

ATM 망에서의 감시 알고리즘 구현

학생회원 이요섭*, 권재우*, 이상길*, 정회원 최명렬*

Implementation of Policing Algorithm in ATM network

Yo-Seop Lee*, Jae-Woo Kwon*, Sang-Kil Lee* *Student Members*

Myung-Ryul Choi* *Regular Member*

요 약

본 논문에서는 ATM 망의 트래픽 제어 기능 중의 하나인 감시 알고리즘을 제안하고 칩(chip)을 설계하였으며, VHDL을 사용하여 구현하였다. 제안한 알고리즘은 우선순위가 높은 셀의 손실을 최소화 하고, 트래픽의 다중화 및 역다중화 과정에서 발생하는 트래픽의 군집성을 해소할 수 있다는 장점을 갖고 있다. 구현한 칩은 입력 모듈과 UPC 모듈, 출력 모듈의 3부분으로 이루어지며, 편의상, UPC 모듈에서는 메모리 제어시 메모리 어드레스 즉, 커넥션 테이블의 index를 VCI(Virtual Channel Identifier)와 동일하게 할당하였다. 또한 UPC 모듈의 VSA(Virtual Scheduling Algorithm)블럭에서 셀의 도착시간을 계수하는 카운터가 랩-어라운드(wrap-around)할 때 생기는 VSA의 오류를 보정해 줄 수 있는 방법을 제안하였다.

ABSTRACT

In this thesis, a policing algorithm is proposed, which is one of the traffic management function in ATM networks. The proposed algorithm minimizes CLR(Cell Loss Ratio) of high priority cells and solves burstiness problem of the traffic caused by multiplexing and demultiplexing process.

The proposed algorithm has been implemented with VHDL and is divided into three parts, which are an input module, an UPC module, and an output module. In implementation of the UPC module's memory access, memory address is assigned according to VCI's LSB(Lowest Significant Byte) of ATM header for convenience. And the error of VSA operation from counter's wrap-around can be recovered by the proposed method. ANAM library 0.25 μm and design compiler of Synopsys are used for synthesis of the algorithm and Synopsys VSS tool is used for VHDL simulation of it

1. 서 론

ATM은 다양한 트래픽 종류들을 수용하면서 사용자와 망이 요구하는 각종 성능을 만족시켜야 하며 이를 위하여 트래픽 제어 능력을 제공해야 한다. 트래픽 제어는 트래픽 체증 제어로서 체증이 일어난 후에 처리하는 대응 제어보다는 체증이 일어나지 않게 미리 예방하는 예방 제어가 더 바람직하다.

ITU-T의 권고안 I.371에는 예방적 트래픽 제어 방법으로 호 수락 제어, 사용 변수 제어(UPC : Usage Parameter Control), 자원 관리, 우선순위 제어 및 트래픽 정형화 등을 정의하였으며, 대응적 트래픽 제어 방법으로는 선택적 셀 폐기 및 명시적 전방향 폭주 표기(EFCI : Explicit Forward Congestion Indication)등을 정의하고 있다. 예방적 제어가 추구하는 기본 목표는 QoS(Quality of

* 한양대학교 전자전기 제어계측공학과 ASIC연구실 최명렬(choimr@asic.hanyang.ac.kr), 이상길(sklee@asic.hanyang.ac.kr),

권재우(kirhoff@asic.hanyang.ac.kr), 이요섭(remnant@asic.hanyang.ac.kr)

논문번호 : 010184-0713, 접수일자 : 2001년 7월 13일

※ 본 연구는 한양대학교 교내 연구비의 지원을 받았습니다.

Service)의 향상으로서 연결 수락 제어와 사용 변수 제어가 있다. 사용 변수 제어는 망이 사용자의 트래픽량과 셀 경로 유효성의 측면에서 감시하고 조치하는 것으로서 사용자의 트래픽 변수가 호 설정 시 협정했던 대로 잘 지켜지고 있는지의 여부를 감시하여 상응하는 조치를 취하는 것이다^[1,2]. 이와 같이 연결 수락 제어 시 협정된 변수가 호 진행 중에 잘 지켜지고 있는지를 항상 감시(policing)하는 기능이 사용 변수 제어이다^[3].

본 논문에서는 셀 손실 우선 순위를 고려하여 우선 순위가 높은 셀(cell loss priority 비트가 0인 셀)의 손실을 최소화하는 개선된 UPC 알고리즘^[4]을 제안하고 그것을 VHDL로 구현하였다. 본 논문의 구성은 다음과 같다. 2장에서는 ITU-T의 권고안에서 소개한 기존의 UPC 알고리즘에 관해 논한다. 3장에서는 본 논문에서 제안한 알고리즘에 대해 기술하고, 4장에서는 제안한 알고리즘을 3개의 모듈로 나누어 칩으로 설계하여 VHDL로 구현한 결과들을 나타내었다. 마지막으로 5장에서는 결론을 맺고 향후 연구과제에 관해 논하였다.

II 기존의 감시(policing) 알고리즘

UPC를 구현하기 위해 가장 기본적으로 요구되어지는 기능은 셀의 적합성(conformance) 여부를 판단하는 것이다. 셀의 적합성을 시험하는 방법으로 가장 많이 사용되어지는 것이 ITU-T의 권고안 I.371에 정의되어 있는 GCRA(Generic Cell Rate Algorithm)이다. GCRA의 구체적인 방법으로 VSA(Virtual Scheduling Algorithm), CS-LBA(Continuous State Leaky Bucket Algorithm) 등이 있으나 본 논문에서 제안한 UPC 알고리즘은 셀의 적합성 시험을 위해 VSA^[5]를 사용하였다.

