SystemC 를 이용한 OpenCable™ Copy Protection Module 의 Physical Layer 설계

이정호, 이숙윤, 조준동
성균관대학교 전기전자컴퓨터공학과
e-mail : mazzana@vada1.skku.ac.kr

A Design Of Physical Layer For OpenCable Copy Protection Module Using SystemC

Jung-Ho Lee, Suk-Yun Lee, Jun-Dong Cho
School of Information & Communication Engineering, SungKyunKwan University

요 약

본 논문은 미국 차세대 디지털 케이블 방송 표준 규격인 오픈케이블(OpenCable™)의 수신제한 모듈인 CableCard 의 Physical Layer 를 SystemC 의 TLM(Transaction Level Modeling)과 RTL(Register-Transfer Level) 모델링 기법으로 설계하였다. 본 논문에서 설계한 CableCard 의 Physical Layer 는 PCMCIA Interface, Command Inteface 그리고 MPEG-2 TS Interface 로 구성된다. CableCard 가 전원이 인가될 때, 카드 초기화를 위하여 동작하는 PCMCIA 인터페이스는 16 비트 PC 카드 SRAM 타입으로 2MByte Memory 와 100ns access time 으로 동작할 수 있게 설계하였다. PCMCIA 카드 초기화 동작이 완료된 후, CableCard 의 기능을 수행하기 위하여 두 개의 논리적 인터페이스가 정의되는데 하나는 MPEG-2 TS 인터페이스이고, 다른 하나는 호스트(셋톱박스)와 모듈 사이의 명령어들을 전달하는 명령어 인터페이스(Command Interface)이다. 명령어 인터페이스(Command Interface)는 셋톱박스의 CPU 와 통신하기 위한 1KByte 의 Data Channel 과 OOB(Out-Of-Band) 통신을 위한 4KByte 의 Extended Channel 로 구성되고, 최대 20Mbits/s 까지 동작한다. 그리고 MPEG-2 TS 는 100Mbits/s 까지 동작을 수행할 수 있게 설계하였다. 설계한 코드를 실행한 후, Cadence 사의 SimVision 을 통해서 타이밍 시뮬레이션을 검증하였다.

1. 서론

현재 우리나라는 아날로그 방송시스템에서 디지털 방송 시스템으로 변모해가는 과도기에 접어들고 있다. 이러한, 방송환경의 급격한 변화와 더불어 디지털방송 콘텐츠 산업의 중요성은 날로 부각되어 가고 있으며, 이와 더불어 중요시되고 있는 것이 막대한 자금과 노력을 투자하여 제작된 디지털방송 콘텐츠에 대한 불법복제 방지기술(CP : Copy Protection) 및 이를 위한 가입자 인증(CAS : Conditional Access System)이다. 내장형 CAS 가 해킹되었을 때는 그것으로부터 사업자를 보호할 방법이 없지만, CableCard 가 있을 경우 이러한 극단적인 상황에서도 큰 비용과 고통없이 CableCard 만을 바꿔 버린다면, 해킹으로부터 사업자를 근원적으로 보호할 수 있다.

이런 배경에서, 미국에서는 2005 년부터 적용을 목표로 미국 차세대 디지털 케이블방송 표준 규격을 CableLab™ 사에서 CableLab™ 프로젝트를 운용하고 있으며, 현재 우리나라에서도 이들 표준을 포함하여 OpenCable™ 방식을 국내 디지털 유선방송 송수신 정합 표준규격으로 채택하여 제정 공포하였다.

본 논문에서 설계된 SystemC 를 이용할 경우 전송 수준 모델링(Transaction-Level Modeling, TLM)이 가능하다. TLM 은 클럭-사이클과 같은 세밀한 수준이 요구되고 데이터 통신이 실제 구현되는 레지스터 트랜스터 수준 모델링(Register-Transfer Level Modeling)보다는 데이터 전송의 기능성이 더욱 강조되어서 모듈의 개별적인 입축력 기반 인터페이스 방식에 비하여 불필요한 상세함을 억제 함으로써, 빠른 시뮬레이션 속도를 얻을 수 있는 장점이 있다.

본 논문의 전체 구성은 다음과 같다. 2 장에서는 CableCard 의 기본 구조에 대해서 살펴보았으며, 3 장에서는 CableCard 의 Physical Layer 에 대한 설명과 PCMCIA Interface 및 Command Interface 설계에 대해서 설명하였다. 4 장에서는 C++ 기반의 SystemC 를 이용해서 설계한 본 연구의 시뮬레이션 결과를 나타내었다. 7 장에서는 결론과 앞으로의 진행 방향에 대해서 설명하였다.

