

B-ISDN용 ATM AAL 계층의 구현 기술

손승일* 김형교**

◆ 목 차 ◆

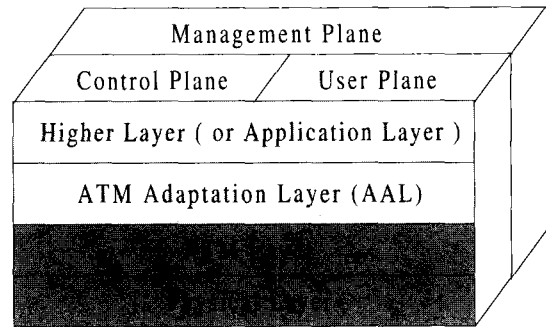
- | | |
|-----------------------|-----------------------|
| 1. 서론 | 4. 수신단 ATM AAL 계층의 설계 |
| 2. AAL 유형의 연구 | 5. AAL 모듈의 검증 |
| 3. 전송단 ATM AAL 계층의 설계 | 6. 결론 |

1. 서론

ATM(Asynchronous Transfer Mode)은 데이터, 비디오, 음성 등과 같은 다양한 트래픽을 전송하는 차세대 멀티미디어 모바일 통신 시스템인 IMT-2000에서 응용될 수 있을 것으로 예견되고 있는 분야이다[1][2][3]. 이는 높은 대역폭과 다양한 유형의 QoS의 지원이 가능하기 때문에 ATM의 응용은 점점 증가할 것으로 예견되고 있다. ATM 프로토콜 스택 모델은 그림 1처럼 네 개의 계층으로 나뉜다. 물리계층은 전송매체의 규격과 신호의 인코딩 규격을 만족시키는 역할을 수행한다.

ATM 계층에서는 상위계층에서 내려온 정보에 헤더를 첨가하는 일을 포함하여 셀 라우팅, 멀티플렉싱, 그리고 트래픽 제어 등의 일을 수행한다. ATM 적응계층(ATM Adaptation Layer-AAL)에서는 상위 계층의 애플리케이션에 맞도록 정보를 재정리하여 하위 계층으로 전송하는 역할을 담당한다. AAL 계층에서 제공하는 일반적인 서비스들로는 전송 에러의 처리, 큰 데이터 블록을 ATM 셀 크기에 맞도록 분할 및 재조합, 잃어버리거나 잘못 삽입된 셀 처리, 및 흐름제어와 타이밍 제어가 있다.

ATM 프로토콜 스택의 각 계층들과는 별도로 존재하는 플레인(Plane)은 여러 가지 관리 및 유지 역할을 담당하는데, 사용자 플레인은 사용자의 정보를 전송하는



(그림 1) ATM 프로토콜 스택

역할을 담당하고 흐름 제어와 에러 제어를 부가적으로 수행하며 제어 플레인(Control Plane)은 호(Call)의 제어와 연결 상태를 제어한다. 플레인 관리 플레인은 전체적인 시스템을 대상으로 플레인들 간의 상호 연결을, 그리고 계층 관리 플레인은 각 프로토콜 계층들 간의 자원 및 변수들과 관련된 기능을 수행한다.

본 논문에서는 AAL의 기본 구조를 알아보고, AAL 모듈을 설계한다. 설계되는 AAL 모듈은 VHDL 언어를 사용하여 기술되었으며, 합성 후 FPGA에 다운로드하여 기능을 검증하게 된다.

2. AAL 유형의 연구

현존하는 AAL 유형에 대한 간단히 소개하고자 한다. AAL 유형 1은 항등 비트율(CBR)을 지원하며, 송수신단의 클럭 동기를 필요로 하며, 송신단 및 수신단

* 한신대학교 정보통신학과 조교수

** 한신대학교 정보통신학과 부교수

간에 타이밍 정보 전송을 필요로 하며, AAL-1에서 복구되지 못한 정보 에러에 대한 표시 기능을 제공하고 있다[5][6]. 한편, 기존의 AAL 유형들이 저속의 짧은 가변 길이의 패킷을 사용하는 서비스에 비효율적인 면을 개선한 것이 AAL 유형 2이다[7]. 이 방식은 저속이면서 짧고 가변의 길이를 가지는 패킷들에 대해 지연에 민감한 특성을 지니는 응용에 대해 대역폭 활용의 효율성을 제고하고, 처리 지연을 감소시키기 위해 만들어 졌다. AAL 유형 3/4는 B-ISDN의 표준화 초기 단계에서 분리되어 추진되었던 연결형 데이터 서비스와 비연결형 데이터 서비스를 통합된 형태이며, SAR-PDU에 MID를 포함한 2 옥텟의 헤더와 2 옥텟의 트레일러를 매 옥텟마다 포함하고 있다[8]. 그러나 이 방식은 4 옥텟의 AAL-3/4 SAR-PDU 헤더 및 트레일러는 다중화 기능을 제공하기는 하나, 오버헤드를 부가하게 된다.

AAL 유형 5는 AAL-3/4의 다중화 기능을 제거하고, SAR-PDU의 전체 48 옥텟을 사용자 정보 전송에 사용할 수 있게 한 방식이다[9]. 오늘날 가장 많이 사용되는 방식이기도 하다. 그리고 AAL 유형 0이 있는데, 여기서는 AAL 유형 5의 형태를 사용하지만 트레일러의 CRC-32를 사용하지 않는다. 마지막으로 아주 작지만 각각의 패킷의 길이가 동일한 패킷들을 연속적으로 전송하는 응용 분야가 있다. 이는 ATM 상으로 MPEG 비디오 패킷을 전송하는 경우가 해당된다. ITU-T에서는 이러한 패킷의 전송을 위해 2개이 MPEG 프레임의 하나의 AAL 5 CPCS-PDU로 가공하여 전송하는 방안을 제안하였다[10]. 이러한 방식으로 전송할 경우 하나의 MPEG 프레임은 188 바이트이고, 2개의 MPEG 프레임 376 바이트가 하나의 CPCS-SDU를 구성하므로 여기에 8 바이트의 AAL 5 트레일러 정보를 부착하면 384 바이트(8 ATM 셀)로 별도의 패딩 정보를 부착하지 않으면서 8개의 ATM 셀을 전송할 수 있는 장점이 있다. 이러한 패킷의 전송은 AAL 유형 5를 사용하지만, 패킷의 길이가 작기 때문에 컨트롤 메모리의 정보를 갱신하기 위해 잦은 호스트 메모리의 액세스를 필요로 한다. 이러한 문제점을 해결하기 위해 본 연구에서는 AAL type 5 Mpeg 모드를 지원한다. 하나의 패킷 단위로 메모리를 관리하지 않고 다수개의 패킷의 길이 정보뿐만 아니라, 패킷의 수에 대한 정보를 컨트롤 메

