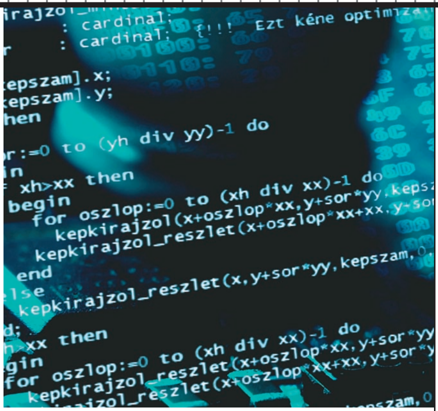
A Survival Kit: Adaptive Hardware/Software Codesign Life-Cycle Model



Dong-hyun Lee and Hoh Peter In, *Korea University*
Keun Lee, *Samsung Electronics*
Sooyong Park, *Sogang University*
Mike Hinchey, *Lero—The Irish Software Engineering Research Centre*

An adaptive hardware/software codesign model enables consumer electronics companies to reduce production cost and maximize profits by progressively modifying codesign options over the life cycle.

The cost of a digital television such as an LG 42-inch LCD—about $1,800 in June 2007—had plummeted to about $1,000 by December 2007. Within a half year of the initial marketing, 44.4 percent was wiped from the price tag. How can a consumer electronics company survive this cutthroat market competition?

Many CE makers that produce not only digital TVs but also mobile phones, digital cameras, digital video recorders, MP3 players, or other CE embedded systems are struggling to resolve this problem. They reduce expenses by lowering manufacturing costs, improving productivity, and using other management methods.

However, few CE producers have considered using hardware/software codesign technology to lower production costs. CE embedded system makers can use hardware/software codesign technology to replace expensive hardware components with inexpensive software components. Manufacturers can dramatically reduce cell phone handset production costs, for example, by replacing radio frequency hardware chips with software programs such as software-defined radio.

Few studies have examined the cost impact of a codesign decision. Existing hardware/software codesign research focuses only on optimizing the hardware and software combination in the design phase, but the optimized combination remains the same thereafter, even though market conditions may change.

We introduce an adaptive life-cycle model of hardware/software codesign to optimize the production cost of CE embedded systems by modifying various combinations of hardware and software components over the life cycle, according to market conditions such as cost, price, revenue, and time-to-market requirements.

TYPICAL HARDWARE/SOFTWARE CODESIGN DEVELOPMENT PROCESS

The embedded system development process is unique, since two different processes—hardware development and software development—are considered in combination. Arnold S. Berger (*Embedded System Design*, CMP Books, 2002) presented a typical embedded system development process, which consists of seven phases, as Figure 1a shows. Hardware/software partitioning (phase 2) and integration (phase 5) are distinctive features that distinguish this process from general software development processes.

Overall, the industry employs a development process that is similar to a typical embedded system development process, especially in terms of the phased structure. Phases 3 and 4, which involve design and implementation, are slightly different, as Figure 1b depicts. Most vendors develop their

Published by the IEEE Computer Society

own software components but rely on hardware suppliers for hardware designs. In other words, hardware component design is based on the production plan, while the software design is based on the development plan. Moreover, hardware implementation (with buyout options) is more rapid than in-house software implementation. Consequently, hardware and software development lead times and costs differ. A strategy is necessary for adaptive optimization of hardware/software partitioning options over the life cycle.

ADAPTIVE LIFE-CYCLE PROCESS

The adaptive life-cycle process of hardware/software codesign involves reducing the production cost and determining an optimized transition sequence via the codesign life-cycle transition sequence. This method is highly applicable to the embedded system field, since it changes hardware/software codesign options for a product to reduce production costs.

Figure 2 is a conceptual diagram of the adaptive hardware/software codesign life-cycle sequence of a CE product. There are three types of components in the product: software-only, hardware-only, and components that can be implemented as both hardware and software.

