ELSEVIER

Microelectronics Journal 34 (2003) 1025–1029

Microelectronics Journal

www.elsevier.com/locate/mejo

Highly configurable platforms for embedded computing systems

Frank Vahid[a,*], Roman Lysecky[a], Chuanjun Zhang[b], Greg Stitt[a]

[a]*Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA*
[b]*Department of Electrical Engineering, University of California, Riverside, CA 92521, USA*

Abstract

Platform chips, which are pre-designed chips possessing numerous processors, memories, coprocessors, and field-programmable gates arrays, are becoming increasingly popular. Platforms eliminate the costs and risks associated with creating customized chips, but with the drawbacks of poorer performance and energy consumption. Making platforms highly configurable, so they can be tuned to the particular applications that will execute on those platforms, can help reduce those drawbacks. We discuss the trends leading embedded system designers towards the use of platforms instead of customized chips. We discuss UCR research in designing highly configurable platforms, highlighting some of our work in highly configurable caches, and in hardware/software partitioning.

Keywords: Platform; System-on-a-chip; Configurable; Field-programmable gate array; Cores; Embedded systems; Low energy; Cache; Hardware/software partitioning; Architecture tuning

1. Introduction

Integrated circuit (IC) chip capacities are increasing at a tremendous rate, leading to system-on-a-chip (SOC) designs. Such capacities allow embedded computing system designers to create single-chip systems with massive functionality. Future IC technologies promise further advances in both transistor capacity and processor speeds. But at some point one begins to ask: How much is enough?

Consider the following analogy. Meeting your family's basic needs on a $20,000 annual salary would be a challenge. Increasing that salary to $40,000 would make a big difference, and $80,000 would be even better. However, at some point, working for further increases would reach a point of diminishing returns. An increase from $100 million to $200 million would be nice, but it probably wouldn't change your life much, and few would even notice the difference between $1 billion and $2 billion.

Similarly, meeting your basic embedded computing needs with a 20,000-transitor silicon budget would also be a challenge. Twenty thousand transistors (roughly the silicon budget two decades ago) are barely enough to implement an 8-bit microprocessor. Increasing the budget to 40,000 transistors would make a big difference, and 80,000 would be even better. Again, at some point, working for further increases would reach a point of diminishing returns. An increase from 100 to 200 million transistors (modern chip sizes) would be nice, but would not change most designers' systems all that much, and few designers would even notice the difference between 1 and 2 billion transistors.

Moore's law states that chip capacity doubles every 18 months. However, ASIC vendor data shows that most designs greatly underuse that capacity. Mainstream embedded systems designers are simply no longer screaming for higher-capacity chips the way they once were. One reason is that a few hundred million transistors are enough to provide plenty of computing ability. Another reason, known as the productivity gap, is that designer productivity increases have not kept pace with chip capacity increases, meaning designers often cannot create designs that utilize all those available transistors.

IC design costs are also increasing at a rapid rate. Table 1 shows sample non-recurring engineering (NRE) costs for different CMOS IC technologies [1]. At 0.8 μm technology, the NRE costs were only about $40,000. However, with each advance in IC technology, the NRE costs have increased dramatically. NRE costs for 0.18 μm designs are around $350,000, and at 0.13 μm, the NRE costs are

* Corresponding author. Tel.: +1-909-787-4710; fax: +1-909-787-4643.
E-mail address: vahid@cs.ucr.edu (F. Vahid).

0026-2692/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0026-2692(03)00171-X

Table 1
IC non-recurring engineering (NRE) costs and turnaround time

	Technology (μm)			
	0.8	0.35	0.18	0.13
NRE (K)	$40	$100	$350	$1000
Turnaround (days)	42	49	56	76

over $1 million. This trend is expected to continue at each subsequent technology node, making it more difficult for designers to justify producing an IC using these technologies.

