

# TCP/IP 프로토콜 스택 프로세서 IP의 VLSI 설계

최병윤, 박성일, 하창수  
동의대학교 컴퓨터공학과

## VLSI Design of Processor IP for TCP/IP Protocol Stack

Byeong-Yoon Choi, Seung-II Park, and Chang-Su Ha  
Department of Computer Engineering Dong-Eui University  
E-mail : bychoi@dongeui.ac.kr

### Abstract

In this paper, a design of processor IP for TCP/IP protocol stack is described. The processor consists of input and output buffer memory with dual bank structure, 32-bit RISC microprocessor core, DMA unit with on-the-fly checksum capability. To handle the various modes of TCP/IP protocol, hardware and software co-design approach is used rather than the conventional state machine based design. To eliminate delay time due to the data transfer and checksum operation, DAM module which can execute the checksum operation on-the-fly along with data transfer operation is adopted. By programming the on-chip code ROM of RISC processor differently, the designed stack processor can support the packet format conversion operations required in the various TCP/IP protocols.

### I. 서론

최근의 급속한 네트워크와 인터넷 기술의 발전으로 전세계 컴퓨터가 하나의 망에 연결되는 시대가 되고 있다. 이러한 시대에 맞추어 OC-768과 같은 고속의 네트워크 기술은 40 Gbps 이상의 전송율을 제공하고 있다.

그러나 이와 동시에 접속 노드의 증대로 네트워크 트래픽은 계속 증가하고 있다<sup>[1]</sup>. 기존 연구 결과에 따르면 인터넷을 통해 전달되는 패킷의 85% 이상이 TCP/IP (Transmission Control Protocol / Internet Protocol)에 기반을 두고 있다<sup>[2]</sup>. 그러나 TCP 프로토콜은 신뢰성을 갖는 전송 능력을 제공하는 대신에 복잡한 상태도에 바탕을 두고 있기 때문에 프로토콜의 전송 속도를 감소시키는 주원이 되고 있다. 이러한 이유로 TCP 프로토콜 대신할 수 있는 여러 가지 단순화된 (light-weight) 프로토콜이 연구되었다<sup>[6]</sup>. 그러나 인터넷의 활성화는 TCP/IP 프로토콜을 사실상의 (De-Facto) 표준으로 만드는 결과를 낳았다. 따라서 TCP/IP 프로토콜을 대체하기 보다는 TCP/IP 자체를 고속으로 처리하기 위한 연구가 필요하게 되었다. 국내의 경우 부산 대학교에서 90년대부터 TCP/IP 프로토콜을 하드웨어로 구현하는 연구를 하여 많은 연구 성과를 발표하고 있다<sup>[5]</sup>. 그 외 국외에서는 워싱턴 대학(Washington University)의 LockWood 교수 연구실에서 TCP Packet를 모니터링하는 TCP Splitter라는 회로를 개발하였다<sup>[3]</sup>. TCP 프로토콜에 비해 IP 프로토콜은 상대적으로 단순하여 구현이 용이하므로, 상대적으로 훨씬 복잡한 TCP 프로토콜 계층의 고속화에 대한 연구가 필요하다. 참고 문헌[4]에서 수행한 연구는 TCP/IP 프로토콜의 모든 기능을 하드웨어로 구현하려는 연구로, 규모는 방대했지만 하드웨어 융통성 문제로 TCP/IP의 다양한 선택사양(Option)을 지원하지 못하고 있으며, 또한 다중화(Multiplexing)와 다중 연결

등을 구현 할 수 없는 제약사항이 있다. 참고 문헌[3]의 TCP/IP 프로토콜 분석에 따르면 TCP 프로토콜의 경우 모든 기능을 하드웨어로 구현하는 것이 비현실적임을 알 수 있다. 그 이유는 다음과 같다.

첫째, TCP 프로토콜의 경우 패킷 당 하나의 타이머와 상태도가 유지되어야 하므로 하드웨어로 모든 타이머와 상태 정보를 구현하는 것은 거의 불가능하다.

둘째, ATM과는 달리 데이터 필드의 크기가 가변적이므로 reassembly buffer를 위해 많은 메모리가 필요하다.

셋째 많은 연결 관계(connection)를 하드웨어로 관리하는 것이 복잡한 상태 와 메모리 문제로 불가능하다. 따라서 TCP 프로토콜의 경우 OS 관련 동작, 상태 관리, 타이머 관리, 연결 관리, Sliding Window 버퍼 제어 등은 소프트웨어에 의해 처리하고, 순수한 데이터 전송 동작과 체크섬 계산 / 검사 동작은 하드웨어로 처리하는 Hardware-Software Co-Design 전략이 바람직하다. 더구나 참고 문헌[5]의 TCP 프로토콜 동작 분석에 따르면, TCP 프로토콜의 대부분 시간은 데이터 전송과 Checksum 계산에 의해 생기는 지연임을 알 수 있다. 그 외에 PIC 마이크로컨트롤러에 TCP/IP의 제한된 기능을 소프트웨어로 구현하여, RS232 직렬 전송 또는 Ethernet Controller를 통한 연결을 통해 WEB Server를 구현하거나, Network-Attacked Device를 구현하는 연구가 활성화되고 있다<sup>[7]</sup>.

