

Diet SODA: A Power-Efficient Processor for Digital Cameras

Sangwon Seo¹, Ronald G. Dreslinski¹, Mark Woh¹, Chaitali Chakrabarti²,
Scott Mahlke¹, and Trevor Mudge¹

¹Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, MI 48109

²School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287
{swseo, rdreslin, mwoh, mahlke, tnm}@umich.edu, chaitali@asu.edu

ABSTRACT

Power has become the most critical design constraint for embedded handheld devices. This paper proposes a power-efficient SIMD architecture, referred to as Diet SODA, for DSP applications. The key design idea is to apply near-threshold operation on a single instruction and multiple data (SIMD) architecture to significantly lower the power consumption. The major features of Diet SODA are very wide SIMD width, scatter/gather data prefetcher, and dual mode operation. A case study was performed on digital still camera (DSC) applications; the results show that Diet SODA achieves $\sim 130x$ better performance and $\sim 340x$ better energy efficiency than a DSP solution.

Categories and Subject Descriptors

C.1.2 [Processor Architectures]: [Multiple Data Stream Architectures (Multiprocessors)]; C.3 [Special-Purpose and Application-Based Systems]: [Signal processing systems]; B.6.1 [Logic Design]: [Design Styles]

General Terms

Algorithms, Design, Performance

Keywords

SIMD, near-threshold, dynamic voltage scaling, digital still cameras

1. INTRODUCTION

Mobile devices have rapidly proliferated and the deployment of the handheld devices will continue to increase at a spectacular rate. As today's devices not only support advanced signal processing of wireless communication data but also provide for richer sets of various applications, power dissipation has become a more important design constraint. Increasing power consumption leads to increasing energy costs as well as impacts chip reliabilities. Therefore, more power-efficient processors for embedded DSP applications are highly required.

Among many DSP applications, high resolution cameras have become an integral part of most cell phone designs. As a result, the market for these cameras has mirrored the spectacular growth in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISLPED'10, August 18–20, 2010, Austin, Texas, USA.

Copyright 2010 ACM 978-1-4503-0146-6/10/08 ...\$10.00.

mobile phones [3]. Furthermore, the expectation is that these mobile cameras produce an image whose quality should approach that of high quality digital still cameras (DSCs). Therefore, a DSC processor needs to be of high-performance to support a large amount of image data and perform the DSC image processing tasks in a highly energy-efficient manner in order to conserve critical battery life for other phone applications.

Traditionally, the DSC image signal processing pipeline is implemented in digital signal processors (DSPs) or application specific integrated circuits (ASICs). DSP-based solutions [2] support high flexibility and handle various DSC algorithms, but they suffer from lower performance and higher energy consumption than ASIC solutions. ASIC-based solutions [4, 5] are highly specialized and optimized for the DSC image signal processing pipeline, but such designs lack flexibility and require longer design time. Therefore, to achieve high processing performance efficiency at low cost while maintaining programmability, hybrid architectures [6] are employed. In these designs, ASICs are typically used for a preview mode, where high processing capabilities are desired, and DSPs are adopted for picture-taking and post-processing modes, where flexibility is more important. However, such a heterogeneous solution is inefficient to build and maintain.

To address these challenges, this paper presents a power-efficient programmable architecture, Diet SODA, that has been optimized for DSC image signal processing. Diet SODA exploits near-threshold operation [20] on a wide-SIMD architecture — SODA [11]. In the near-threshold operation, circuits operate at lower than normal supply voltages, reducing power consumption by $\sim 100x$. Near-threshold operation offers a new opportunity for mobile applications such as DSCs to reduce power consumption. However, the reduction in power consumption comes at a cost of a $\sim 10x$ performance degradation. Diet SODA overcomes these hurdles by exploiting architectural features specific to near threshold operation. The key features of Diet SODA are 1) very wide SIMD width to exploit the significant amount of data level parallelism (DLP) inherent in DSC applications, which helps overcome the frequency loss from operating in the near-threshold region; 2) scatter-gather data prefetcher to support 2D memory access enabled by the latency difference between the full voltage SIMD memory and SIMD data engine operating at near-threshold voltage; and 3) dual voltage modes where the SIMD data engine operates at either full or near-threshold voltage based on processing demands. The customized architecture was implemented in Verilog and synthesized in IBM 90nm technology using Synopsys physical compiler.

The rest of the paper is organized as follows. Section 2 gives a brief overview of near-threshold operation. Section 3 analyzes the computational characteristics of DSC signal processing algorithms for preview mode, picture-taking mode and post-processing mode. Section 4 introduces Diet SODA, the low-power DSP processor for DSCs with an analysis of design choices created by using near-threshold operation. Section 5 presents the performance, power, and comparison analysis of Diet SODA. Section 6 discusses the related work and Section 7 concludes the paper.

