
불안정 상태를 제거한 NoC용 위상차 클럭 동기회로

김강철*

Metastability-free Mesochronous Synchronizer for Networks on Chip

Kang Chul Kim*

요 약

본 논문에서는 미래의 온칩통신 구조로 각광받고 있는 NoC의 GALS 클럭 구조에서 불안정 상태를 제거하기 위한 위상차 동기방법과 위상차 동기회로를 제안한다. 제안된 방법은 송신부의 클럭을 입력 스트로브 신호로 사용하고, 송수신부 클럭의 위상차가 불안정 상태 영역에 존재하더라도 샘플링 결과 값에 따라 클럭의 상승 모서리 또는 하강 모서리 중의 하나를 선택하여 불안정 상태를 피할 수 있다. 고장을 삽입한 로직 시뮬레이션을 통하여 $0^{\circ} \sim 360^{\circ}$ 위상차에서 불안정 상태에 관계없이 위상차 클럭 동기회로가 잘 동작함을 확인하였다. 그리고 제안된 위상차 클럭 동기회로는 위상 검출기가 필요하지 않아 제어기가 간단하며, 모든 회로가 디지털 회로로 구성되어 NoC의 클럭 동기 회로에 적합하다.

ABSTRACT

This paper proposes a metastability-free synchronization method and a mesochronous synchronizer for NoC. It uses the clock transmitted from TX as a strobe and solves the metastability problem by selecting one of rising or falling clock edge depending on the sampling value in RX when the phase difference between clocks is under a metastability window. The logic simulation results show that it works without metastability under $0^{\circ} \sim 360^{\circ}$ phase difference in the synchronizer that a fault is inserted. The mesochronous synchronizer has a simple control logic and is suitable for NoC.

키워드

위상차 클럭 동기회로, 불안정 상태, 동기 모서리, 온칩 통신

Key word

Mesochronous synchronizer, Metastability, Synchronization edge, On-chip communication

* 정회원 : 전남대학교 전기전자통신컴퓨터공학부(kkc@jnu.ac.kr)

접수일자 : 2012. 04. 27

심사완료일자 : 2012. 05. 15

Open Access <http://dx.doi.org/10.6109/jkiice.2012.16.6.1242>

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

I. 서 론

반도체 공정기술의 발달로 인하여 하나의 칩에 수십억 개의 트랜지스터를 내장하는 칩들이 발표되고 있다. 또한 CAD 기술의 향상은 하나의 칩 내에 수십 개의 처리기와 메모리 및 주변회로를 포함하는 SoC의 구현이 가능하도록 하였다.[1-3] 그리고 프로세서, 메모리, 주변장치 등의 많은 IP를 포함하는 MPSoC 구조가 가능하여 통신, 멀티미디어 데이터 처리 등의 연산 작업량이 많은 곳이나, 크기와 전력소모가 제약조건이 되는 임베디드 시스템 등에 사용되고 있다. 그러나 MPSoC의 복잡도 증가는 설계생산성격차 문제[4][5]와 온칩 통신량의 증가를 유발한다. 온칩통신구조는 시스템 성능, 전력소모, time-to-market에 아주 중요한 영향을 미치므로, 최근에는 설계목적에 도달할 수 있도록 SoC 통신구조에 대한 연구, 평가, 설계에 관한 연구가 수행되고 있다.

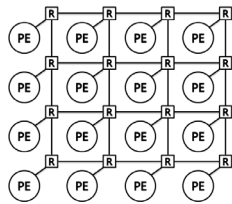


그림 1. NoC 구조
Fig. 1 The structure of NoC

설계규칙의 감소는 여전히 칩내에서 상호연결 성능, 신뢰도, 전력소모 등 기준에 예측하지 못했던 많은 어려움을 발생시키고 있다. 트랜지스터 크기는 비례축소되고 있으나 버스의 길이는 비례축소할 수 없다. 또한 버스는 한 번에 하나의 프로세서 밖에 지원할 수 없으며, 버스에 접속된 모든 시스템은 버스 대역폭을 공유하고 있다. 그리고 프로세서 수가 증가함에 따라 단위면적당 전력소모도 증가하고 있다. 이러한 문제점들을 해결하기 위한 방안으로 최근에 대두되고 있는 방안이 NoC(Networks-on Chip)이다.[6][7] 그림 1과 같이 NoC는 효율성과 확장성을 고려하여 제안된 새로운 내장형 통신구조로서 계층적 프로토콜을 이용하는 네트워크 인터페이스 및 라우터에 의한 패킷 데이터 전송을 통해 공유버스의 문제점들을 해결 할 수 있다. 그러나 같은

실리콘 면적에 더 많은 IP가 집적되고, 전역 신호선의 지연은 비례축소될 수 없어 실제적으로 온칩 통신속도가 시스템 성능의 병목현상이 되고 있다.

