

# 초고속 신축버퍼의 구현

종신회원 홍 유 표\*, 정회원 양 기 주\*\*

## An Implementation of a High Speed Elasticity Buffer

Youpyo Hong *Lifelong Member*, Gijoo Yang *Regular Member*

### 요 약

컴퓨터 네트워크의 지속적인 보급과 멀티미디어에 대한 수요의 급증은 화상 회의 등의 새로운 수요에 대응 할 수 있는 초고속 근거리 통신망의 중요성을 부각시키고 있다. 이러한 초고속 근거리 통신망의 구현을 위해서는 연결된 컴퓨터들 간의 데이터 전송이 지연 없이 실시간으로 수행될 수 있도록 동기화시키는 것이 매우 중요하다. 네트워크상의 모든 컴퓨터들이 거의 같은 주기의 클럭을 사용할 경우, 데이터 정체를 최대한 줄일 수 있는 장점이 있는 반면, 송신단으로부터의 데이터를 수신단에서 받아들일 때 비동기 데이터 샘플링에 의한 준안정성 문제가 발생할 수 있기 때문에 그에 대한 해결을 위한 신축 버퍼가 필요하며, 본 논문에서는 고속 동작용 신축버퍼의 구현에 대해 논의한다.

**Key Words** : Elasticity Buffer, Synchronization, Metastability

### ABSTRACT

The importance of high-speed networking is ever increasing to better support multimedia application such as video conferencing. It is crucial to synchronize the network so that the delay between computers on the network is minimized. In high-speed LAN, for example, most computers use clocks with almost same frequency to minimize the delay for data transmission. However, because of the deviation of transmitter's and receiver's clock frequency and phase difference there can be a metastability problem. Elasticity buffer is to provide a solution for this situation and this paper presents an implementation is a high-speed elasticity buffer.

### I. 서 론

컴퓨터 네트워크의 고속화와 멀티미디어에 대한 수요의 급증은 화상 회의 등의 새로운 수요에 대응 할 수 있는 초고속 근거리 통신망의 중요성을 부각시키고 있다. 이러한 초고속 근거리 통신망의 구현을 위해서는 연결된 컴퓨터들간의 데이터 전송이 지연 없이 실시간으로 수행될 수 있도록 동기화(Synchronize)시키는 것이 매우 중요하다. 즉, 근거리 통신망은 다수의 컴퓨터들로 구성되어 있으며 각각의 컴퓨터들은 고유의 시스템 클럭을 가지고

있는데, 이들 클럭의 속도와 위상을 완전히 일치시키는 것은 현실적으로 불가능하기 때문에 각 컴퓨터 수신부 전단에 신축버퍼라고 불리는 버퍼를 삽입하여 데이터 정체를 최소화하며 고속 실시간 전송이 가능토록 하여야 한다. 신축버퍼는 서로 다른 클럭을 기반으로 동작하는 두 개의 다른 타이밍 영역의 접목지점에서 사용되므로, 수신부 최전단 저장장치에서 준 안정상태<sup>[1]</sup>의 발생가능성이 있다. (준 안정 상태란 플립플롭의 셀업/홀드 조건이 충족되지 못하여 논리적으로 1과 0으로 규정할 수 없는 값이 저장되거나 그 저장 값이 발진하는 현상을 지칭한다.)

\* 본 논문은 IDEC의 지원으로 이루어졌음.

\* 동국대학교 IT학부 전자공학전공 (yhong@dgu.edu), (° : 교신저자)

\*\* 동국대학교 IT학부 정보통신공학전공

논문번호 : KICS2009-06-236, 접수일자 : 2009년 6월 3일, 최종논문접수일자 : 2009년 7월 22일

신축버퍼의 구현에 관한 연구는 FDDI<sup>[2]</sup>, 기가비트 이더넷 등의 초고속 네트워크가 등장하며 본격적으로 시작되었고, Hutchison<sup>[3]</sup>, Horowitz<sup>[4]</sup> 등이 다양한 신축버퍼의 구조를 제안되었으며 Yuan<sup>[5]</sup>과 Micheliogiannakis<sup>[8]</sup> 은 신축버퍼의 VLSI 구현 결과까지 제시한 바 있다.

