

# 네트워크 프로세서의 성능 예측을 위한 고속 이더넷 제어기의 상위 레벨 모델 검증 (Model Validation of a Fast Ethernet Controller for Performance Evaluation of Network Processors)

이 명 진 <sup>†</sup>  
(Myeong-jin Lee)

**요약** 본 논문에서는 SystemC를 이용하여 네트워크 SOC에 적용이 가능한 상위 계층 설계 방법을 제안한다. 본 방식은 실제 양산되고 있는 네트워크 SOC를 기본 플랫폼으로 하여 NAT 라우터에서 보다 높은 변환율을 얻기 위한 최적의 하드웨어 계층 결정을 목표로 한다. 네트워크 SOC에 내장된 고속 이더넷 MAC, 전용 DMA, 시스템 모듈들은 트랜잭션 레벨에서 SystemC를 이용하여 모델링되었다. 고속 이더넷 제어기 모델은 실제 Verilog RTL의 동작을 사이클 단위로 측정된 결과를 토대로 동작이 세부 조정되었다. SystemC 환경의 NAT 변환율은 기존 플랫폼 검증 보드상의 측정 결과와 비교하여  $\pm 10\%$  이내의 오차를 보였고, RTL 시뮬레이션보다 100배 이상의 속도 이득을 보였다. 본 모델은 NAT 라우터에서 성능 저하의 원인을 찾는 SOC 구조 탐색을 위해 사용될 수 있다.

**키워드** : 네트워크 프로세서, 구조 탐색, 네트워크 주소 변환(NAT), 인터넷 게이트웨이, 트랜잭션 레벨 모델링(TLM)

**Abstract** In this paper, we present a high-level design methodology applied on a network system-on-a-chip(SOC) using SystemC. The main target of our approach is to get optimum performance parameters for high network address translation(NAT) throughput. The Fast Ethernet media access controller(MAC) and its direct memory access(DMA) controller are modeled with SystemC in transaction level. They are calibrated through the cycle-based measurement of the operation of the real Verilog register transfer language(RTL). The NAT throughput of the model is within  $\pm 10\%$  error compared to the output of the real evaluation board. Simulation speed of the model is more than 100 times faster than the RTL. The validated models are used for intensive architecture exploration to find the performance bottleneck in the NAT router.

**Key words** : network processor, architecture exploration, Network Address Translation(NAT), Internet gateway, Transaction Level Modeling(TLM)

## 1. 서론

인터넷 기술의 발전에 따라 가정에서도 여러대의 컴퓨터가 사용되고 있으며 소형 사무실(Small Office Home Office)이 급격히 증가하고 있다. 이에 따라 인터넷 접속 라인의 공유에 대한 필요가 대두되었으며, 그림 1과 같은 형태의 인터넷 공유를 위한 장비들이 출시되고 있다. 이들 장비들은 일반적으로 인터넷 접속을 위한 WAN(Wide Area Network) 포트, 다수의 구내 컴퓨터들을 위한 LAN(Local Area Network) 포트, 네트워크

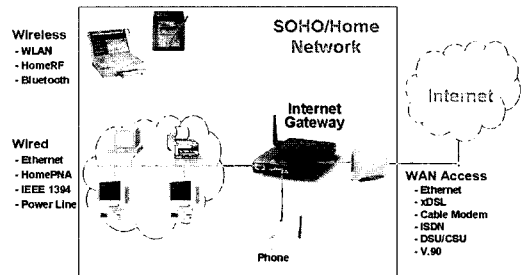


그림 1 인터넷 게이트웨이를 통한 인터넷 공유

<sup>†</sup> 정 회 원 : 경성대학교 전기전자메카트로닉스공학부 교수  
mjlee@ieee.org

논문접수 : 2004년 5월 27일

심사완료 : 2004년 10월 28일

SOC(System on a Chip), 실시간 운영체제(Real-time Operating System, RTOS)와 IP(Internet Protocol) 공

유 소프트웨어 등으로 구성된다.

네트워크 SOC는 RISC 프로세서 코어와 다수의 고속 이더넷 MAC, 이더넷 스위치, 직렬 통신 제어기 등으로 구성된다. SOC 설계시 프로세서 코어와 제어기 들은 일반적으로 IP(Intellectual Property) 형태로 제공되며 단일 칩에 통합된다. 이에 따라 설계의 복잡도가 증가하고, 새로운 IP들의 출현에 따라 설계된 제품의 수명이 과거보다 훨씬 짧아지고 있다. 따라서, 빠르게 변화하는 시장에 적시에 제품을 출시하기 위해서는 설계 단계부터 목표 시스템의 성능을 정확하게 예측하여 설계 기간을 줄이는 것이 중요하다.

