Moscow-Bavarian Joint Advanced Student School 2011

RAZOR: circuit-level correction of timing errors for low-power operation

Shohaib Aboobacker

Technische Universität München

10.04.2011

Contents

1 Introduction

Low-power is the need of the hour in embedded systems. Power consumption has become very critical with increasing clock frequencies and silicon integration. Moreover, along with the low power budget, there is a need for higher levels of performance. For example, today's mobile phones have shown 50X improvement in talk-time per gram of battery, while performing new computationals tasks such as 3D graphics, multimedia and internet access. A single operating point is no longer sufficient to efficiently meet processing and power consumption requirements.

The gap between high performance and low power can be bridged by Dynamic Voltage Scaling (DVS). This means that the voltage can be adapted to meet performance demands of workload. Dynamic power can be reduced by decreasing the supply voltage since it scales quadratically with supply voltage. But this also leads to an increase in propagation delays, thus limiting the maximum frequency of operation.

Running systems at multiple frequency and voltage levels is challenging, because it must be ensured that the operation is correct at the required operating points. The minimum possible supply voltage that results in correct operation is referred to as the critical supply voltage. Below this voltage, delay-induced errors appear in the system. The critical supply voltage depends on a number of environmental and process-related variabilities. These include voltage drops in the power supply network, temperature fluctuations, changes in doping concentration and cross-coupling noise. These variabilities may be data-dependent, which means that they exhibit their worst-case impact on circuit performance only under certain instruction and data sequences, and are composed of both local and global components.

In the traditional approach to find the critical supply voltage, a conservaive supply voltage is selected using corner-analysis, and margins are added to the critical voltage to account for uncertainty in the circuit models and the worst-case combination of variabilities. But this is a very pessimistic approach since the worst-case scenario is highly improbable. Instead, the Razor approach, which is discussed in this paper, can be used. Razor is a new approach to DVS based on dynamic detection and correction of speed path failures in digital designs.

2 Razor

The Razor technique was developed at the Electrical Engineering and Computer Science Department of the University of Michigan. The key idea of Razor is to purposely operate the circuit at sub-critical voltage and tune the operating voltage by monitoring the error rate. This eliminates the need for conservative voltage margins. The trade-off would be between the power penalty incurred from error correction against the additional power savings obtained from operating at a lower supply voltage.

Operating at moderately sub-critical voltages causes only a few critical instructions to fail. If the non-zero error rate is maintaned sufficiently low, then the power overhead from

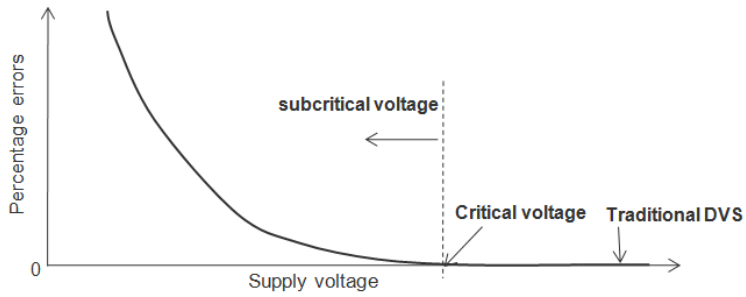

Figure 1: Razor operation at subcritical voltage

correcting these errors is minimal, while the power savings from the reduced operating voltage can be substantial.

2.1 Error detection and correction

In order to detect an error at the circuit level, each flip-flop is augmented by a shadow flip-flop, which is clocked by a delayed clock. If the combinational logic meets the setup time of the main flip-flop, then the main and delayed flip-flops will latch the same value. In this case, the error signal remains low. If the setup time of the main flip-flop is not met, then the main flip-flop will latch a value that is different from the shadow flip-flop. To guarantee that the shadow flip-flop always latches the input data correctly, the input voltage is constrained such that under the worst-case condition, the logic delay does not exceed the shadow flip-flop's setup time. The circuit is shown in Figure 2.