본 논문의 목적은 이러한 앞서 언급된 준 안정상태의 발생가능성을 최대한 억제하면서 고속으로 동작하는 신축버퍼를 구현하는 것이다.

## II. 신축 버퍼 구조

그림 1은 네트워크상의 각 컴퓨터의 수신단에 들어갈 신축버퍼의 역할 및 전체 구성을 나타내고있다. 그림 1은 신축버퍼의 기본 구성을 보여주고 있다. 실제로 많은 고속근거리 통신망의 경우 클럭정보가 데이터와 결합되어 전송되고, 클럭 및 데이터 복원회로(Clock and Data Recovery Circuit, CDR)에 의하여 데이터 신호로부터 클럭 정보를 추출해내는 방식이 많이 사용되고 있으나, 이러한 CDR 회

로는 신축버퍼 전단에 배치되는 것이 통상적이므로 본 논문에서는 데이터와 클럭이 동시에 병렬로 들어오는 것으로 가정하였으며, 유효한 패킷 데이터가 들어옴을 알리는 별도의 신호 ValidData가 있는 것으로 가정하였다.

신축버퍼의 주요 구성요소는 메모리부와 제어부이며, 메모리는 쓰기와 읽기를 동시에 하기 위하여 양방향 포트를 갖는 양방향 포트 FIFO를 사용한다. 데이터가 이 FIFO로 들어올 때는 송신단 클럭인 CLK<sub>w</sub>에 의하여 동기 되어 있으며, 수신단은 자신의 시스템 클럭인 CLK<sub>r</sub>를 이용하여 읽어내게 된다. 쓰기와 읽기 주소는 ADRS<sub>w</sub>와 ADRS<sub>r</sub>로 불리는 두 개의 계수기에 의하여 공급되며, 이들은 각

기 CLK<sub>w</sub>와 CLK<sub>r</sub>에 의하여 동작한다.

신축버퍼 제어부는 읽기와 쓰기를 위한 주소 생성부와 오버플로우/언더플로우 감지부로 구성되어 있다. 수신단으로 전달되어 오는 데이터가 없을 경우에는 유틸부호가 들어오는 것으로 간주하며, 이러한 경우에는 ValidData 신호가 활성화되지 않는 것으로 판별하는 것으로 가정하였다. ValidData신호가 활성화되기 전까지 두 개의 계수기 ADRS<sub>w</sub>와 ADRS<sub>r</sub>는 초기화 상태에 머물러 있게 된다. 유효 데이터가 들어오게 되면 ADRS<sub>w</sub>가 곧바로 초기상태에서 벗어나 쓰기용 주소를 생성한다. ADRS<sub>r</sub>은 버퍼의 반이 차기까지 초기상태에 머무르게 하는데, 데이터가 반이 차기를 기다리는 이유는 오버플로우나 언더플로우가 발생하는데 걸리는 시간을 최대한 연장하기 위해서이다. 가령 읽기 동작이 쓰기가 개시되자마자 시작된다면 CLK<sub>r</sub>이 CLK<sub>w</sub>보다 빠른 경우 곧바로 언더플로우가 발생할 것이다.

## III. 신축 버퍼의 구현

준안정 상태란 데이터가 변하는 동안 샘플링이 이루어지어, 논리 레벨이 부정확한 값이 저장되거나, 저장된 값이 발진하는 등의 비정상적인 현상이 발생하는 것을 일컬으며 신축버퍼에서는 두 가지의 준안정 상태 발생 가능성이 있는데, 본 논문에서는 그러한 준안정 상태 발생가능성을 최소화하며 고속으로 동작하는 신축버퍼의 구현이 목표이다.