2. NEAR THRESHOLD OPERATION

The original SODA architecture was targeted for applications that require substantial processing to meet time-critical tasks in software defined radio applications on a limited energy budget. There are a considerable number of applications, such as DSC, that operate on an even tighter energy budget, but where timing is less critical. A paradigm shift is necessary for these applications to further reduce energy consumption.

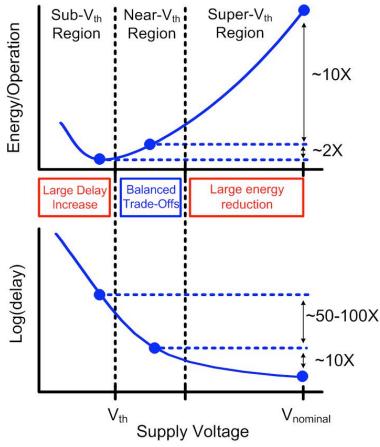


Figure 1: Supply voltage operating regions and the energy and delay associated at each point. The near-threshold region provides considerable energy savings for non-timing critical low power applications such as DSCs.

Near-threshold operation, as described by Zhai et al. [20], defines three regions of operation, pictured in Figure 1. In the superthreshold regime ($V_{dd} > V_{th}$), energy is highly sensitive to V_{dd} due to the quadratic scaling of switching energy with V_{dd} . Hence, voltage scaling down to the near-threshold regime ($V_{dd} \sim V_{th}$) yields an 10x energy reduction at the expense of approximately 10x performance degradation. However, the dependence of energy on V_{dd} becomes more complex as voltage is scaled below V_{th} . In subthreshold regime ($V_{dd} < V_{th}$), circuit delay increases exponentially with V_{dd} , causing leakage energy (the product of leakage current, V_{dd} , and delay) to increase in a near-exponential fashion. This rise in leakage energy eventually dominates any reduction in switching energy, creating an energy minimum.

The identification of an energy minimum led to interest in processors that operate at this energy optimal supply voltage [23]. However, the energy minimum is relatively shallow. Energy typically reduces by only $\sim 2x$ when V_{dd} is scaled from the near-threshold regime to the subthreshold regime, though delay rises by 50-100x over the same region. While acceptable in ultra-low energy sensor-based systems, this delay penalty is not tolerable for a broader set of applications.

The near-threshold region offers an opportunity for applications, such as DSC, to reduce energy further. In order to do so, the design must overcome one hurdle, the 10x increase in delay. This delay impacts the ability of designs to meet more stringent real time constraints without scaling the voltage higher and losing energy efficiency. However, in cases where the application can be parallelized, simply using more near-threshold processing elements can meet the timing constraint with greater efficiency. Near-threshold operation, therefore, has a natural synergy with data parallel environments like SIMD. In a SIMD architecture, the number of functional units can be increased to help meet a timing critical code, provided the application has sufficient DLP.

In addition, the fact that near-threshold operation decreases frequency offers several new and interesting design choices related to the memory system. First, memory devices that are slower and

more energy efficient can be used to replace previously timing critical memories. This will help to reduce the overall energy consumption of the chip. Second, multiple accesses to memory can be performed in one near-threshold clock cycle. This means that data patterns that were impossible at full speed could be achieved using new hardware that scatter-gathers memory requests. And, third, the slower memory allows for elements originally designed to hide long memory latency, i.e. caches and register files, to be turned off or eliminated.

3. DSC ALGORITHM ANALYSIS

3.1 DSC Signal Processing Pipeline

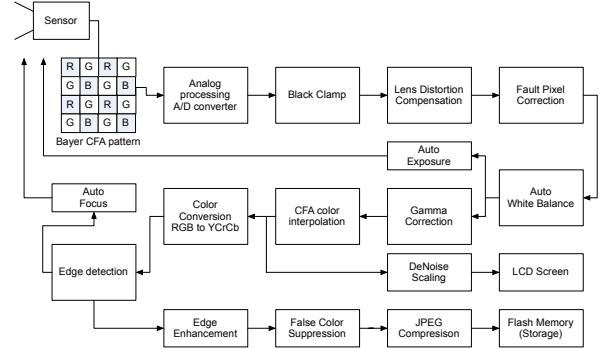


Figure 2: A typical DSC image signal processing pipeline [1], [2]

Figure 2 shows a typical DSC image signal processing pipeline, which performs multiple processing steps to generate a high-quality image [2]. The image is first captured by a CCD or CMOS sensor using a Bayer-pattern [21] color filter array (CFA). Then, the image is digitized with a 10- or 12-bit A/D converter. The *Black Clamp* adjusts the pixel values by subtracting a black offset value from all pixel values. The *Lens Distortion Compensation* adjusts the brightness of pixels depending on the spatial locations and the *Fault Pixel Correction* interpolates defective pixels with neighboring pixels. After all these pre-processing steps, *Auto White Balance* computes the average brightness of each color component and balances the energy of the colors. Based on the brightness information, *Auto Exposure* appropriately adjusts the CCD or CMOS exposure time and gain. After the white balanced image pixels are compensated by *Gamma Correction*, *CFA color interpolation* uses the one-color-per-pixel Bayer-pattern image to interpolate and generate the full color (R, G, and B) resolution for each pixel. The RGB color pixels are filtered by *De-Noise* and scaled down and sent to the LCD screen in preview mode. In picture-taking mode, the noise-filtered images are transformed to the YCrCb color domain. *Edge Detection* detects edges to help *Auto Focus*, and *Edge Enhancement* is performed. Next, *False Color Suppression* occurs, and finally the image is compressed by using *JPEG Compression* and stored in flash memory. Later, post-processing tasks such as *Histogram Calculation*, *Histogram Equalization*, and *Spatial Frequency Filtering* are used to enhance the quality of the stored images.