Furthermore, designing high-end ICs that utilize the available transistor capacity requires significantly more time during design. While the productivity gap is partially responsible for longer design times, the time it takes for a design to be manufactured at a fabrication facility and returned to the designers in the form of an initial IC is also increasing. Table 1 provides the turnaround times for various technology nodes. The turnaround times for manufacturing an IC have almost doubled between 0.8 and 0.13 μm technologies. Longer turnaround times lead to larger design costs and even possible loss of revenue if the design is late to the market. Furthermore, long turnaround times become a larger burden on designers when design flaws lead to multiple respins, delaying a products entrance into the market.

The problems of increasing design costs and long turnaround times are made even more noticeable due to increasing market pressures. Market windows, the time during which a company seeks to introduce a product into the market, are shrinking. The design of new ICs are increasingly being driven by time to market concerns. Due to these concerns, design features or requirements of new systems are often modified into order to get the system to market faster.

Increases in design costs and design time as well as time to market concerns limit the number of situations that can justify producing designs using the latest IC technology. Less than 1000 out of every 10,000 ASIC designs have volumes high enough to justify fabrication at 0.13 μm [1]. Therefore, if design costs and design times for producing a high-end IC are becoming increasingly large, will high-end ICs be produced at all? Yes, but only a few designs will be able to justify doing so. There will always be systems that make use of the transistor capacity of the latest IC technology. However, for most mainstream embedded systems designers, producing a high-end IC is not feasible.

ICs that are sold in small volumes typically have a high per IC cost. On the other hand, ICs with high volume sales have lower per IC costs because the NRE design costs and other initial design costs can be amortized over the high volume. The three plots in Fig. 1 show the idea that cost per IC decreases with higher volumes due to amortization of

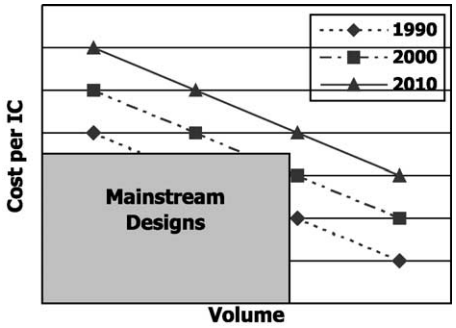

Fig. 1. Modern design technologies have such high initial design costs that they fall out of reach of mainstream design.

design costs. The plots correspond to leading edge IC technology in the 1990s, 2000s and 2010s. For higher volumes of ICs, initial design costs can be amortized over larger numbers, resulting in lower cost per IC—hence, the plots slope down to the right.

Fig. 1 also includes a shaded region illustrating the volumes and acceptable cost per IC for mainstream embedded system products. The top of the region represents the highest acceptable cost per IC, and the right side of the region represents the highest volumes we might expect to see, in mainstream systems. The figure illustrates that 1990s technologies were affordable enough to be considered by mainstream designers, while the 2010s technologies are out of range. Few systems could tolerate the high cost per IC, or have the very high volumes, to justify creating a new IC in those technologies.

High-end chips will still be produced in 2010 technologies, but will either require extremely large volume or have high costs. Thus, to achieve the high volumes, high-end ICs will more likely be produced in the form of prefabricated programmable platform ICs, which could be used across a wide variety of embedded systems.

2. Platforms

A platform IC is a prefabricated SOC that may possess one or more microprocessors, caches, memories, coprocessors, peripherals, and field-programmable gate arrays (FPGAs). Prefabricated configurable platforms have many advantages, including time to market and cost advantages. By purchasing a prefabricated platform, a designer's task shifts from designing the entire IC to programming the desired functionality onto the platform. This design approach eliminates the long turnaround times of IC fabrication and reduces NRE costs. By eliminating long turnaround times associated with manufacturing an IC as well as problematic respins, designers can get new IC designs to market faster with lower costs. Furthermore, faster time to market translates to increased market share, revenue, and ultimately profit.

