Configurable Multi-Processor Platforms for Next Generation Embedded Systems

David Goodwin
Tensilica Inc.
Santa Clara CA 95054
Tel : +1-123-456-7890
Fax : +1-408-986-8919
e-mail :
goodwin@tensilica.com

Chris Rowen
Tensilica Inc.
Santa Clara CA 95054
Tel : +1-408-327-7333
Fax : +1-408-986-8919
e-mail :
rowen@tensilica.com.

Grant Martin
Tensilica Inc.
Santa Clara CA 95054
Tel : +1-408-327-7323
Fax : +1-408-986-8919
e-mail :
gmartin@tensilica.com

Abstract - Next-generation embedded systems in application domains such as multimedia, wired and wireless communications, and multipurpose portable devices, are increasingly turning to multiprocessor platforms as a vehicle for their realization. But entirely fixed platforms composed of entirely fixed components lack the flexibility and ability to be optimized to the application to offer the best solution in any of these areas. Configurability at multiple levels offers a much better chance to optimize the resulting multiprocessor platform. Existing and emerging technologies for configurable and extensible processors and the creation of configurable multiprocessor subsystem platforms offer significant capability to design teams to both differentiate and optimize their products.

I Introduction

Designing multiprocessor platforms for embedded systems has never been easy and the challenges are growing with the rise of portable, multifunction devices incorporating subsystems for communications, multimedia, security and entertainment applications. Completely fixed multiprocessor platforms incorporating fixed components are a poor choice to meet the constraints of performance, cost, energy consumption and design time. Configurable technology in the form of application-specific instruction set processors (ASIP) has been well-proven over many years and many designs. ASIPs provide the opportunity to precisely tailor and optimize the platform to the particular product application's performance, size, and power requirements while still providing the time-to-market and programmability advantages that general-purpose processors have compared to custom logic.

Configurable processor technology allows the definition of programmable processing elements precisely tuned to the particular part of the application to which they will be applied. In addition to tuning the processor cache, memory, special functional units and application-oriented instruction extensions, modern configurable processors allow a wide variety of communication schemes to be supported, including point to point, shared memory and bus-based communications. The wide variety of communication schemes enables the next level of configurability, namely system-level configurability. Developing a configurable multiprocessor platform for the application in which number of processors, the inter-processor communication schemes, and the mapping of portions of the application to processors and communication channels extends the benefits of configurability to the system level.

Such platforms themselves may be application-specific subsystems composed into a more complex system-on-chip (SoC) device that delivers the complete product functionality. Programmable application-specific subsystems built from optimized components make the design and implementation of complex SoCs tractable, manageable, and less risky.

In this paper we briefly overview the evolution of processor design, emphasizing the rise of configurable processor technology. We then discuss the move to multiprocessor platforms, and outline some of the key design issues that must be addressed in applying this architecture to embedded systems. We then draw some conclusions.

II Evolution of Processor Design

The microprocessor has been with us for more than 30 years, and has evolved in response to available silicon technology and to electronic system requirements. Now, shifts in basic silicon scaling and embedded systems are forcing a significant shift in the architecture, the use and the design process for microprocessors.

Three fundamental trends in the design of embedded systems are driving a basic redefinition of the microprocessor. First, total system complexity, especially software complexity, is growing dramatically. Second, the cost, power, and size constrains of high-volume systems (e.g. digital televisions, mobile phones, games, digital cameras, printers, etc.) make high-integration silicon implementation not just attractive, but mandatory. Third, Moore's Law transistor scaling is reaching limits in circuit power density. These trends underlie important shifts in electronic design, including

- greater importance in finding and exploiting system concurrency,
- increased use of processors, not only in traditional control and DSP roles, but also in roles previously reserved for hard-wired multimedia, security, and protocol-processing tasks, and
- a basic shift in the architecture of processors, away from complex one-size-fits-all architectures running at high frequency, to much more energy-efficient processors optimized for each system-on-chip application platform.