Replacing hardware components with software components reduces production costs. The hardware cost is constant in the production stage, after release; the software cost is almost zero during the same process. For this reason, we assume that a transition can occur in only one direction, from hardware to software. Figure 2 shows the transition from hardware-intensive Design A to software-intensive Design D over the life cycle. For simplicity, we also assume that there are no other parameters affecting a component transition from hardware to software.

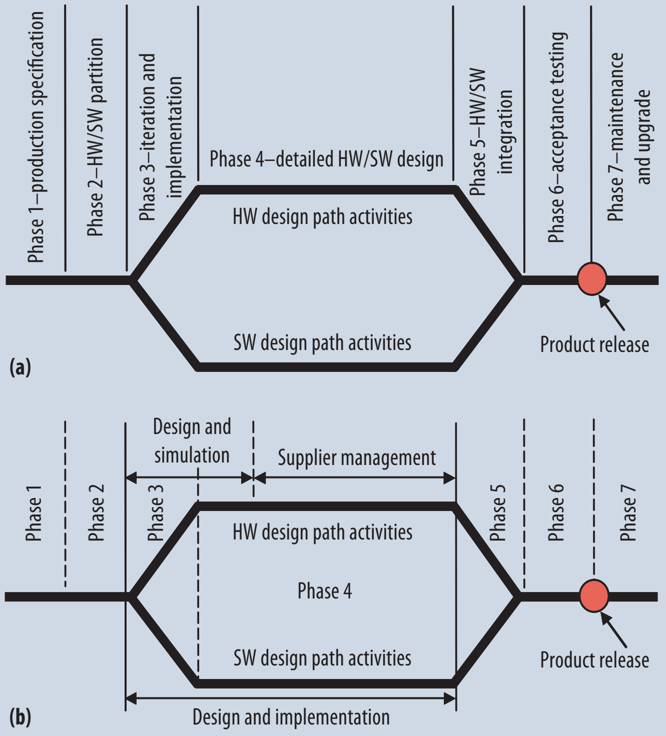


Figure 1. Differences in hardware and software development lead times in the embedded system industry: (a) typical embedded system development process and (b) real-world development process in the embedded system industry.

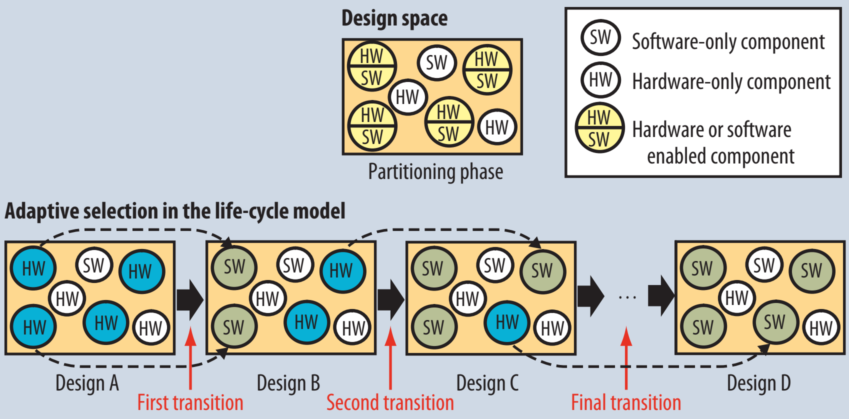


Figure 2. Conceptual diagram of the adaptive hardware/software codesign life-cycle transition sequence.

After release, the optimized hardware/software codesign life-cycle transition sequence determines the design transition sequence in a manner that maximizes revenue from a product in certain production environments—for example, market conditions and the product development schedule. It consists of two steps: *partitioning* and *transition*.