(표 1) 본 논문에서 지원하는 AAL 유형

구 분	주요 구현 방안
AAL type 5	<ul style="list-style-type: none"> · Automatic CRC-32 Computation & Verification · Automatic Padding Information Generation · Automatic Endian Conversion(Little → Big) · Automatic PT field setting after Last Cell Detection
AAL type 5 Mpeg	<ul style="list-style-type: none"> · Implementation of All Function for AAL type 5 · Efficient Support of MPEG Video Over ATM · Minimization of Host System Access · 1 AAL 5 Packet of 2 MPEG Frame(376-byte : 8-cell)
AAL type 3/4	<ul style="list-style-type: none"> · Multiplexing Support by Host System · Automatic CRC-10 Computation & Verification · Automatic Endian Conversion(Little → Big)
AAL type 0	<ul style="list-style-type: none"> · Automatic Padding Information Generation · Automatic Last Cell Information insertion · Automatic Endian Conversion(Little → Big)

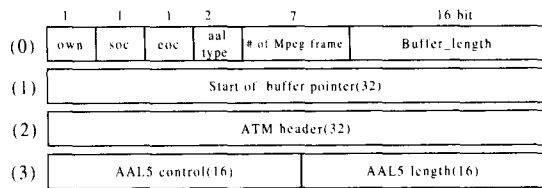
모리에 저장하여 지정된 패킷 수에 대한 전송을 완료 하였을 때 최종적으로 컨트롤 메모리를 갱신하도록 함 으로서 전체적인 성능을 향상시킬 수 있도록 하였다.

표 1은 본 연구에서 지원하는 AAL 유형을 요약하여 나타낸 것이다. 본 연구에서는 위에서 제시한 AAL 유형을 바탕으로 AAL 유형 0과 AAL 유형 3/4 및 AAL 유형 5를 지원하는 ATM 적층 계층에 대한 설계를 수행하며, 이를 VHDL 기술하여 검증한 후 FPGA에 다운로드하여 동작을 확인하였다.

3. 전송단 ATM AAL 계층의 설계

3.1 전송단의 컨트롤 메모리 구성

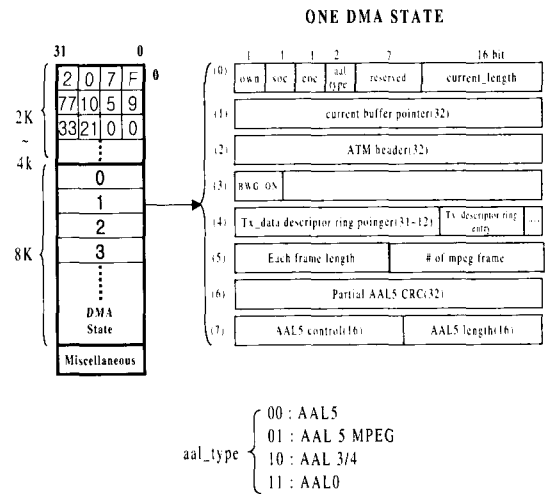
본 논문에서 정의한 호스트 메모리에 존재하는 전송 디스크립터 링의 구조는 그림 2와 같다. Own 비트는 디스크립터의 소유권을 정의하며 1로 세팅되어 있으면 AAL 모듈에서 소유권이 있고, 그렇지 않으면 호스트가 소유권을 갖는다. SOC 신호는 디스크립터 링의 시작 엔트리를 의미하며, EOC는 디스크립터 링의 마지막 엔트리를 의미한다. AAL type은 현재의 링이 어떤 유형의 AAL을 지원하는가를 의미한다. 본 설계에서는 AAL 유형 5와 AAL 유형 3/4 및 AAL 유형 0를



(그림 2) 호스트 메모리상의 Tx 디스크립터 링 구조

지원하고, 효율적인 MPEG 패킷 전송을 위한 AAL type 5 Mpeg 유형이 있다. 7 비트의 “# of Mpeg frame” 필드는 현재 MPEG 패킷을 전송하고 있다는 것을 의미한다. 이것은 MPEG 패킷의 길이가 188 바이트이고, 2개의 패킷을 합쳐 1개의 AAL 5 CPCS-PDU를 구성하기 때문에 성능향상에 유리하다.

그리고 설계된 AAL 모듈이 내부에서 ATM 셀을 처리하는 컨트롤 메모리의 구조를 본 연구에서는 그림 3과 같이 정의하였다. 하나의 패킷에 대해 패킷의 마지막까지 연속적으로 전송하는 경우도 있을 수 있지만, 일반적으로 각 채널에 할당된 QoS(Quality of Service)에 따라 다른 비트율로 전송되기 때문에 호스트가 설정한 각 채널에 대한 전송 상태 정보를 보유하고 있어야 한다. 이러한 전송 상태를 포함하고 있는 컨트롤 메모리의 영역을 DMA 상태 테이블이라 하며, 256 개의 채널을 지원할 수 있도록 하였다. 이러한 스케줄 테이블을 일반적으로 BWG 혹은 CRT(Constant Rate Table) 테이블이라 호칭하는데, 본 연구에서는 BWG 테이블이라는 용어를 사용하여 설명할 것이다. BWG 테이블은 현재 어떤 채널을 수행해야 하는지를 알려주는 정보를 워드의 각 바이트를 통해 채널 정보를 알려 준다. 한 바이트는 0부터 255까지 256 개의 채널을 표현할 수 있으므로, 본 연구에서 설계한 AAL 모듈은 256개의 전송 채널을 지원하고 있다. 만약에 더 많은 채널이 필요할 경우에는 채널을 표시하는 필드의 길이를 확장하면 가능하게 된다. AAL 모듈은 채널 번호의 정보를 사용하여 다음에 처리해야 할 DMA 상태 테이블의 정보를 읽어오게 되며, 테이블의 디스크립터 정보를 해독하여 채널의 활성화 및 비활성화 여부를 결정하게 된다. 그리고 채널이 현재 활성화되어 있으면, 호스트 메모리에서 48 바이트의 셀을 읽어와 AAL 모듈로 전달하게 된다. 또한 수행하는 AAL 동작 모드가 AAL 5 혹은 AAL 5 Mpeg 모드일 경우에



(그림 3) 컨트롤 메모리의 구조 및 1 채널의 DMA 상태 정보

는 이전 셀까지 계산되어 일시적으로 저장되어 있는 partial CRC-32 데이터를 AAL 모듈에 로드하여 다음에 유입되는 AAL 데이터에 대한 CRC-32를 계산하여야 한다. 또한 상태 테이블에 있는 16 비트 current_length 필드는 현재의 디스크립터 버퍼가 전송해야 하는 패킷 데이터의 양을 의미한다. 한 워드(32 비트)를 전송할 때마다 4씩 감소하게 되며, 이 값이 0이 될 경우에는 soc(Start of Chain) 혹은 eoc(End of Chain) 필드의 값을 조사하고 패킷의 마지막일 경우에는 채널의 own 비트를 disable 시키고, 그렇지 않을 경우에는 tx_data_descriptor 링 포인터를 사용하여 다음 번 버퍼를 로드하여 전송을 계속하게 된다.

