From *Xetal-II* to *Xetal-Pro*: On the Road Towards An Ultra Low-Energy and High Throughput SIMD Processor

Yu Pu, *Member, IEEE,* Yifan He, *Member, IEEE,* Zhenyu Ye, *Student Member, IEEE,* Sebastian Moreno Londono, *Student Member, IEEE,* Anteneh Alemu Abbo, *Member, IEEE,* Richard Kleihorst, *Member, IEEE,* and Henk Corporaal, *Member, IEEE*

***Abstract*—Looking forward to the next generation of mobile streaming computing, the demanded energy efficiency of end-user terminals will become ever stringent. The *Xetal-Pro* processor, which is the latest member of the *Xetal* low-power single-instruction multiple-data (SIMD) processor family from Philips, is presented in this paper. The predecessor of *Xetal-Pro*, known as *Xetal-II*, already ranks as one of the most computational-efficient (in terms of GOPS/Watt) processors available today, however it cannot yet achieve the demanded energy efficiency (less than 1 pJ per operation). Unlike *Xetal-II*, *Xetal-Pro* supports ultra-wide supply voltage (V_{dd}) scaling from the nominal supply to the sub-threshold region. Although aggressive V_{dd} scaling causes severe throughput degradation, this can be partly compensated for by the massive parallelism in the *Xetal* family. *Xetal-II* includes a large on-chip frame memory (FM), which cannot be scaled well to an ultra low V_{dd} hence creating a big obstacle to increase energy efficiency. Therefore, we investigate both different FM realizations and memory organization alternatives. A hybrid memory system (HMS), which reduces the non-local memory traffic and enables further V_{dd} scaling, is proposed. For design space exploration of the right number of the scratch-pad memory (SM) entries, the corresponding data locality analysis is provided, too. Moreover, some unique circuit implementation issues of *Xetal-Pro* such as the customized level-shifter are also discussed. Compared to *Xetal-II* operating at the nominal voltage, *Xetal-Pro* provides up to two times energy efficiency improvement even without V_{dd} scaling (essentially a consequence of data localization in the SM) when delivering the same amount of ultra high throughput. With V_{dd} scaling into the sub/near threshold region, *Xetal-Pro* could gain more than ten times energy reduction while still delivering a high throughput of 0.69 GOPS (counting multiply and add operations only). The new insight of *Xetal-Pro* sheds light on the direction of future ultra-low energy SIMD processors.**

***Index Terms*—Xetal, SIMD, hybrid memory system, ultra low-energy, sub/near threshold**

This article has been partially presented at the 47th Annual ACM IEEE Design Automation Conference [1].

Yu Pu is with the University of Tokyo, Japan (email: ypu@iis.u-tokyo.ac.jp). Yifan He, Zhenyu Ye, Sebastian Moreno Londono, and Henk Corporaal are with the Eindhoven University of Technology, the Netherlands (email: {y.he, z.ye, s.moreno, h.corporaal}@tue.nl). Anteneh Alemu Abbo is with the Philips Research, the Netherlands (email: anteneh.a.abbo@philips.com). Richard Kleihorst is with VITO, Belgium (email: richard.kleihorst@vito.be).

I. Introduction

THE latest communication and multimedia standards, such as 4G wireless communication, H.264 and high-definition video, require ultra-high computational performance and ultra-high energy efficiency of end-user devices. While processors like Intel's *Pentium M* and IBM's *Cell* [2] provide excellent computational performance, their energy consumption exceeds far beyond the energy budget of mobile terminals. Instead of these high-end processors, domain-specific streaming processors, particularly massively-parallel Single Instruction Multiple Data (SIMD) processors, are very popular candidates for SoCs within mobile devices. This is because: (*i*) massive parallelism in streaming algorithms typically shows up as data-level parallelism (DLP) which can be inherently exploited by SIMD architectures, thus making SIMD the most common core execution engine on a stream platform. (*ii*) SIMD is a low power architecture as it applies the same instructions to all processing elements (PEs). However, the practice today is that the embedded streaming processor in a cellular phone consumes tens of pJ per operation (pJ/op) and the limited battery capacity is only sufficient for playing video applications for a few hours. Meanwhile, the large power dissipation also worsens the SoCs' thermal and reliability issue, thus requiring expensive cooling techniques. The semiconductor industry hopes that the energy for consumer electronics can be reduced by five to ten times in the next five years. This paper presents our progress in developing the *Xetal-Pro* processor,

which is the latest member of the *Xetal* SIMD processor family from Philips and delivers a significantly improved computational efficiency. Please be aware that, instead of power reduction, *Xetal-Pro* focuses on energy reduction, as energy/operation is the real metric for battery life.

The development of *Xetal-Pro* starts from exploring its predecessor *Xetal-II* [3]. *Xetal-II* has been implemented in a 90 nm CMOS process with 74 mm^2 die area. It consists of 320 PEs, and delivers a peak performance of 107 GOPS on 16-bit data at a running frequency of 84 MHz, with a power budget of 600 mW. Although *Xetal-II* already ranks as one of the most computational-efficient (in terms of GOPS/Watt) processors available today, it still cannot suffice the demanded computational efficiency for emerging mobile computing applications. Compared to *Xetal-II*, *Xetal-Pro* has the following key improvements:

- *Xetal-Pro* supports ultra-wide V_{dd} scaling from a nominal supply to sub/near threshold region. Although aggressive V_{dd} scaling causes throughput degradation, fortunately, the massively-parallel nature of *Xetal-Pro* can partly compensate such degradation. Even operating in the sub/near threshold mode, *Xetal-Pro* can still render a reasonably high throughput that is enough for many low/medium level streaming applications.
- *Xetal-II* utilizes a large SRAM based on-chip frame memory (FM) of 10 Mbit, which allows on-chip storage of multiple VGA frames. While this feature dramatically reduces off-chip traffic and helps enhancing performance and energy efficiency, it also creates a big problem when applying aggressive V_{dd} scaling because typically commercial SRAM cannot operate reliably below 0.7 V [4]. As a result, the SRAM becomes the system energy bottleneck. Alternative realizations exist, like the low-power SRAM cells from MIT [4] [5], or using standard cell memory logic. However, our analysis shows that these alternatives are not effective and not efficient enough for the large on-chip FM. To address this issue, we propose a hybrid memory system (HMS). The "hybrid" implies twofold meanings: (*i*) A hybrid memory architecture, which consists of an accumulator (ACCU) register, a scratchpad memory (SM) and the FM; (*ii*) A hybrid realization using sub-threshold SM in combination with super-threshold FM. A corresponding data locality analysis of the HMS is also provided in this paper. The proposed HMS provides up to two times energy efficiency improvement even at nominal operating condition, essentially a consequence of data localization in the scratch-pad memories.

Three representative kernels, which are typical benchmarks for SIMD architectures [3] [6], are used to examine the *Xetal-Pro* system. These kernels include: (*i*) $N\times N$ non-separable filter (*ii*) $N\times N$ separable filter (*iii*) YCbCr to RGB color-space conversion. We compare the energy consumption of applying each kernel on a VGA frame. The aforementioned new features of *Xetal-Pro* bring a total energy reduction of up to ten times compared to *Xetal-II*. This is achieved when *Xetal-Pro* runs at about 0.4 V. Even at this low-voltage mode, *Xetal-Pro* can still deliver a performance of 0.69 GOPS (counting multiply and add operations only). The Intrinsic Computational Efficiency (ICE) graph in Figure 1 highlights the energy efficiency advantage of *Xetal-Pro* over that of earlier well-known works[1]. Table I summarizes the references for the ICE graph. In general, *Xetal-Pro* gives very new insights on how to design an ultra low-energy SIMD processors, which is suitable for future mobile streaming systems.

The remainder of this paper is organized as follows. Section II presents the related work of *Xetal-Pro*. In Section III, we give an overview of *Xetal-II* and analyze its energy and performance by mapping different benchmark kernels. The energy breakdown of *Xetal-II* clearly shows that the FM energy dominates the total energy. Section IV presents the big challenge of applying ultra-low V_{dd} scaling to the FM implemented with commercial SRAM. Following which, in Section V, we explore alternative V_{dd} scalable FM. Unfortunately, these alternatives are also not effective and not efficient enough in lowering the energy of FM. To resolve this issue, the HMS is introduced in Section VI. Section VII briefly discusses the possible approaches to enhance the yield of *Xetal-Pro* under large process and design variabilities. Finally, Section VIII draws conclusions of this work.

II. Related Work

In the following, the related work is categorized into three subsections: *(A)* sub-threshold designs; *(B)* scratch-pad memory; and *(C)* SIMD processors.

A. Sub-threshold Designs

An emerging trend for lowering energy of digital processors is to scale V_{dd} to the sub/near threshold region, which brings not only quadratic dynamic power

[1]In the ICE curve, only programmable multiply and add operations are counted. Other operations, e.g., shift, dedicated adder tree, etc. are not counted. The energy of 8-bit and 16-bit operations are linearly scaled to 32-bit operations for a fair comparison.

