

ARM 기반의 네트워크용 SoC(System-on-a-chip)

프로세서의 설계 및 구현

박경철^o 나종화
아주대학교 한세대학교
kcpark@waytotec.com^o, jwna@hansei.ac.kr

Design and Implementation of ARM based Network SoC Processor

Kyoungcheol Park^o Jongwhoa Na
Dept. of Systems Engineering, Ajou University
Dept. of Computer & telecommunication Engineering, Hansei University

요약

본 논문에서는 서로 다른 네트워크간의 다양한 프로토콜과 이종의 트래픽을 동시에 처리할 수 있는 네트워크용 SoC (System-on-a-Chip) 프로세서를 구현하였다. 제작된 네트워크 SoC 프로세서는 ARM 프로세서 코어와 ATM(Asynchronous Transfer Mode) 블록, 10/100 Mbps 이더넷 블록, 스케줄러, UART 등을 이용하였고 각 블록은 AMBA (Advanced Microcontroller Bus Architecture) 버스로 연결하였다. SoC 프로세서는 CADENCE사의 VerilogHDL을 이용하여 설계하였고 0.35 μ m 셀 라이브러리를 이용하여 검증하였다. 구현된 칩은 총 게이트수가 312,000개이며 칩의 최대 동작 주파수는 50MHz 이다.

1. 서론

다양한 종류의 네트워크들은 고유의 특성을 갖고 각기 발전해 왔다. 하지만 이런 네트워크들의 개발과 함께 이들의 통합에 대한 요구가 증가되어왔다. 현재까지 LAN의 프로토콜로는 이더넷이 WAN용 프로토콜로는 ATM이 가장 많이 사용되고 있다. 따라서 이들 간의 프로토콜 변환이 많이 발생하고 있는 상황이며 이 문제는 주로 소프트웨어에 의해 처리되어 왔으나 급속하게 발전하는 네트워크의 속도를 고려할 때 성능이 문제가 된다. 더욱이 음성, 영상, 데이터의 다양한 트래픽을 신속하게 처리해야 하는 경우에는 성능이 사업자가 제공할 수 있는 서비스의 종류를 결정하기도 한다.

본 논문에서는 이러한 다양한 프로토콜과 이종의 트래픽을 동시에 처리할 수 있는 네트워크 SoC 프로세서의 구현을 설명한다. 제안된 SoC는 ADSL 기반의 네트워크 환경과의 접속을 하나의 칩으로 구현하여 음성과 데이터를 한 칩에서 처리하여 비용은 물론 성능 면에서도 우월한 네트워크 장비의 개발을 가능하게 하였다.

개발된 네트워크 SoC는 프로세서, 이더넷 블록, ATM 블록, 그리고 스케줄러로 구성된다. 프로세서는 ARM720T를 사용하여 다양한 운영체제, 소프트웨어 등을 활용하기가 용이하다. 다양한 프로토콜들을 처리하기위해 이더넷 블록, ATM 프로토콜 처리 블록을 내장하여 고속의 프로토콜 변환이 가능하다. 스케줄러는 사용자별, 사용 어플리케이션의 트래픽 별로 데이터 전송 속도(data rate)를 관리하는 기능을 수행한다. 이는 인터넷 폰과 같은 데이터 망을 이용한 음성 통화 시 발생하는 통화 품질의 열화를 방지할 수 있다. 또한 사용자의 전송하고자 하는 데이터가 없는 경우 이를 다른 사용자가 사용할 수 있도록 하는 기능을 포함하여 전송 대역폭을 최대한 활용할 수 있도록 하였다.

네트워크 SoC 칩은 SUN사의 Ultra 60 워크스테이션의 solaris 2.7 환경에서 CADENCE사의 verilog HDL을 사용하여 설계하였다. 합성은 SYNOPSIS사의 Design CompilerTM 블록

이용하였다. 최종적으로 검증된 회로는 하이닉스의 0.35 μ m 공정을 이용하여 제작되었다. 구현된 칩은 총 게이트수가 312,000개이며 칩의 최대 동작 주파수는 50MHz 이다.

2. 네트워크 프로세서의 구조와 기능

그림 1은 설계된 네트워크 SoC 프로세서의 전체 시스템 구성도를 나타낸다. 네트워크 SoC 프로세서는 ARM 720T 프로세서 코어, ATM 처리블록, Ethernet MAC 등으로 구성된다.