3.2 Characteristics of DSC Algorithms

In this section, we analyze the key algorithms in the two modes (preview and picture-taking) of DSC signal processing pipeline and post-processing tasks to find opportunities for improving the processing performance and energy efficiency.

Table 1 presents the data level parallelism (DLP) analysis of the DSC signal processing algorithms. Instructions are broken down into three categories: SIMD, scalar, and overhead workloads. The SIMD workload consists of traditional arithmetic/logical functional operations and load/store operations that can be executed in SIMD-fashion. The scalar workload consists of instructions running only

Mode	Task	SIMD	Scalar	Overhead
Preview	Black Clamp	100%	0%	0%
	White Balance	100%	0%	0%
	Auto Focus	71%	14%	14%
Picture-Taking	Gamma Correction	0%	100%	0%
	CFA Interpolation	84%	3%	13%
	Auto Exposure	74%	11%	15%
Post-Processing	Color Conversion	100%	0%	0%
	Edge Detect/Enhance	81%	2%	17%
	Histogram Equalize	37%	44%	19%
	Spatial Freq. Filter	77%	3%	20%

Table 1: Data level parallelism analysis for DSC image signal processing algorithms. Instructions are categorized into three groups: SIMD, scalar, and overhead instructions.

on the scalar datapath such as control instructions and address generations for local SIMD and scalar memories. The overhead workload consists of instructions to assist SIMD computations and scalar computations such as shuffle operations, predication operations, and data movements between the SIMD datapath and scalar datapath. The workloads of each category are calculated based on hand-written assembly codes and are weighted by dynamic execution frequency.

As can be seen in Table 1, most of the DSC signal processing algorithms have significant DLP. Exceptions are *Gamma Correction* and *Histogram Equalization*, where memory access patterns inhibit parallelization. The remaining DSC signal processing algorithms can be grouped into three categories.

(1) Pixel Independent Kernels: In this set of kernels, some basic arithmetic/logical and multiply-and-accumulate (MAC) operations are applied on every pixel independently. Therefore, these kernels can easily be mapped onto a SIMD architecture. *Black Clamp*, *Color Space Conversion*, *Brightness/Contrast Enhancement*, and *Hue/Saturation Enhancements* fall into this category. Although *Gamma Correction* is also a pixel-independent operation, this kernel cannot be easily parallelized on a SIMD architecture because each SIMD lane has to access different memory locations at the same time.

(2) Pixel Dependent Kernels: This set of kernels includes *CFA Interpolation*, *Edge Detection/Enhancement*, and *Spatial Frequency Filtering* that operate on pixels in a 2D neighborhood. The size of the 2D neighborhood is typically 3x3, though 5x5 or 7x7 sizes are also used. Traditional processor architectures spend more than half of the total instructions aligning the 2D data [1]. Therefore, for these kernels, 2D data access must be supported. This is done by a combination of multi-bank memory organization and a SIMD shuffle network.

(3) Statistics Gathering Kernels: The statistical information of the whole or partial frame is gathered for *White Balance*, *Auto Exposure*, *Histogram Calculation* and *Histogram Equalization*. Some of these kernels can be supported by a SIMD adder tree. *Histogram Calculation* is another kernel where memory access patterns inhibit parallelization for a SIMD architecture.

4. DIET SODA ARCHITECTURE

In this section, we propose a power-efficient architecture, referred to as Diet SODA, for DSC processors. Diet SODA exploits key characteristics of the DSC image processing algorithms described in Section 3. This architecture operates in two modes: 1) dual voltage (DV) mode to handle low power applications such as the DSC image processing pipeline, and 2) the full voltage (FV) mode to handle advanced wireless communications. In DV mode, the memory operates at full voltage but the SIMD pipelines operate at near-threshold voltage. In FV mode, the SIMD data engines operate at full voltage as well.

