임베디드시스템에 기반을 둔 시스템온칩 구성에 관한 연구

박춘명*

*한국교통대학교

A Study on Constructing the System-on-Chip based on Embedded Systems

Chun-Myoung Park*

*Korea National University of Transportation

E-mail : cmpark@ut.ac.kr

요 약

본 논문에서는 입베디드세스템에 기초를 둔 시스템온칩을 구성하는 방법을 제안하였다. 제안한 방법은 이전의 방법에 비해 좀 더 콤팩트하고 효과적이다. 이 방법은 높은 수행시뮬레이션을 요구하고 하드웨어/소프트웨어 통합설계 툴을 사용하여 구현을 위한 실행 가능한 규격화된 적절함을 요구한다. 시스템 인터페이스 처럼 이미 존재하고 있는 부품의 재사용은 지원되지만, 작업 이후는 단지 하드웨어/소프트웨어 통합설계 툴의 프로그램에 의해 수행되어진다. 실제 설계 흐름은 모든 프로세스를 통하여 요구되는 구현으로부터 모든 설계 단계 사이의 궤환을 허용하게끔 설명되어진다. 향후 좀 더 진보된 임베디드시스템에 기초를 둔 시스템온칩을 구성하는 방법이 요구된다.

ABSTRACT

This paper presents a method of constructing the system-on-chip(SoC) based on embedded systems. The proposed method is more compact and effectiveness than former methods. The requirements generation start high level performance simulation and then passes to an executable specification suitable for implementation using a hardware/software co-design tool. The reuse of pre-exiting components is supported, as well as synthesis of the system interface, but only after much work is done to program the hardware/software co-design tool. The actual design flow described allows feedback among all design levels, e.g. from implementation up to requirements, throughout the process. In the future, it is necessary to development the advanced method of constructing system-on-chip based on embedded systems.

키워드

Embedded systems, system-on-chip(SoC), co-design, performance, etc.

Ⅰ. 서 론

최근에 21세기 IT 분야의 주요 기술로 부각되고 있는 임베디드시스템과 이에 근간을 둔 시스템온칩 구현에 대한 연구가 활발하게 진행되고 있다.[1-3] 특히, 임베디드시스템의 하드웨어/소프트웨어 통합설계는 시스템온칩의 중요한 방법이다.[4-6] 또한, 최근의 분산환경과 모바일 환경에서의 임베디드시스템의 적용은 매우 중요하다.[7-9] 따라서, 본 논문에서는 하드웨어/소프트웨어 통합설계에 대한 내용과 이에 근간을 둔 시스템온칩 구현에 대한 방법에 대해 논의하였다.

Ⅱ. 설계흐름에 기초를 둔 작업흐름

다음 그림 3-1은 잘 정돈된 설계와 설계흐름에 기초를 둔 작업 흐름 사이의 다름 점들을 보여준다. 그림 3-1에서 전자는 CAD 중심적이고 설계 행위의 이상적인 면을 보여주지만, 후자는 산업계에서 사용되는 설계흐름을 보여준다. 즉 작업의 자세한 공정 설명과 어떻게 수반되는 방법들에 관련된 것이다.[7] 작업흐름에서의 각각의 개별적인 단계는 활동적이다. 활동적인 것은 인간뿐만 아니라 프로그램 또는 상호동작적인 프로그램을

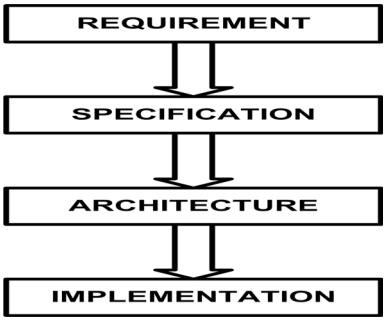
사용하여 수행된다. 작업흐름 패러다임은 특히 SoC에 대해 유효하다. 일찌기 SoC 설계의 국면은 주로 인간의 작업을 포함하고 있는 반면에 후자는 주된 프로그램으르 포함하는 단계이다. 향 후 제공되는 잘 정돈된 작업에 대한 실행 가능한 작업 흐름은 동적이다.[8] 본 논문에서는 Coware간의 동작적인 규격을 요구하는 시스템 레벨사이의 반복적인 정립을 할 수 있는 시스템 레벨 시뮬레이션에 대한 매뉴얼을 설명한다. 설계흐름에 기반을 둔 작업 흐름과 더불어 시스템 설계 팀은 고장난 설계 공정의 다양한 부분에 대한 같은 환경을 다룬다.

