Scratchpad Memory Optimizations for Digital Signal Processing Applications

Syed Z. Gilani
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
gilani@wisc.edu

Nam Sung Kim
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
nskim3@wisc.edu

Michael Schulte
Research and Advanced Development Labs
Advanced Micro Devices
michael.schulte@amd.com

***Abstract*—Modern digital signal processors (DSPs) need to support a diverse array of applications ranging from digital filters to video decoding. Many of these applications have drastically different precision and on-chip memory requirements. Moreover, DSPs often employ aggressive dynamic voltage and frequency scaling (DVFS) techniques to minimize power consumption. However, at reduced voltages, process variations can significantly increase the failure rate of on-chip SRAMs designed with small transistors to achieve high integration density, resulting in low yields. Consequently, the size of transistors in SRAM cells and cell size needs to be increased to satisfy the target yield. However, this can result in high area overhead since on-chip memories consume a significant portion of the die area.**

In this paper, we present a scratchpad memory design that exploits the tradeoffs between SRAM cell sizes, their failure rates, the minimum operating voltage for target yield (V_{ddmin}), and application characteristics to achieve an on-chip memory area reduction of up to 17%. Our approach reduces V_{ddmin}, which allows dynamic and leakage power savings of 42% and 36% respectively with DVFS. Moreover, for error-tolerant DSP applications we allow voltage scaling below V_{ddmin} to achieve further power savings while incurring lower mean error as compared to short word-length memory. Finally, for error-sensitive applications, we propose a reconfigurable memory organization that trades memory capacity for higher precision at a lower V_{ddmin}.

I. Introduction

Many applications in the signal processing domain have real-time constraints and operate in low-power environments. Digital signal processors (DSPs) must provide deterministic execution latencies for these applications and minimize the data transfers to and from main memory. Consequently, most DSPs prefer software-controlled scratchpad memories, where the software explicitly specifies data transfers and placement in the memory, to data caches. Since the data required by DSP applications during computations usually can be determined beforehand, scratchpad memories can significantly improve the energy requirements of the DSP by eliminating unnecessary data transfers.

DSP applications usually exhibit a higher error tolerance than general-purpose applications [1]. Consequently, many DSPs allow reduced precision arithmetic to decrease power consumption. In terms of memory design, this error tolerance can be exploited to decrease the memory area by reducing the memory word length. However, reducing memory word length introduces a hard constraint on the precision of DSP applications. Applications that require a higher precision may need to perform multiple memory accesses and pack and unpack the data in software. All these considerations make word length reductions in memories less suitable for processors intended for a wide range of applications.

Unlike general-purpose computing, DSP applications are usually statically scheduled and extensively profiled to determine their characteristics, such as performance and power consumption. This allows more opportunities for aggressive DVFS in DSPs. However, at reduced voltages, process variations in on-chip memories can considerably increase the failure rate of SRAM cells implemented with small-size transistors. Consequently, the transistors in SRAM cells need to be sized up to make them more robust to withstand process variations and improve memory yield when DVFS is employed for low-voltage operation.

Since on-chip memories constitute a significant portion of die area (60% for the StrongARM®processor [2]), any increase in the cell size due to adopting larger transistors in each cell can greatly impact the overall die area. If the SRAM cells are not sized up, the minimum operating voltage (V_{ddmin}) of the processor is limited by the failure rate of SRAM cells. Many circuit- and logic-level techniques have been proposed to reduce the failure rate of SRAM cells for a given size [3]: However, even with such techniques, the overall failure rate (and thus the yield) of on-chip memories is dominated by their cell size.

In this paper, we explore the trade-offs between SRAM cell sizes, their failure rates, V_{ddmin}, and the precision requirements of applications. We exploit the characteristics of scratchpad memories and DSP applications to reduce memory area while allowing aggressive voltage scaling at high yields. Our approach allows dynamically adjustable memory word lengths according to the precision requirements of the application.