4.1 Diet SODA PE Design

Figure 3 shows the architectural details of a single processing element (PE) of Diet SODA. The PE contains two different voltage domains: full voltage (FV) and dual voltage (DV). DV domain

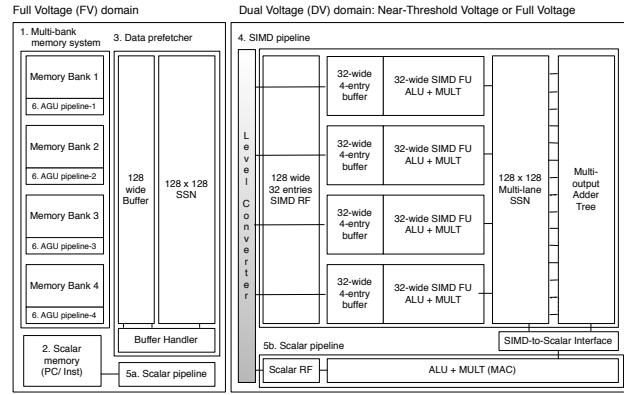


Figure 3: Diet SODA processing element (PE) for DSCs. The PE contains two different voltage domains: full voltage (FV) and dual voltage (DV). DV domain operates at either full or near-threshold supply voltage. The PE consists of: 1) multi-banked SIMD memory; 2) scalar memory; 3) SIMD data prefetcher; 4) SIMD pipeline; 5a) scalar pipeline in full voltage domain; 5b) scalar pipeline in dual voltage domain; and 6) 4-wide address generation unit (AGU) pipeline.

operates at either full or near-threshold supply voltage. The PE consists of: 1) multi-bank SIMD memory; 2) scalar memory; 3) SIMD data prefetcher; 4) SIMD pipeline; 5) scalar pipeline; and 6) 4-wide address generation unit (AGU) pipeline.

The SIMD pipeline consists of a 128-wide 16-bit datapath with a SIMD register file (RF), 128 functional units, 128 4-entry buffers used for intermediate data, a SIMD shuffle network (SSN), and a multi-output adder tree. The SIMD datapath consists of four groups of 32-wide SIMD units. With the support of a multi-banked SIMD memory and an 4-wide AGU pipeline, these groups of SIMD partitions can work on four different memory sections concurrently. There are two scalar pipelines, one in each voltage domain; both pipelines consist of one 16-bit datapath and are used to perform sequential algorithms in addition to coordinating the SIMD units. The 4-wide AGU pipeline handles memory address calculation for the 4-bank SIMD memory and the data prefetcher.

4.2 SIMD Pipeline Width

Although near-threshold operation allows circuits to consume significantly less power, the processing performance also degrades. To compensate for the degraded performance, the number of SIMD lanes is increased.

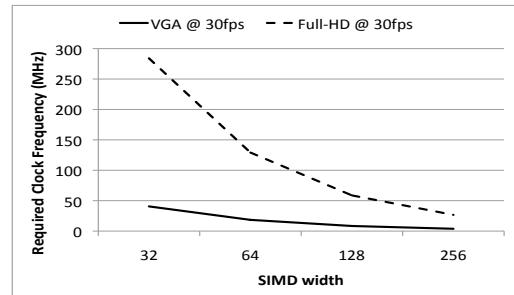


Figure 4: Minimum clock frequencies based on different SIMD width configurations to run the preview mode of DSC signal processing pipeline shown in Figure 2.

The DSC signal processing pipeline for a VGA-size (640x480) image and a full-HD (1920x1080) image at 30 fps is used as the evaluation point to decide the number of SIMD lanes. Figure 4

shows the minimum clock frequency required for VGA and full-HD for different SIMD width configurations — 32, 64, 128, and 256. Thus, to process full-HD images at 30 fps, a 32-wide SIMD pipeline needs to operate at more than 270MHz, while a 256-wide SIMD pipeline needs to operate at around 30 MHz.

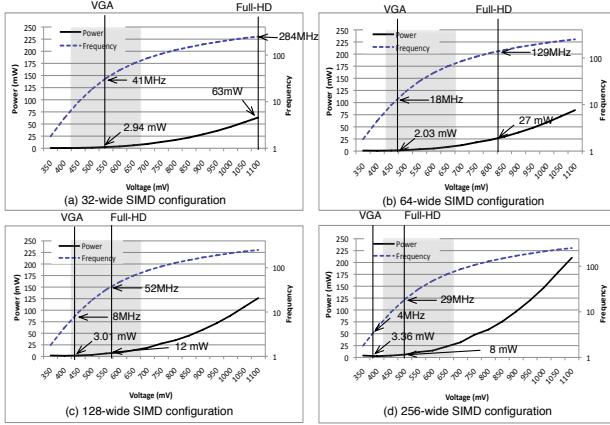


Figure 5: Near-threshold operation is applied to four different SIMD width configurations: 32, 64, 128, and 256. Solid vertical lines provide guidelines for the minimum supply voltage necessary to meet VGA and full-HD processing demands. Gray boxes represent the near-threshold regions.