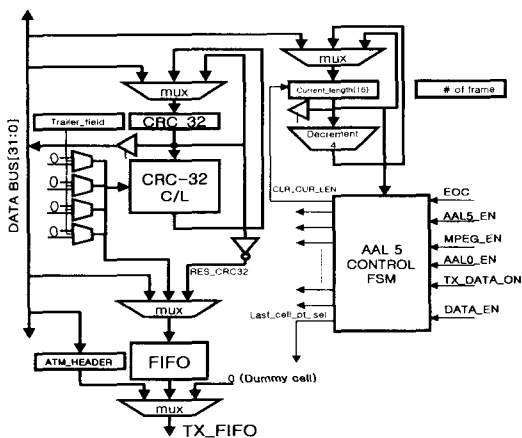
“# of Mpeg frame” 필드는 AAL type 5 mpeg 모드에서 사용하기 위한 필드이다. 이 모드에서 동작할 경우에는 다수의 프레임이 체인으로 연결하여 사용함으로써 새롭게 채널을 설정하지 않아도 되므로, 잦은 외부 메모리의 액세스를 방지할 수 있는 장점이 있다. 그리고 “each frame length”는 mpeg 모드로 동작할 때, 한 프레임의 바이트 수를 정의하기 위해 사용하는 필드이다. 또한 DMA 채널은 반드시 ATM 헤더 정보를 저장하고 있어야 한다. ATM 계층에 각 셀의 전송을 시작할 때 먼저 ATM 헤더 정보를 Tx FIFO에 저장한 후, AAL 모듈에서 처리한 48 바이트의 payload를 저장하게 된다. 그리고 AAL 5 관련 모드의 효율적인 지

원을 위해 CRC를 제외한 4 바이트의 트레일러 정보를 포함하고 있다. 이 정보는 AAL 5 control 필드에는 한 바이트의 CPI와 한 바이트의 UUI 정보를 합한 2 바이트의 정보를 포함하고 있으며, AAL 5 length 필드는 현재 채널을 통해 전송중인 CPCS-SDU의 길이를 나타내며 1부터 65,535의 값을 가질 수 있다.

마지막으로 2 비트의 AAL type 필드는 현재 활성화되어 있는 채널의 AAL 모드를 규정하고 있는 부분이다. "00"일 때는 AAL 5 모드를 의미하며, "01"일 때는 AAL 5 Mpeg 모드를 의미하며, "10"일 때는 AAL 3/4 모드로 동작되고 있음을 의미하며, "11"일 경우에는 AAL 0 모드로 동작되고 있음을 의미한다. 본 연구에서 구현하고 있는 BWG에서 '0'의 값을 가지고 있는 채널인 경우에는 VBR 채널 및 UBR 채널을 지원하기 위해 사용할 수 있다.

3.2 전송단의 AAL 유형 5 모듈의 설계

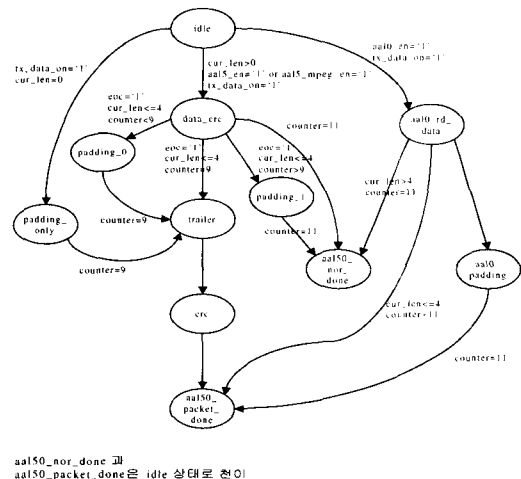
전송단의 AAL 5 관련 모듈에서는 3가지의 기능을 수행하게 된다. 첫 번째는 AAL 5의 순수한 기능과 AAL 5 mpeg 모드, 그리고 AAL 0 모드에 대한 설계가 여기에 속한다. 이에 대한 전체 블록도는 그림 4와 같다. 위에서 설명한 기본 구조를 바탕으로 전송단의 ATM AAL 모듈을 설계하였다. AAL 모듈은 각 셀 타임마다 하나의 셀을 처리하며, 디스크립터 링 생성시 설정된 AAL 유형에 따라 해당되는 AAL 유형을 처리하게 된다.



(그림 4) AAL 유형 5의 전체 블록다이어그램

먼저 AAL 유형 5의 경우에 기본적으로 CRC-32에 대한 계산을 수행하게 된다. 이러한 CRC-32의 계산의 ITU-T I.363.5의 규정을 준수하도록 설계하였다. 따라서 CRC-32의 초기값은 패킷의 전송을 시작할 때 항상 "FFFFFFF"값으로 설정하도록 하였다. 그런데 이 값은 Tx 단의 컨트롤 메모리의 Tx DMA 상태 레지스터에 호스트가 초기값을 설정하도록 하였다. 패킷의 전송을 시작할 때에 이 레지스터를 부분적으로 계산된 CRC-32 값을 저장하게 되며, 해당 채널을 액세스할 때 AAL 모듈로 읽어와 48-byte의 payload에 대해 CRC-32 값을 계산하고 이 때의 중간 결과를 다시 Tx DMA 상태에 저장하게 된다. 그리고 하나의 완전한 패킷에 대한 전송을 하는 마지막 셀을 전송할 때는 최종적으로 계산된 CRC-32를 반전한 값을 보내도록 ITU-T I.363에서 규정하고 있다. 따라서 본 연구에서는 패킷의 마지막을 전송할 때 부착하는 CRC-32 필드의 값은 트레일러의 length 필드까지의 CRC-32를 계산한 후 반전한 결과인 res_crc32 값을 전송하게 된다.

