Hindawi Publishing Corporation
Active and Passive Electronic Components
Volume 2012, Article ID 926753, 11 pages
doi:10.1155/2012/926753

Review Article

Recent Subthreshold Design Techniques

Mohsen Radfar, Kriyang Shah, and Jugdutt Singh

Centre for Technology Infusion, La Trobe University, Melbourne, VIC 3086, Australia

Correspondence should be addressed to Mohsen Radfar, m.radfar@latrobe.edu.au

Received 2 March 2012; Revised 30 April 2012; Accepted 5 May 2012

Academic Editor: Yu-Te Liao

Considering the variety of studies that have been reported in low-power designing era, the subthreshold design trend in Very Large Scale Integrated (VLSI) circuits has experienced a significant development in recent years. Growing need for the lowest power consumption has been the primary motivation for increase in research in this area although other goals, such as lowest energy delay production, have also been achieved through sub-threshold design. There are, however, few extensive studies that provide a comprehensive design insight to catch up with the rapid pace and large-scale implementations of sub-threshold digital design methodology. This paper presents a complete review of recent studies in this field and explores all aspects of sub-threshold design methodology. Moreover, near-threshold design and low-power pipelining are also considered to provide a general review of sub-threshold applications. At the end, a discussion about future directions in ultralow-power design is also included.

1. Introduction

Subthreshold digital circuits are now well known to ultralow-power designers and have frequently and successfully been used for applications such as implanted sensors and mobile peripheral processors. However, performance degradation and high sensitivity to Process/Voltage/Temperature (PVT) variations are the primary challenges that have confined subthreshold circuit implementations to low-activity applications.

There are many recent review papers covering the subjects of sub/near-threshold digital design and their challenges. The goal of this review, however, is to cover a boarder range of designs, to show the interrelation of different solutions for low-power digital design and review the recent updates and new advances in ultralow-power era.

Some of useful and comprehensive reviews are as follows. Authors in [1] presented a detailed review that discussed the advantages/disadvantages of subthreshold design, and mathematical equations required for the subthreshold circuits. The paper also covered variation problems briefly and presented subthreshold design techniques for Static Random Access Memories (SRAM) design followed by design techniques for multiprocessing. These discussions make [1] one of the most inclusive early papers.

In [2] authors have focused on variations and leakage reduction, to a greater extent, and besides DC-DC converters have also been discussed.

In one of the recent reviews, authors (see [3]) presented a detailed review of subthreshold design by exploring subthreshold modelling methods, a few leakage reduction techniques, a short discussion about pipelining/parallelising, and a good review of SRAM. A discussion on other device technologies like Double metal-oxide-semiconductor field-effect transistors (D-MOSFETs) has also been included, which will not be included in this review.

Low-power designers constantly need more evidence to consider subthreshold/near-threshold design as a reliable technique to overcome all challenges. This review has collected all important recent works in this area and has considered previous reviews. Considering the above reviews, giving further thorough explanation of subthreshold definitions and formulations has been left out of the scope of this paper. As a result this paper starts from challenges that have come to light since the above-mentioned reviews and will explain the recent advances. Subsequently, this study will compare their

results to find out effectiveness of these solutions and finally will draw attentions to remaining challenges.

Subthreshold challenges are discussed in Section 2. Recent solutions for coping with PVT variations are discussed in Section 2.1 and in two domains of logic and SRAM design. Section 2.2 covers recent solutions for handling performance degradation problem. Section 3 makes a discussion about the papers that were reviewed. Section 4 proposes some future directions coming out from this review, and Section 5 concludes the paper.

2. Subthreshold Challenges

Due to ultralow voltages, subthreshold design has to usually deal with two major challenges namely subthreshold PVT variations and performance degradation. The following subsections will address each of these challenges with the aid of recent studies.

2.1. Challenge 1: PVT Variation

2.1.1. Logic Design. This subsection discusses techniques that either worsen the variations or eliminate them. The possible side effects of these techniques while decreasing variations have also been considered. Also, delay/performance variations due to PVT variations are discussed in this subsection.

Stacking has been widely used to increase the threshold and hence decrease the subthreshold leakage [4]. However, some drawbacks in this technique that cause variations are important and discussed as follows.

Firstly, although stacked devices exhibit lower current variability, they have a higher probability of logic failure due to insufficient output swing especially at lower supply voltages [5].

It is well known that insufficient output swing can be compensated by upsizing the stacked devices and authors in [5] took its advantage to make up for degraded output levels in stacked devices. The failure rate of 0.13% was targeted and the proposed 32-bit adder with constant yield sizing worked for lower voltages until 300 mV [5]. In addition to making up for PVT variations, upsizing also reduces Drain Induced Barrier Lowering (DIBL) effects and as a result leads to lower power consumption [6]. In fact, an upsize by several nanometres at 32 nm node in a Fan Out of 4 (FO4) inverter can reduce energy per operation by 65% at 10 MHz in 0.3 V and by two orders of magnitude at 10 kHz in smaller than 0.2 V [6].

The second problem in stacking is the reduction in current due to stacked devices which results in loss of speed in subthreshold region. This, however, can be offset by body biasing [7]. For example, in [7] authors proposed complementary hybrid latch flip-flop (CHLFF) for ultralow-power applications and Forward Body Bias (FBB) was used to increase the speed of the PMOS (p-channel MOSFET) stacked network. It was found that reducing the supply voltage to 0.3 V in an NMOS (n-channel MOSFET) stacked flip-flop (FF) causes some failures in corners. After applying FBB to PMOS network, the supply voltage could be reduced to 0.23 V, and the speed was increased three times (to 5 MHz) and the power consumed was 0.159 μW. The same idea was also applied to a sense-amplifier-based flip-flop (SAFF) in [7] and the improved Complementary SAFF (CSAFF) worked properly for supply voltages of even less than 0.3 V and consumed 0.144 μW with double the speed (of 5 MHz).

In lower frequencies ($\sim$100 KHz), however, Forward Adaptive Body Bias (ABB) increases minimum-energy due to threshold voltage reduction [8]. Instead, reverse ABB was used in [8] for an 8-bit multiplier and at low frequencies with 0.2 V power supply. Moreover, it was proved to be more efficient (70% less energy overhead) than adaptive supply voltage scaling (AVS). It was also found that global PT variations might suggest a wrong frequency estimation of minimum-energy point. This may result in an improper device/V_{th} selection specific for low-power (LP) or general-purpose (GP) design and lead to energy overhead higher than 200% at the worst-case corner for energy [8].

