Known errata in *A Pipelined Multi-core MIPS Machine, Hardware Implementation and Correctness Proof*

Saarland University, Computer Science

Institut für Rechnerarchitektur und Parallelrechner

December 16, 2015

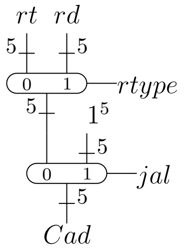
Nine Shades of RAM

1. p.91, typo in the name of the address input, should be

$$Sout(h) = h.S(Sa(h))$$

A Basic Sequential MIPS Machine

2. p.145, figure 99: the figure is wrong, should be



3. p.146, figure 100: typo in the name of the fields of the BCE unit, should be

$$bf[3:1] \quad \text{and} \quad bf[0].$$

Pipelining

4. p.164: typo in the text, should be

...into 5 circuit stages $cir(i)$ with $i \in [1:5]$, such that...

5. p.167: typo in the text, should be

For $k \in [1:5]$ circuit stage $cir(k)$ is input for register stage k and...

6. p.189: typos in the proof, should be

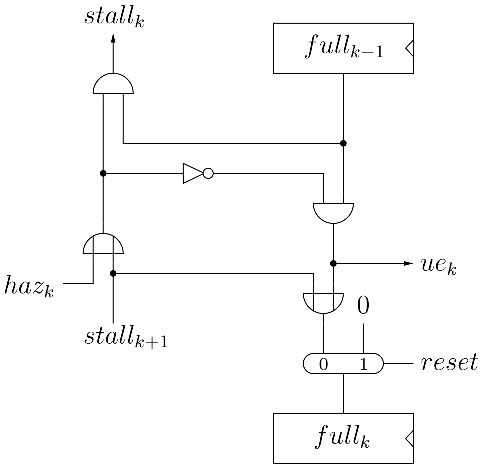
$$\begin{aligned} gprin^t_\pi &= lres^t_\pi \\ &= lres^i_\sigma \\ &= gprin^i_\sigma \end{aligned}$$

and

$$\begin{aligned} gprin^t_\pi &= C.4^t_\pi \\ &= C^i_\sigma \\ &= gprin^i_\sigma \end{aligned}$$

in the cases of resp. load and not-store instructions.

7. p.198, figure 123: the figure is wrong, the multiplexer is missing, should be



Caches and Shared Memory

8. p.214, proof of Lemma 8.1: typo in the first line, should be

$$hhit(h.ca, a) \equiv h.ca.s(a.c) \neq I \wedge h.ca.tag(a.c) = a.t$$

9. p.218: definition of the outputs of the fully associative cache, should be

$$h.ca.data(b) \quad \text{and} \quad h.ca.s(b)$$

10. p.223: typo in the notation, should be

$$aca(i).X^t = aca(h^t.ca(i)).X$$

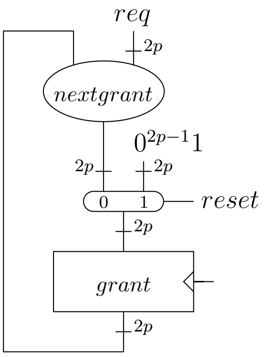
11. p.232: typo in the definition of semantics of the global access, should be

$$\begin{array}{rcl} & \vdots & \\ \forall j : sprot(j) & = & C2(aca(j).s(a), mprot) \\ & \vdots & \end{array}$$

12. p.246, figure 137, comment:

    For *souta* there is a multiplexer forcing the state of the abstract cache to invalid in case of */phit*. For the slave state *soutb* the corresponding mux is not present. The reason is, that the slave only participates in the protocol (and the output of circuit $C2$ is only used) if $bhit = 1$.

13. p.254, figure 140: the figure is wrong, the multiplexer is missing, should be



14. p.262, lemma 8.16: one case is not covered in the proof. When considering $G(i)^{t-1}$ we should neither be in state $w(i)^{t-1}$ (which is present) nor take transition (9) (which is missing). However, the statement of the lemma still holds, while the proof is easily fixable by adding the missing case.

15. p.286, lemma 8.51: typo in the proof, should be

$$q = \max\{t' \mid wait(i)^{t'} \wedge t' < t\}$$

16. p.290, lemma 8.56: typo in the formulation, in the second line it should be

$$\neg acc(i,k).f$$

17. p.291, lemma 8.57: typo in the formulation, should be

$$X \in \{s, tag, data\}$$

18. p.294, lemma 8.58: typo in the formulation, should be

$$X \in \{s, tag, data\}$$

19. p.298, lemma 8.62: the proof uses $\neg acc(i,k).f$

In the proof of lemma 8.64 (one step) the application of lemmas 8.58 (stable slaves) and the data transfer lemma 8.35 (m1) does not work out. Splitting cases on $localw$ in the three lemmas concerned fixes this problem.

20. Lemma 8.35 (m1) ignores the multiplexer at the s input of circuit $C2$ forwarding the next state

$$M = aca(j).s^{t+1}(badin(j)^t)$$

in case of a local write. Thus the correct statement is

$$\begin{aligned} sprotout(j)^{t+1} &= \begin{cases} C2(M, mprotin(j)^t) & localw(j)^t \wedge ca(j).pa^t = badin(j)^t \\ C2(soutb(j)^t, mprotin(j)^t) & \text{otherwise} \end{cases} \\ &= \begin{cases} C2(aca(j).s^{t+1}(badin(j)^t), mprotin(j)^t) & localw(j)^t \wedge ca(j).pa^t = badin(j)^t \\ C2(aca(j).s^t(badin(j)^t), mprotin(j)^t) & \text{otherwise} \end{cases} \end{aligned}$$

21. Lemma 8.58 (stable slaves) must hold for one more cycle in the absence of local writes

$$q(j) \in \begin{cases} [s(i,k)+2 : t] & localw(j)^{s(i,k)+1} \wedge ca(j).pa^{s(i,k)+1} = a \\ [s(i,k)+1 : t] & \text{otherwise} \end{cases}$$

In cycle $s(i,k)+1$ only local writes to address a can end. If no such write ends, one concludes with lemma 8.57 (unchanged cache lines)

$$aca(j).X^{s(i,k)+1}(a) = aca(j).X^{s(i,k)+2}(a)$$

22. In the proof of lemma 8.64 (one step) one uses in the case $w(i)^{e(i,k)}$ first the data transfer lemma 8.35 with $t = s(i,k)+1$ and then lemma 8.58 literally to conclude

$$\begin{aligned} sprotout(j)^{s(i,k)+2} &= \begin{cases} C2(aca(j).s^{s(i,k)+2}(a), mprotin(j)^{s(i,k)+1}).(ch,di) & localw(j)^{s(i,k)+1} \wedge \\ & ca(j).pa^{s(i,k)+1} = a \\ C2(aca(j).s^{s(i,k)+1}(a), mprotin(j)^{s(i,k)+1}).(ch,di) & \text{otherwise} \end{cases} \\ &= C2(aca(j).s^{e(i,k)}(a), mprotout(i)^{e(i,k)}).(ch,di) \end{aligned}$$

A Multi-core Processor

23. p.329: typo in the definition of the instruction cache interface, should be

$$Iin_\pi = \begin{cases} ica.pdout[63:32] & ima_\pi[2] = 1 \\ ica.pdout[31:0] & ima_\pi[2] = 0 \end{cases}$$

24. p.329: typo in the definition of the data cache interface, should be

$$dmout_\pi = dca.pdout$$

25. p.340: index i is incorrect, should be i_y