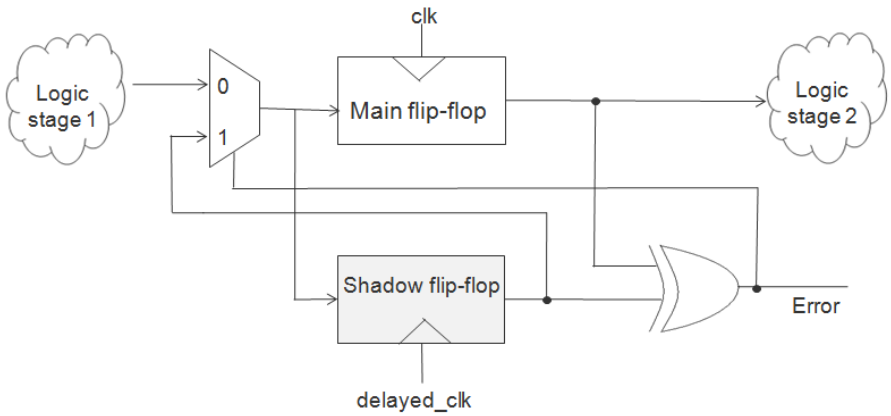

Figure 2: Razor circuit for error detection and correction

The operation of the error detection circuitry is illustrated in the timing diagram in Figure 3. In the first clock cycle, input data D1 meets the setup time of the main flip-flop

and shadow flip-flop, thus both flip-flops latch D1. But in the second clock cycle, the input data D2 does not satisfy the setup time requirement of the main flip-flop. It latches the old data D1 while the shadow latch latches D2. This is detected by the XOR circuitry and the error signal is asserted.

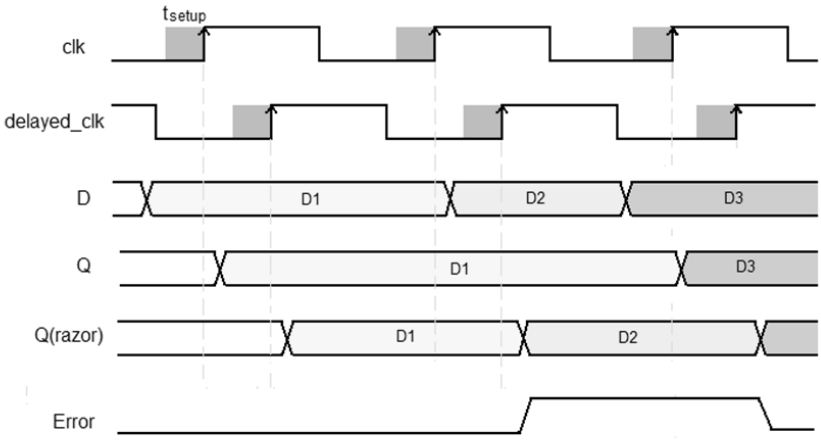

Figure 3: Razor operation

The asserted error signal is used in error correction, where it triggers the correct output value from the shadow latch to be restored to the main flip-flop in the subsequent cycle. This value is then available to the next pipeline stage.

2.2 Circuit-level implementation issues

The power overhead of the error detection and correction circuitry must be kept minimal in order to reap the rewards of the Razor approach. Otherwise the power gains obtained by reducing supply voltage would be cancelled out by the power overheads. One way to reduce the power overhead is by selectively replacing the flip-flops with Razor flip-flops. If the maximum delay at the input of a flip-flop is guarateed to meet the required time, then that flip-flop need not be replaced.

While the circuit is running at subcritical voltage, there is the danger of metastability. For instance, the input of the main latch may transition at the same time as the rising clock edge. Since the minimum critical voltage is constrained such that the setup time of the shadow flip-flop is always met, the shadow flip-flop will be stable. Since the main flip-flop feeds the XOR gate to generate the error signal, this signal would not be reliable in case of metastability. To resolve this, a metastability detector can be used and if metastability is detected, it can be corrected just as in the case of a regular delay failure.