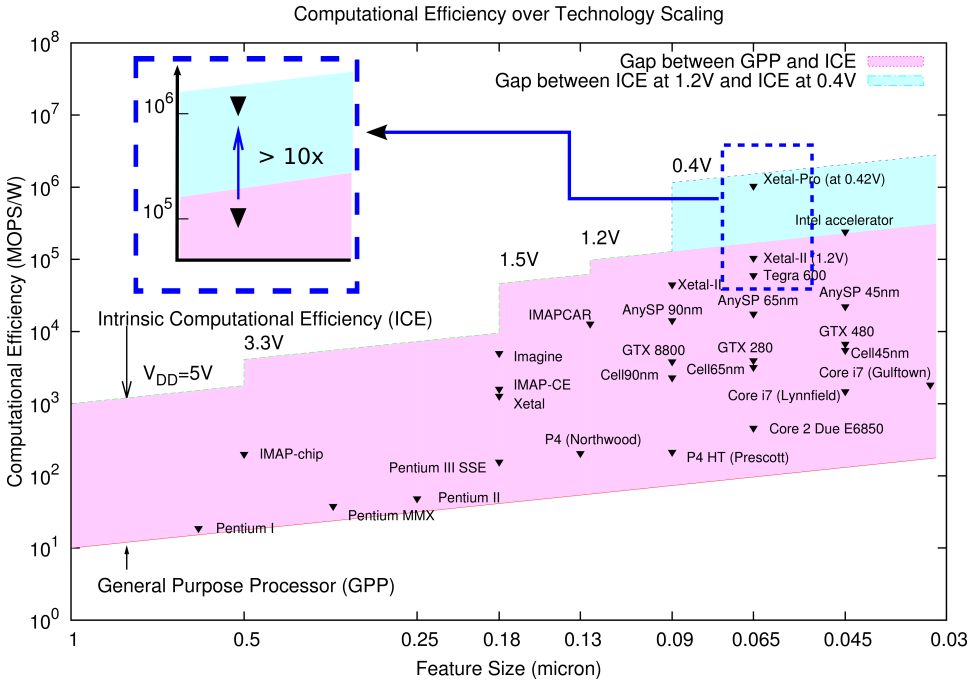


Fig. 1. The ICE graph annotates the energy efficiency of *Xetal-Pro* and earlier well-known works

TABLE I
REFERENCES FOR THE ICE CURVE

Processors	Intel accelerator [7]	IMAP-chip [8]	IMAP-CE [9]	IMAPCAR [6]	Xetal [10]	Xetal-II [3]
GOPS/W	240	0.2	1.8	12.8	1.6	44.6
Processors	Intel CPUs [11]	Imagine [12]	Cell 90nm [2]	Cell 65nm [13]	Cell 45nm [14]	Tegra 600 [15]
GOPS/W	annotated in Fig 1	5	2.3	3.2	5.5	60
Processors	AnySP 90nm [16]	AnySP 65nm [16]	AnySP 45nm [16]	GTX 8800 [17]	GTX 280 [17]	GTX 480 [17]
GOPS/W	14	18	22	3.8	3.9	6.7

savings, but also super-linearly reduced leakage current. Many prototype chips, which can function in the sub-threshold region, have been presented in recent years. These chips include a 180 mV FFT processor in 180 nm CMOS process [18], a 256 Kbit 10-T dual-port SRAM in 65 nm CMOS process [5], which was later improved to 8-T dual-port SRAM [4]. In [19] a single-end sub-threshold SRAM has been developed for extremely low speed applications. A 130 nm and a 180 nm CMOS sensor node processors are presented in [20] and [21], respectively. A TI-MSP430 based digital signal processor (DSP) with integrated DC-DC converter in 65 nm CMOS is presented in [22]. In [23] we presented the *SubJPEG* prototype chip, a 65 nm CMOS 8-bit JPEG co-processor. *SubJPEG* is equipped with four parallel DCT-Quantization engines and delivers 15 fps VGA processing at about 0.4 V. In 2009 Intel also announced its 45 nm CMOS sub-300 mV 4-Way sub-word parallel processors [7].

B. Scratchpad Memory

Using scratchpad memories can help reduce the traffic to higher memory levels significantly when applications show substantial locality [24]. For example, a stream register file (or memory) as used in the *Imagine* architecture [25] can provide high performance with low energy consumption for streaming applications.

C. SIMD Processors

Other than *Xetal-II*, *IMAPCAR* [6] from NEC is another very successful SIMD processor. It includes 128 PEs and each PE is a 4-way 16-bit VLIW with its own 2 KB on-chip memory. It achieves 100 GOPS within a power budget of 2 W. The *IMAPCAR* differs from *Xetal-II* in the VLIW PEs, the per-PE register files, and the index addressing to on-chip memory. Compared to *Xetal-II*, the indirect addressing capability of *IMAPCAR* [6] enables access to different pixel locations by PEs. While this feature facilitates parallelization of some image tasks containing irregular memory access, it leads

to increased energy consumption for most applications with predominantly regular memory accesses. Its successor, *IMAPCAR-II* [26], added the support of switching between SIMD and MIMD mode, which however is out of the scope of this paper. The recently published *AnySP* [16] architecture also proposes a configurable SIMD datapath as the core execution engine in the stream platform. In the annotated ICE graph (Figure 1), GPUs like nVIDIA GTX8800 have different application domains. It is worth mentioning that sub-word parallel processors [7] also benefit from exploiting SIMD parallelism. However, they are not massively-parallel processors for very low-energy applications.

However, to the best of our knowledge, no previous work has analyzed the impact of aggressive V_{dd} scaling on the memory hierarchy in the context of an ultra-low energy massively-parallel SIMD. The comparison between this work and previous works is summarized in Table II.

TABLE II
COMPARISON BETWEEN THIS WORK AND PREVIOUS WORKS

References	Sub-threshold design	Short SIMD	Wide SIMD	Scratchpad memory
[4], [5], [18] – [23]	√			
[7]	√	√		
[24]				√
[6], [16], [25], [26]			√	√
This work	√		√	√

III. EXPLORATION OF *Xetal-II*

As the starting line to the development of *Xetal-Pro*, *Xetal-II*'s architecture, performance and energy breakdown have been carefully analyzed. The detailed energy breakdown of *Xetal-II* is presented. Being the latest derivative of the *Xetal* family, *Xetal-Pro* inherits many low-power peculiarities of the *Xetal-II* processor while removing serious shortcomings that may result in a sub-optimal energy efficiency.

A. Overview of Xetal-II Processor Architecture

The block diagram of the *Xetal-II* processor is depicted in Figure 2(a). The control processor (CP) is a 16-bit, microprocessor without interlocked pipeline stage (MIPS) like processor. The main task of the CP is to control the program flow, handle interrupts, communicate with the outside world and configure other blocks. The linear processor array contains 320 PEs and an integral 10 Mbit FM. Layout and memory considerations necessitate partitioning of the linear processor array into tiles because: (*i*) grouping all PEs and FM into one tile would result in a poor global layout with a very strange aspect-ratio and correspondingly large area. (*ii*) commercial memory generators have limited maximum word width, which also disables this option. However, on the other extreme, using only one PE plus FM per tile results in too many memories with corresponding addressing overhead and global/semi-global wiring overhead, in spite of the fact that it may provide advantages in programming flexibility for certain kernels/applications [27].

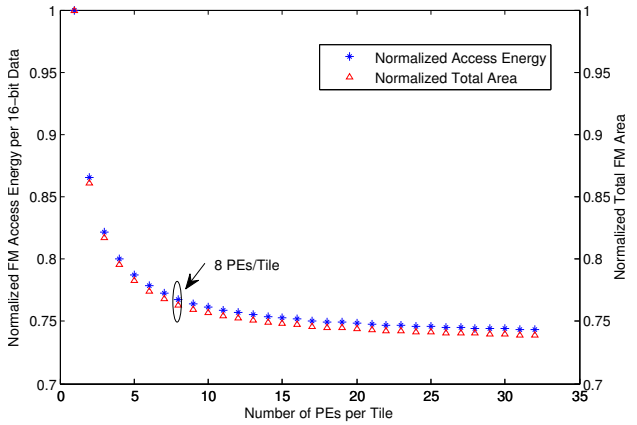


Fig. 3. Number of PEs per tile vs. normalized FM access energy per 16-bit data and normalized total FM area

Apart from silicon area, our primary concern is energy consumption. The metric we used to decide the optimal number of PEs per tile is the energy/area efficiency of the shared FM. Different physical partitions affect both total area and energy consumption per unit data. Figure 3 shows the normalized FM energy per 16-bit data access and normalized total FM area under different partitions. We can see that including 8 PEs (power of two) per tile (thus, 40 tiles in total) is a good choice considering FM access energy efficiency, FM total area efficiency and practical layout constraints.