3. Highly configurable platforms

Platforms need to be highly configurable, or programmable, in order to adapt to different applications and design constraints. Programmability of platforms can come in several forms, like general-purpose processors, field-programmable logic, and tunable architecture parameters like reshapable memory hierarchy, segmented bus structure, optional code and data compression schemes, and variable bit widths. Such programmability uses far more transistors than more customized designs, but with current and future IC technologies, those transistors are readily available.

Other uses of the additional available transistors can be used to develop platforms with built-in optimization capabilities. Such platforms may monitor and optimize a chip's execution, or reduce power by executing operations on specialized components. Other platform capabilities may include on-chip system exploration abilities to determine the best cache configuration, bus structure, or memory hierarchy.

At UCR, we have thus far investigated two aspects of highly configurable platforms, namely configurable cache design, and hardware/software partitioning, both for improved performance and energy consumption.

3.1. Configurable cache

Caches consume a large amount of a microprocessor system's energy—around 50% for some systems [2,3]. Caches come in a variety of shapes and sizes, varying in their total size, associativity, and line size, among other items.

Consider a cache's line size. Caches typically move data to and from off-chip memory in chunks of several bytes, perhaps 16, 32 or 64 bytes, known as line size. When a program exhibits much spatial locality, then a larger line size can reduce the number of microprocessor stalls caused by cache misses. But without spatial locality, a large line size fetches many unnecessary bytes, which not only lengthens cache fill time, but may also evict needed bytes from the cache, thus increasing off-chip memory accesses and microprocessor stalls.

Now consider cache associativity. While higher cache associativity improves hit rates, the increase in hit rate comes at the expense of increased power per access. Furthermore, the best performing cache configuration does not always have the lowest overall energy. Direct mapped caches work well on most examples and have low power per access. However, for some applications they have a very poor hit rate leading to decreased performance and high power due to many misses. On the other hand, set-associative caches have good hit rates on nearly all applications, but come at the cost of high power per access. For many applications, the increased performance does not outweigh the increase in energy consumption.

Finally, consider cache size. Large caches ensure higher hit rates across a range of applications, but wastes energy for a particular application that needs only a small cache.

Designers of mass-produced embedded microprocessors do not know what particular application will run on the chip. Therefore, designers of mass-produced parts will typically choose a particular cache that works best on average across a wide variety of applications. Furthermore, from surveying several popular embedded processors, we see no agreement on the best cache size, line size or associativity. Sizes range from no cache or just 2 Kilobytes, up to 32 or even 64 Kilobytes. Cache line sizes typically range from 16 to 64 bytes, and associativity ranges from direct mapped to eight-way set associative for different microprocessor devices.

We have therefore developed a highly configurable cache utilizing a novel technique called way concatenation, a method for way shutdown, and a configurable line size, such that the cache can be configured to adapt to a particular application [4,5]. A configurable cache could also be extended to include features such as multiple replacement policies, write through and write back policies, or an optional victim buffers to improve cache hit rates.

Fig. 2 summarizes energy results we obtained using our 8 Kilobytes configurable cache (*cfg8Kb*), compared to

