ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

A REVIEW OF LOW POWER PROCESSOR DESIGN

Nalini.D[1] ,Pradeep Kumar.P[2] ,Sundar Ganesh C S[3]

[1,2] Dept of Electrical and Electronics Engineering, PSG College of Technology, Coimbatore ,India
[3]Assistant Professor (SR.G), Dept of Robotics and Automation Engineering, PSG College of Technology, Coimbatore ,India
Email-Id:nalini.nithya@gmail.com,pradeep_nec@hotmail.com,css@raepsgtech.ac.in

Abstract

Power has become an important aspect in the design of general purpose processors. This paper gives a review of various technologies used for low power processor design. Scaling the technology is an attractive way to improve the energy efficiency of the processor. In a scaled technology a processor would dissipate less power for the same performance or higher performance for the same power. Some micro architectural changes, such as pipelining and caching, can significantly improve efficiency. Another attractive technique for reducing power dissipation is scaling the supply and threshold voltages. Unfortunately this makes the processor more sensitive to variations in process and operating conditions. Dynamic voltage scaling is one of the more effective and widely used methods for power-aware computing. DVS approach uses dynamic detection and correction of circuit timing errors to tune processor supply voltage and eliminate the need for voltage margins. Razor, a voltage-scaling technology based on Dynamic detection and correction of circuit timing errors, permits design optimizations that tune the energy in a microprocessor pipeline to typical circuit operational levels. This eliminates the voltage margins that traditional worst-case design methodologies require and allows digital systems to run correctly and robustly at the edge of minimum power consumption.

Keywords: Scaled technology, Dynamic voltage scaling (DVS), Error detection and correction

Introduction

Processor is the heart of the computer. There are lot many processor's in the market. When a processor is designed using processor cores i..e Hardware Description Languages like Verilog-HDL and VHDL (Very High Speed Integrated Circuit Hardware Description Language) it is called soft core processor. It is used for writing a particular version of processor. This helps the designer to check and select the processor for particular application. RISC (Reduced Instruction Set Computer) is an efficient Computer Architecture which can be used for the Low power and high speed applications of the processor. RISC Processors are important in application of pipelining. The heart of the processor is the Instruction Set Architecture (ISA) used for developing it. The total worthiness of the processor depends on utilizing the Instruction Set Architecture.

However a lot of research is being carried out in the field of processor's to satisfy the performance issues. But now a days it is mandatory to use a machine which is efficient in the terms of speed, power, performance and size. Though there are tradeoffs between all the performance parameter's, Research is being carried out to satisfy all the above performance parameters.

A critical concern for embedded systems is the need to deliver high levels of performance given ever-diminishing power budgets. This is evident in the evolution of the mobile phone: in the last 7 years mobile phones have shown a 50X improvement in talk-time per gram of battery1, while at the same time taking on new computational tasks that only recently appeared on desktop computers, such as 3D graphics, audio/video, internet access, and gaming. As the breadth of applications for these devices widens, a

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

single operating point is no longer sufficient to efficiently meet their processing and power consumption requirements.

Lowering clock frequency to the minimum required level exploits periods of low processor utilization and allows a corresponding reduction in supply voltage. Because dynamic energy scales quadratically with supply voltage, DVS can significantly reduce energy use. Enabling systems to run at multiple frequency and voltage levels is challenging and requires characterizing the processor to ensure correct operation at the required operating points. We call the minimum supply voltage that produces correct operation the *critical supply voltage*. This voltage must be sufficient to ensure correct operation in the face of numerous environmental and process-related variabilities that can affect circuit performance. These include unexpected voltage drops in the power supply network, temperature fluctuations, gate length and doping concentration variations, and cross-coupling noise. These variabilities can be data dependent, meaning that they exhibit their worst-case impact on circuit performance only under certain instruction and data sequences and that they comprise both local and global components. Christo Ananth et al. [10] discussed about a system, a low power area reduced and speed improved serial type daisy chain memory register also known as shift Register is proposed by using modified clock generator circuit and SSASPL (Static differential Sense Amplifier based Shared Pulsed Latch). This latch based shift register consumes low area and low power than other latches. There is a modified complementary pass logic based 4 bit clock pulse generator with low power and low area is proposed that generates small clock pulses with small pulse width. These pulses are given to the conventional shift register that results high speed. The system is designed by the Cadence virtuoso 180 nm technology. The Maximum supply voltage for the system, clock source and input source are 1.8V. The complementary pass logic based proposed system reduces the area about 7% for the total system and about 23% for the 4 bit clock pulse generator circuit. The Power is reduced by 26% than the conventional system. The speed is improved about 7% than the existing system. This approach to DVS has the advantage that it dynamically adjusts the operating voltage to account for global variations in supply voltage drop, temperature fluctuation, and process variations. However, it cannot account for local variations, such as local supply-voltage drops, intradie process variations, and cross-coupled noise. Christo Ananth et al. [12] proposed a system which can achieve a higher throughput and higher energy efficiency. The S-BOX is designed by using Advanced Encryption Standard (AES). The AES is a symmetric key standard for encryption and decryption of blocks of data. In encryption, the AES accepts a plaintext input, which is limited to 128 bits, and a key that can be specified to be 128 bits to generate the Cipher text. In decryption, the cipher text is converted to original one. By using this AES technique the original text is highly secured and the information is not broken by the intruder. From that, the design of S-BOX is used to protect the message and also achieve a high throughput, high energy efficiency and occupy less area.