ATM 셀의 헤더(header) 내에 있는 CLP 비트는 송신 측에서 셀을 전송할 때에 부여하는 것으로 CLP = 0 셀은 ATM 망 내에서 손실되지 않기를 희망하는 우선순위가 높은 셀을 의미한다. 한편 CLP = 1 셀은 우선순위가 낮은 셀을 의미한다. ITU-T(International Telecommunications Union-Telecommunication) 권고안 I.371에서 정의한 UPC는 태깅을 선택 사항(option)으로 두었으며, 이것을 그림 1에 도시하였다. 그림 2은 태깅을 허용하는 경우와 태깅을 사용하지 않는 경우를 모두 나타낸 것이다.

그림 1에서 사용한 기호의 정의는 다음과 같다.

- T_0 : CLP = 0 셀에 대한 셀 도착 간격, VSA에서 I와 동일.
- T_{0+1} : CLP = 0 셀과 CLP = 1 셀의 합에 대한 셀 도착 간격.
- VSA(x, y) : 셀 도착 간격 x, 허용 오차 y로 VSA 수행.
- τ_0 : CLP = 0 셀에 대한 허용오차, VSA에서 τ 와 동일.
- τ_{0+1} : CLP = 0 셀과 CLP = 1 셀의 합에 대한 허용오차.

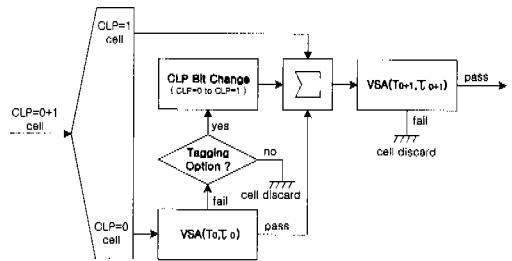


그림 1. ITU-T 권고안의 UPC 알고리즘

태깅을 허용하는 UPC 알고리즘은 CLP = 0 셀에 대해서 VSA(T_0, τ_0)와 VSA(T_{0+1}, τ_{0+1})을 적용하여 셀 손실률을 최소로 하려고 하였다. 그러나 VSA(T_{0+1}, τ_{0+1})을 수행하는 시점에 따라 CLP = 0 셀의 손실률이 차이가 날 수 있다. 즉 VSA(T_{0+1}, τ_{0+1})을 수행하는 과정에서는 CLP = 0 셀과 CLP = 1 셀에 대한 구분이 없기 때문에 VSA(T_0, τ_0)을 통과한 적합한 CLP = 0 셀이 VSA(T_{0+1}, τ_{0+1})을 수행하는 과정에서 적합치 못한 셀로 판단되어 폐기되는 경우가 발생한다.

그림 2와 같이 셀이 유입되는 경우에 CLP = 0 셀들은 VSA(T_0, τ_0)에 의해 모두 적합한 셀로 판단되었으나 VSA(T_{0+1}, τ_{0+1})에서는 적합치 않은 셀로 판단되어 모두 폐기되어진다. 즉 ITU-T UPC 알고리즘은 UPC를 시작하는 위치에 따라 CLP = 0 셀에 대한 손실률이 높게 나타나는 경우가 발생하게 된다

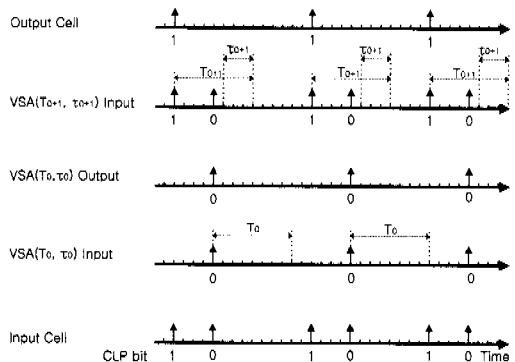


그림 2. ITU-T 사용 변수 제어 알고리즘의 문제점 ($T_0 = T_{0+1} = 8$ cell slots, $\tau_0 = \tau_{0+1} = 3$ cell slots의 경우)

III 제한한 감시(policing) 알고리즘

그림 3에 본 논문에서 구현한 셀 손실 우선순위를 고려한 UPC 알고리즘을 나타냈다. 구현한 알고리즘은 손실 우선 순위가 낮은 $CLP = 0$ 셀의 손실을 최소화 하기 위해 UPC 출력단에 1개의 셀을 저장할 수 있는 버퍼를 사용하였으며, $VSA(T_0, \tau_0)$ 을 통과한 $CLP = 0$ 셀이 $VSA(T_{0+1}, \tau_{0+1})$ 을 거치면서 적합치 못한 셀로 판단된 경우에 해당 셀을 버퍼에 저장하고, $VSA(T_{0+1}, \tau_{0+1})$ 을 만족할 수 있도록 일정 시간을 지연한 후에 전송하는 것이다.

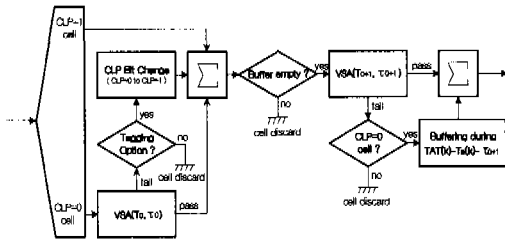


그림 3. 셀 손실 우선순위 기반의 UPC 알고리즘

그림 4에 구현한 UPC 알고리즘을 적용했을 경우의 예를 보였다. 이것은 그림 2와 동일한 조건 즉 $CLP = 0$ 셀에 대해서 최악의 조건을 가정한 것이다. $CLP = 1$ 셀과 $CLP = 0$ 셀이 모두 통과하였고, 트래픽의 균집성이 향상되었다. $CLP = 0$ 셀 중에서 $VSA(T_{0+1}, \tau_{0+1})$ 를 만족하지 못한 셀은 $TAT(k) - Ta(k) - \tau_{0+1}$ 시간 동안 버퍼에 저장된 후 전송된다. 이러한 시간 지연은 입력 트래픽의 균집성을 해소시키는 이점이 있다.