The novel contributions of this paper include:

(i) Area and V_{ddmin} reduction according to the on-chip memory footprints of applications (Section III),
(ii) Area and V_{ddmin} reduction by exploiting the precision requirements of applications (Section IV),
(iii) Low-voltage operation (below V_{ddmin}), exploiting the error tolerance of DSP applications (Section IV), and

978-3-9810801-7-9/DATE11/©2011 EDAA

(iv) Dynamic reconfiguration to provide up to 32-bit precision at low V_{ddmin} (Section V).

II. SRAM Cell Failure Probability and V_{ddmin} versus Cell Size

A. Impact of Transistors Size on RDF and LER

The main source of SRAM cell failures at low voltage is due to random dopant fluctuations (RDF) and line edge roughness (LER) [4]. These sources result in device mismatches in SRAM cells that employ symmetric, cross-coupled circuit topology, making such cells unstable at low voltage. The degree of device mismatches, i.e., the standard deviations of each transistors threshold voltage (V_{th}) and channel length (L) are given by [5]:

$$\sigma_{V_{th}} = \sqrt{\frac{q}{3\epsilon_{ox}}} \cdot \sqrt{T_{oxe}(V_{th0} - V_{FB} - 2\phi_B)}\sqrt{WL} \quad (1)$$

$$\sigma_L = \sigma_{L_{LER}}/(\sqrt{1 + \frac{W}{W_c}}) \quad (2)$$

where q is the charge of an electron (1.6×10^{-19}),ϵ_{ox} is the permittivity of SiO_2 (3.5×10^{-13}), T_{oxe} is the gate-oxide thickness, V_{th0} is V_{th} at zero body bias, V_{FB} is the flat-band voltage, ϕ_B is the Fermi potential, W is the transistor width, $L_{LER} = 0.5$ and $W_c = 15nm$ for 32nm or smaller technologies. As shown by Equations 1 and 2, the magnitude of the variations in V_{TH} and L increases as transistor size decreases. As a result, typical SRAM cells that use minimum geometry transistors to improve integration density also exhibit a large amount of device mismatch, which exacerbates the negative impact of such device mismatches on SRAM failures at low voltage. Thus, to achieve a low V_{ddmin} at a target yield, on-chip caches must use (i) larger transistors that exhibit less variation and/or (ii) special circuit and logic techniques, e.g., read/write assist, redundancy, ECC, etc. [4], [6], [7] that mitigate the negative impact of variation on SRAM reliability.

B. Impact of Cell Size on Cell and Cache Failure Probability

To estimate the failure probability of SRAM cells for our study, we use a most probable failure point (MPFP) analysis method, similar to [8]. The $\sigma_{V_{th}}$ values for NMOS and PMOS transistors with W equal to the minimum L in the high-performance 32nm predictive technology model (PTM) [9] is $24mV$ and $29.2mV$, respectively [10]. We used a baseline cell that has the minimum width ($W = 3\lambda$) for all six transistors, and created a layout using a TSMC $0.18\mu m$ technology design rule and Cadence® Virtuoso® layout editor to estimate the area of SRAM cells.

Each SRAM cell consists of a pair of pull-down (PD), pass-gate (PG), and pull-up (PU) transistors; the widths of PD, PG, and PU transistors are represented by W_{PD}, W_{PG}, and W_{PU}, respectively. Since W_{PG} is often equal to or smaller than W_{PD}, its contribution to overall area is not significant. Thus, the sum of W_{PD}, W_{PG}, and a fixed cost (e.g., the minimum spacing between the active to N-well) determines the overall area of each SRAM cell. Finally, the cell height is often fixed in industry SRAM cells while $W_{PD} + W_{PU}$ determines the width of the cell; the minimum size cell has $W_{PD} + W_{PU}$ equal to 6λ plus the fixed width cost. For each cell with a constant $W_{PD} + W_{PU} = W_{CONST}$, we sweep the size of W_{PD}, W_{PG}, and W_{PU} to minimize cell failure probability, $P_{FAILCELL}$ which is the maximum of the read or write failure probabilities of a cell for given W_{PD}, W_{PG}, and W_{PU}.

	c0	c1	c2	c3	c4	c5
$\mathbf{W_{PD}}$	3λ	5λ	8λ	11λ	14λ	16λ
$\mathbf{W_{PU}}$	3λ	4λ	4λ	4λ	4λ	5λ
$\mathbf{W_{PG}}$	3λ	5λ	8λ	11λ	14λ	16λ
Relative Area	1.00	1.12	1.23	1.35	1.46	1.58