Ⅲ. CoWare를 수반하는 하드웨어/ 소프트웨어 통합 설계

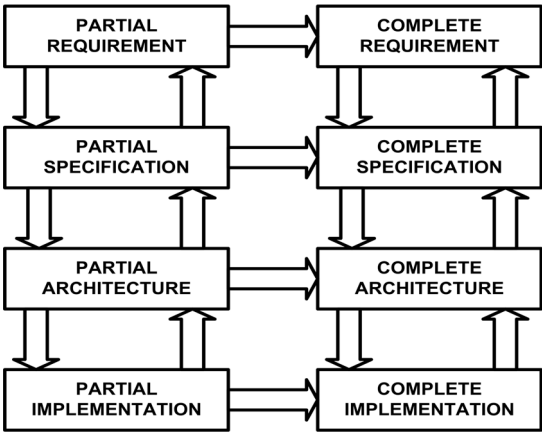
모든 시스템의 부분으로써 분석되어 져야 할 노드 제품의 플래너(Planner)는 소프트웨어 프로그램이 가능한 CPU 코어(Core) 상에서 제일 좋게 실행된다. 그리고 그 기능은 하드웨어 칩상에 가장 잘 구현된다. 이들 아키텍쳐 결정은 요구되는 칩상의 메모리 총 용량, 전체적인 다이(die) 크기, 시스템 수행, 전력 소비 등에 극적인 충격을 가질 것이다. 설계 과정에 늦게 시스템 분할의 지연은 아키텍쳐 결정이 게이트들과 더불어 만들어 진다. CoWare 하드웨어/소프트웨어 통합설계 환경은 존재하고 있는 언어를 사용하여 하드웨어/소프트웨어 부품의 상호 규격을 허용한다.[6-7]

Ⅳ. 결론

본 논문에서는 최근에 21세기 차세대 IT 기술의 주요 기술중에 하나인 임베디드시스템에 기반을 둔 시스템온칩 구성에 대한 방법에 대해 논의하였다. 시스템 레벨 설계 툴의 산업계 채택에서의 주요한 단계는 혼합 절대 레벨을 가로지르는 IP 블록의 정당성과 모델 생성의 개선이다. 특별한 블록에 대한 가장 표현 가능한 언어 형태에서의 존재하고 있는 블록이 설명되어지므로, 많은 형식(Format)과 코딩 스타일이 사용되어진다. 이러한 이종간의 실현 스터일은 어려운 절대 부품의 라이브러리의 생성 과정을 만든다. 필요한 것은 정당성의 방법을 개선시키고, 자동적으로 생성하고, 시스템 레벨 프레임 작업안에 존재하는 블록을 캡슐화 하는 것이다. 존재하고 있는 실현 레벨 모델에 반한 손으로 생성된 정대 색인에 대한 새로운 정당성을 위한 방법과 좀 더 좋은 툴은 라이브러리 생성 과정에 대한 속도를 필요로 한다. 이 경우에 시스템 레벨 시뮬레이션은 결정적으로 중요하다. 향 후 손으로 생성하는 과정은 실현 시뮬레이션 의 동작의 관찰로부터 시스템 레벨 모델을 생성하는 자동 툴에 의해 수행되어질 것이다. 제안한 방법은 기존의 방법에 비해 효과적이며, 규칙성과 신뢰성이 개선되었다. 하지만, 향 후 좀 더 진보된 방법의 연구를 필요로 한다.



(a) Refinement based design flow



(b)Workflow based design flow

그림 4-1. 설계 흐름의 비교
Fig. 4-1. The comparison of design flow

참고문헌

[1] Edward A. Lee, “What’s Ahead for Embedded Software?,“ IEEE Computer, September 2012, pp.18-26

[2] Wayne Wolf, “What Is Embedded Computing?” , IEEE Computer, pp 136-137, January, 2010.

[3] Steve Furber, ARM System-on-chip Architecture, Adison-Wesley, 2011.

[4] G. De Micheli and M. Sami, hardware/ Software Co-Design. Kluwer Academic Publishers, Norwell, MA, 2000.

[5] F. Balarin, M. Chiodo, P. Giusto, H Hsieh, A. Jurecska, L. Lavagno, C. Passerone, A. Sangiovanni-Vicentelli, E. Sentovich, K. Suzuki and B. Tabbara, hardware-Software Co-Design of Embedded Systems - The Polis Approach, Kluwer Academic Publishers, MA, 2013.