본 논문에서 설계한 PCMCIA Interface 는 아래 그림 1 에서 OCI-H2 에 해당된다.

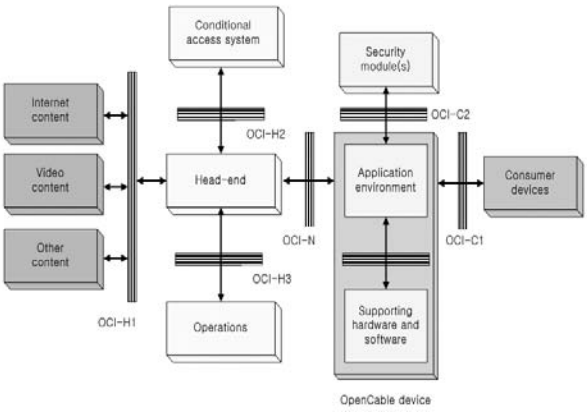

그림 1. OpenCable 구성

2. CableCard 의 기본 구조

아래 그림 2 는 HDTV 용으로 사용 되어 지는 디지털 셋톱박스에 탈부착이 가능한 형태인 CableCard 의 전체적인 기본 구조를 블록별로 나타낸 것이다. SystemC 를 이용해서 설계한 부분은 PCMCIA 에서 CableCard 로 성격이 바뀌게 하는 PCMCIA 와 그것의 Interface Logic 의 Physical Layer 이다. Interface Logic 은 MPEG-2 의 데이터 이동과 쌍방향 통신이 가능하게 하는 OOB 프로세서의 데이터가 송수신될 수 있게 설계하였다.

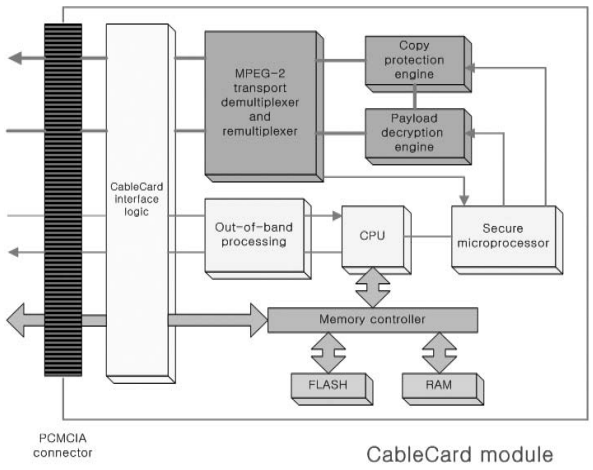

그림 2. CableCard Security Module Block Diagram

3. CableCard 의 Physical Layer 구조

물리적 인터페이스 상에 포함되는 인터페이스는 PCMCIA Interface 와 2 개의 논리 인터페이스가 정의된다. 논리 인터페이스 중 첫 번째는 MPEG-2 TS(Transport Stream) 인터페이스이고, 두 번째인 명령어 인터페이스는(Command Interface) 호스트와 모듈 사이에서 명령을 전달하는 역할을 한다.

MPEG-2 TS 인터페이스는 MPEG-2 TS 패킷을 양방향으로 전달한다. 모듈이 트랜스포트 스트림에서 모든 서비스의 액세스를 허용하고 호스트가 이 서비스를 선택한 경우, 서비스를 전달하는 패킷은 디스크램블 상태로 복귀하고 다른 패킷은 수정되지 않는다. 트랜스포트 스트림 인터페이스 상에서 상수는 모듈을 통해 지연되고, 대부분의 경우 관련된 모든 물리적 계층 조건 논리는 보존된다.

명령어 인터페이스는 모듈과 호스트에서 실행되는 어플리케이션 간에 모든 통신을 전달한다. 이 인터페이스는 비트 설정 상황에 따라서 Data Channel 과 Extended Channel 으로 나누어진다. Data Channel 을 통해서 호스트 – CableCard 간에 주고 받는 데이터들은 CPU 간의 통신으로써, Resource 를 주고 받는다. 반면에, Extended Channel 을 통해서 호스트 – CableCard 간에 주고 받는 데이터들은 OOB(Out-Of-Band) 메시지들을 주로 송수신 한다.

