Journal of Novel Applied Sciences
Available online at www.jnasci.org
©2014 JNAS Journal-2014-3-6/572-577
ISSN 2322-5149 ©2014 JNAS

VIRTUAL PROTOTYPE SYSTEMS (APPLICATION OF EMBEDDED SYSTEMS)

Mohammad Reza Pourmir

Computer Engineering Department, Faculty of Engineering, Zabol University, Zabol, Iran

Corresponding author: Mohammad Reza pourmir

ABSTRACT: By today's standards, early microprocessor-based systems were simple, not least because they typically employed only a single processor (possibly with a few co-processors, such as a floating-point co-processor) with a relatively simple instruction set running at low clock frequency. This processor communicated with a small number of comparatively simple memory and peripheral devices by means of a single 8-bit or 16-bit data bus with a simple read/write and signaling protocol. Those days have long gone. There is currently a tremendous growth in the development of systems that involve tens or hundreds of complex processors and hardware accelerators in closely coupled or networked topologies. In addition to tiered memory structures and multi-layer bus structures, these super systems - which may be executing hundreds of millions to tens of billions of instructions per second - feature extremely complex software components, and this software content is currently increasing almost exponentially. Aggressive competition makes today's electronics markets extremely sensitive to time-to-market pressures. This is especially true in consumer markets such as cell phones, where the opportunity for a new product to make an impact can sometimes be as little as two to four months. However, a recent report showed that more than 50 percent of embedded system developments run late, while 20 percent either fail to meet their requirements specifications or are cancelled in their entirety. The problem is that, in conventional system development environments, hardware design precedes software development. This sequential process simply cannot support the development of today's super systems. This article first introduces examples of super systems and outlines the problems presented by increasing system size and complexity. The concept of architecture-driven design based on the use of virtual system prototypes (VSPs) is then discussed as a potential solution. Finally, a productivity, development time, and risk comparison is made between the back-end engineering resource loading associated with the conventional environment and the front-end loading resulting from the architecture-driven, VSP-based methodology.

Keywords: hardware/software co-design, rapid system prototyping,Design-space exploration, mobile terminal, SDL, SoC, VSP.

INTRODUCTION

In some respects, the term "super system" may be misleading, because it may cause some readers to imagine a physically large implementation. Actually, a supersystem is often realized on a single system-on-chip (SoC) device.

For example, a modern cell phone may contain an SoC comprising several general-purpose central processing units (CPUs), and one or two digital signal processing (DSP) units, controlling 40 or more peripheral devices providing control functions, multimedia functions, 2D and 3D graphics functions, crypto functions, camera interfaces, and a variety of other interfaces such as WiFi and USB.

The DSPs with associated accelerator devices provide a variety of base band processing, filtering, modulation, and decoding functions. Having multiple cores allows a broader range of processing traffic to be handled in real-time, which is a critical requirement for many of today's applications.

Moving away from the handheld portion of the wireless network, the base stations controlling wireless communications systems are themselves a hierarchy of closely-coupled multi-processor systems. For example, a typical base station capable of executing billions of instructions per second can comprise five to 20 major subsystems and more than 100 individual processors.

In addition to multiprocessor implementations, today's super systems employ tiered memory structures. Some of the memory elements will be tightly coupled to individual processing engines by means of dedicated busses, other memory subsystems may be local to a cluster of processing engines, and yet other memory units may be shared between multiple groups of processing engines. Each of these memory subsystems may have different speed requirements, different bus widths, and use different clock domains.

In today's supersystems, different processing engines can have separate buses for control, instructions, and data, and each of these complex buses can feature a wide variety of structures and protocols. In addition to the general-purpose processor buses, there may be a variety of dedicated peripheral buses, tightly-coupled memory buses, external memory buses, and shared memory buses.

Many of these buses will feature pipelined structures with multiple transaction requests and responses scheduled in the pipeline. The bus system also may employ sophisticated cross-bar switches that can attempt multiple read and write operations simultaneously.

