Xetal-Pro: An Ultra-Low Energy and High Throughput SIMD Processor

Yifan He, Yu Pu
Eindhoven University of Technology, the Netherlands
{y.he, y.pu}@tue.nl

Zhenyu Ye
Eindhoven University of Technology, the Netherlands
z.ye@tue.nl

Sebastian M. Londono
Eindhoven University of Technology, the Netherlands
s.moreno@tue.nl

Richard Kleihorst
VITO, Belgium
richard.kleihorst@vito.be

Anteneh A. Abbo
Philips Research, the Netherlands
anteneh.a.abbo@philips.com

Henk Corporaal
Eindhoven University of Technology, the Netherlands
h.corporaal@tue.nl

ABSTRACT

This paper presents *Xetal-Pro* SIMD processor, which is based on *Xetal-II*, one of the most computational-efficient (in terms of GOPS/Watt) processors available today. *Xetal-Pro* supports ultra wide V_{DD} scaling from nominal supply to the sub-threshold region. Although aggressive V_{DD} scaling causes severe throughput degradation, this can be compensated by the nature of massive parallelism in the *Xetal* family. The predecessor of *Xetal-Pro*, *Xetal-II*, includes a large on-chip frame memory (FM), which cannot operate reliably at ultra low voltage. Therefore we investigate both different FM realizations and memory organization alternatives. We propose a hybrid memory architecture which reduces the non-local memory traffic and enables further V_{DD} scaling. Compared to *Xetal-II* operating at nominal voltage, we could gain more than 10× energy reduction while still delivering a sufficiently high throughput of 0.69 GOPS (counting multiply and add operations only). This work gives a new insight to the design of ultra-low energy SIMD processors, which are suitable for portable streaming applications.

Categories and Subject Descriptors

C.1 [**Processor Architectures**]: Multiple Data Stream Architectures (Multiprocessors)—*Single-instruction-stream, multiple-data-stream processors (SIMD)*

General Terms

Algorithms, Design, Performance

Keywords

Xetal-Pro, Hybrid Memory System, Low-Energy, SIMD

1. INTRODUCTION

To enhance computational performance and energy efficiency of the latest video standards, such as H.264 and MPEG4, stream processors are often integrated in SoCs within portable devices. Among these stream processors, massively-parallel Single Instruction Multiple Data (SIMD) processors are very popular because (1) SIMD is a low power architecture since it applies the same instructions to all processing elements (PEs) and (2) massive parallelism in streaming applications typically shows up as data-level parallelism (DLP) which is naturally supported by SIMD architectures. However, practice today is that the embedded streaming processor in a cellular phone consumes tens of pJ per operation (pJ/op) and the battery capacity is only sufficient for playing video applications for a few hours. Meanwhile, the large power dissipation also worsens the chip's thermal issue. To significantly improve energy efficiency for future mobile streaming applications, this paper presents our progress in developing the *Xetal-Pro* processor, which will be the newest child of the *Xetal* processor family from Philips.

The predecessor of *Xetal-Pro* is *Xetal-II*[1], which has been implemented in a 90 nm CMOS process with 74 mm^2 die area. It has 320 PEs, and delivers a peak performance of 107 GOPS on 16-bit data when running at 84 MHz, with a power budget of 600 mW. Compared to *Xetal-II*, *Xetal-Pro* has the following improvements:

(1) It supports ultra-wide-range V_{DD} scaling from a nominal supply to sub/near threshold supply. Although aggressive V_{DD} scaling will cause throughput degradation, the massively-parallel nature of *Xetal-Pro* can compensate for such degradation. Even operating in the sub/near threshold mode, it still renders a reasonably high throughput.

(2) *Xetal-II* includes a large SRAM based on-chip frame memory (FM) of 10 Mbit, which allows on-chip storage of multiple VGA frames. This dramatically reduces off-chip traffic and helps to enhance performance and energy efficiency. However, it causes a problem when applying aggressive V_{DD} scaling. SRAMs typically cannot operate reliably below 0.7 V[11]. Alternative realizations exist, such as the low-power SRAM cells from MIT[2][11], or using standard cell memory logic. However, our analysis shows that these alternatives are not effective for the large on-chip FM. To address this issue, we propose a hybrid memory system (HMS), containing (1) A hybrid memory architecture: consisting of an ACCU register, a scratchpad memory (SM), and the FM; (2) A hybrid realization: sub-threshold SM in combination with super-threshold FM.

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.
DAC 2010, June 13-18, 2010, Anaheim, California, USA.
Copyright 2010 ACM ACM 978-1-4503-0002-5 ...$10.00.