TABLE I: Relative area and transistor widths of the SRAM cells

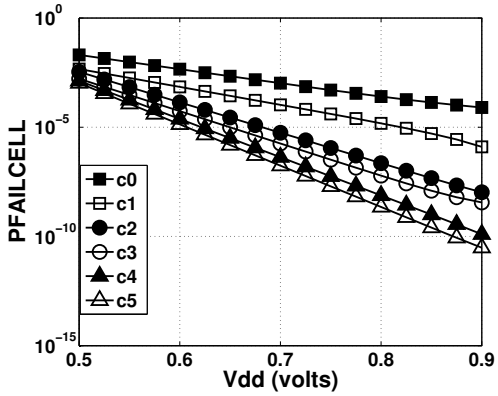
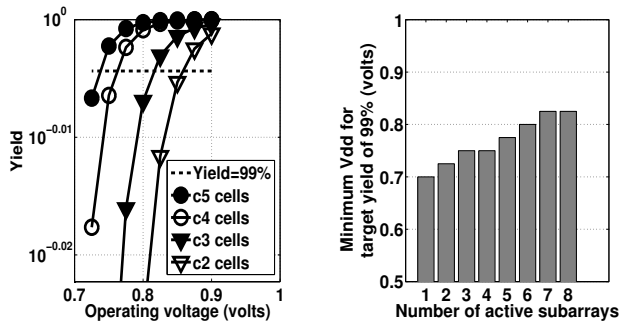


Fig. 1: $P_{FAILCELL}$ vs. operating voltage

Table I tabulates the optimized cell area for each cell (relative to the area of $c0$) and Figure 1 plots $P_{FAILCELL}$ versus V_dd for different cell sizes. Each cell size can have W_{CONST} equal to 9, 12, 15, 18, or 21λ. All the possible combinations of W_{PD}, W_{PG}, and W_{PU} were exhaustively searched to minimize the failure probability at $600mV$ with the step value equal to 0.5λ for a given cell size, i.e., W_{CONST}. In 32nm technology with a given $\sigma_{V_{th}}$ for NMOS and PMOS transistors, the P_{FAIL} is fairly high for the small cells, $c0$ and $c1$. This is because transistors with smaller W_{PD}, W_{PG}, and W_{PU} exhibit larger $\sigma_{V_{th}}$ and σ_L as can be seen by examining Equations 1 and 2. The (on-current) characteristics of these cells are more sensitive to V_{th} and L variability, leading to higher failure probabilities for these cells than for larger cells. Note that $P_{FAILCELL}$ decreases by orders of magnitude at the same voltage as cell size increases; the $P_{FAILCELL}$ for the largest two cells, $C4$ and $C5$, are very close to Intel's data based on the 45nm technology [5].

III. Memory footprint-based optimizations

The on-chip memory footprint of DSP applications can vary considerably. While some applications, such as channel decoding, may require storage of large blocks of data in the scratchpad memory, most DSP applications are likely



(a) Yield vs. operating voltage (V_{dd}) for 32-KB memory (b) V_{ddmin} for different numbers of active subarrays

Fig. 2: V_{ddmin} required for a memory composed of a single and multiple SRAM cell types

to perform computations on small blocks of streaming data. Table II shows the on-chip data memory requirements of some common DSP applications. LDPC decoding for the IEEE-802.16e standard (WiMAX) has the highest memory requirement of 31 KB. Therefore, we chose a memory size of 32 KB to ensure the working set of each application fits in the scratchpad memory.