The presence of the delayed clock leads to the possibility of a hold path violation for short paths. This is illustrated in Figure 4. If the logic delay for the input D1 is too small (fast path), then at the rising edge of the delayed clock, the Razor flip-flop can latch in the

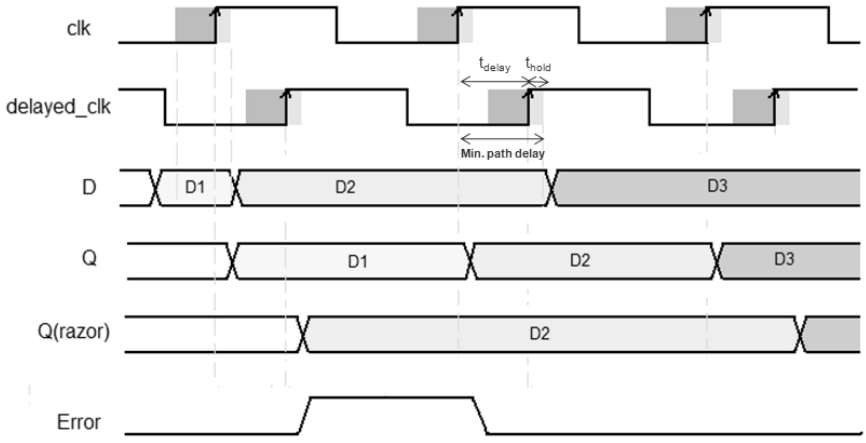

Figure 4: Short path constraint

new data D2, which was intended for the second cycle. This means that an error signal can be asserted even if the main flip-flop had latched in the correct data. This leads to a short path constraint where the minimum path delay t_{pd} is given by

$$t_{pd} = t_{delay} + t_{hold}$$

where t_{delay} is the delay between the main clock and the delayed clock and t_{hold} is the hold time of the shadow flip-flop. The relation means that a large clock delay increases the severity of the short path constraint and increases power overhead because of the need for additional buffers. On the other hand, a small clock delay reduces the margin between the main flip-flop and the shadow latch, and hence reduces the amount by which the supply voltage can be dropped below the critical supply voltage.

3 Pipeline error recovery using Razor

Pipelined processing is used to split a long logic path into stages so that the frequency of operation can be increased. A 5-stage pipeline is discussed here with the stages Instruction Fetch(IF), Instruction Decode(ID), Execute(EX), Memory Access(MEM) and Register Write-Back(WB). In the basic functioning of an error-free pipeline, each pipeline stage would work on one instruction at a time.

In this section, two approaches to implement pipeline error recovery will be discussed. The first is a simple but slow method based on clock gating, while the second method is a much more scalable technique based on counterflow pipelining.

3.1 Recovery with clock gating

If any pipeline stage detects an error, then the entire pipeline is stalled for one cycle by gating the next clock edge. During the additional clock cycle, each stage recomputes its result using the shadow flip-flop as input. Since all stages reevaluate their result, any number of errors can be tolerated in a single cycle and forward progress is guaranteed. The circuitry for this is illustrated in Figure 5.