설계시 목표 SOC의 성능 평가를 위해서는 RTL 시뮬레이션과 FPGA(Field Programmable Gate Array) 프로토타이핑(prototyping)이 사용될 수 있다. 그러나, RTL 시뮬레이션은 RTOS 환경에서 응용 프로그램을 수행시키기까지 매우 많은 시간을 필요로 한다. FPGA 프로토타이핑은 실제 시스템 클럭 속도와 같은 동작 환경을 구축하거나, 캐쉬 크기와 시스템 버스 속도 등과 같은 하드웨어 변수들을 가변시키면서 성능을 측정하기가 어렵다. 따라서, 목표 시스템의 성능을 만족시킬 수 있는 하드웨어 구조 탐색시 시스템의 세부 동작들을 상위 계층의 동작으로 추상화하여 모델링하는 기법이 필요하다.

이를 위해 하드웨어와 소프트웨어의 동시 시뮬레이션(co-simulation)이 가능한 Seamless[1]와 CoCentric[2]과 같은 상용 시뮬레이션 환경들이 발표되었다. SystemC[3]는 시스템 동작 레벨과 RTL 레벨에서 설계를 위한 시뮬레이션 커널과 C++ 클래스 라이브러리로 구성된다. SystemC를 이용한 설계와 검증을 위해 CoCentric 환경이 제공되는데 C나 C++ 언어로 개발된 소프트웨어는 SystemC 블록으로 변환이 가능하고 CoCentric 환경에서 시뮬레이션 된다. 응용 소프트웨어의 실행은 목표 프로세서의 트랜잭션-레벨(transaction-level) 또는 핀 레벨(pin-level) 모델을 이용하여 시뮬레이션 될 수 있다. 그러나, 핀-레벨 모델은 실행 시간이 트랜잭션-레벨 모델에 비해 길다.

본 논문에서는 SystemC를 이용하여 네트워크 SOC에 내장된 고속 이더넷 제어기와 주요 시스템 모듈들을 설계하고, RTOS 환경에서 NAT[4] 성능을 측정한다. 목표 네트워크 SOC는 현재 시장에서 그 기능과 성능이 충분히 인정되어 양산중이며, 인터넷 공유기에 널리 사용되고 있다. 그러나, 현재의 NAT 변환율은 설계 당시의 예상 성능에 미치지 못한다. 따라서, 향후 새로운 방식에 의한 인터넷 접속의 광대역화가 이루어질 경우에 대비하여 NAT 변환율 저하의 원인을 파악하고, 최대의 NAT 변환율을 갖는 최적의 네트워크 SOC 구조를 찾는

것이 필요하다. 고속 이더넷 제어기와 주요 시스템 모듈들은 트랜잭션-레벨에서 모델링되었고, 성능은 네트워크 SOC의 실제 검증 보드 상에서 측정된 결과와 비교되었다. 성능 측정과 비교를 통하여 고속 이더넷 제어기 모델은 실제 하드웨어와 비교시 성능 오차가  $\pm 10\%$  이내가 되도록 보완되었다. 이와 같이 검증된 모델을 이용하여 RISC 코어의 캐쉬 크기, 고속 이더넷 전용 DMA의 버퍼 크기, CPU 클럭 속도 등을 가변시키면서 향후 설계될 네트워크 SOC의 NAT 변환율이 최대가 될 수 있도록 구조 탐색을 수행하였다.

## 2. 인터넷 게이트웨이를 위한 목표 네트워크 SOC

인터넷 게이트웨이는 일반적으로 RISC 코어와 두 채널 이상의 고속 이더넷 인터페이스가 내장된 네트워크 SOC를 핵심 부품으로 사용한다. 그림 2는 본 논문에서 성능 개선 및 구조 탐색을 목표로 하는 네트워크 SOC인 S3C2510A[5]의 블록도를 나타낸다. 목표 네트워크 SOC는 ARM940T RISC 코어, 두 채널의 고속 이더넷 MAC과 전용 DMA, ATM SAR(Segmentation and Reassembly) 제어기, USB 호스트, PCI 호스트 등으로 구성된다. 여러 네트워크 제어기들, 운영체제, 네트워크 응용 소프트웨어의 적절한 조합을 통해, 본 네트워크 SOC는 다양한 유형의 인터넷 접속 방식, 홈 네트워크 환경, 네트워크 응용 서비스들을 지원할 수 있다.

향후 출시될 네트워크 SOC들 역시 무선랜, 블루투스, IEEE1394 등과 같이 새롭게 채택될 몇 가지 네트워크 제어기를 제외하고는 기본 구조는 본 논문에서 성능 개선을 목표로 하는 네트워크 SOC의 구조와 유사할 것으로 예상된다. 따라서, 목표 네트워크 SOC는 향후 네트워크 SOC 설계시 성능 예측을 위한 기준 플랫폼으로 사용될 수 있다.