To investigate how much voltage/frequency scaling can be achieved while still meeting the performance requirements, the power consumption for each SIMD width configuration was measured. First, a representative test circuit was laid out in IBM 90nm technology, parasitic extraction was performed and annotated. Then, SPICE simulations were done to determine the voltage, frequency, and power characteristics at different supply voltages. To obtain power numbers, the SIMD pipeline logic was then synthesized with Synopsys Physical Compiler and scaled to match the representative test circuit. Figure 5 shows power consumption and achievable clock frequencies depending on the corresponding supply voltage for each candidate SIMD width. The solid vertical lines provide guidelines on what the minimum supply voltage is required to process VGA and full-HD images at 30 fps. The results show that although a 32-wide SIMD data engine is capable of handling VGA processing requirements, to support full-HD images, a SIMD width of greater than 32 is required. On the other hand, wider SIMD widths do not always guarantee better energy efficiency due to the additional hardware and critical path delay increases, resulting in a higher minimum clock frequency. In this paper, a 128-wide SIMD configuration is chosen to maximize the benefit of using near-threshold operation while maintaining the real time processing constraints of both VGA and full-HD. With this configuration, the supply voltage needs to be 600mV using a clock frequency of 50MHz.

4.3 Scatter-Gather Data Prefetcher

While operating in the DV mode, the SIMD memory operates significantly faster than the SIMD pipeline. Therefore, two-dimensional (2D) data accesses can be achieved by performing multiple memory accesses to the same memory banks in a single cycle of the SIMD pipeline. There is also sufficient time to perform complicated shuffling operations before delivering the data. Because of these non-traditional memory access patterns and additional shuffling capabilities, a significant reduction in the required number of SIMD instructions can be obtained.

Figure 6 shows the process of data alignment using the SIMD data prefetcher. First, the required data is read from a multi-bank memory. Then the data prefetcher stacks the data in the location

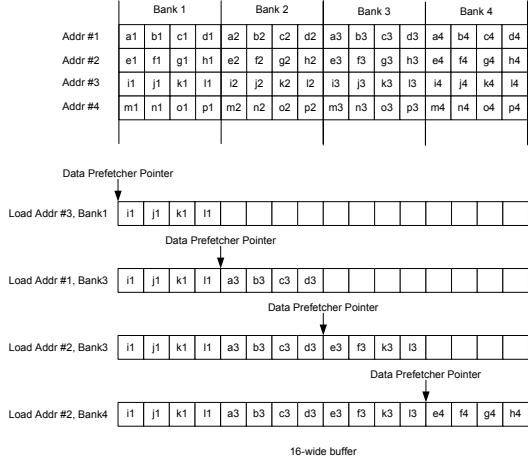


Figure 6: Example of complex data shuffling with 4-bank 4-wide SIMD memory, SIMD data prefetcher, and 16-wide buffer.

indicated by the data prefetcher pointer. The pointer then advances to the next data section and repeats the process for the next load operation. In addition, with the support of SSN, more complex shuffle operations can be implemented.

4.4 Operating Modes

In this section, dual voltage (DV) and full voltage (FV) modes in Diet SODA are described. Table 2 provides the configuration of each component of Diet SODA PE for each mode.

Components	DV Mode	FV Mode
1. Multi-Bank SIMD Memory	on	on
2. Scalar Memory	on	on
3. Data Prefetcher - Buffer/Buffer Handler	on	off
PE 3. Data Prefetcher - SSN	on	on
4. SIMD pipeline - SIMD RF	off	on
4. SIMD pipeline - Other modules except SIMD RF	on@NTV	on
5a. Scalar pipeline	on	off
5b. Scalar pipeline	on@NTV	on
6. 4-wide AGU pipeline	on	on

Table 2: Architectural modules that are turned on and off for dual voltage (DV) and full voltage (FV) modes.

4.4.1 DV Mode

In the DSC signal processing pipeline, preview and picture-taking tasks are performed in DV mode because these tasks do not require very high data processing rates. Consequently, the supply voltage of the SIMD data engine is operated at near-threshold voltage to significantly lower energy consumption. More specifically, the SIMD pipeline and scalar pipeline (5b in Figure 3) in the DV domain operate at near-threshold voltage, while the SIMD memory, scalar memory, SIMD data prefetcher, and 4-wide AGU pipeline operate at full voltage. As can be seen in Table 2, the SIMD RF is switched off because the latency of the SIMD memory is much lower than that of SIMD data engine so the SIMD pipeline is capable of directly handling data from the SIMD memory. This results in a reduction of energy consumption by eliminating SIMD RF accesses. The 4-entry buffer in each SIMD lane operates as a small RF to hold recently produced values for consumption by subsequent instructions.