시스템의 크기나 성능에 제한을 받지 않는 클럭 구조로 GALS(Globally Asynchronous Locally Synchronous)가 제안되었다.[8] 이 방법은 한 시스템을 독립적인 클럭 모듈로 나누고, 각각의 하부 시스템은 자신의 클럭에 의해서 GS(Global Synchronous) 방법으로 동기를 맞추어 동작하며, 하부 시스템 사이에는 비동기 블록을 삽입하여 통신하는 방법이다[9].

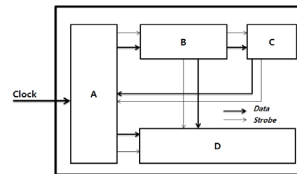


그림 2. 위상차 동기방식
Fig. 2 Mesochronous synchronization

GALS에서 IP들은 스큐를 무시할 수 있는 영역내에서 동기적으로 동작하며, 전역신호선에서는 비동기 방식을 사용하여 스큐가 문제가 되지 않도록 한다.[10] 동기식과 비동기식의 양자균형 방법으로 위상차 동기방식이 있다.

그림 2는 위상차 동기방식의 한 방법으로 클럭의 분배가 버스 속에 포함된 것을 알 수 있다.[11] 스트로브(strobe) 신호가 각 링크 속에 있으므로 그블록의 지역 클럭으로 사용될 수 있다. 이 방식에서는 클럭 위상에 관한 제어가 없으므로 불안정 상태를 피하기 위하여 특별한 제어방법과 클럭동기회로의 설계에 대한 연구가 필요하다.[12]

본 논문에서는 미래의 온칩통신 구조로 각광받고 있는 NoC의 기술 개발을 위한 GALS 클럭 구조에서 불안정 상태를 제거하기 위한 위상차 동기방법과 위상차 동기회로를 제안한다. 제안된 방법은 송신부의 클럭을 입력 스트로브 신호로 사용하고, 송수신부의 클럭 위상차가 불안정 상태 영역에 존재하더라도 샘플링 결과 값에 따라 클럭의 상승 모서리 또는 하강 모서리 중의 하나를 선택하게 되므로 불안정 상태를 피할 수 있다. 본 논문의 2장에서는 국내외 연구동향을 간단히 설명하고, 3장에서 논문에서 제안한 위상차 동기 방법을 제안한다.

그리고 4장에서는 3장에서 제안한 방법을 이용하여 위상차 동기회로를 설계한다. 5장에서는 제안한 회로에 고장을 삽입하여 불안정 상태에 관계없이 위상차 동기회로가 잘 동작하는 것을 보여주고, 6장에서 결론을 기술한다.

II. 관련 연구 동향

NoC가 온칩통신구조의 대안으로 대두되면서 GALS에서 사용될 수 있는 위상차 동기 방식에 관련된 논문이 최근에 많이 발표되고 있다. Mesgarzadeh[11]는 두 블록 사이에 하나의 링크를 사용하여 데이터와 동시에 클럭을 전송하는 구조를 사용하여 고속으로 동작하는 칩에서도 사용이 가능하지만 위상 검출기와 주파수 배수기가 사용되어 회로가 복잡하다. Bojnor[13]는 [11]에서 제안된 회로의 단점을 보완하여 양모서리 동기 플립플롭(double-edge triggered flip-flops) 기반 위상차 동기회로를 제안하였다. 이 구조는 완전 주문형 설계에 적합하며, 다음 블록으로 신호 전달 시에 동기 모서리가 달라질 수 있는 단점을 가지고 있다. M 단계 FIFO 동기회로에 기반한 구조가 [14]에서 제안되었으나 불필요한 면적을 차지하고 전력을 소모하는 단점을 가지고 있다. 이 구조는 Caputa[15]에 의해 칩으로 구현되었다. Semiat[16]는 가장 간단하고 효율적인 brute-force 동기회로이지만 완전 주문형 회로 설계에 적합하고, 동작 주파수 낮은 단점을 가지고 있다.