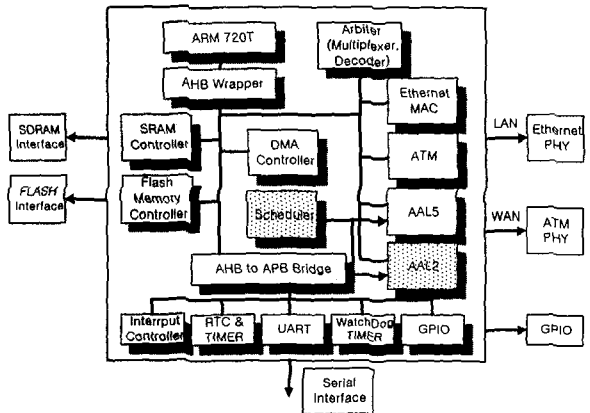


그림 1. 네트워크 프로세서의 구조

프로세서 코어는 ARM720T로서 내부 8KB cache, write buffer와 MMU(Memory Management Unit)를 보유하여 임베디드 Linux 또는 Pocket PC 등의 임베디드 OS의 포팅이 가능하다. 기타 UART, DMA 제어장치, Watch dog 타이머, 인터럽트 제

어장치, 타이머 제어장치 등은 개개의 기능블록으로 분할하여 설계 되었으며 개별적으로 시뮬레이션을 수행하였다. 내부 기능 블록들의 연결을 위한 시스템 버스는 ARM의 AMBA 버스를 채택하였다. AMBA는 고성능 내장 프로세서의 내부 데이터 전송 표준으로서 AHB (Advanced High-performance Bus) 와 APB(Advanced Peripheral Bus)로 구성된다. 이제 각 블록을 상세히 설명한다.

2.1 ATM 처리블록

광역망(WAN)용 프로토콜로 많이 사용되는 ATM 처리부의 세부 구성모듈은 다음과 같다. ATM 처리 모듈은 상위 계층과의 데이터 전송을 ATM 프로토콜에 맞게 구현해서 53 바이트의 셀로 주고받도록 하는 모듈로서 다음과 같은 여러 가지 모듈들로 구성되어 있다. 음성 traffic을 담당하는 AAL(ATM Adaptation Layer)2 모듈과 고속의 데이터 트래픽을 처리하는 AAL5 모듈, 그리고 ATM layer의 기능을 수행하는 ATM 모듈, 이를 물리계층으로 보내주는 UTOPIA 인터페이스 모듈, 호스트 인터페이스 모듈등으로 구성되어 있다.

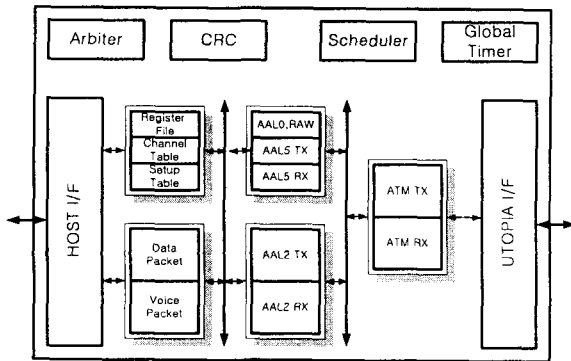


그림 2 . ATM 처리블록의 구성도

AAL5와 AAL2모듈은 각각 송신부와 수신부로 나누어서 구현하였다. 송신부의 경우는 셀 슬롯 단위로 동작하며, 이는 스케줄러 모듈에서 발생시켜 준다. 즉 스케줄링된 셀 레이트(cell rate)에 따라 그에 해당하는 시간 단위로 동작하도록 슬롯을 보내준다. 한 개의 셀 슬롯 이내에 하나의 cell(53bytes)를 전달하는 것이다. 그러나, DMA를 사용하는 경우 DMA의 완료 시점을 정확히 이 가정에 맞출 수 없으므로 이를 완화하기 위한 버퍼를 두고 있다. AAL0 module은 AAL type0 로서 특별한 AAL처리를 하지 않고 upper layer 로부터 전달받은 데이터를 ATM cell payload size 로 나누어 ATM layer 로 전달하며, 수신시에도 ATM layer로부터 전달받은 ATM cell payload 를 upper layer 로 전달하는 기능을 수행한다.