The papers reviewed so far have examined different techniques using special circuits like Flip-Flops (FF), adders, or multipliers. Following papers, however, have considered more extensive circuits like processors. Their performance is compared in Table 1. For example, the above claims about Body Biasing (BB) can be verified by looking at [9] which uses three BB voltages: forward, zero, and reverse BB produced by a BB Generator with a PV monitor which is an inverter (temperature variations were not studied). BB also was utilised to prevent failure caused by NMOS/PMOS mismatches which in practice modulates the β-ratio adaptively in sub-V_{th} region such that the switching threshold (V_M) will be close to $1/2V_{\text{DD}}$. The inverter V_M is compared against two reference voltages. If $V_M < V_{\text{REF1}}$, signifying that the NMOS is stronger than the PMOS, forward BB will be then applied to the PMOS network. Conversely, if $V_M > V_{\text{REF2}}$, the NMOS network is forward body biased. It was also pointed out that the application of BB can successfully alter the β-ratio and decrease the distribution of V_M, leading to increased robustness in subthreshold circuit [9].

Authors in [10] used the same idea and proposed a configurable V_{th} balancer using BB to reduce the V_{th} mismatch between NMOS and PMOS transistors, so that both the functional and the timing/speed yields were increased. This speed improvement was because both the PMOS and NMOS transistors were forward-biased when the balancer was turned on. The V_{th} balancer, however, applies FBB at Typical-Typical, Fast-Fast, and Slow-Slow corners which accounts for a faster design but FBB could have been cancelled in such corners, especially the Fast-Fast one, to save more energy and to be able to scale voltage even more. Voltage and Temperature variations results are also missing. The logic gates with more than four parallel transistors or four-stacked transistors were discarded from cell library to decrease leakage current variability, and ratioed logics were substituted with nonratioed logics [10].

Although the above-mentioned studies decreased voltage for specific circuits, more discussion is needed on use of Dynamic Voltage Scaling (DVS) in GP designs. DVS is considered inferior to Dynamic Frequency Scaling (DFS) at facing variability in subthreshold voltages when energy

Table 1: Comparison of effect of different techniques for further power-performance improvements.

Criteria	References			
	[9]	[10]	[11]	[12]
Technique	ABB	V_{th} balancing by BB	Frequency scaling	ABB and $W + L$ sizing
Technology	0.13 μm	65 nm	0.13 μm	0.13 μm
Circuit	8×8 FIR Filter	JPEG coprocessor	General-purpose sensor processor	8-bit processor
Frequency	98 KHz at 280 mV 240 Hz at 85 mV	2.5 MHz at 400 mV	833 KHz at 200 mV	77–354 KHz
Performance Impact	2.6x faster at 200 mV 1.25x faster at 1.2 V	Delay improvement from 14 ns to 10 ns	1.09x faster due to library selection	3.6x improvement at 300 mV
Energy/Power Impact	40 nW at 85 mV	0.75 pJ per cycle at 400 mV 1.0 pJ per cycle at 450 mV	2.6 pJ/instruction at 360 mV	3.5 pJ/inst at 350 mV at 354 KHz 515 fJ/inst at 290 mV at 77 KHz
Min voltage (mV)	85 [13]	350	200	140 (24% reduction to zero BB)
Min energy voltage (mV)	85	350	360	350 (total) 290 (core)
Pros	Mismatch prevention	Mismatch prevention, fast	Frequency optimisation	Preventing PT variations, L sizing
Cons	Temp variations not addressed	Not energy conservative, VT variations not addressed	Temp variations and mismatches were not fully investigated	A complicated off-chip BB system

efficiency is the key performance criterion [11]. In fact, it is not limited to the choice of DVS and DFS only. A study in [11] clarifies that many of the area optimal and performance optimal designs, at super threshold voltages, are not suitable for subthreshold voltages (therefore their library was re-characterised for subthreshold operations and, to maximise the robustness, some cells were eliminated from it).

It was also reported in [12] that DVS is more energy-efficient for high target frequencies (i.e., GP designs) while ABB is more energy efficient for low target frequencies (low-power designs) over the frequency range of 30–300 kHz. In fact, [12] verifies [11] for use of DFS and approves that DVS should be substituted in subthreshold design with more energy efficient techniques. Authors in [12] have again used ABB for eliminating performance variations. It was also proved that energy decreases by applying a Reverse BB (RBB) and increases with a FBB for subthreshold circuits.

Recent studies show that one should certainly consider the effect of sizing in addition to BB and stacking. The Body Biasing techniques are usually chip level whilst sizing can be applied at both chip or block level and also along gate width or length. For selection of the most appropriate one (chip/block, Width/Length), [12, 14] investigate some experiments at $V_{DD} = 300$ mV. It was shown that a processor with W sizing (Proc B) and another one with both $W + L$ sizing (Proc C), along critical paths, are 22% and 85% faster, respectively, than a processor with minimum sizing (Proc A). But for Proc C this improvement came at %14 energy penalty with respect to Proc A that could alternatively be achieved by a ~7% energy penalty with increasing V_{DD} by 20–30 mV in Proc A. This suggests that although L sizing is superior to W sizing, it is only appropriate for block-level performance tuning, and not as an entire chip performance variations solution.

Beside sizing, it has to be pointed out that a processor implemented in a 0.18 μm technology is 7.7 times larger than a similar processor in a 65 nm technology, but analysis discloses that total energy is reduced by 647 times [4]. This is a very desirable trade-off, especially when the size of a product is determined by the battery size. Moreover, in [8] authors verified the same idea by stating that minimum energy level is 30% higher in 45 nm technology (at 30 MHz) than in 130 nm technology (at 0.7 MHz).

2.1.2. SRAM Design. Although the mentioned techniques generally decrease the power usage in processors by about 20%, but still SRAM maintains a large proportion of power consumption in chips from 30% in runtime to 90% in standby time [4]. This is because usually more area (more than 50%) is allocated to on-chip caches with every new processor generation. And, That is due to the attractive characteristics of SRAMs such as low activity and high transistor density [15] and also due to power and performance optimisation as a result of placing memory as close as possible to processor.

Low-Power applications like wireless mobiles or sensor processors usually need to have two modes of working, that is, high performance and low-power/standby. The latter mode is usually the source of leakage power consumption, which mostly happens in SRAMs (especially in standby time). It is well known in low-power design that when the activity and voltage reduce, leakage and PVT variations will become the most important factors, as before. This rule is applicable to SRAMs too and this section continues with a review of recent techniques in dealing with these issues. It is worth pointing out that leakage reduction techniques always help tackling PVT variations in SRAM because they usually make SRAMs more robust and error free; therefore

some leakage reduction techniques have also been reviewed. Table 2 compares the most important criteria for SRAM designs.