Even the average modern car contains from 20 to 80 processors performing a huge range of tasks and executing several hundred million to several billion instructions per second (Figure 1).

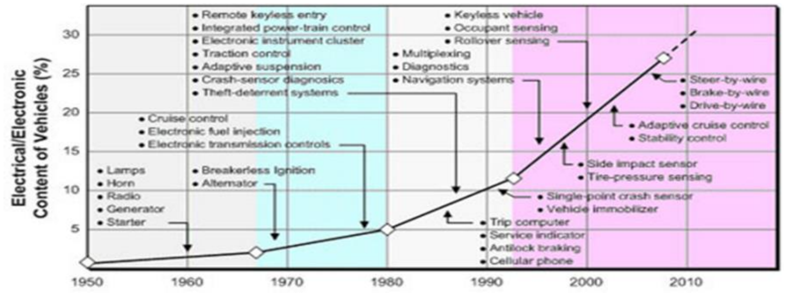


Figure 1. the electronics content of automobiles is growing at an ever-increasing rate[2]

The Virtual System Prototype Revolution

A VaST virtual system prototype represents a revolutionary change for target software development — it removes the need to develop code on host workstations and test by downloading to target single- board computers for execution and debugging. Software is developed on a virtual prototype of the target system rather than on a host system. Code is then executed and debugged as it is written — not months later when the silicon is available The VaST virtual system prototype simulates fast enough that it is no longer necessary to use either host-based or hardware-based (ICE, emulation or prototype) development approaches. Since the entire simulation is cycle-accurate, it can be used to develop both software with strict real-time requirements and software that interacts intimately with the hardware. A VaST virtual prototype is a PC-based surrogate for the actual chip or system into which the software will eventually be integrated.

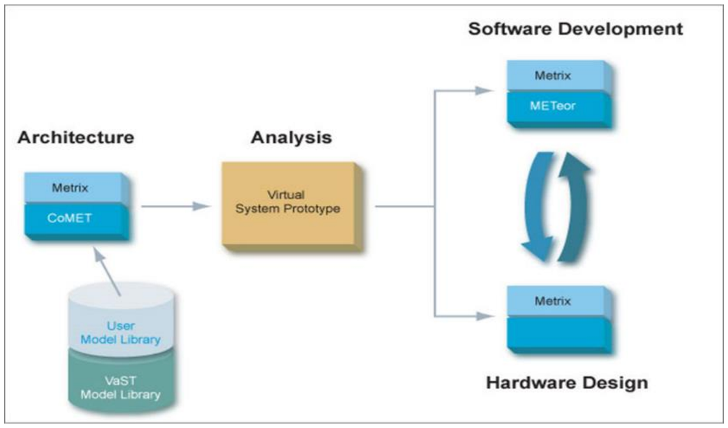
Building and using VaST virtual system prototypes does not require major changes to a software development methodology. Because a VaST virtual system prototype executes the identical binary code used on the real hardware the choices of compiler and source-code control software are not restricted. Even real-time operating systems available only in binary form will run correctly. Software tools are the standard cross-development C/C++ compilers together with debuggers suited to the target processor(s).

VaST's visualization tool, Metrix, provides visibility into the internals of the virtual system prototype, making debugging much easier. Observability, coupled with the ability to walk through and traverse the hardware/software interface, using break-pointing and single-stepping, provides a potent verification and debugging environment. Models can contain extensive error checking, simulation time can be frozen, even in a multi-processor system, since all processors will stop when one hits a breakpoint or is single-stepped. These unique advantages of the virtual system prototype approach translate directly into higher productivity and higher quality.

VaST's virtual processor modeling technology supports the development of an operating system (OS) and its device drivers within a target system, as well as porting existing OSes to target systems. The observability that VaST's technology brings to the hardware/software design process means that OS development and OS porting cycles are significantly shortened. Booting a real-time OS takes seconds to execute on a VaST high-performance virtual processor model as opposed to hours on an instruction set simulator (ISS).