첫 번째로 ADRS<sub>r</sub>의 초기화 해제를 위해서는 FIFO의 절반이 유효한 데이터로 찾아가를 판별하여야 하며, 이를 위해서는 항상 ADRS<sub>w</sub>의 값이 FIFO의 중간 주소값과 같은지를 비교하여야 하는데, 이 비교동작은 호스트의 시스템 클럭인 CLK<sub>r</sub>에 동기 되어야 하므로 준안정 상태 발생 가능성이 존재한다. 이 경우 준안정 상태 발생 가능성을 최소화 하기 위해서는 이미 널리 사용되고 있는 기법인 오버샘플링(Oversampling), 즉 ADRS<sub>w</sub>의 출력을 CLK<sub>r</sub>로 동기된 두 개의 플립플롭을 연이어 경유시키는 방법으로 처리한다.

두 번째로, 오버플로우 또는 언더플로우를 검사하는 과정에서 두 주소 생성기의 출력값이 같은지를 비교하여야 하는데 이 때에도 ADRS<sub>w</sub>값을 CLK<sub>r</sub>를 이용하여 판독하여야 하므로 준 안정상태의 발생 가능성이 존재한다. 이 경우에도 준 안정상태 자체는 앞서 언급한 오버샘플링 기법을 적용하여 줄일 수 있으나 다음에서 설명할 유효 데이터

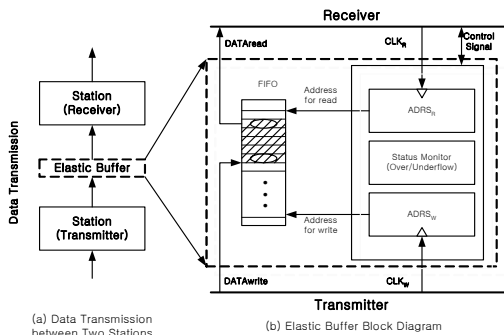


그림 1. 신축버퍼의 구성도

유실의 가능성이 존재한다.

준안정 상태란 데이터가 변하는 동안 샘플링이 이루어지거나, 논리 레벨이 부정확한 값이 저장되거나, 저장된 값이 발진하는 등의 비정상적인 현상이 발생하는 것을 일컫는다. 신축버퍼에서는 두 가지의 준안정 상태 발생 가능성이 있는데, 본 논문에서는 그러한 준안정 상태 발생가능성을 최소화하며 고속으로 동작하는 신축버퍼의 구현이 목표이다.

첫 번째로  $ADRS_R$ 의 초기화 해제를 위해서는 FIFO의 절반이 유효한 데이터로 찾아가를 판별하여야 하며, 이를 위해서는 항상  $ADRS_W$ 의 값이 FIFO의 중간 주소값과 같은지를 비교하여야 하는데, 이 비교동작은 호스트의 시스템 클럭인  $CLK_R$ 에 동기되어야 하므로 준안정 상태 발생 가능성이 존재한다. 이 경우 준안정 상태 발생가능성을 최소화하기 위해서는 이미 널리 사용되고 있는 기법인 오버샘플링, 즉  $ADRS_W$ 의 출력을  $CLK_R$ 로 동기된 두 개의 플립플롭을 연이어 경유시키는 방법으로 처리한다.

두 번째로, 오버플로우 또는 언더플로우를 검사하는 과정에서 두 주소 생성기의 출력값이 같은지를 비교하여야 하는데 이 때에도  $ADRS_W$ 값을  $CLK_R$ 를 이용하여 판독하여야 하므로 준안정상태의 발생 가능성이 존재한다. 이 경우에도 준안정상태 자체는 앞서 언급한 오버샘플링 기법을 적용하여 줄일 수 있으나 다음과 같은 유효 데이터 유실의 가능성이 존재한다.

유효한 데이터가 유실되는 경우는 크게 두 가지가 있는데, 첫 번째로 데이터가 변하는 상황에서 샘플링이 이루어질 경우, 미세한 타이밍 차이에 의하여 변화 이전, 혹은 변화 이후의 데이터가 샘플링되는 것은 정상적인 상황이지만, 전혀 상관이 없는 값으로 샘플링이 되는 경우가 존재한다. 가령, 4비트  $ADRS_W$ 를 가정할 경우 그 출력 값이 1111에서 0000으로 변하는 상태에서  $CLK_R$ 를 이용하여 샘플링 하는 경우, 각 비트마다의 타이밍 미세한 차이에 의하여 1111과 0000이외의 값, 예를 들면 1010등과 같이 전혀 상관이 없는 값이 샘플될 수 있다. 이러한 현상은 주소데이터를 그레이코드<sup>[6]</sup>로 부호화하여 사용함으로써 해결할 수 있다.