Again, starting from stacking, authors in [4] further reduce leakage in the bitcell by stacking in the cross-coupled inverters of SRAM as well as other retentive gates. It was found that the leakage sensitivity to number of stacked devices becomes linear for more than two stacked ones. Therefore, a stack height of two was utilised. Moreover, it was shown that length increase of the devices in the cross-coupled inverters leads to a more area-efficient reduction in leakage. It was also observed that IMEM (Instruction MEMory) and DMEM (Data MEMory) consume 89% of the standby power while the CPU consumes only 7% of the power when it is power gated. A particular architecture was proposed for storing frequently used procedures in Instruction Read Only Memory (IROM) while storing application specific instructions in IMEM. Because ROM can be power gated during standby mode, it is beneficial to put as many instructions in IROM as possible. For further leakage reduction, in [4], a specific entry in DMEM is power gated only if a special free-list indicates that the entry is idle.

Apart from gating, sizing and stacking and, as will be seen in following papers, usually various write/read assists are used for preventing subthreshold region failures in SRAMs.

Write/read assists that are designed for subthreshold region, however, might severely impact high-voltage performance [16]. To address this fact, authors in [16] proposed an SRAM with a reconfigurable three different write-assist architecture. By combining different circuits optimized for both subthreshold and superthreshold voltages and employing reconfigurability to switch between them, their SRAM operated from 1.2 V down to 250 mV. Effectiveness of Ultra DVS (UDVS) was also examined in [16]. Consider a memory in low-power mode (0.4 V) and accessed every 2 μs with each access causing active energy consumption. It was observed that leakage power decreased 40x by scaling from 1.2 V (without UDVS) to 0.4 V (with UDVS). But UDVS circuitry consumed energy as well and energy consumption in both with and without UDVS during low-power mode became equal just after five accesses (or 10 μs). As a result, only if a system stays in the low-power mode longer than 10 μs, then it is more beneficial to utilise UDVS, otherwise it will consume more energy.

The work done in [17] (which has already been reviewed in review papers [2, 3]) is similar to [16]. A buffered read was employed to guarantee read stability, and for enabling subthreshold write and read, a peripheral control on both the bit-cell voltage and the read-buffer's foot voltage was performed without degrading the bit-cell's density. Authors also amended the Sense Amplifiers (SAs) and, by means of redundancy, the problem of area-offset tradeoff in SAs was mitigated, which in return decreased read errors by 5x compared to upsizing.

Instead of using the traditional differential structure, authors in [18] used a single-ended cell with a full transmission gate at one side. By the elimination of the second bitline, the cost of having one additional wordline was balanced. One obvious benefit of this design was the ability of the bitline to be driven from rail to rail removing the necessity of sense amplifier (which usually leads to density and variability problems in differential designs). Furthermore, the noise was isolated, during a read operation, to the single bitline which made this design essentially more robust to read failures than differential design. During the write operation, the supply voltage was gated on the feedback inverter to make up for the degraded write margins. Upsizing was also utilised to handle process variations.

Continuing the discussion about read assists, in [19] four operational modes Retention, Read, Write, and a new proposed mode called Accessed Retention mode (AR-mode), for the SRAM cell were defined. This new mode signified those SRAM cells when they were located on an accessed row but they were not selected to be read or written, which is an approach similar to [16, 17]. These cells did not discharge their bitlines, hence saved energy. It also increased the read noise margin of the accessed cell. In addition, it was shown that using RBB in subthreshold region the design led to a low leakage current for all nonselected cells. Furthermore, due to the super-threshold voltage setting for the selected cells (for read operation) the cell access time was reduced dramatically while the stability of the AR-mode cells was maintained.

Moving to write assists, in [15] authors offered a differential 10T bitcell that efficiently split read and write operations and as a result achieved high cell stability. The write assist transistors in the cell were boosted to make up for weak writability. Each four columns were connected to a common ground voltage driver with dynamic-threshold MOS to lessen process variations. The driver's pull-down device was forward-biased during read to increase the drive current.

As it was seen before, in some SRAMs other cells sharing a word line are subject to hold stability problem while a cell is being written [15]. Some solutions, that implement adjacent bits as the same logic word, make the SRAMs exposed to multiple bit-soft errors (which is more critical in subthreshold SRAMs). The offered column-by-column write control in [15] caused the hold stability of adjacent cells not to be affected during a write. Dynamic Differential Cascade Voltage Switch Logic (DCVSL) scheme was also used for read access. In this scheme, bitline leakage noise is offset by the drive current of a keeper, providing large bitline swing. While holding, bitline leakage subthreshold current was considerably decreased because of stacked bitline leakage path.

In [9], however, instead of using read/write assists, a Schmitt-Trigger-(ST-) based 10-transistor SRAM cell was proposed with the idea of making the characteristics of the inverter pair of the bitcell near the ideal inverter which is fundamental for a robust cell operation. The positive feedback from extra transistors adaptively altered the V_M of the inverter depending on the direction of input transition ($0 \rightarrow 1$ input transition or vice versa). The proposed ST bitcell took advantage of differential operation and delivered a better noise immunity.

Although cell design plays an important role in decreasing delay and energy, the SRAM architecture is also another effective part. Multi-tier SRAMs in System On Chip (SOC) design is a common technique to prevent costly out of chip memory accesses as well as to increase performance.

Table 2: Comparison of effects of different techniques in SRAM designing.