목표 네트워크 SOC의 주된 응용은 인터넷 서비스 제공자(ISP)에 의해 할당되는 공중 IP 주소(public IP address)와 사설 IP 주소들 간의 네트워크 주소 변환(NAT)이다. 한 채널의 고속 이더넷 제어기에는 공중 인터넷 접속을 위해 사용되며, 다른 채널은 사설 네트워크 접속을 위해 사용된다. 본 논문에서는 대다수의 인터넷 공유기가 성능 지표로 삼고 있는 NAT 변환율을 시스템 관점의 성능 지표로 삼고, 고속 이더넷 제어기의 모델링 및 구조 탐색을 수행한다.

## 3. SystemC를 이용한 고속 이더넷 제어기와 전용 DMA의 모델링과 검증

### 3.1 목표 네트워크 SOC의 SystemC 모델링

그림 3은 목표 네트워크 SOC의 SystemC 모델링을 위

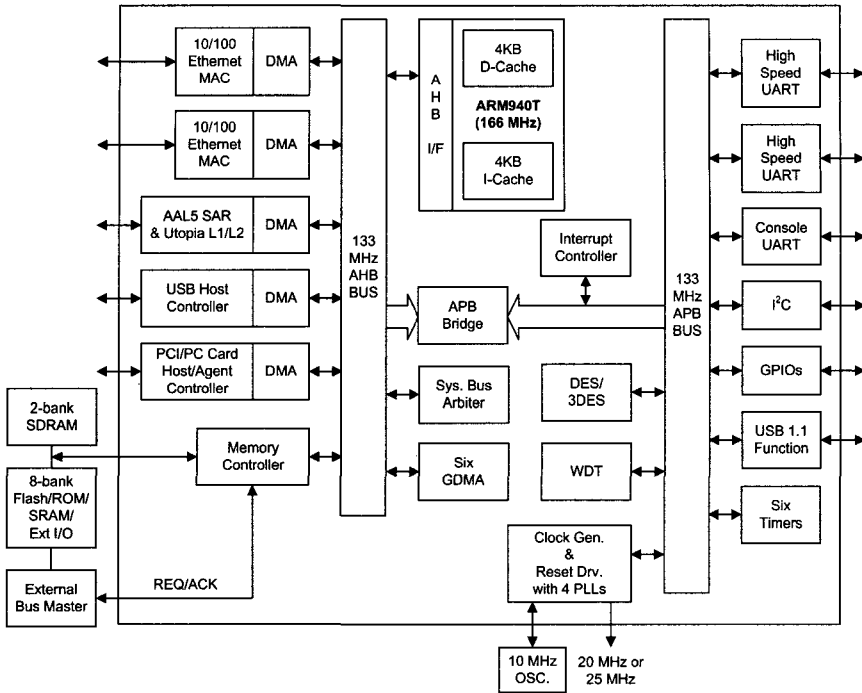


그림 2 목표 네트워크 SOC의 블록도

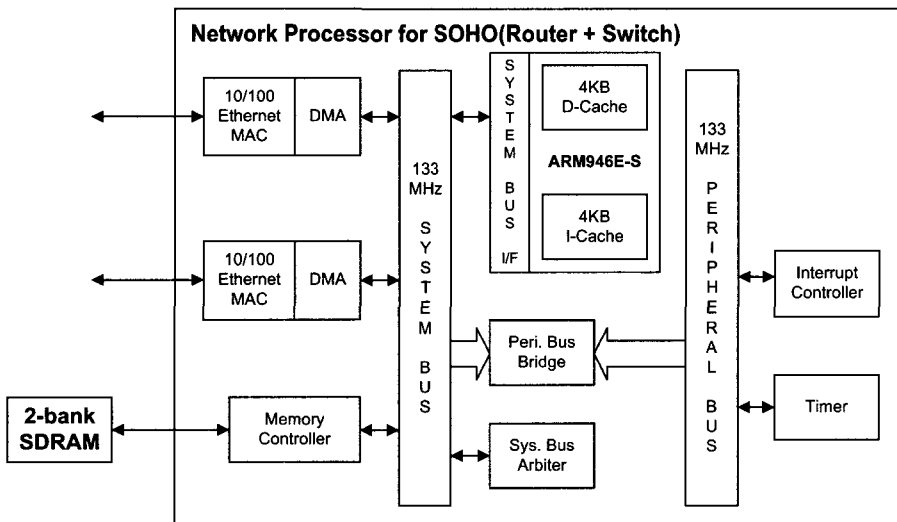
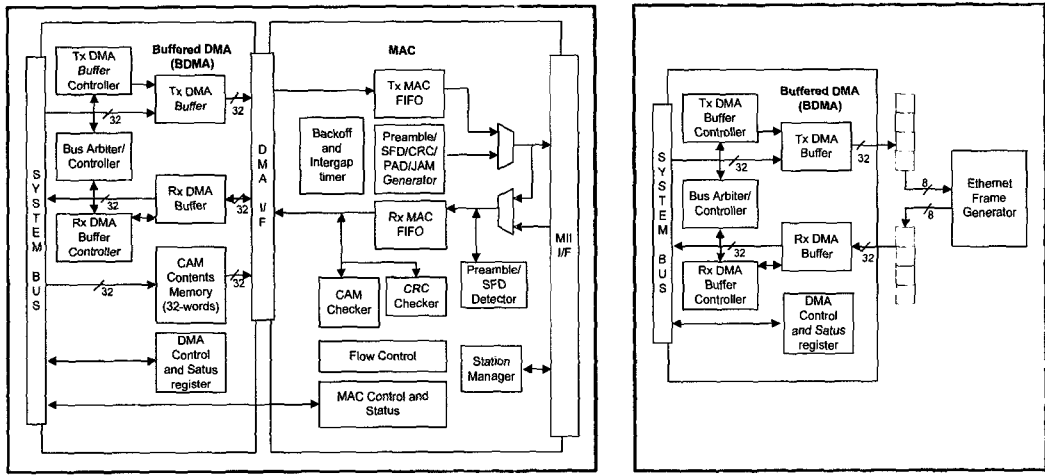


