

인텔 IXP28x0 네트워크 프로세서 및 응용

민경주, 권택근 (충남대학교)

I. 서론

최근 SoC (System on Chip) 기술의 발전으로 최대 10 Gbps의 처리율을 갖는 네트워크 프로세서가 개발되고 있다. 네트워크 프로세서는 기존의 ASIC (Application Specific Integrated circuit) 또는 FPGA (Field Programmable Gate Array) 등 하드웨어가 수행하던 고속의 패킷 처리 기능을 소프트웨어 기반으로 처리하도록 함으로써 다양한 기능의 패킷 처리를 저비용으로 단시간 내에 개발할 수 있는 장점을 갖고 있다^[1].

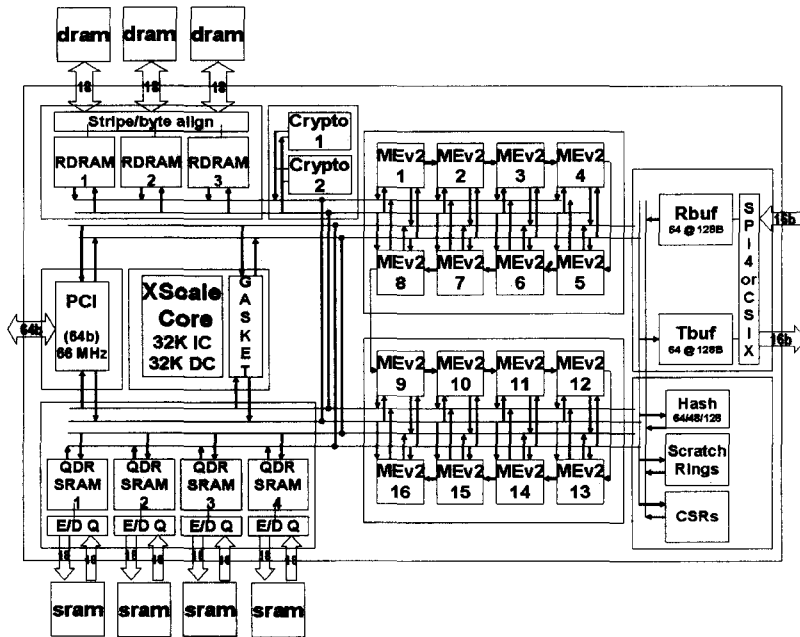
최근의 통신 시스템은 패킷 포워딩의 단순 기능에서부터 멀티서비스, 통합 서비스, 그리고 보안 서비스 등 다양한 기능의 제공이 필수적이다^[2]. 그리고 이러한 요구 사항은 ISP (Internet Service Provider) 및 사용자의 환경에 따라 다양하게 변화하고 있는 실정이다. 따라서 유연성에 기반한 시스템의 개발은 개발 기간과 비용을 줄이는 중요한 요인이 되고 있다. 따라서 네트워크 프로세서에 기반한 시스템의 구축은 이러한 점에서 가장 적합한 시스템 개발 방법 중 하나이다.

90년대 후반부터 시작된 네트워크 프로세서 칩의 개발은 최근들어 변화를 겪고 있다. 초기에

IBM, C - Port, Agere 등에서 주도하던 네트워크 프로세서는 하드웨어 기반의 패킷 처리에 비해 낮은 성능을 제공하는 반면, 최근의 Intel, EZChip 등 후발 네트워크 프로세서 칩 개발 업체는 10 Gbps 급 패킷 처리 성능을 하나의 칩으로 제공하고 있다.

EZChip은 Full - duplex 10 Gbps 처리율을 갖고, Intel의 IXP28x0은 Half - duplex 10 Gbps 처리율을 갖는다^[3]. EZChip은 패킷 분류 (classification)를 위한 하드웨어를 포함하는 반면, Intel IXP28x0 중 IXP2850은 암호화 / 복호화 엔진을 탑재하고 있다. 일반적으로 EZChip은 하드웨어에 의한 고속 패킷 처리에 유리하고, Intel IXP 계열의 네트워크 프로세서는 소프트웨어 기반 패킷 처리에 있어 유연하다. 따라서 네트워크 프로세서간 특징이 있어 응용에 따라 최적의 네트워크 프로세서가 달라질 수 있고, 여기서는 패킷 처리의 유연성에 보다 큰 비중을 둔 Intel IXP28x0에 대하여 논하도록 한다.

