An Architecture for Low-Power High-Performance Embedded Computing

***Ronald G. Dreslinski*[1], *Qi Zheng*[1], *Robert P. Higgins*[3], *Johann Hauswald*[1], *David Blaauw*[1], *Trevor Mudge*[1], *Chaitali Chakrabarti*[2], *Jon Ballast*[3], *Warren Snapp*[3]**

[1] Department of Electrical Engineering
and Computer Science,
University of Michigan, Ann Arbor, MI

[2] School of Electrical Computer and
Energy Engineering,
Arizona State University, Tempe, AZ

[3] Boeing Co.,
Seattle, WA

Abstract: *There is a growing number of systems that require high-performance low-power embedded solutions. Recently the DARPA PERFECT program set forth an efficiency goal of 75 GFLOPS/W to enable embedded computing in unmanned aerial vehicles (UAVs). As a result of analyzing a typical UAV workload, wide-angle motion imaging (WAMI), we are able to specify a new architecture to enable as much computing on board the UAV as possible.*

This paper presents the proposed heterogeneous architecture, which includes: 1) accelerators to enable extreme energy-efficiency for key kernels; 2) throughput accelerators, such as GPGPUS or SIMD engines, for highly parallel work; 3) near-threshold parallel general purpose processors to facilitate path-divergent parallel code; and, 4) voltage boosting techniques to improve the speed of the general purpose processors to enable single-thread computation.

Computational Pyramid

In order to design the architecture for a UAV system, first the workloads must be characterized. For this task we explore the wide-angle motion imaging (WAMI) application process. In this image processing pipeline, shown in Figure 1, a high-resolution camera takes images with billions of pixels several times per second. These images are then analyzed to identify objects of interest (e.g. cars). Once the objects have been identified they are tracked between images to determine their tracks (e.g. a car's path). These tracks are then classified into events (e.g. a car doing a u-turn). Finally, a machine learning algorithm is used to identify threats based on combinations of the events (e.g. coordinated attacks). Figure 2 shows an illustration of the WAMI algorithm annotated with relevant data and descriptions. The goal of the PERFECT program is to integrate as much of the computational pyramid on board the UAV as possible in a futuristic 7nm process. The more work done on board decreases the amount of data each UAV must stream to the ground. This increases the number of UAVs that can broadcast in the same area, and decreases the probability of detection and interception.

The architecture we propose is targeted to handle the diverse types of behavior at each level of the pyramid. We begin by exploring the amount and type of parallelism available at each level of the pyramid. At the bottom of the pyramid the amount of data is enormous, as each image is billions of pixels. However, the amount computation per pixel is small. The types of kernels run at this level of the

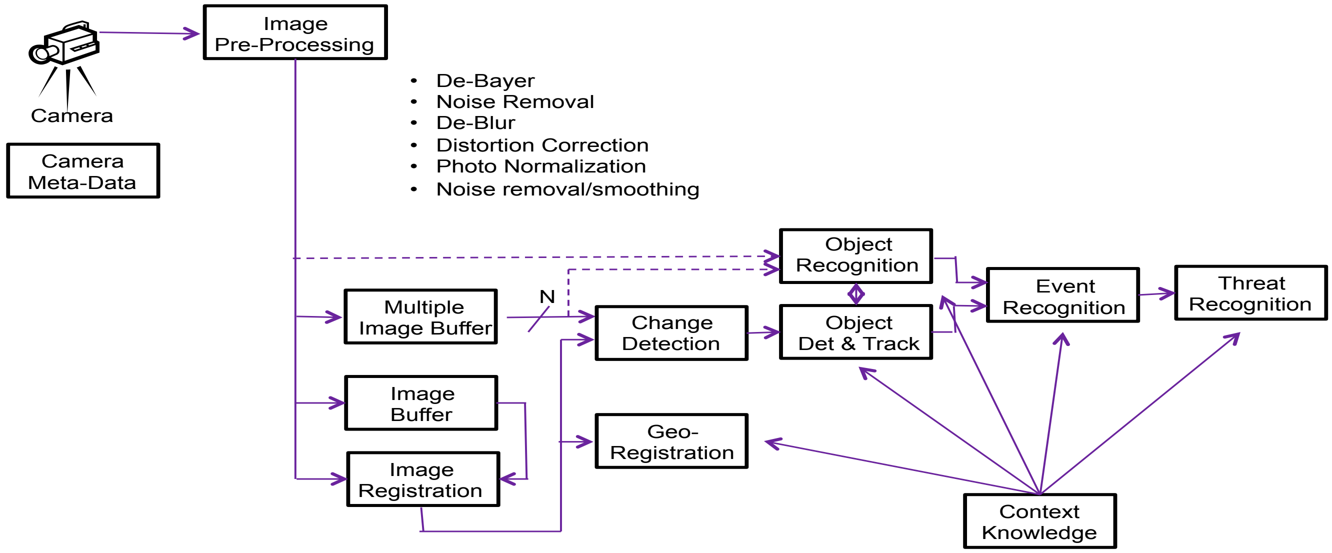

Figure 1: Processing Pipeline flow for Wide-Angle Motion Imaging (WAMI).