데이터가 유실되는 두 번째 경우로서 오버플로우나 언더플로우를 검사하는 회로가  $CLK_R$ 을 클럭으로 사용하는데 만약  $CLK_W$ 의 주기가  $CLK_R$ 주기보다 빠르면  $ADRS_W$ 의 값을 놓치는 경우가 발생한다. 이를 방지하기 위하여  $ADRS_R$ 의 값을  $ADRS_W$ 의

값과도 비교하고, 동시에 180° 쉬프트된  $ADRS_W$ 과도 비교를 하면,  $CLK_W$ 의 주기가  $CLK_R$ 주기보다 두 배 이상 빠르지 않다는 전제하에, 두 비교 중 최소한 하나는  $ADRS_W$ 의 변화가 감지된다. 근거리통신망 상의 컴퓨터 시스템 클럭 주기의 차이는 두 배 이상 나지 않는 경우가 일반적이므로 두 배까지의 주기차를 해결하는 것은 큰 의미를 갖는다. 이를 위하여 본 신축버퍼 제어부에서는  $ADRS_W$ 출력을 호스트 클럭과 호스트클럭이 180° 쉬프트된 클럭에 의해 두 번 샘플한다.

본 논문에서 제안된 신축버퍼는 동작 속도측면에서 매우 유용한 동적회로의 한가지인 TSPC(True Single Phase Circuit)<sup>[7]</sup>스타일을 이용하여 구현하였으며, 180° 쉬프트는 단일 극성의 래치를 추가함으로써 쉽게 구현 가능하며 그 자세한 설명은 다음과 같다. 그림 2는 완성된 신축버퍼 제어부의 구성을 보여주고 있는데,  $ADRS_W$ 의 출력이 N형 회로로부터 생성되고 그 출력을 오버샘플링하기 위하여 두 개의 플립플롭을 경유하여  $ADRS_R$ 의 출력과 비교된다. 만약 두 개의 주소가 동일하면 오버플로우나 언더플로우가 발생한 것으로 판단되어 호스트로부터 적절한 조치가 취해지도록 사용될 것이다.  $ADRS_W$  출력이 180° 쉬프트 되도록 하기 위해서는 N형 계수기로부터 생성된  $ADRS_W$ 의 출력을 일단 한 개의 플립플롭을 경유시킨 뒤, 또 다른 플립 플롭 대신 단지 P형 래치만을 경유시킴으로서 Precharge와 Evaluation 위상을 반전시켜 실질적으로 180° 쉬프트된  $ADRS_W$ 를 얻도록 구성하였다.

전술된 회로 이외에 신축버퍼 구현을 위해 필요한 모듈은 비교기이다. 비교기는 덧셈기 등과 마찬가지로 많은 디지털 회로에서 사용되고 있는 매우 기본적인면서도 간단한 회로이다. 그러나, MSB부터 하위 비트로 차례로 비교하여야 하는 방식을 그대로 채용하는 방식은 동작속도가 크게 떨어지고, 고속 뺄셈기를 이용하는 경우도 속도에 속도 향상에

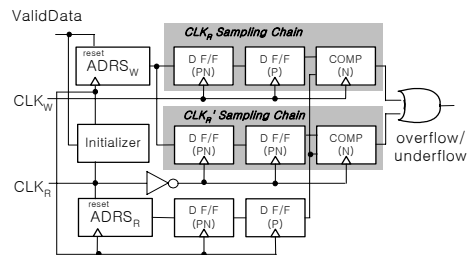


그림 2. 신축버퍼 제어부

제약이 있다. 비교기의 동작속도를 향상시키기 위하여 한쪽 데이터의 값이 다른 쪽보다 크다고 판단할 수 있는 모든 경우를 동시에 계산하는 방식의 비교기가 특허 [8] 를 통해 제안되었으며. 그림 3은 그러한 개념에 기초하여 4비트 데이터 두 개의 크기를 비교하는 회로의 구성도이다.