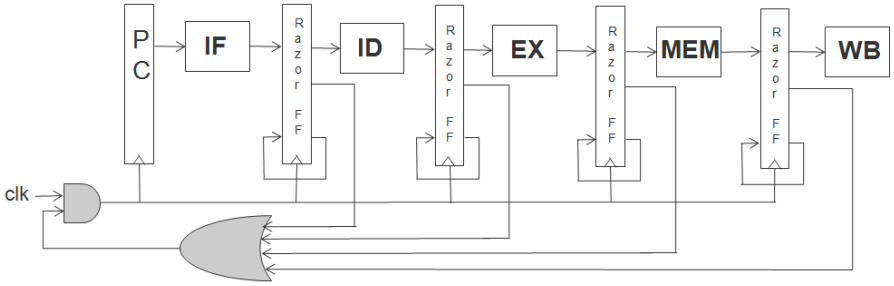

Figure 5: Clock gating mechanism for pipeline error recovery

The pipeline timing diagram in Figure 6 shows the pipeline recovery for an instruction that fails in the ID stage of the pipeline. The second instruction computes an incorrect result in the ID stage of the pipeline. This is detected in the 4th cycle, but only after the EX stage has computed an incorrect result using the errant value obtained from the ID stage. After the error is detected, there is a clock stall in the 5th cycle, when the correct value is taken from the shadow flip-flop of the ID stage and the EX stage is reevaluated. From the 6th cycle, normal pipeline operation resumes.

Cycle / Instr.	1	2	3	4	5	6	7	8	9	10
1	IF	ID	EX	MEM	stall	WB				
2		IF	ID*	EX*	EX	MEM	WB			
3			IF	ID	stall	EX	MEM	WB		
4				IF	stall	ID	EX	MEM	WB	
5					stall	IF	ID	EX	MEM	WB

Figure 6: Pipeline timing with clock gating

3.2 Recovery with counterflow pipelining

Clock gating is not useful in aggressively clocked designs because of its effect on processor cycle time. Instead, a fully pipelined error recovery mechanism based on counterflow

pipelining techniques can be used. This approach, illustrated in Figure 7, places negligible timing constraints on the baseline pipeline design at the expense of extending pipeline recovery over a few cycles. When a pipeline error is detected, a bubble signal is asserted that initiates the flushing of all the instruction stages following the errant instruction. Backward propogation of the flush train occurs, and the program counter is reset with the instruction following the failed instruction. Meanwhile, after the failed stage computation, the correct value from the Razor shadow flip-flop is used in the following cycle, allowing the errant instruction to continue with the correct inputs. Note that the instruction preceding the failed instruction was allowed to complete without interruption.

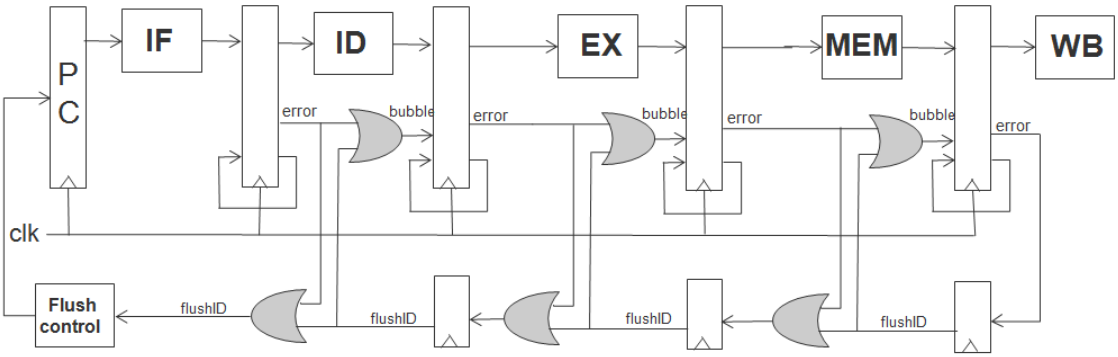

Figure 7: Counterflow mechanism for pipeline error recovery

Figure 8 shows the pipelined timing diagram with the counterflow mechanism. The 2nd instruction fails at the ID stage, which is detected in the 4th cycle. A bubble is then propagated from the EX stage towards WB, while the flush train is initiated towards the first stage. As a result, the IF and ID stages are flushed in the 5th and 6th cycles and the PC is set to instruction 3, which is restarted in the 7th cycle. Meanwhile, the errant instruction continued after a single-cycle stall, while instruction 1 was not affected at all.