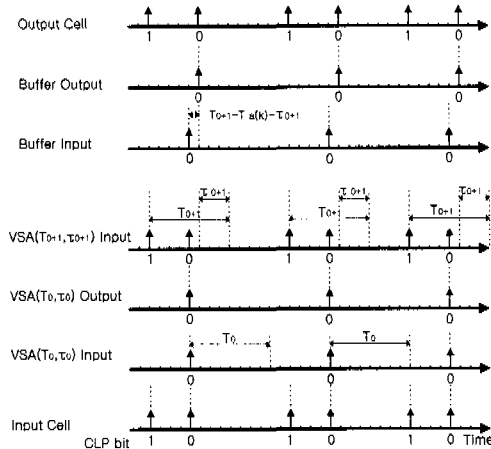


그림 4. 제한한 UPC 알고리즘의 예($T_0=T_{0+1}=8$ cell slots, $\tau_0=\tau_{0+1}=3$ cell slots의 경우)

IV 제안된 알고리즘의 VHDL 설계 및 구현

1 구현 환경 및 전체 구조

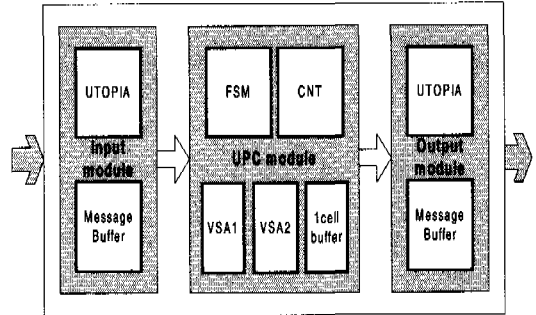


그림 5. 구현한 칩의 전체 구조

제한한 알고리즘의 합성을 위해서 아남 $0.25 \mu m$ 라이브러리(library)를 이용하였으며 VHDL을 사용하였다^[7,8]. VHDL 해석기로는 Synopsys사의 VHDL 어널라이저(analyzer)를 이용하였고, VHDL 시뮬레이터(simulator)로는 VHDL 디버저(debugger)를 사용하였다. 마지막으로 디자인 어널라이저(design analyzer)를 이용한 회로 합성을 통하여 실제 칩으로 제작될 회로를 얻었다.

그림 5에 구현한 칩의 전체 구조를 도시하였다. 그림에서 볼 수 있듯이 구현한 칩은 크게 입력 모듈, UPC 모듈, 출력 모듈의 3부분으로 나누어진다.

2. 입력모듈의 설계 및 구현

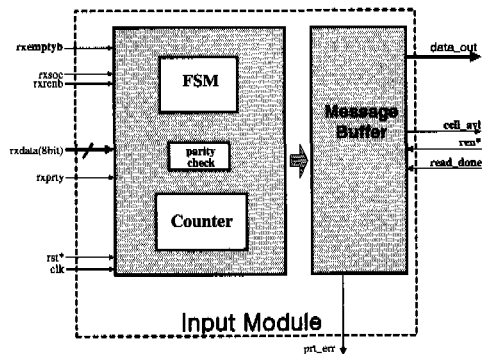


그림 6. 입력 모듈의 설계

그림 6에서 입력 모듈에 대한 블록도를 나타내었다. 입력 모듈은 물리적인 계층과 구현한 UPC칩과의 인터페이스를 담당하는 UTOPIA(Universal Test & Operations PHY interface for ATM) Rx (Receiver) 블록과 한 개의 셀을 담을 수 있는 FIFO 3개로 이루어져 있다. 본 논문에서는

UTOPIA 블록을 Specification Level 1, Version 2.01을 바탕으로 구현했다.

3. UPC 모듈의 설계 및 구현

그림 7에 설계한 UPC 모듈을 도시하였다. UPC 모듈은 크게 3가지 기능을 가지고 있다. 주요 3가지 기능은 ATM 헤더의 VCI 값을 어드레싱하므로써 메모리를 액세스하는 것과 VSA를 실행하는 기능, 마지막으로 처리가 끝난 셀을 분리되어 있던 헤더와 페이로드를 합쳐서 출력 모듈로 보내는 기능이다. 따라서 UPC 모듈은 필요한 각 기능을 수용하기 위해서 헤더 레지스터와 FSM, VSA 블록과 메모리, 그리고 다중화기(MUX)로 구성된다.

1) FSM 블록의 설계 및 구현

FSM(Finite State Machine) 블록은 그림3에서 보여준 알고리즘을 실행하는 블록이다. 이 블록에서는 먼저 ATM 헤더 내에 있는 CLP 비트 값에 의해 VSA₀ 알고리즘을 실행할 것인지 VSA₀₊₁ 알고리즘을 실행 할 것인지를 결정한다. CLP='0'일 경우에는 VSA₀ 알고리즘이 실행되며, 이때 ATM 헤더 내의 VCI LSB(Lowest Significant Byte)를 참조하여 메모리에 저장되어 있던 필요한 파라미터를 읽어온다. CLP='1'일 경우에는 버퍼가 비었는지를 확인할 수 있는 buffer_empty 신호를 보고서 다음 동작