본 연구에서는 비교기의 동작속도를 개선시키기 위하여 저자에 의해 기존 게이트 수준 비교기의 개념을 트랜지스터 수준에서 최적화 시킨 비교기[9]를 사용하였다. 그림 4는 TSPC형 회로스타일에 기반을 둔 4비트 비교기의 회로도도를 보여주고 있다. 비교기만의 독립적 시뮬레이션 결과 제안된 비교기는 약 0.5ns정도의 지연시간을 가지며 [10], [11]과 같은 기존의 비교기들에 비하여 약 2배 가량 빠르게 동작하는 것으로 측정되었을 뿐만 아니라, 트랜지스터의 수효가 기존 비교기들에 비해 대폭 감소하여 전력 소모량 측면에서도 현재까지 가장 우수한 것

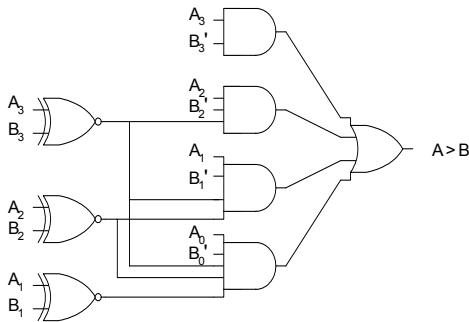


그림 3. 기존의 비교기<sup>[8]</sup>

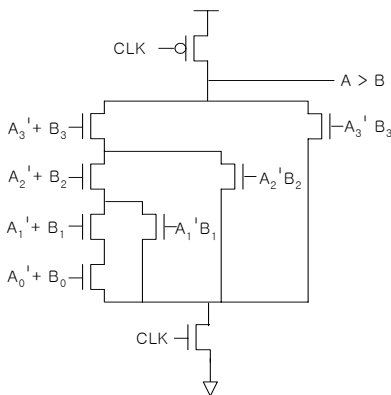


그림 4. 고속 비교기

으로 알려진 [10]에 비하여 약 3배 가량 우수한 것으로 측정되었다.

#### IV. 구현 및 실험결과

본 논문에서 제안된 신축버퍼는 0.25 $\mu$ m CMOS 공정을 통하여 설계되었으며 그림 5는 설계된 신축 버퍼의 레이아웃을 보여주고 있다. 칩의 크기는 패드 등의 IO셀 들을 제외하면 대략 1.5 $\times$ 1.5 mm이며 Hspice를 이용한 시뮬레이션 결과 약 475MHz까지 정상적인 오버플로우 및 언더플로우의 감지가 가능한 것으로 측정되었으며, 최대 속도 동작시 약 900mW의 전력을 소모하는 것으로 측정되었다.

VLSI 구현 결과까지 보고된 [5]의 경우, 설계된 신축 버퍼의 최대 동작속도가 200MHz에 머물렀고, 최근 보고된 [8]의 경우 300MHz 까지의 동작이 가능한 것으로 보고 되었다. [8]의 경우는 0.45 $\mu$ m CMOS 공정을 사용하였기 때문에 동일 공정 사용 시 제안된 신축버퍼의 동작속도 향상폭은 더욱 클 것을 예상할 수 있다.

FIFO로는 널리 사용되고 있는 [10]의 레지스터 파일을 이용하였으며 그 크기는 32x8비트를 사용하였고, 바로 이 레지스터 파일의 동작속도가 전체회로의 동작 속도를 제한하는 것으로 측정되었다. 따라서, 대상 응용 분야에 따라 레지스터 파일의 크기가 감소하거나 보다 진보된 FIFO를 사용할 경우 전체회로의 구동속도는 크게 향상될 것으로 예측된다. 신축버퍼의 제어부만을 따로 동작시키는 경우 약 1GHz이상의 고속 동작이 가능한 것으로 측정되었으며, 신축버퍼 조절부내에서 가장 저속으로 동작하는 회로는 계수기로서 FIFO크기에 따라 그 속도가 민감하게 영향을 받게 됨을 예측할 수 있다.