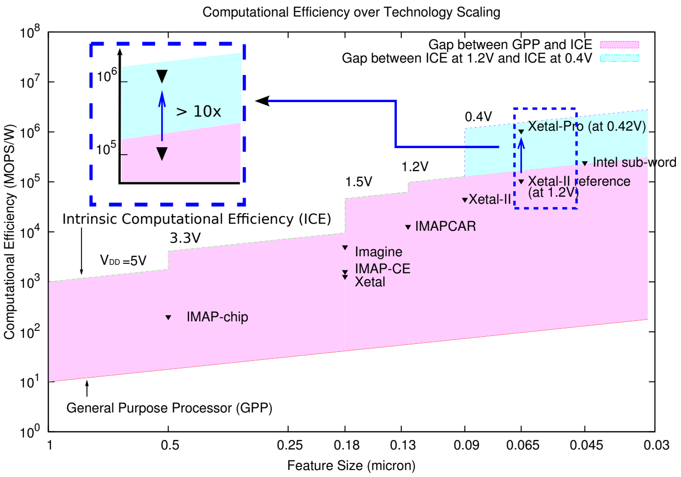


Figure 1: ICE Curve Extended with V_{DD} Scaling

To test our system, a general kernel-based filter operation was chosen, which is a representative application for SIMD processors[1][8]. The proposed features bring a total energy reduction of more than 10× compared to *Xetal-II*. *Xetal-Pro* then runs at about 0.4 V, while it can still achieve 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]. Other issues, such as energy breakdown based on the synthesis results using 65 nm low-power libraries, implementation choices of the hybrid memory architecture, and enhancing yield under large variability, are also covered in this paper. This work gives new insights on how to design low energy SIMD processors, which are suitable for future portable streaming systems.

2. RELATED WORK

2.1 Sub-threshold Designs

Several prototype chips that 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[12], and a 256 Kbit 10-T dual-port SRAM in 65 nm CMOS process[2], which has later been improved to 8-T dual-port SRAM[11]. A 130 nm and a 180 nm CMOS sensor node processors are presented in [13] and [10], respectively. A TI-MSP430 based DSP processor with integrated DC-DC converter in 65 nm CMOS is presented in [7]. The *SubJPEG* prototype chip, a 65 nm CMOS 8-bit JPEG co-processor, is presented in [9]. It is equipped with 4 parallel DCT-Quantization engines and delivers 15 fps VGA processing at about 0.4 V. The physical design techniques of *SubJPEG* are migrated to *Xetal-Pro*. Recently Intel announced its 45 nm CMOS 300 mV 4-Way sub-word parallel processor[5].

2.2 SIMD Processors

Other than *Xetal-II*, *IMAPCAR*[8] from NEC is another 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. Subword parallel processors[5] also benefit from using parallelism, however, they are not massively-parallel processors for very low-energy applications.

2.3 Scratchpad Memory

It is well-known that using scratchpad memories may reduce the traffic to higher levels substantially when applications show substantial locality[3]. For example, a stream register file (or memory) as used in the Imagine architecture[4] can provide high performance with low energy consumption for streaming applications.

However, 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.

3. EXPLORATION OF XETAL-II

Xetal-Pro is a derivative of the *Xetal* family. It inherits many peculiarities of the *Xetal-II* processor. As the starting line to the development of *Xetal-Pro*, *Xetal-II*'s architecture, performance and energy breakdown was carefully analyzed.

3.1 Xetal-II Processor Architecture

The block diagram of the *Xetal-II* processor is indicated in Figure 2(a). The control processor (CP) is a 16-bit, 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. Layout and memory considerations necessitate partitioning of the linear processor array, containing 320 PEs and an integral 10 Mbit FM, into tiles. The number of PEs per tile is based on the energy/area efficiency analysis of the shared FM, as well as the layout constraints. 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 having 8 PEs (power of 2) per tile (thus, 40 tiles in total) is a good choice considering FM access energy efficiency, FM total area efficiency, and 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 be further compounded with other operations (e.g. absolute, or negative). All instructions are executed in a single cycle. The FM is built from 40 commercial SRAM modules (128bit×2048) with a pseudo-dual port interface to provide single cycle read and write accesses. This data memory stores both the frame data and the intermediate results. The relatively large capacity of the FM allows on-chip storage of multiple VGA frames or images with higher resolution. 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, PEs is used to refer to the logic part (processing elements and communication network) of the tile, and FM is used to refer to the memory part of the tile.

[1] In our ICE curve, only multiply and add operations are counted, and the energy of 8-bit and 16-bit operations are linearly scaled to 32-bit operations.