Criteria	References						
	[4]	[16]	[17]	[18]	[19]	[15]	[9]
Technique	Stacking and length sizing, power gating in standby, and compression	Buffered read, and reconfigurable UDVS support	Buffered read, control of supply, buffered voltages and SA	Single-ended, 2% bit redundancy, body, header and footer bias	Segmented virtual grounding	Column-wise write, DCVSL read control	Schmitt trigger based
Main novelty	Using ROM and new cell design	Reconfigurability	Redundancy in SAs	New cell design	Super-threshold read	Soft-error addressing	Using ST design
Technology	0.18 μm	65 nm	65 nm	0.13 μm	0.13 μm	90 nm	0.13 μm
Size (bits)	64	64 K	256 K	2 K	40 K	32 K & 49 K	4 K
Frequency (KHz)	~35 at 450 mV 121 at 500 mV	200,000 at 1.2 V 500 at 250 mV	25 at 350 mV	205 at 300 mV 21.5 at 210 mV	100,000 at 400 mV	581.4 at 300 mV 0.5 at 160 mV	620 at 400 mV
Area overhead	910% to 6 T	Not reported	30% to 6 T	42% to 6 T [20]	8% to 6 T	61% to 8 T	~200% to 6 T
Total leakage/size (pA)	Not reported	~700 at 1.2 V ~30.5 at 250 mV	~24 at 350 mV ~21 at 300 mV	~122 at 300 mV	27 at 400 mV	~24.11 at 300 mV	~90 at 400 mV
Energy/access/size (fJ)	~0.000058 at 500 mV	0.167 at 400 mV	~0.396 at 350 mV	0.488 at 340 mV 0.38 at 300 mV	0.17 at 400 mV	0.056 at 300 mV (Write) 0.094 at 300 mV (Read)	50% and 18% lower dynamic and leakage power to 6 T at 175 mV
Min voltage (mV)	450	250	350	193	360	160	160
transistors	14 T	8 T	8 T	6 T	6 T	10 T	10 T
Bit error rate	Not reported	Read static noise margin (SNM) eliminated	Read SNM eliminated	2% at 120 mV	3.5% at 330 mV	60.3 mV mean Read and ~91 mV mean Hold SNM at 300 mV	~56.5 mV mean Read and ~118 mV mean Hold SNM at 400 mV [21]
Min energy voltage (mV)	450	400	350	340	Not reported	160	160
Pros	Low energy	High performance	Low read error rate	Variability aware design	Very high performance	Low energy, high read SNM	Low voltage, high read SNM
Cons	Large area overhead, SNM not discussed	PVT variations not discussed	Low frequency	Still high leakage current	Not DVS enabled	Leakage increase at typical temp	Large area overhead

Besides, the low speed of subthreshold SRAMs limits the ability of subthreshold cores whose speed is usually more than subthreshold SRAMs [22]. As a result, a discussion about optimum subthreshold SRAM architecture is necessary.

Authors in [22] observed that optimal L1 size increases from 64 KB to 128 KB for targets below 76 MHz since L2 starts to relatively consume more energy. Even though a larger L1 causes more energy per access, the energy saved from decreasing L2 accesses (in lower frequencies and because of larger L1) outweighs any increase in the L1. Moreover, it was observed that optimal energy consumption is obtained at near-threshold region (400–500 mV) and at a frequency of ~15 MHz–50 MHz.

Another example is [23] in which by means of a multilevel SRAM, a high-performance design has become possible. The proposed design supported ultra V_{DD} scaling from a nominal to sub/near threshold voltages. In order to reduce off-chip traffic and improve performance and energy efficiency, a large on-chip frame memory (FM) of 10Mbit was embedded, which allowed keeping Video Graphics Array (VGA) frames. However, as discussed before, when dealing with V_{DD} scaling, usual SRAMs cannot work reliably below 700 mV. Therefore, a Hybrid Memory Architecture (HMA) was proposed to decrease the access rate from processors to the FM by employing the data locality in the scratchpad memory (SM). Within the proposed HMA, there existed three characterized memories to hold the data: (1) ACCU register: short-term data; (2) SM: intermediate-term data; (3) FM: long-term data.

On the other hand, near-threshold operation in logic decreases frequency compared to super-threshold one. This speed degradation, however, suggests several new and interesting design opportunities about memory system selection [24]. Firstly, memory technologies (like 130 nm or 180 nm devices as discussed in Logic Design Section 2.1.1) and designs that are slower and more energy efficient can substitute timing critical memory designs. This will help to decrease the total energy of the chip while memory is working in super-threshold voltage and logic in near-threshold. Furthermore, multiple accesses to memory can be carried out in one near-threshold clock cycle of logic. This means that more parallel data can be fetched and be processed in one cycle. And finally, the slower memory possibility allows caches, register files, and other elements that are originally designed to compensate long memory latency, to be turned off or removed. Therefore, a pipeline was implemented so that in a single cycle of the Single Instruction Multiple Data (SIMD) pipeline, multiple memory access was possible. It was also showed that wider SIMD widths do not always provide less energy consumption because of the additional hardware and increase of critical path delay.

2.2. Challenge 2: Performance Degradation. As stated in introduction, performance decrease is another challenge in subthreshold circuits which is usually addressed by whether parallelism or pipelining. As it will be seen later, pipelining is popular in super-threshold designs because it usually needs many circuitries for controlling the pipeline which when operated in subthreshold voltages will result in a leaky system. As a result, near-threshold voltage operation was considered for pipelined circuits by many researchers. In the super-threshold region, energy is extremely sensitive to V_{DD} due to the quadratic dependence of active energy on V_{DD}. Therefore, voltage scaling down to the near-threshold voltages yielded 10x energy reduction at the expense of nearly 10x performance decrease [25]. Interestingly, energy reduces by only ~2x when V_{DD} is scaled down from the near-threshold region to the subthreshold region, but at the same time delay rises dramatically by 50–100x. As a result, authors in [25] concluded that huge amount of performance could be recovered by just backing off a bit and working in near-threshold region.

More about near-threshold design: authors in [26] also found that the rate of delay change with respect to supply voltage change ($\delta t_d/\delta V$) is very huge in near-threshold regime. A 200 mV change in supply voltage from 0.3 V to 0.5 V leads to approximately 30x change in performance. The concept was proved by offering a two V_{DD} design that were only 50 mV apart and as suggested a small voltage supply rise caused very considerable speedup. Dual-V_{DD} assignment was applied at the level of entire rows in the layout in order to restrict the surplus cost of dual voltage distribution and no level shifters were utilised because these dual voltages were not more than 100 mV apart. It was showed that the maximum speed-up (with V_{DDL} = 0.4 V and V_{DDH} = 0.45 V) was ~45%, which was equal to what is obtained by powering up all cells to the V_{DDH}.

Above studies demonstrated that near-threshold voltages are necessary for higher speed demands. Keeping this fact in mind, this section continues with articles that have tried to increase performance more by parallelism and pipelining within sub/near-threshold region.

2.2.1. Parallelising. One good example of parallelism has been illustrated in JPEG cores in [10]. With sub/near threshold techniques explained in Logic Design Section 2.1.1, it became best suitable for low-energy and medium-frequency applications, such as mobile image processing.