4.4.2 FV Mode

Recent DSCs support video recording at full-HD (1920x1080) resolution. This necessitates additional processing capability, and therefore requires Diet SODA to operate in FV mode. In this mode,

Components	DV mode		FV mode	
	Area (mm ²)	Area (%)	Power (mW)	Power (%)
SIMD banked-memory (64KB)	3.41	33%	28	23%
SIMD Register Files (4KB)	1.58	15%	0	0%
SIMD Buffer (1KB)	0.41	4%	3	2%
SIMD ALU/Multiplier, SSN	2.26	22%	23	19%
SIMD Adder Tree	0.12	1%	1	1%
SIMD pipeline+Clock+Routing	0.68	7%	12	10%
Data Prefetcher	1.63	16%	33	27%
Scalar/AGU Pipelines & Misc.	0.18	2%	22	18%
Total	90nm(1V@400MHz, 600mV@500MHz)	10.27	100%	122
				1228
				100%

Table 3: Area and Power Summary of Diet SODA for Preview Mode of Full-HD Images at 30 fps. For comparison, the results of both DV mode and FV mode are presented.

the SIMD pipeline operates at full voltage along with the SIMD memory and 4-wide AGU pipeline. On the other hand, the SIMD data prefetcher is turned off because there is no time slack between the SIMD memory and the SIMD data engine to prefetch data in advance. Also, the scalar pipeline (5a in Figure 3) in the FV domain is turned off and another scalar pipeline (5b in Figure 3) in the DV domain works for the overall system. In this mode, the SIMD RF is switched on so that faster operations are supported.

5. RESULTS AND ANALYSIS

5.1 Methodology

The DSC image signal processing pipeline algorithms are implemented in C to evaluate system performance, memory requirements, and non-parallelizable bottlenecks. Next, the C benchmark codes [22] are transformed to assembly codes for Diet SODA. The Diet SODA processor is implemented as an RTL Verilog model and synthesized for IBM’s 90nm technology using the Synopsys Physical Compiler. The clock frequency is targeted for 400MHz @ 1V, and the power numbers for the SIMD data engine are scaled down for 50MHz @ 600mV by the process shown in Section 4.2.

5.2 Area and Power

The area and power breakdown of this processor are presented in Table 3. The preview mode of full-HD images at 30 fps consumes about 122mW and 1228mW in DV mode and FV mode, respectively. Therefore, the DV mode offers about ~ 10 x better power efficiency.

About 68% of the total power dissipation in DV mode is consumed by SIMD memory, the scalar/AGU pipeline and data prefetcher operating at full voltage. In particular, the SIMD memory and data prefetcher consume a large part of the power because the number of SIMD memory accesses and shuffle operations is increased due to the SIMD RF being switched off. However, the SIMD data engine operating in DV mode consumes ~ 21 x less power than the SIMD datapath operating in FV mode, which offsets the increased SIMD memory power and highlights the advantage of using near-threshold operation.

5.3 Performance

Table 4 presents the latencies of DSC processing algorithms - preprocessing (*Black Clamping, Lens Distortion Compensation, Fault Pixel Correction, White Balance, Gamma Correction*), *CFA Interpolation*, *Color Space Conversion*, *Edge Detection/Enhancement, Scaling*, and *JPEG Compression*. As can be seen in Table 4, the preview modes of both VGA and Full-HD images are processed within a time constraint of 33 ms thus meeting the 30 fps requirement.

CFA Interpolation and *Edge Detection/Enhancement* are the most demanding workloads taking about 60% of the processing time. While most algorithms deals with only one color component (R, G, or B) per each pixel location, *CFA interpolation* generates all of three components for each pixel locations. Therefore, the memory size and workload for this interpolation algorithm are increased. *Edge Detection/Enhancement* works with only one component per

Task	Latency (VGA)	Latency (Full-HD)
Black Clamp, Distortion Compensation, Fault Pixel Correction, White Balance, Gamma Correction	0.57 ms	3.89 ms
CFA Interpolation	1.02 ms	6.67 ms
Color Conversion	0.36 ms	2.43 ms
Edge Detection Edge Enhancement	0.82 ms	5.29 ms
False Color Suppression Scaling	0.31 ms	2.11 ms
Total	3.08 ms	20.38 ms

Table 4: The Latencies of DSC signal processing pipeline algorithms for the preview mode of a VGA image and a Full-HD image.

each pixel location, but 3x3 matrix convolutions in this task require significant processing time and shuffling for MAC calculations and realignments.

5.4 Comparison With Other Solutions

The DSC image signal processing pipeline in Figure 7 [1] is used to compare the performance of Diet SODA with one high-end commercial DSP and one coarse-grained reconfigurable image stream processor — TI TMS320C64x [7] and CRISP [1]. The pipeline is divided into three task groups: 1) color gain adjustment, gamma correction and CFA interpolation; 2) noise reduction and smooth filter; and 3) color space conversion and edge enhancement.

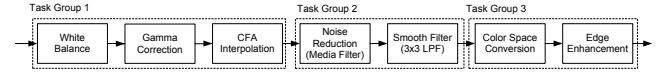


Figure 7: A Test DSC image signal pipeline [1]

Table 5 shows the execution time comparison with TMS320C64x, CRISP, and Diet SODA for a 4072x2720 image. Results show that Diet SODA is approximately 140x and 1.6x faster than TMS320C64x and CRISP, respectively. The wide SIMD datapath allows the DSC image signal processing algorithms to operate on many pixels at the same time. In addition, scatter-gather data prefetcher helps data alignment issues.