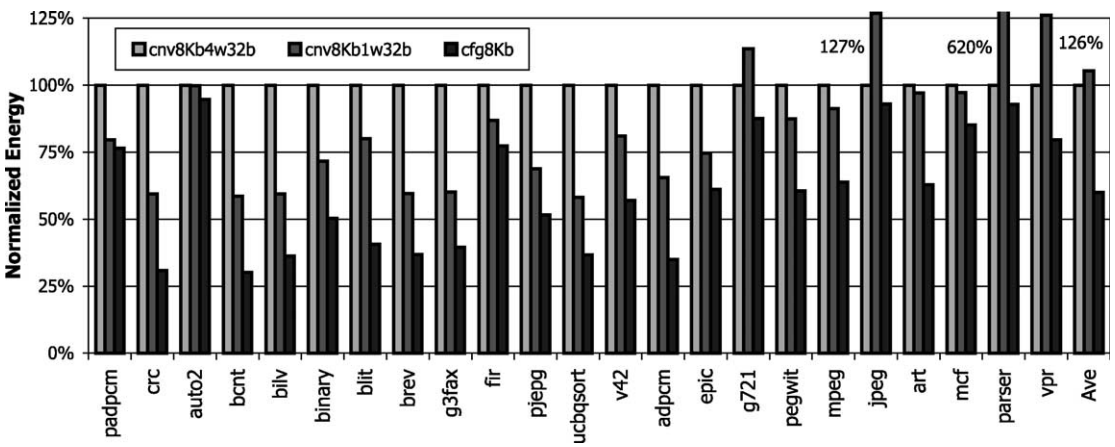

Fig. 2. A configurable cache saves memory-access-related energy on every benchmark we studied, averaging over 40% energy savings, compared to conventional four-way set-associative and direct-mapped caches.

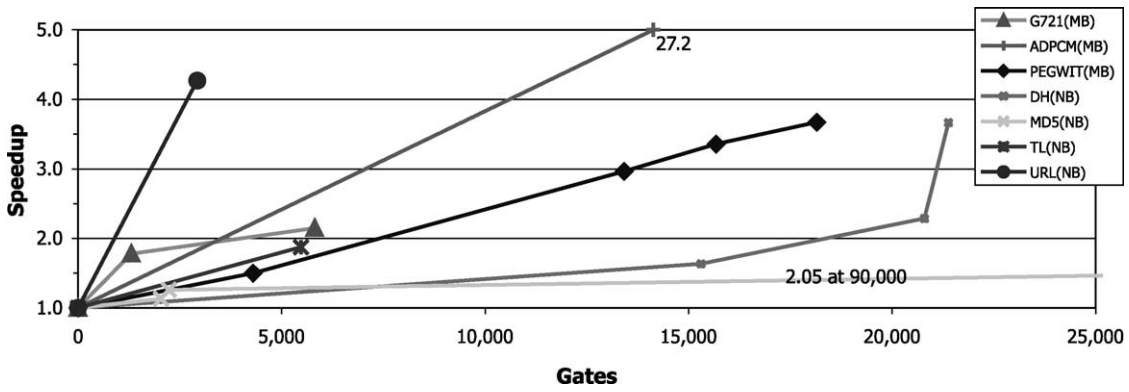

Fig. 3. Relationship between FPGA size and speedup for several benchmarks, obtained through hardware/software partitioning.

a conventional 8 Kilobytes four-way set associative cache with a 32 byte line size (*cnv8Kb4w32b*) and a conventional 8 Kilobytes direct mapped cache with a 32 byte line size (*cnv8Kb1w32b*), using benchmarks drawn from Powerstone [2], MediaBench [6], and SPEC2000. Energies are normalized to the conventional four-way cache's energy. The configurable cache saves energy over the conventional caches for *every* benchmark. The average savings of energy related to memory accesses is over 40%, and as high as 70% in several cases.

3.2. Embedded configurable logic

The availability of transistors using current IC technologies has led to the recent appearance of platforms combining a microprocessor with configurable logic. Triscend [7] provides two such platforms, the E5 and A7, which use an 8051 and ARM7 microprocessors, respectively, combined with up to 40,000 configurable logic gates. The Atmel FPSLIC [8] (field-programmable system-level IC) is a similar platform using an eight-bit microprocessor and up to 40,000 configurable gates. The Altera Excalibur [9] is a higher-performance platform, combining an ARM9 with a 2 million gate FPGA. The Xilinx Virtex II Pro [10] combines up to four PowerPC processors with 50,000 configurable gates.

One beneficial use of the configurable logic on platforms is to improve the performance of the software running on the microprocessor. Hardware/software partitioning using configurable logic is a well-known technique for achieving software speedup. Large speedups are possible because frequent regions of code may consist of many instructions, taking hundreds of cycles to execute. However, configurable logic can execute the same region of code in just a few cycles, due to the ability to perform many operations in parallel and higher efficiency for bit-level operations. In addition to speedup, we have shown that configurable logic can reduce system energy for numerous examples [11,12].