그림 3 NAT 성능 시뮬레이션을 위한 목표 네트워크 SOC의 SystemC 모델

해 단순화된 환경이다. 두 채널의 고속 이더넷 전용 비퍼형 DMA(BDMA)와 메모리 컨트롤러, 버스 중재기 (bus arbiter), 인터럽트 제어기, 타이머 등이 SystemC 로 모델링 되었다. ARM 코어와 AMBA[6] 버스는 Synopsys사에서 제공하는 DesignWare SystemC 라이

브러리를 사용하였고, 모델링과 구조 탐색을 위해 CoCentric System Studio[2]가 사용되었다.

SystemC 모델링 환경은 RTL 환경과 비교시 ARM 모델에서 차이가 있다. RTL 환경은 ARM940T를 사용하는 반면 SystemC 환경에서는 ARM946E-S를 사용



(a) 고속 이더넷 제어기의 하드웨어 구조 (b) 고속 이더넷 제어기의 SystemC 모델  
 그림 4 고속 이더넷 제어기의 SystemC 모델을 이용한 단순화

하였다. ARM946E-S는 I/D(Instruction/Data) 캐쉬 크기를 1Mbytes까지 증가시킬 수 있고, TCM(Tightly Coupled Memory)을 사용할 수 있으며, 버스가 INCR8 (incremental 8) 버스트로 동작한다. ARM946E-S 모델을 ARM940T와 가능한 동일하게 하기 위해, I/D 캐쉬 크기를 4K/4K를 사용하였고, TCM을 사용하지 않았다. 비록 AMBA AHB[6] 버스에서 ARM946E-S의 INCR8과 ARM940T의 INCR4 버스트 동작에는 약 28.1~34.6%의 사이클 단위 차이<sup>1)</sup>가 존재한다. 그러나, 시스템 버스 증대기 규격상 ARM 코어는 시스템 버스를 최대 50%까지만 점유할 수 있고, INCR8이 ARM 코어와 메모리 사이에서만 사용되며, 20% 이상의 버스 자원이 최대 속도의 입력 트래픽에 대해서도 사용되지 않기 때문에 BDMA의 동작이 ARM 코어의 버스 점유율에 많은 영향을 받지 않을 것으로 예측되었다. 따라서, BDMA의 이더넷 프레임 입출력시 동작이 Verilog RTL과 비교시 10% 이하의 사이클 단위 오차를 갖도록 모델링될 수 있어서, 본 논문에서는 SystemC 환경의 시스템 레벨 성능 지표인 NAT 변환율의 목표 오차를 실제 하드웨어 대비 +10% 이내로 설정하였다.

3.2 SystemC를 이용한 고속 이더넷 제어기의 모델링

모델의 시뮬레이션 속도를 증가시키고 정확도를 높이기 위해, NAT 응용시 사용되지 않는 제어기들은 SystemC 환경에서는 제거되었다. 그림 4(a)의 고속 이더넷

제어기는 MAC과 BDMA로 구성되는데 그림 4(b)와 같이 단순한 형태의 BDMA로 모델링되었다.

고속 이더넷 MAC은 단순히 수신된 이더넷 프레임의 경계를 구분하고, 프레임의 에러를 확인 후 보고하는 기능과 전송 매체로의 접근을 제어하는 기능을 수행한다. 따라서, 수신된 이더넷 프레임에 에러가 없고, 고속 전송이 가능한 전-이중(full-duplex) 모드에서 동작한다고 가정하면 매체 접근 제어가 필요하지 않으므로, 시스템 성능에 영향을 주지 않으면서 고속 이더넷 MAC을 외부의 이더넷 프레임 생성기로 대체가 가능하다.