이러한 기존 연구의 분석 결과를 바탕으로 본 연구에서는 TCP/IP 프로토콜을 하드웨어와 소프트웨어로 나누어 구현하는 방식으로, 다양한 프로토콜과 다수의 선택사항을 고려하여 RISC 마이크로프로세서를 기반으로 하는 TCP/IP 프로토콜 스택 프로세서를 설계하였다.

## II. TCP/IP 프로토콜

1982년 ARPANet을 이용하기 위한 통신용 프로토콜로 개발된 TCP/IP 프로토콜(TCP/IP Protocol Suite) 구조는 기본적인 송수신의 역할을 담당하는 TCP/IP 프로토콜과 그 외에 TCP/IP 프로토콜의 기능을 보충하여 주는 프로토콜들로 구성되어 있다. 그림 1은 현재 사용되고 있는 TCP/IP 프로토콜과 주변 프로토콜들과, OSI 7 계층과의 관계를 보여주고 있다.

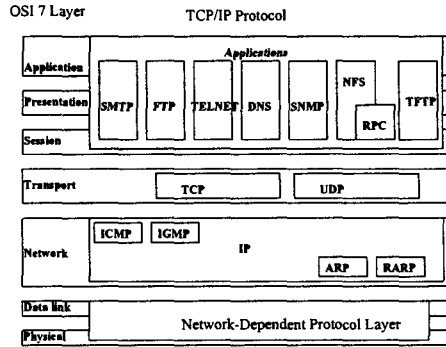


그림 1. OSI 7 계층과 TCP/IP 프로토콜 계층과의 관계

## III. TCP/IP 프로토콜 스택 프로세서

### 3.1 구현 범위 및 설계 사양

TCP/IP 프로토콜의 성능을 저하시키는 요인을 밝히기 위한 참고 문헌[5]의 연구 결과는 표 1과 같다. 20 Mhz의 68020 CPU를 사용하는 SUN-3/60 워크스테이션에서 Berkeley TCP를 수행시켜 TCP/IP 프로토콜의 오버헤드를 측정된 결과이다.

표 1. TCP/IP 프로토콜의 오버헤드

구분	처리 시간	단위
User-System Copy	200us	Byte
TCP Checksum	185us	
Network-Memory Copy	385386us	
Ethernet Drivers	100us	Packet
TCP+IP+ARP Protocols	100us	
OS Overhead	240us	

본 논문에서는 표 1의 분석 데이터를 바탕으로 설계한 TCP/IP 프로토콜 스택 프로세서 코어는, 다음과 같은 특징을 갖고 있다. 전체 TCP/IP 프로토콜을 구현하는 대신에 NOC (Network-on-Chip) 환경에서 TCP/IP 프로토콜 스택 동작에서 가장 많은 연산 시간이 소요되는 데이터 전송 동작과 Checksum 계산 및 프로토콜 계층간 패킷 형식 변환 동작을 지원하는 코어 IP(Intellectual Property) 개발을 설계 사양으로 한다. 특히 TCP 프로토콜의 경우 타이머 관리, 슬라이딩 윈도우 관리 등의 버퍼제어가 필요한데, 이러한 작업은 호스트 프로세서에서 소프트웨어로 구현하도록 한다.

즉 스택 프로세서에서는 TCP 계층의 Checksum 계산부분과 IP 계층을 구현한다.

### 3.2 TCP/IP 프로토콜 스택 프로세서 구조

3.1절에서 정의한 설계 사양을 바탕으로 구현한 TCP/IP 프로토콜 스택 프로세서 코어 IP의 구조는 그림 2와 같다.