Historically, microprocessor architectures and application-oriented SOC architectures have followed wholly

1-4244-0630-7/07/$20.00 ©2007 IEEE.

independent paths. Processors were inherently generic while SOC designs become increasingly focused on the specific needs of the target system.

When processors were developed separately from the rest of the system, this generic optimization served system needs adequately. However, the emergence of automatic generation of complete processor hardware and software dramatically improves processor efficiency. And in the era of application-specific chips, creating a new processor for each new chip family no longer carries a cost or risk penalty.

State-of-the-art processor generation offers four dimensions to the configuration of the microprocessor – instruction set, memory system, processor interface, and processor control functions [1][2]. Together, these enable quick generation of an enormous range of optimized processors at various degrees of application specialization. Moreover, these processors incorporate versatile inter-processor communications channels that permit order-of-magnitude improvement in communications bandwidth and energy efficiency. Automatically generated processors change the design of electronic systems in two basic ways:

1. Tuned processors are smaller, faster and more energy efficient than the generic processors that preceded them.
2. Tuned processors also serve as an alternative to hard-wired logic blocks, bringing full programmability from high-level languages to high-performance functions.

These two effects make it both possible and desirable to build application-oriented chips using processors as a basic building block – 5 or 6 processors per chip has become routine, even in consumer systems and chips with almost 200 32-bit processors are used in production network routers. The optimization of processor configuration and inter-processor communication has been a central theme in system-on-chip development. This shift from generic processors, often with hardwired logic accelerators, to configurable processor-based system design appears fundamental to improved design productivity and end-product efficiency.

III. Moving to Multiprocessor Platforms

Many interesting products can be designed incorporating a single processor with perhaps some embedded hardware to accelerate certain functions, but in the embedded area we see many products incorporating multiple processors on chip. From the cell phones of the late 1990's, incorporating a RISC control processor and a DSP for voice encoding/decoding, multiprocessor SoC designs have become more and more frequent [3]. We see desktop and server machines switching from single processors clocked at ever increasing frequencies, to multi-core designs in order to offer increased performance while keeping power dissipation and heat under control. In the deeply embedded product space, splitting an application into multiple asymmetric processors clocked at a lower frequency and using a lower operating voltage saves large amounts of energy in comparison to a single processor with the same computational burden running at a faster rate.

Often, a multiprocessor SoC utilizes 2 to 10 processors, but predictions from many sources anticipate a future where 10s to 100s of processors will be integrated onto a single SoC. Indeed, Cisco has designed an SoC for its CRS-1 router, the silicon packet processor, integrating 192 Xtensa processors onto a single chip. These processors are partitioned to deal in a data-wise parallel manner with multiple packets or packet streams simultaneously.

Single processor designs could often be done with relatively rudimentary system level design tools. However, multiprocessor SoC is beginning to require much more complex Electronic System-Level (ESL) tools and models [4].

Two of the most important decisions in moving to an MPSoC design approach are to decide on the basic concurrency architecture on the multi-processor side, and to decide on the basic description of concurrency on the software side – what is often called the "programming model". Many of the multi-core and multiprocessor architectures being advocated are based on symmetric multiprocessing approaches (SMP) – homogeneous cores sharing a large coherent memory space – and the programming models often advocated for SMP are multithreading ones. While this may be reasonable for large general purpose server machines or the desktop, where the applications are often not known in advance, and there is no specificity possible with the processor cores, it seems unlikely to be the best approach for deeply-embedded data-intensive applications in well-defined domains. This is especially true from a configurable and extensible processor perspective.

For this class of problems, a more likely architectural choice is asymmetric multi-processing (AMP), in which each processor is configured to best match the tasks that are going to be mapped to it, or at least match the domain of processing to which the processor will be dedicated (for example, an audio or video codec processor). Because AMP is still utilizing processors, they can have general additional applications mapped to them in future even without specific instruction extensions. But they can utilize the large gains in performance and energy efficiency possible with configuration if enough is known about the applications in advance.