BDMA와 다른 버스 슬레이브(slave)들의 시스템 버스 인터페이스는 실제 하드웨어와 사이클 단위의 정확도를 갖도록 모델링되었다. BDMA 모델에서는 전송, 수신, 버스 동작을 위해 두 개의 FSM(Finite State Ma-

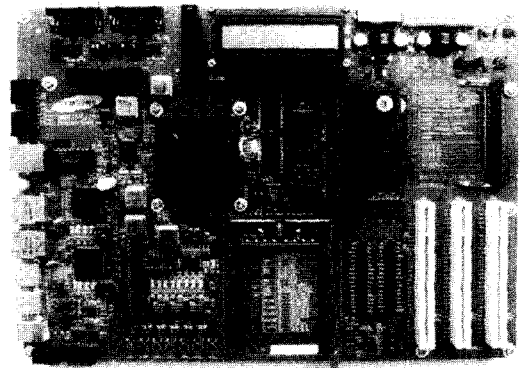


그림 5 목표 네트워크 SOC의 검증 보드

1) INCR8과 INCR4에 의한 8-word read 사이클 수는 page hit시 각각 17과 26이고, page miss시 23과 32이다.

chine)이 사용되었다. 또한, 실제 응용 보드 상에서 사용되었던 uCLinux[7] 커널과 응용 프로그램들을 동일하게 SystemC 환경에서도 사용하기 위해 장치 구동 드라이버(device driver)에서 사용되는 제어와 상태 레지스터들이 모델에서도 구현되었다.

BDMA의 모든 기능 블록들은 SystemC를 이용하여 TLM(Transaction Level Modeling) 방식으로 모델링되었다. RTL로 설계된 시스템은 핀-레벨(pin-level) 동작을 모델링시 고려해 주어야 하지만, TLM 방식은 상위 레벨에서 시스템의 동작을 기술하므로 모델링이 간단하고 시뮬레이션 속도가 빠르다. 예를 들어, TLM 방식으로 작성된 BDMA 모델은 이더넷 프레임 생성기를 포함하여 2,341 라인의 SystemC 코드로 구성되는데 비해 실제 RTL은 17,948 라인의 Verilog 코드로 구성된다. 또한, TLM 방식의 모델링시 표 3에서와 같이 100 배 이상의 시뮬레이션 속도의 증가를 확인할 수 있는데, RTOS 상에서 NAT 라우팅 응용을 실행할 경우 SystemC 기반의 BDMA 모델은 4시간 이내의 시간이 소요되나, Verilog RTL 시뮬레이션은 약 22일의 수행 시간이 예상된다.

**3.3 고속 이더넷 제어기의 BDMA 모델 검증**

검증은 구현 모델의 동작이 실제 시스템과 동일한지 확인하는 과정으로, 모델의 동작을 실제 시스템과 비교하고 그 차이점을 보완하는 그림 6과 같은 반복적인 과정을 통해 검증이 진행된다.

BDMA의 모델 검증을 위해서 세 가지 검증 단계가 사용되었다. 첫째, 목표 네트워크 SOC의 기능 검증을 위한 브리지 테스트(bridge test)가 수행되었다. 브리지 테스트는 네트워크 SOC 내부의 라우팅 경로에 존재하는

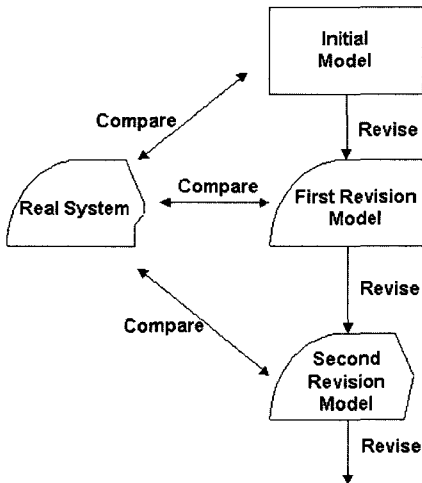


그림 6 BDMA 모델의 설계 과정

BDMA, 메모리, CPU 등이 바르게 동작하고 한 채널로 입력된 이더넷 프레임이 다른 채널로 정확하게 출력되는지를 확인하는 하드웨어 테스트이다.

둘째, 모델의 사이클 단위 정확도가 이더넷 제어기의 Verilog RTL과 비교되었다. 시스템 버스 인터페이스 신호들은 트랜잭션 레벨에서 BDMA를 통한 처리 지연이 이더넷 프레임의 수신 시점과 송신 시점 각각에 대해 첫 번째 시스템 버스 트랜잭션 시점과의 지연을 측정함으로써 비교되었다. 표 1은 모델의 사이클 단위의 정확도를 실제 Verilog RTL과 비교한 결과인데, BDMA 송신 지연을 제외하고는 10% 이내의 오차를 보임을 알 수 있다.