Current_length 필드의 경우에는 현재의 디스크립터가 제공하는 데이터 중에서 전송을 위해 남아있는 데이터의 바이트 수를 의미한다. 따라서 데이터 버스 폭이 32 비트(4 바이트)이므로 데이터 트랜잭션이 발생할 때마다 current_length는 4씩 감소하도록 하였다. 그런데, 만약 마지막 payload 워드를 수신한다고 할 때, 유효한 데이터가 1바이트에서 4바이트가 될 수 있다.



(그림 5) Tx AAL 5관련 모드의 상태 머신

마지막 payload가 4바이트일 경우에는 별도의 처리를 하지 않아도 상관없으나, 4 바이트 미만의 데이터를 수신할 경우에는 4 바이트에서 모자라는 만큼 데이터가 "00"인 패딩 바이트로 채워져야 한다. 이를 위해 CRC-32 C/L부 앞부분에서 멀티플렉서를 사용하여 지원하도록 설계하였다. 그리고 4씩 감소하도록 한 decrement는 마지막 payload에서 유효한 데이터만큼을 current_length에 빼서 결과가 0이 되도록 구현하였다.

정상적인 payload 데이터는 tx_aal_data[31:0] 버스를 통해서 전달받게 되며, 내부의 CRC_32 레지스터, current_length 레지스터 등은 tx_data_bus[31:0] 양방향 버스를 통해서 데이터 트랜잭션을 수행하도록 설계하였다.

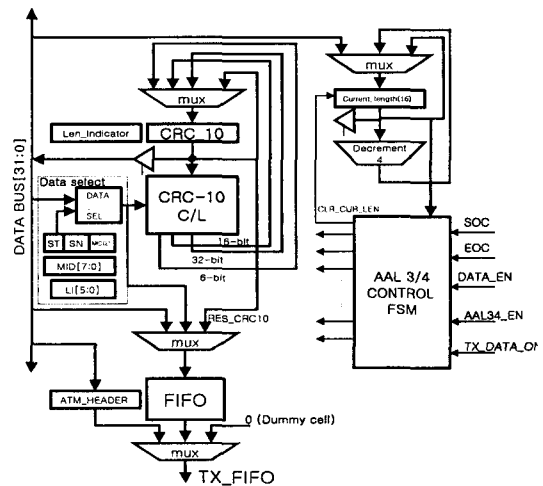
Tx aal 5 관련 모드 상태 머신은 그림 5와 같으며, tx aal 5 관련 전체 제어를 수행하는 부분이다. 따라서 이 블록은 AAL 5, AAL 5 mpeg 및 AAL 0 모드에 대한 제어를 수행하게 된다.

먼저 AAL 5 제어 상태 머신이 활성화되기 위해서는 호스트가 컨트롤 레지스터의 tx_en 신호를 '1'로 write 해야 한다. Tx_en 신호가 활성화되면 전송단은 패킷 전송을 위한 정상 모드가 되어 셀 타임(cell time) 간격으로 BWG 테이블을 참조하면서 해당 채널에 대한 ATM 셀을 전송하게 된다. 그리고, DMA 상태 테이블에 있는 첫 번째 워드 값을 디코딩하여 aal_en, mpeg_en 및 aal0_en 신호 중의 하나만이 활성화된 정보를 받아서 payload 전송 모드에 진입하게 된다. 그런데, 이러한 모드에 진입하기 위해서는 tx control 블록에서 tx_data_on 신호를 먼저 활성화시켜야 한다. 이 신호는 현재 선택된 채널이 활성화되어 있고, tx DMA 상태 테이블에서 데이터 로드를 완료하였으므로 AAL 5 상태 머신이 조만간 payload 데이터가 전송된다는 것을 알려준다. 이러한 조건을 만족할 때, data_en 신호가 활성화되면 한 워드의 payload 데이터가 도달하였으므로 이 데이터에 대해 해당되는 데이터 처리를 수행하도록 명령하게 된다. 이 때 CRC-32를 계산할 뿐만 아니라 자동적으로 endian conversion을 수행하게 된다. 왜냐하면 물리계층을 통해서 전송되는 데이터는 big endian 유형으로 전달될 뿐만 아니라 CRC-32의 계산도 big endian 유형으로 계산하기 때문이다. 그리고 EOC(End Of Chain) 신호가 활성화되어 있을 경우에는 현재 처리중인 패킷의 마지막 버퍼를 처리하고 있을

을 알려준다. 따라서 eoc가 세팅되어 있는 상황에서 current_length 필드가 zero가 되면 하나의 완전한 패킷의 전송이 완료되었기 때문에 제어권을 호스트에 넘기기 위해 own 비트를 '0'으로 리셋한다.

3.3 전송단의 AAL 유형 3/4 모듈의 설계

전송단의 AAL 유형 3/4는 AAL 유형 5와는 다르게 사용자 데이터 포맷이 멀티플렉싱을 지원하고 있다. 사용자 데이터 포맷은 항상 48 바이트의 배수가 되도록 호스트가 정렬하게 되며, 48 바이트의 처음 4 바이트를 ST(Segment Type), SN(Sequence Number) 및 MID (Multiplexing Identification) 정보로 채우고 나머지 10 비트의 CRC-10 계산 결과가 삽입될 자리를 확보하고 있다. 그림 6에 나타난 바와 같이 CRC-10의 계산을 위해 사용되는 입력은 32 비트만을 사용하지 않고 있다. 왜냐하면, CRC-10을 제외한 나머지 payload가 32 비트의 정수배가 되지 않기 때문이다. 그렇기 때문에 CRC-10에 유입되는 데이터중에서 payload의 마지막을 수행할 경우에 32 비트를 채우지 못하고 남은 데이터가 존재하게 되는데, 바로 이 데이터가 22 비트의 데이터이다. 이러한 22 비트의 데이터에 대한 CRC-10 계산을 완료해야 완전한 CRC-10 계산 결과가 되며, 계산의 입력에 사용된 22 비트와 CRC-10의 계산 결과 10 비트가 합쳐져 32 비트를 구성하며, 이것이 마지막