Speed, Energy and Voltage Scaling

Both circuit speed and energy dissipation depend on voltage. The speed or clock frequency, f, of a digital circuit is proportional to the supply voltage, V_{dd}:

$$f \alpha \square V_{dd}$$

The energy E necessary to operate a digital circuit for a time duration T is the sum of two energy components:

$$E = SCV^2_{dd} + V_{dd} I_{leak} T$$

where the first term models the *dynamic power* lost from charging and discharging the capacitive loads within the circuit and the second term models the *static power* lost in passive leakage current—that is, the small amount of current that leaks through transistors even when they are turned off. The dynamic power loss depends on the total number of signal transitions, S, the total capacitance load of the circuit wire and gates, C, and the square of the supply voltage. The static power loss depends on the supply voltage, the rate of current leakage through the circuit, I_{leak}, and the duration of operation during which leakage occurs, T. The dependence of both speed and energy dissipation on supply voltage creates a tension in circuit design: To make a system fast, the design must utilize high voltage levels, which increases energy demands; to make a system energy efficient, the design must utilize low voltage levels, which reduces circuit performance.

Dynamic voltage scaling has emerged as a powerful technique to reduce circuit energy demands. In a DVS system, the application or operating system identifies periods of low processor utilization that can tolerate reduced frequency. With reduced frequency, similar reductions are possible in the supply voltage. Since dynamic power scales quadratically with supply voltage, DVS technology can significantly reduce energy consumption with little impact on perceived system performance.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

Conventional processor design

To minimize this power, Technology scaling, voltage scaling, clock frequency scaling, reduction of switching activity, etc., were widely used.

The two most common traditional, mainstream techniques are:

1. Clock Gating:

Clock gating is a technique which is shown in Fig. 1 for power reduction, in which the clock is disconnected from a device it drives when the data going into the device is not changing. This technique is used to minimize dynamic power.

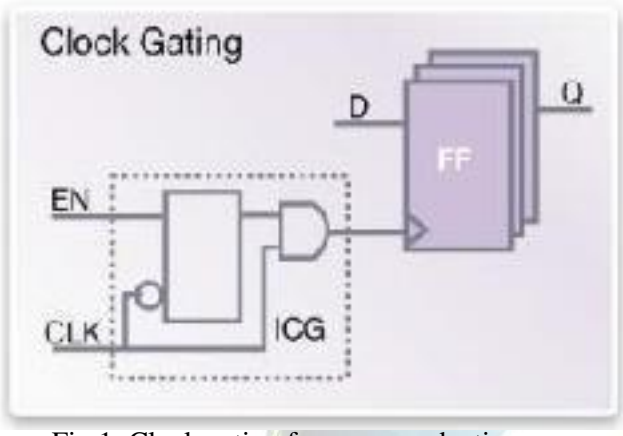

Fig 1: Clock gating for power reduction.

Clock gating is a mainstream low power design technique targeted at reducing dynamic power by disabling the clocks to inactive flip-flops.

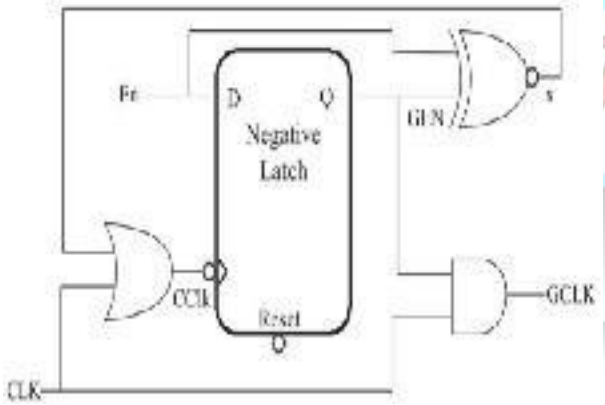

Fig 2: Generation of gated clock when negative latch is used.

To save more power, positive or negative latch can also be used as shown in Fig. 2 and Fig. 3. This saves power in such a way that even when target device's clock is 'ON', controlling device's clock is 'OFF'. Also when the target device's clock is 'OFF', then also controlling device's clock is 'OFF'. In this more power can be saved by avoiding unnecessary switching at clock net .