Application	Memory footprint
Cholesky decomposition (16 x 16)	1 KB
Singular value decomposition (16 x 16)	2 KB
QR decomposition (16 x 16)	3 KB
MPEG4 decoding	4 KB
1024-point Fast Fourier transform	12 KB
128-tap FIR filter	16 KB
2048-point Fast Fourier transform	24 KB
LDPC decoding (block length = 2304 bits)	31 KB

TABLE II: On-chip memory footprints of benchmarks

Figure 2(a) shows the yields at different operating voltages for a 32 KB memory composed of $c5$, $c4$, $c3$, or $c2$ cell types. For the memory designed with $c2$ cells, the area can be reduced by 22% compared to $c5$ cells. However, the memory requires a higher V_{ddmin} $(0.875V)$ to achieve the same target yield. Cell sizes $c0$ and $c1$ can achieve higher area reductions but result in extremely low yields even at higher operating voltages. Their yields are not visible in Figure 2(a).

To balance the area and power requirements, we optimized the scratchpad memory design by tailoring it to the needs of the most common applications. As shown in Table II, most of the applications have a low memory footprint that can fit within 4 KB of memory. Applications such as 1024-point FFT and 128-tap FIR filters have memory footprints in the range 4 KB to 16 KB, while only a few applications have higher memory requirements. We thus designed the scratchpad memory such that, for the most common applications, the required yield is achieved at a lower V_{ddmin}. For applications with higher memory requirements, we trade off the power benefit of low-voltage operation to achieve area reduction.

Scratchpad memories are usually designed using multiple subarrays of SRAM cells. The data addresses provided to the memory are first pre-decoded to select one or a subset of the subarrays. Then, the rest of the address bits are used to select the word line within the selected subarray(s).

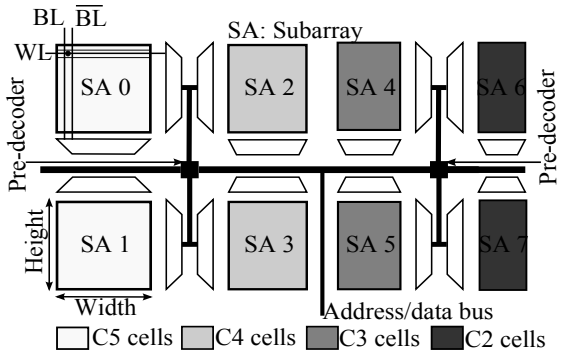


Fig. 3: Scratchpad memory organization optimized to the memory footprints of applications

We partitioned the scratchpad memory into eight subarrays of 4 KB each, as shown in Figure 3. The subarrays are arranged so that the different cell sizes only affect the width of subarrays; only the width varies for different SRAM cell sizes while the height is fixed (cf. Section II). Since the height of the subarrays remains the same, memory layout does not incur any wasted area. Since the data placement in scratchpad memories is under software control, the number of active subarrays for a given application can be pre-determined. Thus for applications that require 4 KB or less memory, only one subarray is activated and the others are turned off. This allows us to achieve the target yield of 99% at a lower $V_{ddmin}(0.700V)$ compared to the case in which all subarrays are active; the number of cells impact the overall failure probability (and thus V_{ddmin}).

To maximize the DVFS potential for applications with smaller memory footprints, the first two subarrays are composed of $c5$ cells because they have the lowest failure rate. Since the number of applications that have a higher memory requirement gradually decreases, we sized the subsequent subarrays accordingly. Specifically, the next two subarrays are composed of $c4$ cells while the last four subarrays are designed with $c3$ and $c2$ cells as shown in Figure 3.

Figure 2(b) shows the V_{ddmin} required when different numbers of subarrays are active. As the memory footprint of applications increases, a gradual increase in V_{ddmin} is required to achieve the target yield. Since most applications are likely to utilize fewer than eight subarrays, this organization achieves an overall power reduction with DVFS compared to a design with only $c3$ or $c2$ cells.

This memory organization has 11% less area than a memory composed only of $c5$ cells. Moreover, compared to a memory composed of $c5$ cells alone, we reduced the V_{ddmin} from 0.75V to 0.700V for applications with a memory requirement of 4KB or less by only activating one subarray. This

corresponds to dynamic and leakage power reductions of 29% and 26%, respectively in the $32nm$ technology node when DVFS is employed to reduce the processor power. These power numbers were determined based on the peak frequency at V_{ddmin} for the two memory designs. Assuming there is no change in the switched capacitance (C) and switching activity (α) of the processor, the dynamic power reduction can be estimated using the Equation $P_{dynamic} = \alpha C V_{ddmin}^2 f$. The leakage power reductions are estimated by determining the leakage current values at V_{ddmin} for the two designs and using the Equation $P_{leak} = I_{leak} V_{ddmin}$. Based on Table II, one or two subarrays will suffice for most applications.