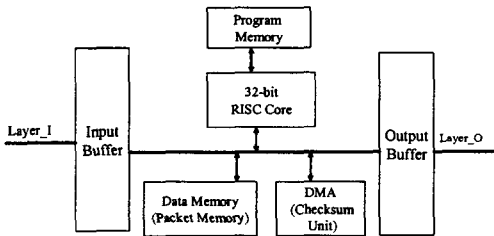


그림 2. TCP/IP 스택 프로세서 코어 블록도

설계된 프로세서는 크게 입력 버퍼 메모리 와 32 비트 RISC 코어, 체크섬 계산 기능을 갖는 DMA(Direct Memory Access) 제어기, 패킷 메모리 와 출력 버퍼 메모리로 구성된다. 체크섬 회로는 TCP/IP의 송·수신부에 사용되기 위해 체크섬 생성 기능과 체크섬 오류 감지기능을 갖고 있다. TCP와 IP 계층의 다양한 선택 사양(Option)을 상태로 구현하는 것은 융통성이 없고 지원 프로토콜의 확장에 제약을 야기하므로, 본 연구에서는 TCP/IP 동작 분석을 바탕으로 28개의 전용 명령을 갖는 32 비트 RISC 마이크로프로세서를 설계하여<sup>[8]</sup>, RISC Core가 데이터 형식 변경과 제어 기능을 담당하도록 하여, 프로그램 메모리 내용 교체를 통해 다양한 옵션을 지원할 수 있도록 하였다. RISC 내부 구조는 기존 마이크로프로세서와 달리 Von-Neumann 구조가 아닌 Harvard 구조를 갖고 있으며, 5단 파이프라인과 함께 TCP/IP의 프로토콜 처리에 적합한 전용 명령어를 갖고 있다. 그리고 패킷(데이터) 메모리는 RISC 코어의 연산 결과의 중간 값을 저장하는 데이터 메모리 역할과 함께, 입력 버퍼 메모리에서 읽은 데이터를 일시적으로 유지하는 Packet Memory 역할을 동시에 수행한다.

그림 2에서 입력 버퍼 메모리는 송신 동작시에는 상위 계층에서 입력 데이터를 받고, 수신 동작의 경우에는 하위 계층에서 데이터를 입력받게 된다. 입력 버퍼에 담긴 데이터와 제어 정보를 RISC 코어가 해독하여 적절한 프로토콜 처리와 함께 DMA와 Checksum Unit을 이용하여 생성된 체크섬을 출력 버퍼 메모리로 전달한다. 반면 수신시의 경우에는 하위 계층에서 데이터를 받아서 데이터를 프로토콜 처리에 맞게 조정하

며, 체크섬 오류여부를 검사함과 동시에 상위 계층으로 변경된 데이터를 보낸다. 입력 버퍼 메모리는 입력 데이터 버퍼, 입력 헤더 버퍼, 입력 제어 버퍼, 플래그 레지스터로 구성된다. 플래그 레지스터를 제외한 3가지 버퍼는 그림 3과 같이 이중 बैं크(dual-bank) 구조를 갖고 있어서, 내부 RISC Core 동작시 입출력 동작이 가능하도록 하여, 입출력에 따른 오버헤드를 최소화하도록 하였다. 입출력 버퍼의 경우 FIFO(First-In First Out) 구조를 하지 않은 이유는, TCP/IP 프로토콜의 경우 ATM 프로토콜과는 달리 데이터 필드의 크기가 가변적이지 때문에, 하드웨어는 많이 필요하지만 dual-bank 구조가 제어가 용이하다고 판단했기 때문이다. 출력 버퍼도 유사한 이중 बैं크 구조를 갖고 있다.

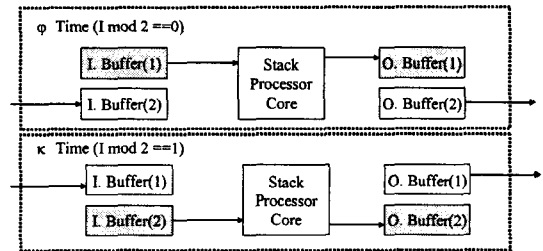


그림 3. 이중 बैं크(Dual Bank) 구조를 갖는 입출력 버퍼 동작

Checksum 생성 및 Checksum Error 감지 기능을 갖는 DMA 제어기는, 다양한 프로토콜 지원을 위해 다음과 같은 3가지 동작 모드를 갖는다.

- 1). DMA 동작 모드
- 2). DMA + Checksum 모드
- 3). Checksum 모드

즉 3번째 Checksum 모드는 체크섬 보조 프로세서와 유사한 동작을 수행한다. 체크섬 회로는 32 비트 데이터 버스를 통한 DMA 데이터 전송시, 데이터 버스 값을 래지 하였다가 32 비트 데이터를 상·하위 16 비트씩 나누어 병렬적으로 1의 보수 연산을 통해 체크섬을 계산한 후, 마지막에 상위와 하위 체크섬에 대해 1의 보수 덧셈과 1의 보수 연산 과정을 통해 최종 체크섬을 얻거나, 체크섬 에러를 감지한다. 그리고 DMA는 데이터의 근원지로 입력 버퍼와 Packet 메모리를 사용하며, 목적지로 Packet Memory와 출력 버퍼를 사용한다.

