Precise Read Control in a File Design of Low Power Consuming Register

Rashi Malhotra

Research Scholar, Singhania University

Abstract:- Only instructions such as register-register arithmetic, store, conditional-branch, and shift instructions require fetching both source operands. Traditionally, battery operated products have represented a key application of low power electronics. A lot of powers saving techniques have been developed for these kinds of applications.

---♦-------------------------------------

1.1 MOTIVATION

However, traditional embedded processors, which meet the price and power objectives of battery operated products, cannot deliver the performance required by new applications such as interactive digital media products. To the gap, new microprocessors appeared that are exclusively focused on low-cost and low-power applications. The Strong ARM microprocessor is the 1st example of this generation of high performance embedded microprocessors. For these products, power was reduced by lowering supply voltage, using low-power circuit/logic techniques, reducing functionality, reducing control complexity and using slower clock frequencies.

These processors are usually no more than 32-bit wide, and they are typically implemented as simple single issue, in-order pipelines [25, 72]. Such processors are highly optimized for power, however, the listed features also result in a significant performance loss, which is not an option in the high-performance market. Therefore, a large fraction of source operand data is discarded because of over fetching of operands from the regfile. Over fetching operands creates extra unnecessary regfile switching activity contributing to the power consumption. The measurement profiling shown in Figure 3-1 shows on average, each instruction requires 1.3 source operands; 70% of dynamic instructions require only one source operand. So, a precise-read-control regfile has an potential of decreasing the regfile read activity by 35%.

1.2 IMPLEMENTATION

One of the most straightforward implementations of precise read control is by adding an opcode pre-decoder prior to the word line drivers in the regfile as shown in Figure 1-2. a hardware unit is introduced for performing the register renaming. This approach can be very useful when there is a need to preserve binary compatibility with earlier architectures without the need for recompilation or any modification of the software binaries. The major disadvantage of a hardware rename unit is that it incurs area, timing, and power overheads. Additional hardware incurs an area overhead; the mapping extends the register file access time, and the active circuitry leads to larger dynamic power dissipation during access. However, if the timing overhead imposed by hardware rename unit is acceptable, then the approach is still superior to the monolithic register file from an energy-efficiency point of view. We find from our experiments that the area, timing and energy overheads of such a hardware rename unit are: 3%, 60% and 10% respectively over the Software approach. Several approaches can be taken to handle the renaming problem in software, starting from re-compilation of the source code to a post-compilation approach where the register numbers in the software binary are re-numbered after the code generation. We have chosen the latter approach in this work, as it is the closest to offering binary compatibility for existing software, while still benefiting from register banking. This approach does not incur any time, area and energy overheads as in the hardware approach.

In many RISC processors, Register 0 is hard-wired to zero, implying that this register is not subject to renaming. Various other registers may have specific meanings according to the respective calling conventions, however, in an application specific environment, it may be permissible to violate them, as long as we have access to the entire application binary (and consequently, ensure that the renaming is consistent over the entire application).

When the word line is not enabled, the read bit line value retains its pre-charged value and no switching occurs. We also keep the pre-charge transistors turned on to avoid switching their gate capacitance. The precise-read-control regfile has only an AND-gate area overhead because the opcode pre-decoders are part of the original bypassing interlock circuit. There is no latency overhead if the opcode pre-decoder utilizes the first half of the cycle to finish performing all its necessary decoding and is able to provide the issue signal in time for the read bit line enables in the second half of the cycle. However, if the opcode decoding cannot finish in the first half of the cycle, precise read control is going to add latency to the regfile. The precise-read-control regfile handles NOP instructions differently from shift left logical (SLL) instructions even though they have the same opcode. Since NOP instructions do not require any operands, the opcode decoders disables both read operand fetches.

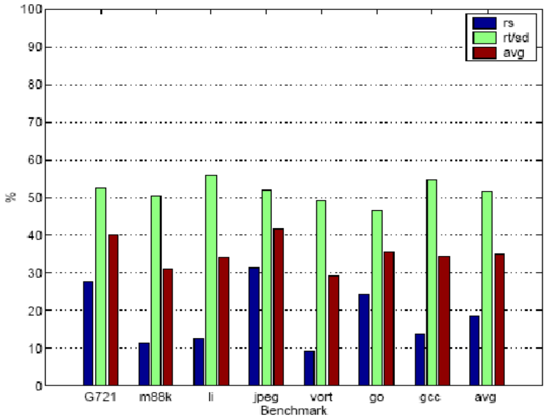

Figure 1-1: Percentage of discarded operands due to over fetching.

1.3 RESULTS

Figure 1-3 shows the power savings of precise-read-control in comparison with the base case scenario. The power saving ranges from 16% to 31% across measurements with an average of 23%. Our technical approach to the problem posed above consists of three steps:

First, we identify targets for power reduction within microprocessor architectures.

At this step we determine where the power is heavily consumed, or will be heavily consumed in next-generation processors and why.

Second, we reduce power consumption at the identied architecture and design points with minimal performance impact. This step involves determining new traders in the design of micro architecture between the performance and power.

As a third step we developed a methodology for optimizing and comparing different micro-architectures for energy efficiency. Extensive simulation of the baseline and proposed micro-architectures is used to prove the potential improvement of the energy efficiency.

In order to accomplish the 1st step, we have developed basic energy models for the most critical structures of the chip. Since the future growth in performance of modern superscalar processors is predicted on exploiting higher and higher levels of Instruction-Level Parallelism (ILP), particular attention is given to those structures in a micro architecture where energy per access grows with increasing amount of ILP exploited by a processor. Latches deserve special consideration on their own, because of their highest importance for both performance and power considerations.