Ingemar[17]는 모든 블록들이 동기 방식으로 동작하도록 기본 블록은 GS 모델을 사용하고, 간단한 동기 타이밍 교정 절차를 가지도록 설계되었다. 그리고 블록 사이에 양방향 데이터 링크가 존재하고 클럭 통로에 적절한 시간 지연회로가 있어야 한다는 2 가지 가정이 필요하며, 회로가 복잡하여 많은 면적을 차지하고 전력소모가 크다는 단점을 가지고 있다. Loi[18]는 3D NoC에서 양방향 위상차 동기회로를 이용한 통신 구조를 제안하였다. 이 구조는 front-end와 back-end로 구성된 각 층의 위상차 동기회로를 보여준다. 이 논문에서는 3D NoC를 고려하여 위상차 동기회로를 설계하였으나 위상 검출기를 사용하고 있고, 동기 타이밍 여유가 작은 단점을 가지고 있다. Ludovici[19]는 [18]에서 발생된 문제점을 개선한 것으로 위상 검출기를 제거 하였으며, 송신부와 수

신부를 통합하여 성능을 향상시킨 것이다. 그러나 위상 검출기가 제거되는 과정에서 플립플롭의 수가 증가하게 되고, 지연시간 여유를 가지기 위하여 brute-force 설계방법을 이용하여 완전주문형 설계 방법에 적합하다는 단점을 가지고 있다.

지금까지 설명된 위상차 클럭 동기회로는 불안정 상태를 피하기 위하여 위상 검출기를 사용하거나 brute-force 방식을 사용하여 주문형 설계방식에만 사용할 수 있고, 반도체 공정에 많이 의존하게 되는 단점을 가지고 있다. 따라서 동기속도가 빠르고 회로가 단순하며, 불안정 상태를 원천적으로 제거 가능한 위상차 클럭 동기회로의 설계가 요구되고 있다.

III. 불안정 상태 제거 위상차 클럭 동기회로

본 연구에서는 플립플롭의 불안정 상태에 영향을 받지 않고, 회로가 간단하여 전력소모가 적어 3D NoC에서도 사용이 가능한 위상차 클럭 동기회로를 설계한다.

본 연구에서 제안하는 위상차 클럭 동기회로는 그림 3과 같이 TX(송신부)로부터 데이터와 클럭 CLK_TX를 함께 받아 수신부의 내부 클럭과 동기시킨다. CLK_RX는 동기회로와 수신부에서 사용되는 지역 클럭이다.

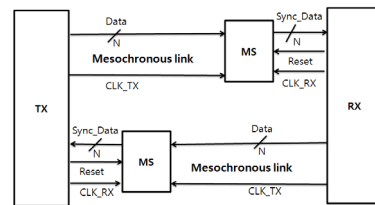


그림 3. 위상차 클럭 시스템 구조
Fig. 3 Mesochronous clocking system

불안정 상태를 피하기 위하여 샘플링 모서리와 동기 모서리(Synchronization Edge: SE)를 정의한다. 샘플링 모서리는 TX(송신부)에서 전송된 입력 데이터를 샘플링하기 위하여 사용된 모서리로 CLK_TX에 동기된다. SE는 그림 4와 같이 위상차 클럭 동기회로에서 샘플링 데이터가 CLK_RX에 동기되어 RX(수신부)로 보내지는 모서리를 의미한다.

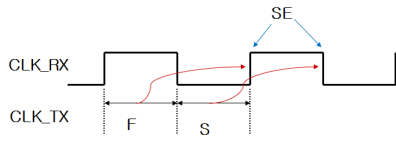


그림 4. SE의 정의
Fig. 4 Definition of SE

본 논문에서는 CLK_TX와 CLK_RX는 같은 주파수를 가지고 위상만 다르다고 가정하고, 위상차 동기회로의 지연(latency)을 줄이고 불안정 상태를 피하기 위하여 2가지 방법을 제안한다. 첫 번째로 SE는 CLK_RX의 상승 또는 하강 모서리로 설계 시에 결정되지 않고 전송부와 수신부 사이의 물리 링크에서 발생하는 클럭 스퀴에 의해 결정된다. 즉 스트로브(strobe) 신호로 사용되는 CLK_TX의 상승 모서리가 CLK_RX의 첫(두) 번째 반주기 F(S) 동안에 발생하면 그림 4와 같이 동기회로의 SE는 CLK_RX의 다음 상승(하강) 모서리가 된다. 시스템 리셋(system reset)이 동작한 후 CLK_TX의 첫 번째 상승 모서리가 동기회로에 들어오자마자 SE 결정 과정이 수행된다. 제안된 방법은 기존의 방법보다 동기시간을 반주기 이상 줄일 수 있다. 동기회로의 지연 td는 식 (1)과 같이 계산된다.