IV. Precision-based optimizations

DSP applications in general have more computational error tolerance than general-purpose applications. Consequently, DSPs often employ reduced precision arithmetic to decrease the power dissipation of the processor. If the word length of the scratchpad memory is also reduced according to the precision requirements of the application domain, substantial savings can be achieved in terms of both area and power.

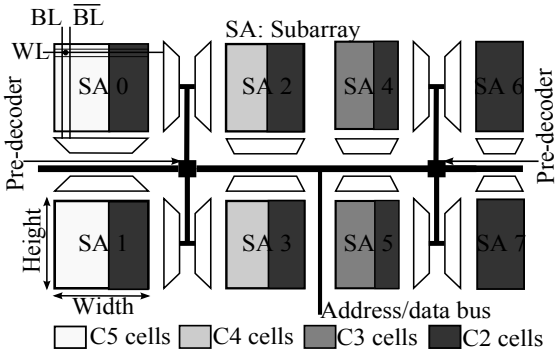


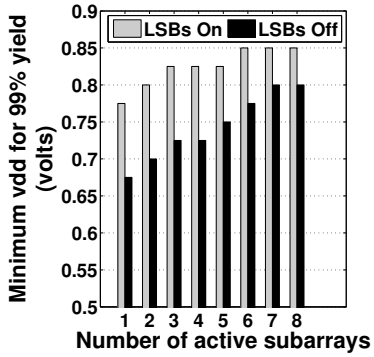
Fig. 4: Scratchpad memory organization optimized to the precision requirements of applications

However, a major drawback of this approach is that the precision is limited to the word size. Modern DSPs often need to support a wide range of applications. The error tolerances of these applications can vary considerably and a hard limit on the memory word length can increase the errors beyond the acceptable bounds for applications that require a higher precision.

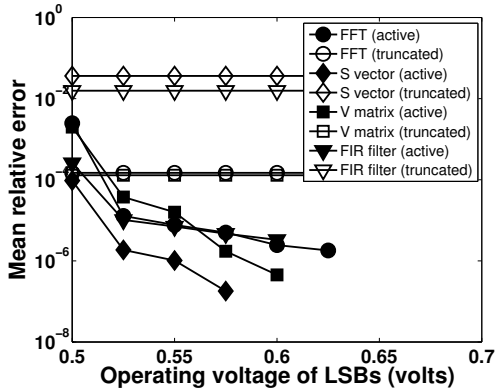
While some DSP applications require longer word-lengths, for most common DSP applications, such as digital filters and short FFTs, shorter word lengths are sufficient. Even for applications that perform floating-point arithmetic, power constraints may necessitate light-weight floating-point hardware with reduced significand precision and fewer exponent bits [1]. However, it is likely that emerging applications will require higher precisions. Therefore, hardware must provide support for longer word lengths.

Based on the varying precision requirements, we propose a novel approach that trades off the operating voltage with precision while still reducing overall area. Specifically, we use the more robust cells for the 16 most significant bits (MSBs) of each memory word. These MSBs are used to implement the most common DSP applications that do not require the complete 32-bit precision. For the 16 least significant bits (LSBs), we use $c2$ cells to reduce memory area. Although these $c2$ cells are less robust and increase the error probability of the memory, we can achieve a lower failure rate and a higher yield by increasing the supply voltage.

A pseudo-layout of memory is shown in Figure 4. Each subarray contains the $c2$ cells for the LSBs. The height of each subarray is determined by the number of words while the width is determined by the width of cells in each word. The inclusion of cells of different sizes only decreases the width of each subarray. Each subarray still contains one word per row. From the pseudo-layout shown in Figure 4, it can be seen that a variation in width of the subarray does not incur any routing complexity for the bit-lines or word-lines. Thus, any area overhead due to layout complexity is avoided.