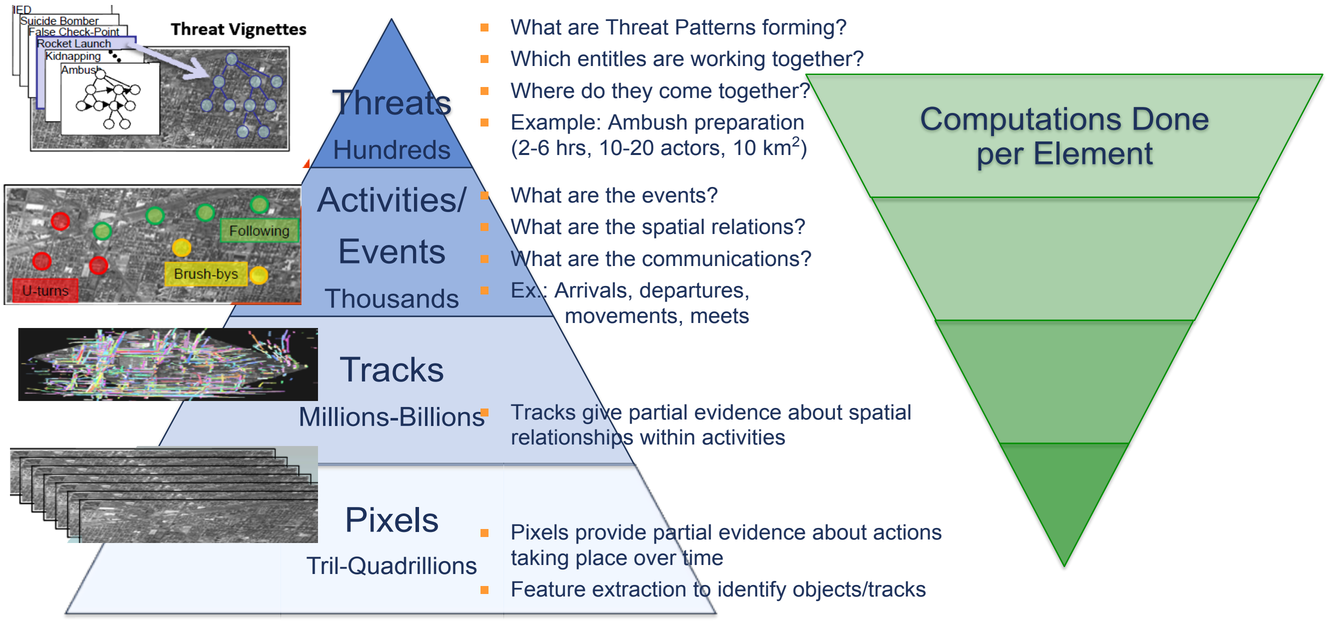
IED
Suicide Bomber
False Check-Point
Rocket Launch
Kidnapping
Ambush
Threat Vignettes
Following
Brush-bys
U-turns
Threats
Hundreds
Activities/
Events
Thousands
Tracks
Millions-Billions
Pixels
Tril-Quadrillions
What are Threat Patterns forming?
Which entitles are working together?
Where do they come together?
Example: Ambush preparation
(2-6 hrs, 10-20 actors, 10 km^2)
What are the events?
What are the spatial relations?
What are the communications?
Ex: Arrivals, departures, movements, meets
Tracks give partial evidence about spatial relationships within activities
Pixels provide partial evidence about actions taking place over time
Feature extraction to identify objects/tracks
Computations Done per Element

Figure 2: Characterization of wide angle motion imaging (WAMI) workload for unmanned aerial vehicles (UAV). The amount and type of parallelism changes at each level of the pyramid.

pyramid are debayer, deblur, image registration, etc. Several of these kernels can be easily mapped to specialized accelerators that operate on large arrays of data. As we move up the pyramid the amount of data decreases, however the work done per each data point increases. As we move up the pyramid we leverage accelerators for general purpose throughput computing, similar to GPGPUs. These throughput accelerators are programmable, but all operate on the same control-flow path. Continuing up the pyramid the data continues to decrease and the computation per element continues to increase. In addition the code tends to become more control-code dominated, with many divergent paths. This level of the algorithm doesn't easily map to throughput engines and needs parallel processors each with their own control-code. Finally, near the top of the pyramid we run into less parallelism and more need for single thread performance. In these cases we need a single high-performance core to get the work done quickly. In the following section we will detail the architecture that targets this diverse set of behaviors in an energy-efficient manor.

