Journal of Digital Contents Society Vol. 5. No. 1. March. 2004(pp. 61-65)

하드웨어/소프트웨어 통합설계에 기초한 임베디드시스템 구성

박춘명*

요 약

본 논문에서는 현재의 IT 분야의 핵심 기술 중에 하나인 임베디드시스템을 하드웨어/소프트웨어 통합설계 방법론에 기초를 두고 구성하는 방법의 한 가지를 제안하였다.

제안한 방법에 기초하여 각 종 진보된 멀티미디어시스템을 구현할 수 있으리라 사료되며, 또한 이를 기반으로 좀 더 진보된 각종 디지털컨텐츠를 구현할 수 있으리라 전망되며, 특히 최근 대두되고 있는 모바일 컨텐츠 구현에 적용이 기대된다.

A Construction of Embedded Systems based on Hardware/Software Co-Design

Chun-Myoung Park*

ABSTRACT

This paper presents a method of constructing the embedded systems based on hardware/software co-design approach that was key methodology.

The proposed method was important technology enable to implement advanced multimedia systems and digital contents creating that are rapidly growing of the new information technology.

Key words : Embedded System, Hardware/Software Co-Design, Advanced Multimedia System, Mobile Digital Contents

1. Introduction

In recently, the IT(Information Technology) fields are bring up the most important fields in 21th century.[1-5] As we know well, key technologies of the 21th IT fields are embedded systems, multimedia systems and digital contents etc. These technologies are fused each other, then emerging the new life paradigm for our life style.[6-10]

In this paper, we propose a method of constructing the embedded systems that is the important technology be able to implement advanced multimedia systems and digital contents creating.

During the last years, embedded system engineers have come across significant changes in system due to increasing complexity of the systems[1,6,11-12].

To fulfill the aforementioned requirements, system engineering calls for methodologies and tools that will enable the production of highly complex embedded systems, with shorter development times and reduced cost. Significant research activity is currently taking places in the area of hardware/software co-design, in order to cope with the design problems of modern embedded systems[1,10-11].

2. Key in Embedded System Design

During the last years there has been a lot of activity around embedded system design.This has been caused mainly for two reasons. First, the

※ 제일저자(Frist Auther) : 박춘명 , 주소 : 충북 충주시 이류면 검단리123 충주대학교 전기·전자 및 정보공학부 컴퓨터공학전공,
전화 : 043) 841-5346, Fax : 043) 841-5340,
E-mail : cmpark@chungju.ac.kr
접수일 : 2004년 2월 15일, 완료일 : 2004년 5월 28일

* 정회원, 충주대학교 전기·전자 및 정보공학부 컴퓨터공학전공/글로벌미디어연구소(GMRI)

development of electronic technology allows the integration of increasingly complex systems on a single chip. This has risen the need for new techniques and tools to cope with the complexity of modern designs. Second, the market pressure demands for shorter development times and cost effective designs. The net result is significant research activity in the area of embedded systems, and especially in field of hardware/software co-design. The design of systems that have been developed by the different hardware-software co-design approaches have revealed innovations in several design aspects : specification, estimation, partitioning, prototyping, co-simulation, etc. Nevertheless, most approaches support specification languages that are either fixed or not widely used or in adequate for system level specification. The methodological support is limited to the description of the design phases and the tools that should be used in each phase, while continuous subsystems are not support by most methodologies. Telecommunication and multimedia computing are among the fastest growing segments of the micro-electronics market today.

At the same time, programmability is becoming increasingly important for facilitating flexible designs that can be customized with differentiating features for use in multiple products.

To facilitate flexible low-cost designs in short design time, emerging designs are based on heterogeneous embedded system architectures.

For example, the following Fig.1 describes embedded systems for mobile phone and Fig.2 shows embedded systems for home networking(or home automation).