(a) V_{ddmin} required for 99% yield



(b) Mean relative error with $c2$ cells for 16 LSBs vs. 16-bit word length

Fig. 5: V_{ddmin} and mean relative error improvement

Specifically, our approach provides a dynamically adjustable precision, by turning the LSBs on or off according to the precision requirements of the application. Consequently, we achieve a high yield at lower V_{ddmin} when complete precision is not crucial to the application. For applications that require complete 32-bit precision we enable the LSBs and increase

the operating voltage to satisfy the yield requirement.

Figure 5(a) shows the V_{ddmin} required for different memory footprints for 16- and 32-bit precision. For applications that do not demand complete precision, the operating voltage can be reduced while achieving the same yield by turning off the 16 LSBs. Consequently, we achieve an area saving of 17% and reduce V_{ddmin} from $0.75V$ to $0.675V$. For the $32nm$ technology node, the lower V_{ddmin} allows dynamic and leakage power savings of 42% and 36%, respectively with DVFS for the processor compared to the memory composed of $c5$ cells alone.

Comparison with truncation

Although our approach consumes more area than truncating the memory word size to 16 bits, we can allow implementation of a diverse range of applications on the processor by dynamically adjusting the precision and operating voltage. Moreover, our approach can provide high precision at a lower V_{ddmin} than a short 16-bit word-length memory. Specifically, in our approach, if – instead of turning off the LSBs – we operate them at the same voltage as required by the MSBs, we still provide a much better precision than short word-length memory. This is because, while the LSBs are more likely to cause errors, the mean error is likely to be much smaller than with truncation.

Figure 5(b) shows the mean relative error in three applications, the 1024-point FFT, the 16 x 16 SVD and the 128-tap FIR filter. The graph shows the errors in the output of the single-precision floating-point FFT algorithm, the singular values (S) and the matrix V of the SVD algorithm, and the output of the FIR filter, when LSBs are not turned off and the memory is operated at a lower voltage. Although the FFT and FIR filter algorithms require more than one subarrays (3 subarrays for the FFT and 4 subarrays for the FIR filter) and thus, a higher a V_{ddmin} of $0.825V$ to achieve 99% yield, we operate the memory at $0.675V$, the operating voltage required when only the MSBs of one subarray are active. In Figure 5(b), the voltage of the LSBs is gradually increased from $0.5V$ to show the error trend of the algorithms. Although we do not see any errors at $0.675V$, even if the error lines are projected towards higher voltages, the mean relative error is likely to be less than 10^{-6}. Compared to the short word-length memory operating at $0.75V$ with the 16 LSBs truncated, we achieve a much higher precision at a lower operating voltage.

V. Reconfiguration

The area and operating voltage reduction achieved through precision-based optimizations comes at the cost of a higher V_{ddmin} for applications that require full 32-bit precision. To ameliorate the impact of the high failure rate of the LSBs, we propose a reconfigurable scratchpad organization that can trade off memory capacity for a higher precision at the same V_{ddmin}.

Reconfiguration is achieved by selecting two memory subarrays during the pre-decoding stage such that the LSBs are read from the more robust cells of the next subarray. This allows

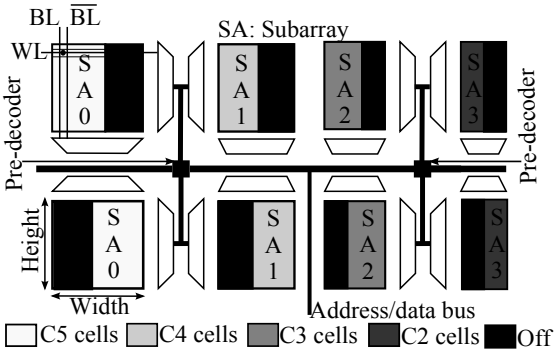


Fig. 6: Reconfigured scratchpad memory organization for high-precision applications