A comprehensive study of parallelism can be found in [22] which studies all the main factors influencing the energy of a system such as size of L1 cache and cluster, number of clusters and V_{DD} and V_{th} selection within a cluster. Firstly, memory tends to operate at speeds faster than the core at subthreshold region, and therefore the idea of using more than one cluster that share one memory was proposed. It was shown that the energy optimal point is 2 cores per cluster with 2 clusters. This point brought about a 53% increase in energy efficiency compared to traditional multiprocessor designs. For targets above 150 MHz, the optimal number of clusters increased from 2 to 3. Interestingly, a design with increased number of clusters, and therefore total cores, required less energy than a design with a scaled voltage of the smaller number of cores and with same constrains.

2.2.2. Pipelining. Pipelining in subthreshold region leads to leaky circuits, as discussed before, and as a result, if still a subthreshold pipeline is necessary, it has to be very simple.

Table 3: Comparison of pipelining strategies.

Criteria	References				
	[27]	[28]	[29]	[30]	[31]
Technique	Instruction isolation	DVFS and critical path isolation under temperature variations	Variable clock in times of process variations	Soft edge flip-flop	Flow-through latch between stages and selection of different voltages
Technology	45 nm	BPTM 70 nm	90 nm	PTM 65 nm	PTM 32 nm
Circuit	32-bit in-order 5-stage dual-pipeline processor with IA32	in-order superscalar pipeline with the Alpha ISA	32-bit microprocessor	34-bit pipelined adder	6 stages pipelined FPU
Frequency	1.25 GHz	1.5–3 GHz	~0.1–1 GHz	2–2.5 GHz	Improves BIPS/W by 47% (actual frequency not reported)
Area and/or frequency impact	28% performance reduction due to instruction isolation	~4.5% area overhead /3.4–11% frequency overhead	2.6% area overhead/13%–50% performance improvement	5–20% performance improvement	40% performance improvement
Energy/Temperature impact	13% power reduction	Reduces temperature by 6.6–9%	3% energy overhead	19% power saving (4.9 mW)	Not reported
Min voltage	740 mV for ADD 680 mV for XOR and AND	700 mV	Scaling from 1.2 V to 1 V	Scaling from 1.2 V to 1.05 V (5–20% V_{DD} reduction)	Scaling from 1.4 V to 0.95 V
Pros	DVS enabled	Temperature variations tolerant, DVFS enabled	Low energy and area overhead	Rather large power reduction	Rather large performance improvement
Cons	Performance reduction	PV variations not discussed	Not supporting very low voltages	Not supporting very low voltages, PVT variations not discussed	Not supporting very low voltages

For example in [32] considering subthreshold operation difficulties like PVT variations, it was concluded that variations are distributed over the length of a path which makes shallow pipelines with high-FO4 delay per stage more advantageous. Hence, in [11] a 2 stage pipeline implementation was selected for the processor.

Because of above study and as review of different literature shows, discussion about low-power pipelining is more logical in near-threshold than subthreshold era and the rest of this subsection focuses on near-threshold region.

Generally, there are three main sources of energy consumption in pipeline: instructions, circuits (including datapath, registers, and control), and synchronisation plan. As it will be explained in detail, each instruction has its own specific energy usage. And depending on synchronisation strategy running between stages, active energy differs. While Section 2.1.1 covered data path and control circuits, some specific concerns are discussed in this subsection. Moreover, as some techniques useful for near-threshold pipelining (such as how to cope with PVT variations in pipeline) were located in low power super-threshold researches, inevitably these studies have also been included (see Table 3).

Starting from instruction, the first discussion will be about isolation. It has been reported in [27] that as supply voltage reduces, ADD instruction operates correctly until 0.74 V, while logical instructions (XOR and AND) tolerate V_{DD} scaling down to 0.68 V. Therefore it was proposed that isolating ADD operation lets the ALU operate at 0.68 V by providing 2-cycles for ADD operation and 1-cycle for other instructions, resulting in more power savings. For ADD, the ALU saved another 23% power because halving the frequency at the reduced V_{DD} also decreased power consumption at the cost of performance degradation.

Authors in [28] have considered both instructions and datapath. Any possible delay failure in specific instructions such as ADD (under process variation and voltage scaling) was prevented by adaptively extending the clock period to two-cycles while all standard operations were single cycle. Execution datapath was changed so that whenever a failure in operations became probable, those operations could be executed in two cycles. The prediction was done by utilising a small predecoding logic. In addition, if the temperature exceeded a threshold value, a lower supply voltage (V_{DDL}) was applied to the execution unit. Once the temperature fell

below the threshold, nominal supply (V_{DDH}) was reapplied. It is also interesting that during execution, only EX stage received V_{DDL} while all other pipeline stages received V_{DDH}.

In synchronisation, it is often attempted to modify the clock so that the slack time in datapath is used for compensation of variations or voltage scaling. For example in [29] by considering instruction, datapath and clocking, authors associated a variable delay with each pipeline stage, and a table of delays were adjusted to meet the delay of each specific instruction. Whenever the delays of all stages were elapsed, a new clock was created and therefore, some clocks were shortened and the overall speed was increased. This variable delay unit was located close to the corresponding datapath to be subject to the similar PVT conditions. A delay selector reads the inputs of the pipeline to choose appropriate delay value from operation selection table.

In [30], with focusing on synchronisation plan, a new soft edge flip-flop (SEFF) was proposed to postpone the clock of the master latch to produce a window along which both master and slave latches were active. This window, which is called the transparency window, allowed timing slacks to pass between adjacent pipeline stages. The delayed clock was created by employing an inverter chain and sizing them in order to maintain the desired delay. Available slacks at stages were passed to the previous stages, providing previous stages with surplus of borrowed time. Since positive slacks were available in all stages of the pipeline, as a result of this time borrowing, the clock could be increased or circuit voltage could be decreased to reduce the power consumption.

In [31], a design was presented with a flow-through latch between two stages so that clocking of that latch added an extra half cycle to the pipeline. This half cycle provided extra time borrowing to absorb delays due to process variation in the previous stages. Gating the latch and switching between the modes with and without the extra latency allow for postfabrication tuning. A voltage interpolation was also used to deliberately select different effective voltages needed for each stage to run at a single nominal frequency. Therefore, if pipeline ran slowly due to process variation, there were two ways to obtain the nominal operating frequency. One option was connecting more stages to V_{DDH} so that the effective voltage increased. Another option as discussed before was extending two stages with a latch in between to a single stage to provide additional time for execution while decreasing energy by switching more stages to V_{DDL}.