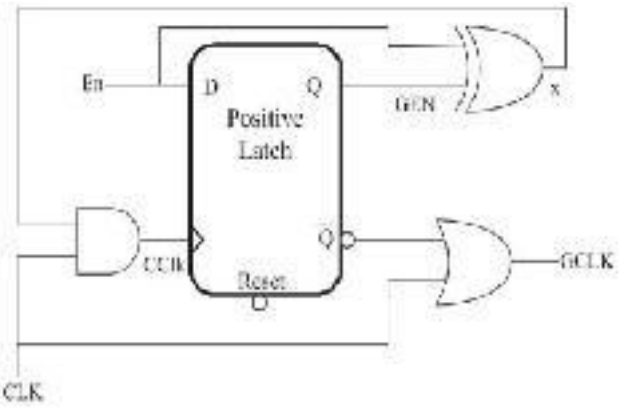

Fig 3: Generation of gated clock when positive latch is used.

2. Multi-Vth optimization/ (Multi Threshold - MTCMOS):

MTCMOS is the replacement of faster Low-*Vth* (Low threshold voltage) cells, which consume more leakage power, with slower High-*Vth* (high threshold voltage) cells, which consume less leakage power. Since the High-*Vth* cells are slower, this swapping can only be done on timing paths that have positive slack and thus can be allowed to slow down. Hence multiple threshold voltage techniques use both Low *Vt* and High *Vt* cells. It uses lower threshold gates on critical path while higher threshold gates off the critical path .

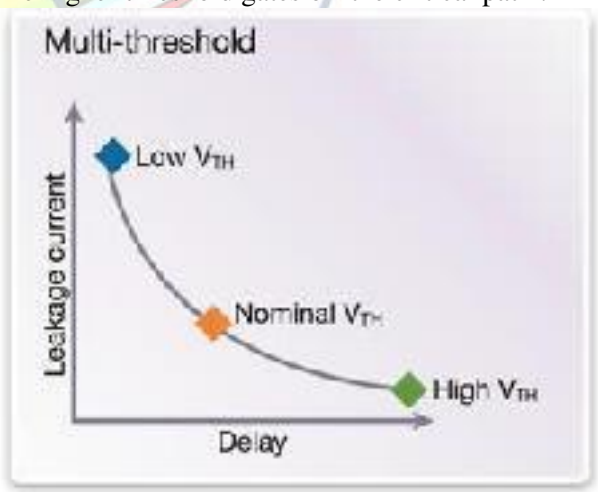

Fig 4: Variation of threshold voltage with respect to the delay and leakage current.

Above figure shows the variation of threshold voltage with respect to the delay and leakage current. As *Vt* increases, delay increases along with a decrease in leakage current. As *Vt* decreases, delay decreases along with an increase in leakage current. Thus an optimum value of *Vt* should be selected according to the presence of the gates in the critical path. As technologies have shrunk, leakage power consumption has grown exponentially, thus requiring more aggressive power reduction techniques to be used.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

Several advanced low power techniques have been developed to address these needs. The most commonly adopted techniques today are in below:

1) Dual *VDD*

A Dual *VDD* Configuration Logic Block and a Dual *VDD* routing matrix is shown in Fig.5.

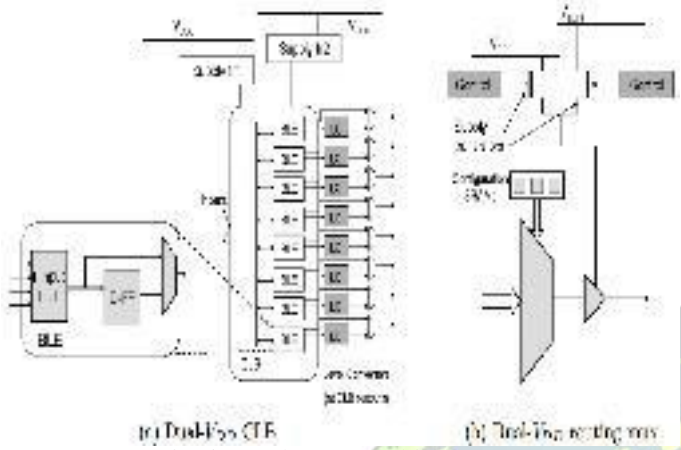

Fig 5: Dual *VDD* architecture.

In Dual *VDD* architecture, the supply voltage of the logic and routing blocks are programmed to reduce the power consumption by assigning low-*VDD* to non-critical paths in the design, while assigning high-*VDD* to the timing critical paths in the design to meet timing constraints as shown in Fig. 6. However, whenever two different supply voltages co-exist, static current flows at the interface of the *VDDL* part and the *VDDH* part. So level converters can be used to up convert a low *VDD* to a high *VDD*.

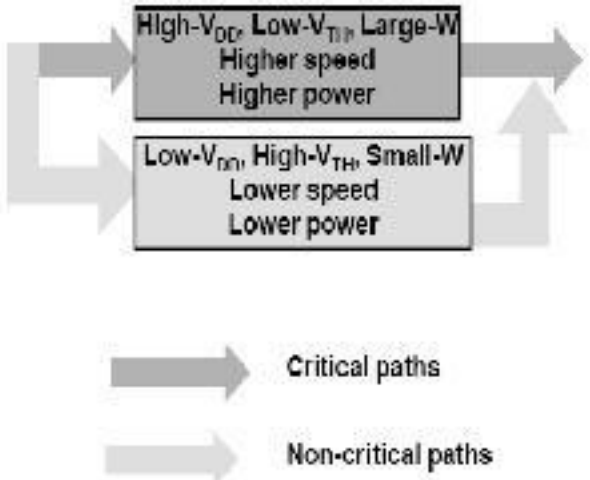

Fig 6: High *VDD* for critical paths and low *VDD* for non-critical paths.