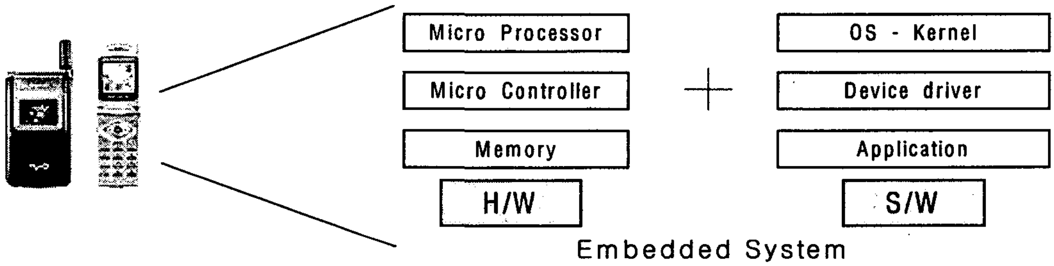

Fig.1 Embedded Systems of mobile phone

phone

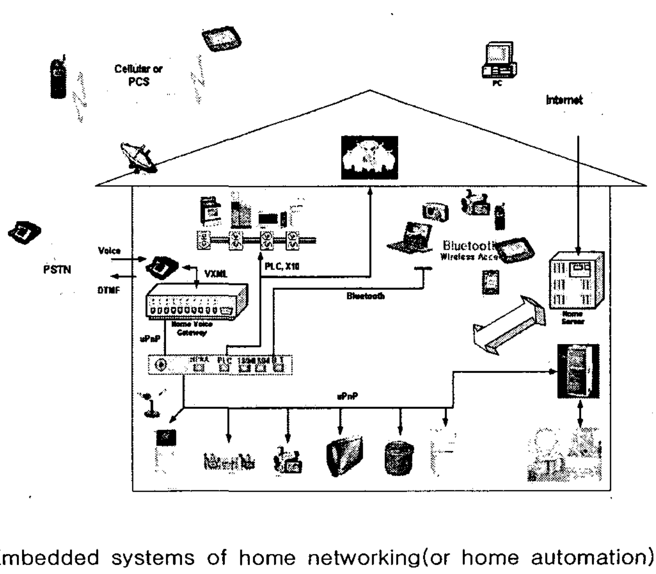

Fig.2 Embedded systems of home networking(or home automation)

2.1 Recent Trends in Electronic System Design

Among the main current research activities we could cite methodological support, multi-formalism specification, hard real-time analysis and design reuse.

Methodological support is required due to the complexity of modern systems, where not only powerful tools are needed, but also the methodologies that guide the design process.

The methodological support to the design process cab be conceived from various points of view. This means that the methodology should define the global strategy to follow in the *development of a system, the different design* phases considered and the criteria to move from one phase to the next in the design process. Specification requirements: analog vs. discrete, software vs. hardware, electronic vs. mechanic, control dominated vs. data dominated, real time requirements, safety critical systems, Several requirements can be identified in this context: (1)support for standard or widely used specification languages, (2)support for different application domains, and (3)methodological support for overall design process.

Additionally, most co-design environments focus on digital electronic systems, despite the fact that there are cases of embedded systems that contain parts for controlling continuous subsystems.

2.2 Hardware/Software Co-Design

The different hardware-software co-design approaches that can be found in the literature make use of a wide range of specification languages; several approaches make use of implementation oriented languages similar to C. There are also approaches that use state oriented specification languages.

An important issue for correct co-design is the search for a highly compositional lean formal approach that crosses the hardware/software boundary and enables us to keep up with the fast growth in the complexity and variety of electronic devices and their associated software.

In the hardware industry, simulation has often been considered synchronous with verification. The design process usually still consists of developing an implementation from an informal specification without the use of formal design techniques.

3. Co-design Methodology for Advanced Embedded Systems

The use of multiple formalisms for system specification, allowing co-modeling of the various subsystems in a seamless way. Also, Efficient system partitioning. Prototyping and testing *supported by tools* : the use of tools associated to different notations marks it possible to generate prototypes that can be evaluated in order to assess the correctness of the approach.