본 고의 구성은 다음과 같다. II장에서는 Intel IXP28x0의 세부 구조에 대해 살펴보고, III장에서는 이러한 네트워크 프로세서를 이용한 개발의 적용을 위한 IPv4 포워딩의 마이크로 블록과 네



〈그림 1〉 IXP2850 구조

트위크 프로세서를 이용한 하드웨어 및 응용에 대해 소개하고 결론 및 향후과제는 IV장에서 기술한다.

II. 구조

인텔 IXP2850 네트워크 프로세서의 내부 구조를 살펴보면 [그림 1]과 같다.

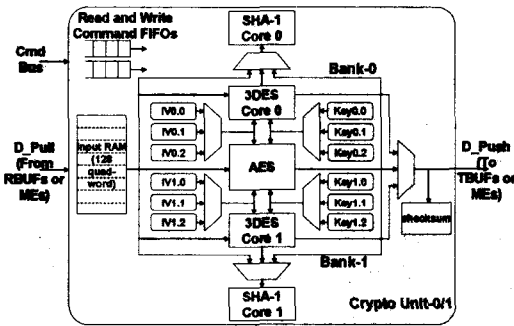
패킷 저장을 위한 3개의 대용량 DRAM 인터페이스와 테이블 관리를 위한 4개의 고속 SRAM 인터페이스를 갖고, 패킷 입출력 인터페이스로, SPI - 4.2와 CSIX 인터페이스를 선택할 수 있다. 고속의 패킷 처리는 16개의 마이크로 엔진 (ME: microengine)에서 담당하고, 제어를 위한 복잡한 기능은 XScale에서 처리되도록 한다. XScale core는 700 MHz의 속도로 ARM V5TE (Version 5

Thumb instruction Enhanced) core와 거의 동일한 명령어를 처리하므로 Linux나 VxWorks 등 범용 운영체제가 수행 될 수 있다^[2,3,4,5]. 마이크로엔진은 1.4 GHz의 속도로 특수 명령어를 처리하는 RISC (Reduced Instruction Set Computer) 기반 프로세서로 16 개의 마이크로엔진을 파이프라인 형태로 구성하거나 부하 분산의 방법으로 구성하여 패킷 처리를 병렬화한다⁶⁾. 마이크로엔진에 대한 보다 자세한 사항은 다음 절에서 다루도록 한다.

그리고 두 개의 크립토유닛 (Crypto Unit)은 IXP2800에는 존재하지 않는 IXP2850에만 존재하는 유닛으로 IPSec (Internet Protocol Security) 등에서 암호화 및 복호화 기능을 담당하는 하드웨어이다. 외부 인터페이스와의 밀접한 관련이 있는 MSF (Media Switch Fabric)은 수신된 메시

<표 1> IXP2400, IXP2800, IXP2850의 특징

XScale 코어	600MHz	700MHz	700MHz
마이크로엔진	8@600MHz	16@1.4GHz	16@1.4GHz
SHaC 유닛	지원	지원	지원
MSF	SPI-3 Utopia, CSIX-L1	SPI-4, CSIX-L1	SPI-4, CSIX-L1
PCI 제어기	지원	지원	지원
SRAM제어기	2	4	4
DRAM제어기	1, DDR	3, Rambus	3, Rambus
암호화기능	미지원	미지원	지원



<그림 2> IXP2850의 크립토 유닛 구조

지 처리를 위한 Rbuf (Receive Buffer)와 Tbuf (Transmit Buffer)가 있으며 SPI - 4나 CSIX는 각각 10기가 외부 인터페이스와 스위치 패브릭을 위한 인터페이스이다. 마지막으로 포워딩을 위한 정보를 저장하기 위한 해쉬(Hash)나 스크래치 링(Scratch Ring), 제어 및 상태 레지스터 (CSR: Control & Status Register)가 IXP2850의 주요 구성이다.