이 결정된다. VSA₀ 알고리즘 실행 후 셀이 적합하면 CONF1 신호는 '1'이고 적합하지 않으면 '0'이 된다. 셀이 적합하지 않을 경우에 tag_op이 '1'이면 CLP bit을 1로 바꾼 다음, 버퍼를 체크하는 단계로 가지만, tag_op이 '0'이면 해당 셀은 버려진다. 이제 Buffer_empty 신호로 버퍼가 비어 있으면 VSA₀₊₁ 알고리즘이 실행되지만 버퍼에 데이터가 있으면 해당 셀은 버려진다. VSA₀₊₁ 알고리즘 실행 후 셀의 적합 여부에 따라 CONF2 신호가 생성된다. CONF2 신호가 '0'일 경우 즉, 현재 셀이 적합하지 않다고 판정 될 경우 CLP0 값이 얼마인지를 확인하게 된다. CLP0 값이 '1'인 경우는 TAT(k) - Ta(k) - τ₀₊₁ 시간만큼 버퍼링 후 외부로 셀이 전달되며 CLP0 값이 '0'일 경우는 셀은 폐기된다. 반면, CONF2 신호가 '1'이면 해당 셀은 최종적으로 적합한 셀로 판정되고 외부로 전달된다.

2) VSA 블록의 설계 및 구현

VSA 블록이 연산을 수행할 경우 실시간 카운터가 계수하는 수가 유한하다는 문제가 생기게 된다^[6]. 즉, 카운터는 일정 수만큼만 계수하고 램-어라운드하여 '0'으로 돌아올 경우에도 VSA의 연산과정에서는 램-어라운드하지 않고 연속적으로 계수를 해서 나온 수로 연산을 행해야 정확한 결과가 나오는 것이다. 따라서 VSA 블록의 구현시 카운터의 램-어라운드 특성을 고려하는 것은 필수적이다.

3) 램-어라운드 문제 해결 방법

본 논문에서는 카운터의 램-어라운드 특성을 고려한 설계 방법으로 카운터가 램-어라운드 할 때마다 플래그를 셋팅(setting)하여 카운터가 램-어라운드 했음을 표시하고, 카운터가 현재 계수한 수 대신에 현재 계수한 수와 카운터의 최대 계수 값을 합산한 값을 사용하였다.

(가)Ta를 구현 시 하나의 카운터를 사용하면 카운터가 램-어라운드 하는 순간이 같으므로, 모든 커넥션 테이블의 파라미터를 업데이트 해야한다. 이는 한순간에 너무 많은 시간을 요하므로 그 순간에 들어오는 셀을 제때에 처리하지 못하게 된다. (나)반약에 각 커넥션마다 하나씩의 Ta를 나타내는 카운터가 있다고 가정하자. 모두 동기가 되어 있다고 가정하면 이들도 동시에 램-어라운드가 발생하므로 이 경우에도 위와 같은 문제가 발생한다. 이를 변경하여 각 커넥션마다의 카운터가 한 셀 클럭씩 차이가 난다면 전체적으로 봐서 램-어라운드가 발생하는 횟

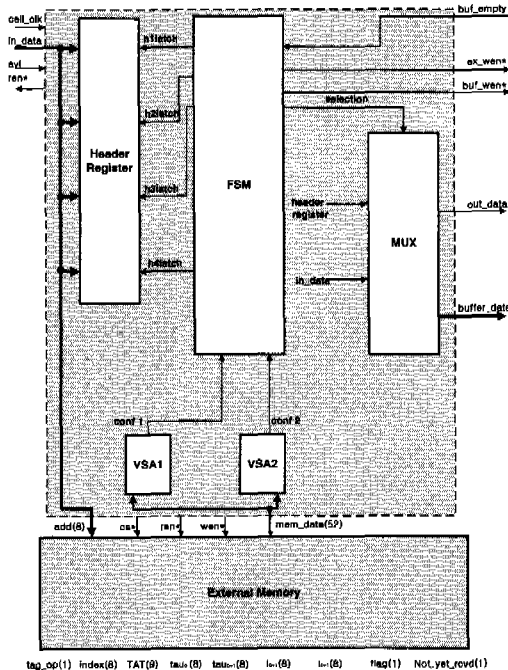


그림 7. UPC 모듈의 설계

수는 똑같지만 동시 발생이 아니라 분산 발생이 된다. 즉, 한순간의 셀 클럭에서 동시에 n개의 커넥션을 업데이트하는 게 아니라 각 셀 클럭마다 하나의 커넥션의 카운터가 램-어라운드 하므로 그 커넥션 하나만을 업데이트하면 된다.

(다) 이를 반대로 해석하면, Ta 카운터는 하나이고, 램-어라운드 정의틀 0번 커넥션은 카운터가 255에서 0이 되는 순간으로 1번 커넥션은 카운터가 254에서 255가 되는 순간으로 하면, 하나의 Ta를 사용하면서 매 클럭마다 단지 하나의 커넥션만 램-어라운드 발생하여 하나의 커넥션 파라미터를 업데이트하면 된다. 이를 위하여 가상시간 클럭(Virtual Time Clock)을 $VTCn = RTC + nth \text{ Connection'Index}$ 로 정의하였다. 즉, 실제시간 클럭(RTC)은 하나이지만, 각 커넥션마다 VTCn을 갖고 VTCn이 255에서 0일 때 램-어라운드 발생하며 커넥션마다 VTC가 다르므로 분산적으로 발생하게 된다.