$$t_d = (1/2 + (|180^\circ - PD|)/360^\circ) t_{CLK_RX} \quad (1)$$

여기서 t_{CLK_RX} 는 CLK_RX의 클럭 주기이며, PD는 CLK_TX와 CLK_RX의 위상차이다. 다른 로직과 동기를 맞추기 위하여 동기가 결정되어야 한다면 SE는 상승 또는 하강 모서리 중의 하나가 결정되며, 스트로브 신호의 위치에 따라서 반주기 더 지연이 발생할 수 있다. 비동기 클럭 시스템에서 입력 데이터를 정확하게 얻기 위하여 신호를 샘플링할 때 동기를 맞추는 것이 필요하다. 그러나 동기 신호와 데이터의 두 신호 변화가 거의 동시에 일어나면 동기가 실패하여 데이터를 잃어버릴 수 있다. 즉 입력과 플립플롭의 제어신호가 비동기적으로 사용될 때 불안정상태가 발생할 수 있다. 플립플롭의 경우 set-up 시간과 hold 시간이 준비되지 않으면 플립플롭의 두 출력의 전압차가 거의 없는 불안정 상태로 들어갈 수 있으며, 이 때 전체 시스템이 불안정하고 신뢰도가 없는 상태가 될 수 있다.

이러한 문제는 플립플롭을 사용하는 지역 클럭 발생기를 가지고 신호의 한 개에 동기시켜 해결할 수 있다. D

플립플롭은 0 또는 1의 2개의 상태 중의 하나를 가질 수 있으므로 불안정 상태는 결국 두 상태 중의 하나로 수렴한다. 그러나 수렴시간이 길어져 값이 결정되기 전에 다음 동작을 수행하게 되면 틀린 결과를 발생하게 된다. 비동기 신호를 클럭에 동기시키는 회로에서 동기 실패가 발생하는 시간 MTBF(Mean Time Between Failures)은 식 (2)와 같이 계산된다.

$$MTBF = 1/(f_m \times f_{clk} \times t_d) \quad (2)$$

여기서 f_m 은 비동기 신호의 주파수, f_{clk} 는 클럭 주파수, t_d 는 임계시간 창 길이다. 위상차 클럭 시스템에서 MTBF는 식 (3)으로 주어진다.

$$MTBF_{MC} = 1/(f_{clk_TX} \times f_{clk_RX} \times t_d) = 1/(f_{clk_TX}^2 \times t_d) \quad (3)$$

두 번째로 본 연구에서는 입력 클럭의 위치에 관계없이 불안정 상태를 피할 수 있는 방법을 제안 한다. 완벽하게 불안정 상태를 제거할 수는 없지만 two flops의 동기회로가 사용된다면 불안정 상태가 발생할 확률은 많이 줄어들 것이다.[20] 위상차 동기 방식도 비동기 클럭 시스템과 같은 조건을 가지고 있다. 데이터 래치 에러(data latching error)를 피하기 위하여 MW(Metastability Window)는 입력 데이터가 플립플롭에서 안전하게 래치될 수 없는 기간으로 정의 된다. 즉 그림 5와 같이 set-up 시간과 hold 시간의 합으로 정의된다. 스트로브 신호가 MW 내에서 발생하면 래치의 결과 값은 불안정할 수 있다. 그러므로 대부분의 위상차 동기 회로에서는 MW를 피하기 위하여 위상 검출기를 사용한다. 그림 5에서 MW의 A 부분에서 f 기간 내에 첫 번째 CLK_TX의 상승 모서리가 발생한다고 가정하자. 입력 래치는 링크에서 들어오는 입력 데이터를 받아들이고, 불안정 상태 때문에 그 값들은 0 또는 1의 값을 가지게 될 것이다. 그 값이 0(1)이라면 현재(다음) 클럭의 두(첫) 번째 반주기 S0(F1)에서 채워진 값과 같은 0(1)의 안정된 상태로 여겨지고, SE는 그림 5와 같이 CLK_RX의 다음 하강(상승) 모서리에서 결정될 것이다. MW의 A 부분에서 b 기간 내에 첫 번째 CLK_TX의 상승모서리가 발생한다고 가정하자. 입력 래치는 링크에서 들어오는 입력 데이터를 받아들이고, 불안정 상태 때문에 그 값들은 0 또는 1의 값을 가지게 될 것이다. 그 값이 1(0)이라면 현재(전) 클럭

의 두(첫) 번째 반주기 F1(S0)에서 채취된 값과 같은 0(1)의 안정된 상태로 여겨지고, SE는 CLK_RX의 다음 상승(하강) 모서리에서 결정될 것이다.