여기서 IXP2400^[3], IXP2800^[4], 그리고 IXP2850의 시스템적인 측면에서의 중요한 차이

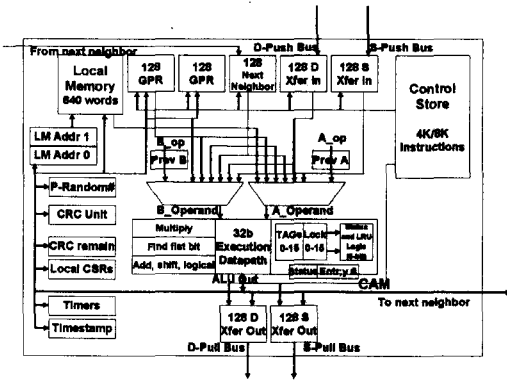
점을 살펴보면 [표 1]과 같다.

[표 1]을 보면, IXP2400과 IXP2800은 많은 차이점을 보이고 있다. XScale 코어를 비롯하여 각 마이크로엔진의 성능, MSF 등에도 많은 차이를 보인다. 반면 IXP2800과 IXP2850의 경우는 데이터의 암호화/복호화를 담당하는 크립토 유닛의 유무에서만 차이를 갖는데, IXP2850에서 도입된 크립토 유닛의 내부도가 다음의 [그림 2]에 나타나 있다.

[그림 2]에서 보는 바와 같이 크립토 유닛은, 핵심인 AES (Advanced Encryption Standard), 두 개의 3 - DES (triple Data Encryption Standard), 입력 데이터에 대해 1 - way 해쉬 함수를 계산하는 SHA - 1 (Secure Hash Algorithm), HMAC 그리고 체크섬(checksum) 어큐뮬레이터 등으로 구성되어 있다.

1. 마이크로엔진 구조

[그림 1]에서 전술한 바와 같이 인텔 네트워크 프로세서의 핵심 중 하나인 마이크로엔진은 두



(그림 3) MEv2의 구조

개의 클러스터로 구분되어 있고, 각 클러스터에는 8개의 마이크로엔진으로 구성되어 있다. 각 마이크로엔진을 구분하기 위한 식별자를 위해서는 0 또는 1번째 클러스터의 몇 번째 마이크로엔진인가로 구분한다. 가령 1번째 클러스터의 3번째 마이크로엔진(그림 1)의 11번 마이크로엔진의 경우 식별자는 1:3과 같이 '클러스터명:클러스터에서의 n번째 마이크로엔진'이 된다. 각각의 마이크로엔진의 세부적인 구조는 다음의 [그림 3]과 같다.

[그림 3]의 구조를 보면 MEv2 (Micro Engine version 2)는 여러 개의 메모리 블록과 레지스터들로 구성되어 있고, 인스트럭션에 의해 연산장치 (ALU: Arithmetic & Logical Units)를 거쳐, 다음 마이크로엔진이나, 다른 레지스터 또는 유닛 등으로 결과를 내보내는 역할을 수행한다. 위 그림에서 보이는 각각의 장치들을 살펴보면 다음과 같다.

- ① Local Memory : 각 마이크로엔진은 640 롱 워드(Long Word)의 로컬 메모리를 가지고 있는데, 레지스터보다는 지연시간이 길지만 SRAM 및 DRAM에 비해 빠른 접근이

가능하다. 로컬 메모리는 번지로 접근이 가능하며 칩 내부에는 스크래치패드 링을 할당하고, 마이크로엔진 내부에는 이 로컬 메모리를 할당하는데, 엔진 내부에 있어 속도 향상에 우수한 성능을 보인다.