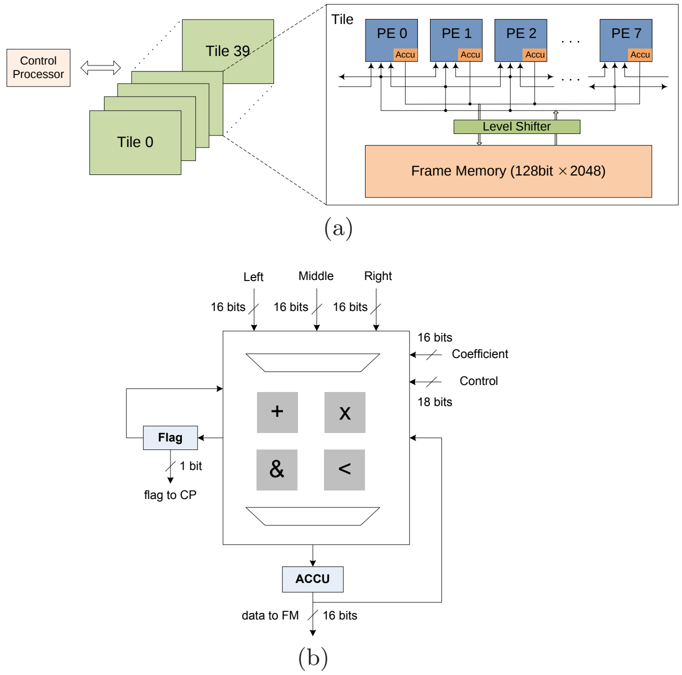


Figure 2: (a) Block Diagram of *Xetal-II* Architecture; (b) Structure of the 16-bit PE

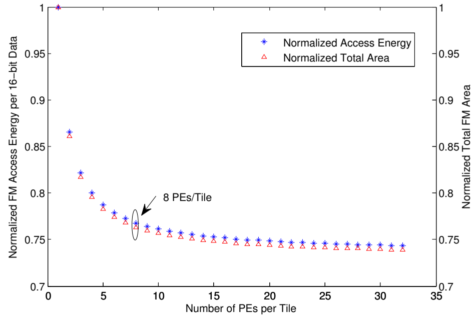


Figure 3: Number of PEs per tile vs. normalized FM access energy per 16-bit data and normalized total FM area

3.2 Energy/Performance Analysis

As a reference of *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 SV_T CMOS digital standard cell library. The V_T of this process is about 0.41˜0.42 V. The SRAM was synthesized with a commercial low-power memory generator (choosing HV_T for bit cells) in the same process technology. The impact of the long global decoded instruction wires have been considered based on post-layout analog simulation results. The whole system can run at 125 MHz with 1.2 V voltage supply, and offers 80 GOPS throughput (counting multiply and add operations only) with each PE processing 1 inst/cycle. The critical path is the FM read access plus the PE (MAC) operation.

To analyze the system energy breakdown, we chose a general kernel-based filter operation as a representative application for all algorithms with an N×N convolution: smoothing operations (linear, Gaussian), derivative operations (Gaussian gradient, Laplacian), color reconstruction filters, morphological operations, etc.[6]. Its high regularity and large potential of DLP makes it very suitable for SIMD processing. The mapping of a 5×5 non-separable filter kernel on the reference (*Xetal-II*) processor is shown in Figure 4. The filter kernel executed on each PE is described in Algorithm 1. A total of 25 instructions are required to process each pixel. Figure 5 depicts the energy breakdown of the reference processor when running this filter. The average energy consumption is 240.8 pJ/pixel (9.6 pJ/inst). About 69% of the total energy is consumed by the FM, while the PEs consume 26%. Compared with the 40 tiles (PEs + FM), CP and the global decoded instruction wires (from CP to the input of each tile) consume much less energy. To effectively reduce the total energy, the tiles are the focus of our further study.

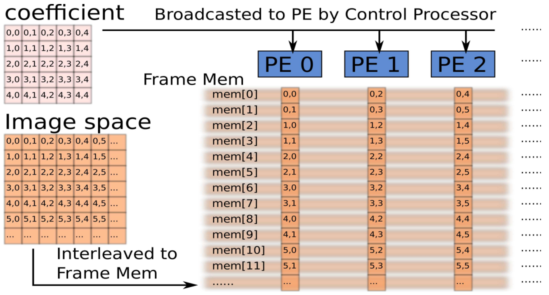


Figure 4: A 5×5 filter applied on the VGA image (interleaving factor = 2)

Algorithm 1: A 5×5 filter kernel applied on the VGA image (interleaving factor = 2). Assume image height is H. Each PE can read the memory on its left (mem.L) and right (mem.R). Results of pixel at mem[i] are written to mem[2H+i].

```
for  $h = 2$ to $(H-3)$ do
    accu ← c[0,0] × mem.L[2h-4];
    accu ← accu + c[0,1] × mem.L[2h-3];
    accu ← accu + c[0,2] × mem[2h-4];
    accu ← accu + c[0,3] × mem[2h-3];
    accu ← accu + c[0,4] × mem.R[2h-4];
    ... // other accu for the output at mem[2H+2h]
    accu ← accu + c[4,0] × mem.L[2h+4];
    accu ← accu + c[4,1] × mem.L[2h+5];
    accu ← accu + c[4,2] × mem[2h+4];
    accu ← accu + c[4,3] × mem[2h+5];
    mem[2H+2h] ← accu + c[4,4] × mem.R[2h+4];
    ... // accu for the output at mem[2H+2h+1]
    mem[2H+2h+1] ← accu + c[4,4] × mem.R[2h+5];
end
```