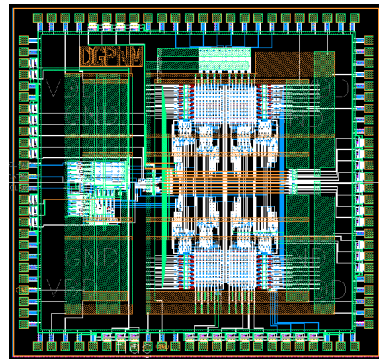


그림 5. 신축버퍼 레이아웃

## V. 결 론

본 논문에서는 초고속 근거리 통신망의 실시간 고속 데이터 전송을 위한 신축버퍼를 제안하여 실험 결과를 통하여 고속 동작을 검증하였다. 비동기 샘플링에서의 준 안정상태는 그레이 코드 인코딩과 오버샘플링을 통하여 해결하되 TSPC회로 스타일로 구현함으로써 고속 동작이 가능하도록 하였으며, 상대적으로 속도가 느린 비교기의 동작 속도를 대폭 향상시키어 신축버퍼 제어부 전체의 동작속도를 크게 향상시키었다.

## 참 고 문 헌

- [1] T. J. Chaney, C. E. Monlar, "Anomalous Behavior of Synchronizer and Arbiter Circuits," IEEE Transactions on Computers, pp. 421 - 422, April 1973.
- [2] C. Mead, L. Conway, "Introduction to VLSI systems," Addison Wisley.
- [3] J. D. Hutchison, C. Baldwin, and B. W. Thomson, "Development of the FDDI Physical Layer," Digital Technical Journal Vol. 3 No. 2 Spring 1991.
- [4] P. Horowitz, and W. Holl, "The Art of Electronics," Cambridge.
- [5] J. Yuan, and C. Svensson, "High-speed CMOS Circuit Technique," IEEE Journal of Solid-State Circuits, Vol.24, No.1, pp62-70, February 1989.
- [6] W. Hwang, R. V. Joshi., W. H. Henkels.: A 500-MHz, 32-Word×4-Bit, Eight-Port Self-Resetting CMOS Register File, IEEE Journal of Solid-State Circuits, Vol. 34, Issue 8, pp.1108 - 1117, Aug. 1999.
- [7] B. Xiaoping, Z. Hongwei, "Study of Digital Video Interface (DVI) Architecture Design in Computer System," International Conference on Communications, Circuits and Systems, pp.1200-1203, 2007.
- [8] G. Michelogiannakis, J. Balfour and W. J. Dally, "Elastic-Buffer Flow Control for On-Chip Networks," IEEE 15th International Symposium on High Performance Computer Architecture, pp.151- 162, 2009.
- [9] H. Shin, J. Lee, Y. Hong, "Comparator circuit

and method," United States Patent, Patent No. 7,103,624, B2, Sept. 5, 2006.

- [10] R. D. Adams, D. A. Evans, R. C. Flaker, "High Speed Greater Than Or Equal To Compare Circuit," United States Patent 5,592,142, 1997.
- [11] S. S. Park, "Binary Comparator," United States Patent 6,014,074, 2000.

### 홍 유 표 (Youpyo Hong)

중신회원



1991년 2월 연세대학교 전기공학과 학사

1993년 5월 University of Southern California 전기공학과 석사

1998년 8월 University of Southern California 컴퓨터

공학과 박사

1998년 7월~1999년 2월 Synopsys, Hillsboro, Senior Engineer

1999년 3월~현재 동국대학교 IT학부 전자공학전공 부교수

<관심분야> 멀티미디어 칩 설계, SOC 설계

### 양 기 주 (Gijoo Yang)

정회원



1984년 5월 미국 위스콘신대 전산학 학사

1986년 5월 미국 미시간대 전산학 석사

1991년 12월 미국 텔라웨어대 전산학 박사

1992~1995 KT 연구소 선임

연구원

1995년 9월~현재 동국대학교 IT학부 정보통신공학 전공 교수

<관심분야> 네트워크, 데이터 마이닝