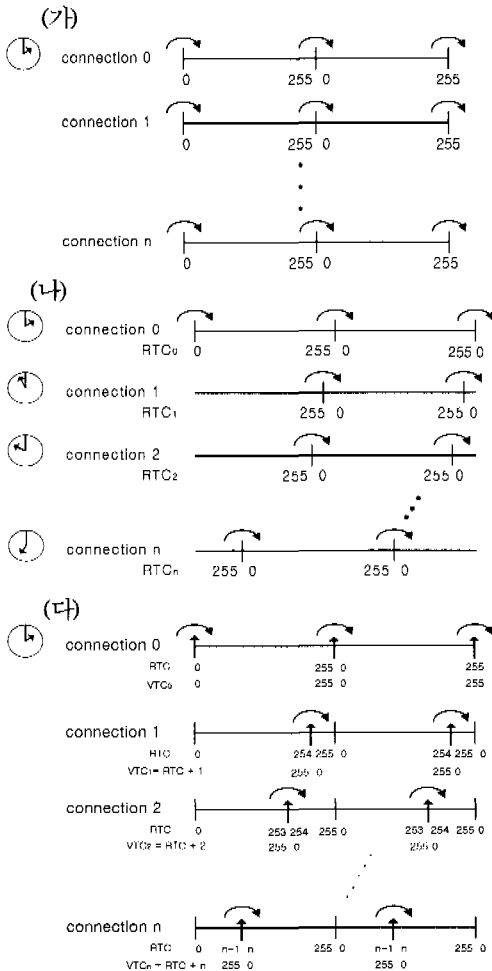
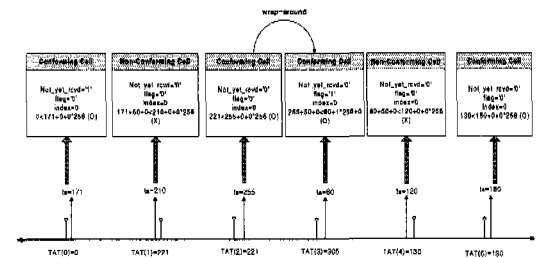


그림 8에서 카운터가 램-어라운드하는 경우를 나타내었다. 램-어라운드를 수용하는 방법을 상술하는데 초점을 맞추기 위해서 실제적인 VSA의 연산에서 r 값을 고려하지 않고 처리하였다. 또한 다중의 가상 연결들이 혼재하는 실제 망에서와는 다르게 단일 연결에 대해서만을 기술하였다. 그림에서 셀이 도착하는 시간인 ta 는 카운터로서 계수 되어지는 값이다. 사용한 카운터는 255를 센 다음 0으로 램-어라운드 된다. 셀이 시간에 따라 VSA에서 어떻게 처리되는 지를 보여주기 위하여 인덱스(index)가 0인 셀 즉, VCI가 0인 셀이 연속적으로 도착한다고 가정하였다. 셀이 도착하는 것을 회색 화살표로 나타내었고 박스의 맨 위에 VSA를 실행한 결과,

셀의 적합성을 판정한 결과를 표시하였다. 아래 박스 안에 카운터의 램-어라운드를 위한 3개의 신호가 변화하는 것을 나타내었으며 그 아랫줄에는 VSA가 수행하는 $TAT < Ta + index + flag * 256$ 의 변화를 보여준다. 가장 처음 들어오는 셀의 경우 $TAT(0)$ 는 0이 된다. VSA로부터 Ta와 비교되는 TAT의 값은 0이 되므로 당연히 $TAT < Ta + index + flag * 256$ 를 만족하게 되어 적합한 셀이 된다. 적합한 셀로 판정되었으므로 그 다음에 도착하는 셀의 TAT(1)의 값은 이전 셀의 $Ta + I + index$ 가 되어 221이 된다. 이에 따라 셀은 부적합 판정을 받고 세 번째 경우에는 이전 셀이 부적합 판정을 받았으므로 TAT의 값은 이전 셀의 TAT의 값과 동일하게 된다. 그런데 $Ta = 255$ 에서 0으로 램-어라운드 하면서 플래그를 '1'로 셋팅한다. 플래그가 셋팅 되면 다음번 셀의 적합성을 판정시 Ta값이 계속해서 증가한 것처럼 256을 더해 주게 된다. 따라서 4번째 경우처럼 $TAT + I$ 와 $Ta + 256$ 이 비교된다. 여기서 Not_yet_rcvd는 카운터에 의해 계수된 Ta가 255에서 0으로 변할 경우 플래그가 '1'로 활성화되는데 처음 카운터가 시작될 경우도 0부터 시작되기 때문에 램-어라운드가 아닌 경우에도 플래그가 '1'로 활성화되는 문제를 해결하기 위해 쓰였다. 여기서 I값은 50이다.



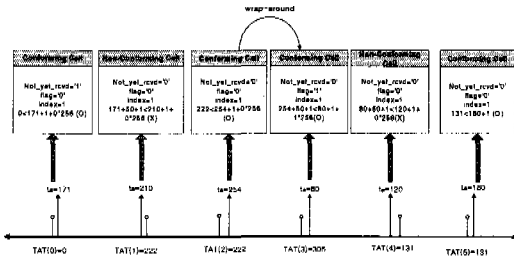


그림 8. VSA 블록에서의 램-어라운드 수용 예.

4) 메모리 블록의 설계 및 구현

표 1은 외부메모리의 구성도를 보여 준다. 램-어라운드 수용을 위한 변수들과 알고리즘 실행 시 필요한 변수들로 구성된다. 이 변수들은 각 커백션마다 또는 VSA 알고리즘 실행결과에 따라 갱신되어진다.

$I_{0\pm}$ 는 커백션을 맺기 전에 CAC(Connection Admission Control)를 수행함으로써 결정되는 변수들이다.

