Computing in Memory Architectures for Digital Image Processing

Luke Roth and Lee Coraor
The Pennsylvania State University
Dept. of Computer Science & Engineering
220 Pond Lab, University Park, PA 16802
roth@cse.psu.edu

David Landis, Paul Hulina, and Scott Deno
The Pennsylvania State University
Center for Design & Computing
11 EE West, University Park, PA 16802
dll2@psu.edu

Abstract

Continuing improvements in semiconductor fabrication density are enabling new classes of System-on-a-Chip architectures that combine extensive processing logic and high-density memory. Many of the capabilities of these new architectures can be custom tailored to the demands of real-time digital image processing. This paper evaluates several candidate designs, using the criteria of image processing performance, flexibility, manufacturability, and fabrication cost.

1 Introduction

1.1 Computing-in-Memory Overview

The rapid growth in silicon fabrication density has spawned computing architectures that would not have been imaginable ten years ago. In 1989, a state-of-the-art microprocessor such as the Intel 386 contained about 275,000 transistors, and ran at a maximum clock speed of 33 MHz. Today's equivalent CPU, Intel's Pentium II Xeon, contains nearly 7.5 million transistors, plus up to 62 million transistors in its 1MB L2 cache on a second die, and has a clock speed of 450 MHz. By 2009, the Silicon Industry Association expects a high-end microprocessor to contain approximately 84 million logic transistors. DRAM density is increasing at an even more rapid pace. By 2009, state-of-the-art DRAMs are expected to have a capacity 64 times as large as today – 16 Gbits, or 2Gbytes on a single silicon chip.

This additional silicon real estate provides many new design opportunities. General-purpose microprocessors are integrating more and more memory onto the die in the form of multiple-level SRAM caches. Digital signal processors are going to larger word sizes, floating point capabilities, and integration of multiple DSPs and local SRAM on a single die.

Fabrication density is now becoming large enough that entire complex problem sets can be mapped into the memory space of a single DRAM. This fact, plus the enhanced fabrication capabilities of today's silicon fabrication lines, opens up the possibility of greatly increased performance from "system-on-silicon" designs. It is now possible to integrate any combination of CPU cores, DRAM, custom logic, FPGA cells, and analog circuitry - all of the necessary components for a wide variety of computing systems - on the same die. With today's fabrication technology, a single chip containing 16 MB of DRAM and 500K gates of custom logic - sufficient for a RISC CPU core - could be integrated into a single die. This could provide many benefits in power consumption, small size, and cost for handheld and embedded applications. For real-time image processing applications, eight 1-MB frame buffers could be integrated together with 1M gates of custom logic for high-bandwidth, compute-intensive applications. This would provide many benefits – increased system integration, lower power consumption, smaller physical size, and improved memory bandwidth and latency.

1.2 Why Digital Image Processing?

Many applications have been proposed for combined logic/DRAM designs – from the parallel processing PPRAM project underway at Kyushu University [1] to the IRAM project at the University of California-Berkeley [2] that is aimed at the workstation CPU market. Our research focuses on Digital Image Processing (DIP), one application that seems well-matched to the benefits and drawbacks that are characteristic of a CIMA (Computing-in-Memory Architecture) design.

Digital image processing generally takes a specific digitized image, represented by a two-dimensional array of values, and transforms it into a form that is more suited for its intended purpose. This "intended purpose" could be transmission over a low-bandwidth network, in which case the application would involve compressing and decompressing the image for faster transmission. Alternatively, it could be restoring scanned images of old, faded photographs, in which case a complex string of filters may be applied. Applications frequently require that this processing be completed in real time, as when the image data is being generated by a camera or other sensor and must be displayed to the user. This leaves a fixed amount of time – typically 1/30 of a second – for the processing of a complete image. While the number of possible image processing algorithms is quite large, they can generally be decomposed into simpler operations that fall into several of the generic categories identified in the following subsections.