2) Clustered Voltage Scaling (CVS)

This is a technique to reduce power without changing circuit performance by making use of two supply voltages. Gates of the critical path are run at the lower supply to reduce power, as shown in Fig. 7. To minimize the number of interfacing level converters needed, the circuits which operate at reduced voltages are clustered leading to clustered voltage scaling.

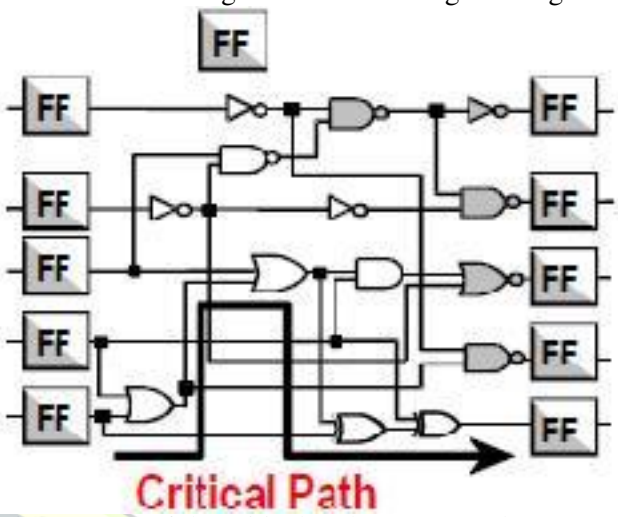

Fig 7: Gates of the critical paths are run at lower supply.

Here only one voltage transition is allowed along a path and level conversion takes place only at flipflops.

3) Multi-voltage (MV)

MV deals with the operation of different areas of a design at different voltage levels. Only specific areas that require a higher voltage to meet performance targets are connected to the higher voltage supplies. Other portions of the design operate at a lower voltage, allowing for significant power savings. Multi-voltage is generally a technique used to reduce dynamic power, but the lower voltage values also cause leakage power to be reduced.

4) Dynamic Voltage and Frequency Scaling (DVFS)

Modifying the operating voltage and/or frequency at which a device operates, while it is operational, such that the minimum voltage and/or frequency needed for proper operation of a particular mode is used is termed as DVFS, Dynamic Voltage and Frequency Scaling

Razor approach

Razor, a new approach to DVS, is based on dynamic detection and correction of speed path failures in digital designs. Its key idea is to tune the supply voltage by monitoring the error rate during operation. Because this error detection provides in-place monitoring of the actual circuit delay, it accounts for both global and local delay variations and doesn't suffer from voltage scaling disparities. It therefore eliminates the need for voltage margins to ensure always-correct circuit operation in traditional designs. In addition, a key Razor feature is that operation at subcritical supply voltages doesn't constitute a catastrophic failure but instead represents a trade-off between the power penalty incurred from error correction and the

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

additional power savings obtained from operating at a lower supply voltage.

Error-Tolerant DVS

Razor is an error-tolerant DVS technology. Its error-tolerance mechanisms eliminate the need for voltage margins that designing for "always correct" circuit operations requires. The improbability of the worst-case conditions that drive traditional circuit design underlies the technology.

Voltage margins

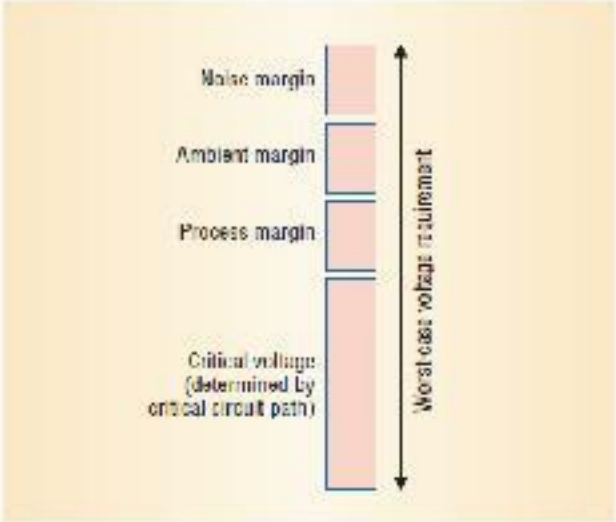

Fig 8: Critical voltage margins and to meet worst case reliability requirements

Above figure shows margins for factors that can affect the voltage required to reliably operate a processor's underlying circuitry for a given frequency setting. First, of course, the voltage must be sufficiently high to fully evaluate the longest circuit computation path in a single clock cycle. Circuit designers typically use static circuit-level timing analysis to identify this *critical voltage*. To the critical voltage, they add the following voltage margins to ensure that all circuits operate correctly even in the worst-case operating environment:

- *Process margins* ensure that performance uncertainties resulting from manufacturing variations in transistor dimensions and composition do not prevent slower devices from completing evaluation within a clock cycle. Designers find the margin necessary to accommodate slow devices by using pessimistically slow devices to evaluate the critical path's latency.
- *Ambient margins* accommodate slower circuit operations at high temperatures. The margin ensures correct operation at the worst-case temperature, which is typically 85-95°C.
- *Noise margins* safeguard against a variety of noise sources that introduce uncertainty in supply and signal voltage levels, such as di/dt noise in the supply voltage and cross-coupling noise in logic signals. The sum of these voltages defines the minimum supply voltage that ensures correct circuit operation in even the most adverse conditions.

The sum of these voltages defines the minimum supply voltage that ensures correct circuit operation in even the most adverse conditions.

Error rates

Fig 9 illustrates the relationship between voltage and error rates for an 18 × 18-bit Noise margin, Ambient margin, Process margin, Critical voltage (determined by critical circuit path) ,Worst-case voltage requirement multiplier block running with random input vectors at 90 MHz and 27°C. The error rates are given as a percentage on a log scale.

The graph also shows two important design points:

- *no margin*—the lowest voltage that can still guarantee error-free circuit operation at 27°C, and
- *full margin*—the voltage at which the circuit runs without errors at 85°C in the presence of worst-case process variation and signal noise. Traditional fault-avoidance design methodology sets the circuit voltage at the full margin point.

As Fig 10 shows, the multiplier circuit fails quite gracefully, taking nearly 180 mV to go from the point of the first error (1.54 V) to an error rate of 1.3 percent (1.36 V). At 1.52 V, the error rate is approximately one error every 20 seconds—or one error per 1.8 billion multiply operations.

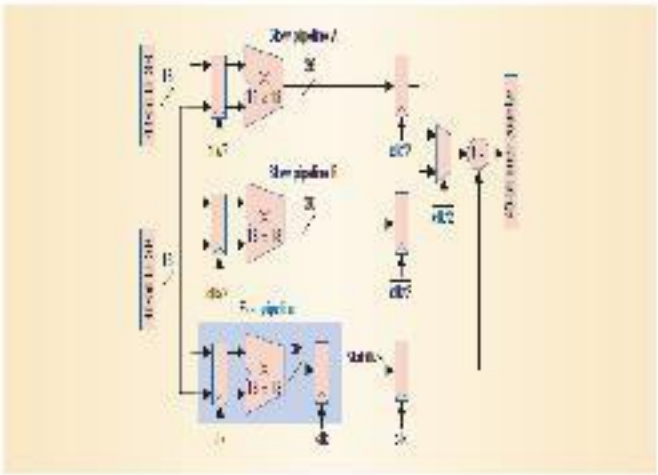

Fig 9. Error-rate test for 18 × 18-bit multiplier block. Shaded area in the test schematic indicates the multiplier circuit under test.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

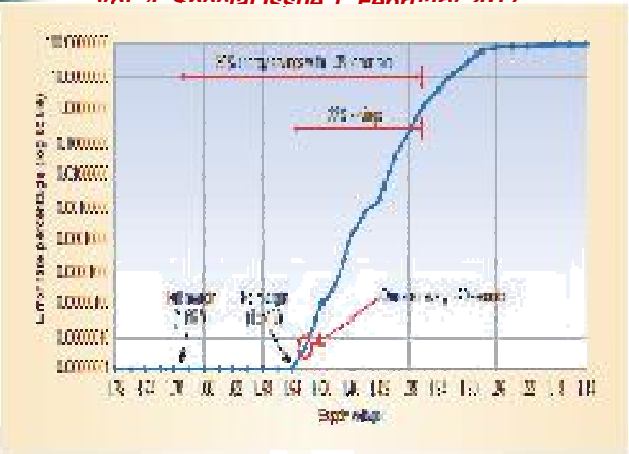

Fig 10. Measured error rates for an 18 × 18-bit FPGA multiplier block at 90 MHz and 27°C.

The gradual rise in error rate is due to the dependence between circuit inputs and evaluation latency. Initially, only circuit inputs that require a complete critical-path reevaluation result in a timing error. As the voltage continues to drop, the number of internal multiplier circuit paths that cannot complete within the clock cycle increases, along with the error rate. Eventually, voltage drops to the point where none of the circuit paths can complete in the clock period, and the error rate reaches 100 percent. Clearly, the worst-case conditions are highly improbable. The circuit under test experienced no errors until voltage has dropped 150 mV (1.54 V) below the full margin voltage. If a processor pipeline can tolerate a small rate of multiplier errors, it can operate with a much lower supply voltage. For instance, at 330 mV below the full margin voltage (1.36 V), the multiplier would complete 98.7 percent of all operations without error, for a total energy savings (excluding error recovery) of 35 percent.

