Incorporating Virtual Prototype in Product Development to Reduce Time-To-Market of a Mobile Device Product

Nirjith Prakash¹ Dr. M.N. Shanmukha Swamy²
¹²Department of Electronics and Communications Engineering
Sri Jayachamarajendra College of Engineering, Mysore, Karnataka 570006, India

Abstract—The Time-to-market constraint is a significant challenge to any embedded systems product, especially a mobile device because of its highly-competitive market. Traditional methods of serialized hardware and software development often fail to meet the Time-to-market constraint. This demands concurrent hardware and software development. Virtual prototyping provides a solution to these new development challenges.

Key words: Virtual Prototype, Time-To-Market

I. INTRODUCTION
The design complexity of mobile phones has increased drastically in the last few years, and keeps increasing to support additional features[1]. This impacts the complexity of the software written to these systems, also to increase. The integration of increasingly complex hardware and software is a significant challenge for developing next-generation mobile devices.

The increased complexity and intense time to market (TTM) has rendered traditional methods of serialized hardware and software development—where the vast majority of software is developed and verified after the silicon design, as inefficient and impractical. These factors demand concurrent hardware and software development [4][3]. Virtual prototyping provides a solution to these new development challenges and requirements.

II. TRADITIONAL DEVELOPMENT APPROACH

Traditional development approach views hardware and software development process as two serialized processes which happens one after another as can be seen in the figure1.

Fig. 1: shows how virtual prototypes accelerate time-to-market with higher quality and fewer resources.

III. VIRTUAL PROTOTYPE APPROACH

A virtual prototype is a software-simulation-based, architectural-level model of an embedded system. Since a virtual prototype has same capabilities as the hardware prototype, real-world effects and environment can be modeled into the VP. This lessens the hardware-software integration effort in the later stages of the project and ensures the hardware prototypes will work when built. The virtual prototype model includes single or multiple processors, hardware peripheral components, multiple buses, and can even have models of mechanical subsystems which are part of the overall system.

These models can be written in any language. But for more accurate representation of the hardware, it is most likely to be written in SystemC, as it is a system design language which lets the engineers to design both the hardware and software components together as these components would exist on the final system, but at a high level of abstraction. SystemC is a single, unified design and verification language that expresses architectural and...
other system-level attributes in the form of open-source C++ classes[5]. With SystemC, designers can apply object-oriented capabilities to hardware design.

With Virtual Prototype development approach, the whole hardware system, the compiler toolchain, the software and the debugging tools are now located inside the developer's desktop PC. The developer can analyze the performance of the system, study the throughput, find bottlenecks or have a look at the task scheduling and therefore has the possibility to optimize, for instance the bus and memory architecture. Virtual Prototype enables development that is not possible, or is very difficult, with the physical or actual product[6]. It is even possible to model hardware control elements like buttons or a touchscreen as shown in Figure 2.

![Fig. 2.1: Traditional software development and debugging](image1)

Fig. 2.1: Traditional software development and debugging

![Fig. 2.2: Virtual Platform software and hardware development and debugging](image2)

Fig. 2.2: Virtual Platform software and hardware development and debugging

The integration of hardware, software architecture into a single development environment has enormous advantages, and positive effects on every aspect of the project. A single IDE also prevents the traditional problem of changes rippling back up to the architecture, based on issues discovered only during detailed hardware and software implementation, which occurs at end stages of product development.

Software development being pushed up in the project timeline due to availability of virtual prototype means the overall development schedule is accelerated and the time-to-market of the product is reduced. The availability of virtual prototype prompts software team not to wait for the hardware prototypes to be available to run and debug their code.

Running the software code on the virtual prototype models of the hardware, and swapping in RTL when available, dramatically reduces the risk of finding problems during final integration with the actual hardware.

IV. CONCLUSION

Virtual prototyping is the best solution to the new development challenges such as increasing hardware complexity, faster time to market. Software developers’ reliability on Virtual prototypes for testing and debugging the software is increasing day-by-day, and they no longer need to wait for the hardware availability. Concurrent hardware - software development is a boon resulting from incorporating virtual prototype, which helps to meet aggressive product development schedules.

A high-performance, cycle-accurate virtual prototype, together with tools that enable Concurrent hardware-software development for embedded-processor-based electronic systems has been shown to:

- Reduce product development risk by about 67%.
- Reduce development bench cost by about 50%.
- Reduce engineering development time by about 25%.
- Reduce engineering resources by about 25%.
- Improve product quality by 50%.

Incorporating Virtual prototype in product development enables leading semiconductor and electronics companies to deliver more competitive and higher quality products up to 6-to- 9 months faster than waiting for production of hardware.

REFERENCES