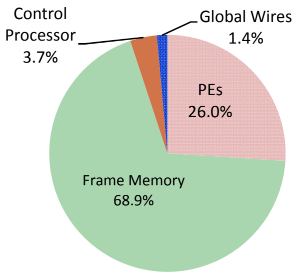
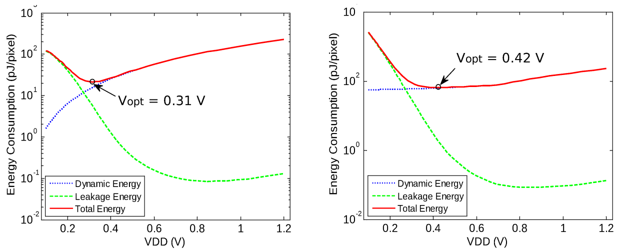


Figure 5: Energy breakdown of the *Xetal-II* processor at 1.2 V when executing a 5×5 non-separable filter kernel. Note that tiles (PEs + FM) consume 95% of the total system energy.



(a) (b)

Figure 6: V_{DD} vs. energy consumption when processing one pixel with a 5×5 filter kernel (a) assuming ideal SRAM voltage scaling; (b) SRAM only scales to 0.7 V

4. CHALLENGE OF ULTRA-WIDE-RANGE VDD SCALING

In this section, ultra-wide-range V_{DD} scaling is applied to the most energy-consuming part, i.e. the tile. The energy consumption when processing one pixel (applying the 5×5 filter kernel, 25 instructions in total) is used as comparison metric throughout the remaining parts of this paper. Figure 6(a) depicts the energy consumption curve under different supply voltages. Note that here we assume the SRAM can scale to sub-threshold as well as the standard cells. This is an unrealistic assumption, 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 10× reduction of the energy consumption ideally achievable, compared to operating at 1.2 V.

However, with voltage scaling, the maximal frequency (thus the maximal throughput each PE can achieve) also decreases dramatically (the lower curve of Figure 7), which causes severe performance loss. Fortunately, with 320 PEs, we can still achieve reasonably high performance even at very low voltage. The upper curve of Figure 7 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[2].

Figure 6(a) presents the ideal lower energy consumption bound of the reference processor. In practice, commercial SRAM cannot operate reliably below 0.7 V. Figure 6(b) shows the practical V_{DD} scaling result (SRAM only scales to 0.7 V). The minimal energy consumption (65.1 pJ/pixel) is reached when the logic part is scaled to 0.42 V. Compared with the nominal voltage supply, the energy reduction is only a factor of 3.5, far behind the 10× ideally achievable reduction. Note that here about 88% of the energy is consumed by the FM.

The tile energy consumption for different V_{DD} is compared in Figure 8(a). We can see that even when PEs are aggressively scaled to near threshold, it only reduces an extra 15% of the energy compared to that when both PEs and SRAM are supplied at 0.7 V. Thus, in our case, unless the FM can also scale further, it does not make too much sense to aggressively scale the standard-cell (PEs) part due to the low *energy gain/performance loss* ratio.

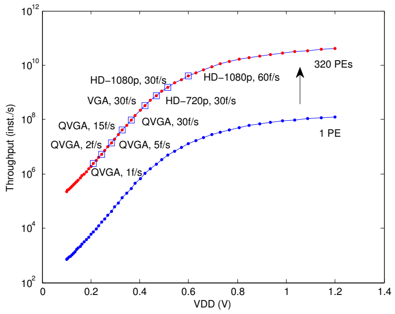


Figure 7: Impact of V_{DD} scaling on system throughput of 1 PE (lower curve) and 320 PEs (upper curve). The blue squares on the upper curve indicate the supported resolution and frame rate with 320 PEs when executing a 5×5 filter kernel

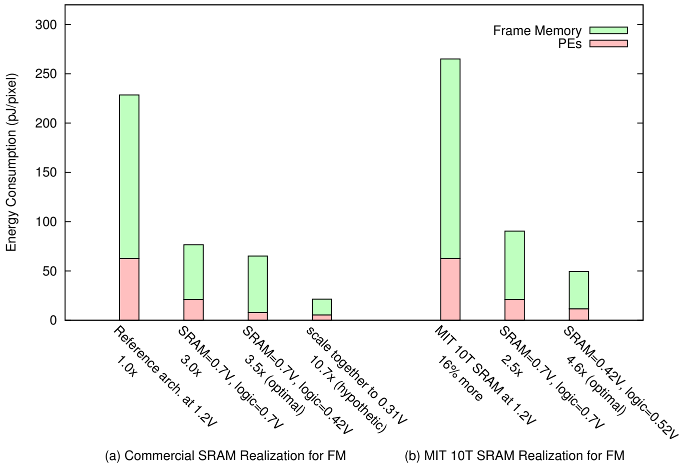


Figure 8: Tile (reference processor) energy consumption for different V_{DD}

5. EXPLORATION OF VDD SCALABLE FM

Commercial SRAM is the bottleneck of V_{DD} scaling. Based on the analysis above, 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 SRAM[2][11] and the standard-cell synthesized memory are two possible choices.