5) UPC 모듈의 구현 결과

그림 9에서는 UPC 블록의 시뮬레이션 결과를 보여주고 있다. CELLCLK에 따라 해당 커백션의 플래그를 활성화해 주기 위해서 외부 메모리를 액세스

표 1. 외부 메모리의 구성(단위 : bit)

	tagop	index	TAT	r ₀	r ₀₊₁	I ₀	I ₀₊₁	flag	Not_yet_rcvd
size(bit)	1	8	10	8	8	8	8	1	1

Tagop : 태깅 옵션(Tagging Option)

Index : 커백션의 순번

TAT : 가상 도착 시간

r₀ : CLP = 0셀에 대한 허용 오차

r₀₊₁ : CLP = 0셀과 CLP = 1셀의 합에 대한 허용오차

I₀ : CLP = 0셀에 대한 셀 도착 간격

I₀₊₁ : CLP = 0셀과 CLP = 1셀의 합에 대한 셀 도착 간격

flag : 카운터의 램-어라운드 발생 유무 표시

Not_yet_rcvd : 아직 처리 해야할 셀이 없음을 나타냄

스하고, AVL 신호로 처리할 셀이 있음을 알 수 있다. 셀을 처리하기 위해서 필요한 CLP 값과 VCI 값은 ATM 헤더에서 가져오고, 프로세스에 따라 VSA 알고리즘을 실행하기 위해 외부메모리를 액세스한다. VSA 실행 결과로 CONF1과 CONF2 신호가 생성된다. CONF2 신호와 CLP0값에 의해 셀이 바로 출력 모듈로 보내지거나 버퍼에 쌓였다가 보내지거나 버려진다. 위 시뮬레이션에서는 VSA 알고리즘 실행 후 CONF1 신호가 '1'이 되고 버퍼가 비어 있으므로 VSA₀₋₁ 알고리즘이 실행된다. 그 결

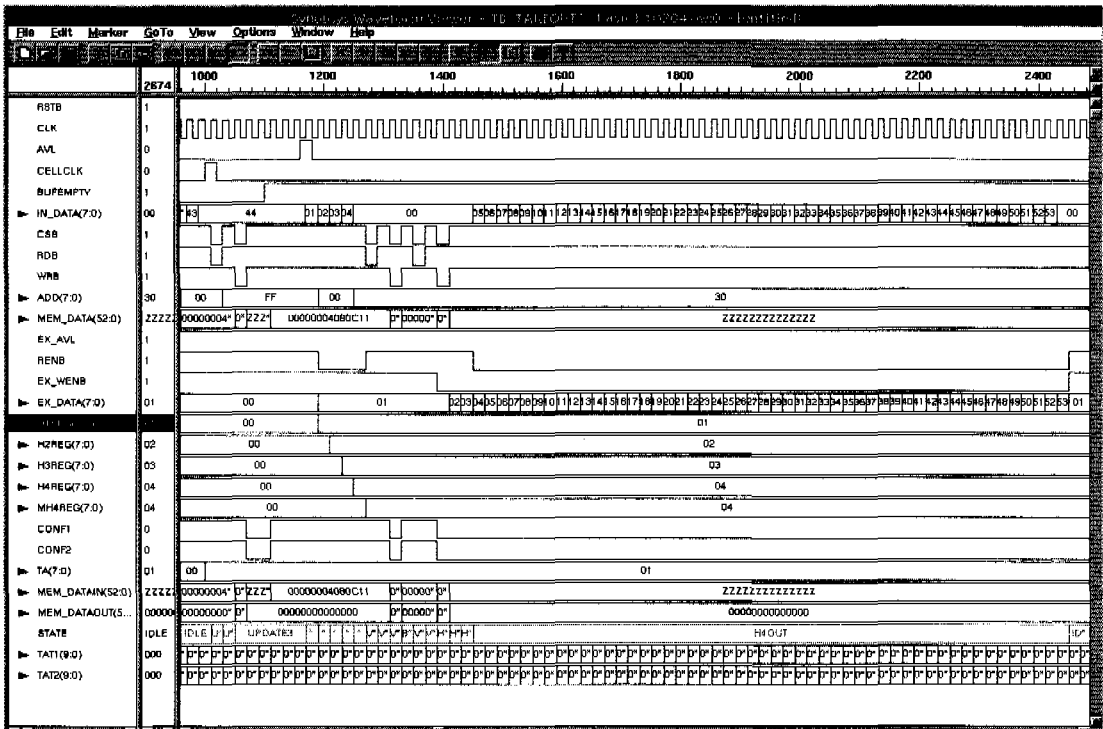


그림 9. UPC 모듈의 시뮬레이션 결과

과 CONF2 신호가 '1'이므로 셀은 최종적으로 적합한 셀에 해당하고 바로 출력 모듈로 보내지는 것을 나타내고 있다.

4. 출력 모듈의 설계 및 구현

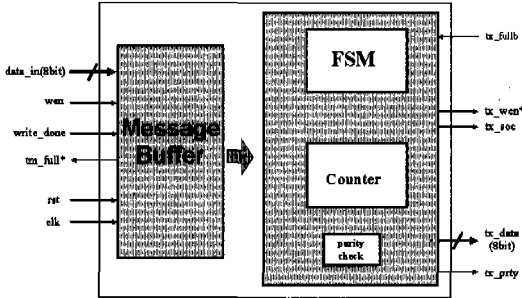


그림 10. 출력 모듈의 설계

그림 10은 출력 모듈을 도시하였다. 입력 모듈과 유사한 구조로서 전달될 셀은 먼저 메시지 버퍼에 쌓였다가 UTOPIA를 통해 물리적 계층으로 보내진다. 메시지 버퍼는 입력 모듈과 같이 FIFO 3개로 구성되어 있다.