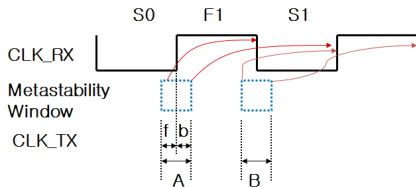


그림 5. 불안정 상태 영역 및 SE
Fig. 5 Metastability window and SE

스트로브 신호가 윈도우 B에서 발생하면 SE는 위의 설명과 같은 방법으로 래치의 값에 따라 CLK_TX의 다음 상승 또는 하강 모서리에서 결정될 것이다. 그림 5에서 MW의 f 기간 동안에 입력 래치가 1을 가지면 동기회로의 지연시간은 최대 MW의 반주기 더 늘어날 것이다. 본 연구에서 제안된 MS의 최대 지연 t_{ms} 는 식 (4)와 같이 계산된다.

$$t_{ms} = (1/2 + ((180^\circ - PD)/360^\circ))t_{CLK_RX} + t_{mw}/2 \quad (4)$$

여기서 t_{mw} 는 MW 기간이다. 최소 클럭주기는 식 (5)에 의해서 얻어진다.

$$t_{CLK_TX}/2 \geq t_{mw}/2 + t_{mwd} + t_{count} \quad (5)$$

여기서 t_{mwd} 는 MW 지연로직의 지연시간이고, t_{count} 는 multiplexer의 카운터의 지연시간이다.

IV. 위상차 클럭 동기회로 구조

수신부의 MS는 송신부에서 전송한 데이터/제어 신호와 클럭을 받아들이며, 클럭은 스트로브 신호로 사용된다. 데이터와 스트로브 신호는 수신부에 같은 시간에 도달한다고 가정한다. 제안된 위상차 동기회로는 그림 6과 같이 DFF, DFF 제어 카운터, 2-to-1 multiplexer, multiplexer 카운터, 동기 모서리 결정 발생기로 구성된다.

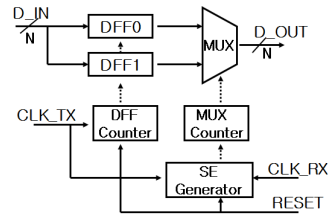


그림 6. 불안정 상태 없는 위상차 동기회로 블록 다이어그램
Fig. 6 Block diagram of metastability-free mesochronous synchronizer

본 논문에서는 송신부의 데이터를 저장하는 입력 버퍼로 2개의 DFF를 사용한다. 그리고 송신부에서 데이터와 함께 전송된 클럭은 비중첩 클럭을 만드는 과정에서 지연이 발생하므로 DFF의 제어 클럭 신호로 사용이 가능하다. 2개의 DFF에 사용되는 클럭은 2개의 비중첩 클럭으로 그림 7(a) 회로로부터 발생되며, (b)의 파형을 가지게 된다.

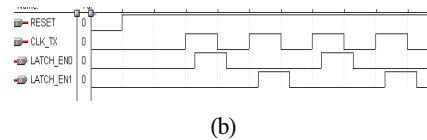
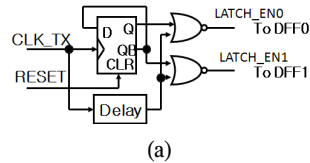


그림 7. DFF 제어신호발생기 및 출력파형
(b) 출력 파형 (a) DFF 제어신호 발생기
Fig. 7 DFF control signal generator and Waveform
(a) DFF control signal generator (b) Output waveform

위상차 클럭 동기회로의 가장 핵심이 되는 SE 선택회로는 CLK_TX의 위치에 따라 CLK_RX의 동기 모서리를 선택해야 한다. 그림 8(a)는 SE 신호를 결정하는 회로이며, (b)는 불안정상태를 제거하기 위해 CLK_RX 모서리를 선택하는 회로이다. CLK_RX가 DFF의 입력 신호이고, CLK_TX가 에지트리거 클럭으로 사용되고 있다. 두 신호가 거의 동시에 발생하면 불안정상태가 발생할 수 있고, 그 출력은 예측할 수 없으며, 경우에 따라서는 1 또는 0이 아닌 다른 값을 가질 수도 있다. 본 논문에서 제안

하는 회로는 3장에서 설명한 것과 같이 1 또는 0 중에서 어떤 값을 출력하더라도 정상적으로 동작할 수 있다. 그리고 2개의 플립플롭 사이에 AND 게이트를 삽입하여 플립플롭의 출력이 중간 값을 가질 경우에도 두 번째 플립플롭의 입력에 1 또는 0의 값이 입력될 수 있도록 설계하였다. 또한 처음에 선택된 CLK_RX의 모서리가 다음 클럭에서 불안정상태 때문에 변경될 수도 있으므로 두 번째 클럭부터는 래치의 enable 단자의 입력을 0으로 만들어서 다시 SE 모서리 선택을 하지 않도록 지연회로 (Delay + Disable signal)를 삽입하였다. 즉 RESET 신호가 발생하지 않는 한 래치의 enable 단자의 입력 값이 변경되지 않도록 하는 회로가 필요하다.