The MIT SRAM (10T) can be scaled to below 0.4 V. However, it consumes more access energy at nominal voltage and occupies 66% more cell area compared to the commercial 6T SRAM[2]. The area efficiency (*SRAM cell array area/SRAM total area*) of our FM (6T SRAM) is 70%. If this FM is realized by the 10T SRAM, more than 30% area overhead will be added to each tile. The much lower speed of the MIT SRAM is also severe. The reported maximal speed is 2.5× slower than the commercial SRAM with the same word width and depth that we are using. This severely degrades the performance at both nominal and scaled voltage. Moreover, the high leakage power (about 100 μW at 1.2 V) also prevents it from scaling to ultra low voltage, as the leakage energy

[2] 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.

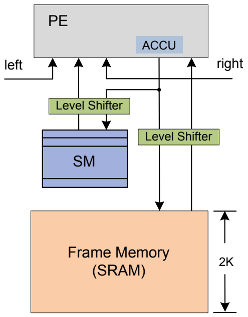


Figure 9: Proposed Hybrid Memory Architecture

increase quickly counteracts the reduction of the dynamic energy. Figure 8(b) presents the energy consumption when FM is realized by the MIT 10T SRAM. 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 memory is not applicable in our case.

The standard-cell realization of large on-chip SRAM is also not applicable. According to our synthesis result, it consumes much more power and area than the MIT 10T SRAM at nominal voltage. So, to reach our goals (ultra-low-energy, ultra-wide-voltage-range, and medium-to-high-throughput SIMD), architecture improvements are required.

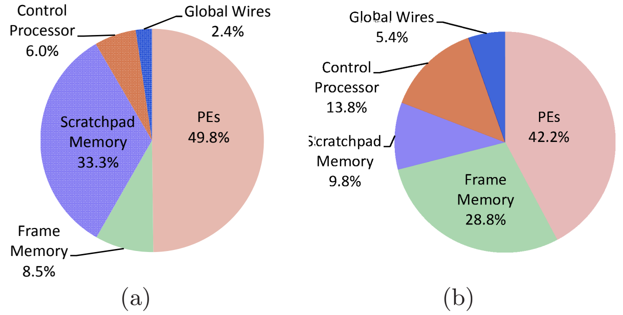
6. MEMORY HIERARCHY EXPLORATION

Since V_{DD} scalable FM is not applicable in our case, we propose a hybrid memory architecture to (1) exploit the often available data locality and reduce the non-local memory traffic and (2) enable further V_{DD} scaling.

6.1 Proposed Hybrid Memory Architecture

The Hybrid Memory Architecture (HMA) is proposed to reduce the access rate from PEs to the FM by exploiting the data locality in the scratchpad memory (SM) (Figure 9) and enable further memory V_{DD} scaling. Within the proposed HMA, we have three characterized memories to hold the data: (1) ACCU register: short-term data; (2) SM: intermediate-term data; and (3) FM: long-term data. Both the FM and the SM are directly accessible by the PE, with SM consumes less energy per access due to its much smaller size. 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 a few μW leakage overhead. The critical path of the system is also not changed (FM read access plus PE operation). Notably, coupled with the index addressing, the SM can also be used as a look-up table for complex and irregular operations.

The SM is dual-ported with 128-bit word width and 32 entries. The reasons that we chose this relatively large number of entries are (1) to enable more applications with large working windows (e.g. motion estimation) or higher resolutions (>VGA) to fully exploit data locality and (2) to demonstrate that even with such a (relatively) large size, we can still reach more than 10× energy gain. 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. For the 5×5 filter kernel, the implementaton on the proposed architecture requires one extra instruction.



(a) (b)

Figure 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.

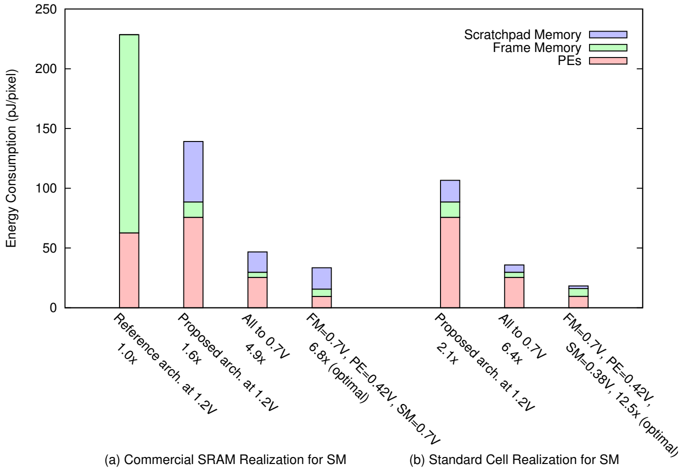


Figure 11: Tile (proposed architecture) energy consumption for different V_{DD}

6.2 Exploration of HMA Implementation

The proposed HMA consists of ACCU, SM, and FM. In Section 5, we have shown that V_{DD} scalable memory is not applicable for the large on-chip FM. So, commercial SRAM is used. Clearly, the ACCU register is most properly implemented by standard cells. In this section, we exploit the implementation choices for the SM.

