IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1 Issue 5, July 2014.
www.ijiset.com
ISSN 2348 – 7968

An Approach for Early Software Development to Reduce Time to Market in today's Mobile Market

Sammidi Mounika[1] and Renuka B S[2]

[1]Electronics and Communications Department, Sri Jayachamarajendra college of Engineering, Mysore, Karnataka 570006, India

[2]Electronics and Communications Department, Sri Jayachamarajendra college of Engineering, Mysore, Karnataka 570006, India

Abstract

In today's mobile market, the faster time to market is a significant challenge. In mobile devices where new software and new hardware are being created, software developers must often wait for the hardware design to be finalized before they can begin detailed coding. Software developers must also wait for devices such as integrated circuits and printed circuit boards to be manufactured to test their code in a realistic environment. These Serialized hardware and software developments often fail to meet product development schedules. Virtual prototyping provides a platform for pre-software development and verification and enables concurrent software and hardware development.

Keywords: *Time-To-Market Serialized Hardware and Software Development, Virtual Prototyping, Concurrent Hardware and Software Development.*

1. Introduction

The design of mobile phones is becoming increasingly complex to support additional features. There is a significant increase in the number of cores and the processing power. The embedded processor cores feature in more percentage of today's System on Chip designs. The number of cores in a mobile device is increasing rapidly and with it, total device complexity. Accordingly, writing software for these complex multi core systems became a great challenge than building hardware. The amount of interaction between the cores is increasing as well, meaning that there are many more opportunities for problems to exist in temporal interactions, resource conflux and performance issues. The software development costs will also continue to rise due to the increased system complexity. As a result, the software development and verification became an essential part of the product development life cycle [1].

In order to stop software being on the critical path of product development it is necessary to find some way of enabling software development to be conducted in parallel with the hardware development. What is needed is the virtual prototype of the hardware of a mobile phone that is available very early in the development program on which the software can be verified [2].

2. Traditional Methods

Without a virtual prototype, the software development team has few options available to them for integrating and debugging code. The options available to the software development team are

2.1 Wait for the hardware development to be completed

This has the advantage that it is a true representation of the product being bit and detailed timing accurate. But it is far too late in the product development cycle and delays product release. It also means that any compatibility problems found must be worked around in the software.

2.2 Use a previous generation product

A previous generation product can be used sometimes if it is similar enough. This can be expensive and still relies on the hardware availability for new capabilities.

2.3 Create an early prototype

In some cases emulation, rapid prototyping using Field Programmable Gate Arrays can make the hardware available beforehand. They are also bit true and cycle accurate. But this is still after hardware design has been completed and can be expensive.

Virtual prototypes are one of the tools that software engineers are increasingly turning to for testing and debugging the software and in some cases virtual prototypes are the only solutions that can provide an

ISSN 2348 – 7968

answer. Even the real hardware cannot address some of them [3].

3. Virtual Prototypes

A virtual prototype is a software-simulation-based, architectural-level model of the embedded system. Since this system model has the same capabilities as the hardware prototype, real-world effects can be modeled. That ensures the hardware prototypes will work when built and minimizes the hardware-software integration effort in the late stages of the project. The system model can include processors, buses, hardware peripheral components and even models of mechanical subsystems that are part of the overall system. Virtual prototype becomes the golden reference for the hardware and software development effort. The processor models in the virtual prototype are closely linked to code development tools. When software updates are compiled, the same binaries that will run in the final system can be executed within the virtual prototype. That allows system architects to evaluate candidate architectures by running realistic software loads. It also lets software engineers debug their code using architectural models, long before the detailed Register Transfer Level (RTL) hardware design is complete, thus enabling true parallel development of hardware and software [4].

The models can be written in any language. But most of the time they will be in C, C++ or a language called systemC. SystemC is a C++ class library that adds capabilities such as concurrency and some notions of time into a model. This makes it possible to consistently and accurately model the way in which various pieces of the system interact with each other. SystemC is the language used to assemble various models and to perform necessary coordination [5].

3.1 Characteristics of Virtual Prototypes

Early availability: Virtual prototypes are available before the physical system (RTL, board, test bench, etc.).

Software left-shift: Virtual prototyping enables software to be developed before the hardware is available. Early availability enables concurrent hardware, software and system development.

Easier accessibility: Virtual prototypes are software packages, and as such they can be duplicated and distributed worldwide to a large number of users in a matter of minutes rather than weeks. They provide a simplified development environment executable on every developer's desktop.

Increased productivity: Virtual prototyping results in faster edit-compile-debug cycle productivity. In addition, the system execution is deterministic and debugging is non-intrusive.

Reduced dependency: Software has dependency only on virtual prototype models for its development and hence inter-team coordination is smooth and is less susceptible to environment and set up issues.

Multi-core/multi-processor support: Debugging the software running on one processor can be tough enough and when the problem occurs on multiple cores, it is still tougher to debug. What is required is not just to look at each core but need to look at the status of entire system in an abstract manner.

4. Concurrent Hardware and Software Development

The incorporation of a virtual prototype into the development flow can have a significant number of benefits. The ability to start software development early in the project, before the detailed hardware design has started, greatly accelerates the overall development schedule. The software team no longer has to wait until hardware prototypes are available to run and debug their code. Running the code against the system-level models of the hardware, and swapping in RTL when available, greatly increases the chances that there will be no problems found during final integration with the actual hardware [3].

5. Conclusions

Virtual prototyping is the best way to face challenges such as faster time to market. It enables concurrent hardware and software development which helps to meet aggressive product development schedules. The software developers no longer need to wait for the hardware availability to test their code. The proposed approach enables to simulate complex virtual platforms in a faster and more effective way and hence the software/firmware development will be faster. This methodology enables leading semiconductor and electronics companies to deliver more competitive and higher quality products up to 9 months faster than waiting for production of hardware.

ISSN 2348 – 7968

References

[1] Bryan Schauer,"Multicore Processors – A Necessity", released september 2008, [online]. Available: http://www.csa.com/discoveryguides/multicore/review.pdf

[2] Kalypso White Paper by Bill Poston & Joe Dury, "Semiconductor Product Lifecycle Management- Industry Adoption, Benefits and The Road Ahead".

[3] Arjen Damstra, "Virtual prototyping through co-simulation in hardware/software and mechatronics co-design", April 2008.

[4] S.H. Choi and H.H. Cheung, "Virtual Prototyping for RapidProduct Development" Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong SAR, China [online].
Available: http://cdn.intechopen.com/pdfs-wm/28871.pdf

[5] IEEE Standard SystemC® Language Reference Manual, IEEE Computer Society Sponsored by the Design Automation Standards Committee

Sammidi Mounika Master of Technology 2012-2014, Bachelor of Technology 2008-2012.

Renuka B S Master of Technology, Bachelor of Technology, Associate Professor, Department of Electronics and Communications, Sri Jayachamarajendra College of Engineering.