	TMS320C64x [7]	CRISP [1]	*Diet SODA PE
Task Group 1	6440 ms	220 ms	110 ms
Task Group 2	20550 ms	110 ms	80 ms
Task Group 3	9690 ms	110 ms	80 ms
Total	36680 ms	440 ms	270 ms

Table 5: Execution Time Comparison with TI TMS320C64x, CRISP, and Diet SODA. Task Group 1 - White Balance, Gamma Correction, CFA Interpolation; Task Group 2 - Noise Reduction, Smooth Filter; Task Group 3 - Color Space Conversion, Edge Enhancement. *Diet SODA operates in DV mode.

Table 6 shows comparisons of technology, area, power consumption, and normalized energy with TMS320C64x and CRISP. Normalized power and area results for 90nm technology are estimated using a quadratic scaling factor based on Predictive Technology Model [8]. The results show that the energy efficiency of Diet SODA is more than 340x better than that of TMS320C64x and also comparable to that of the reconfigurable image stream processor, CRISP [1]. Even though we show comparable energy and only slightly improved performance numbers over the CRISP design, our design is more flexible and maintainable because the reconfigurable interconnection and heterogeneous processing elements of CRISP must be manually designed before fabrication.

6. RELATED WORK

The real-time constraints of media (image/video) applications require high-performance but low-power processors for portable de-

	TMS320C64x [7]	CRISP [1]	Diet SODA PE
Tech.	0.13um	0.18um	90nm
Freq.	600MHz	115MHz	400MHz, 50MHz
Power	718mW@1.2V	218mW@1.8V	122mW@DV**
Area*	34.5mm ²	1.9mm ²	10.3mm ²
Energy*	11k	9.3	32.4

Table 6: Chip Statistics and Energy Comparison with TI TMS320C64x, CRISP and Diet SODA. *Area and energy are normalized to 90nm technology. **Diet SODA operates in DV mode - 1V and 600mV.

vices. In addition, as pre-/post-processing to improve image/video quality are becoming more important, flexibility is another important decision factor. To satisfy real-time constraints and programmability, three types of image/video processors have been used: SIMD, stream processors and reconfigurable processors.

SIMD-based processors such as SODA [11], NXP's EVP [13], Sandbridge's Sandblaster [12] and Icera's DXP [14], use multiple processing elements working in SIMD fashion to exploit high data level parallelism. These types of architectures usually support VLIW execution and use software-managed scratchpad memories to meet the real time constraints. Each SIMD processor includes special characteristics such as very wide SIMD width in SODA [11], deeply pipelined execution for chained operations in DXP [14], and multi-threading in Sandblaster [12]. Although many DLP-intensive DSC algorithms are efficiently implemented in SIMD manner [15, 16], SIMD-based processors usually suffer from large power consumption and hardware cost because of high bandwidth requirements. Diet SODA uses near-threshold operation to reduce energy consumption.

Stream processors such as Imagine [9] and SPI [10] have proved to be efficient solutions for media processing applications. Stream processors organize an application explicitly into streams of data and compute-intensive kernels. A host processor sends stream instructions to the processors and the arithmetic clusters in the processors operate in SIMD fashion. In addition, stream processors employ data locality and concurrency by compounding complex SIMD kernel computations to reduce the number of vector register read/write operations and power dissipation. Although existing stream processors achieve high performance for media applications, complex architectural components are an overkill for DSC applications.

Reconfigurable architectures can be classified into two types: coarse-grain and fine-grain. Coarse-grained reconfigurable architecture such as REMARC [17] have been used for media processing. In addition, ADRES [18] automatically maps applications onto coarse-grained reconfigurable arrays that are tightly coupled to VLIW processors and exploits loop-level and instruction-level parallelism to maximize functional units. XiSystem's XiRisc [19] is an example of a fine-grain reconfigurable architecture.

Diet SODA differs from all these architectures in that it operates in the near-threshold voltage region. This enables new memory system designs and radically reduced power consumption.

7. CONCLUSION

In this paper, we have proposed a programmable substrate for an ultra-low power signal processor using near-threshold operation. Near-threshold operation reduces energy but suffers from degraded performance, but this can be overcome by using parallelism. DSC algorithms on SIMD architectures offer an abundant amount of data level parallelism, forming a natural synergy with near-threshold operation. In addition, because memory systems operate at faster rates than SIMD data engines in near-threshold operation mode, scatter-gather prefetcher was introduced to exploit latency difference and lower instruction counts. Diet SODA also uses a dual voltage mode to increase performance for kernels that requires high processing power. Our results show that Diet SODA with a 128-lane SIMD unit operating at 600mV and 50MHz in an IBM 90nm technology can

meet the processing requirements of full-HD resolution at 30 fps while consuming only 122mW. This is on the order of 130x better performance and approximately 340x better energy efficiency over a DSP solution, and provides a more flexible solution than equivalently powered ASIC designs.