5. 구현결과

제안한 감시 알고리즘은 입력 모듈, UPC 모듈, 출력 모듈의 3 부분으로 구성되어 있다. 입력 모듈

과 출력 모듈은 UTOPIA와 3개의 FIFO큐로 구성하였으며 UPC 모듈은 핵심 블록인 VSA₀과 VSA₀₊₁ 블록을 비롯하여 CLP=0인 셀을 일정시간 만큼 지연했다가 출력 모듈로 전송하는 블록이 포함되어 있다. 3가지 모듈을 합친 전체 블록의 시뮬레이션 결과를 그림 11에 도시하였다. 여기서 data_in은 입력 데이터를 나타내고 data_out은 출력 데이터를 나타낸다. cell_clk은 셀 타임 슬롯(time slot)이다. 각 셀 타임 슬롯에서는 카운터의 램퍼라운드를 수용하기 위해서 메모리에 있는 플래그를 활성화해 주어야 한다. 위 그림에서 메모리를 액세스하기 위해서 CSB(Chip Select low active), RDB(Read low active), Wrb(Write low active) 신호가 활성화되는 것을 알 수 있다. avl2 신호에 의해 메시지 버퍼에 처리해야 할 셀의 유무를 알 수 있고 위 그림에서는 avl2 신호가 활성화되었으므로 셀을 처리하기 위해서 외부메모리에 있는 TAT, τ, I 값을 불러온다. 이 셀은 VSA₀과 VSA₀₊₁ 블록의 연산을 거쳐 적합하다는 판정을 받고 셀이 data_out 버스로 출력된다. 그림 12에서는 디자인 어빌라이저 (design analyzer)를 이용한 회로 합성을 통하여 얻어진 실제 칩으로 제작될 회로를 도시하였다. 합성한 전체 블록의 게이트 수는 외부 메모리를 제외하고 2,387개이다.