When building energy models for the critical structures of a superscalar microprocessor, we are mostly interested in relative energy estimates that would allow us to compare energy complexity of dierent architectures. Since very accurate absolute values are not needed for the architectural level analysis, we tried to keep the energy models simple. On the other hand we included into the energy models the latest circuit-level innovations that could improve the energy efficiency of the critical structures. Actually, we attempted to 2nd the lower bound on the energy dissipation that can be achieved or approached by dierent circuit techniques. This makes our energy models particularly valuable for architectural studies.

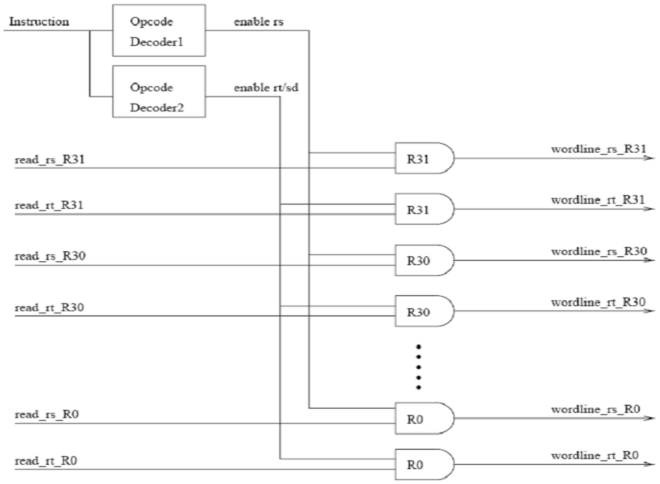

Figure 1-2: Implementation of precise read control.

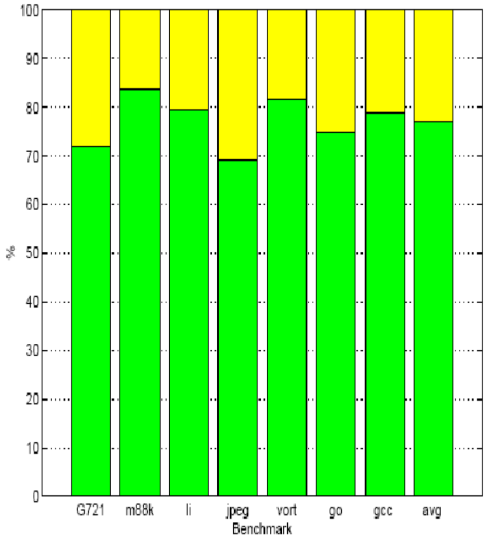

Figure 1-3: Comparative power consumption for the base case regfile and the precise-read control regfile.

REFERENCES:-

1. K. Asanovi´c. Vector Microprocessors. PhD thesis, University of California at Berkeley, May 1998.

2. T. D. Burd and B. Peters. Power analysis of a microprocessor: A study of an implementation of the MIPS R3000. Technical report, ERL Technical Report, University of California, Berkeley, May 1994.

3. R. Y. Chen, R. M. Owens,M. J. Irwin, and R. S. Bajwa. Validation of an architectural level power analysis technique. In DAC '98. Proceedings of the 35th Annual Design Automation Conference, San Francisco, CA, June 1998.

4. D. R. Gonzales. Micro-RISC architecture for the wireless market. IEEE Micro, 19(4):30–37, July/August 1999.

5. J. L. Hennessy and D. A. Patterson. Computer Architecture — A Quantitative Approach, Second Edition. Morgan Kaufmann, 1996.

6. L. Huffman and D. Graves. MIPSpro Assembly Language Programmer's Guide. Technical Report 007-2418-002, Technical Report, Silicon Graphics, 1996.

7. A. Kalambur and M. J. Irwin. An extended addressing mode for low power. In Proceedings of the IEEE Symposium on Low Power Electronics, pages 208–213, August 1997.

8. G. Kane and J. Heinrich. MIPS RISC Architecture (R2000/R3000). Prentice Hall, 1992.

9. R. E. Kessler. The Alpha 21264 microprocessor. IEEE Micro, 19(2):24–36, March/April 1999.

10. P. Landman. High-level power estimation. In Proceedings ISLPED, pages 29–35, Monterey, CA, 1996.

11. L. Nagel. SPICE2. Technical Report ERL-M520, ERL Technical Memo, University of California, Berkeley, 1975.

12. J. Ousterhout, G. Hamachi, R. Mayo,W. Scott, and G. Taylor. Magic: A VLSI Layout System. Proc. 21st Design Automation Conference, pages 152–159, 1984.

13. J. Rabaey. Digital Integrated Circuites. Prentice Hall, 1996.

14. J. Scott. Designing the low-power M*CORE architecture. In Power Driven Micro architecture Workshop at ISCA98, Barcelona, Spain, June 1998.

15. V. Stojanovic and V. G. Oklobdzija. Comparative analysis of master-slave latches and flip-flops for high-performance and low-power system. IEEE Journal of Solid-State Circuits, 34(4):536–548, April 1999.

16. M. Tremblay, B. Joy, and K. Shin. A three dimensional register data for superscalar processors. In Proceedings of the 28th Annual Hawaii International Conference on System Sciences, pages 191–201, January 1995.

17. N.P. van der Meijs and A.J. van Genderen. SPACE Tutorial. Technical Report ET-NT 92.22, Technical Report, Delft University of Technology, Netherlands, 1992.

18. N. Weste and K. Eshraghian. Principles of CMOS VLSI Design, Second Edition. Addison Wesley, 1993.

19. V. Zyuban and P. Kogge. Split register data architectures for inherently low power microprocessors. In Power Driven Micro architecture Workshop at ISCA98, Barcelona, Spain, June 1998.