- ② GPR (General Purpose Register: 범용 레지스터) : 각 마이크로엔진은 256개의 32비트 범용 레지스터를 갖는다. [그림 3]에서 알 수 있듯이, 두 개의 뱅크에 범용 레지스터가 할당되는데, 각각의 뱅크는 128개의 범용레지스터를 갖는다.
- ③ NN (Next Neighbor) 레지스터 : 인텔 IXP2850 네트워크 프로세서는 두 개의 클러스터에 모두 16개의 마이크로엔진을 가지고 있는데, 각 마이크로엔진간 통신을 위해 이 레지스터를 사용하고 마이크로엔진간 데이터를 전달하는 기능을 수행한다. NN은 32비트 레지스터로 모두 128개가 존재한다.
- ④ SRAM 트랜스퍼 레지스터 : 비동기 메모리 연산을 지원하기 위해 사용되는 레지스터로 SRAM으로 들어올 때는 '쓰기만 가능'하고 SRAM에서 나갈때는 '읽기만 가능'하다. 마이크로엔진으로 들어올 때와 나갈 때를 위해 이 SRAM 트랜스퍼 레지스터는 각각 128개씩 존재함을 [그림 3]의 하단부에서 보이고 있다.
- ⑤ DRAM 트랜스퍼 레지스터 : SRAM 트랜스퍼 레지스터와 유사한 기능을 위해, 각 128개씩 존재한다.
- ⑥ ALU : SRAM, DRAM 또는 기타 레지스터를 통해 들어온 결과를 처리하는 연산장치이다.
- ⑦ CRC (Cyclic Redundancy Check) 유닛 및

〈표 2〉 마이크로엔진 내부 레지스터

GPR	reg	@reg
SRAM Transfer	\$reg	N/A
DRAM Transfer	\$\$reg	N/A
Neighbor	n\$reg	N/A
Local Memory	N/A	N/A

CRC remain : 패킷의 CRC를 검사하는 기능을 수행하며, 이러한 기능은 매우 빈번히 사용되고, 소프트웨어적으로 처리하기에는 너무 복잡하므로, 하드웨어를 통해 수행된다.

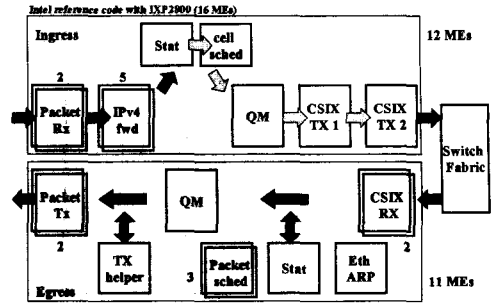
- ⑧ 의사 난수발생기 (Pseudo - random number generator) : 마이크로엔진에서 난수를 생성할 때 사용하는데, 해쉬나 혼잡제어 등에 활용할 수 있다.
- ⑨ Timer 및 Timestamp : 마이크로엔진간 동기화를 하거나, 시간의 흐름을 제어하는 기능을 수행한다.

위에서 살펴본 바와 같이 인텔 IXP2850 네트워크 프로세서를 이용하기 위해서는 다양한 레지스터를 다루게 된다. 이러한 레지스터를 이용해 마이크로코딩을 하게되면, 다음의 [표 2]와 같은 규칙을 따라 접근하여 읽기/쓰기가 가능하다.

상대 레지스터는 마이크로엔진에서 수행되는 쓰레드마다 존재하는 동일한 이름의 레지스터이므로 쓰레드 로컬 (thread local) 레지스터라고도 불린다. 절대 레지스터는 쓰레드에 공유되는 레지스터로 전역 (global) 레지스터라고도 불린다.

III. Intel IXP28x0을 이용한 라우터 개발

인텔 네트워크 프로세서는 필요한 응용에 맞



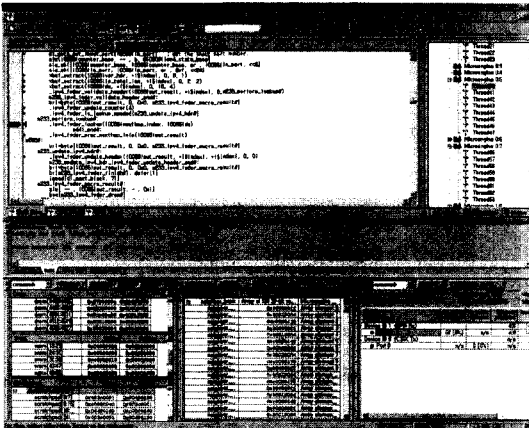
〈그림 4〉 10x 1Gb IPv4/v6 마이크로 포워딩 마이크로 블록

게 IPv6 주소변환기나, 초고속 라우터 개발 등에 이용될 수 있다. 본 장에서는 네트워크 프로세서를 이용한 초고속 라우터의 개발 과정을 소개한다. 인텔 IXP28x0을 이용하여 초고속 라우터를 개발하기 위해서는 다음과 같은 절차를 따라 수행된다.