3-1. PCMCIA Interface 설계

본 논문에서 설계한 PCMCIA Interface 는 호스트와 CableCard 간의 데이터 전송을 위하여 듀얼 모드로 설계하였다. PCMCIA 에 전원이 들어오면 PCMCIA 내부 메모리와 호스트와의 데이터 송수신을 통하여 모듈이 CableCard 임을 확인하는 초기화를 실행한다. PCMCIA 내부 메모리에는 PCMCIA 가 CableCard 임을 나타내는 CIS 정보가 저장되어 있고, Configuration Register 에는 I/O 인터페이스로 전환되었을 때 카드 상태에 관한 정보를 호스트와 송수신한다. 초기화를 마치고 나면, 초기화시 사용되던 핀들의 일부가 재배열 된다. 아래 그림 3 에서 Control 블럭은 핀 재배열이 가능하도록 설계 되었다. 핀 재배열이 되면, CableCard 의 MPEG-2 Interface 와 Command Interface 가 PCMCIA Interface 를 통하여 호스트와 통신할 수 있게 된다.

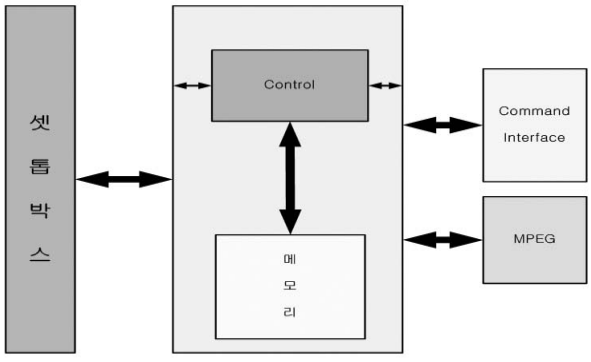

그림 3. PCMCIA 호스트-CableCard Interface 구조

옆의 그림 2 에서 보면 알 수 있듯이 PCMCIA 는

호스트와 모듈간의 인터페이스 역할을 하고 있다. PCMCIA 는 호스트에서 받은 신호를 내부 컨트롤 블록에 의하여 메모리와의 데이터 전송과 모듈과의 데이터 전송을 결정한다. 또한, PCMCIA 가 I/O 인터페이스로 변화되었을 때, 바뀌는 핀은 호스트와의 관계에서만 정의해 주면 된다. 메모리나 모듈과 연결된 핀들은 호스트에서 받은 핀을 그대로 내 보내주는 것으로 설계가 되었다. 각각에 쓰인 signal 들은 초기화에 관련된 것들, 카드 read 에 관련된 것들, 카드 write 에 관련된 것들, 핀 재배열에 관련된 것들이 있다.

설계된 CableCard 의 PCMCIA Interface 모듈이 지원하는 사양은 다음과 같다.

1) 모듈에 대한 8 비트 병렬 입력 및 별도의 8 비트 병렬 출력, 제어 신호 그리고 바이트 클럭으로 구성되는 MPEG-2 데이터 트랜스포트 스트림 인터페이스.

2) 8 비트 양방향 데이터 버스, 어드레스 그리고 제어 신호로 구성되는 호스트와 모듈간 명령 인터페이스.

3) 호스트가 모듈 상의 카드 정보 구조(Card Information Structure : CIS)를 읽어 들이고 모듈을 정상 작동 모드로 구성하게 하는 attribute 메모리 인터페이스.

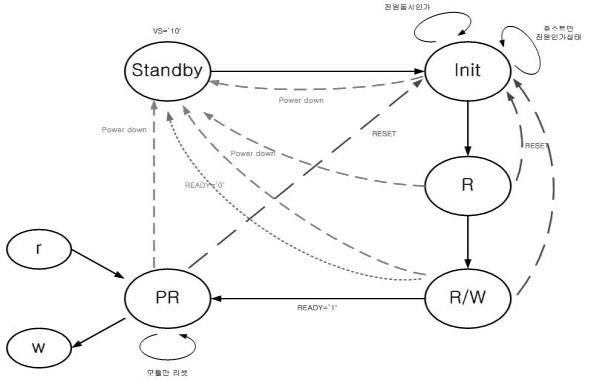

그림 4. PCMCIA Interface 의 동작에 따른 상태도

그림 4 는 위의 내용에 따른 PCMCIA Interface 의 초기화와 I/O Interface 전환되기까지의 과정을 나타낸 전체적인 상태도를 나타낸 것이다.

3-2. 명령어 인터페이스(Command Interface)의 설계

하드웨어 인터페이스는 PC 카드 인터페이스 상에 4 바이트씩의 어드레스 공간을 가지는 여러 개의 레지스터로 구성된다. 두 개의 어드레스 라인(A0, A1)만 인터페이스에 의해 사용된다. 아래 그림 5 은 설계한 Physical Layer 를 중심으로 설계한 Link Layer 와 PCMCIA 간의 도식도이다. 이동하는 데이터들은 포인터를 사용해서 데이터의 사이즈와 버퍼의 시작번지를 주고 받는 형태로 설계되었다.