#### IV. 회로 검증 및 성능 분석

설계된 TCP/IP 스택 프로세서 회로는 Verilog HDL 을 통해 검증하였으며<sup>[9-10]</sup>, 이중 뱅크 구조를 갖는 입출력 버퍼, 프로그램 메모리, 패킷 메모리를 제외한 32 비트 RISC 코어와 체크섬 기능을 갖는 DMA 장치는, IDEC에서 제공하는 삼성 0.35um 공정 표준 셀로 합성한 결과 각각 약 22,000 과 1,680 게이트로 구성되며, 최대 동작 주파수는 167 Mhz를 가짐을 확인하였다. 그리고 설계된 TCP/IP 스택 프로세서 코어를 송신부 IP로 구현하기 위해 내부 프로그램 메모리를 한 결과 120개의 명령어가 사용되었다. 그림 4는 Verilog-XL로 본 연구에서 설계한 TCP/IP 스택 프로세서 코어를 검증한 파형을 나타낸다.

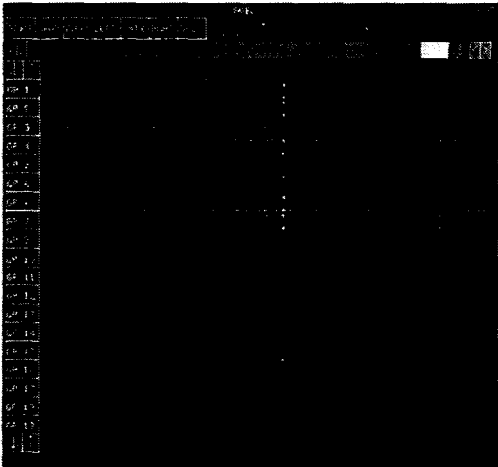


그림 4. TCP/IP 스택 프로세서 코어 검증 파형

#### VI. 결론 및 추후 연구

본 논문은 TCP/IP 프로토콜을 구현하기 위한 스택 프로세서 코어를 Verilog HDL로 설계하였다. 현재 송신부에 대한 내부 프로그램 메모리 내용은 완성하여 검증하였으며, 수신부에 대한 RISC 내부 프로그램은 개발 중에 있다. 그리고 Xilinx FPGA와 PCI 인터페이스 환경에서 검증을 하기 위한 작업을 진행하고 있다. 송신부와 함께 수신부에 대한 내부 프로그램 메모리 코드가 완료되면, 설계된 회로를 네트워크 프로세서와 네트워크 인터페이스 보드(Network Adaptor Board)에 IP 형태로 사용될 수 있을 것으로 판단된다.

#### 감사의 글

본 연구는 산업 자원부 반도체 연구조합 시스템 IC 2010과제의 연구비로 수행되었으며, 회로 설계에 사용된 CAD 소프트웨어는 IDEC 지원에 의한 것입니다.

#### 참고문헌

- [1] L. Roberts, "Internet Still Growing Dramatically Says Internet Founder," <http://www.caspiannetworks.com/press/release/08.15.01.shtml>, Aug, 2001.
- [2] Marc Necker, Didier Contis, and David Schimmel, "TCP-Stream Reassembly and State Tracking in Hardware", Proc. of the 10th Annual IEEE Symposium on Field-Programmable Custom Computing Machines(FCCM'02), pp.1-2, 2002.
- [3] David V. Schuehler, and John W. Lockwood, "TCP Splitter: A TCP/IP Flow Monitor in Reconfigurable Hardware," IEEE Micro January-February, 2003, pp.54-59.
- [4] 진교홍, 고속 실시간 통신을 위한 TCP/IP 프로토콜의 하드웨어 설계 및 구현, 부산대학교 컴퓨터 공학과 공학 박사 논문 1997. 8.
- [5] D. Clack and V. Jacobson, "An Analysis of TCP Processing Overhead," IEEE Communications Magazine, vol. 27, no.6, pp.23-29, June, 1989.
- [6] W. Doeringer, and D. Dykeman, etc., "A Survey of Light Weight Transport Protocols for High-Speed Networks," IEEE Trans. on Communications, vol. 38, no.11, pp.2025-2039, Nov., 1990.
- [7] Jeremy Bentham, TCP/IP Lean : Web Server for Embedded System, 2nd Edition, CMP, 2002
- [8] 박성일, 최 병 윤, "32 비트 RISC 마이크로컨트롤러 설계", 2003년도 대한 전자 공학회 하계 학술 발표 대회 논문집 발표 예정
- [9] 최 병 윤, "Verilog HDL을 사용한 디지털 시스템 설계 및 실습", 부산대학교 IDEC, 2002. 1
- [10] 최 병 윤, "RISC 프로세서 구조", 부산대학교 IDEC, 2001. 1.