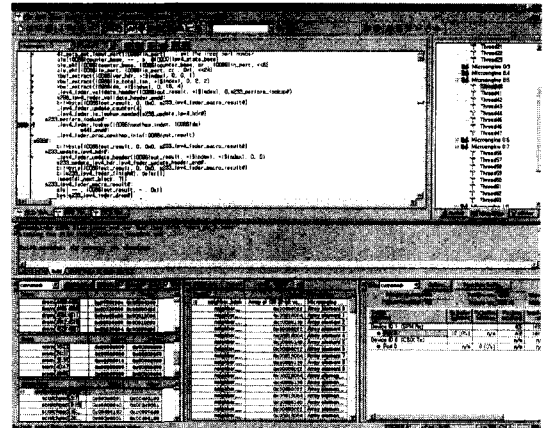
- ① 마이크로 블록의 설계
- ② SDK를 이용한 시뮬레이션
- ③ IXP28x0에 적용

1. 마이크로 블록

2장에서 설명한 바와 같이, 인텔 IXP28x0 네트워크 프로세서를 사용한 10 Gbps 성능의 시스템은 상향 진입지점(Ingress) 및 하향 진입지점(Egress)에 각각의 네트워크 프로세서를 사용하고, 이 각각의 네트워크 프로세서는 16개의 마이크로엔진으로 구성되어 있다. 따라서 16개의 마이크로엔진이라는 제한된 자원을 개발 목적에 맞추어 할당할 필요가 있다. 또한 목표 성능을 달성하기 위하여 마이크로엔진을 병렬로 연결하거나 순차적으로 처리하는 등 병렬 처리를 해



<그림 5> SDK 3.5를 이용한 시뮬레이션



<그림 6> NP 서버보드의 개발 환경

야한다. 현재 구현된 IPv4의 라우터 기능 개발을 위한 마이크로 블록과 이에 따른 마이크로엔진의 할당을 다음의 [그림 4]에서 보여주고 있다.

[그림 4]는 IXP2800을 이용하기 위해 인텔에서 레퍼런스로 제공한 기본적인 마이크로 블록들을 보여주는데, 마이크로 블록 위의 숫자는 그 마이크로 블록에 할당된 마이크로엔진의 수를 의미한다. Ingress에서는 패킷 포워딩을 위한 마이크로 블록에 마이크로엔진을 가장 많이 할당하고, Egress에서는 패킷 스케줄링에 마이크로엔진을 많이 할당한다. 마이크로 블록은 빌딩 블록과 같이 원하는 기능을 수행할 블록을 설계하고, 그 마이크로 블록에 마이크로엔진을 할당하는 방식으로 작업이 수행되기 때문에, 여분의 마이크로엔진 (Ingress : 4, Egress : 5)을 효율적으로 배분할 필요가 있다.

2. SDK 시뮬레이터

마이크로 블록을 설계하고, 그 세부적인 기능에 대해 마이크로 코딩을 위한 개발툴로 SDK

(Software Development Kit) 3.5를 사용하는데, 이 툴은 Micro - C와 Microcode를 지원할 수 있다. 하지만 네트워크 프로세서를 이용한 라우터 개발 시 하드웨어에 종속적인 특성한 microcode를 이용해 개발하기로 한다. 이러한 마이크로코드를 이용한 마이크로코딩을 컴파일하고 미리 시뮬레이션 할 수 있는데, 이러한 SDK를 이용한 시뮬레이션 과정이 다음의 [그림 5]에 나타나 있다.