그리고, PCMCIA 와 Physical Layer 간에는 각각의 레지스터에 속해 있는 비트 설정 상황에 따라 Data Channel 데이터와 Extended Channel 데이터가 송수신될 수 있게 설계되었다.

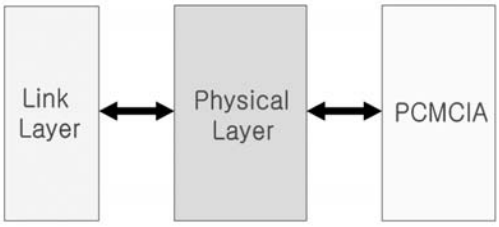

그림 5. PCMCIA - Link Layer 간의 Description

아래 표 1 은 인터페이스 상에 존재하는 4 바이트의 레지스터를 나타낸 것이다.

오프셋	레지스터
0	데이터 레지스터
1	명령/상태 레지스터
2	크기 레지스터(LS)
3	크기 레지스터(MS)

표 1. 하드웨어 인터페이스 레지스터의 맵

데이터 레지스터를 통해서 모듈과 데이터를 송수신하고, 오프셋 1 에서 읽으면 상태 레지스터이고 오프셋 1 에서 쓰면 명령 레지스터이다. 크기 레지스터에서 오프셋 2 는 하위 바이트이고 오프셋 3 은 상위 바이트를 나타내는 것이다.

Bit	7	6	5	4	3	2	1	0
	DA	FR	R	IIR	R	R	WE	RE

표 2. 상태 레지스터의 비트 맵

DA(데이터 있음) : 모듈이 호스트로 전송할 데이터를 가지고 있을 경우 '1'로 설정된다.

FR(사용가능) : 모듈이 호스트로부터 자유롭게 데이터를 수신할 수 있고, 모듈의 하드웨어 재설정이나 RS 명령에 의한 초기화가 끝날 때에 '1'로 설정된다.

IIR : POD 모듈이 인터페이스가 재초기화되는 것을 요청하기 위해서 사용되어진다. 이 비트는 data channel 과 extended channel 둘 다에 존재한다. POD 모듈이 IIR flag 를 셋팅할 때, POD 는 RS flag 가 셋팅될 때 IIR flag 를 또한 리셋해야한다.

WE(쓰기 오류)와 RE(읽기 오류) : 읽기 또는 쓰기 동작시 크기 오류를 나타낸다.

Bit	7	6	5	4	3	2	1	0
	DAIE	FRIE	R	R	RS	SR	SW	HC

표 3. 제어 레지스터의 비트 맵

DAIE : 이 비트가 설정이 되면, 상태 레지스터의 DA 비트가 '1'로 설정될때 POD 모듈에서 호스트로 인터럽트 발생한다.

FRIE : 이 비트가 설정되면, 상태 레지스터의 FR 비트가 '1'로 설정될때 POD 모듈에서 호스트로 인터럽트 발생한다.

RS(Reset) : 인터페이스를 재설정할 때 '1'로 설정한

다. 모듈 전체가 재설정되지는 않는다.

SR(Size Read) : 모듈에 최대 크기의 버퍼를 제공할 것을 요구할때, '1'로 설정한다. 데이터가 전송된 후 호스트는 '0'으로 재설정한다.

SW(Size Write) : 모듈에 사용할 버퍼 크기를 알릴 때 '1'로 설정한다. 데이터가 전송된 후 호스트는 '0'으로 재설정한다.

HC(Host Control) : 호스트는 데이터 쓰기 시퀀스를 시작하기 전에 '1'로 설정한다.

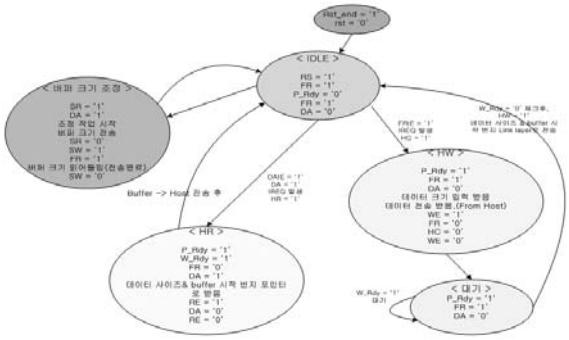

그림 6. 명령어 인터페이스의 동작에 따른 상태도

위의 그림 6 은 앞에서 설명된 명령어 인터페이스(Command Interface) 동작에 따른 상태도를 나타낸 것으로, 이에 따라 설계되었다.