(그림 6) Tx AAL 유형 3/4의 전체 블록 다이어그램

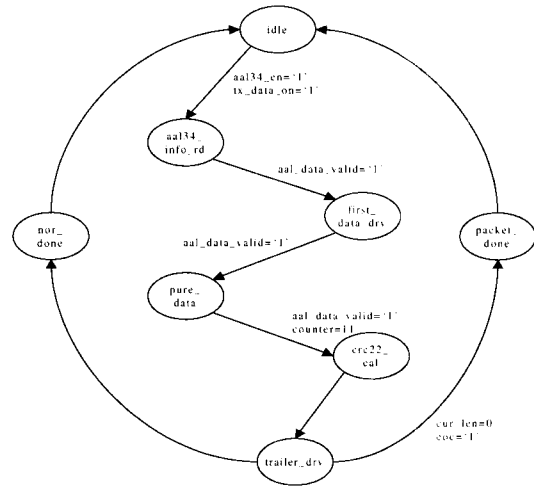
payload 데이터가 된다. 본 연구에서는 CRC-10을 계산하고 검증하는 로직을 입력 파라미터를 사용하여 구현하는 방안으로 설계하였으며, 입력되는 비트수가 차후에 64 비트로 확장되더라도 내부의 하드웨어에 대한 수정을 가하지 않고 입력 파라미터만을 변경함으로써 CRC-10을 계산할 수 있는 모듈을 개발하여 실제 설계에 이용하였다.

하나의 셀을 처리해야 하는 셀 타임이 되면, BWG 테이블에서 처리해야 하는 채널을 읽게 되고, 해당 채널이 활성화되어 있는 경우, 32 비트 tx_data_bus를 경유하여 tx DMA 레지스터로부터 필요한 필드의 내용을 로드하게 된다. 이러한 작업이 완료된 이후에 호스트로부터 48 바이트의 payload 데이터를 읽어오게 된다.

AAL 모드가 AAL 3/4일 경우에 본 블록에서는 읽어오는 payload 데이터에 대해 CRC-10을 비롯한 전반적인 데이터 조작이 시작된다. CRC-10의 경우에는 초기값을 항상 0으로 설정하도록 되어 있으며, 연속적으로 유입되는 셀들에 대한 CRC-10을 계산하는 것이 아니라 48 바이트 단위로 CRC-10 계산이 수행되어 payload의 마지막에 CRC-10 계산 결과를 전송하므로 임시저장 장소에 이 값을 저장하고 있을 필요가 없다.

Tx FIFO 윗 단에 있는 FIFO는 전송단 AAL 3/4 모듈에서 전송 ATM 셀 유형의 전송 순서에 맞도록 endian 변환된 payload 데이터를 저장하고 있다. Current_length 필드는 실제적으로는 AAL 5 모듈과 공동적으로 사용하는 필드이며, 여기서의 기능 또한 AAL 5에서의 동작과 동일한 방식으로 수행되게 된다. ATM 계층에 전송 가능한 완전한 payload가 FIFO에 담기면, AAL 3/4 모듈은 먼저 ATM 헤더 정보를 Tx FIFO에 구동하고 48 바이트의 payload를 전달하게 된다. Dummy 셀 (Idle 셀)을 전달하는 경우는 물리 계층의 전송 속도를 유지하기 위해 일정한 셀 타임동안 전송할 수 있는 채널에 데이터가 없을 경우에 더미 셀을 사용하여 ATM 계층에 데이터를 전달하기 위해 이용한다.

그림 7은 전송단의 AAL 3/4 모듈의 제어를 수행하기 위한 상태 머신을 나타내고 있다. 이 상태 머신이 동작하기 위해서는 현재 채널이 aal34_cn 신호를 활성화하고 있으며, payload 데이터가 입력된다는 것을 알려주는 정보 tx_data_on 신호가 활성화되어야 동작 모드에 진입하게 된다. 이 상태에서 최초의 유효한



(그림 7) 전송단의 AAL 3/4 모듈 제어 상태 머신

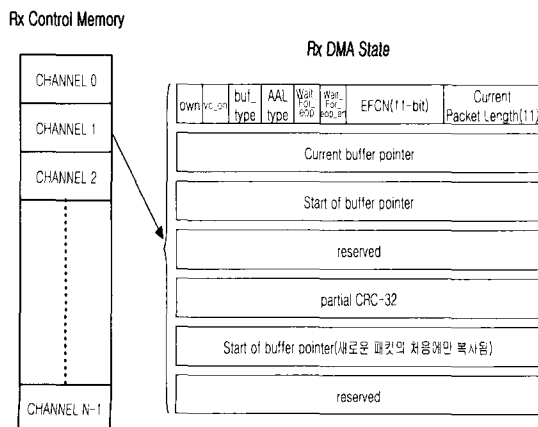
payload 데이터가 입력되고 있다는 것을 알려주는 aal_data_valid 신호가 활성화되면 4 바이트의 헤더 정보가 유입되고 있음을 의미하므로 이를 AAL 3/4 헤더 정보 저장 레지스터에 저장한다. 그 다음부터 aal_data_valid 신호가 유입되면 이전에 래치한 16 비트와 현재 유입되는 16 비트 데이터를 결합한 32 비트 입력에 대한 CRC-10을 계산함과 동시에 endian 변환을 수행하여 FIFO에 저장하게 된다. 그리고 현재 유입된 나머지 16 비트 데이터는 다음 데이터가 유입될 때 사용하기 위해 버퍼에 저장하는 역할도 수행한다. 이러한 방식으로 마지막 payload 데이터까지를 처리하고 첫 번째 유효 payload 데이터 사이클에서 래치한 6 비트의 LI (Length Indicator) 정보와 마지막 유효 payload 데이터 사이클에서 래치해 둔 16 비트 데이터를 결합한 22 비트의 데이터에 대해 CRC-10을 계산하게 되고, CRC-10 계산 결과 10비트와 22 비트 입력 데이터를 결합한 32 비트 데이터를 FIFO에 전송함으로써 완전한 48 바이트의 payload를 전달하게 된다.

또한 정상적인 패킷의 처리를 완료하였을 경우에는 nor_done 신호를 한 클럭 사이클 동안 활성화한 후에 초기 상태에서 대기하게 되며, 패킷의 마지막 payload를 처리하였을 경우에는 packet_done 신호를 활성화하여 payload 데이터에 대한 전송을 종료하게 된다.