Each PE has a two-stage pipeline and shares the instruction fetch and decode stage of the CP. Figure 2(b) shows the structure of the 16-bit PE, which is equipped with a local register (ACCU) for immediate result feedback and a flag register (FLAG) for guarded instruction execution. Each PE supports 16-bit ADD/SUB, MUL, MAC, logical operations, which can further be compounded with other operations (e.g., absolute, negative, etc.). All instructions are executed in a single cycle. The FM consists of 40 SRAM modules (each 128bit$\times$2048[2]) with a pseudo-dual port interface to provide single cycle read and write accesses. This data memory can store both the frame data and the

[2]One SRAM module per tile. Since each tile consists of eight 16-bit PEs, the data width of the SRAM module is 128 bit.

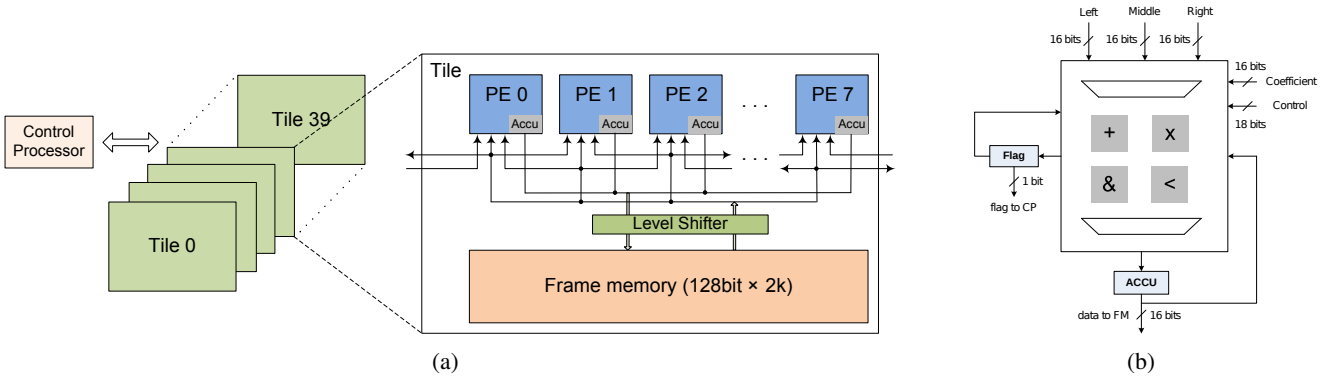


Fig. 2. (a) Block Diagram of *Xetal-II* Architecture; (b) Structure of the 16-bit PE

intermediate results. The relatively large capacity of the FM allows on-chip storage of multiple VGA frames or images with higher resolution, reducing in this way the off-chip traffic. The communication network between the FM and PEs enables PEs to directly access the memory (FM) data of its left and right neighbors. To provide better control of V_{dd} scaling, the tile is divided into logic and memory voltage domains, coupled with level-shifters. For simplicity, in the following sections of this paper, PEs is used to refer to the logic part, including processing elements and communication network of the tile; FM is used to refer to the memory part of the tile.

B. Energy/Performance Analysis of Xetal-II

Xetal-Pro is designed in a 65 nm CMOS process. As a reference for *Xetal-Pro*, we migrated the *Xetal-II* processor from 90 nm to 65 nm technology. The logic part was synthesized with TSMC 65 nm Low-power (LP) SV_t CMOS digital standard cell library. LP process is superior over general-purpose (GP) process for medium/low end SoCs, because the LP feature can make leakage energy one to two orders of magnitude lower than with the GP feature. The V_t of our process is about 0.41 to 0.42 V[3]. The SRAM was synthesized with a well-known commercial low-power memory generator, which uses minimum size and HV_t devices of the same process technology for bit-cells to further constrain leakage energy without violating timing constraints. The impact of the long global wires for decoded instruction plus intermediate repeaters have been considered based on post-layout analog simulation results. The whole *Xetal-II* system can run at 125 MHz with 1.2 V voltage supply at 25 °C room temperature, and offers 80 GOPS throughput (320 PEs in total, two operations per cycle per PE, and counting multiply and add operations only) with each PE processing one inst/cycle. The critical path is the FM read access plus the MAC operation within the PE.

To analyze the system energy breakdown, we use three representative application kernels which are typical benchmarks for SIMD processors. These kernels are: (*i*) $N\times N$ non-separable filter, (*ii*) $N\times N$ separable filter, and (*iii*) YCbCr to RGB color-space conversion. Besides the popularity of these kernels, the other major reason for choosing them as our benchmark is that they show up quite different data locality characteristics. The data in an $N\times N$ non-separable filter can be reused N^2 times while the data in an $N\times N$ separable filter is only reused $2N$ times. YCbCr-RGB conversion is a pixel-to-pixel operation, so there is no data sharing between pixels.

The mapping of three kernels on the reference *Xetal-II* processor is shown in Figure 4. Due to page limit, only the mapping of non-separable filer kernel is described here, in Algorithm 1. The mapping of other two kernels can be described similarly. We take VGA (640×480 pixels) image with interleaving factor of two as an example[4]. In the case of color conversion, the Y, Cb, and Cr values of a pixel are assumed to be stored in consecutive rows in the FM. Each PE can read the memory on its left (mem.l) and right (mem.r). The image height is represented by H (H is equal to 480 for VGA format).

Table III summarizes the energy breakdown of the reference *Xetal-II* processor when running the three benchmark kernels. It is worth mentioning that in Table III, the global wires do not include the intra-tile part, which is already included in the PEs. The summation of energy consumption percentage of both intra and inter-tile global wires takes around 5%-7% for the three kernels.

[3]This V_t is according to the foundry definition. However, the actual point between the exponential region and the linear region of this process is around 650mV according to device simulations.

[4]With interleaving factor of two, one image line is stored in two rows of the frame memory. Pixels at the odd (even) columns of the image line are stored in the odd (even) rows of the frame memory.

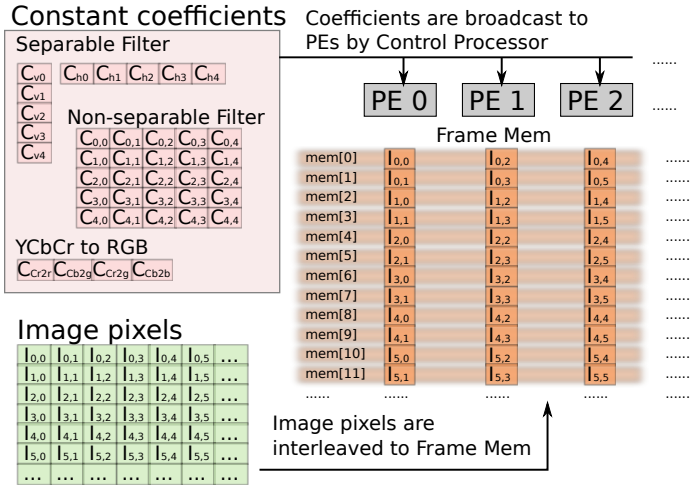


Fig. 4. Mapping of YCbCr-to-RGB, non-separable filter and separable filter on the baseline architecture.

Algorithm 1: A 5×5 non-separable filter kernel mapped on the baseline architecture. The mem.l and mem.r represent the memory of a PE's left neighbor and right neighbor respectively. The image height, H, is equal to 480 for VGA format.

```
for i = 2 to (H − 3) do
    accu ← C0,0 × mem.l[2i-4];
    accu ← accu + C0,1 × mem.l[2i-3];
    accu ← accu + C0,2 × mem[2i-4];
    accu ← accu + C0,3 × mem[2i-3];
    accu ← accu + C0,4 × mem.r[2i-4];
    ... // other accu for output at mem[2H+2i]
    accu ← accu + C4,0 × mem.l[2i+4];
    accu ← accu + C4,1 × mem.l[2i+5];
    accu ← accu + C4,2 × mem[2i+4];
    accu ← accu + C4,3 × mem[2i+5];
    mem[2H+2i] ← accu + C4,4 × mem.r[2i+4];
    ... // accu for output at mem[2H+2i+1]
    mem[2H+2i+1] ← accu + C4,4 × mem.r[2i+5];
end
```

For the three kernels, the total energy is dominated by the energy of the tiles (i.e., PEs and FM). Compared with the 40 tiles, the CP and the global decoded instruction wires consume much less energy. Therefore, to effectively reduce the total energy, the tiles are the focus of our further exploration.

