Disclaimer: "The contents of this document are scribe notes for The University of Texas at Austin EE382V Spring 2007, Computer Architecture: User System Interplay". The notes capture the class discussion and may contain erroneous and unverified information and comments.

Low Power and Circuit Variability

Lecture #09: Monday, 19 February 2007
Lecturer: Mattan Erez
Scribe: Jae Wook Lee
Reviewer: Mattan Erez

1 Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation

1.1 Reference

[1] Dan Ernst, Nam Sung Kim, Shidhartha Das, Sanjay Pant, Rajeev Rao, Toan Pham, Conrad Ziesler, David Blaauw, Todd Austin, Krisztian Flautner, Trevor Mudge, "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation," *micro, p. 7, 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-36), 2003.*

2 Discussion Points

2.1 What is the problem being solved?

- Minimize power usage by **maximizing** voltage scaling
- Better than worse case or average case. But, a big concept in this paper is adjusting to the worst-case while occasionally exceeding it and doing better than the conventional margins.
- Secondary, recovering from circuit-level speculation

2.2 Who are the intended users?

2.2.1 Who are the intended readers?

- Computer architects
- Circuit designers implementing microarchitecture in a circuit level: circuit-level techniques are invisible to computer architects in some degree
- EDA tools (designer)
- System/Chip-level power supply designers

2.2.2 Who are the intended users who will actually get benefits from this idea?

- Mobile users who must be able to tolerate unexpected delay(slowdown) - up to 50%.
  - There are minor variations in performance over entire running time. The performance is not degraded much in overall, but significant momentarily.
  - Reducing power consumption is universally good - not going to make any harm to users
  - A voltage modulator used here is a slow device, and if there are some rapid environmental changes affecting to the voltage regulator, users will be affected also.
- Data centers
- Lowering margins may increase soft-error propensity, which is due to noise, not much for particle strikes.

2.3 What is unique about the suggested solution?

2.3.1 What is Razor?

- Two paths: Razor utilizes two latches. One latch assumes the previous path was fast, and the second latch with delayed timing checks that it really was.
- Designed to tolerate timing violations, not computation error. That is, they didn't intend to check logic circuit itself.
- Suggested method is only applicable to between pipelining logic.

2.3.2 What is unique in Razor?

- Latch speculation and recovery ← global, "cut-in-closer"
  - Shadow latch is not unique.
- How does it compare to triple-latch ← local
  - Which one is area efficient? Which one is energy efficient? If we choose circuit-level approach, we don't need to speculate and do recovery action, but this technique is confined to local area. On the other hand, latch speculation and recovery mechanism in the Razor is more global and more aggressive. Although it needs recovery and re-execution overhead when there is something wrong, this technique can give us more performance potentially. Thus, there is some trade-off in terms of area, complexity, or performance.
- Assumption: local voltage regulators
  - The voltage regulator does not cover the entire chip, but their target area. So, the data dependency can give us more headroom compared to global voltage regulator.
- Operate outside margins: take errors and recover from it.
- Introduce control mechanism to balance power
- Meta-stability detector
  - More susceptible to the meta-stability problem with voltage supply less than marginal one.
- Creative adaptation of the counterflow pipeline
  - **Counterflow pipeline**: The name of counterflow pipeline came from the fact that data values propagate in one way, and control signal generated propagates in the the other way. There is no global control. If we reach a certain stage, and that stage generates a stall, the stall signal propagates in the opposite direction to the data flow direction, while the stall signal propagates globally in the global control scheme. So, this is a distributed control.
- Benefits of asynchronous design in a synchronous flow
  - Asynchronous design uses lower power, can operate the highest possible speed for given computation, but has to deal with handshake overheads.
- Simplifies verification: kinds of true, and we'll talk about DIVA later.

2.4 How is the idea evaluated?

2.4.1 Evaluations and some critiques

- A little bit of everything to convince different aspects of their idea to the extend they are concerned.
  - They built prototype circuits. To analyze how it behaves on the program, they have a SPCIE model. To know how it reacts in terms of whole program, they built C-Model - cycle-level simulator- reflecting the SPICE model, since the SPICE model is slow. Basically, they did modelling, emulating, and simulating to show different aspects of their idea.
- But, they ignored recovery power overhead. The worst case, however, is bounded to 50%.
- " Memory accesses are non-speculative", but addresses pass through Razor F/F.
- Evaluating pipeline with adder only. How about the rest of circuit in terms of voltage scaling?
  - Actually, ALU of modern superscalar processor consumes relatively small fracture of power in the entire processor.
- Not giving some specific numbers showing Razor's superior performance over the classic DVS techniques, but just mentioning that.

2.4.2 Problems / Improvement

- Control depends on actual data
- Razor on critical path only(Shouldn't all paths be critical)
  - When design is done optimally, many paths are close to critical. But, it is hard to judge early in the design stage, so potentially multiple iterations are needed(it is true in any case).
  - A half-cycle away clock was used and it can't be much further away from the that.
- Used spec to evaluate for embedded usage
- No sensitivity analysis
  - adder design: They used only one design of adder, Kogge-Stone adder. Other adder designs are possible and may offer very different trade-offs.
  - pipeline parameters

  - single f/V point
  - no analysis on the bottlenecks of the processor.

- Mostly on circuit-level analysis, but little implementation-level analysis

2.5 Was the evaluation in line with the stated user requirements?

True, but they could be more convincing. This problem has been discussed in the previous subsection.

2.6 Was technology a factor in the problem or solution?

- We always need less power.
- We need to cope with the process variability
- CAD designer
- Voltage regulator's technology: Currently, Razor is limited to the slow operation of voltage regulator.

2.7 Were new tools or software techniques introduced?

Definitely not part of this paper.

2.8 How may users with other requirements be affected?

- How about users who want to use other adder designs?
- How about users who usually use the adder 100% compared to whom with much less usage?
- They didn't show the slowdown profiling, just showed average slowdown
  - For example, in MPEG decoder, missing 1 frame in every 30 frame vs. 3 frames in a row in every 90 frame is a totally different thing.
- average performance analysis vs. real-time performance analysis

2.9 How can the discussion of this paper be generalized in the context of the class?

- Minimize power usage by adjusting voltage supply to the worst-case while occasionally exceeding it.
- Recover from circuit-level speculation.
- But, the technique was used only on critical path, which is hard to be identified early in the design stage.
- creative ways of solving problems
    - Another important point in this paper is creative ways of solving problems. To test circuit-level technique, here to measure circuit-level timing error, they built and used FPGA-model, which is an unique way to see the process variability on FPGA as well as to see their algorithm. Other things are using SPICE model to get good feelings on how things are and using these parameter to build C-level simulations.

The purposes of this class are as follows.

- To think about how the idea of paper was related to the big picture of that, not just seeing how the idea is evaluated.
- To see multiple ways of solving different problems in the different levels of system. There can be different ways of trade-off. For example, circuit level/microarchitecture level or microarchitecture level/OS level. This paper was chosen so that papers that don't necessarily only deal with microarchitecture could be brought out.
- To see how the evaluations are done and to criticize the evaluations.