

## 혼합 설계 환경에서의 하드웨어-소프트웨어 인터페이스 설계

장 준 영, 배 영 환

전자통신연구원 회로소자연구소 집적회로연구부 시스템설계자동화팀

전화 : 042-860-6680

### A Hardware-Software Interface Design in the Codesign Environment

System Design Automation Team  
ETRI-Micro-Electronics Technology Laboratory  
E-mail : jychang@etri.re.kr

#### Abstract

In this paper, A target architecture and interface synthesizer are proposed for processor-embedded codesign. The target architecture has the form of ARM processor based on AMBA. The interface synthesizer automatically generates an interface circuit for the communication between HW and SW. A memory map is used as the communication channel and an interrupt-based interface is applied for synchronized communication between HW and SW modules. In order to verify the function and performance of proposed target architecture and the interface synthesizer, practical test example is applied. Experimental results show the proposed interface synthesizer functioned correctly in the HW/SW codesign environment.

#### I. 서론

최근 디지털 시스템의 복잡도가 증가하고 다양한 시스템 설계 기술이 등장함에 따라 하드웨어와 소프트웨어를 동시에 설계함으로써 효율적이고 빠른 설계를 가능하게 하는 혼합 설계에 대한 관심이 집중되고 있다.

하드웨어와 소프트웨어가 혼합된 설계(codesign)는 컴퓨터 제어 통신 기기 및 멀티미디어 기기, 정보 기기 단말기를 비롯한 대부분의 전자 시스템에 이용되며, 고밀도 고성능의 복잡한 시스템을 초고집적으로 설계하기 위해서는 하드웨어와 소프트웨어가 혼합된 시스템을 설계하는 기술이 필요하다[1,2].

하드웨어-소프트웨어 혼합 설계 과정은 시스템 기술 언어를 사용하여 시스템의 동작을 시스템 수준에서 기술하고 시스템 시뮬레이션으로 시스템 사양을 검증한 후 (2) 시스템을 하드웨어와 소프트웨어 요소로 분할하고 (3) 하드웨어 요소는 상위수준합성과 논리합성으로 합성, (4) 소프트웨어 요소는 소프트웨어 합성기와 컴파일러를 통해 내장된 마이크로 프로세서에서 실행될 코드를 생성한 다음 (5) 시스템 통합(system integration)을 통해 각 요소를 통합하여 혼합 시뮬레이션이나 하드웨어 prototype 구현을 통해 시스템의 동작을 검증한다. 하드웨어-소프트웨어가 혼합된 프로세서 내장형 시스템의 통합 설계에서 분할된 이종의 시스템 모듈 즉, 하드웨어와 소프트웨어 모듈간의 데이터 및 제어 신호의 통신을 위해서 인터페이스를 생성하는 과정에서 많은 오류가 발생하고 긴 시간을 필요로 하는 작업이다[2,6]. 또한 목적 원판의 구성에 따

라서 하드웨어 소프트웨어 인터페이스 설계가 달라질 수 있다. 하드웨어-소프트웨어 인터페이스 설계는 혼합 설계 환경에서 필수적인 부분으로 시스템의 성능에 큰 영향을 미칠 뿐 아니라 최종 설계 결과의 성공 여부에 큰 영향을 미치는 중요한 매우 중요한 단계이다. 본 논문에서는 최근의 정보단말기 및 멀티미디어 기기, IC 카드 등에 이용되고 있는 저전력 32 비트 마이크로 프로세서인 ARM7TDMI[3]와 On-chip 버스 구조를 가진 AMBA(Advanced Microprocessor Bus Architecture)[4,5]로 구성된 목적 원판을 설계하고, 하드웨어-소프트웨어 데이터 통신 부담을 줄여서 전체 시스템의 성능을 만족하는 하드웨어-소프트웨어 인터페이스 모듈을 자동으로 생성하는 인터페이스 합성기를 구현한다.