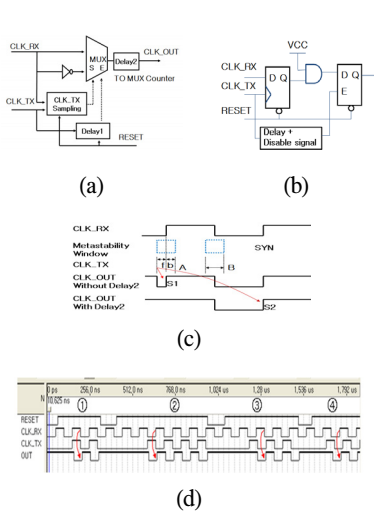


그림 8. SE 발생기 및 파형. (a) SE 발생기 (b) 단순화된 샘플링 로직 (c) Delay1이 있는 것과 없는 것의 CLK_OUT (d) 파형
 Fig. 8. SE Generator and Waveform. (a) SE Generator (b)Simplified sampling logic (c) CLK_OUT with and without Delay1 (d) Waveform

그림 5에서 MW A의 앞부분 f 시간 영역에서 불안정상태가 발생한 경우를 가정하자. DFF의 출력이 0인 경우는 정상 동작한 경우이며, 출력이 1인 경우는 불안정상태 때문에 출력이 다르게 나타난 경우이다. 이 때 SE 발생기는 CLK_RX의 현재 주기보다 1 주기 앞에서 CLK_TX가 발생한 것으로 인식하여 (c)의 delay2가 없는 CLK_OUT 파형과 같이 현재 주기의 상승모서리를 SE로 결정하는 오류를 발생하게 된다. 따라서 MUX의 출력에 Delay2 회로가 추가되어야 한다.

Delay2가 있는 CLK_OUT 파형은 지연 회로를 추가한 회로의 출력 파형이다. 그리고 MWA의 b 기간에 불안정상태가 발생하여 DFF의 출력이 0이 된 경우에는 바로 다음 하강 모서리가 SE로 결정되므로 위상차 클럭 동기회로에서 사용할 수 있는 최대 클럭의 주기는 식 (5)와 같이 결정된다. (d)는 CLK_RX와 CLK_TX의 위상차가 $0^\circ < PD \leq 180^\circ$ 인 경우(②,④)에 CLK_RX가 SE로 결정되고, $180^\circ < PD \leq 360^\circ$ 인 경우(①,③) CLK_RX_bar가 SE로 결정되는 것을 보여준다.

그림 9는 1비트 데이터에 대한 위상차 클럭 동기회로와 로직 시뮬레이션 결과를 보여준다.

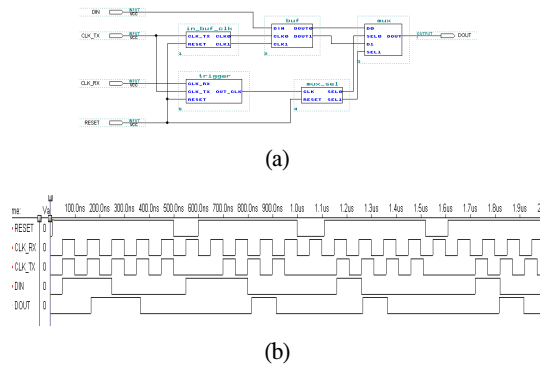


그림 9. 위상차 클럭 동기회로 및 시뮬레이션 결과
 (a) 위상차 동기회로 블록 다이어그램
 (b) 로직 시뮬레이션 결과

Fig. 9 Mesochronous synchronizer and waveform
 (a) Block diagram of mesochronous synchronizer
 (b) Logic simulation result

V. 시뮬레이션 결과 및 고찰

로직 시뮬레이션 과정 동안에 불안정상태를 발생시키기 어려우므로 회로에 고장을 삽입하여 불안정상태가 발생하는 것과 같은 효과를 낼 수 있다. 그림 10(a)는 CLK_RX와 CLK_T가 40° 위상차를 가지고 있으며 불안정상태가 발생하지 않은 경우의 시뮬레이션 결과이다. 즉 CLK_RX가 1(high)인 상태에서 CLK_TX의 상승모서리가 발생하여 CLK_RX가 1로 정상적으로 인식된 경우이다. CLK_TX의 상승모서리가 발생한 다음 CLK_RX 상승모서리에서 OUT_CLK의 상승모서리가 발생한다. 이 파형은 CLK_RX에 비하여 지연이 발생하지만 같은