2.2 Ethernet 10/100Mbps MAC

그림3은 Ethernet 블록의 구성도이다. Ethernet MAC은 수신부, 송신부, CRC (Cyclic Redundancy Code) generation / checker, 주소 식별 로직, FIFO와 FIFO 제어 로직, DMA 제어 로직으로 구성되어 있다.

MAC client가 frame의 전송을 요구하였을 때, CSMA/CD MAC sub-layer의 Transmit Data Encapsulation component는 client가 제공한 data로부터 frame을 만든다. Frame의 시작 부분에 preamble과 Start Frame Delimiter를 붙인다. Client가 제공한 data에 따라 minimum frame size를 충족시키기 위해 PAD data를 덧붙일 수도 있다.

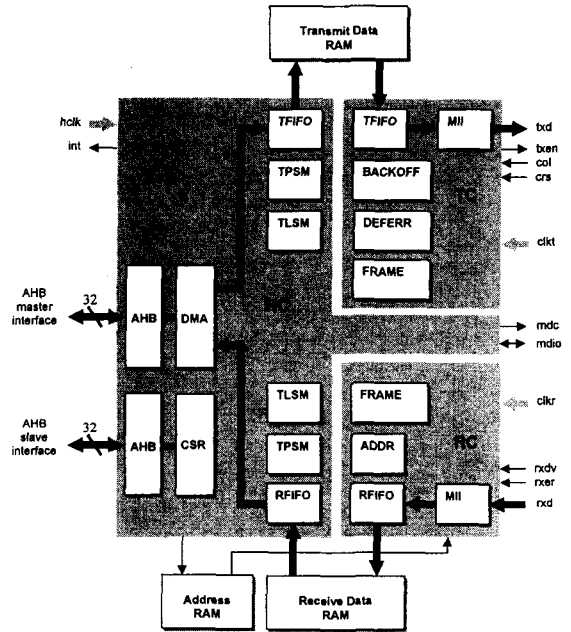


그림 3. Ethernet 처리 블록의 구성도

또한 destination address, source address, length/type field와 error detection을 위해서 frame check sequence field도 붙인다. MAC sub-layer의 Transmit Media Access Management component에 의해 frame의 전송은 이루어진다.

2.3 Peripheral 모듈

네트워크 SoC 프로세서는 외부 메모리용으로 ROM, FLASH, SDRAM Memory 인터페이스를 지원한다. ROM은 시스템 초기화를 구현하는 Bootloader용으로, Flash 메모리는 Bootloader를 제외한 시스템 S/W를 저장한다. SDRAM은 네트워크 SoC 프로세서가 수행하는 응용 S/W 또는 데이터 수행 중에 발생하는 데이터 부분이 저장된다. ROM, Flash Interface는 Static Memory Controller가 그 기능을 수행하고, SDRAM interface는 SDRAM Controller가 동작을 수행한다.

Static Memory Controller는 non-clocked 메모리 서브시스템의 asynchronous 페이지 모드 read 동작과 burst mode ROM device의 asynchronous burst mode read access 기능을 지원한다. SMI(Static Memory Interface)는 8, 16 과 32 비트 폭의 외부 메모리 데이터 path를 가질 수 있고, little-endian과 big-endian을 지원한다. 또한, 8개 메모리 bank까지 가질 수 있고, 32개의 programmable wait states를 지원한다.

SDRAM Memory Controller는 4 개의 독립적인 chip select를 제공하고, controller와 SDRAM 사이의 quad-word 전송을 지원한다. 또한, 1, 2, 3 CAS latency와 SDRAM의 CKEOut[3:0]과 CLKOut을 동적으로 제어하는 파워 saving 모드를 지원한다. 위에서 설명한 SMI와 SDRAM Controller는 모두 AMBA 2.0 버스 규격의 AHB Slave로 구성되도록 설계되었다.