IV. Challenge of Ultra-Wide-Range V_{dd} Scaling

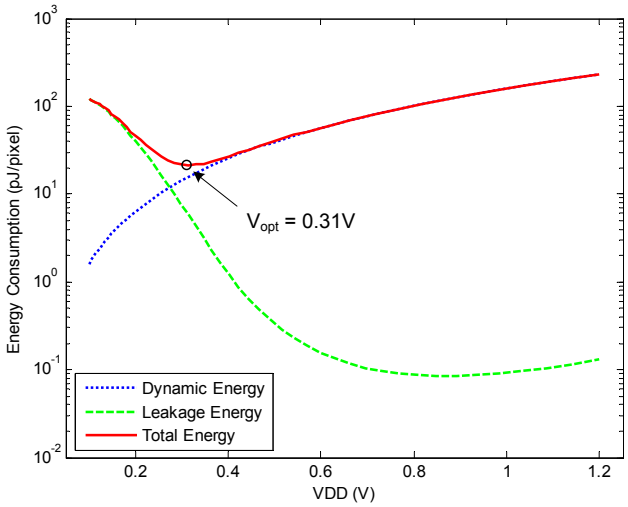
V_{dd} scaling is one of the most effective means to bring quadratic dynamic energy savings to standard-cell based logic, i.e., $E_{logic} \propto C_{load}V_{dd}^2$, where C_{load} is the loading capacitances including both gate and interconnection wire capacitances. The V_{dd} scaling range of commercial processors is normally limited to about 2/3 of nominal supply due to two fundamental problems at an ultra-low V_{dd}: (*i*) high yield loss in the presence of process variations (*ii*) severe throughput degradation. To mitigate the first problem, the physical design techniques, which we have developed in the *SubJPEG* processor [23] to improve sub-threshold logic's yield, are applied to *Xetal-Pro*. To solve the second problem, the nature of the massive parallelism of the *Xetal* family can be used to compensate the throughput degradation, as will be discussed soon.

TABLE III
Energy breakdown of the *Xetal-II* processor at 1.2 V for three benchmark kernels

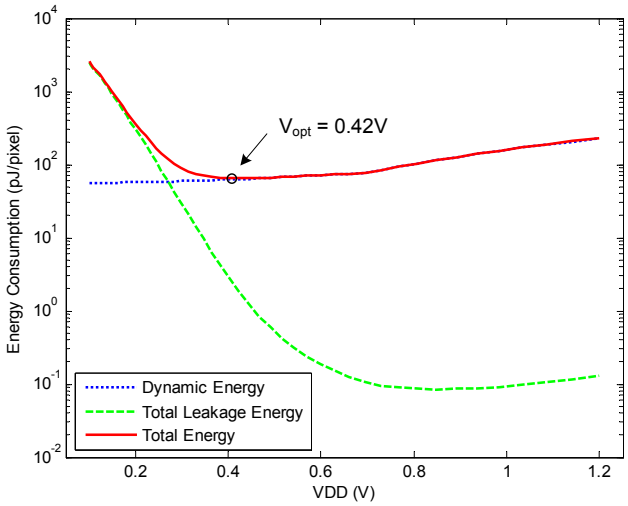
Benchmark	PEs	FM	CP	Global Wires	Total (pJ/pixel)
5×5 non-separable filter	26.0%	68.9%	3.7%	1.4%	240.8
5×5 separable filter	23.5%	71.9%	3.3%	1.3%	106.5
YCbCr to RGB	14.7%	81.3%	2.9%	1.1%	109.9

Compared to pure logic, V_{dd} scaling is even more difficult when applied to SRAM. First, the rapidly deteriorating read/hold static noise margin (SNM) of bit-cells causes severe reliability issues. A very small amount of injected noise can cause the bit-cell's state to flip. Thus, all commercial SRAMs achieving high density strictly prohibit operating below 0.7 V. Second, SRAM's energy cannot scale quadratically with V_{dd}. SRAM bit-cells' energy, which usually dominates total SRAM's energy, can be approximated as $E_{bitcell} \propto C_{bitline}V_{dd}V_{swing}$ in a single cycle. $C_{bitline}$ is the loading capacitance on a SRAM bitline. V_{swing} is the bitline swing, which must exceed a minimum magnitude required by sense-amplifiers to make correct decoding. V_{swing} cannot scale linearly with V_{dd}. Therefore, both bit-cells' energy and total SRAM's energy only scale sub-quadratically with V_{dd}, while the energy of other SRAM components like sense-amplifiers, wordline and bitline drivers, address decoders can scale as well as logic. Third, SRAM's speed degrades even faster with V_{dd} scaling, compared to that of pure logic. This implies that SRAM becomes the system performance bottleneck if both SRAM and logic scale to the same ultra-low V_{dd}.

Assume that ultra-wide-range V_{dd} scaling is applied to the most energy-consuming part, i.e., the tile. While V_{dd} scaling lowers the dynamic energy, the leakage energy increases due to a prolonged cycle time. As a result, there exists an energy-optimal V_{dd} point where the total energy is minimal. Pursuing a lower V_{dd} than this optimal V_{dd} point makes leakage energy dominate total energy hence rendering no additional energy benefits. As an example, the energy consumption of processing one pixel when applying the aforementioned 5×5 non-separable filter kernel (25 instructions in total) is used as comparison metric. Figure 5(a) depicts the energy consumption curve



(a)



(b)

Fig. 5. V_{dd} vs. energy consumption when processing one pixel with a 5×5 non-separable filter kernel (a) assuming ideal SRAM voltage scaling; (b) SRAM only scales down to 0.7 V

under different supply voltages. The power and delay of standard cells are characterized with the recently released Penn State-Philip (PSP) transistor model from Philips, which has superior accuracy over the BSIM4 transistor model when modeling low-voltage circuits. Note that here we unrealistically assume that the SRAM can be scaled to sub-threshold as well as the standard cells, just to show the lower bound of energy reduction by V_{dd} scaling. The optimal point in this case occurs at V_{dd} = 0.31 V. At this point, the tile consumes 21.4 pJ/pixel, leading to a ten times reduction of the energy consumption ideally achievable, compared to operating at nominal 1.2 V.

However, with V_{dd} scaling, the maximum frequency each PE can achieve also decreases dramatically hence causing severe performance degradation, as shown by

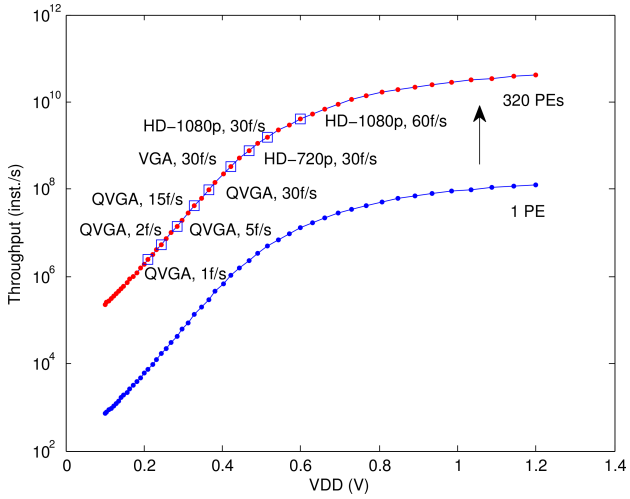


Fig. 6. Impact of V_{dd} scaling on system throughput of 1 PE (lower curve) and 320 PEs (upper curve). The blue squares on the upper curve indicate the supported resolution and frame rate with 320 PEs when executing a 5×5 non-separable filter kernel

the lower curve of Figure 6. Fortunately, with 320 PEs processing in parallel, this performance loss can be largely compensated. This shows the unique advantage of massively-parallel SIMD architecture to outperform other processor architectures in energy efficiency and throughput. The upper curve of Figure 6 depicts the supported resolution and frame rate at different V_{dd} when running the 5×5 non-separable filter kernel by 320 PEs. Above 0.6 V and above 0.42 V, HD-1080p (1920×1080) 60 frames/s and VGA (640×480) 30 frames/s can be supported in real time respectively. Even when V_{dd} goes down to about 0.33 V, we can still run many low-end applications, such as QVGA at 15 frames/s[5].

Figure 5(a) presents only the ideal lower energy consumption bound of the reference processor. Because commercial SRAM's V_{dd} cannot scale well to below 0.7 V, Figure 5(b) shows the practical V_{dd} scaling result when SRAM only scales to 0.7 V. The minimal energy consumption (65.1 pJ/pixel) is obtained when the logic part is scaled to 0.42 V. Compared to operating at the nominal voltage supply, the energy reduction is only a factor of 3.5, far behind the ten times ideally achievable reduction. It should be noted that here about 88% of the total energy is consumed by the FM.

The tile energy consumption at different V_{dd} is compared in Figure 7. We can see that even when PEs' V_{dd} is aggressively scaled to sub/near threshold, it only reduces

[5]As indicated in Algorithm 1, it requires 25 instructions to implement the 5×5 non-separable filter kernel on VGA resolution or higher (interleaving factor≥2). However, QVGA format requires 5 additional instructions, as not all of the 5×5 pixels are directly accessible.

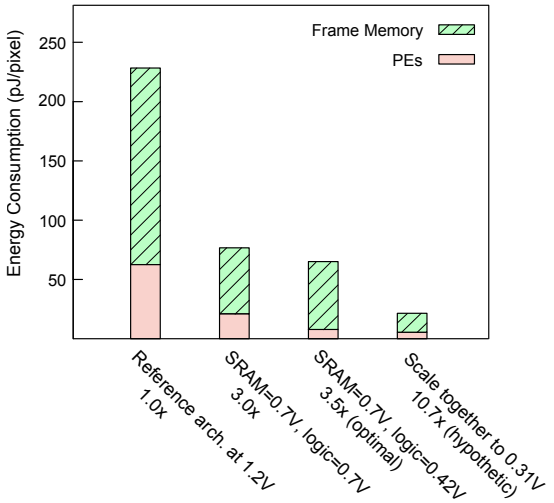


Fig. 7. Tile (*Xetal-II* Reference Processor) Energy Consumption for different V_{dd}

an extra 15% of the energy compared to that when both PEs and SRAM are supplied at 0.7 V. Thus, unless the FM can also scale further, it does not make too much sense to aggressively scale the V_{dd} of standard-cell (PEs) part due to the low energy gain and high performance loss. This conclusion holds true for other kernels.