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 new architecture requires one extra instruction to implement the 5×5 filter kernel, the energy consumption per pixel (tile part) at nominal voltage is still 1.6× less than that of the reference processor. After voltage scaling (Figure 11(a)), a total of 6.8× reduction can be reached at the optimal point (FM = 0.7 V, SM = 0.7 V, and PE = 0.42 V) with a throughput of 0.88 GOPS. Note that more than half of the energy consumption goes to the SM at this point. Thus, further reduction requires an SM with better scalability.

Similar to the analysis we did for FM in Section 5, two other potential choices for the SM, the MIT low-power SRAM and the standard cells, are investigated, both of which have better voltage 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 HMA, i.e. a sub-threshold SM in combination with super-threshold FM. Figure 11(b) shows the energy consumption of this proposed architecture (SM is realized by the standard cells). After scaling, a total of 12.5× 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 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× energy reduction (i.e. < 1 pJ/16-bit op) while still delivering a throughput of 0.69 GOPS, sufficient to execute a 5×5 convolution kernel on VGA at 43 frames/s.

7. ENHANCING YIELD UNDER LARGE VARIABILITY

Design and manufacturing variabilities, including process variations (both inter-die and intra-die in 65 nm technology and below), temperature changes, supply noise and clock skew, largely impact *Xetal-Pro*'s performance, especially at very low voltage. For example, our simulation shows that at 0.4 V V_{DD} under 25 °C room temperature, the $3\sigma/\mu$ of the critical path delay inside each PE can be higher than 50%! To keep a high yield up to industrial standards, *Xetal-Pro* uses the techniques developed in *SubJPEG*. Currently we are also exploring post-silicon tuning, which can push performance (almost) back to typical even at worst corner case. The regular layout of *Xetal-Pro* partitions each tile as an island to implement individual V_{DD} and body-biasing tuning. The energy overhead due to a dedicated central monitor, which configures tiles to select their desirable V_{DD}s and body-biasing voltages from an off-chip programmable DC-DC unit, should be negligible in such a large system. We also observe that, *Xetal-Pro*'s large number of tiles/PEs helps tightening the leakage and total energy distributions among dies according to the central limit theorem. In addition, adoption of the massively-parallel architecture also enables the possibilities for fault-tolerant redundancy, which is our future work.

8. MAPPING OF KERNELS ON BASELINE AND PROPOSED ARCHITECTURE

To make things clear, I create a new section for the mapping. The mapping of three kernels on old Xetal architecture (without scratchpad memory) is shown in Figure 12.

The pseudocode of mapping YUV-to-RGB, non-separable filter and separable filter on the baseline archtiecture. Assume the input image is in VGA format (640 × 320 pixels) with interleaving factor of two. Each PE can read the memory on its left (mem.L) and right (mem.L). Assume the image is of height H (H is equal to 320 for VGA format).

The mapping of three kernels on Xetal-Pro (with scratchpad memory) is shown in Figure 13.