4. 수신단 ATM AAL 계층의 설계

4.1 수신단 AAL 유형 5 관련모듈의 설계

수신단의 AAL 유형 5 관련 모듈의 설계는 전송단의 기능과 역으로 동작하게 된다. 우선 유틸피아 프로토콜을 통해 셀 단위로 ATM 셀을 수신하게 된다. 수신단 측에서는 먼저 수신된 셀에 대해 FIFO로부터 4 바이트의 ATM 헤더를 읽는다. 헤더 정보의 VPI 혹은 VCI 필드의 정보를 활용하여 RX DMA 상태 테이블을 참조한다. 그림 8과 같이 DMA 테이블의 첫 번째 워드에 존재하는 own 비트가 1로 세팅되어 있고, vc_on이 1로 세팅되어 있으면 현재 처리하는 ATM 셀이 활성화되어 있음을 의미한다. 그리고 2 비트의 Rx AAL type 필드가 있는데, "00"이면 AAL type 5의 셀을 처리함을 의미하고, "01"이면 AAL type 3/4의 셀을 처리하도록 채널이 설정되어 있음을 의미하며, "10"이면 AAL type 0 셀을 처리하도록 채널이 설정되어 있음을 의미한다. 그리고 wait_for_eop 신호는 현재 채널을 통해 수신되는 셀에서 패킷 사이즈 오류나 CRC 오류가 발견되면 패킷의 종료시까지 유입되는 나머지 패킷들을 폐기하도록 지시하는 신호로 작동한다. Partial CRC-32 레지스터는 AAL 5 유형의 패킷을 수신하는 과정에서 계산된 결과를 임시적으로 저장하기 위한 레지스터로 동작한다.



(그림 8) 수신단 컨트롤 메모리의 DMA 테이블 구조

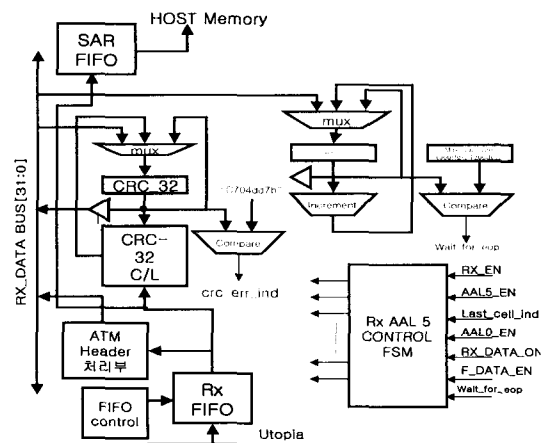
패킷의 마지막 셀이 도착하였을 때 CRC-32 필드를 포함하여 48 바이트 payload 전체에 대해 계산을 완료하였을 때 ITU-T I.363.5에서 규정한 나머지 값과 동일한 값인지를 비교하여 오류 유무를 판별하게 된다.

EFCN(Explicit Forward Congestion Notification) 필드는 ATM 헤더의 PT 필드의 3 비트중 가운데 비트가 1로 세팅되어 네트워크를 통해 전송되는 과정에서 트래픽 혼잡을 경험하였음을 의미하며, 그렇지 않으면 트래픽 혼잡을 경험하지 않았음을 의미하는데, 트래픽 혼잡을 경험한 셀의 수를 카운트하기 위해 사용되는 필드이다.

그리고 Current Packet Length 필드는 현재의 채널을 통해 수신되고 있는 패킷에 대한 수신한 셀의 개수를 저장하고 있는 필드이다. 새로운 셀이 유입될 때마다 1씩 증가하게 된다. 그리고 테이블에 나타난 포인터는 호스트 메모리의 버퍼 주소를 알려주는 레지스터이다. 실제로 OAM 셀은 채널 0에 할당하여 관리하도록 하였다.

그림 9는 Rx AAL type 5 유형에 대한 구현 블록도를 나타낸 것이다. 크게 Rx FIFO 부와 ATM 헤더 처리부, CRC 검증 부, Rx AAL 5 컨트롤 FSM 부 및 SAR FIFO 부 그리고 현재의 도달한 셀의 수를 계산하고, 최대 패킷 길이 위반을 체크하는 블록으로 구성되어 있다.

먼저 AAL 5 CRC 검증 블록에 대해 설명하면, 이 블록은 전송단의 CRC-32 블록과 유사한 기능을 수행하게 된다. 다른 점이 있다면 패킷의 마지막까지 입력된 데이터에 대해 CRC 값을 계산한다. 그리고 이때까지 계산된 CRC-32 값이 "C704DD7B" 값과 같은지를



(그림 9) Rx AAL type 5 유형의 구현 블록도

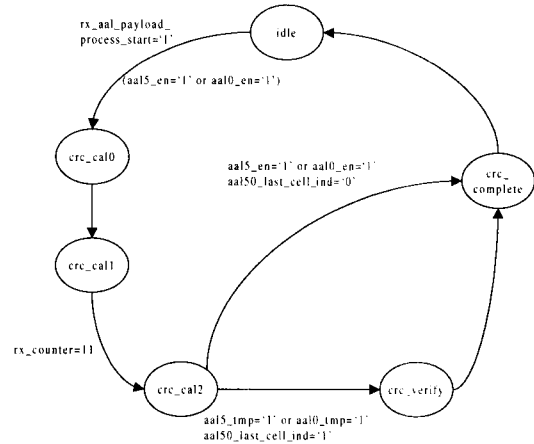
판별한다. 두 값이 같으면 패킷의 송신 중에 데이터 오류가 없었음을 의미하며, 그렇지 않으면 전송중에 패킷의 오류가 있었음을 의미한다. 판결 결과는 `crc_err_ind` 신호를 통해서 통보하게 된다.

`wait_for_eop` 신호를 발생하는 블록은 AAL type 5의 패킷이 최대 65,535 바이트를 넘지 않아야 한다는 규정을 수신되는 패킷 데이터에 대해 조사하게 된다. 만약에 이 규정을 위반할 경우에는 `wait_for_eop` 신호를 발생하여 Rx DMA 상태 테이블에 있는 해당 비트를 세팅하게 되고, 그렇게 되면 동일 채널에 대해 차후에 입력되는 셀 데이터에 대해 패킷의 마지막을 알리는 신호가 탐지할 때까지 수신한 셀을 폐기하게 된다.

Rx FIFO는 utopia 프로토콜[11]을 준수해 셀 단위로 수신한 데이터를 보관하고 있는 메모리이다. 새로운 셀이 수신되면, ATM 헤더 처리부가 헤더 정보를 해석하여 해당 채널의 DMA 정보를 로드한 후 48 바이트 payload 데이터에 대해 AAL type에 따라 적절한 AAL 기능을 수행한 후에 SAR FIFO에 데이터를 넘겨준다. 이 때 수신단에서는 동시에 endian 변환을 수행한다. 수신단은 big endian 데이터를 little endian 데이터로 변환하게 된다.