V. Exploration of V_{dd} Scalable FM

As shown from the above analysis, commercial SRAM module creates a big obstacle for V_{dd} scaling. To resolve this challenge and to further reduce the total energy consumption of the *Xetal-II* SIMD processor, one potential solution is to look for a V_{dd} scalable FM. Recent MIT low-power dual-port SRAM [4] [5] and the standard-cell synthesized memory are two possible choices. The MIT work achieves ultra-low V_{dd} operation by adding extra devices within the bit-cell. The standard-cell based memory can also approach ultra-low V_{dd} because: (*i*) they are not limited by density constraint and composition style, so transistor up-sizing, buffer insertion and logic re-construction (which optimizes boolean expressions) can be used freely during synthesis. (*ii*) they can employ hierarchical topology, which prevents high fan-out and relieves shared architecture.

The V_{dd} of MIT 10-T low-power SRAM can be scaled to below 0.4 V. However, it has several drawbacks in our case. First, it occupies 66% more cell area compared to the commercial differential 6-T SRAM [5]. When the FM is implemented with 6-T commercial SRAM, the ratio between SRAM bit-cell array area and SRAM total area is 7/10. If this FM is realized by the 10-T SRAM, more than 30% additional area overhead will be imposed to each tile. Second, it consumes more access energy at nominal voltage. Besides, the high leakage power (about 100 μW at 1.2 V) also prevents it from scaling to very low V_{dd}, as the leakage energy increase will quickly counteract the reduction of the dynamic energy. Table IV presents the energy consumption when FM is realized by the MIT 10-T SRAM, in comparison with commercial SRAM realization for FM. Third, the much lower speed of the MIT SRAM is quite severe. The reported maximal speed is 2.5 times slower than the commercial 6-T SRAM with the same word width and depth that we are using. This severely degrades the performance at both nominal and scaled voltage. The maximum energy gain it can reach is rather small in contrast to its high area, performance and reliability overhead. So we conclude that, the MIT 10-T memory is not applicable in our case. These problems also exist for other sub-threshold SRAM works [4] [19].

The standard-cell realization of large on-chip SRAM is also not applicable. According to our synthesis result, although it can be faster than commercial SRAM, it consumes even more energy and area than the MIT 10-T SRAM at nominal voltage. This limits the standard-cell based memory designs to only very small arrays. Therefore, to reach our goals of ultra-low-energy, ultra-wide-voltage-range, and medium-to-high-throughput SIMD processor, architecture improvements are required.

VI. Hybrid Memory System

Since V_{dd} scalable FM is not applicable in our *Xetal-Pro*, in this section we propose a hybrid memory system (HMS) to exploit the often available data locality and reduce the non-local memory traffic and to enable further V_{dd} scaling. The "hybrid" implies two things: (*i*) a hybrid memory architecture consisting of an ACCU register, a scratchpad memory (SM), and the FM; (*ii*) a hybrid realization consisting of sub-threshold SM and super-threshold FM.

A. The HMS Scheme

The HMS is shown in Figure 8. Within the proposed HMS, we have three types of characterized memories to hold the data: (*i*) ACCU register for short-term data storage; (*ii*) SM for intermediate-term data storage; and (*iii*) FM for long-term data storage. Both the FM and the SM are directly accessible by the PE to provide the source/destination operands, which means that they are at the same memory hierarchy. This design choice not only increases the flexibility of memory access, but also reduces the penalty when few data locality can be exploited by the SM. Compared to FM, SM consumes much less energy per access due to its much smaller

TABLE IV
TILE (REFERENCE PROCESSOR) ENERGY CONSUMPTION WITH MIT 10-T SRAM REALIZATION FOR FM.

Benchmarks	pJ/pixel at 1.2 V	Compare [a]	pJ/pixel after optimal V_{dd} scaling [b]	Compare [c]
5×5 non-separable filter	265.0	16.0%↑	49.6	1.3×↓
5×5 separable filter	118.3	16.0%↑	21.4	1.4×↓
YCbCr to RGB	124.5	18.0%↑	21.1	1.5×↓

[a]Compare to the energy consumption with commercial SRAM realization for FM (at 1.2V).
[b]FM and PEs are scaled to different sub/near threshold voltages, to reach an optimal combination for energy efficiency.
[c]Compare to the energy consumption with commercial SRAM realization for FM (after optimal scaling).

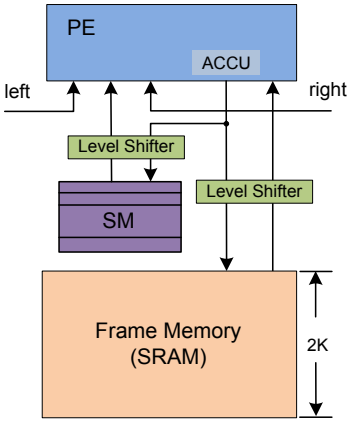


Fig. 8. Proposed Hybrid Memory Architecture (HMS)

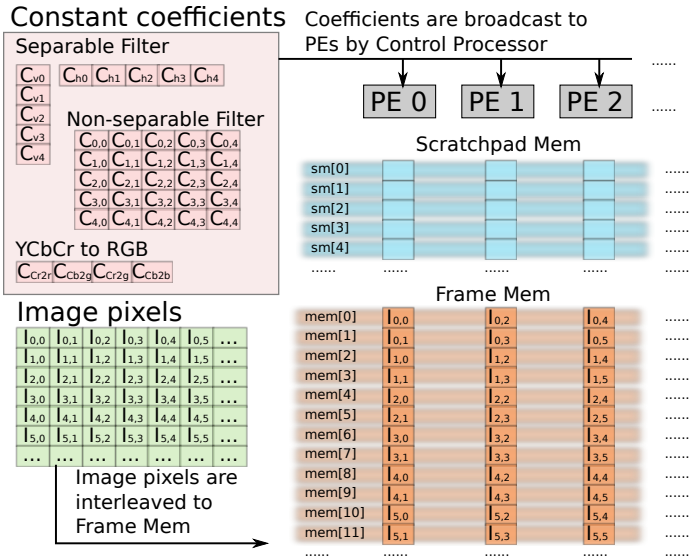


Fig. 9. Mapping of YCbCr-to-RGB, non-separable filter and separable filter on the proposed architecture.

size. It is worth mentioning that the SM supports all the addressing modes of FM, which makes it very friendly to access. For the low-level image/video processing (target domain of SIMD), most applications contain spatial data locality. When no data locality is exploitable, the SM can be bypassed and clock-gated with only a few μW leakage overhead. It should be noted that the critical path of the system is also hardly changed, i.e., FM read access plus PE operation. In addition, when coupled with index addressing, the SM can also be used as a look-up table for index based algorithms.

The SM in *Xetal-Pro* is decided to be dual-ported with 128-bit word width and 32 entries. The reasons that we choose this relatively large number of entries are (*i*) to enable more applications with large working windows (e.g., motion estimation, etc.) or higher resolutions (>VGA) to fully exploit their data locality; (*ii*) to demonstrate that even with such a (relatively) large size, we can still reach more than ten times energy gain. In Section VI-D, we will further justify this choice in details. The 32-entry SM (commercial SRAM realization) adds about 15% area to the tile. Fewer entries can slightly reduce the area overhead and energy consumption, but fewer applications can benefit from this HMA. The programming model of the proposed architecture is also slightly different since there is an extra memory (SM) to utilize. The mappings of the three kernels on *Xetal-Pro* are shown in Figure 9. The mapping of non-separable filter kernel on the architecture with HMS is shown in Algorithm 2. The mapping of other two kernels can be described similarly.

B. Instruction Set Extension

Compared to *Xetal-II*, the instruction format of *Xetal-Pro* is almost the same because the SM has the same addressing modes as the FM and they are mapped to a continuous memory space. However, since the source operand can be read from and the result can be sent to one extra location (SM), the total number of instruction types increases from user's point of view. By categorizing the instructions based on (*i*) what the data source (src) and destination (dest) are; (*ii*) if data is operated (OP) or only moved (MV) to a different location, the main difference of PE instructions between the two architectures is described in Table V.