Accelerates Product Development

VaST provides tools and models for embedded system design. Users create a software simulation-based model of a system on a chip (SoC) or a portion of a system. This virtual system prototype (VSP) is initially used for quantitative architectural exploration and optimization. When finalized the virtual system prototype becomes the golden reference model that enables concurre nt development of hardware and software.



Customers use the VaST CoMET and METeor tools — together with virtual processor, bus and peripheral models — to develop virtual system prototypes (VSPs) as early as the architecture phase of the embedded systems design. The high performance, timing accuracy, observability and controllability of virtual system prototypes make them indispensable for embedded systems design

Superior Systems Engineering Tools for Embedded Systems Design

VaST provides manufacturers of consumer electronics such as cameras and printers with the systems engineering tools required to get to market sooner with higher quality products that cost less to build. Using VaST's tools and virtual system prototyping technologies, consumer electronic designers rapidly explore multiple design alternatives without having to build costly and time-consuming physical prototypes.

Many areas of consumer electronics, such as digital cameras, MP4 players, printers and digital radios, contain increasingly large amounts of embedded software interacting with complex hardware. Much of this software operates in an environment of inflexible real-time constraints. All consumer electronics products are developed in an environment of intense cost sensitivity where design challenges cannot afford to be solved by over-engineering the SoC.

Optimal Embedded Systems Design Increases Profit Margins

Using VaST embedded systems design tools and models to create an optimal architecture for a SoC can make a big difference to the profit margin for the chip and the system built around it. For example, a cache memory that is unnecessarily large results in wasted silicon area and thus increased cost, while a cache memory that is too small results in lower performance than optimum for the application. VaST's virtual system prototypes enable architectural exploration that produces quantitative results on aspects such as power consumption, performance and cost.

In most consumer products, aspects of the system performance are visible to the end purchaser as capabilities of the product, such as the rate at which a digital camera can take photos or the ability of an MP4 player to play music continuously under all circumstances. The VaST suite of systems engineering tools and models allows developers to evaluate architectures and then run the hardware and software together to ensure that the product will have the right performance and features to meet market requirements.

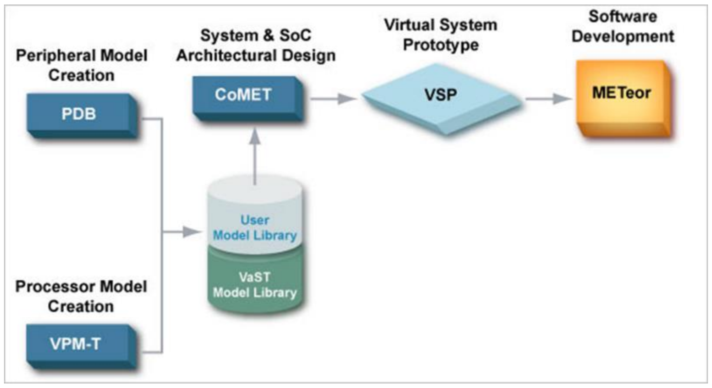
A growing number of consumer products are battery powered. The power measurement capabilities of VaST's virtual system prototype solutions means that estimating the actual power usage for features such as the number of photographs per recharge can be minimized before the real chip is available.

Using VaST's virtual system prototype, embedded software development can begin even before the detailed hardware design has been finalized. Porting of operating systems and communication stacks, together with the development of both driver and application code, can be overlapped with the hardware design and development. This concurrent design and development means that product cycle times and time to market for new and iterated products are substantially reduced.