The most obvious benefit of a CIMA design for DIP applications is the increased memory bandwidth and lower memory latency available on-chip. If an on-chip DRAM could transfer 128 bits every 6 nanoseconds (performance comparable to that claimed by current first-generation mixed fabrication processes [3]), an entire 1MB frame buffer could be streamed through memory in less than 1/2000 of a second. This corresponds to a pixel processing rate of 2.67 billion 8-bit pixels/second. Of course, this requires a processing unit fast enough to keep up with this tremendous bandwidth. An optimal CIMA design must provide a great deal of computational parallelism to exploit the full bandwidth available. At a clock rate of 166MHz, at least 16 8-bit pixels are available to be processed each cycle.

This increased bandwidth, which is critical for DIP applications due to the memory-intensive nature of most DIP algorithms, can be exploited in a CIMA design. For example, one study using traditional DSP's and a variety of common algorithms – threshold detection, erode/dilate, median, convolution, and others – reported a ratio of up to two memory accesses for each floating point calculation [4].

The lack of temporal locality in most DIP applications makes most caching systems ineffective. Because of this, our research focuses on CIMA designs using little or no cache, typically using only a small instruction cache and no data cache. While this might adversely affect performance on those few DIP algorithms which do show temporal locality, this is largely compensated for by the low latency of the DRAM.

While temporal locality is limited in most DIP applications, spatial locality is often quite high – if a certain pixel of the frame buffer is accessed, the most likely next accesses are its immediate neighbors. With a bus width of 128 bits, a window of up to 16 pixels could be loaded in a single DRAM cycle. Depending on the addressing scheme, this could be a 4x4-pixel window sufficient for a simple filter, or one line of up to 16 pixels from a larger window.

Due to the lower logic density of a mixed DRAM/logic manufacturing process, it is impractical to implement a CIMA design with all the functionality of one of today's leading edge microprocessors together with enough DRAM to provide sufficient program and image data space. Fortunately, because of the small data sizes associated with most DIP operations (8-16 bit), the computational power of a modern 32- or 64-bit processor with floating-point math capabilities is not needed. Instead, an array or vector of simpler 8-bit processing elements (PEs) tailored to the specific needs of DIP applications is sufficient.

1.3 Paper Overview

This paper starts with a brief description in section 2 of the most important factors – both performance benefits and obstacles to manufacturing – that must be considered in the evaluation of a proposed CIMA architecture. Section 3 continues with an evaluation of four architectural styles that are fairly representative of current CIMA proposals. Finally, section 4 presents some conclusions and a description of future work in this area.

2 Important Issues for CIMA Designs

2.1 Performance Issues

The ability to integrate DRAM and computing logic on the same Integrated Circuit (IC) die gives CIMA designs several important advantages. First and most obviously, the physical size and weight of the overall design can be reduced. As more functions are integrated on each chip, fewer chips are needed for a complete design.

Second, the integration of a large memory onto the same die as the processing logic provides the opportunity for designs with much higher bandwidth than would be practical in a traditional design. By running a 256-bit bus from a DRAM bank, it would be possible to achieve a bandwidth of up to 4.8GB/second from each memory bank present on the chip. For a CIMA system with four banks of memory, the total available bandwidth would be 19.2GB/second. This is more than an order of magnitude larger than that available in traditional systems. For comparison, the main memory system of an AlphaServer 8200 provides a maximum memory bandwidth of only 1.2GB/second.

By eliminating the need for driving signals off-chip, the power consumption of the memory system is also greatly reduced. This is clearly beneficial for contemporary portable and battery-powered applications. In the future, this benefit may become critical for traditional computing systems as on-chip power consumption rapidly becomes a performance limiting factor.

Finally, by integrating DRAM memory onto the die it is possible to tailor the size, memory word width, and bus width to an extent not possible with traditional DRAMs, in which size and width are typically fixed at powers of 2. As an example, a dedicated frame buffer for an image processing unit may require storage for one 640x480 resolution, 8-bit greyscale image, which for performance reasons shold be able to fetch at least four pixels at a time - a total of 2.34Mbits, with a 32 bit bus. Using conventional DRAMs, a design would need to integrate two 2Mbit, 16-bit wide DRAMs to fulfill this requirement, of which over 40% is wasted space, or another similar configuration.