파형을 가지고 있어 위상차 클럭 동기회로가 정상적으로 동작하고 있음을 알 수 있다. (b)는 회로에 고의로 고장을 삽입하여 불안정상태를 발생시켜 시뮬레이션 한 결과이다. 즉 40° 위상차를 가지고 CLK_RX가 1인 상태에서 CLK_RX의 상승모서리가 발생하였지만 불안정상태가 발생하여 CLK_RX가 0(low)로 인식된 경우이다. CLK_RX의 상태가 1이지만 0으로 인식된다. 그러나 본 논문에서 제안한 위상차 클럭동기회로는 CLK_RX의 현재 보다 반 주기 앞의 0 상태로 인식하여 OUT_CLK는 CLK_TX가 발생한 바로 다음 CLK_RX의 하강모서리에 동기되어 출력 클럭을 만들어 정상적으로 동작하게 된다.

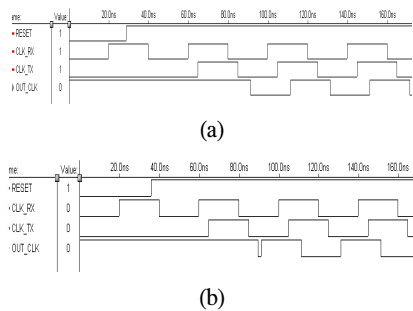


그림 10. 위상차가 40° 일 때 불안정상태가 없는 경우와 발생한 경우의 출력 클럭 파형
 (a) 40° 위상차에서 불안정상태 없는 OUT_CLK 파형
 (b) 40° 위상차에서 불안정상태 있는 OUT_CLK 파형
 Fig. 10 Waveform of OUT_CLK with and without metastability by 40° phase difference
 (a) Waveform of OUT_CLK without metastability by 40° phase difference (b) Waveform of OUT_CLK with metastability by 40° phase difference

그림 11은 4 bit의 입력 데이터를 가진 위상차 클럭 동기회로 고의로 불안정상태를 발생시켜 시뮬레이션한 결과이다. (a)/(b)는 CLK_RX가 1/0인 상태에서 CLK_TX의 상승모서리가 발생하였지만 불안정상태가 발생하여 CLK_RX가 0/1으로 인식되었을 경우에 대한 시뮬레이션 결과를 보여준다. 비록 불안정상태 때문에 CLK_RX의 1/0인 상태에서 0/1으로 인식되었지만 바로 앞의 CLK_RX의 0/1 상태에서 정상적으로 인식된 것과 같은 결과를 얻을 수 있다. 즉 DOUT은 지연을 가지고 CLK_RX의 는 하강/상승 모서리에서 DIN과 같은 결과 값을 보여준다.

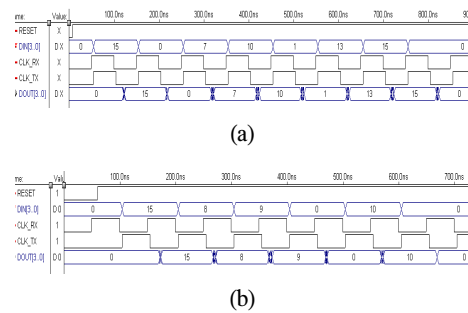


그림 11. 20°, 200° 위상차를 가진 4비트 입력에 대한 MS의 로직시뮬레이션 (a) PD=20° (b) PD=200°
 Fig. 11 Logic simulations for MS with 4-bit input by 20° and 200° phase difference (a) PD=20° (b) PD=200°

표 1은 기존에 발표된 위상차 클럭 동기회로와 본 논문에서 제안한 회로에서 중요한 특징들을 비교한 것이다.

표 1. 본 논문의 제안한 회로와 기존 회로 비교
 Table. 1 Comparisons between the proposed and conventional synchronizers

	본 논문	[12]	[20]	[21]
Metastability-free synchronizer	Y	Y	N	N
phase detector	N	Y	Y	N
360° Coverage of phase difference	Y	Y	Y	Brute-force
DLL	N	Y	N	N

VI. 결론

설계규칙의 감소는 칩내에서 상호연결, 신뢰도, 전력 소모 등 기존에 예측하지 못했던 많은 어려움을 발생시키고 있다. 특히 MPSoC 등에서는 상호연결과 버스 대역폭이 칩 성능의 병목현상이 되고 있으며, 이를 해결하기 위한 방안으로 NoC가 연구되고 있다. 클럭 스쿠 문제를 해결하기 위한 NoC의 클럭 시스템으로 GALS와 동기 회로가 연구되고 있다.