TABLE V
MAIN DIFFERENCES OF PE INSTRUCTIONS BETWEEN *Xetal-II* AND *Xetal-Pro* (INSTRUCTION FORMAT: OPERATION DEST, SRC1, SRC2)

No.	*Xetal-II*				*Xetal-Pro*			
	op	*dest*	*src1*	*src2*	*op*	*dest*	*src1*	*src2*
1	*OP*	ACCU	FM	COEF / ACCU	*OP*	ACCU	SM	COEF / ACCU
2	*OP*	FM & ACCU	FM	COEF / ACCU	*OP*	SM & ACCU	SM	COEF / ACCU
3	*MV*	ACCU	FM	–	*OP*	FM & ACCU	SM	COEF / ACCU
4	*MV*	FM & ACCU	FM	–	*OP*	ACCU	FM	COEF / ACCU
5					*OP*	SM & ACCU	FM	COEF / ACCU
6					*OP*	FM & ACCU	FM	COEF / ACCU
7					*MV*	ACCU	SM	–
8					*MV*	SM & ACCU	SM	–
9					*MV*	FM & ACCU	SM	–
10					*MV*	ACCU	FM	–
11					*MV*	SM & ACCU	FM	–
12					*MV*	FM & ACCU	FM	–

Algorithm 2: A 5×5 non-separable filter kernel mapped on the proposed architecture, with unrolling factor of 5.

```
for i = 4 to (H − 1) do
    // Load one image row into scratchpad.
    sm[8] ← mem[2i];
    sm[9] ← mem[2i+1];
    // Apply 5 × 5 convolution.
    accu ← C0,0 × sm.l[0];
    accu ← accu + C0,1 × sm.l[1];
    accu ← accu + C0,2 × sm[0];
    accu ← accu + C0,3 × sm[1];
    accu ← accu + C0,4 × sm.r[0];
    ... // other accu for output at mem[2H+2i]
    accu ← accu + C4,0 × sm.l[8];
    accu ← accu + C4,1 × sm.l[9];
    accu ← accu + C4,2 × sm[8];
    accu ← accu + C4,3 × sm[9];
    mem[2H+2i] ← accu + C4,4 × sm.r[8];
    ... // accu for output at mem[2H+2i+1]
    mem[2H+2i+1] ← accu + C4,4 × sm.r[9];
    ... // Remaining 4 image rows in
    ... // the unrolled code.
end
```

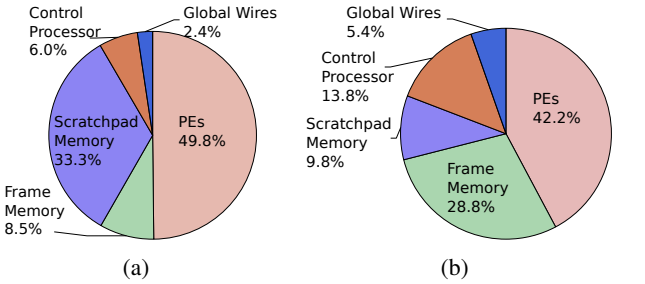


Fig. 10. System energy breakdown of the proposed architecture (a) at 1.2 V, and SM is realized by the commercial SRAM (151.9 pJ/pixel); (b) sub-threshold SM in combination with super-threshold FM (22.6 pJ/pixel), CP and global wires are only scaled to 0.7 V.

C. Exploration of HMS Implementation

ACCU, SM, and FM are the three components of the proposed HMS. Since the large on-chip FM cannot be implemented with V_{dd} scalable memory, it is therefore implemented with commercial low-power and high-density SRAM. Obviously, the ACCU register is most proper to be implemented by standard cells. The remaining question is how to implement the SM. In this section, we explore the implementation choices for the SM.

Taking the 5×5 non-separable filter as an example, Figure 10(a) shows the energy breakdown of the proposed architecture at 1.2 V when the SM is realized by the commercial SRAM. Although the *Xetal-Pro* architecture requires one extra instruction to implement this kernel compared to *Xetal-II*, the energy consumption per pixel (tile part) at nominal voltage is still 1.6 times less than that of the *Xetal-II*. Let us assume that commercial SRAM is used as the SM. Figure 11(a) shows that, after V_{dd} scaling, a total of 6.8 times reduction can be reached at the optimal point where FM = 0.7 V, SM = 0.7 V, and PE = 0.42 V. At this point *Xetal-Pro* delivers a throughput of 0.88 GOPS. However, it should be noted that more than half of the energy is consumed by the SM at this point. Thus, further energy reduction needs a SM with better V_{dd} scalability.

Similar to the analysis we did for FM in Section V, two other possible choices for the SM, i.e., the MIT low-power SRAM and the standard cells, are investigated, both of which have better V_{dd} scalability than commercial SRAM realization. According to our synthesis results, the standard-cell realization of the 128bit×32 dual-port memory is the best in terms of energy efficiency and speed. Thus, we propose a hybrid realization of our HMS, i.e., a sub-threshold standard-cell based SM in combination with super-threshold commercial SRAM based FM. Figure 11(b) shows the energy consumption of this proposed architecture. After scaling, a total of

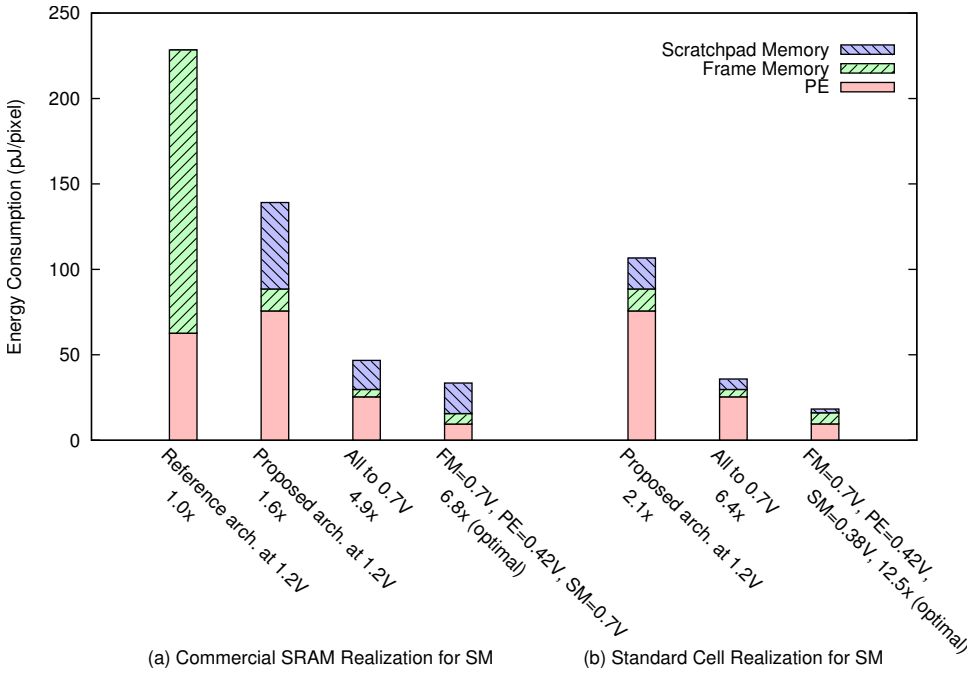


Fig. 11. Tile (proposed architecture) energy consumption for different V_{dd}

12.5 times energy saving (tile part) can be reached.

Figure 10(b) shows the system energy breakdown when the minimal energy consumption is achieved. Note that we only conservatively scale CP and global wires (together they consume 5% of the total system energy at nominal) to 0.7 V. Compared to *Xetal-II* operating at nominal voltage, *Xetal-Pro* gains more than 10 times energy reduction (240.8 pJ/pixel vs. 22.6 pJ/pixel), i.e., $<$ 0.5 pJ/16-bit op, while still delivering a throughput of 0.69 GOPS with 1.08 MHz frequency, sufficient to execute a 5×5 convolution kernel on VGA at 43 frames/s. Table VI compares the tile part energy consumption between the reference *Xetal-II* processor and the *Xetal-Pro* processor. Even for the YCbCr to RGB conversion kernel which has little locality to be exploited, 1 pJ/16-bit op is achieved.

It is worth introducing the implementation of level-shifter (LS) in the HMS. Different from conventional LS which converts from 2/3 nominal supply to full nominal supply, the LS in *Xetal-Pro* should be capable of converting between signals from a sub-threshold VDDL supply domain (e.g., 0.4 V) to a super-threshold VDDH supply domain (e.g., 0.7 V). A two-stage LS, as shown in Figure 12(a) is proposed in this work to deliver robust, fast, and energy-efficient operation. Each stage uses a normal cross-coupled differential inverter. For low voltage inputs, the first stage handles the majority of the up-conversion, and the second stage is mainly to restore and re-shape the final output signal. To ensure the correct functioning of the LS, the NMOS and PMOS devices are carefully sized so that the cross-coupled NMOS pull-down devices can overpower the PMOS pull-up devices in the presence of process variability. In addition, LV_t devices are used in the LS to enhance operation speed and reliability. LSs are small circuits in the overall system, so the impact of increased leakage by using LV_t devices is negligible in such a big system (the leakage power increase of each LS is in pW range, whereas the total system power is in mW range). Figure 12(b) shows the transient response to pull-up a input signal from 0.4 V to 0.7 V. The transition delay is 2 ns at this low frequency mode. When VDDL is in the super-threshold region, the transition delay can be less than 100 ps. It should be noted that using LV_t does not impact the dynamic energy. It also does not bring any area overhead.