2.2 Manufacturability Issues

The mixing of DRAM and logic on the same die presents several new design and processing challenges. High-volume semiconductor manufacturing processes are highly optimized for the particular characteristics and features demanded by the product, e.g. high switching rates for logic processes, and long retention times for DRAM processes. A hybrid manufacturing process must of necessity compromise between these two characteristics, leading to sub-optimal performance – estimates have ranged from a 30% to 100% decrease in overall logic speed [5]. However, as recently as August 1998, several major manufacturers (Siemens and IBM) have announced first-generation mixed processes that claim little to no reduction in logic performance [6,7].

Traditional stacked-capacitor DRAM cells create a non-planar surface which limits the number of metal interconnect layers. This, in turn, limits the density of the logic in the design and increases die size and cost. The use of trench-capacitor DRAM cells alleviates this problem, at the cost of a slightly more complex set of masks. However, this technology may not be available in many logic fabrication processes.

DRAM manufacturers take advantage of the tremendous regularity of the DRAM array to provide a high level of manufacturing defect tolerance by utilizing redundant rows and columns of DRAM cells. This increased defect tolerance allows them to use less conservative design rules, which improves layout density. The logic sharing the die in a CIMA design will not have this same level of redundancy, reducing overall IC yield.

While system power consumption will decrease due to the elimination of external driver circuitry and buses, on-chip power consumption will typically increase due to the added high-speed logic. This could be a problem for the integrated DRAM, since higher junction temperatures reduce DRAM retention times.

The high-speed logic of the design will also generate a much larger amount of digital noise than is typically present on a DRAM die. As the sense amplifiers rely on detecting very small voltage swings (on the order of millivolts), this could lead to soft errors from the DRAM.

Finally, there are some obstacles to overcome from a market perspective. The DRAM memory market has traditionally been very commodity-oriented: components and interfaces are standardized among manufacturers, and differentiation is achieved primarily through memory size and pricing. Because of this emphasis, a significant benefit is required in order to justify even a small increase in cost, and design improvements requiring a change in the basic interface are integrated at a very conservative pace. This can be seen in the slow adoption of recent high-speed memory architectures such as RAMBUS which, while promising a large performance increase, carries a significant cost overhead and is accessed differently than traditional DRAM.

3 CIMA Architecture Evaluation

The range of design characteristics that fall under this architectural style is quite rich and varied. Over the past few years, many architectures have been proposed that each embody a distinct set of the advantages and disadvantages that CIMA offers. Of these, four major design styles represent the majority of the variations on the CIMA theme: the Processor-In-Memory, the Vector DRAM processor, the Multiprocessor-on-a-chip, and the merged CPU / FPGA / DRAM. Figure 1 contains four block diagrams, which serve to illustrate each of the basic internal organizational structures of these four CIMA architectural styles.

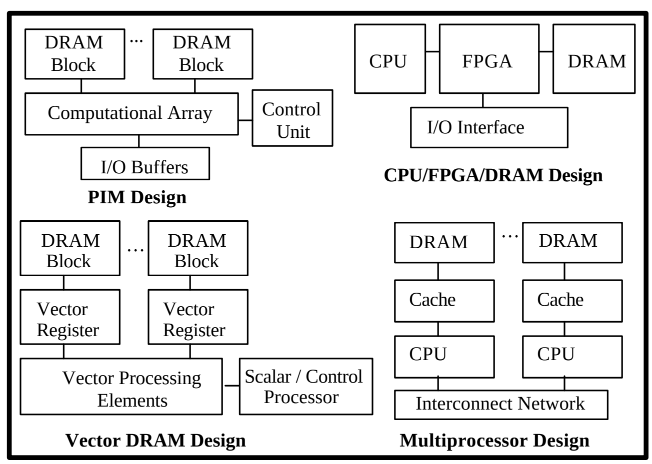

Figure 1. Block Diagrams of CIMA Architectures

3.1 Processor-In-Memory Designs

3.1.1 **PIM Overview.** One of the first of the new generation of CIMA designs to be proposed and built was the C-RAM design [8], developed by the University of Toronto in 1992. A similar design, the Terasys PIM array [9], was announced in 1995. This design style features a large array of simple computation elements (typically over 1,000) which are built into the DRAM arrays. These processing elements are usually integrated at the output of the sense amplifiers and are controlled by a single control unit. Because of this single control unit, this architecture operates as a SIMD (Single Instruction, Multiple Data) processor – in any given DRAM access cycle, every processor in the array executes the same instruction on its own local data.