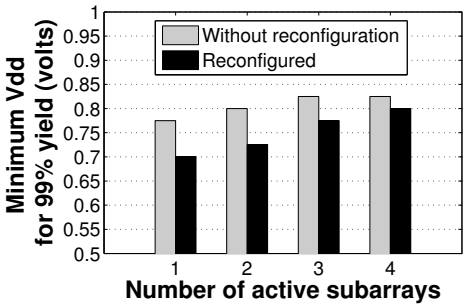


Fig. 7: Operating voltages required with and without reconfiguration

us to turn off the cells with high failure rates in each subarray and consequently decrease the V_{ddmin}. With 4 KB subarrays, this reduces our scratchpad memory capacity by half but provides complete 32-bit precision at a lower V_{ddmin} while still achieving an overall area reduction. The organization of these subarrays is shown in Figure 6. As compared to Figure 4, pairs of subarrays are combined to form a single logical subarray. The MSBs are read/written to the top subarrays while the LSBs use the bottom subarrays. The orientation of MSBs and LSBs in each alternate subarray is switched to reduce the complexity associated with multiplexing the bit lines.

Figure 7 shows the V_{ddmin} reduction corresponding to the size of the memory footprints. Since the memory capacity has been reduced by half, only four effective subarrays can be used by applications. However, we achieve V_{ddmin} reduction when any number of these subarrays are active.

VI. Related work

Chang *et al.,* proposed an approach to reduce the SRAM area and energy for MPEG decoding. They exploited the error tolerance of MPEG decoding to integrate $8T$ and $6T$ SRAM cells for the MSBs and LSBs respectively [14]. Our approach provides a flexible alternative that spans a wide range of applications.

Cho *et al.,* [15] proposed a reconfigureble SRAM architecture that uses two different voltage domains for the MSBs and LSBs. The number of bits in each voltage domain is

TABLE III: Summary of results

Design	Capacity	Precision	Area reduction	Minimum V_{ddmin} (One subarray active)	Maximum V_{ddmin} (All subarrays active)
All cells are $c5$ (Baseline)	32 KB	32-bit	0%	0.750V	
All cells are $c2$	32 KB	32-bit	22%	0.875V	
Footprint-based optimization	32 KB	32-bit	11%	0.700V	0.850V
Precision-based optimization	32 KB	Variable	17%	0.675V(16-bit) 0.775V(32-bit)	0.800V(16-bit) 0.825V(32-bit)
Below V_{ddmin} operation	32 KB	32-bit	17%	0.675V(Application dependent)	
Reconfiguration	Variable (16 KB)	Variable (32-bit)	17%	0.700V(32-bit)	0.800V(32-bit)

reconfigureable during run-time to achieve power savings. Our approach uses different cell sizes for the MSBs and LSBs to achieve area and V_{ddmin} reduction.

V_{ddmin} reduction can also be achieved using error correction codes (ECC). However, ECC based approaches incur area and delay penalties that render them less beneficial for first level caches and scratchpad memories [16]. All of our results assume that ECC are not employed in the scratchpad memory. In the presence of ECC, our optimizations will achieve higher V_{ddmin} reduction.

Amelifard *et al.,* proposed a hybrid cell SRAM for leakage power reduction. They employed different cell configurations for SRAM cells to exploit the delay variations corresponding to the location of cells in the SRAM to reduce its leakage power [11].

Significance compression schemes have been proposed by Ghosh *et al.,* [12] and Canal *et al.,* [13]. These schemes can reduce memory footprint of applications by compressing leading zeros/ones in fixed-point data. However, these approaches are data-dependent and can not be used for statically turning off subarrays are compile time.

VII. Conclusion

We proposed a scratchpad memory design that exploits the trade offs between cell sizes, memory yield, V_{ddmin}, and application characteristics to achieve area and voltage reductions. We can achieve an area reduction as high as 17% while still supporting multiple precisions and low V_{ddmin} for most applications. We exploited the error-tolerance of DSP applications to reduce the operating voltage below V_{ddmin}. Even with the errors incurred due to low-voltage operation, our approach provides a much higher precision than a short word-length memory. This allows further power reduction through voltage scaling without violating the error-margin of the application. Table III presents a summary of our results. For applications that require complete 32-bit precision, we proposed a dynamically reconfigurable subarray organization. With dynamic reconfiguration, we can support complete 32-bit precision at a lower V_{ddmin} with decreased memory capacity. Thus, our proposed design has a variable capacity, 32 KB (without reconfiguration) or 16 KB (with reconfiguration), and variable precisions of 16 or 32 bits. The proposed design can thus adapt efficiently to the disparate requirements of DSP applications and can also be applied to other error tolerant application domains.