표 1 사이클 단위 정확도 비교

	RTL	Model	Error
DMA Rx Request Latency	693	675	2.6%
DMA Tx Latency	75	67	10.7%
Burst Write Transaction	11	11	0.0%
Burst Read Transaction	19	19	0.0%
Single Write Transaction	8	8	0.0%
Single Read Transaction	16	16	0.0%

표 2 NAT 변환율 비교

Frame(byte)	Eval. Board(Mbps)	Model(Mbps)	Error
1514	58.60	62.20	6.14%
256	10.50	11.08	5.52%
64	2.68	2.95	10.07%

이들 결과로부터, 시스템 버스에 가해지는 트랜잭션들은 모델과 실제 하드웨어가 거의 동일함을 알 수 있다. 이는 네트워크 SOC를 모델링한 SystemC 환경의 성능 데이터가 실제 네트워크 SOC의 검증 보드 상에서 측정된 결과와 매우 정확하게 일치할 것임을 의미한다. 네트워크 SOC 내부의 모든 동작은 시스템 버스를 경유하여 이루어지기 때문에 버스 트랜잭션을 기준으로 한 BDMA 동작의 사이클 단위 정확도는 전체 시스템 성능을 결정하는 매우 중요한 기준이 될 수 있다.

마지막으로 시스템 레벨의 정확도인 NAT 변환율이 비교되었다. 측정을 위해 uCLinux 커널을 SystemC 모델과 목표 네트워크 SOC의 검증 보드에 동일하게 탑재시켰다. NAT 변환율은 표 2와 같이 이더넷 프레임 크기와 프레임 간의 간격을 가변시키면서 다양한 입력 비트율에 대해 측정되었다. 이더넷 프레임 크기는 일반적으로 64~1514 바이트의 길이를 갖는데, 동일한 수신 비트율에 대해 프레임의 크기가 작을수록 단위 시간 동안 라우팅 횟수가 증가한다. 따라서, 작은 프레임들에 대해서는 보다 빈번한 인터럽트 서비스 프로그램 호출과 이

에 따른 ARM 코어에 의한 버스 트랜잭션 양이 증가한다.

시뮬레이션 결과로부터, SystemC 모델의 NAT 변환율 오차는 실제 시스템과 비교시 10% 이내임을 알 수 있고, 시뮬레이션 속도 역시 표 3에서와 같이 Verilog RTL에 비해 100배 이상 빨라짐을 알 수 있다.

표 3 시뮬레이션 속도 비교

Model	Verilog RTL	Ratio
15,000 cycle/sec	112 cycle/sec	134

3.4 검증된 BDMA 모델의 응용

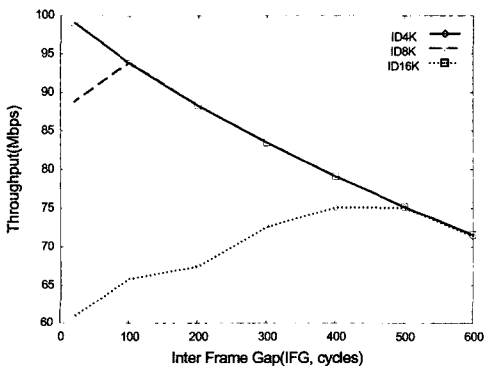
NAT 성능 최대화를 위한 최적의 네트워크 SOC 구조 탐색 대부분의 NAT 변환을 저하는 시스템 버스 자원의 일시적인 감소에 따라 BDMA 버퍼의 넘침이나 고갈에 의해 발생한다. NAT 변환을 최대화를 위해서는 하드웨어와 소프트웨어 모두 시스템 버스 자원을 효과적으로 사용할 수 있도록 설계되어야 한다. 검증된 SystemC 모델들은 최적의 하드웨어 계수들을 결정하고 이에 따른 NAT 변환율 성능을 예측하는데 사용될 수 있다. 본 구조 탐색에서는 SystemC 모델과 실제 네트워크 SOC에 대해 동일한 RTOS와 장치 구동 드라이버를 사용함으로써 소프트웨어의 최적화에 따른 성능 개선의 영향은 배제하였다.

주어진 네트워크 프로세서 환경에서 NAT 성능에 영향을 주는 하드웨어 변수를 찾기 위해 다음과 같은 방식으로 구조 탐색을 수행하였다. 첫째, I/D 캐쉬 크기를 변화시키면서 NAT 변환율을 측정하였고, 동시에 시스템 내부의 병목 지점을 파악하기 위해 버스 점유율과 버스 사용 시점 중첩(contention)을 분석하였다. 둘째, BDMA 버퍼 크기를 변화시키면서 버퍼의 넘침과 고갈

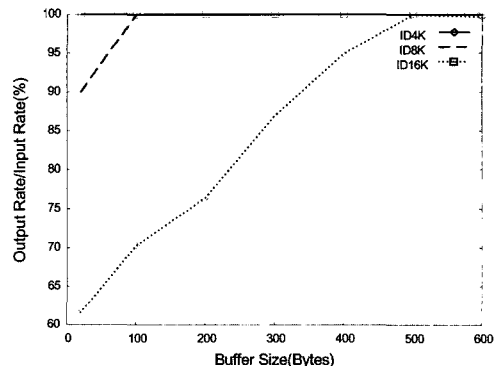
이 성능에 미치는 영향을 실험하였다. 마지막으로 시스템 버스와 CPU에 대해 클럭 동기화 모드의 성능을 실험하였다. 클럭 동기화 모드에서 시스템 버스의 클럭 속도는 네트워크 프로세서에 내장된 주변 장치들의 동작 속도에 의해 대부분 결정되고 고정되기 때문에, CPU 클럭 속도를 증가시키면서 실험을 수행하였다.