3.1.2 **PIM Performance.** This style of CIMA design is generally the most capable of exploiting the enormous on-chip bandwidth of the DRAM, as the computation elements are integrated directly into the DRAM outputs. From an architectural standpoint, this design is also the simplest and can achieve the highest theoretical performance. In our performance projections, this design achieved the best performance of any of the systems reviewed on a number of easily parallelizable tasks such as averaging filters and thresholding. For performance figures for this and later architectures, see Table 1 below (For details on how these numbers were obtained see [10]).

Table 1. Performance of CIMA Alternatives

Architecture Type	Histogram Time	Filter Time	Threshold Time	Necessary Die Size
Traditional RISC	19ms	67ms	17ms	300mm^2
Processor-in-Memory	71ms	0.6ms	0.1ms	350mm^2
Vector DRAM-333MHz	28ms	50ms	0.9ms	430mm^2
Single-Chip Multiprocessor: 4 processors at 333MHz	8ms	25ms	6.5ms	450mm^2
Merged CFD 16K LE's no fast-carry	37ms	6ms	1.0ms	500mm^2
Merged CFD 16K LE's fast-carry logic	12ms	1.2ms	1.0ms	530mm^2

To date, these types of massively parallel, simple-processing-element SIMD designs have only been successful on a limited set of applications that can be easily parallelized. For applications with a significant amount of serial computation, the speedup from this architecture is very limited. On the more complex histogram algorithm, for example, this architecture had the second-worst performance of the architectures reviewed. For this reason, most PIM designs best function as co-processors, executing the functions that they are optimized for, while leaving applications that are mostly serial to the host processor. However, image processing may prove to be one area where this type of parallelism exists in a large percentage of the target applications.

3.1.3 **PIM Manufacturability**. The Processor in Memory approach has some serious practical drawbacks associated with the close proximity of the processor logic and the memory array. Even though this design has the simplest architecture, some serious complications arise in the actual design and fabrication. Most DRAM block cores are highly optimized, and can only be modified with difficulty. While prototype systems developed using in-house designs may be modified easily, it is difficult to achieve the same level of integration with the highly optimized designs used by the large DRAM manufacturers.

Also, because of the tight integration with the DRAM cell, there is space for processing elements of only a limited size. The C-RAM and PIM designs both contain only single-bit processors, significantly limiting the set of applications that the architectures can accelerate. Note, however, that a larger, more capable functional processing unit design would consume more die area, reducing the number of parallel functional units which can be integrated on a single IC.

The tight integration of the processing logic to the DRAM core also provides little insulation between the design logic and the DRAM cells, which may create problems with digital noise and heat production. However, the simple bit-serial processing elements are likely to create less noise than a large, high-speed datapath, reducing the magnitude of this problem.

Since these simple processing elements are closely tied to the outputs of the DRAM arrays, their speed will necessarily be limited by the cycle time of the DRAM array – a rate much slower than a stand-alone CPU design. This has the advantage of keeping power consumption and digital noise low. Also, this aspect of the design makes the lower logic switching speeds available in a hybrid logic/DRAM process fairly irrelevant. The cost of this type of CIMA design is likely to be lower than any of the other designs evaluated here. The added die area required by the processing logic is fairly small, and the highly parallel nature of the design provides the opportunity to use logic redundancy to improve yield.

From a market standpoint, this design suffers from being somewhat of a hybrid design – for a memory design, the extra logic and die area will increase production costs above that of a traditional DRAM, limiting its appeal in the DRAM market. However, the programming difficulty of the architecture and the limited application set may also keep it from being a mainstream processing solution.