Razor error detection and correction

Razor relies on a combination of architectural and circuit-level techniques for efficient error detection and correction of delay path failures. Fig 11 illustrates the concept for a pipeline stage. A so-called shadow latch, controlled by a delayed clock, augments each flipflop in the design. In a given clock cycle, if the combinational logic, stage L1, meets the setup time for the main flip-flop for the clock's rising edge, then both the main flip-flop and the shadow latch will latch the correct data. In this case, the error signal at the XOR gate's output remains low, leaving the pipeline's operation unaltered. If combinational logic L1 doesn't complete its computation in time, the main flip-flop will latch an incorrect value, while the shadow latch will latch the late-arriving correct value. The error signal would then go high, prompting restoration of the correct value from the shadow latch into the main flip-flop, and the correct value becomes available to stage L2.

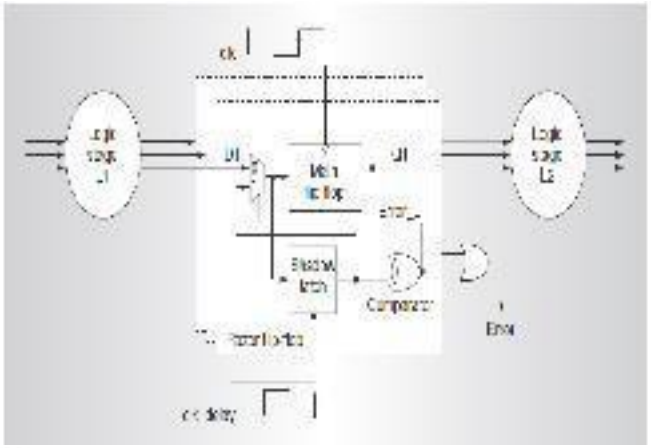

Fig 11: Pipeline stage augmented with razor latches and control lines.

To guarantee that the shadow latch will always latch the input data correctly, designers constrain the allowable operating voltage so that under worst-case conditions the logic delay doesn't exceed the shadow latch's setup time. If an error occurs in stage L1 during a particular clock cycle, the data in L2 during the following clock cycle is incorrect and must be flushed from the pipeline. However, because the shadow latch contains the correct output data from stage L1, the instruction needn't reexecute through this failing stage. Thus, a key Razor feature is that if an instruction fails in a particular pipeline stage, it re-executes through the following pipeline stage while incurring a one-cycle penalty. The proposed approach therefore guarantees a failing instruction's forward progress, which is essential to avoid perpetual failure of an instruction at a particular pipeline stage.

Circuit-level implementation issues

Razor-based DVS requires that the error detection and correction circuitry's delay and power overhead remain minimal during errorfree operation. Otherwise, this circuitry's power overhead would cancel out the power savings from more-aggressive voltage scaling. In addition, it's necessary to minimize error correction overhead to enable efficient operation at moderate error rates. There are several methods to reduce the Razor flip-flop's power and delay overhead, as shown in Fig 12. The multiplexer at the
Razor flip-flop's input causes a significant delay and power overhead; therefore, we moved it to the feedback path of the main flipflop's master latch, as Fig 12 shows.

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

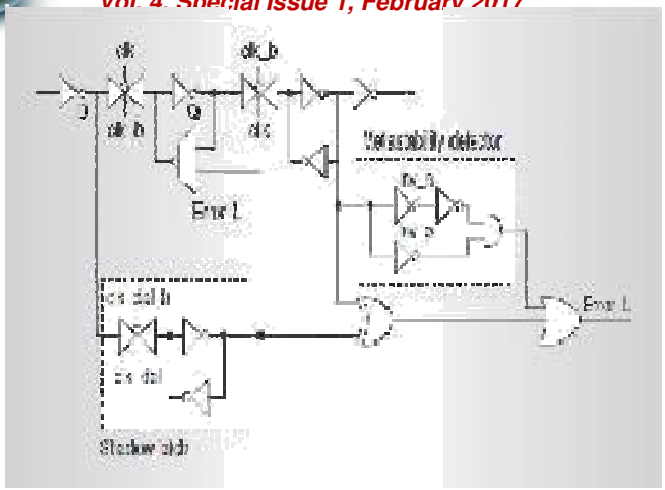

Fig 12: Reduced-overhead Razor flip-flop and metastability detection circuits.