## II. 혼합 설계를 위한 목적 원판

혼합 설계를 위한 목적 원판 구조는 AMBA를 기반으로 ARM7TDMI 프로세서 코어를 탑재한 구조이다 [3]. AMBA는 ASB(Advanced System Bus)와 APB(Advanced Peripheral Bus)로 구성된다.

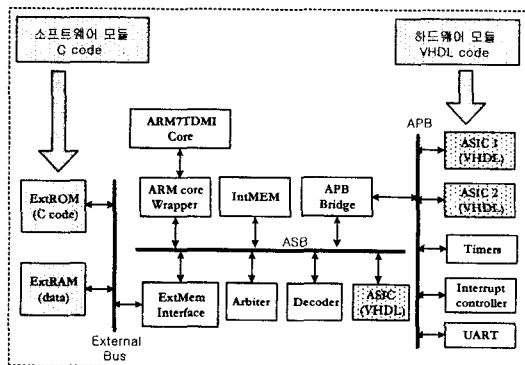


그림 1. ARM/AMBA 기반 목적 원판

ASB는 고속의 데이터 access를 필요로 하는 시스템 모듈들로 구성되며, decoder, arbiter, 외부 메모리 인터페이스, 내부 메모리, ARM 코어와 ASB를 연결해주는 Wrapper, APB 버스와의 다리 역할을 해주는 APB Bridge등으로 구성된다. APB는 주변장치를 위한 시스템 버스로서, Timer, 인터럽트 제어기, UART등의 기본적인 주변 모듈이 연결되면 하드웨어 모듈들이 APB에 연결된다. APB는 주로 느린 데이터 access를 필요로 하는 모듈로 구성되며 속도가 느린 반면 매우 적은 양의 전력을 소모한다. 그림 1과 같은 목적 원판 구조에 분할된 하드웨어는 APB나 ASB에 ASIC 모듈로 매핑하고 소프트웨어 모듈은 ROM 데이터로 만들어

메모리에 적재하여 ARM 코어에서 프로그램을 실행한다. 다음은 하드웨어와 소프트웨어 모듈사이 데이터 송수신을 위한 하드웨어와 소프트웨어간 인터페이스에 대해서 설명한다.

## III. 하드웨어-소프트웨어 인터페이스

하드웨어와 소프트웨어간의 인터페이스 합성 단계에서는 구현하고자 하는 시스템을 하드웨어 부분은 VHDL로, 소프트웨어 부분은 C 언어로 작성한다. 인터페이스 합성 과정을 통해서 그림 2와 같이 소프트웨어 부분에는 인터페이스를 위한 device driver 함수를 자동으로 합성하며, 하드웨어 부분에서는 버스 구조에 근거한 하드웨어 인터페이스 모듈을 자동으로 생성한다. 이 인터페이스 모듈에서는 버스 프로토콜에 따라 데이터 버스와 주소 버스를 access하며 각 데이터 신호에 할당된 주소에 따라 데이터 통신을 수행한다.