그리고 AAL 5 컨트롤 FSM 부는 AAL 5 관련 패킷에 대한 전체적인 제어를 관할한다. 이에 대한 내용은 그림 10에 나타나 있다. ATM 셀 처리부에서 내부 레지스터 로드 및 AAL 유형 판별 등을 완료한 후에 `tx_aal_payload_process_start` 신호를 발생한다. 이 신호를 받으면 CRC-32 계산 모드로 진입하게 된다. Rx FIFO로부터 payload 데이터를 읽어서 CRC-32를 계산함과 동시에 endian 변환을 수행하여 SAR FIFO에 저장하게 된다. 그리고 ATM 헤더 처리부에서 디코딩한 `aal50_last_cell_ind` 신호를 수신할 경우에는 payload의 마지막 데이터를 처리한 후 CRC-32 검증을 수행한다. 그렇지 않을 경우에는 해당 셀까지 계산된 CRC-32 값을 다시 Rx DMA에 저장함으로써 한 셀의 수신 과정을 종료하게 된다.

SAR FIFO에 저장된 데이터를 호스트 메모리로 DMA를 통해 전달되게 된다. 그리고 수신단은 호스트에 항상 48 바이트 단위로 호스트 메모리에 전달하도록 하였다. 패딩 정보 및 CRC-32 정보, CPI 및 UII 정보등에 대한 처리는 호스트가 패킷 수신을 완료한 후에 처리하게 된다.

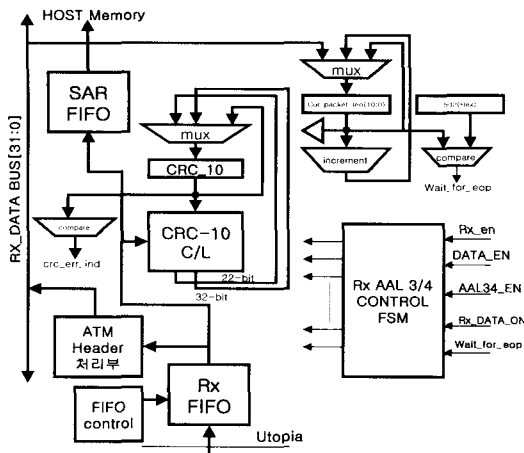


(그림 10) 수신단 AAL 5 및 AAL 0의 상태 머신

4.2 수신단 AAL 유형 3/4 모듈의 설계

Rx AAL 3/4에 대한 구현 블록도는 그림 11에 나타나 있다. 여기서 ATM 헤더 처리부 및 utopia 프로토콜로 구현한 Rx FIFO 블록 그리고 `cur_packet_len` 블록은 AAL 5 모듈의 설계에서와 동일한 방식으로 동작한다. 또한 CRC-10 검증 블록 또한 전송단의 CRC-10 계산 방식과 동일한 방식으로 동작한다. 그러나 수신단의 CRC-10 블록은 48 바이트의 payload에 대한 CRC-10 검증을 수행하게 된다. CRC-10 검증은 매 48 바이트 단위로 이루어지기 때문에 AAL 5와는 달리 중간 계산 결과를 저장할 필요가 없다. Payload 48 바이트에 대한 CRC-10을 계산한 최종 결과가 0이면 수신된 ATM 셀이 수신중에 오류를 발생하지 않았음을 나타낸다. 그리고 결과가 0이 아니면 수신중에 오류가 발생하였음을 나타낸다. 이러한 오류를 알려주는 신호는 `crc_err_ind` 신호이다. 이 신호가 활성화되면 `wait_for_eop` 비트를 설정하여 차후에 입력되는 동일 채널의 ATM 셀을 패킷의 마지막에 도달할 때까지 폐기시킬 수 있도록 설계하였다. 그리고 SAR FIFO에서 payload 데이터를 전달할 때, endian 변환을 수행한 결과를 저장하여 호스트 메모리에 저장되는 유형으로 변환한다.

AAL 3/4 모듈의 경우에는 특히 수신되는 payload에 대한 재조립을 수행하게 된다. AAL 3/4 SAR-PDU의 2 바이트 헤더 및 2 바이트 트레일러를 합쳐서 payload의 마지막에 전송해야 한다. 이러한 방식으로 데이터



(그림 11) 수신단 AAL 3/4 유형 구현 블록도

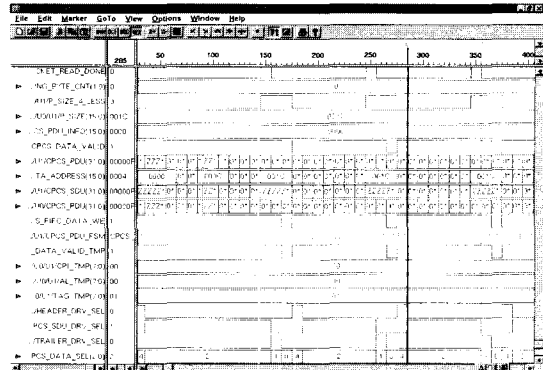
포맷을 변경하는 이유는 호스트가 AAL 3/4 패킷을 효율적으로 다룰 수 있도록 도와준다.

5. AAL 모듈의 검증

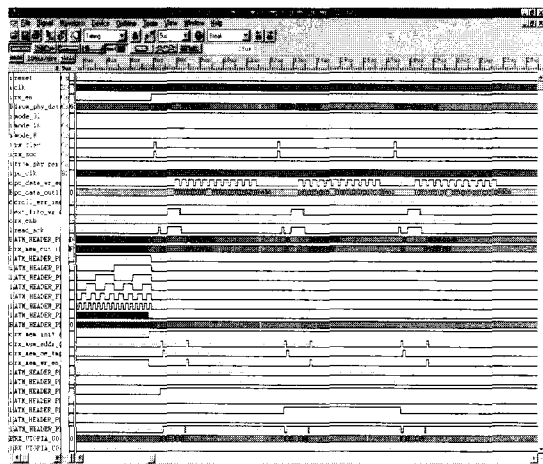
HDL을 사용하여 기술된 AAL 모듈에 대해 시뮬레이션 수행하였다. 본 논문에서는 전송단의 AAL 3/4에 대한 시뮬레이션 결과를 그림 12에 실었다. 시뮬레이션 수행 결과 ST, SN, 및 MID 필드 등을 SAR-PDU의 헤더에 부착하고, LI 및 CRC-10을 계산 및 생성하여 트레일러에 부착하여 ATM 헤더를 부착한 후 유토피아 프로토콜을 통해 물리 계층으로 전달되는 것을 확인하였다. 이 밖에도 AAL 5 및 AAL 0에 대해서도 다양한 시뮬레이션을 수행하였으며, CRC-32 계산 및 패딩 바이트 부착을 비롯한 트레일러 자동 부착 기능을 수행하는 것을 확인하였다. 그리고, AAL 0의 경우에는 CRC-32 계산을 제외하고는 AAL 5의 기능과 유사하다.