In this section we outline our framework for hardware/software co-design.

The process of modeling a system, albeit sequential or concurrent, timed or untimed, needs a suitable computational model. We take the view that a computation defines mathematically an abstract architecture upon which applications will execute. A system is a collection of agents(which is our unit computation), possibly executing concurrently and communicating (a)synchronously via communication links. Systems can themselves be viewed as single agents and composed into large systems. Systems may have timing constraints imposed at three levels: system wide communication deadlines, agent deadlines and sub-computation deadlines(within the computation of individual agent).

It is important to note that when we talk about system we do not make any distinction between software or hardware. We simply talk of a set of agent collaborating to achieve the desired behavior. Some of those agents *may* be realized(or implementation) in software and some in hardware.

4. Identification of Main Components and the Notation

The identification of the subsystems and the assignment of the corresponding notation are not straightforward. It can be that a preconceived idea of the design that influences the chosen notation and tools, or that a somehow required set of tools and notations determines the subsystems in which the design is split.

The final strategy recommended in the method is a mixture of both approaches. It has to be taken into account that the method and toolset aims to be as general possible, so that it can be tailored to different types of user and systems. The methodology toolset supports in fact a set of subsets of notations. The viable combinations depend on the possible design flow or least co-simulation facilities available for the selected subset.

4.1 Information Exchange between Subsystems

The second aspect that comes into consideration in the architectural design phase, after the identification of components, is the information exchange between the subsystems described in different formalisms. The logical information that needs to be exchanged consists of control information. This phase has to define the mapping between the logical information and the specific constructs in the selected notations.

4.2 Model Validation

The final sub-phase in the system architecture design is the initial validation of the design. At this level, the design description is a logical model, in which system resource are not considered. At this stage, it is possible to validate that the model fulfills system requirements, and that these are complete and consistent. In addition, it is also possible to make an initial evaluation of the subsystem cohesion and coupling, and estimate the communication overhead. Being a logical model, it is not possible to validate other requirements, such as very detailed functionality, timing, cost, performance, etc. Finally, validation of the notational decisions is also required, not only as far as their interfaces and external relations are concerned, but also to see if the internal characteristics of the subsystem natch the descriptive capabilities of the notation. Finally, this initial validation sub-phase should address the task of defining black box test cases based on the use of the system.

5. Co-Modeling and Detailed Design

This phase should undertake the refinement of the previously identified components. It is very important in design so to bear in mind the restrictions imposed in each vase on the interface by the other components in the system. That is, they have to comply with the information exchanges schema defined in the previous phase. Special care has to be taken if the detailed design of a component is used as an input to a co-modeling tool, because co-modeling is based on translation of whole components from one notation to the other, and severe restrictions are imposed on the sub-language covered by the translator.

5.1 Model Partitioning

The design is already detailed enough in this phase, and the step is to partition it into software modules running on different processors or hardware/software modules. As a result, a virtual prototype emerge prior to the final implementation. The resulting prototype at this point, takes into account hardware resources as well. It has been decided which modules will be implemented as software running in processors or as hardware. At this stage, it is possible to make an initial evaluation of additional requirements such as the time response of the system, performance, hardware costs, communication overhead, etc. For this purpose, it is possible to use tools such as model animators, emulators, hardware simulators, etc. In the case of hardware-software partitioning, to bridge the communication gap between hardware and software, an additional module is required that will enable the hardware signals to be realized by software and vice versa. The output of this phase is a virtual prototype that reflects the final partitioning scheme and fulfills the functional and non-functional constraints posed by the system requirements.

5.2 Targeting and Implementation

The final system implementation is the next step following a successful system partitioning. The final C code must be generated taking into account the operating system on which the software part will run. Operating system selection is critical in embedded system design since it must fulfil requirements such as real-time features, specialized schedule policies, multiple threads, etc., in a reasonably small kernel. Additionally, it must contain primitives that will ease the communication between hardware and software.