Hence, the Razor flip-flop introduces only a slight increase in the critical path's capacitive loading and has minimal impact on the design's performance and power. In most cycles, a flip-flop's input will not transition, and the circuit will incur only the power overhead from switching the delayed clock, thereby reducing Razor's power overhead. Generating the delayed clock locally reduces its routing capacitance, which further minimizes additional clock power. Simply inverting the main clock will result in a clock delayed by half the clock cycle. Also, many noncritical flip-flops in the design don't need Razor. If the maximum delay at a flip-flop's input is guaranteed to meet the required cycle time under the worst-case subcritical voltage, the flip-flop cannot fail and

doesn't need replacement with a Razor flip-flop. It is found that in the prototype Alpha processor, only 192 flip-flops out of 2,408 required Razor, which significantly reduced the Razor approach's power overhead. For this prototype processor, the total simulated power overhead in error-free operation (owing to Razor flipflops) was less than 1 percent, while the delay overhead was negligible. Using a delayed clock at the shadow latch raises the possibility that a short path in the combinational logic will corrupt the data in the shadow latch. To prevent corruption of the shadow latch data by the next cycle's data, designers add a minimum-path-length constraint at each Razor flip-flop's input. These minimum-path constraints result in the addition of buffers during logic synthesis to slow down fast paths; therefore, they introduce a certain power overhead. The minimum-path constraint is equal to clock delay t_{delay} plus the shadow latch's hold time, t_{hold}. A large clock delay increases the severity of the short-path constraint and therefore increases the power overhead resulting from 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, hence reducing the amount by which designers can drop the supply voltage below the critical supply voltage. In the prototype 64-bit Alpha design, the clock delay was half the clock period. This simplified generation of the delayed clock while continuing to meet the short-path constraints, resulting in a simulated power overhead (because of buffers) of less than 3 percent.

Recovering pipeline state after timing-error detection

A pipeline recovery mechanism guarantees that any timing failures that do occur will not corrupt the register and memory state with an incorrect value. There are two approaches to recovering pipeline state. The first is a simple method based on clock gating, while the second is a more scalable technique based on counterflow pipelining.

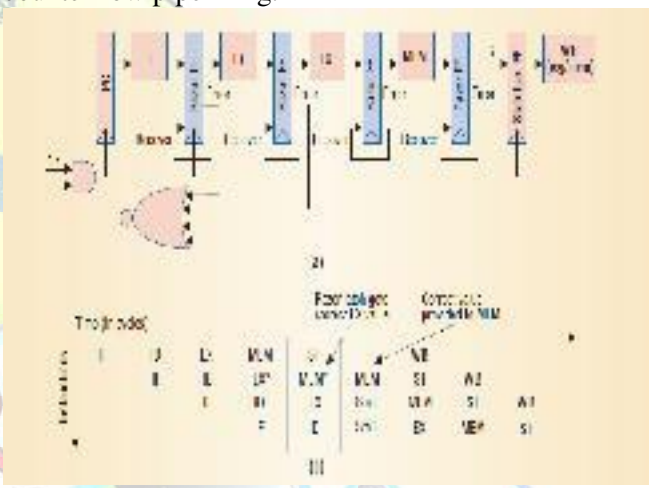

Fig 13: Pipeline recovery using global clock gating. (a) Pipeline organization and (b) pipeline timing for an error occurring in the execute (EX) stage. Asterisks denote a failing stage computation. IF = instruction fetch; ID = instruction decode; MEM = memory; WB = writeback.

Above figure illustrates pipeline recovery using a global clock-gating approach. In the event that any stage detects a timing error, pipeline control logic stalls the entire pipeline for one cycle by gating the next global clock edge. The additional clock period allows every stage to recompute its result using the Razor shadow latch as input. Consequently, recovery logic replaces any previously forwarded errant values with the correct value from the shadow latch. Because all stages reevaluate their result with the Razor shadow latch input, a Razor flip-flop can tolerate any number of errant values in a single cycle and still guarantee forward progress. If all stages ail each cycle, the pipeline will continue to run but at half the normal speed. In aggressively clocked designs, implementing

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 1, February 2017

global clock gating can significantly impact processor cycle time.

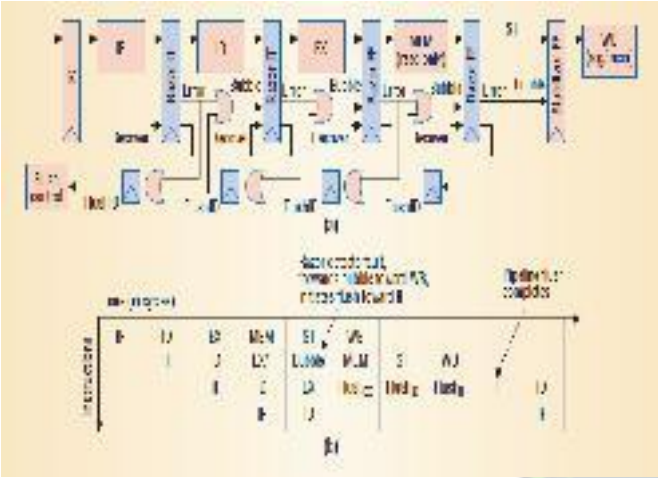

Fig 14: Pipeline recovery using counterflow pipelining. (a) Pipeline organization and (b) pipeline timing for an error occurring in the execute (EX) stage. Asterisks denote a failing stage computation. IF = instruction fetch; ID = instruction decode; MEM = memory; WB = writeback.