그림 13은 수신단의 탐 모듈에서의 시뮬레이션 파형을 보여준다. Rx 유토피아 프로토콜을 통해 유입되는 셀에 대해 AAL 유형을 해석하고 이에 따라 payload를 처리하는 과정이 그림에 나타나 있다.

그 밖의 모든 상황에 대한 시뮬레이션을 수행하였으며, 시뮬레이션을 통해 각 모듈에 대한 기능 검증은 완료하였다.



(그림 12) 전송단 AAL 유형 3/4의 시뮬레이션 파형

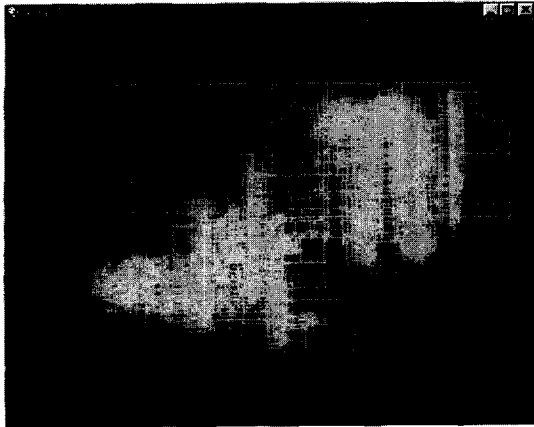


(그림 13) 수신단의 탐 모듈의 시뮬레이션 파형

(표 2) FPGA로 구현한 ATM AAL 모듈의 사양

구분	전송단(Tx)AAL	수신단(Rx)AAL
Target Library	vertexe(System Library V.1.37)	
FPGA Type	XCV1000E6H240C	
# of External I/O	240핀	
# of gates	80,064	66,568
Frequency	35MHz	50MHz

설계된 모듈은 자일링스의 FPGA XCV1000EHQ240 칩을 사용하여 검증하였다. FPGA로 구현한 ATM AAL 모듈에 대한 결과는 표 2에 나타났다. 그림 14는 본 논문에서 설계한 AAL 레이어 모듈에 대한 레이아웃 결과를 보여준다.



(그림 14) 설계된 ATM AAL 모듈의 레이아웃

송수신단을 합하여 사용된 전체 게이트 수는 약 15만개이며, 35MHz 이상에서 동작하는 것을 확인하였다.

6. 결 론

본 논문에서는 B-ISDN용 AAL 계층을 설계하였다. 설계된 모듈은 자일링스의 FPGA XCV1000EHQ240 칩에 다운로드 하여 기능을 검증하였다. 설계된 모듈의 게이트 수는 전송단은 8만 게이트이며, 수신단은 6만6천 게이트였고, 동작 주파수의 경우 전송단은 35MHz, 수신단은 50MHz였다. 설계한 모듈들은 ATM, Ethernet 및 SONET 등과 같은 네트워크 및 통신 분야에서 효율적으로 이용될 수 있을 것으로 예견된다.

설계한 AAL 모듈은 622Mbps 이상의 고속의 ATM 셀 전송을 지원하기 위해 설계되었으며, 본 연구에서 제공한 ATM AAL 모듈을 통해서 응용 분야에 적합하도록 주변 회로를 첨가하여 고속의 ATM NIC(Network Interface Card) 카드를 비롯한 각종 관련 분야에서 효과적으로 사용될 수 있을 것으로 예견된다.

Acknowledgement

- [1] 본 연구는 과학기술부와 산업자원부의 선행 핵심 IP 개발 사업의 일환으로 연구되었음.
- [2] 본 연구의 일부는 IDEC의 CAD 툴에 지원에 의해 이루어 졌음.

참 고 문 헌

- [1] Rainer Handel, Manfred N Huber, and Stefan Schroder, ATM Networks : Concepts, Protocols, and Applications, Addison-Wisley, 1998
- [2] Othmar Kyas, ATM Networks, Addison, 1994
- [3] Masahide Hatanaka, Toshihiro Masaki, Takao Onoye, and Koso Murakami, "VLSI Architecture of Switching Control for AAL Type 2 Switch", IEICE Trans. Fundamentals, Vol.E83-A, No.3, pp435-441, Mar. 2000.
- [4] 김영탁, 김종근, 조유제, 데이터통신 및 컴퓨터망, 정익사, 1999
- [5] ITU-T Recommendation I.363, B-ISDN AAL Specification, Mar. 1993.
- [6] ITU-T Recommendation I.363.1, B-ISDN AAL Specification : Type 1 AAL, Aug, 1996
- [7] ITU-T Recommendation I.363.2, B-ISDN AAL Specification : Type 2 AAL, Sep, 1997
- [8] ITU-T Recommendation I.363.2, B-ISDN AAL Specification : Type 3/4 AAL, Sep, 1997
- [9] ITU-T Recommendation I.363.5, B-ISDN AAL Specification : Type 5 AAL, Aug, 1996
- [10] The ATM Forum, Audiovisual Multimedia Services : Video on Demand Specification 1.1, Mar. 1997
- [11] The ATM Forum, UTOPIA 3 Physical Layer Interface, Nov., 1999
- [12] 차태호, 전춘기, 비주얼 C++에 의한 인터페이스 응용, 북두출판사, 2001

● 저자 소개 ●



손승일

1985년~1989년 : 연세대학교 전자공학과(공학사)
1989년~1991년 : 연세대학교 전자공학과(공학석사)
1992년~1998년 : 연세대학교 전자공학과(공학박사)
1998년~2002년 : 호남대학교 컴퓨터공학과 조교수
2002년~현재 : 한신대학교 정보통신학과 조교수
관심 분야 : ATM 통신 및 보안, ASIC 설계, 컴퓨터구조



김형교

1978년 : 서울대학교 전기공학과(공학사)
1980년 : 서울대학교 전자공학과(공학석사)
1993년 : Georgia Institute of Technology School of Electrical Eng.(Ph.D.)
1983년~1985년 : 인덕 대학 전자과 교수
1993년~1995년 : 한국전자통신연구원 선임연구원
1995년~1997년 : 상명대학교 정보과학과 교수
1997년~현재 : 한신대학교 정보통신학과 부교수
관심분야 : DSP, VLSI Signal Processing, 무선인터넷 응용