Although the near-threshold techniques bring a large opportunity for energy-efficient architecture designs like Diet SODA, they suffer from large delay variations due to increased process variability. Therefore, our next research steps are to assess the effects of variations in near-threshold operations on a SIMD architecture, and to explore architectural design spaces to tolerate the process variability.

Acknowledgment

Thanks to Yongjun Park for his help and feedback. This work was supported in part by NSF grants CSR 0910699 and CSR 0910851 and by ARM.

8. REFERENCES

- [1] J. C. Chen and S.-Y. Chien, "CRISP: Coarse-Grained Reconfigurable Image Stream Processor for Digital Still Cameras and Camcorders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1223-1236, Sep. 2008.
- [2] K. Illgner, et al., "Programmable DSP platform for digital still cameras," *Proceedings of IEEE International Conference on Acoustics, Speech, Signal Processing ICASSP '99*, vol. 4, pp. 2235-2238, Mar. 1999.
- [3] C. Chute, "Worldwide Digital Still Camera 2009-2013 Forecast," *IDC*, Apr. 2009.
- [4] H. Zen, et al., "A new digital signal processor for progressive scan CCD," *IEEE Transactions on Consumer Electronics*, vol. 4, no. 2, pp. 289-296, May 1998.
- [5] N. Nakano, et al., "Digital still camera system for megapixel CCD," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 2, pp. 289-296, May 1998.
- [6] D. Talla, et al., "Anatomy of a portable digital mediaprocessor," *IEEE Micro*, vol. 24, no. 2, pp. 32-39, Mar.-Apr. 2004.
- [7] S. Agarwala, et al., "A 600MHz VLIW DSP," *IEEE International Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002*, vol. 2, pp. 38-390, 2002.
- [8] Nanoscale Integration and Modeling (NIMO) Group, "Predictive technology model (PTM)," [Online]. Available: <http://www.eas.asu.edu/~ptm/>. [Accessed Nov. 16, 2009]
- [9] B. Khailany, et al., "Imagine: media processing with streams," *IEEE Micro*, vol. 21, no. 2, pp. 35-46, Mar./Apr., 2001.
- [10] B. Khailany, et al., "Programmable 512 GOPS Stream Processor for Signal, Image, and Video Processing," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 202-213, Jan. 2008.
- [11] Y. Lin, et al., "SODA: A low-power architecture for software radio," *33rd International Symposium on Computer Architecture, 2006. ISCA '06*, pp. 89-101, 2006.
- [12] J. Glossner, et al., "The sandbridge SDR communications platform," *Joint IST Workshop on Mobile Future, 2004 and the Symposium on Trends in Communications. SympoTIC '04*, pp. ii-ix, 24-26, Oct. 2004.
- [13] K. V. Berkel, et al., "Vector Processing as an Enabler for Software-Defined Radio in Handsets From 3G+WLAN Onwards," *EURASIP Journal on Applied Signal Processing*, pp. 2613-2625, Jan. 2005.
- [14] S. Knowles, "The SoC Future is Soft," *IEEE Cambridge Processor Seminar*, Dec. 2005.
- [15] O. Vermeulen, et al., "Ultra Fast Grey Scale Face Detection Using Vector SIMD Programming," *Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, 2007. SITIS '07*, pp. 585-592, 16-18, Dec. 2007.
- [16] C. Wu, et al., "Mapping Vision Algorithms on SIMD Architecture Smart Cameras," *First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDS '07*, pp. 27-34, 25-28, Sept. 2007.
- [17] T. Miyamori and K. Olukotun, "REMAR: Reconfigurable multimedia array coprocessor," *IEICE Trans. Inf. Syst.*, vol. E82-D, no. 2, pp. 389-397, 1999.
- [18] B. Mei, et al., "ADRES: An Architecture with Tightly Coupled VLIW Processor and Coarse-Grained Reconfigurable Matrix," *Lecture Notes in Computer Science*, vol. 2778/2003, pp. 61-70, Sep. 2003.
- [19] A. Cappelli, et al., "XiSystem: a XiRisc-based SoC with a reconfigurable IO module," *IEEE International Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005*, vol. 1 10-10, pp. 196-193, Feb. 2005.
- [20] B. Zhai, et al., "Energy efficient near-threshold chip multi-processing," in *Proc. of the ACM/IEEE International Symposium on Low-Power Electronics Design*, pp. 32-37, 2007.
- [21] B. Bayer, "Color Imaging Array," *U.S. Patent 3 971 065*, Jul. 1976.
- [22] D. Philips, "Image Processing in C", *R&D Publications Inc.*, 1994.
- [23] A. Wang and A. Chandrakasan, "A 180mV FFT processor using subthreshold circuit techniques," in *Proc. of the IEEE International Solid-State Circuits Conference*, pp. 292-299, 2004.