karatsuba Offman Multiplier) (2 + 4 + 4 points)

For bit vectors $a[n-1:0]$ and $b[n-1:0]$ one can compute $a * b$ by the following[1]:

Let

$$\begin{aligned} r &= a[n-1:n/2] * b[n-1:n/2] \\ s &= (a[n-1:n/2] + a[n/2-1:0]) * (b[n-1:n/2] + b[n/2-1:0]) \\ t &= a[n/2-1:0] * b[n/2-1:0] \end{aligned}$$

Then $a * b = 2^n * r + (s - r - t) * 2^{n/2} + t$.

1. Proof the correctness of this formula.
2. Build a multiplier which uses the above formula.
3. Proof that the cost of your construction is in $O(n^{\log 3})$

[1] n is a power of two