6. Conclusion

This paper present a method of construction of embedded systems based on hardware/software co-design. The proposed method enable to apply recent multimedia systems that is rapidly growing of the new information technology area. Also, the proposed method introduce efficient construction method for advanced embedded systems for the future.

For the future, more efficient embedded systems are needed for the sake of more advanced multimedia systems, digital contents including mobile digital contents. Therefore, we research above mentioned embedded systems at now.

References

[1] Frank Vahid and Tony Givargis, Embedded Systems Design - A Unified hardware/Software Introduction, John Wiley & Sons, Inc., 2002.

[2] Rae Earnshaw and John Vince, Digital Contents Creation, Spring Verlag, 2001.

[3] Ramesh Kari, David Goodman, System-Level Power Optimization for Wireless Multimedia Communication, Kluwer Academic Pub., 2002.

[4] Frank Nack,"The future in Digital Media Computing is Meta," IEEE Multimedia, vol.11, no.2,pp.10-13, April /June, 2004.

[5] Luca Benini, Francesco Menicheiii, Nauro Ovivier," A Class of Code Compression Schemes for Reducing Power Consumption in Embedded Microprocessor Systems," IEEE Trans. Computers, vol.53, no.4, pp .467-482, April, 2004.

[6] Kamal, Embedded Systems : Architecture, Programming and Design, McGraw Hill, 2002.

[7] Nagao Katashi, Digital Content Annotation and Tran-scoding, Artech House, 2003.

[8] Ralf Steinmetz, Klara Nahrstedt, Multimedia Systems, Spring Verlag, 2004.

[9] Abdulmotaleb El Saddik, Stefan Fischer, Ralf Stein-metz,"Reusable Multimedia Content in Web-Based Learning Systems," IEEE Multimedia, vol.8, no.3, pp .30-38, July/September, 2001.

[10] David Hwang, Bo-Cheng Lai, Patrick Schaumont, Kazuo sakiyama, Yi Fan, Shenglin Yang, Aiireza Hodjat, Ingrid Verbauwhede,"Design Flow for HW/SW Acceleration Transparency in the Thumb Pod Secure Embedded System," DAC'03, pp.60-65, Ana-heim, CA, USA, June 02-06, 2003.

[11] Haris Lekatsas, Jorg Henkel, Srimat Chakradhar, Ve-nkata Jakkula, Murugan Sankaradass," CoCo : A Hardware/Software Platform for Rapid Prototyping of Code Compression Technologies," DAC'03, pp .306-311, Anaheim, CA, USA, June 02-06, 2003.

[12] A. Krstic, W.-C. Lai, L Chen, K.-T. Cheng, S. Dey," Embedded Software-based Self-Testing for SoC Design," DAC'02, pp.355-360, NewOrleans, LA, USA, June 10-14, 2002.

박 춘 명

Chun-Myoung Park received B. S., M.S., and Ph.D. degree fromIn-ha University, in 1983, 1986, 1994 respectively. He is professor of major of Computer Engineering of School of Electrical Electronic and Information Engineering(EEI) from 1995 in Chung-ju National University. He was a visiting Scholar in 2002 ~ 2003 at School of Information and Computer Science(ICS) and Center for Embedded Systems(CECS) at University of California, Irvine (UCI) USA.

He is also Director of Global Media Research Institute(GMRI) from 2002. Also he is a director of Korea Digital Contents Society(DCS) from 2000, Korea Multimedia Society(KMMS) from 2001, and chungbuk branch in Institute of Electronics Engineers of Korea(IEEK) form 2001 respectively.

His research interests include Embedded Systems, Ubiquitous Computing, Advanced Multi-media Systems, Advanced Digital Contents, Next Generation Digital Logic Systems and Computer Architecture, and Microprocessor Application etc.