Known Errata in 'Silvia M. Mueller and Wolfgang J. Paul. *Computer Architecture: Complexity and Correctness.* Springer, 2000'

Maintained by Mark Hillebrand (`mah at cs dot uni-sb dot de`)

Date: 2008/03/05 22:01:48

1. Reported by Jörg Fischer, Saarland University
   Page 8, Table 2.1: the delay of AND, OR is 2.

2. Reported by Jörg Fischer, Saarland University
   Page 22: the brackets $\langle\cdot\rangle$ are missing in the equation chain $z = \ldots$.

3. Reported by Ulan Degenbaev, Saarland University
   Page 26: In Figure 2.15, the signal $s0[n:m]$ should include the topmost 0, otherwise the bit widths are not correct. Also, the formulae for the cost and delay of the conditional sum incrementer should refer to $C_{\mathrm{inc}}(\cdot)$ and $D_{\mathrm{inc}}(\cdot)$ instead of $C_{\mathrm{CCI}}(\cdot)$ and $D_{\mathrm{CCI}}(\cdot)$.

4. Reported by Mark Hillebrand, Saarland University
   Page 29: in the equation for (G_i, P_i) the indices for the generate and propagate signals are swapped, it should read
   $$(G_i, P_i) = (g_i, p_i) \circ \cdots \circ (g_0, p_0) = (g_{0,i}, p_{0,i}) = (c_i, p_{0,i})$$

5. Reported by Mark Hillebrand, Saarland University
   Page 29: the optimization for the g output at the bottom of the page is wrong, the derivation should read $g = \overline{\overline{g_2 \vee g_1 \wedge p_2}} = \overline{\overline{g_2} \wedge \overline{g_1 \wedge p_2}}$. Cost and delay are not affected since $C_{\mathrm{nor}} = C_{\mathrm{nand}}$ and $D_{\mathrm{nor}} = D_{\mathrm{nand}}$.

6. Reported by Dirk Leinenbach, Saarland University
   Page 31: at the end of first paragraph it should be $s_{n-1} = p_{n-1} \oplus c_{n-2}$ instead of $s_{n-1} = p_{n-1} \oplus c_{n-1}$.

7. Reported by Dirk Leinenbach, Saarland University
   Page 33: indices in Figure 2.24 are wrong. It should be r_{n-i} and r_{n-i-1} instead of r_i and r_{i-1}.

8. Reported by Warren E. Ferguson, Intel
   Page 44: since $\langle a \rangle \in \{0, \ldots, 2^n - 1\}$ and $B_{2j} \in \{-2, \ldots, 2\}$, it holds
   $$C_{2j} \in \{-2^{n+1} + 2, \ldots, 2^{n+1} - 2\},$$
   $$D_{2j} \in \{0, \ldots, 2^{n+1} - 2\}.$$
   Since D_{2j} has an $(n+1)$–bit representation d_{2j} (which was not the case for the faulty range $D_{2j} \in \{0, \ldots, 2^{n+1}\}$), this does not affect the further arguments.

9. Reported by Jörg Fischer, Saarland University
   Page 46: The definition of $S'_{2j,2k}$ must be

$$S'_{2j,2k} := \sum_{t=j}^{j+k-1} \langle g_{2k} \rangle \cdot 4^{t-1}.$$

10. Reported by Jörg Fischer, Saarland University
    Page 47, Lemma 2.7, induction base: $\cdots < 2^{n+6} \cdot 2^{2j-2} = \ldots.$

    Induction step: the right bracket $\rangle$ is in the wrong place.

11. Reported by Jochen Preiss, Saarland University
    Page 47, Lemma 2.7: the second line of the inequality chain must be

$$\cdots < 2^{n+2j+2k} + 2^{n+5} \cdot 2^{2j+2k-4}.$$

12. Reported by Mark Hillebrand, Saarland University
    Page 60: in the formula for the accumulated delay $A(O(i))$ of the output circuit $O(i)$ the data path circuits should be indexed by $j-1$, i.e.,

$$A_{O(i)} = D_{O(1)} + \sum_{j=2}^{i} (D_{DP(j-1)} + D_{O(j)}) \ .$$

13. Reported by Jochen Preiss, Saarland University
    Page 67, line 1: replace beqz by sgri.

14. Reported by Mark Hillebrand, Saarland University
    Page 90, Figure 3.20: the label of the sucessor state of $jalR$ and $jalI$ should be changed from wbI to wbL.

15. Reported by Mark Hillebrand, Saarland University
    Page 94, Table 3.12:

    - In the decode state, the output signal Pce should read as $PCce$.
    - In the decode state, the signal $shiftI$ should only be activated conditionally depending on the instruction word. This would make the automaton a Mealy automaton. To avoid this, the condition $shiftI(IR)$ can be computed in the locally in the instruction register environment.