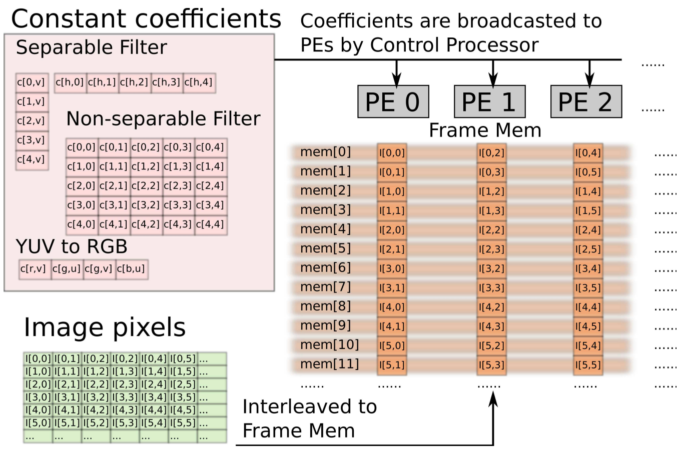

Constant coefficients
Separable Filter
c[0,v]
c[1,v]
c[2,v]
c[3,v]
c[4,v]
c[h,0] c[h,1] c[h,2] c[h,3] c[h,4]
Non-separable Filter
c[0,0] c[0,1] c[0,2] c[0,3] c[0,4]
c[1,0] c[1,1] c[1,2] c[1,3] c[1,4]
c[2,0] c[2,1] c[2,2] c[2,3] c[2,4]
c[3,0] c[3,1] c[3,2] c[3,3] c[3,4]
c[4,0] c[4,1] c[4,2] c[4,3] c[4,4]
YUV to RGB
c[r,v] c[g,u] c[g,v] c[b,u]
Coefficients are broadcasted to PEs by Control Processor
PE 0
PE 1
PE 2
Frame Mem
mem[0]
mem[1]
mem[2]
mem[3]
mem[4]
mem[5]
mem[6]
mem[7]
mem[8]
mem[9]
mem[10]
mem[11]
I[0,0] I[0,1] I[1,0] I[1,1] I[2,0] I[2,1] I[3,0] I[3,1] I[4,0] I[4,1] I[5,0] I[5,1]
I[0,2] I[0,3] I[1,2] I[1,3] I[2,2] I[2,3] I[3,2] I[3,3] I[4,2] I[4,3] I[5,2] I[5,3]
I[0,4] I[0,5] I[1,4] I[1,5] I[2,4] I[2,5] I[3,4] I[3,5] I[4,4] I[4,5] I[5,4] I[5,5]
Image pixels
I[0,0] I[0,1] I[0,2] I[0,2] I[0,4] I[0,5]
I[1,0] I[1,1] I[1,2] I[1,3] I[1,4] I[1,5]
I[2,0] I[2,1] I[2,2] I[2,3] I[2,4] I[2,5]
I[3,0] I[3,1] I[3,2] I[3,3] I[3,4] I[3,5]
I[4,0] I[4,1] I[4,2] I[4,3] I[4,4] I[4,5]
I[5,0] I[5,1] I[5,2] I[5,3] I[5,4] I[5,5]
Interleaved to Frame Mem

Figure 12: Mapping of YUV-to-RGB, non-separable filter and separable filter on the baseline architecture.

Algorithm 2: A 5×5 non-separable filter kernel mapped on the baseline architecture

```
for  h = 2 to (H − 3) do
    accu ← c[0,0] × mem.L[2h-4];
    accu ← accu + c[0,1] × mem.L[2h-3];
    accu ← accu + c[0,2] × mem[2h-4];
    accu ← accu + c[0,3] × mem[2h-3];
    accu ← accu + c[0,4] × mem.R[2h-4];
    ... // other accu for the output at mem[2H+2h]
    accu ← accu + c[4,0] × mem.L[2h+4];
    accu ← accu + c[4,1] × mem.L[2h+5];
    accu ← accu + c[4,2] × mem[2h+4];
    accu ← accu + c[4,3] × mem[2h+5];
    mem[2H+2h] ← accu + c[4,4] × mem.R[2h+4];
    ... // accu for the output at mem[2H+2h+1]
    mem[2H+2h+1] ← accu + c[4,4] × mem.R[2h+5];
end
```

Algorithm 3: A 5×5 separable filter kernel mapped on the baseline architecture

```
for  h = 2 to (H − 3) do
    // horizontal convolution with a new row
    // result is stored to mem[tmp+modulo(h,5)]
    accu ← c[h,0] × mem.L[2h-4];
    accu ← accu + c[h,1] × mem.L[2h-3];
    accu ← accu + c[h,2] × mem[2h-4];
    accu ← accu + c[h,3] × mem[2h-3];
    mem[tmp+modulo(h,5)] ←
                    accu + c[h,4] ×mem.R[2h-4];
    // vertical convolution with previous results
    // of horizontal convolution
    accu ← c[0,v] × mem[tmp+modulo(h-4,5)];
    accu ← accu + c[1,v] × mem[tmp+modulo(h-3,5)];
    accu ← accu + c[2,v] × mem[tmp+modulo(h-2,5)];
    accu ← accu + c[3,v] × mem[tmp+modulo(h-1,5)];
    mem[2H+2h] ←
            accu + c[3,v] × mem[tmp+modulo(h,5)];
    ... // accu for the output at mem[2H+2h+1]
    mem[2H+2h+1] ← accu + c[4,4] × mem.R[2h+5];
end
```

9. CONCLUSION

This paper presents *Xetal-Pro*, the first work to combine ultra-wide-range V_{DD} scaling to massively parallel SIMD architectures. While aggressive V_{DD} scaling leads to ultra low energy per operation, it also causes severe throughput degradation. *Xetal-Pro* compensates these losses by its massively-parallel nature. The predecessors in the Xetal family, such as