Considering the effect of clock in power consumption, as another synchronising plan, it should be pointed out that asynchronous pipelines are playing a vital role in recent designs. Although they are beyond the scope of this review, taking the idea of eliminating the clock in synchronised world, one can find [33] with Moebius pipeline proposed. In this pipeline, a stage sent a COMPLETE signal to the previous stage when the computation was done and at the same time held the result until the COMPLETE signal from the next stage came. This way in addition of saving clock power, the available slack in the path was utilised very efficiently and moreover variations were dealt with in a better way.

This subsection is concluded with a survey in Pipelining [34]. This survey states that the single-issue in-order architecture, as also the case for above studies, is appropriate for very low-energy design points, while the quad-issue out-of-order is only suitable at very high-performance applications [34]. It was also discovered that the dual-issue in-order and out-of-order processors were efficient for many different kind of design performances.

3. Discussion

In Section 2.1.1, the major problems of stacking, a technique which is used for decreasing leakage, were addressed by upsizing [5] and BB [7]. BB was also utilised by [9, 10, 12] for dealing with variations. In [12] also BB was preferred over DVS for lower frequencies. DVS was also inferior to DFS as reported by [11]. Moreover, L sizing was proved in [14] to be more power/performance efficient than W sizing but unlike BB which was employed at whole chip level, $L + W$ sizing was only useful at block level. Finally using older technologies for subthreshold design was encouraged by [4, 8] which results in a huge energy saving.

In SRAM Design Section 2.1.2, again stacking was used by [4] for leakage reduction and effects of upsizing and gating were studied for decreasing leakage.

Different read/write assists methods were discussed like in [16] reconfigurable write assist, supported UDVS from super- to subthreshold voltages. For increasing read stability, new read-buffer techniques were used in [16, 17, 19]. Read problem was addressed by a novel single-ended cell design in [18] and the SA with common problems of density and variability were also eliminated.

Weak writability was made up for in [15] by a new write assist technique and multiple-bit soft error in subthreshold voltages was considered. In [9], ST based design removed the read/write assist necessity.

Power/performance optimum cache sizes and hierarchical SRAM designs were discussed in [22, 23], respectively. Finally the benefit of old technologies together with near-threshold voltage for SRAM design was again emphasised in [24].

Moving to performance degradation Section 2.2, authors in [25] suggested near-threshold voltages for better performance results while benefiting from its low power advantages. Avoiding costly level shifters, in [26] near-threshold voltage gained the same speed-up as super-threshold voltages. Seeking for more performance, authors in [10] used parallel cores with sub/near-threshold voltages. Looking for power/performance optimum number of cores, in [22] 2 cores per cluster with 2 clusters for targets below 150 MHz were recommended.

In processors world, however, pipelining is crucial but in [32] it was pointed out that subthreshold design limits pipelines to 2 stages when variations are considered. Therefore, near-threshold design was again highlighted.

Some techniques for low-power pipelining, useful for near-threshold voltages, then were necessary. In [27] more power-consuming instructions were isolated which helped to save more energy. The same idea was used in [28] which

prevented the being executed instructions from failure in times of variations.

Exploiting time slacks in stages was the most important technique for handling variations in pipelines. In [29], for example, different delays due to differences between instructions was utilised and as it will be seen, many kinds of FFs have been designed to benefit from time slacks. In [31] also time borrowing was done using a latch inserted inside a stage which also helped in postfabrication tuning.

Finally in [34] it was acknowledged that the single-issue in-order architecture is appropriate for very low-power design goals.

This suggests that complicated instructions and architectures are not appropriate for low-power pipelines as they lead to a leaky and error-prone structure.

It can be concluded from Table 1 that power consumption can be reduced by using Body Biasing for compensating the variations and mismatch between V_{th} of pullup and pull-down network. In SRAM designs using reconfigurability and employing different read/write assists to isolate nonaccessed cells are important factors for both speed and power in accord to Table 2. Table 3 also showed that, for optimising power and speed, using synchronisation strategies is as important as instruction isolation.

This review also presents future directions in next section but discussion about time borrowing is necessary before proceeding (readers are referred to [25, 35] for a complete understanding of error detection techniques in pipelines). First of all it should be noted that, under process variations, the SEFF delay should be changed. It means a technique should evaluate variation and apply different postsilicon SEFF delays so that variations are compensated. A system which can calculate variations should be designed and integrated. Razor [36] is such a technique that has already been used. However, Razor method might pick up the wrong frequency. For example, if an AND is followed by an ADD, as ADD is more prone to variations and tends to cause error, Razor detects this error and increases clock period. However, the subsequent instruction, which is an AND, does not need this frequency reduction.

One way to find out if an FF has caught true data, as Razor does, is comparing the data with a delayed clock data. Another way, however, is calculating if an incoming data has violated the setup and hold times of FF and, based on that, latching an erroneous data, as done in [37]. This idea is actually similar to Razor II [38] and has the same problem as Razor. Moreover, Razor also needs a minimum short path delay because when a clock is triggered, shadow latch at Razor will be waiting for a late coming signal, by means of a delayed clock, but at same time the short path results may change the shadow latch data, before critical path of previous clock discloses its data. Furthermore, there is a probability that metastability propagates through the error detection logic and causes metastability of the restore signal itself, which has been addressed in next versions by adding more circuitry like [39]. In [35] (which is an advanced form of [37]) a new FF is proposed that can handle both short and critical path errors and moreover the FF can recover critical path errors, like Razor, and also can predict short path errors. But this technique incurs a large area and is suitable for super-threshold voltages and it still has the same problem as discussed before.

Another problem of these in situ monitors is their activity, area and energy overhead. Authors in [40] use the high clock phase as the error-detection window for the short path problem, where minimum delay paths must not arrive before the falling clock edge. Latch transparency feature was also taken advantage of and by above assumption, the extra master latch of FF was eliminated and energy was reduced and metastability was tackled using transparency.

Using instruction isolation and due to instruction specific delays, however, the rate of violation resulting from different instructions delays will be cancelled and just those violations caused by temperature will emerge. Authors in [27] present a comprehensive research by putting all these idea together along with using error correction (the details of error correction scheme was not published however) for its pipeline and LUT for keeping the delay of different instructions. The next section will present the future directions in near-threshold and subthreshold design techniques.

4. Future Directions

The requirement of different delays by different instructions means that a transparency window is needed whose size can be changed by different instructions. Employing SEFFs for this purpose is a unique technique that to our best knowledge has never been used so far. By designing this new FF, a transition detector can detect long path delays during transparency window and therefore setting an error signal for tuning the size of window. Moreover, different instructions also lead to different delays being applied which as a result decrease the rate of errors.