3.2 Vector DRAM Designs

3.2.1 **Vector DRAM Overview.** The CIMA design that has received the most attention has been the IRAM project at the University of Berkeley. This ambitious design attempts to take advantage of the high bandwidth and low latency offered by on-chip DRAM by integrating a complete vector processor and a large portion of the main memory of the system onto the same die. This design benefits from moving the processing logic further away from the DRAM, both physically (by segregating the memory and DRAM logic on the die) and conceptually (by providing a more conventional programming model), which eliminates some of the main problems of the PIM design. However, this distance also removes one of the primary benefits of the CIMA design, as the vector architecture is not as capable as the PIM architecture of exploiting the full bandwidth available from the on-chip memory. The design of the vector processor by the IRAM project is rather aggressive, building the equivalent of a full Cray T90-class vector processor into its design. For DIP applications, a more modest design with simpler 8/16 bit pipelines may be more suitable.

3.2.2 **Vector DRAM Performance.** The vector processor programming model is well understood, and there has been a great deal of research into vectorizing the most common types of applications. Consequently, the performance projections of the vector processor were much more balanced for this processor than for the PIM processor. However, the vector processor could not match the performance of the PIM on the most highly parallel algorithms.

3.2.3 **Vector DRAM Manufacturability.** This architectural style manages to avoid many of the drawbacks of the PIM architecture by allowing the datapath logic to be physically separated from the DRAM arrays. This isolates the sensitive DRAM arrays from the noisiest parts of the logic circuit, and allows existing DRAM cores to be embedded in the design without extensive modification. It also slightly increases the memory latency, as memory data must now be routed from the memory to the vector processor. However, this delay is still significantly less than that of going to off-chip memory. To reduce this penalty as much as possible, the IRAM design uses a crossbar interconnect between the DRAM arrays and the processor load/store ports.

This design is likely to be much more expensive from a manufacturing standpoint than the PIM array for several reasons. First, a full-fledged vector processor – even one with smaller 8/16 bit vector pipelines – will require much more die area than the array of simple logic elements in the PIM design. Second, the percentage of die area dedicated to non-redundant logic will be higher, decreasing manufacturing yield. Because of the regular nature of the array of vector units, a "spare" vector unit could provide some logic redundancy, increasing the yield at the cost of a slight increase in die size (approximately 1%).

Unlike the PIM design, in which the maximum speed was limited by the DRAM cycle time, the speed of this type of processor is limited only by the transistor switching time provided by the fabrication process. This switching speed is likely to be significantly slower in a process optimized for DRAM; however, a more sophisticated process such as the SA-27E Embedded DRAM process recently announced by IBM [11] which provides separate gate oxide thicknesses for logic and DRAM transistors would minimize this speed penalty.

3.3 Multiprocessor-On-A-Chip Designs

3.3.1 **MOC Overview.** Between the two extremes of one single powerful processor and an array of tiny one-bit processors there is a wide range of possible architectures. One option that has been explored is to integrate a small number of moderately powerful processors onto the same die, each with a moderate amount of DRAM main memory. The on-chip interconnect between the processors provides extremely low-latency communication, allowing efficient parallel execution of a wide variety of applications. An example of this type of architecture is the PPRAM architecture being developed at Kyushu University [1].

3.3.2 **MOC Performance.** By avoiding the central-control-unit bottleneck that limits the performance of many applications that cannot be easily parallelized, this type of architecture can accelerate performance on a wider range of applications than either the PIM or Vector DRAM design. However, the cost (die size) per functional unit is quite high, as an entire CPU with a moderate amount of memory is required for each processing unit. This limits the number of functional units on-chip to a small number (2 or 4 in first-generation designs). The maximum speedup of this architecture is then fairly limited compared to the 1000+ processing nodes of the PIM design. Because of the limited speedup on massively parallel applications, which are fairly common in image processing, this type of architecture may be more suited to traditional workstation or server applications than to DIP.

3.3.3 **MOC Manufacturability.** This architecture faces several obstacles when implemented in a current-generation fabrication process. Even with today's tremendous fabrication density, it would be impracticably expensive to manufacture a single-chip multiprocessor design with a total of more than roughly 32MB of DRAM main memory. With each additional processor, the amount of main memory available for each processor will shrink proportionately – a four-processor chip would have only 8MB per processor. This amount of memory will be insufficient for many problem sets, forcing a high percentage of inter-processor and off-chip communication and reducing the overall performance of the design. As fabrication technology improves, this will become a less important concern.