Cycle / Instr.	1	2	3	4	5	6	7	8
1	IF	ID	EX	MEM	WB			
2		IF	ID*	EX*	bubble	MEM	WB	
3			IF	ID	flushID	flushIF	IF	ID
4				IF				IF

Figure 8: Counterflow mechanism for pipeline error recovery

4 Voltage control system

The voltage control system adjusts the supply voltage based on the monitored error rates. It works to maintain a constant small non-zero error rate. This optimal value depends on factors like cost of error recovery and performance requirements. At regular intervals the error rate of the system is measured. If the error rate is low, then there is a possibility to further lower the voltage. On the other hand, an increase in the error rate indicates failing timing constraints, and voltage should be increased. The Razor voltage control system is illustrated in Figure 9 . E_{ref} is the optimal error rate to which the measured rate E_{sample} is compared. The voltage regulator adjusts the supply voltage depending on the difference between E_{ref} and E_{sample}.

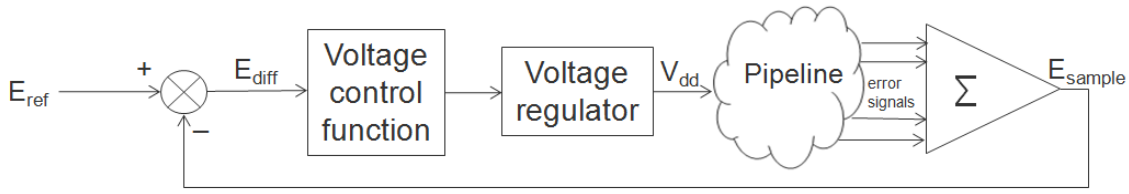

Figure 9: Voltage control system for Razor

Experimental results have shown that the power savings on reducing the supply voltage below critical voltage can be significant. The increase in error rate with decreasing supply voltage corresponds to an increase in the power required for error correction. Below a particular voltage, this power overhead would be greater than the processing power. Therefore, the optimum voltage would be the one which leads to a balance between the power savings with reduced supply voltage and the power overhead of correction. Figure 10 shows the relationship between the power consumption and the supply voltage.

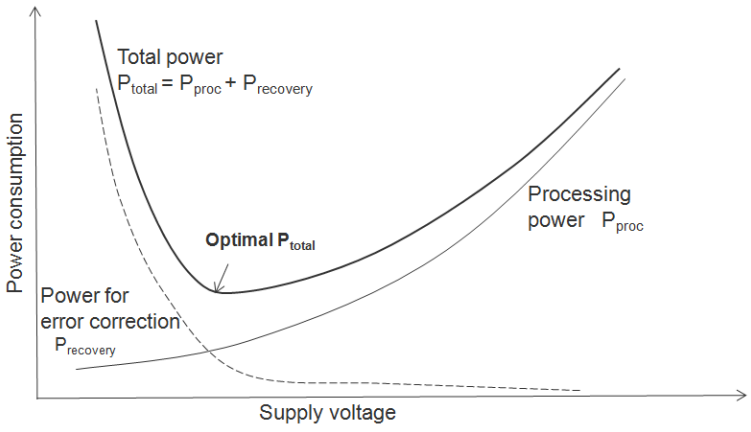

Figure 10: Razor power savings

5 Conclusion

The Razor approach to dynamic voltage scaling was presented in this paper. It involves operation at subcritical voltages and in-circuit detection and correction of the resulting errors. The shadow flip-flop is used to latch in the data with a delayed clock, and the issues in implementing such a system were discussed. Razor can be used in pipelined processing and the methods to recover a pipeline in case of a timing error were presented. The optimum supply voltage needs to be obtained, such that there is a balance between the power savings obtained by the reduced supply voltage and the power overhead due to error correction.

References

[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*, 36th Annual International Symposium on Microarchitecture (MICRO-36), December 2003