Like the approach in [27], an instruction delay can be predicted, with a Look-Up Table (LUT), and be applied to SEFF and if timing is violated because of error detection, LUT entry should be updated with an increased delay. This LUT can be implemented in a ROM as it keeps data permanently after once filled. In another technique, if just process variations are of concern, error detection can be eliminated by just postsilicon evaluations and this way error detection circuitry can be clock gated or totally discarded by an offchip error detection scheme.

Another opportunity of performance improvement can be created by employing instruction isolation using an extra transparent FF (TFF) in EX stage of pipeline, so that long delay instructions can use two stages and short delays use one stage (See Figure 1) and even more stages are applicable. Depending on situation, these TFFs can be transparent or functional. This way it is not necessary to stall the pipeline for energy consuming instructions to be completed by two cycles (or more). Also, a short instruction can be completed quickly, and depending on other instructions on pipeline, they can go through the transparent FFs without clocking (which again saves energy). Having said that, the EX stage design should be changed so that long instructions are split to two (or more) parts in order to implement each part in

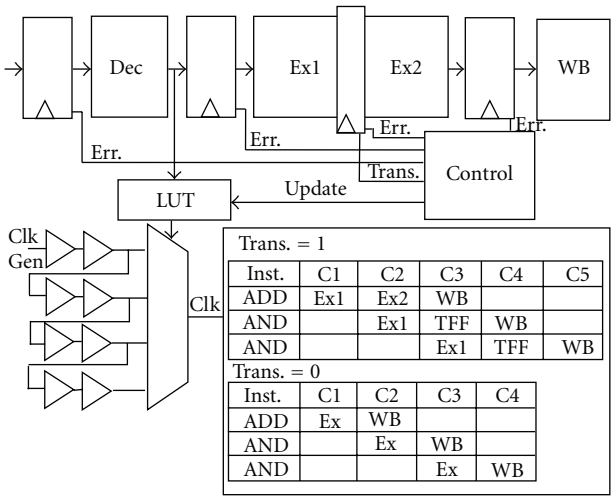

Figure 1: Proposed architecture with a sample instruction flow. When Trans. = 1, pipeline is working in LP mode, otherwise the high-speed mode is applied.

one stage of pipeline with almost the same delay. Moreover, for taking advantage of DVS, the FFs can be one of those with error correction and time borrowing features [40]. When the pipeline needs more speed and can handle higher voltages, the TFF becomes transparent and pipeline can work with higher frequencies. And once pipeline needs to consume less power and lower frequencies can be tolerated, by activating TFF, DVS helps the pipeline to work with the lowest voltage possible. When the optimum voltage is applied, the error detection scheme helps the pipeline with DFS to handle PVT variations. LUT can also be used to keep record of appropriate delays for different instructions.

5. Conclusion

Looking at different aspects of low-power design, one can immediately find out that, apart from emerging new device technologies, it is impossible to maintain such a design without approaching to near threshold/subthreshold regions. However, these regions bring about many issues that researchers have been engaged in for recent years. These problems include leakage increase, PVT vulnerability and performance degradation. This paper tried to present a complete review of recent attempts for solving these problems and compared them to find out which ones have been more effective. This work also proposes future directions based on the outcome of the review and currently the paper authors are involved in implementation of these proposed directions.

References

[1] S. Hanson, B. Zhai, K. Bernstein et al., "Ultralow-voltage minimum-energy CMOS," *IBM Journal of Research and Development*, vol. 50, no. 4-5, pp. 469–490, 2006.

[2] J. Kwong and A. P. Chandrakasan, "Advances in ultra-low-voltage design," *IEEE Solid-State Circuits Newsletter*, vol. 13, pp. 20–27, 2008.

[3] S. K. Gupta, A. Raychowdhury, and K. Roy, "Digital computation in subthreshold region for ultralow-power operation: a device-circuit-architecture codesign perspective," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 160–190, 2010.

[4] S. Hanson, M. Seok, Y. S. Lin et al., "A low-voltage processor for sensing applications with picowatt standby mode," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1145–1155, 2009.

[5] J. Kwong and A. P. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proceedings of the 2006 International Symposium on Low Power Electronics and Design (ISLPED '06)*, pp. 8–13, New York, NY, USA, October 2006.

[6] D. Bol, R. Ambroise, D. Flandre, and J. D. Legat, "Interests and limitations of technology scaling for subthreshold logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 10, pp. 1508–1519, 2009.

[7] F. Moradi, D. T. Wisland, H. Mahmoodi, A. Peiravi, S. Aunet, and T. V. Cao, "New subthreshold concepts in 65 nm CMOS technology," in *Proceedings of the 10th International Symposium on Quality Electronic Design (ISQED '09)*, pp. 162–166, San Diego, Calif, USA, March 2009.

[8] D. Bol, D. Flandre, and J. D. Legat, "Technology flavor selection and adaptive techniques for timing-constrained 45 nm subthreshold circuits," in *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 21–26, New York, NY, USA, August 2009.

[9] K. Roy, J. P. Kulkarni, and M. E. Hwang, "Process-tolerant ultralow voltage digital subthreshold design," in *Proceedings of IEEE Topical Meeting on Silicon Monolithic Integrated Circuits in RF Systems (SiRF '08)*, pp. 42–45, Orlando, Fla, USA, January 2008.

[10] Y. Pu, J. P. De Gyvez, H. Corporaal, and Y. Ha, "An ultra-low-energy multi-standard JPEG Co-processor in 65 nm CMOS with sub/near threshold upply voltage," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 3, pp. 668–680, 2010.

[11] B. Zhai, S. Pant, L. Nazhandali et al., "Energy-efficient subthreshold processor design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 8, pp. 1127–1137, 2009.

[12] S. Hanson, B. Zhai, M. Seok et al., "Exploring variability and performance in a sub-200-mV processor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 4, pp. 881–890, 2008.

[13] M. E. Hwang and K. Roy, "ABRM: adaptive β-ratio modulation for process-tolerant ultradynamic voltage scaling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 2, pp. 281–290, 2010.

[14] S. Hanson, B. Zhai, M. Seok et al., "Performance and variability optimization strategies in a sub-200 mV, 3.5 pJ/inst, 11 nW subthreshold processor," in *Proceedings of IEEE Symposium on VLSI Circuits*, pp. 152–153, Kyoto, Japan, June 2007.