Architecture

The architecture we propose to handle the WAMI class of workloads is heterogeneous and targets the differing amounts and styles of parallelism that the workload exhibits. Figure 3 presents a diagram of our architecture. We start by building on our previous experience with 3D-stacked near-threshold processors, Centip3De [1]. The system is created by stacking a mix of DRAM and Non-Volatile Memory in a single 3D-Integrated stack with logic layers on top. We focus on better ECC designs to tolerate both hard and soft errors in the memory system [2]. The cores are built in heterogeneous clusters that are interconnected through a novel low-power high-radix crossbar [3].

Each core cluster is comprised of three components that target the 4 styles of parallelism exhibited in the WAMI workload. First, the cluster contains coherent accelerators. These accelerators target key kernels to provide extreme energy efficiency for tasks at the bottom of the pyramid. The accelerators are kept coherent with the other processing elements in the cluster to avoid the power and latency that would be required to DMA data in non-coherent accelerators. Initial work has begun on accelerators for FFT and De-blur. Second, the system contains a SIMD unit for throughput acceleration. This processing element targets data-parallel work that exhibits no or small amounts of path divergence in the control code. By working on several elements in parallel SIMD lanes, the cost of fetching and decoding instructions can be amortized across many lanes, reducing power. The initial kernels we are targeting for the SIMD unit are de-bayer, image registration, difference calculation, etc.

Finally, the last processing element targets the final two forms of workload characteristics. This processing element we call a virtual fat core architecture [1, 4]. Here we take a general purpose processor and create a near-threshold version and place many in parallel. These cores are able to work on parallel work that is path divergent. For example when walking a binary-tree to find elements, each core walks the tree looking for an element and takes a unique path. Finally, when a single-threaded kernel or a