The hardware/software partitioning step generates design options for a product and determines the initial version of the product with certain input parameters—for example, hardware/software components, time to market, and hardware/software development lead time—

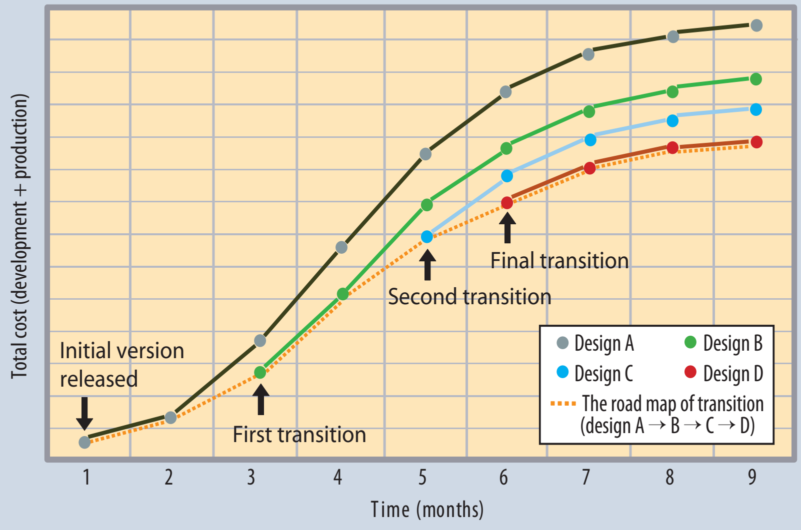


Figure 3. Road map of the transition sequence in the life cycle. The life-cycle transition plan is to adopt design A for months 1-3, design B for months 3-5, design C for months 5-6, and design D for months 6-9.

as well as the costs of developing hardware/software, based on hardware/software cost models. Among these parameters, time to market is the main market constraint on the hardware/software partitioning process, while developing and producing hardware and software are the main cost factors. We have adopted Cocomo II (B. Boehm, *Software Cost Estimation with Cocomo II*, Prentice Hall, 2000) as the model for estimating the software costs, based on the assumption that the hardware costs represent the aggregate price of the hardware components.

The hardware/software transition step determines a design transition sequence with the options generated in the hardware/software partitioning step. This step provides a road map for the optimization of the design transition sequence for maximization of revenues. Figure 3 shows the total cost of design versions over the life cycle, as proposed in Figure 2. Estimating the total cost of each version requires information about market conditions such as the time to market and the estimated number of product sales.

The first transition occurs when the development of design B is complete. The development lead time of software is generally longer than that of hardware, hence it takes time to replace hardware components with software components. The second and final transitions occur when designs C and D are complete. The yellow dotted line in Figure 3 indicates the transition sequence over the life cycle for cost minimization. Thus, the life-cycle transition plan is to adopt design A for months 1-3, design B for months 3-5, design C for months 5-6, and design D for months 6-9. With the road map of the design transition sequence, embedded system vendors can manage the development and production plan over the life cycle with respect to price cuts.

Previous studies on hardware/software codesign mainly focused on development independent of the market requirements, concentrating on various aspects of resource utilization such as the processor, memory, size, power consumption, and timing. The adaptive hardware/software codesign life-cycle transition model considers not only the development cost and lead time but also the production cost and product life cycle. It can maximize profits for an embedded system product, even in cutthroat, red-ocean market conditions.

Currently, we are investigating several industry projects funded by the Information Technology Research Center research program of the Institute of Information Technology Advancement to estimate the return on investment of the proposed model. Contact Hoh P. In (hoh_in@korea.ac.kr) for further information on research results.

Dong-hyun Lee *is a PhD candidate at Korea University. Contact him at tellmehenry@korea.ac.kr.*

Hoh Peter In *is an associate professor at Korea University. Contact him at hoh_in@korea.ac.kr.*

Keun Lee *is a principal engineer at Samsung Electronics. Contact him at gskeun@gmail.com.*

Sooyong Park *is a professor at Sogang University. Contact him at sypark@sogang.ac.kr.*

Mike Hinchey*, Software Technologies column editor, is the codirector of Lero—The Irish Software Engineering Research Centre. Contact him at mike.hinchey@lero.ie.*