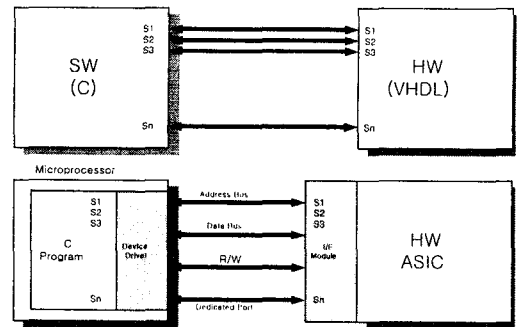


그림 2. 하드웨어-소프트웨어 인터페이스

### 3.1 인터페이스 채널

소프트웨어-하드웨어간의 인터페이스는 동일한 통신 프로토콜로 구현된 경우에는 별도의 인터페이스 회로가 필요 없지만 그렇지 않을 경우 이들 사이에 인터페이스 모듈을 합성하여 삽입한다. 하드웨어-소프트웨어의 인터페이스는 소프트웨어가 수행하되 마이크로 프로세서의 인터페이스를 이용하여 하드웨어 모듈과 통신을 하는데 하드웨어 부분에는 마이크로 프로세서와 통신을 위한 인터페이스 모듈을 자동으로 생성하여 삽입하고 소프트웨어 부분에는 디바이스 드라이버를 만들어 주어야 한다. 하드웨어-소프트웨어간의 통신 인터페이스 채널은 전용 포트, 버스, FIFO, 공용 메모리를 이용하는 방법이 있다[2]. 전용 포트를 이용하는 방법은 그림 3에서와 같이 point-to-point 연결로서 다른 모듈간에 전용 포트를 이용하는 방법이다. 메모리 주소 공간상에서 특정 주소를 할당하여 할당된

주소를 통해서 주소 버스와 데이터 버스를 이용하여 통신하는 메모리 맵에 의한 방식 등이 있다.

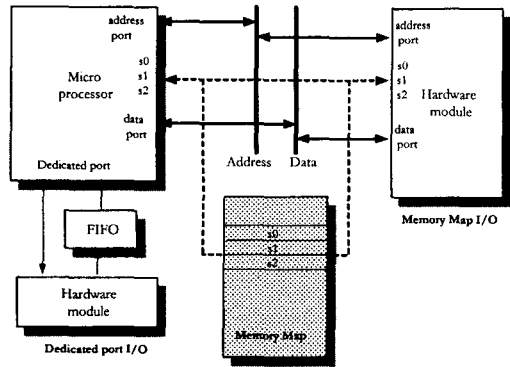


그림 3. 메모리 맵을 이용한 인터페이스 채널

본 논문에서는 전용 포트와 같이 포트의 수에 제한을 받고 별도의 포트 할당 알고리즘이 필요로 하는 문제를 해결하기 위해서 소프트웨어 모듈과 하드웨어 모듈사이의 인터페이스를 위해 메모리 맵을 이용한다.

### 3.2 인터페이스 프로토콜

하드웨어와 소프트웨어 사이의 송수신 데이터들의 동기화를 위한 인터페이스 프로토콜이 필요하다. 인터페이스 프로토콜은 설계 제약 조건에 따라 메시지 전송, 공용메모리, 인터럽트에 의한 동기화 기법 등이 있다[2]. 목적 원판에서는 인터럽트가 가능한 구조이고 또한 다른 방법 보다 빠른 인터페이스 방법이므로 동기화 프로토콜로 인터럽트 동기화 기법을 이용한다. 목적 원판에서 적용된 인터럽트에 의한 동기화 방법에 대해서 설명한다.

1. 하드웨어 모듈에서 버스에 데이터를 적재한다.
2. 하드웨어 모듈에서 인터럽트 신호를 마이크로프로세서에 전달한다.
3. 마이크로프로세서는 인터럽트 신호를 받아서 소프트웨어 모듈에 인터럽트 신호를 보낸다.
4. 소프트웨어 모듈은 인터럽트 신호를 인식하고 인터럽트벡터에 의해 정해진 인터럽트 서비스 루틴을 호출하고 데이터를 처리하여 정해진 메모리 공간에 자료를 적재한다.
5. 소프트웨어 모듈은 인터럽트 서비스 루틴의 종료 신호를 마이크로프로세서에 전달한다.
6. 마이크로프로세서가 하드웨어 모듈에 인터럽트 종료 신호를 보내면 하드웨어 모듈은 정해진 메모리 공간에서 결과 데이터를 읽고 다음 작업을 처리한다.