그림 7은 캐쉬 크기의 NAT 변환율에 대한 영향을 나타낸다. ARM946E-S[6] 코어는 I/D 캐쉬 크기를 4kbytes에서 1Mbytes까지 가변시킬 수 있는데, 캐쉬 크기를 8k/8k 이상으로 변화시키면 최대의 NAT 변환율이 도달 가능하다. 4k/4k 크기에서는 데이터 캐쉬의 읽기 실패 비율(read miss ratio)이 약 9.88%이고, 쓰기 실패 비율(write miss ratio)은 약 14.6%이었다. 8k/8k 크기에서 데이터 캐쉬의 읽기 실패 비율은 3.4%, 쓰기 실패 비율은 3.93%로 감소하였다. I/D 캐쉬 크기가 각각 4k/4k와 8k/8k인 경우에 대해 버스 점유율을 비교하면, CPU(write)와 BDMA1의 점유율에 큰 차이를 보인다. 그림 8에서 BDMA0은 메모리에 데이터를 전송하고, BDMA1은 메모리로부터 해당 데이터를 수신한다. 그림 9에서 캐쉬 크기의 증가로 CPU(write) 동작이 증가되었고, BDMA1이 보다 많은 데이터를 수신하며, 그 결과 NAT 변환율이 증가하였음을 알 수 있다. 이는 캐쉬 크기를 증가시킴에 따라 CPU와 BDMA들에 의한 버스 사용 시점 중첩(bus contention)이 상당히 감소하였음을 의미한다.

그림 10은 고정된 4k/4k의 캐쉬 크기에 대해 BDMA 버퍼 크기에 따른 NAT 변환율의 변화를 보여준다. BDMA 버퍼는 일시적인 버스 자원 고갈을 대비하여 데이터를 임시로 저장하는 역할을 한다. 그러나, 큰 버퍼가 버스 대역폭의 증가를 의미하지는 않기 때문에, 수신률이나 송신률에 비해 버스 대역폭이 작은 상황이 일



(a) Throughput (Mbps)



(b) Input vs. Output Ratio (%)

그림 7 I/D 캐쉬 크기의 NAT 변환율에 대한 영향

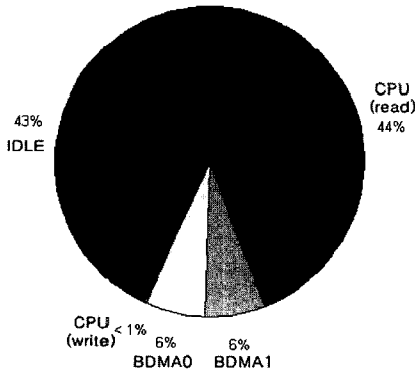


그림 8 버스 점유율(4k/4k I/D캐쉬)

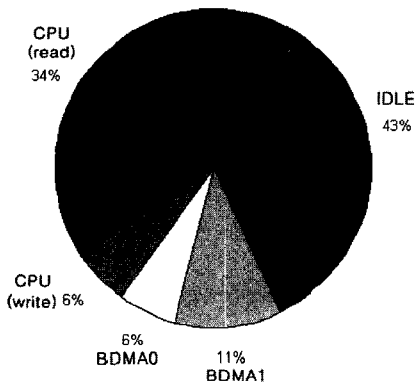


그림 9 버스 점유율(8k/8k I/D캐쉬)

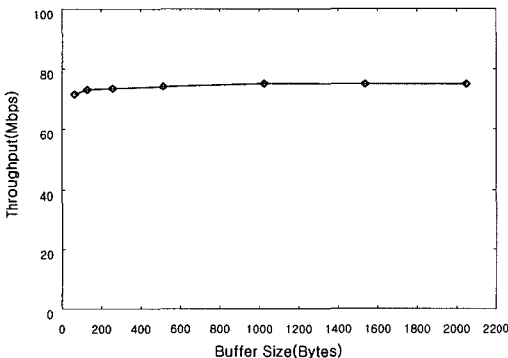


그림 10 BDMA 버퍼 크기의 NAT 변환율에 대한 영향 (프레임간 간격=400 cycles)

표 4 시스템 버스에 동기된 CPU 클럭 속도별 NAT 변환율(Mbps)

I/D Cache	Bus/CPU Clock Frequency(MHz)				
	133/133	133/266	133/399	133/532	Target
4k/4k	58	62	66	69	98
8k/8k	98	98	98	98	98

정 시간 이상 계속되면 버퍼는 넘침이나 고갈에 의해 수신 프레임의 삭제나 송신 프레임의 오류를 유발한다. 따라서, BDMA 버퍼 크기의 NAT 성능에 대한 영향은 I/D 캐쉬 크기가 미치는 영향보다 훨씬 적다. 시뮬레이션 결과로부터 BDMA 버퍼 크기는 512bytes 이상이 일시적인 버스 자원 고갈에 따른 데이터를 흡수하기에 충분함을 알 수 있다.