4. 실험 결과

아래 그림 7 은 C++ 기반의 SystemC 로 작성된 코드를 실행한 Extended Channel 의 실행 결과이다.

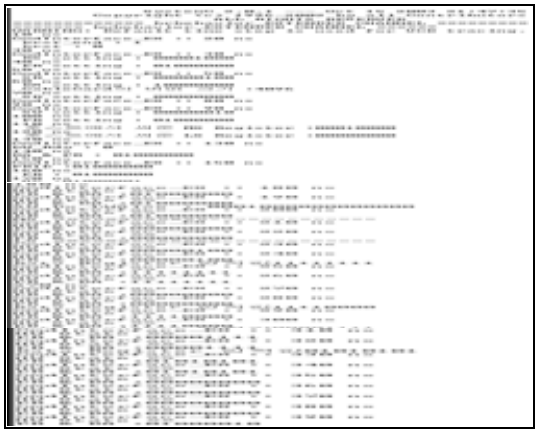

그림 7. Extended Channel 의 실행 결과

아래 그림 8 은 실행 결과를 Cadence 사의 SimVision 을 사용해서 설계한 Physical Layer 와 Link Layer 간의 Extended Channel 데이터 이동을 나타낸 시뮬레이션 결과이다. Data Channel 데이터의 시뮬레이션 결과도 데이터의 이동은 이와 마찬가지이다.

동작하는 순서는 PCMCIA 가 초기화를 마친 후, I/O Interface 로 전환하여 CableCard 로 동작한다는 신호를 받으면, 인터페이스의 초기화 및 IDLE 상태가 된다. 그 다음으로 호스트는 CableCard 가 가지고 있는 버퍼의 크기를 읽어서, 얼마만큼의 크기를 가지는가를 결정해서 CableCard 와 버퍼 크기를 조정하게 된다. 그리고 나서 Link Layer 로 전송 대기 상태가 되고, 설정이 되면 Link Layer 로 데이터가 전송되게 된다. Link Layer 에서 PCMCIA 를 거친 다음 호스트로 이동하는 데이터의 송신은 이와 반대가 된다.

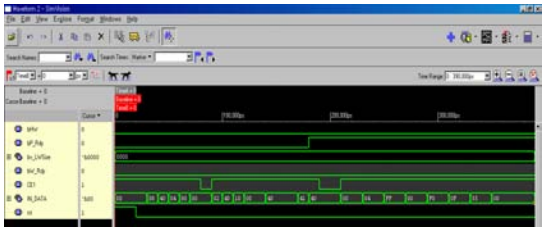

그림 8. CableCard 의 Physical Layer Simulation

5. 결론

본 논문에서는 디지털 케이블 TV 의 셋톱박스에 분리 삽입 가능한 CableCard 모듈의 PCMCIA Interface 와 Physical Layer 를 설계하였다.

우리 나라에서는 미국의 OpenCable 방식을 디지털 방송과 데이터 서비스의 표준으로 정했지만, 아직까지 연구가 활발히 진행되지 않고 있는 상황이다. 이런 시점에서 분리형 CableCard 의 개발은 국가적인 차원에서 원천 기술 개발을 위해서 연구되어야 할 것이다.

앞으로의 연구 진행은 고속 플랫폼 검증에 유용한 SystemC 를 이용하여 Transaction Level Modeling(TLM) 방법으로 CableCard 의 datapath 블럭의 상위 레벨 설계를 수행하고, 전체 통합 검증을 AHB 중에서 ARM7TDMI 를 이용하여 할 수 있도록 할 것이다.

6. Acknowledgement

본 논문은 IDEC 의 지원으로 수행되었슴.

참고문헌

[1] OC-SP-HOSTPOD-IF-113-030707, OpenCable HOST-POD Interface Specification.
[2] OC-SP-HOST-CFR-I12-030210, OpenCable Host Device Core Functional Requirements
[3] SCTE 28 2003, HOST-POD Interface Standard.
[4] PC Card Standard 7.0
[5] PCMCIA System Architecture 16-Bit PC Cards, second edition.
[6] EIA-679-B(Part B)
[7] 임기택, 최광호, 위정욱, 서정욱, 전자부품 연구원 "OpenCable 용 POD 모듈의 Out-of-Band Processor 개발", 대한전자공학회 하계종합학술대회 논문집 P101-104, 2001
[8] Michael Adams, Don Dulchinos, "OpenCable", Time Warner Cable, Cable Television Laboratories, IEEE Communications Magazine P98-105, 2001
[9] Thorsten Grotker, Stan Liao, Grant Martin, Stuart Swan, "System Design With SystemC", Kluwer Academic Publishers