## IV. 인터페이스 합성기

ARM/AMBA기반의 목적 원판에 하드웨어-소프트웨어 인터페이스 모듈을 자동으로 생성하는 인터페이스 합성 방법에 대해서 설명한다. 인터페이스 합성기는 통신 인터페이스 채널로 메모리 맵을 이용하고, 송수신 데이터의 동기화를 위해서 인터럽트 방법을 이용한다. 먼저 인터페이스 모듈에서 사용하는 인터페이스 테이블을 읽어서 인터페이스에 필요한 내부 신호를 생성한다. 이때 사용하는 인터페이스 정보는 입출력 신호, 데이터의 크기, 데이터의 방향(HW->SW, SW->HW), 데이터의 타입, 신호의 동작 코드, 신호에 할당된 메모리의 주소, 전달되어지는 실제 값 등을 포함한다. 인터페이스에 필요한 메모리 맵을 할당하고 할당된 메모리 맵을 통해서 Send 함수와 Receive 함수를 이용하여 데이터를 송수신 한다. 그림 4에서 하드웨어 인터페이스 모듈에서는 Send 함수를 이용하여 제어 신호와 데이터를 메모리 맵에 저장하고 인터럽트를 발생한다. Receive 함수에서는 소프트웨어 인터페이스 모듈에서 인터럽트 종료 신호가 발생되면 계산된 결과를 하드웨어 모듈에서 받아들인다.

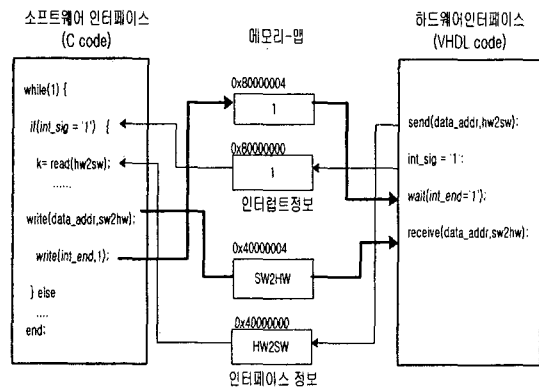


그림 4. 하드웨어-소프트웨어 인터페이스 과정

소프트웨어 인터페이스 모듈의 Read 함수에서는 인터럽트 신호가 발생되면 정해진 메모리 맵에서 데이터를 읽어서 해당하는 인터럽트 처리 루틴을 수행하고 Write 함수에서 계산된 결과를 정해진 메모리 주소에 적재한 후에 인터럽트 종료 신호를 발생한다. 따라서 메모리 맵과 인터럽트를 기반으로 한 Send/Receive 함수에 의해서 하드웨어와 소프트웨어간의 인터페이스가 이루어지는 것을 알 수 있다. 다음은 간단한 예제를 통해 인터페이스 합성 결과를 검증한다.

### V. 목적 원판에 적용 및 결과

인터페이스 합성기를 거쳐서 생성된 하드웨어 인터페이스와 소프트웨어 인터페이스인 디바이스 드라이버를 ARM/AMBA 기반의 목적 원판에 적용하여 시뮬레이션한 결과를 기술한다. 인터페이스 합성기에 의해 생성된 하드웨어 및 소프트웨어 인터페이스 모듈을 목적 원판에 적용하였다. 목적 원판에서 이용되는 프로세서인 ARM 코어는 40MHz에서 동작하는 스마트 모델을 이용하였고 AMBA 주변 모듈들은 VHDL로 설계된 시뮬레이션 및 합성 모델이다. 그림 5는 구현된 목적 원판의 Schematic Diagram이다.

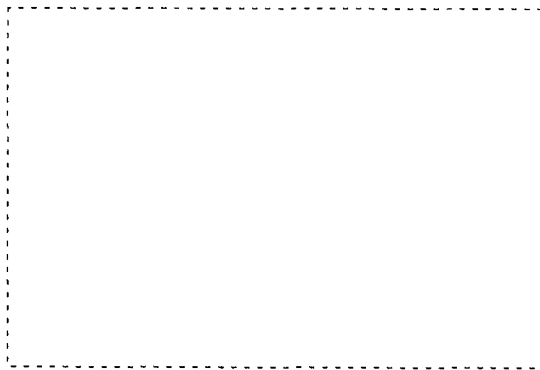


그림 5. ARM/AMBA 기반의 목적 원판

그림 6은 목적 원판과 인터페이스 합성기의 동작을 검증하기 위해서 1 word 데이터를 하드웨어 모듈에서부터 소프트웨어 모듈로 송수신 과정을 시뮬레이션 한 결과이다.