Fig 14 illustrates this approach, which places negligible timing constraints on the baseline pipeline design at the expense of extending pipeline recovery over a few cycles. When a Razor flip-flop generates an error signal, pipeline recovery logic must take two specific actions. First, it generates a *bubble* signal to nullify the computation in the following stage. This signal indicates to the next and subsequent stages that the pipeline slot is empty. Second, recovery logic triggers the *flush train* by asserting the ID of the stage generating the error signal. In the following cycle, the Razor flip-flop injects the correct value from the shadow latch data back into the pipeline, allowing the errant instruction to continue with its correct inputs. Additionally, the flush train begins propagating the failing stage's ID in the opposite direction of instructions. At each stage that the active flush train visits, a bubble replaces the pipeline stage. When the flush ID reaches the start of the pipeline, the flush control logic restarts the pipeline at the instruction *following* the failing instruction. In the event that multiple stages generate error signals in the same cycle, all the stages will initiate recovery, but only the failing instruction closest to the end of the pipeline will complete. Later recovery sequences will flush earlier ones.

Conclusion

Power is the next great challenge for computer systems designers, especially those building mobile systems with frugal energy budgets. Technologies like Razor enable "better than worst-case design," opening the door to methodologies that optimize for the common case rather than the worst. Optimizing designs to meet the performance constraints of worst-case operating points requires enormous circuit effort, resulting in tremendous increases in logic complexity and device sizes. It is also power-inefficient because it expends tremendous resources on operating scenarios that seldom occur. Using recomputation to process rare, worstcase scenarios leaves designers free to optimize standard cells or functional units—at both their architectural and circuit levels—for the common case, saving both area and power.

On average, Razor reduced simulated consumption by more than 40 percent,
compared with traditional design-time DVS and delay-chain-based approaches.

References

[1] T. Pering, T. Burd, and R. Brodersen, "The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms," *Proc. Int'l Symp. Low Power Electronics and Design* (ISLPED 98), ACM Press, 1998, pp. 76-81.

[2] T. Mudge, "Power: A First-Class Architectural Design Constraint," *Computer*, vol. 34, no. 4, Apr. 2001, pp. 52-58.

[3] S. Dhar, D. Maksimovic, and B. Kranzen, "Closed-Loop Adaptive Voltage Scaling Controller for Standard-Cell ASICs," *Proc. Int'l Symp. Low Power Electronics and Design*

[4] W. F. Ellis, E. Adler, and H. L. Kalter, "A 2.5-V 16-Mb DRAM in 0.5-mm

[5] CMOS technology", in *IEEE International Symposium on Low PowerElectronics*. ACM/IEEE, Oct. 1994, vol. 1, pp. 88–9.

[6] K. Itoh, K. Sasaki, and Y. Nakagome, "Trends in low-power RAM circuit technologies", in *IEEE International Symposium on Low Power Electronics*. ACM/IEEE, Oct. 1994, vol. 1, pp. 84–7.

[7] Dan Ernst, Shidhartha Das, Seokwoo Lee, David Blaauw, Todd Austin, Trevor Mudge, University of Michigan, Nam Sung Kim," *Razor : circuit-level correction of timing errors for low power operation*", Published by the IEEE Computer Society.

[8] "Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation" in *the 36th Annual International Symposium on Microarchitecture (MICRO-36),*

[9] R. Sivakumar1, D. Jothi1, Department of ECE, RMK Engineering College, India," *Recent Trends in Low Power VLSI Design*", published

in International Journal of Computer and Electrical Engineering.

[10]Christo Ananth, S. Dinesh, "Area power and speed optimized serial type daisy chain memory using modified CPG with SSASPL," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, India, 2015, pp. 344-349,IEEE Data retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7475302&isnumber=7475237

[11]Pillai, P., & Shin, K. G. (Dec. 2001). Real-Time dynamic voltage scaling for low-power embedded operating systems. *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles.*

[12] Christo Ananth, H. Anusuya Baby, "S-Box using AES Technique", International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 3, March – 2014, pp 285-290

[13]K. Flautner et al., "Drowsy Caches: Simple Techniques for Reducing Leakage Power," *Proc. 29th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, 2002, pp. 148-157.

[14]J. Ziegler et al., "IBM Experiments in Soft Fails in Computer Electronics," *IBM J. Research and Development*, Jan. 1996, pp. 3-18.
P. Rubinfeld, "Managing Problems at High Speed," *Computer*, Jan. 1998, pp. 47-48.

[15]T. May and M. Woods, "Alpha-Particle-Induced Soft Errors in Dynamic Memories," *IEEE Trans. Electron Devices*, vol. 26, no. 2, 1979, pp. 2-9.

[16]J. Ziegler, "Terrestrial Cosmic Rays, *IBM J. Research and Development*, Jan. 1996, pp. 19-39.

[17]Weste, N., & Eshraghian, K. (1993). *Principle of CMOS VLSI Design: A System Perspective,* 2nd ed. NewYork: Addison–Wesley.

[18]Roy, K., *et al*. (February 2003). Leakage current mechanisms and leakage reduction techniques in deep