표 4는 시스템 버스에 동기된 CPU 클럭 속도가 NAT 성능에 미치는 영향을 나타낸다. 시스템 버스 클럭 속도는 133MHz로 고정하였고, CPU 클럭 속도를 버스 클럭 속도의 네 배까지 증가시키면서 NAT 성능을 측정하였다. 캐쉬 크기가 4k/4k인 경우 CPU 클럭 속도의 증가는 NAT 변환율을 다소 증가시키나 큰 영향을 주지는 않는다. 캐쉬 크기가 8k/8k인 경우 NAT 성능은 CPU 클럭 속도에 상관없이 최대 속도인 98Mbps에 도달함을 확인할 수 있다.

본 절에서는 검증된 BDMA 모델을 이용하여 네트워크 프로세서의 구조 탐색을 수행하였다. I/D 캐쉬 크기는 NAT 성능에 가장 큰 영향을 미치는 하드웨어 계수로써, 네트워크 프로세서 성능 예측을 위해 BDMA의 버퍼 크기나 CPU 클럭 속도 등보다 우선하여 고려되어야 할 것이다. 본 논문에서 검증된 BDMA 모델과 구조 탐색 환경은 전용 DMA를 갖는 기타 네트워크 제어기나 범용 DMA의 동작 규격에 맞도록 수정된 후 이들을 내장한 네트워크 프로세서의 구조 탐색에 쉽게 적용 가능하다.

#### 4. 결론

본 논문에서 네트워크 SOC의 성능 예측을 위해 SystemC를 이용하여 네트워크 SOC에 내장된 고속 이더넷 제어기와 시스템 모듈들의 모델을 설계하고 검증하였다. 설계된 모델은 기능 검증, 사이클 기반 정확도 검증, 시스템 관점의 NAT 성능이 실제 Verilog RTL과 네트워크 SOC와 비교되었다. SystemC 환경의 NAT 변환율은 실제 시스템과 비교시 ±10% 이내의 성능 오차를 보였으며, 시뮬레이션 속도는 RTL 시뮬레이션에 비해 100 배 이상의 이득을 얻을 수 있었다. 구현된 SystemC 모델의 단순화와 일부 시스템 모듈들의 트랜잭션 레벨 모델링을 통해 추가의 시뮬레이션 속도 향상이 기대된다.

본 논문에서 구현된 BDMA 모델과 SystemC 환경은 네트워크 제어기용 DMA를 내장하는 향후 네트워크 SOC 설계시 성능 예측 및 최적의 하드웨어 구조 결정을 위해 이용될 수 있다. 또한, 성능 예측 실패에 따른 불필요한 제품의 수정을 줄이고, RTOS 상의 장치 드라이버의 선행 개발 등에 응용 될 수 있어서 SOC 개발 기간을 단축하고, 개발 비용을 감축할 수 있을 것으로 기대된다.

## 참 고 문 헌

- [1] B. Bailey, R. Klein, and S. Leef, Hardware/Software co-simulation strategies for the future, <http://www.mentor.com/>
- [2] CoCentric System Studio Data Sheet, Synopsys, <http://www.synopsys.com/>, 2002.
- [3] Open SystemC Initiative, <http://www.systemc.org>, 2001.
- [4] K.Egcvang, P.Francis, The IP Network Address Translator(NAT), Internet RFC 1631, May 1994.
- [5] S3C2510A, 32-bit RISC Microprocessor, <http://www.samsung.com/>, 2003.
- [6] <http://www.arm.com>, 2003.
- [7] Embedded Linux/Microcontroller Project, <http://www.uclinux.org>, 2003.



이 명 진

1990년 3월~1994년 2월 한국과학기술원 전기및전자공학과 학사. 1994년 3월~1996년 2월 한국과학기술원 전기및전자공학과 석사. 1996년 3월~2001년 8월 한국과학기술원 전자전산학과 박사. 2001년 3월~2004년 2월 삼성전자 반도체 총

괄 책임. 2004년 3월~현재 경성대학교 전기전자공학 전임 강사. 관심분야는 영상 압축/신호처리, 멀티미디어 통신, 인터넷 QoS, 네트워크 하드웨어 설계