본 논문에서는 NoC의 클럭 시스템으로 사용될 수 있는 불안정 상태가 없는 위상차 클럭 동기회로를 설계하였다. 비동기 또는 위상차 회로에서 발생할 수 있는 불안정 상태를 해결하기 위해 동기 모서리를 정의하고, 송수

신부 클럭 위상차에 따라 자동으로 동기 모서리를 선택하는 방법을 제안하였다. 그리고 고장을 삽입한 로직 시뮬레이션을 통하여 0°~360° 위상차에서 불안정 상태에 관계없이 클럭 동기회로가 잘 동작함을 확인하였다. 제안한 회로는 위상 검출기가 필요하지 않아 제어가 간단하고, 모든 회로가 디지털 회로로 구성되어 NoC의 클럭 동기회로에 사용할 수 있을 것이다.

참고문헌

- [1] Mark Bohr, "The New Era of Scaling in an SoC World," ISSCC 2009, pp. 23-28, Feb. 2009.
- [2] ISSCC 2011 TRENDS REPORT, <http://isscc.org/>, ISSCC, 2011.
- [3] Reid J. Riedlinger, et al., "A 32nm 3.1 Billion Transistor 12-Wide-Issue Itanium Processor for Mission-Critical Servers," ISSCC 2011, pp. 84-86, Feb. 2011.
- [4] Nikil Dutt, et al. "Tutorial 5: SoC Communication Architectures : Technology, Current Practice, Research, and Trends," IEEE 20th Int'l Conference on VLSI Design, 2007.
- [5] Francesco Vitullo, et al., "Low-Complexity Link Microarchitecture for Mesochronous Communication in Networks-on-Chip," IEEE Trans. on Computers, Vol. 57, No. 9, pp. 1196-1201, Sep. 2008
- [6] Sudeep Pasricha, Nikill Dutt, On-Chip Communication Architecture, Morgan Kaufmann, 2008.
- [7] Giovanni De Micheli, Luca Benini, Networks on Chips, Morgan Kaufmann, 2006.
- [8] Daniel M. Chapiro, Globally-Asynchronous Locally-Synchronous Systems, Ph. D. thesis, Stanford University, Oct. 1984.
- [9] Jens Mutersbach, et al., "Globally-Asynchronous Locally-Synchronous Architectures to Simplify the Design of On-Chip Systems," 12th International ASIC/SOC Conference, pp. 317-321, Sep. 1999.
- [10] Francesco Vitullo, et al., "Low-Complexity Link Microarchitecture for Mesochronous Communication in Networks-on-Chip," IEEE Trans. on Computers, Vol. 57, No. 9, pp. 1196-1201, Sep. 2008
- [11] Behzad Mesgarzadeh, Christer Svensson, Atila Alvandpour "A New Mesochronous Clocking Scheme for Synchronizatoin in SoC," ISCAS 2004, pp. II-605-II-608, 2004.
- [12] W. J. Dally, J. W. Poulton, Digital System Engineering, Cambridge Univ. Press, 1998.
- [13] Mahdi Nazm Bojnordi, et al., "An Efficient Clocking Scheme for On-Chip-Communications," APCCS 2006, pp. 119-122, 2006
- [14] A. Edman, C. Svensson, "Timing Closure through Globally Synchronous, Timing-Portioned Design Methodology," Proc. 45th DAC, pp. 71-74, 2004.
- [15] Peter Cauputa, Christer Svensson, "An On-Chip Delay- and Skew-Insensitive Multi-cycle Communication Scheme," ISSCC 2006, pp. 24.5, 2006.
- [16] Y. Semiat, R. Ginosaur, "Timing Measurements of Synchronization Circuits," Proc. Ninth Int'l Symp.. Advanced Research in Asynchronous Circuits and Systems, pp. 68-77, May 2003.
- [17] Ingemar Soderquist, "Globally Updated Mesochronous Design Style," IEEE Journal of Solid-State Circuit, Vol. 38, No. 7, Jul. 2003
- [18] Igor Loi, er al., "Developing Mesochronous Synchronizers to Enable 3D NoCs," 2008 EDAA, pp. 1414-1419, 2008.
- [19] Daniele Ludovici, er al., "Comparing Tightly and Loosely Coupled esochronous Synchronizers in a NoC Switch Architecture," 2009 NOCS, pp. 2009.
- [20] Ran Ginosar, "Fourteen Ways to Fool Your Synchronizer," ASYNC03, pp. 1-8, 2003.

저자소개

김강철(Kang Chul Kim)

한국정보통신학회 논문지
제14권 제10호 참조