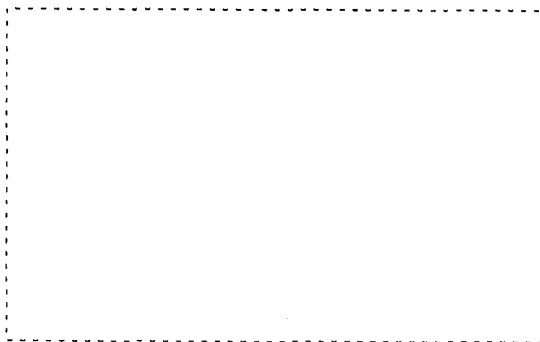


그림 6. 인터페이스 시뮬레이션 결과

그림 6 에서 하드웨어 모듈의 s\_val에 0000FFFF가 적재되고 인터럽트 신호인 intben이 활성화되면 ARM 코어에서 수행 중인 프로그램에 따라 메모리 맵 상의

XXXX0004 위치에서 0000FFFF를 읽는다. 소프트웨어 모듈에서는 처리된 데이터를 하드웨어 모듈의 r\_val의 주소인 XXXX0008에 적재하면 하드웨어 모듈에서 데이터를 받아들인다. 시뮬레이션의 결과와 같이 목적 원판에서 메모리 맵과 인터럽트 신호에 의해 하드웨어와 소프트웨어의 인터페이스가 정확하게 동작하는 것을 알 수 있다. 따라서 인터페이스 합성기를 ARM/AMBA를 이용한 SoC(System-On-Chip) 설계의 응용 분야인 정보 기기 단말기, 멀티미디어 기기 및 IC 카드 등의 분야에 이용할 수 있다.

### VI. 결론

본 논문에서는 시스템 설계에서 필요한 상위 수준에서의 빠른 prototyping을 위해서 ARM/AMBA 기반의 목적 원판을 설계하였고 하드웨어 및 소프트웨어간의 인터페이스 모듈을 자동으로 생성하기 위한 인터페이스 합성기를 구현하였다. 통신 인터페이스 채널로서 메모리 맵을 이용하고, 인터럽트 방식을 이용한 인터페이스 프로토콜을 이용하였다. 또한 인터페이스 합성기를 통해서 생성된 하드웨어 및 소프트웨어 모듈들이 최근에 정보통신용 단말기, 멀티미디어 기기 및 IC 카드 등에 많이 사용되고 있는 ARM/AMBA 기반의 목적 원판에서 잘 동작하는 것을 시뮬레이션을 통해서 확인하였다. 추후 다중 프로세서 다중 하드웨어 모듈로 구성된 혼합 설계 환경에 적합한 인터페이스 합성기의 개발에 대한 연구를 지속할 예정이다.

### 참고 문헌

- [1] Felice Balarin, et al, Hardware-Software Co-Design Of Embedded Systems, The POLIS Approach, Kluwer Academic Publishers, 1997.
- [2] Ralf Niemann, Hardware/Software Co-design For Data Flow Dominated Embedded Systems, Kluwer Academic Publishers, 1998.
- [3] Steve Furber, ARM System Architecture, Addison-Wesley, 1996.
- [4] ARM Ltd, Introduction to AMBA , Document Number ARM DVI 0010A, 1996.
- [5] ARM Ltd, Example AMBA System, User Guide , Document Number ARM DUI 0092A, 1998.
- [6] B. Lin, S. Vercauteren, H. De Man, Embedded architecture co-synthesis and system integration, Hardware/Software Co-Design, 1996. (Codes/CASHE '96), Proceedings., Fourth International Workshop on , 1996.