16. Reported by Richard Pfeifer, Saarland University
    Page 110f., Theorem 4.1: using the induction hypothesis for $i-2$ in the induction step is not well-founded. Instead, if $bjtaken_i = 1$, the proof of $PC'_i = PC_{i+1}$ can be changed as follows:

$$\begin{aligned}
PC'_i &= PC'_{i-1} + imm_i && \text{(because } bjtaken_i = 1\text{)} \\
&= PC_i + imm_i && \text{(by the induction hypothesis} PC'_{i-1} = PC_i\text{)} \\
&= PC_{i-1} + 4 + imm_i && \text{(because} bjtaken_{i-1} = 0\text{)} \\
&= btarget_i \\
&= PC_{i+1} && \text{(because } bjtaken_i = 1\text{)}
\end{aligned}$$

17. Reported by Richard Pfeifer, Saarland University
Page 148f., Section 4.4.3: in the first three displayed formulas in this section it should read

$$I_\pi(1,T) = i \quad \text{instead of} \quad I_\pi(k,T) = 1\ ,$$
$$I_\pi(1+\alpha,T) = i-\alpha \quad \text{instead of} \quad I_\pi(1+\alpha T) = i-\alpha\ \text{, and}$$
$$I_\sigma(1+\alpha,T') = i-\alpha \quad \text{instead of} \quad I_\sigma(1+\alpha T') = i-\alpha\ .$$

18. Reported by Richard Pfeifer, Saarland University
Page 149, Lemma 4.9: the hypothesis for this lemma should read 'Suppose the hypothesis of Theorem 4.7 holds'.

19. Reported by Mark Hillebrand, Saarland University
Page 132, Duration of Reset: the reset signal must be active long enough to permit an instruction memory access *and* the deactivation of reset must coincide with an acknowledgment of the instruction memory (otherwise an already updated $PC' = 4$ might be used for the initial instruction fetch).

20. Reported by Philipp Ritter, Saarland University
Typo: on Page 301 in Figure 6.23 the input signal $DMRw$ should read as $MDRw[31:0]$, as in the text

21. Reported by Dirk Leinenbach, Saarland University
Page 322, property 3 of representable numbers: $(-2^{-e_{min}}, 0]$ must be $(-2^{e_{min}}, 0]$

22. Reported by Christoph Berg, Saarland University
Page 323, 4th paragraph: $e = [\![e[n-1]]\!]_{bias}$ must be $e = [\![e[n-1:0]]\!]_{bias}$

23. Reported by Dirk Leinenbach, Saarland University
Page 329, end of last line of proof of Lemma 7.1: it should be $= r(x')$ instead of $= r'(x)$.

24. Reported by Chris Jacobi, Saarland University
In the `unpack`-circuit (page 355), sub-circuits `lzero(53)` and `CLS(53)` are used, even though these circuits were only designed for powers of two in Chapter 2.

One can derive `lzero(53)` from `lzero(64)` by padding the input with 1's, since $lz(x) = lz(x\ 1^k)$.

The cyclic shifter `CLS(53)` in the `unpack`-circuits can be replaced by a logic right shifter `LRS(53)`. It is easy to design logic right shifters for non-power-of-two inputs.

25. Reported by Chris Jacobi, Saarland University
Page 383. In the circuit `Sign/ExpMD`, a carry-in is fed into the 4/2-adder, although 4/2-adders do not feature a carry-in. To fix this, add a 3/2-adder(7) which adds `lza`, `lzb` (or the inverse) and a constant 1. The output of the new 3/2-adder is sign-extended and fed into the 4/2 adder, instead of the inputs `lza` and `lzb`.

26. Reported by Chris Jacobi, Saarland University
Page 392, 5th item: the input factoring shall not satisfy $f_r[-1:0] = 00 \Rightarrow OVF = 0$, but

$$f_r[-1:0] = 00 \Rightarrow e \leq e_{max}.$$

Otherwise, the correctness argument for the OVF1-computation in circuit `Flags` is wrong (e.g. $e_r = e_{max} + 1, f_r = 0.5$, i.e. $f_r[-1:0] = 00$; no overflow occurs, although OVF1 is asserted).

The new condition for the input factoring is satisfied

- by the adder, since the delivered exponent is the maximum of the input exponents, and hence $e_r \leq e_{max}$.
- by the multiplier, since it delivers $f_r < 1$ only if one the the operands is denormal, and hence the sum if the input exponents $e_r \leq e_{max}$.

27. Reported by Chris Jacobi, Saarland University
In circuits `ExpNorm` (page 400), a carry-in is fed into the compound-adder, although compound adders do not feature a carry-in (cf. Chapter 2). To fix this, incorporate the constant increment by 1 into the constant in the 3/2-adder.

28. Reported by Chris Jacobi, Saarland University
Page 407: the circuit for the rounding decision (Figure 8.34) does not conform with Table 8.5 (e.g., $s = 0, r = st = 1$, mode r_u). Replace the `XOR` gate by an `XNOR` gate.