3. 시뮬레이션

설계된 네트워크 프로세서의 모든 동작에 대한 기능 검증은 실제 ARM 명령어를 사용하여 실제와 같은 상황으로 프로그래

밍한 후 Verilog HDL 모델을 이용하여 이상적인 기능 검증은 수행하도록 시뮬레이션을 수행하였다. 시뮬레이션은 먼저 각각의 모듈별로 수행되고 블록별로 인터페이스를 검증하여 전체 동작을 확인 하는 Bottom-up 방식을 사용하였다. ATM과 Ethernet 프로토콜 처리블록과 ARM 코어가 정상적인 동작을 수행하는지 확인하고 이를 시스템 내에 집적 하였을 때 원하는 출력을 내는지를 검증하였다. 다음은 부분별 시뮬레이션 결과이다.

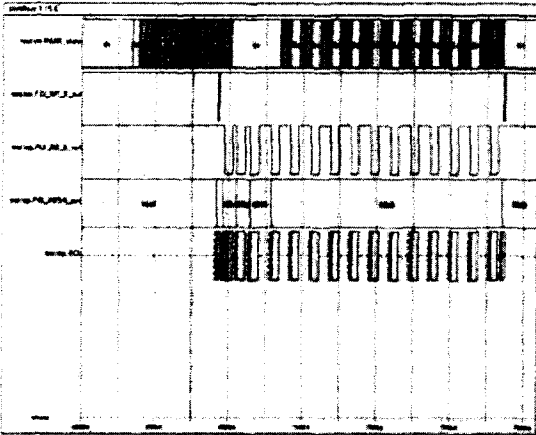


그림 4 ATM 셀 수신과정



그림 5 물리계층 칩과의 접속과정

시뮬레이션 결과와 논리합성후 게이트 지연 시간등과 같은 실제회로 특성을 고려하여 전체 시스템이 정확히 동작함을 확인하였다. 그림 6은 네트워크 프로세서의 논리합성 결과이다.

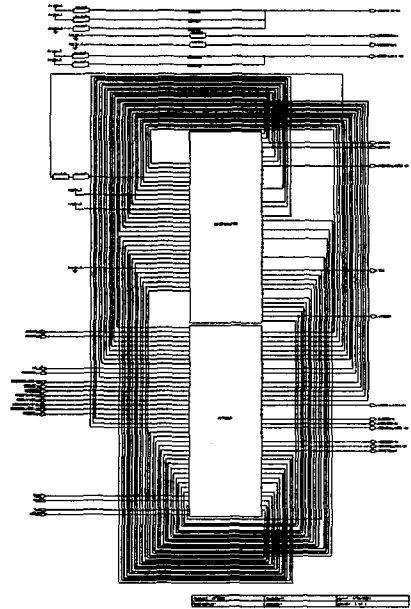


그림 6 ARM720T Synthesis 결과

4. 결론

본 논문에서는 ADSL/VDSL 및 음성 데이터 통합 장비인 IAD(Integrated Access Device)에 사용 되는 네트워크 프로세서의 구조 및 설계에 관하여 설명하였다. 설계된 프로세서는 50MHz의 시스템 클럭에서 ATM 데이터는 52Mbps까지 이더넷은 최대 100Mbps까지 처리할 수 있다. Verilog HDL로 설계된 후 하이닉스의 0.35um 공정을 이용하여 제작 되었고 게이트수는 총 312,000개 이다. 패키지는 QFP 208 pin으로 되어 있다. 네트워크 프로세서는 급격한 통신의 초고속화 및 데이터, 영상 음성의 통합지원 경향에 따라 그 수요가 폭발적으로 증가하고 있으며 앞으로도 많은 관심과 연구가 필요한 분야라 생각된다.

5. 참고 논문

- 1) Steve Furber, " ARM System Architecture", Addison-Wesley,1996.
- 2) "ARM7TDMI Data sheet", Advanced RISC Machines Ltd (ARM),1995.
- 3) Dave Jagger, "ARM Architectural Reference Manual", Prentice Hall, London, 1996.
- 4) "Programming Techniques (ARM DUI 0021A)", Advanced RISC Machines Ltd (ARM), 1996.
- 5) "The ARM7TDMI Debug Architecture", Advanced RISC Machines Ltd (ARM), 1996.
- 6) "IEEE Standard 1149.1 Test Access Port and Boundary-Scan Architecture", IEEE, 1990.
- 7) Lawrence T. Clark, " An Embedded 32-bit Microprocessor Core for Low-power and High-Performance Applications", IEEE Journal of solid-state circuits, Vol. 36, No.11, November 2001.