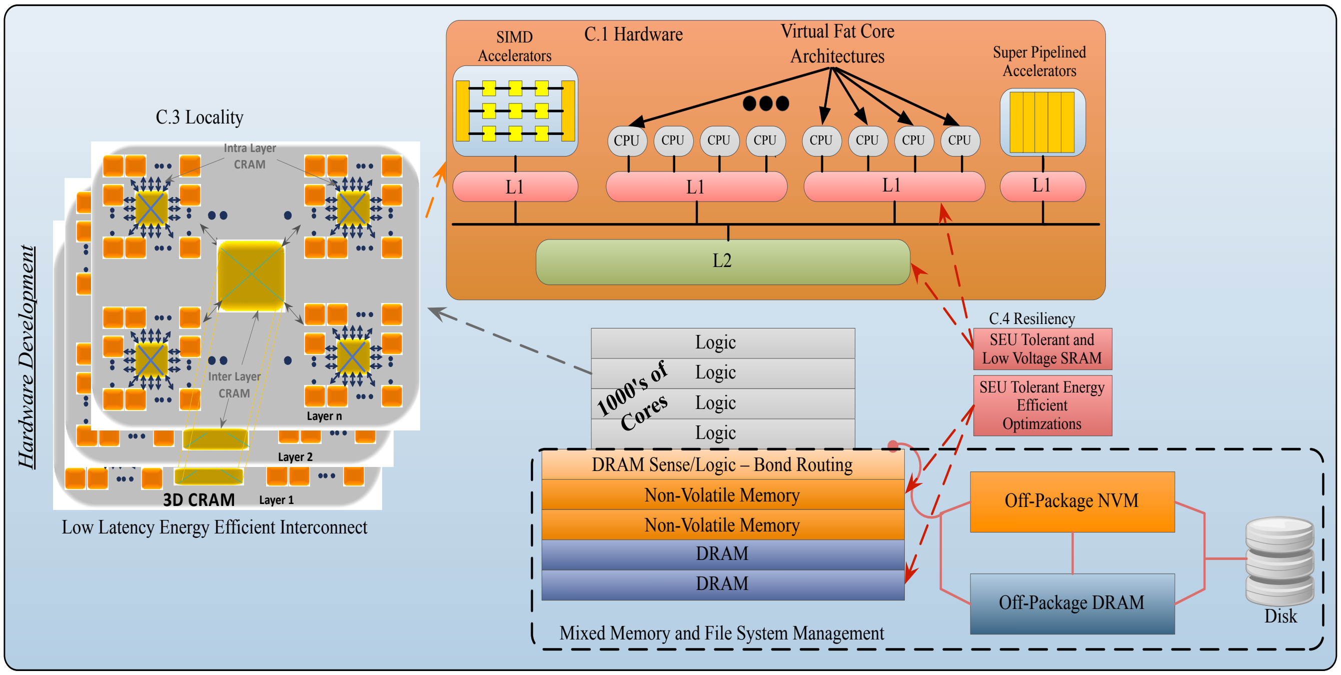

Figure 3: Proposed architecture diagram. The system is a 3D-stacked near-threshold architecture [1]. The stack includes techniques to improve soft and hard error protection through circuit techniques and ECC [2]. The core clusters are made of three elements to target different forms of parallelism. The processing elements are connected through a high-radix low-power crossbar [3].

bottleneck appears in the path-divergent code, the near-threshold cores can voltage boost the remaining core, using our new boosting technique [5], to appear as a much faster core (hence virtual fat core). To make room for the virtual fat core in the thermal budget some of the remaining near-threshold cores may need to be power-gated.

Initial 7nm Design and Estimates

An initial analysis of the workload requirements in terms of number of processing elements, inter-kernel data bandwidth, and intra-kernel data storage yielded the following design. A cluster comprised of an FFT accelerator, one throughput accelerator unit with 32 integer lanes and 16 floating-point lanes, and four in-order virtual fat-core CPUs. The local L2 cache was sized to 9MB. A total of 1,400 clusters are included in the chip. Existing designs of each component was scaled using ITRS roadmap projections and predictive technology models to 7nm. The total chip area that results is around 2,000mm^{2,} spread across 4 3D-stacked layers each of 500mm^{2}. The total computational support of the system is 11,900 GFLOPs, assuming a load/store rate of 30%. The total power of the system is estimated at 51 Watts. This yields a total efficiency of 233 GFLOPS/W, well above the 75 GFLOPS/W requirement of the PERFECT program. Of course this is raw processing power, the next stage of the PERFECT project is to determine how mapping the applications to this architecture de-rate this number. Depending on the amount of serial computation, communication overheads, and cache miss penalties this number will be reduced. Our belief is that with such a large margin (3.1x) the architecture will still meet the requirements.

Acknowledgements

The work depicted in this paper was sponsored (or partially sponsored) by Defense Advanced Research Projects Agency (DARPA) under agreement #HR0011-13-2-0006. The content, views and conclusions presented in this document do not necessarily reflect the position or the policy of DARPA or the U.S. Government, no official endorsement should be inferred.

References

1. D. Fick, R. G. Dreslinski, B. Giridhar, G. Kim, S. Seo, M. Fojtik, S. Satpathy, Y. Lee, D. Kim, N. Liu, M. Weikowski, G. Chen, T. Mudge, D. Sylvester, D. Blaauw. *"Centip3De: A 3930 DMIPS/W Configurable Near-Threshold 3D Stacked System With 64 ARM Cortex-M3 Cores."* IEEE International Solid-State Circuits Conference (ISSCC). February 2012, pp. 190-191.

2. B. Giridhar, M. Cieslak, D. Duggal, R. Dreslinski, H. Chen, R. Patti, B. Hold, C. Chakrabarti T. Mudge, D. Blaauw."*Exploring DRAM Organizations for Energy-Efficient and Resilient Exascale Memories*", Proc. of Super Computing (SC) 13, November 2013, 12pp.

3. S. Satpathy, K. Sewell, T. Manville, Y. Chen, R. Dreslinski, D. Sylvester, T. Mudge, D. Blaauw. *"A 4.5Tb/s 3.4Tb/s/W 64×64 switch fabric with self-updating least recently granted priority and quality of service arbitration in 45nm CMOS."* IEEE International Solid-State Circuits Conference (ISSCC), February 2012, pp. 478-479.

4. B. Zhai, R. Dreslinski, D. Sylvester, T. Mudge, D. Blaauw. *"Energy Efficient Near-threshold Chip Multi-processing."* International Symposium on Low Power Electronic Design (ISLPED). August 2007.

5. Nathaniel Pinckney, Matthew Fojtik, Bharan Giridhar, Dennis Sylvester, and David Blaauw, *"Shortstop: An On-Chip Fast Supply Boosting Technique,"* IEEE Symposium on VLSI Circuits (VLSI-Symp), June 2013.