A straightforward partitioning technique consisting of moving critical loops to hardware running in the configurable logic has been shown to achieve excellent speedups. The main reason this simple partitioning technique is effective is that for most benchmarks the majority of execution time is spent in just a few loops. These loops are usually implemented with a small amount of code, implying that in most cases a hardware implementation will require little area.

Fig. 3 shows speedups we obtained from hardware/software partitioning using configurable logic (in particular, using a Triscend A7 device). The *x*-axis represents the number of gates required to achieve the reported speedups. The examples shown include benchmarks from the MediaBench [6] and NetBench [13] benchmark suites. For URL and ADPCM, big speedups are possible using a small amount of gates. Almost all of the examples achieve respectable speedups using fewer than 25,000 gates—a modest number of gates in today's era of million-gate FPGAs. Excluding ADPCM, which is an outlier, the average speedup using just 25,000 gates was 3.1.

4. Conclusions

Cost and market pressure trends point strongly towards the use of pre-fabricated platforms in mainstream embedded system design. Such platforms must be highly configurable in order to achieve acceptable performance and energy across a wide range of applications. We have investigated two aspects of such configurability. We have designed a highly configurable cache and shown that tuning the cache configuration to a particular application can yield significant energy savings. We have partitioned software among a microprocessor and same-chip FPGA and shown significant speedups and energy savings. Many other researchers are working in the direction of highly configurable architectures too.

Extensive future work remains, not only in designing configurable architectures, but also in developing automated tools to assist in finding the best configuration. Towards this end, we are investigating not only the development of traditional desktop tools, but also desktop tools that interact

with an executing platform, and even platforms that transparently tune themselves to an executing application.

Acknowledgements

This work was supported in part by the National Science Foundation (grants CCR-9876006 and CCR-0203829), a Department of Education GAANN fellowship, and by the Semiconductor Research Corporation.

References

[1] Z. Or-Bach, Panel: (when) will FPGAs kill ASICs?, 38th Design Automation Conference, 2001.
[2] L. Lee, B. Moyer, J. Arends, Instruction fetch energy reduction using loop caches for embedded application with small tight loops, Proceedings of the International Symposium on Low Power Electronics and Design, 1999.
[3] S. Segars, Low power design techniques for microprocessors, Proceedings of the International Solid-State Circuits Conference, 2001.
[4] C. Zhang, F. Vahid, W. Najjar, Energy benefits of a configurable line size cache for embedded systems, Proceedings of the IEEE International Symposium on VLSI Design, Feb 2003.
[5] C. Zhang, F. Vahid, W. Najjar, A highly configurable cache architecture for embedded systems, Proceedings of the 30th International Symposium on Computer Architecture, June 2003.
[6] C. Lee, M. Potkonjak, W. Mangione-Smith, MediaBench: a tool for evaluating and synthesizing multimedia and communications systems, Proceedings of MICRO, 1997.
[7] Triscend Corp., http://www.triscend.com, 2003.
[8] Atmel Corp., http://www.atmel.com, 2003.
[9] Altera Corp., http://www.altera.com, 2003.
[10] Xilinx, Inc., http://www.xilinx.com, 2003.
[11] G. Stitt, F. Vahid, The energy advantages of microprocessor platforms with on-chip configurable logic, IEEE Des. Test Comput (Nov/Dec 2002) 36–43.
[12] G. Stitt, B. Grattan, J. Villarreal, F. Vahid, Using on-chip configurable logic to reduce embedded system software energy, Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines, Napa Valley, April 2002, pp. 143–151.
[13] G. Memik, B. Mangione-Smith, W. Hu, Netbench: a benchmarking suite for network processors, UCLA CARES Technical Report 2001_2_01, 2001.