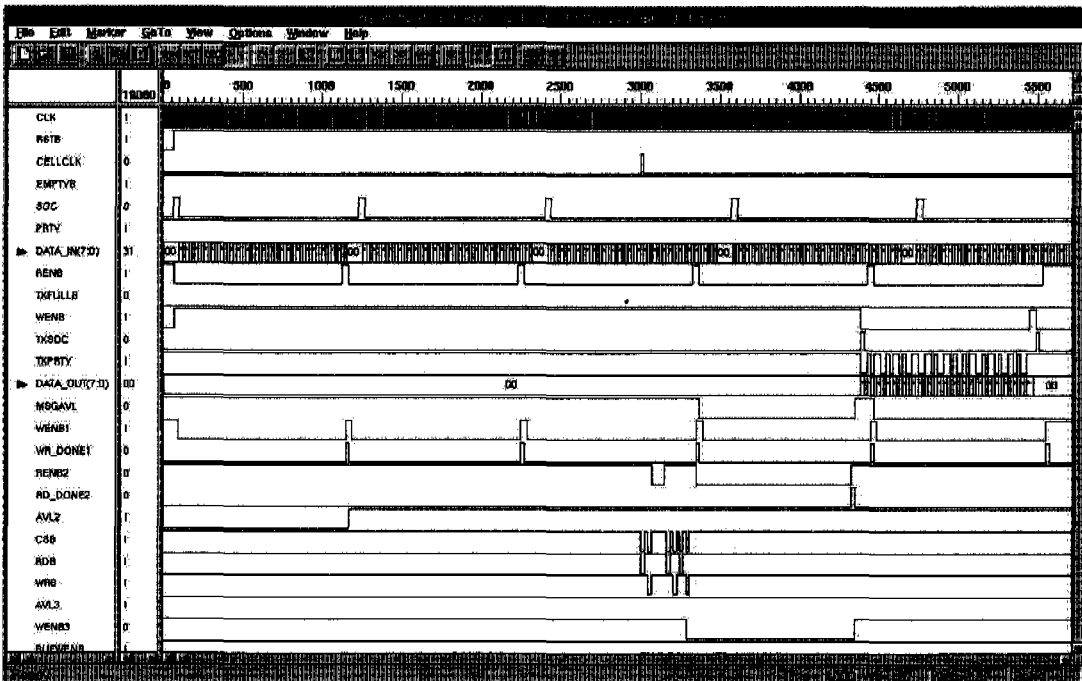


그림 11. 전체 블록 VHDL 시뮬레이션 결과

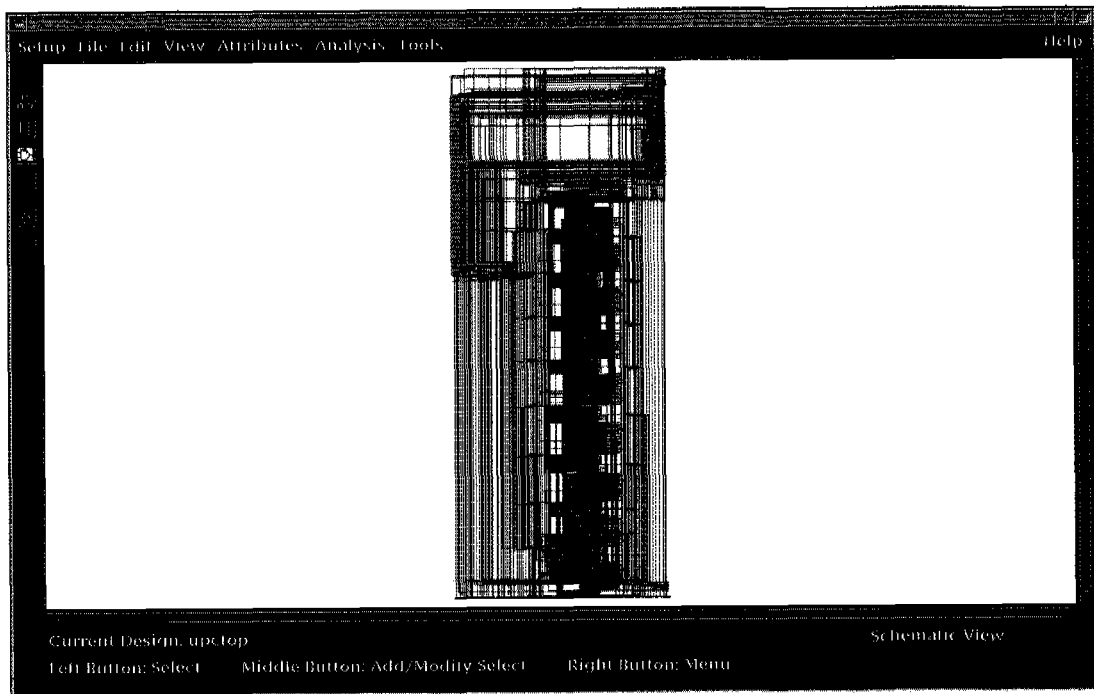


그림 12. 전체 블록의 합성결과

V. 결론

본 논문에서는 ATM 망의 트래픽 제어를 위한 감시 알고리즘을 제안하고 VHDL로 구현하였다. 구현한 칩은 셀손실 우선순위가 낮은 CLP=0 셀의 손실을 최소화 하기 위해 출력 모듈에서 하나의 셀을 저장할 수 있는 셀 버퍼를 사용하여, $VSA(T_0, \tau_0)$ 를 통과한 CLP=0 셀이 $VSA(T_{0+1}, \tau_{0+1})$ 을 수행하면서 만족치 못한 셀로 판단될 경우 해당 셀을 1셀 버퍼에 저장하고 일정 시간 지연 후에 전송함으로써 기존의 ITU-T I.371에서 제안한 UPC 알고리즘의 문제점을 개선하였다. 1셀 버퍼를 거치는 CLP=0 셀은 일정 시간 지연 후에 전송되므로 셀 간 시간간격(time between cells)이 커지게 되고, 따라서 트래픽의 균집성을 해소하는 효과를 얻을 수 있다.

구현한 칩은 입력 모듈과 UPC 모듈, 출력 모듈의 3 부분으로 구성된다. 특히 FSM 블록에서의 메모리 제어시 VCI를 참조하여 저장되어 있는 VSA 변수들을 읽는 방법을 사용하였고, 셀의 도착시간을 계수하는 카운터가 램-어라운드시 VSA에서 생길 수 있는 오류를 보정하는 알고리즘을 제안하고 구현하였다. 제안한 알고리즘의 구현을 위해서 아남 0.25 μm 라이브러리(library)를 이용하였다.

Synopsys사의 VHDL 어넬라이저(analyzer)를 이용하여 외부메모리를 제외한 회로의 합성결과로 2,387개의 게이트가 소요되었다.

참고문헌

- [1] ITU-T Recommendation I.371, "Traffic Control and Congestion Control," Frozen Issue-Paris, Mar. 1995.
- [2] ATM Forum Technical Committee, "Traffic Management," Raleigh, NC. Mar. 1994
- [3] D. Hong, T. Suda, and J. Bae, "Survey of Technique for Prevention and Control of Congestion in an ATM Network," *Proceeding of ICC*, pp. 204-201, 1991.
- [4] 조태경, 최병욱 "셀 손실 우선순위 기반의 사용 변수 제어," *한국통신학회 논문지 '99-2 Vol. 24 No.2A*
- [5] ATM Forum, Traffic Management Specification Version 4.1, March. 1999.
- [6] Fang-Jang KUO and Jung-Shyr WU, "Design of Multi-Connection Shaper and Enforcer for Usage Parameter Control in ATM Networks,"

IEICE Trans. Communication, Vol. E79-B No.1, pp. 8-16, Jan. 1996.

- [7] K. C. Chang, "Digital Systems Design with VHDL and Synthesis," *IEEE Computer Society Press Activities Board*, pp. 187-250, 1999.
- [8] Weng Fook Lee, "Coding and Logic Synthesis with SYNOPSIS," Academic Press, pp.57-225, 2000

이 요 섭(Yo-Seop Lee) 준회원



2001년 2월 : 한양대학교
전자컴퓨터공학부 졸업
2001년 3월~현재 : 한양대학교
전자전기제어계측공학과
석사과정
<주관심 분야> ATM, MPLS,
통신칩 설계

최 명 렬(Myung-Ryul Choi) 중신회원



1983년 : 한양대학교 전자공학과
학사
1985년 : 미시간주립대학교
컴퓨터공학과 석사
1991년 : 미시간주립대학교
컴퓨터공학과 박사

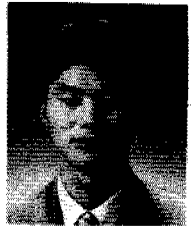
1991년 3월~10월 생산기술 연구원 전자정보실용화
센터 조교수

1991년 11월~1992년 8월 생산기술연구원산하 전
자부품종합기술연구소 선임연구원

1992년 9월~현재 : 한양대학교 제어계측공학과
부교수

<주관심 분야> ASICs, 신경회로망 칩 설계, 스마트
카드 응용, up/DSP 응용, Wireless ATM,
ITS

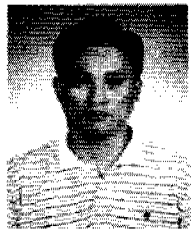
권 재 우(Jae-Woo Kwon) 학생회원



1999년 2월 : 한양대학교
제어계측공학과 졸업
1999년 3월~2001년 9월 :
한양대학교 전자전기제
어계측공학과 석사
2001년 10월~현재 : 국방과학
연구소 연구원

<주관심 분야> ATM, ATM Traffic management

이 상 길(Sang-Gil Lee) 준회원



1991년 2월 : 충남대학교
전자공학과 졸업
1993년 2월: KAIST 전기 및
전자공학과 석사
2001년 2월~현재: 한양대학교
전자전기제어계측공학과
박사과정

<주관심 분야> ATM, MPLS, 통신칩 설계