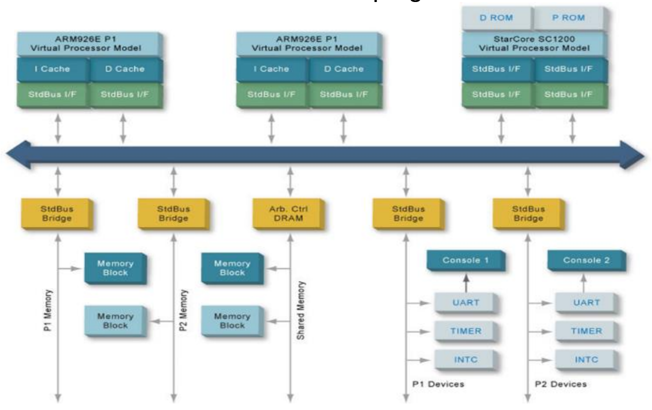
CoMET® System Engineering Environment

CoMET® is a system engineering tool that enables the creation of a software simulation-based virtual system prototype (VSP) of a system-on-a-chip. Initially used for architectural optimization and analysis the VSP goes on to become the environment for the >concurrent design of hardware and software. With CoMET you can design, simulate, analyze, and optimize complex embedded systems and quantitatively evaluate performance while running real software applications.

METeor® Software Development Environment



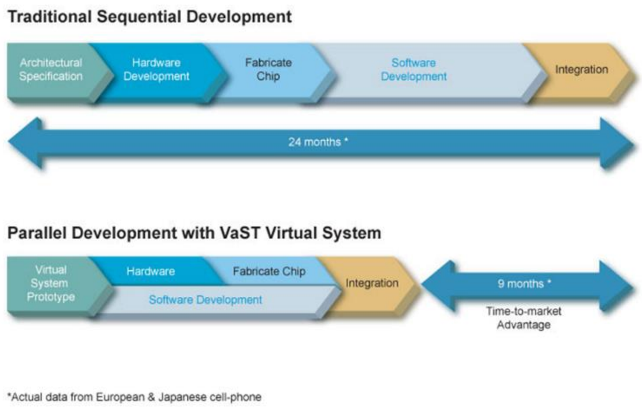
METeor® is an interactive, real-time software development environment for embedded systems and system-on-chip (SoC). With METeor you can use the virtual system prototype (VSP) created with CoMET to develop, simulate, debug and optimize embedded software many months before the hardware becomes available. Metrix™ offers system architects, hardware designers and software developers insight into the behavior and performance of the hardware and software components of virtual system prototypes. The Metrix environment streams, post-processes, and presents selected data from a VSP simulation, facilitating rapid debugging through the visualization of dynamic functionality. Peripheral Device Builder (PDB) enables the rapid creation of VaST models of peripheral devices such as interrupt controllers, DMA controllers, memory controllers, timers, and many others. PDB automates much of the model coding process and provides a common code base helping to ensure code consistency and quality.



Virtual Processor Models

VaST virtual processor models (VPMs) are both fast and accurate, running at speeds of up to 200MIPS on off-the-shelf PCs. A VaST VPM is configurable and provides an unprecedented ability to observe the behavior of executing software and selected processor internal registers. VaST has a broad selection of popular processors from leading IP vendors. VaST provides system engineering tools that enable customers to customize VPMs, as well as turnkey VPM development services.

Virtual Processor Models in the VaST library



Architecture-driven, VSP-based design

In order to satisfy the requirements for architecting and implementing today's supersystems, the development environment must address several major issues:

- Since for supersystems, software is on the critical development path, it is no longer viable to wait for hardware to become available before commencing software development.
- Both software and hardware are undergoing phenomenal increases in generational complexity, but software complexity is increasing at a higher rate.
- Natural language specification documents comprise thousands of pages which take months to write, and this time adds to the design period. Specification documents are inadequate for expressing and communicating the requirements unambiguously for complex supersystem developments with short design cycles.

One solution is to use a virtual system prototype (VSP), which is a functionally-accurate and timing-accurate software model of the entire system. Unlike instruction set simulator (ISS)-based prototype solutions — which typically achieve sub-MIPS of performance — today's best VSP prototypes are based on simulation engines that offer both high performance and timing accuracy.