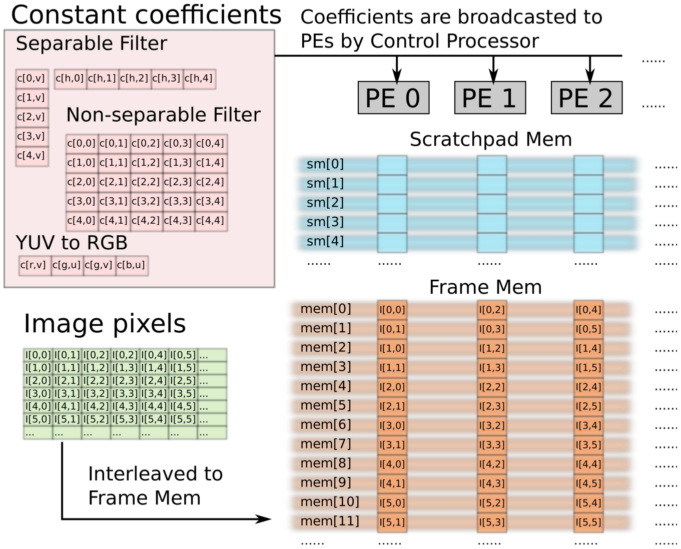

Constant coefficients
Coefficients are broadcasted to PEs by Control Processor
Separable Filter
c[0,v] c[1,v] c[2,v] c[3,v] c[4,v]
c[h,0] c[h,1] c[h,2] c[h,3] c[h,4]
Non-separable Filter
c[0,0] c[0,1] c[0,2] c[0,3] c[0,4]
c[1,0] c[1,1] c[1,2] c[1,3] c[1,4]
c[2,0] c[2,1] c[2,2] c[2,3] c[2,4]
c[3,0] c[3,1] c[3,2] c[3,3] c[3,4]
c[4,0] c[4,1] c[4,2] c[4,3] c[4,4]
YUV to RGB
c[r,v] c[g,u] c[g,v] c[b,u]
PE 0
PE 1
PE 2
Scratchpad Mem
sm[0]
sm[1]
sm[2]
sm[3]
sm[4]
Frame Mem
mem[0] I[0,0] I[0,2] I[0,4]
mem[1] I[0,1] I[0,3] I[0,5]
mem[2] I[1,0] I[1,2] I[1,4]
mem[3] I[1,1] I[1,3] I[1,5]
mem[4] I[2,0] I[2,2] I[2,4]
mem[5] I[2,1] I[2,3] I[2,5]
mem[6] I[3,0] I[3,2] I[3,4]
mem[7] I[3,1] I[3,3] I[3,5]
mem[8] I[4,0] I[4,2] I[4,4]
mem[9] I[4,1] I[4,3] I[4,5]
mem[10] I[5,0] I[5,2] I[5,4]
mem[11] I[5,1] I[5,3] I[5,5]
Image pixels
I[0,0] I[0,1] I[0,2] I[0,2] I[0,4] I[0,5] ...
I[1,0] I[1,1] I[1,2] I[1,3] I[1,4] I[1,5] ...
I[2,0] I[2,1] I[2,2] I[2,3] I[2,4] I[2,5] ...
I[3,0] I[3,1] I[3,2] I[3,3] I[3,4] I[3,5] ...
I[4,0] I[4,1] I[4,2] I[4,3] I[4,4] I[4,5] ...
I[5,0] I[5,1] I[5,2] I[5,3] I[5,4] I[5,5] ...
Interleaved to Frame Mem

Figure 13: Mapping of YUV-to-RGB, non-separable filter and separable filter on the proposed architecture.

Xetal-II, include a large on-chip frame memory (FM), which cannot operate reliably at ultra low voltage. Therefore, we proposed a hybrid memory architecture with a hybrid realization, which not only exploits the often available data locality, but also enables further V_{DD} scaling. Compared to the reference (*Xetal-II* migrated to 65 nm technology) design, more than 10× energy reduction is achieved, while still delivering a throughput of 0.69 GOPS. The result makes *Xetal-Pro* an attractive building block for future low-power MPSoCs.

10. REFERENCES

[1] 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.

[2] B. Calhoun and A. Chandrakasan. A 256kb sub-threshold SRAM in 65nm CMOS. In *IEEE Int. Solid-Stace Circ. Conf*, pages 2592–2601, 2006.

[3] P. Francesco, P. Marchal, D. Atienza, L. Benini, F. Catthoor, and J. Mendias. An integrated hardware/software approach for run-time scratchpad management. In *Proceedings of the 41st annual conference on Design automation*, pages 238–243. ACM New York, NY, USA, 2004.

[4] N. Jayasena, M. Erez, J. Ahn, and W. Dally. Stream register files with indexed access. In *High Performance Computer Architecture, 2004. HPCA-10. Proceedings. 10th International Symposium on*, pages 60–72, 2004.

[5] H. Kaul, M. A. Anders, S. K. Mathew, S. K. Hsu, A. Agarwal, R. K. Krishnamurthy, and S. Borkar. A 300mV 494GOPS/W Reconfigurable Dual-Supply 4-Way SIMD Vector Processing Accelerator in 45nm CMOS. In *IEEE Int. Solid-Stace Circ. Conf*, pages 260–263, 2009.

[6] R. Kenneth. *Castleman. Digital image processing.* Prentice Hall Press, Upper Saddle River, NJ, 1996.

[7] J. Kwong, Y. 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.

[8] 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*, pages 1–12.

[9] Y. Pu, J. de Gyvez, H. Corporaal, and Y. Ha. An Ultra-Low-Energy/Frame Multi-Standard JPEG CO-Processor in 65nm CMOS with Sub/Near-Threshold Power Supply. In *IEEE Int. Solid-Stace Circ. Conf*, pages 146–147, 2009.

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

[11] 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, 2008.

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

[13] 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 *VLSI Circuits, 2006. Digest of Technical Papers. 2006 Symposium on*, pages 154–155, 2006.