이러한 일련의 과정이 정상적으로 수행이 된다면 그 결과로 마이크로엔진이 실행할 수 있는 uof 파일이 생성이 되는데, 이 파일을 IXP28x0에 실제 적용하여 시뮬레이션을 통해 얻은 결과와 동일한 동작을 하는지 확인하면 된다. [그림 6]은 ATCA (Advanced Telecom and Computer Architecture)^[7] 기반으로 개발중인 NP (Network Processor) 블레이드에 사용될 NP 서버보드의 개발 환경을 보이고 있다.

IV. 결론 및 향후 과제

초고속만을 향해 달려가던 통신 시스템이

1.25 대란을 경험하면서, 사용자 및 ISP들로부터 보안 기능의 강화와 다양한 부가서비스 등에 대한 요구가 있어왔다. 이러한 서비스를 제공하기 위한 핵심 장비로 라우터를 들 수 있는데, 나날이 발전하는 비정상적인 공격이나, 새로운 형태의 침해에 적용하기 위한 대안으로 네트워크 프로세서가 대두되었다. 네트워크 프로세서는 다양한 형태의 침해 패턴이나 새로운 형태의 공격 등에 대해 빠른 성능과 빠른 대응 능력을 가질 수 있는 것으로 판단된다. 이는 Microcode라는 새로운 형태의 소프트웨어를 통해 개발이 되기 때문에, 패킷의 보다 유연성 있는 처리가 가능할 뿐만 아니라, 최근 하드웨어 개발자들에게 이슈로 떠오르고 있는 ATCA에 기반한 NP 블레이드를 개발함으로써 보다 다양한 형태의 통신망에 적용이 가능할 것으로 예상된다.

현재 네트워크 프로세서를 이용하여 초고속 라우팅 기능을 가진 패킷 필터링, 트래픽 모니터링 등 침해 감시 및 대응이 가능한 라우터가 현재 개발중이고, 개발중인 NP 블레이드를 실제 라우터로 적용하여 패킷 포워딩 기능을 비롯하여 다양한 네트워크 정보보호 기능을 개발할 계획이다.

==== 참고문헌 =====

- [1] Erik J. Johnson and Aaron R. Kunze, "IXP2400/2800 Programming", Intel Press, 2003.
- [2] W. Feghali, B. Burres, G. Woltich, and D. Carrigan, "Security: Adding Protection to the Network via the Network Processor," Intel Technology Journal, <http://developer.intel.com/technology/itj/2002/volume06issue03/> Aug 2002.
- [3] Intel, Intel® IXP2400 Network Processor, "<http://www.intel.com/design/network/products/npfamily/ixp2400.htm>", Jun.20, 2004.
- [4] Intel, Intel® IXP2800 Network Processor "<http://www.intel.com/design/network/products/npfamily/ixp2400.htm>", Jun.20, 2004.
- [5] B. Carlson, Intel® Internet Exchange Architecture and Applications: A Practical Guide to IXP2XXX Network Processors, Intel Press, 2003.
- [6] Intel, Intel® IXP2800/IXP2850 Network Processor: Hardware Reference Manual, 2002.
- [7] PICMG, Advanced TCA, <http://www.picmg.org/> Jul. 2004.

저자소개



민 경 주

2000년 충남대학교 컴퓨터공학과 공학사
 2002년 충남대학교 컴퓨터공학과 공학석사
 2002년-현 재 충남대학교 컴퓨터공학과 박사과정
 2002년-2004년 한국전자통신연구원 표준연구센터
 연구원
 주관심분야 네트워크 프로세서, 정보보호, 통신프로
 토콜표준화, VoIP



권 택 군

1988년 서울대학교 컴퓨터공학과 공학사
 1990년 서울대학교 컴퓨터공학과 공학석사
 1996년 서울대학교 컴퓨터공학과 공학박사
 1992년-1998년 LG 전자 정보통신연구소 연구원
 1993년-1994년 미국 Washington Univ. 방
 문연구원
 2002년-2003년 미국 Erlang Technology
 방문연구원
 1998년-현 재 충남대학교 전기정보통신공학
 부 부교수
 주관심분야 네트워크 프로세서, 통신 시스템, 네트워크
 정보보호, 초고속 통신망, 시스템 소프트웨어