D. Data Locality Analysis for HMS

To achieve ultra low energy, domain specific processors often exploit the locality of typical kernels in their application domains. In the case of *Xetal-Pro*, the size of scratchpad memory is a crucial factor for energy consumption. Scratchpad memory of small size may not accommodate the potential locality of the kernels, which causes spilling to the energy consuming frame memory. On the other hand, oversized scratchpad memory, which is more than enough for the potential locality of the kernels, consumes more energy but cannot further reduce the access to frame memory. In order to decide the optimal size of scratchpad memory for energy consumption,

TABLE VI
TILE ENERGY COMPARISON BETWEEN THE REFERENCE *Xetal-II* PROCESSOR AND THE *Xetal-Pro* PROCESSOR

Benchmarks	*Xetal-II* (65nm CMOS, max.125 MHz at 1.2 V)			*Xetal-Pro* (65nm CMOS, max.125 MHz at 1.2 V)		
	inst./pixel	pJ/pixel at 1.2 V	optimal pJ/pixel [a]	inst./pixel	pJ/pixel at 1.2 V	optimal pJ/pixel [b]
5×5 non-separable filter	25	228.6 (1.0×)	65.1 (3.5×↓)	26	106.6 (2.1×↓)	18.3 (12.5×↓)
5×5 separable filter	10	101.6 (1.0×)	29.5 (3.4×↓)	11	51.7 (2.0×↓)	10.1 (10.1×↓)
YCbCr to RGB	9	105.4 (1.0×)	32.6 (3.2×↓)	9	63.9 (1.6×↓)	16.8 (6.3×↓)

[a]FM is scaled to 0.7 V, PEs are scaled to the sub-threshold region.
[b]FM is scaled to 0.7 V, PEs and SM are scaled to the sub-threshold region.

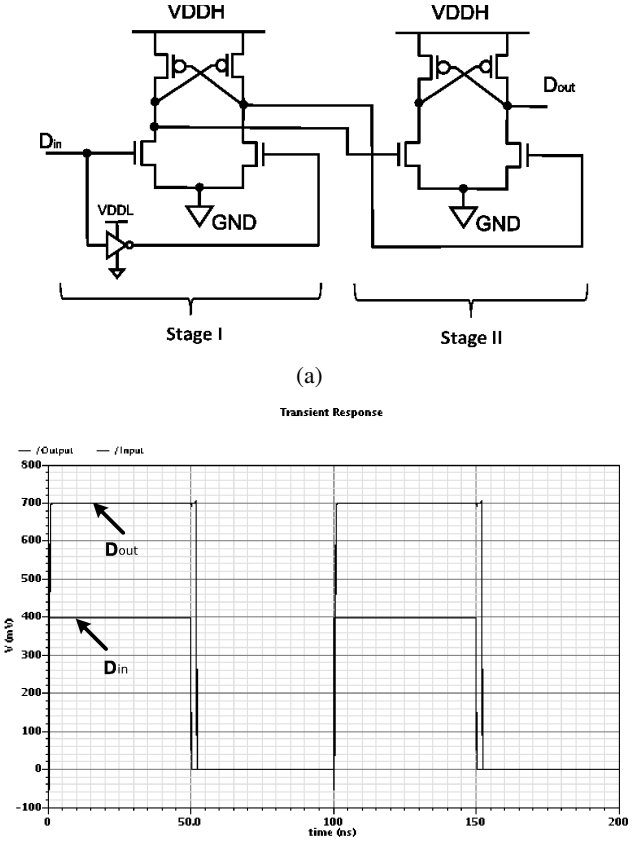


Fig. 12. (a) 2-stage level shifter (b) transient response from 0.4 V input signal to 0.7 V output signal

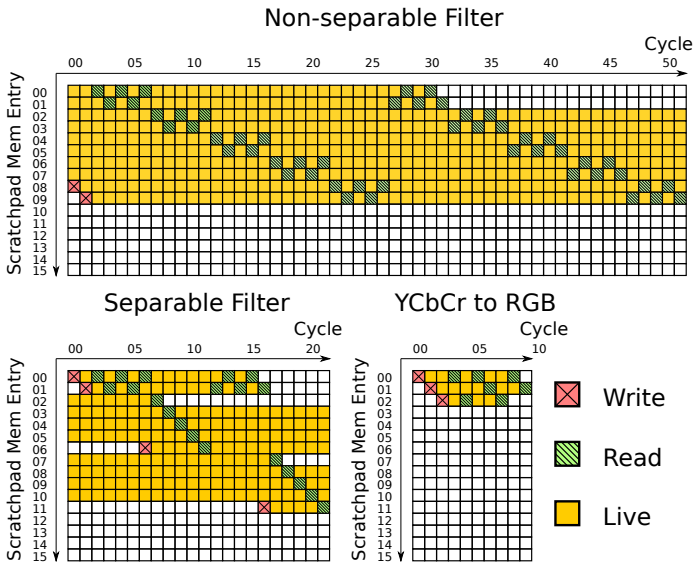


Fig. 13. Usage of scratchpad memory for three kernels: non-separable filter, separable filter, and YCbCr to RGB.

experiments are performed to analyze the locality of the kernels.

Figure 13 shows the usage of scratchpad memory for three kernels. This experiment is based on a few assumptions. First, data is stored back to frame memory only when the data cannot be reused later by the same kernel. Second, loop unrolling is performed as long as it can reduce total access to the frame memory, e.g., unrolling the interleaved pixels of the same image line. Third, intermediate results are either written to ACCU register, when it can be immediately reused by the next instruction, or written to a new location of the scratchpad memory, when the reuse distance is larger than one instruction. Fourth, only the loop body is shown in the trace, but data with reuse distance beyond the loop body is still marked as live.

According to the experiment shown in Figure 13, the scratchpad memory of size 16 will be enough for the potential locality of the kernels. However, the size of scratchpad memory optimized for the locality of the kernels is not necessarily optimal for energy consumption. For example, further reducing scratchpad size, e.g., to 8 entries, will cause spill to frame memory, but on the other hand reduce the access energy of scratchpad memory. This trade-off is not obvious unless kernels are mapped to architectures with different sizes of scratchpad memory. The next experiment is performed to analyze this trade-off.

In Figure 14, three kernels are mapped onto architectures with different sizes of scratchpad memory. The energy is obtained at nominal voltage (1.2 V). The energy per pixel of each benchmark is normalized to the energy per pixel of that benchmark on the baseline

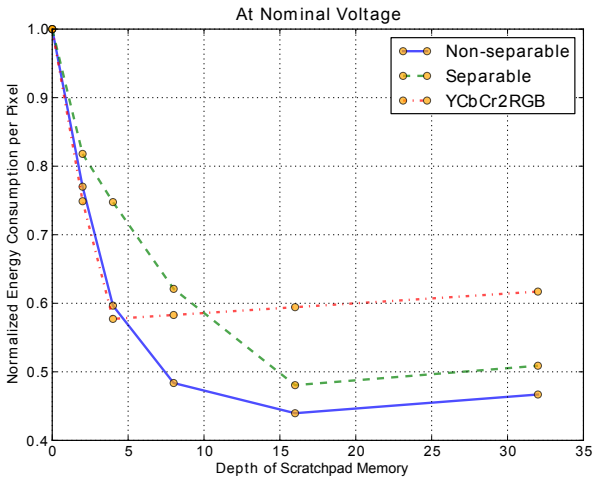


Fig. 14. Normalized energy per pixel for different sizes of scratchpad memory.

architecture, which has no scratchpad memory. For each size of scratchpad memory, the kernels are heavily optimized for energy consumption.

According to Figure 14, the optimal size of scratchpad memory for energy consumption of each benchmark matches the potential locality of the benchmark. The optimal sizes of scratchpad memory for the benchmarks are not the same. Scratchpad memory of 16 entries will be the optimal size on average for the three kernels. However, image processing applications often contain multi-pass filters, which has larger potential locality than a single filter. And applications with larger working windows or higher resolutions require a scratchpad memory of more entries too. To accommodate these cases, it is decided to use a scratchpad memory of 32 entries in *Xetal-Pro*. Based on the experiment, the energy consumption of 32 entries scratchpad memory consumes less than 5% extra energy compared to that of 16 entries. Therefore, this decision is justified.

VII. Yield Improvement for *Xetal-Pro* under Large Variability

The complete *Xetal-Pro* system includes also many supportive peripherals such as data-in-processors (DIP) and data-out-processors (DOP). We are aware that *Xetal-Pro*'s performance at very low V_{dd} can be largely impacted by design and manufacturing variabilities including process variations (both inter-die and intra-die), temperature changes, supply noise and clock skew, etc. To keep *Xetal-Pro*'s high yield up to the industrial standards, not only do we apply the circuit techniques developed in the *SubJPEG* processor, currently we are also exploring the following two approaches:

- Using post-silicon tuning to push performance (almost) back to typical even at the worst corner case. The array-like regular layout of *Xetal-Pro* partitions each tile physically as an island to implement individual V_{dd} and body-biasing tuning. A dedicated central monitor collects information from variability sensors and configures tiles to select their desirable supply and body-biasing voltages from programmable DC-DC units. The energy overhead of this central monitor should be negligible in such a large SoC.
- Adoption of the massively-parallel architecture enables great possibilities for fault-tolerant redundancy. This is a big advantage of SIMD architecture. In addition, *Xetal-Pro*'s large number of tiles and PEs can help tightening the leakage and total energy distributions among dies according to the central limit theorem. The energy impact of the PEs that have lower V_{th}s attempts to cancel out the energy impact of the PEs that have higher V_{th}s.