ACKNOWLEDGEMENT

This work was supported by two NSF grants (CCF-0953603 and CCF-1016262) and generous gift grants from Microsoft and AMD.

References

[1] Fang, F., Chen, T., and Rutenbar, R. A., "Lightweight floating-point arithmetic: case study of inverse discrete cosine transform," *EURASIP J. Appl. Signal Process.* , 879–892 (2002).

[2] Manne, S., Klauser, A., and Grunwald, D., "Pipeline gating: speculation control for energy reduction," *SIGARCH Comput. Archit. News* **26**(3), 132–141 (1998).

[3] Thomas, O., Reyboz, M., and Belleville, M., "Sub-1V, robust and compact 6T SRAM cell in double gate MOS technology," *IEEE International Symposium on Circuits and Systems* , 2778–2781 (2007).

[4] Zhang, K., Bhattacharya, U., Chen, Z., Hamzaoglu, F., Murray, D., Vallepalli, N., Wang, Y., Zheng, B., and Bohr, M., "A 3-Ghz 70MB SRAM in 65nm CMOS technology with integrated column-based dynamic power supply," *IEEE International Solid-State Circuits Conference* , 474–611 Vol. 1 (2005).

[5] Wilkerson, C., Gao, H., Alameldeen, A. R., Chishti, Z., Khellah, M., and Lu, S.-L., "Trading off cache capacity for reliability to enable low voltage operation," *SIGARCH Comput. Archit. News* **36**(3), 203–214 (2008).

[6] Khellah, M., Kim, N. S., Ye, Y., Somasekhar, D., Karnik, T., Borkar, N., Hamzaoglu, F., Coan, T., Wang, Y., Zhang, K., Webb, C., and De, V., "PVT-variations and supply-noise tolerant 45nm dense cache arrays with diffusion-notch-free (DNF) 6T SRAM cells and dynamic multi-vcc circuits," *IEEE Symposium on VLSI Circuits, 2008* , 48–49 (2008).

[7] Schuster, S., "Multiple word/bit line redundancy for semiconductor memories," *IEEE Journal of Solid-State Circuits* **13**(5), 698–703 (1978).

[8] Khalil, D., Khellah, M., Kim, N.-S., Ismail, Y., Karnik, T., and De, V., "Accurate estimation of SRAM dynamic stability," *IEEE Transactions on Very Large Scale Integration Systems* **16**(12), 1639–1647 (2008).

[9] http://www.eas.asu.edu/ ptm, "Predictive technology model."

[10] http://www.itrs.net/links/2009ITRS/Home2009.htm, "ITRS 2009 edition."

[11] Chang, I. J., Mohapatra, D., and Roy, K., "A voltage-scalable & process variation resilient hybrid SRAM architecture for MPEG-4 video processors," in [*Design Automation Conference*], 670–675, ACM (2009).

[12] Cho, M., Schlessman, J., Wolf, W., and Mukhopadhyay, S., "Accuracy-aware SRAM: a reconfigurable low power SRAM architecture for mobile multimedia applications," in [*Asia and South Pacific Design Automation Conference*], 823–828 (2009).

[13] Mohr, K. and Clark, L., "Delay and area efficient first-level cache soft error detection and correction," in [*International Conference on Computer Design*], 88 –92 (2006).

[14] Amelifard, B., Pedram, M., and Fallah, F., "Low-leakage SRAM design with dual Vt transistors," in [*International Symposium on Quality Electronic Design*], 729–734 (2006).

[15] Ghosh, M., Shi, W., and Lee, H.-H., "CoolPression - a hybrid significance compression technique for reducing energy in caches," in [*SOC Conference*], 399 – 402 (2004).

[16] Canal, R., González, A., and Smith, J. E., "Very low power pipelines using significance compression," in [*ACM/IEEE international symposium on Microarchitecture*], 181–190 (2000).