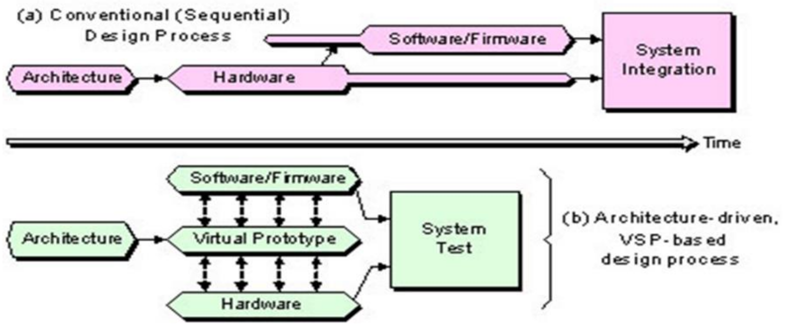
For example, a single processor VSP-based simulation can achieve anywhere between 50 to 200 MIPS, while multi-processor systems with tiered memory structures and multi-level buses can be simulated at 10 to 100 MIPS per processor, depending on the configuration.

These levels of accuracy and performance mean that VSP-based environments can support the concept of architecture-driven design. First, VSPs by nature support the rapid design of experimental systems and then facilitate architectural exploration and evaluation by supporting rapid iteration of the hardware and software that define the system. Accurate measurements of systems that model real-world behavior under real-world data processing and software workloads allow system architects to make accurate decisions as well as hardware/software tradeoffs early in the design process.

Furthermore, the ability of the VSP to run real software workloads yields invaluable information in guiding the system architects to an optimal architecture. This information includes:

- Bus contention and bandwidth utilization
- Processor capabilities and utilization
- Working set match/mismatch to cache size and mapping policies
- Latency in system response to internal and external events
- Correlation of external, hardware, and software conditions and events
- Transaction and algorithm throughput

Once an optimal architecture has been determined, the VSP model assumes the mantle of the executable specification. This means that the hardware design teams can use the VSP as a "golden reference model" against which they can verify the functionality of the hardware portions of the design



VaST Virtual System Prototypes Cut Development Cycles

Wireless handset development is a system-level problem, entailing development of sophisticated hardware and a significant and growing body of software to perform the many functions demanded by the market. With high volumes and short product lifecycles, bringing a product to market early is the key to profitability.

The growing complexity of embedded designs requires new design methods and automation. A complex multimedia handset typically consists of a digital signal processing subsystem, a subsystem with a general-purpose microprocessor and often a third processor to handle multimedia functions. These subsystems are connected by complex multi-level on-chip buses to which are connected many peripheral models such as timers, the radio interface and a subscriber identity module (SIM) card interface.

VaST's electronic system-level (ESL) design tools, models and virtual prototyping technology enable the design and development of these increasingly complex wireless devices faster andwith increased confidence in their functionality and performance.

CONCLUSION

In this paper we propose a hardware/software co-design for embedded systems. Within this Software components are abstractly modeled using SDL, while hardware components are emulated in software using the concept of virtual prototyping. This approach allows for fast and early investigations of several design options for both hardware and software due to the low effort and high speed associated with such modeling techniques. We customize the processor and memory subsystem of the platform by rapidly obtaining a suitable design configuration which meets the required timing constraints and provides 80 % packet processing speedup compared to other unoptimized implementations. In addition, the achieved design parameters provide a balanced power/area consumption trade. For further study, different Hw/Sw partitioning configurations can be applied and evaluated in a similar way. As a future work, we will adapt the software part to make efficient utilization of multi-core architectures and make further investigations about the performance gain, power consumption and scalability of such a multi-core based communication system.

REFERENCES

http://www.eetimes.com/
http://www.vastsystems.com
http://www.wikipedia.org//
http://www.vastsystems.com//
Embedded computer systems: architectures, modeling, and simulation: 5th... By Timo D. Hämäläinen
Technology for the United States Navy and Marine Corps, 2000-2035: becoming... By National Research Council (U.S.). Naval Studies Board
Virtual prototyping: virtual environments and the product design process By J. Rix, Stefan Haas, José Teixeira.