VIII. Conclusion

This paper presents our progress in developing the *Xetal-Pro* processor, which is the latest member of the *Xetal* SIMD processor family from Philips. *Xetal-Pro* is the first work to combine wide-range V_{dd} scaling with highly parallel SIMD architectures. While aggressive V_{dd} scaling leads to ultra low energy/op, it also causes severe throughput degradation. *Xetal-Pro* compensates this degradation by its massively-parallel nature. The predecessors in the *Xetal* family, such as the *Xetal-II*, include a large on-chip frame memory (FM), which cannot scale well to an ultra low V_{dd} hence creating a big obstacle to increase energy efficiency. Therefore, we proposed a hybrid memory system which not only exploits the often available data locality, but also enables further V_{dd} scaling. Compared to the reference design, i.e., *Xetal-II* migrated to 65 nm CMOS technology, the new architecture provides up to two times energy efficiency improvement even at the nominal operating voltage when delivering the same amount of throughput. At the ultra low-energy mode, more than 10 times energy reduction is achievable, while still delivering a throughput of 0.69 GOPS. The preliminary result makes *Xetal-Pro* a very promising building block in multiprocessor SoCs (MPSoCs) for future low-energy mobile streaming computing.

Acknowledgment

The authors would like to thank Leo Sevat from NXP Research Eindhoven for helpful discussions about the

physical implementation of *Xetal*-family processors, and Professor Takayasu Sakurai from the University of Tokyo for sharing his insights on memory energy modeling. We also would like to thank anonymous reviewers for their helpful comments and suggestions.

REFERENCES

[1] Y. He, Y. Pu, R. Kleihorst, Z. Ye, A. A. Abbo, S. M. Londono, and H. Corporaal. Xetal-pro: an ultra-low energy and high throughput simd processor. In *ACM/IEEE Design Automation Conference*, pages 543–548, 2010.

[2] D. Pham, S. Asano, M. Bolliger, M. N. Day, H. P. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, M. Riley, D. Stasiak, M. Suzuoki, M. Wang, J. Warnock, S. Weitzel, D. Wendel, T. Yamazaki, and K. Yazawa. The design and implementation of a first-generation CELL processor. In *IEEE International Solid-State Circuits Conference*, pages 184–185, 2005.

[3] A. Abbo, R. Kleihorst, V. Choudhary, L. Sevat, P. Wielage, S. Mouy, B. Vermeulen, and M. Heijligers. Xetal-II: A 107 GOPS, 600 mW massively parallel processor for video scene analysis. *IEEE Journal of Solid-State Circuits*, 43(1):192–201, 2008.

[4] N. Verma and A. Chandrakasan. A 256 kb 65 nm 8T subthreshold SRAM employing sense-amplifier redundancy. *IEEE Journal of Solid State Circuits*, 43(1):141–149, 2008.

[5] B. Calhoun and A. Chandrakasan. A 256-kb 65-nm subthreshold SRAM design for ultra-low-voltage operation. *IEEE Journal of Solid-State Circuits*, 42(3):680–688, 2007.

[6] S. Kyo and S. Okazaki. IMAPCAR: A 100 GOPS in-vehicle vision processor based on 128 ring connected four-way VLIW processing elements. *Journal of Signal Processing Systems*, 62(1), pages 1–12, 2008.

[7] H. Kaul, M. A. Anders, S. K. Mathew, S. K. Hsu, A. Agarwal, R. K. Krishnamurthy, and S. Borkar. A 300 mV 494 GOPS/W reconfigurable dual-supply 4-Way SIMD vector processing accelerator in 45nm CMOS. In *IEEE International Solid-State Circuits Conference*, pages 260–263, 2009.

[8] Y. Fujita, N. Yamashita, and S. Okazaki. Integrated memory array processor: A prototype VLSI and a real-time vision system. In *IEEE Computer Architectures for Machine Perception*, pages 82–91, 1993.

[9] S. Kyo, T. Koga, and S. Okazaki. IMAP-CE: A 51.2 GOPS video rate image processor with 128 VLIW processing elements. In *IEEE International Conference on Image Processing*, pages 294–297, 2001.

[10] R. Kleihorst, A. Abbo, A. van der Avoird, M. Op de Beeck, L. Sevat, P. Wielage, R. van Veen, and H. van Herten. Xetal: A low-power high-performance smart camera processor. In *IEEE International Symposium on Circuits and Systems*, pages 215–218, 2001.

[11] Intel. Intel processor spec finder. http://processorfinder.intel.com/.

[12] B. Khailany, W. J. Dally, A. Chang, U. J. Kapasi, J. Namkoong, and B. Towles. VLSI design and verification of the Imagine processor. In *IEEE International Conference on Computer Design*, pages 289–294, 2002.

[13] J. Pille, C. Adams, T. Christensen, S. Cottier, S. Ehrenreich, T. Kono, D. Nelson, O. Takahashi, S. Tokito, O. Torreiter, O. Wagner, and D. Wendel. Implementation of the CELL broadband engine in a 65nm SOI technology featuring dual-supply SRAM arrays supporting 6GHz at 1.3V. In *IEEE International Solid-State Circuits Conference*, pages 322–606, 2007.

[14] O. Takahashi, C. Adams, D. Ault, E. Behnen, O. Chiang, S. R. Cottier, P. Coulman, J. Culp, G. Gervais, M. S. Gray, Y. Itaka, C. J. Johnson, F. Kono, L. Maurice, K. W. McCullen, L. Nguyen, Y. Nishino, H. Noro, J. Pille, M. Riley, M. Shen, C. Takano, S. Tokito, T. Wagner, and H. Yoshihara. Migration of CELL broadband engine from 65nm SOI to 45nm SOI. In *IEEE International Solid-State Circuits Conference*, pages 86–597, 2008.

[15] M. Toksvig, J. Mathieson, and B. Cabral. NVIDIA Tegra: Enabling stunning handheld graphics & HD video. In *HOT CHIPS, Presentation*, 2008.

[16] M. Woh, S. Seo, S. Mahlke, T. Mudge, C. Chakrabarti, and K. Flautner. AnySP: Anytime anywhere anyway signal processing. *IEEE Micro*, 30(1):81–91, 2010.

[17] NVIDIA. Geforce. http://www.nvidia.com/.

[18] A. Wang and A. Chandrakasan. A 180-mV subthreshold FFT processor using a minimum energy design methodology. *IEEE Journal of Solid-State Circuits*, 40(1):310–319, 2005.

[19] B. Zhai, S. Hanson, and D. Sylvester D. Blaauw. A variation-tolerant sub-200mV 6T subthreshold SRAM. *IEEE Journal of Solid-State Circuits*, 44(10):2338–2348, 2008.

[20] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, and T. Austin. A 2.60 pJ/inst subthreshold sensor processor for optimal energy efficiency. In *IEEE Symposium on VLSI Circuits*, pages 154–155, 2006.

[21] M. Seok, S. Hanson, Y. S. Lin, Z. Foo, D. Kim, Y. Lee, N. Liu, D. Sylvester, and D. Blaauw. The Phoenix processor: A 30pW platform for sensor applications. In *IEEE Symposium on VLSI Circuits*, pages 188–189, 2008.

[22] J. Kwong, Y. K. Ramadass, N. Verma, and A. Chandrakasan. A 65 nm Sub-V_t microcontroller with integrated SRAM and switched capacitor DC-DC converter. *IEEE Journal of Solid-State Circuits*, 44(1):115–126, 2009.

[23] Y. Pu, J. Pineda de Gyvez, H. Corporaal, and Y. Ha. An ultra-low-energy multi-standard JPEG co-processor in 65nm CMOS with sub/near threshold supply voltage. *IEEE Journal of Solid-State Circuits*, 45(3):668–680, 2010.

[24] P. Francesco, P. Marchal, D. Atienza, L. Benini, F. Catthoor, and J. M. Mendias. An integrated hardware/software approach for run-time scratchpad management. In *ACM/IEEE Design Automation Conference*, pages 238–243, 2004.

[25] N. Jayasena, M. Erez, J. H. Ahn, and W. J. Dally. Stream register files with indexed access. In *IEEE International Symposium on High Performance Computer Architecture*, pages 60–72, 2004.

[26] A. Prengler and K. Adi. A reconfigurable SIMD-MIMD processor architecture for embedded vision processing applications. In *SAE World Congress & Exhibition*, 2009.

[27] Y. He, Z. Zivkovic, R. Kleihorst, A. Danilin, and H. Corporaal. Real-time implementations of hough transform on SIMD architecture. In *ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–8, 2008.