Configurable multiprocessor platforms supporting a wide variety of communications architectures will allow the decision as to specific communications schemes to be supported and their fundamental characteristics to be deferred until the platform is utilized as a subsystem in a more complex SoC. There is also a case that can be made for fixed multiprocessor platforms composed of application-optimized configured processors and components, but clearly, keeping the maximum flexibility in the platform to support late configuration for a derivative product is best if the design methodology supports it.

To make software application writing, porting, mapping and design space exploration easier, it is highly desirable to find efficient programming abstractions and models so that the concurrency possible in the application, whether multi-threading or pipelined dataflow style or some mix of styles, can be explicitly exposed without tying the code to a very specific choice of numbers and kinds of processors. Some API libraries supporting particular programming models have emerged out of large system parallel programming research and work on embedded applications is yielding new approaches such as TTL [5] and ESPAM [6]. However, there still is a need for some standardization in this area.

IV. Multiprocessor Design Problems to be Solved

There are several key design questions that must be answered in developing a multiprocessor platform for specific next-generation embedded applications and products:

- Extraction of concurrency from the application software or models – decades of research into trying to automate the discovery of concurrency in software, but very little in the way of generally applicable results have emerged. The best that can be done still seems to be to give API libraries to software writers and ask them to expose concurrency or parallelism manually.
- Homogeneity and Heterogeneity – for many applications, heterogeneous AMP is the right approach. But homogeneous SMP is attractive when the applications cannot be predicted in advance. How can we combine both approaches where needed? How do the two domains communicate and synchronize?
- How many processors are enough for an application, and are they best designed or configured as the composition of discrete subsystems or as the melding of them to share resources?
- What kind of on-chip communications architectures are best? Does network-on-chip (NoC) have wide applicability? How do we effectively mix NoC, buses, shared memory and point to point approaches?
- Scalability: It is relatively easy to visualize how to get to 10, 20, and perhaps up to 100 processors in a complex SoC composing many multiprocessor subsystems. How do you get to 500 to 1000 or more? Do we even want to? What are the practical limits?

Although many design methods and some tools are just beginning to emerge that deal with these issues, there are many advances still required and many discoveries yet to be made in research, and applied to practical industrial design.

V. Summary and Conclusions

This brief paper has outlined some of the shifts in processor design approaches and the key requirements of multiprocessor platforms for embedded systems that encourage the use of configurable technology on all levels of the design process. The key design questions for multiprocessor platform design methodology have been summarized.

The evolution of complex next-generation embedded systems will encourage a growing use of multiprocessor platforms in their design, but the desire to optimize the products will demand a greatly increased use of configurable processor technology to build configurable platforms that enable the design of application specific products to meet cost, performance and energy consumption constraints.

References

[1] C. Rowen and S. Leibson, *Engineering the Complex SOC*, Prentice-Hall PTR, 2004.

[2] S. Leibson, *Designing SOCS with Configured Cores: Unleashing the Tensilica Diamond Cores Technology*, Elsevier Morgan Kaufmann, 2006.

[3] A. Jerraya, and W. Wolf (Editors), *Multiprocessor Systems-on-Chip*, Elsevier Morgan Kaufmann, 2005.

[4] G. Martin, "Overview of the MPSoC Design Challenge", *Design Automation Conference 2006*, pp. 274- 279.

[5] P. van der Wolf, E. de Kock, T. Henriksson, W. Kruijtzer, and G. Essink, "Design and programming of embedded multiprocessors: an interface-centric approach", CODES+ISSS 2004, pp. 206-217.

[6] H. Nikolov, T. Stefanov, and E. Deprettere, "Multi-processor System Design with ESPAM", CODES+ISSS 2006, pp. 211-216.