[15] I. J. Chang, J. J. Kim, S. P. Park, and K. Roy, "A 32 kb 10T sub-threshold sram array with bit-interleaving and differential read scheme in 90 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 2, pp. 650–658, 2009.

[16] M. E. Sinangil, N. Verma, and A. P. Chandrakasan, "A reconfigurable 8T ultra-dynamic voltage scalable (U-DVS) SRAM in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 11, pp. 3163–3173, 2009.

[17] N. Verma and A. P. Chandrakasan, "A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 141–149, 2008.

[18] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester, "A variation-tolerant sub-200 mV 6-T subthreshold SRAM," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 10, pp. 2338–2348, 2008.

[19] M. Sharifkhani and M. Sachdev, "An energy efficient 40 Kb SRAM module with extended read/write noise margin in 0.13 μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 2, pp. 620–630, 2009.

[20] B. Zhai, D. Blaauw, D. Sylvester, and S. Hanson, "A sub-200 mV 6T SRAM in 0.13 μm CMOS," in *Proceedings of the 54th IEEE International Solid-State Circuits Conference (ISSCC '07)*, pp. 332–606, San Francisco, Calif, USA, February 2007.

[21] J. P. Kulkarni, K. Kim, and K. Roy, "A 160 mV, fully differential, robust schmitt trigger based sub-threshold SRAM," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '07)*, pp. 171–176, New York, NY, USA, August 2007.

[22] B. Zhai, R. G. Dreslinski, D. Blaauw, T. Mudge, and D. Sylvester, "Energy efficient near-threshold chip multi-processing," in *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED '07)*, pp. 32–37, Portland, Ore, USA, August 2007.

[23] Y. He, Y. Pu, Z. Ye et al., "Xetal-pro: an ultra-low energy and high throughput SIMD processor," in *Proceedings of the 47th Design Automation Conference (DAC '10)*, pp. 543–548, Anaheim, Calif, USA, June 2010.

[24] S. Seo, R. G. Dreslinski, M. Who, C. Chakrabarti, S. Mahlke, and T. Mudge, "Diet SODA: a power-efficient processor for digital cameras," in *Proceedings of the 16th ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED '10)*, pp. 79–84, Redondo Beach, Calif, USA, August 2010.

[25] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[26] M. R. Kakoee, A. Sathanur, A. Pullini, J. Huisken, and L. Benini, "Automatic synthesis of near-threshold circuits with fine-grained performance tunability," in *Proceedings of the 16th ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED '10)*, pp. 401–406, Austin, Tex, USA, August 2010.

[27] S. E. Lee, C. Wilkerson, M. Zhang, R. Yavatkar, S. L. Lu, and N. Bagherzadeh, "Low power adaptive pipeline based on instruction isolation," in *Proceedings of the 10th International Symposium on Quality Electronic Design (ISQED '09)*, pp. 788–793, March 2009.

[28] S. Ghosh, J. H. Choi, P. Ndai, and K. Roy, "O2C: occasional two-cycle operations for dynamic thermal management in high performance in-order microprocessors," in *Proceedings of the 13th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '08)*, pp. 189–192, Bangalore, India, August 2008.

[29] N. Toosizadeh, S. G. Zaky, and J. Zhu, "Varipipe: low-overhead variable-clock synchronous pipelines," in *Proceedings of IEEE International Conference on Computer Design, ICCD 2009*, pp. 117–124, October 2009.

[30] M. Ghasemazar, B. Amelifard, and M. Pedram, "A mathematical solution to power optimal pipeline design by utilizing soft edge flip-flops," in *Proceedings of the 13th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '08)*, pp. 33–38, Bangalore, India, August 2008.

[31] X. Liang, G. Y. Wei, and D. Brooks, "ReVIVaL: a variation-tolerant architecture using voltage interpolation and variable latency," in *Proceedings of the 35th International Symposium on Computer Architecture (ISCA '08)*, pp. 191–202, June 2008.

[32] B. Zhai, L. Nazhandali, J. Olson et al., "A 2.60 pJ/inst subthreshold sensor processor for optimal energy efficiency," in *Proceedings of the IEEE Symposium on VLSI Circuits, Digest of Technical Papers (VLSIC '06)*, pp. 154–155, June 2006.

[33] M. G. Jeong, T. Nakura, M. Ikeda, and K. Asada, "Moebius circuit: dual-rail dynamic logic for logic gate level pipeline with error gate search feature," in *Proceedings of the 19th ACM Great Lakes Symposium on VLSI (GLSVLSI '09)*, pp. 177–180, Boston, Mass, USA, May 2009.

[34] O. Azizi, A. Mahesri, B. C. Lee, S. J. Patel, and M. Horowitz, "Energy-performance tradeoffs in processor architecture and circuit design: a marginal cost analysis," in *Proceedings of the 37th International Symposium on Computer Architecture (ISCA '10)*, pp. 26–36, Saint-Malo, France, June 2010.

[35] K. Hirose, Y. Manzawa, M. Goshima, and S. Sakai, "Delay-compensation flip-flop with in-situ error monitoring for low-power and timing-error-tolerant circuit design," *Japanese Journal of Applied Physics*, vol. 47, no. 4, pp. 2779–2787, 2008.

[36] D. Ernst, K. Nam Sung, S. Das et al., "Razor: a low-power pipeline based on circuit-level timing speculation," in *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-36*, pp. 7–18, 2003.

[37] M. J. Turnquist and L. Koskinen, "Sub-threshold operation of a timing error detection latch," in *Proceedings of the Research in Microelectronics and Electronics, 2009. PRIME 2009. Ph.D.*, pp. 124–127, July 2009.

[38] D. Blaauw, S. Kalaiselvan, K. Lai et al., "Razor II: in situ error detection and correction for PVT and SER tolerance," in *Proceedings of the IEEE International Solid State Circuits Conference (ISSCC '08)*, pp. 399–622, February 2008.

[39] S. Das, S. Pant, D. Roberts et al., "A self-tuning DVS processor using delay-error detection and correction," in *Proceedings of the Symposium on VLSI Circuits, 2005, Digest of Technical Papers*, pp. 258–261, June 2005.

[40] K. A. Bowman, J. W. Tschanz, N. S. Kim et al., "Energy-efficient and metastability-immune timing-error detection and instruction-replay-based recovery circuits for dynamic-variation tolerance," in *Proceedings of the IEEE International Solid State Circuits Conference (ISSCC '08). Digest of Technical Papers*, pp. 402–623, February 2008.