The problems posed by power consumption and digital noise, while still present, can be alleviated somewhat by careful physical design and layout. As in the Vector DRAM design, the processing logic is somewhat independent of the DRAM arrays, with memory accesses taking place across a local memory bus instead of integrating processing logic directly at the DRAM outputs.

Because of the large amount of resources required for each processing node, this design is likely to be the most expensive of the designs that we have evaluated in terms of total die area. While the DRAM cores can tolerate manufacturing defects, there is still a large percentage of the die – both processing logic and interconnect between the processing elements – that is susceptible to manufacturing defects, lowering yield. Providing redundant interconnect and cache lines, if possible, could improve yield at the expense of an even larger die area. As in the Vector DRAM design, the limiting factor in the speed of the individual processing units will be determined by the fabrication process; fabrication in a process that has been optimized for embedded DRAM will dramatically improve performance and density.

3.4 Merged CPU/FPGA/DRAM Designs

3.4.1 **Merged CFD Overview.** With the rapid increase in the use of reprogrammable SRAM-based FPGA's has come a new architecture possibility: merging traditional CPU and DRAM technology with programmable logic to create reconfigurable processing units that can adapt to changing processing needs. There are a number of variations to this design style, from the DRAM/FPGA chip proposed by NEC Corporation (capable of reconfiguring itself dynamically in under 100ns [12]) to the Dynamic Instruction Set Computer (DISC) being developed at Brigham Young University [13]. Generic FPGA-only designs have been used in many digital signal processing and image processing applications. However, the combination of FPGA resources with a traditional CPU provides a guarantee of reasonable performance on algorithms which are not suited for FPGA processing, as well as a simpler means of programming and coordinating the system.

The addition of DRAM to the FPGA design provides a large, fast memory space sufficient to hold many FPGA configurations. This offers fast reprogrammability as well as the capability to store reasonable amounts of image data. Current IC fabrication densities permit only a limited amount of functionality to be integrated on a single CIMA die. A fairly large ($300mm^2$) die in today's 0.25 micron technology would be able to accommodate roughly 8MB DRAM, 2500 FPGA logic elements (43,000 equivalent gates), and a reasonably powerful RISC processor (of about 3M transistors). This estimate of IC density is extrapolated from data provided in figures from [2, 6,14]. An equivalent-cost, $340mm^2$ die in next generation's 0.18-micron technology, however, could contain a RISC processor together with a more reasonable 8192 logic elements (each a 3-input lookup table or equivalent design), equivalent to approximately 170,000 gates, and 16MB of DRAM.

3.4.2 **Merged CFD Performance.** Because of its flexibility, this hybrid design provides some of the most impressive performance scores of any of the architectures reviewed on a wide variety of applications. In the few applications which the FPGA array could not accelerate, the simpler CPU core could not match the performance of the Vector DRAM design. However, given sufficient FPGA resources this design outperformed all but the PIM design on the more easily parallelizable algorithms.

3.4.3 **Merged CFD Manufacturability.** FPGA designs have a relatively low functional density (10 to 100 times less dense than full-custom designs [15]). Thus, this type of design requires a large die area to be able to integrate an adequate amount of main memory and a large enough FPGA array to handle a wide range of possible applications. Consequently, it is likely that at least one generation of fabrication process improvements will be required before a design based on this architecture becomes economically feasible.

The large area occupied by this type of design directly implies a higher cost to manufacture. However, the regularity of FPGA designs may allow redundancy to be used to tolerate manufacturing defects, as is typical in DRAM arrays. While there are many obstacles to implementing this type of error-tolerance (limitations in design software, irregular signal routing times, and large FPGA cell size), this could provide a tremendous improvement in fabrication yield, as the FPGA array will occupy a large percentage of the total die area.

Depending on how closely linked the FPGA core is to the embedded DRAM, there may be significant issues with the digital noise from the FPGA affecting the performance of the DRAM. However, the power-per-unit-area of FPGA arrays tends to be much lower than that of custom logic, except for during reconfiguration [14]. This is mostly due to the lower percentage of transistors that are switching at any given time. The majority of transistors in a typical FPGA cell are statically storing the configuration of the cell, not dynamically determining the outputs of the cell.

4 Conclusions and Future Work

A summary of the factors described above can be seen in Table 2. While no single architecture is clearly superior in every category, it is still possible to draw some interesting comparisons. The Merged CFD design provided the best performance over the widest variety of algorithms, at a significant cost. For a small set of applications, the PIM design provides excellent performance and a low manufacturing cost, provided several manufacturing problems can be solved. The Vector DRAM design seems to be the easiest to implement, benefiting from (but not requiring) the most optimized manufacturing process. However, its performance on DIP applications is mediocre compared to the other reviewed designs.

In the long term, as fabrication density continues to improve at a rapid pace, the die size restraint will become the least important of the factors. In this context, the Merged CFD design seems to have the most promise for future designs.

Because of the apparent long-term advantage of the Merged CFD architecture, future work will focus on evaluating variations on this design style. A more quantitative analysis will be performed on these design variations using software simulation.

Table 2. Feature Comparison of CIMA Architectures

Architecture	Peak Performance	Application Flexibility	Logic / DRAM Integration	Composition (Logic / SRAM / DRAM)	Required Die Size
Traditional RISC	Poor	Good	None	63% / 37% / 0%	Average
Processor-in-Memory	Excellent	Poor	Tight	8% / 2% / 90%	Small
Vector DRAM	Fair	Fair	Average	25% / 5% / 70%	Average
Multiprocessor-on-a-Chip	Fair	Fair	Average	30% / 4% / 66%	Large
Merged CPU/FPGA/ DRAM	Good	Excellent	Variable	20% / 30% / 40%	Large

5 References

[1] K. Murakami, K. Inoue, and H. Miyajima, "Parallel Processing RAM (PPRAM)", Nov.1997 http://kasuga.csce.kyushu-u.ac.jp/~ppram

[2]D. Patterson, T. Anderson, R. Fromm, K. Keeton, ,C. Kozyrakis, R. Thomas, and K. Yelick, "A Case for Intelligent RAM: IRAM", IEEE Micro, April 1997

[3] http://www.siemens.de/semiconductor/edram

[4] "Local Memory Enhances Performance of Machine Vision Application", http://www.alacron.com/news/art04.html

[5] D. Patterson, T. Anderson, R. Fromm, K. Keeton, ,C. Kozyrakis, R. Thomas, and K. Yelick, "A Case for Intelligent RAM: IRAM", *IEEE Micro*, April 1997 pp. 34-44

[6] IBM Press Release: "IBM chip advance spurs 'system-on-a-chip' products", Feb. 22, 1999

[7] K. Schoneman, "A Modular Embedded DRAM Core Concept in 0.24 micron Technology", *Proceedings of MDTD '98*, 24-25 August 1998

[8] D. Elliot, W. Snelgrove, M. Sturm, "Computational RAM: A Memory-SIMD Hybrid and its Application to DSP", CICC '92, May 6, 1992

[9] M. Gokhale, B. Holmes, K. Iobst. "Processing in Memory: The Terasys Massively Parallel PIM Array", *IEEE Computer*, April 1995 pp.23-31

[10] D. Landis, P. Hulina, S. Deno, L. Roth, and L. Coraor, "Evaluation of Computing in Memory Architectures for Digital Image Processing Applications", CEDCC Technical Report, TR-99-01, April 1999.

[11] http://www.chips.ibm.com/asics/products/edram

[12] M. Motomura et al., "An Embedded DRAM-FPGA Chip with Instantaneous Logic Reconfiguration", 1997 Symposium on VLSI Circuits Digest of Technical Papers, June 12-14 1997 pp.55-56

[13] M. Wirthlin, B. Hutchings, "A Dynamic Instruction Set Computer", *Proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines*, April 1995 pp. 99-107

[14] Xilinx 1999 Databook p.161 ff.

[15] O. Albaharna, P. Cheung, and T. Clarke, "Area & Time Limitations of FPGA-based Virtual Hardware", *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors*, Oct. 10-12 1994 pp.184-189