SoC에 적용 가능한 GALS(Globally Asynchronous Locally Synchronous) 시스템용 고성능 wrapper 설계

오명훈*, 김춘호**, 박석재**, 최호용**, 이동익*
*광주과학기술원 정보통신공학과, **충북대학교 반도체공학과
*광주광역시 북구 오룡동 1번지, **충북 청주시 흥덕구 개신동 산 48번지
전화 : 062-970-2248 Fax : 062-970-2204
Email : {mhoh, dilee}@kjist.ac.kr, hychoi@cbucc.chungbuk.ac.kr

요약

본 논문에서는 SoC에 적용할 수 있는 GALS (Globally Asynchronous Locally Synchronous) 시스템을 위한 접속장치인 wrapper를 설계한다. 종래의 pausible clocking 방법에 기반한 wrapper와는 달리, 단순하고 견고한 구조로 고성능 데이터 전송 메커니즘을 수행한다. 본 wrapper 설계에서는 내부 클록과 외부 데이터 통신의 병렬수행이 가능한 구조를 가지고, 비동기셀을 사용하여 구조의 단순화를 꾀한다. 또한, 비동기 합성 툴을 사용하고 트랜지스터 수준에서 최적화 설계를 한다. 설계 결과, 종전의 같은 병렬 데이터 전송 메커니즘을 지원하는 wrapper에 비해 본 논문에서 제안된 wrapper는 65.2%의 면적 감소와 12.1%의 latency의 단축을 얻었다.

1. 서론

최근, 공정 기술과 집적회로설계 기술의 발전으로 보드 수준에서 구현되었던 시스템이 한 개의 칩으로 구현되는 시스템온칩(SoC : system-on-chip) 구현이 가능하게 되었다. 또한, 설계 및 검증에 필요한 시간과 노력을 줄이기 위해서, 기능 및 성능이 검증된 설계 데이터, 즉 IP(intellectual property)를 재활용하는 설계방법을 도입하여, 시스템온칩 설계생산성을 획기적으로 향상시키려는 설계방법이 주목을 받고 있다[1].

그러나, IP를 이용한 SoC 설계에 있어서 기존의 전역 클록을 이용한 동기식 설계 기법을 사용하는 경우, 클록 속도 증가에 따른 클록 스큐(skew)와 지터(jitter) 문제를 해결해야하고, 클록 배분을 위한 전력 소모가 증가하게 된다. 더구나, 소자의 지연시간에 비해 상대적으로 더 늘어난 전송선로의 지연시간을 고려하여 설계하여야 하고, IP간 클록 주파수의 차이에 기인한 설계 시간의 증가로 시장요구에 빠른 대응이 곤란하다.

한편, 비동기식 설계 기법은 전역 클록을 사용하지 않고, 데이터 전송을 지연시간에 무관한 핸드세이크 프로토콜(handshake protocol)에 의해서 수행한다는 점에서 이러한 문제점들을 해결할 수 있는 대안으로 제시될 수 있다. 하지만, 비동기 회로부분의 규모가 커질 경우에는 설계 복잡도가 증가하고, 테스팅이 쉽지 않으며, 설계를 뒷받침할 비동기 CAD 툴이 부족하다.

비동기식 설계 기법의 단점을 보완하고 동기식 설계 기법의 문제점을 아키텍처 상에서 보다 근본적으로 해결할 수 있는 방안으로 GALS (globally asynchronous locally synchronous) 시스템이 제시되고 있다[2]. GALS 시스템은 기본적으로 전역적(globally) 단일 클록을 사용하지 않고, 서로 독립적인 클록에 의해 동작하는 여러 개의 소규모 LS (locally synchronous) 모듈로 구성되며, 모듈간의 데이터 전송은 특화된 접속장치(wrapper)를 통해서 비동기 핸드세이크 프로토콜에 의해 수행된다. 그러므로, SoC 측면에서 GALS 시스템은 다음과 같은 장점을 갖는다.

- 전역 클록을 사용하지 않으므로 클록 스큐, 지터, 전력소비 문제의 해결
- IP의 이식성과 재사용성의 증가
- 다른 타이밍을 가진 IP사이에서 비동기 프로토콜에 의한 안정된 데이터 전송
- 대규모의 비동기 회로 설계 시 발생하는 문제점의 최소화

이러한 GALS 시스템에서는 LS 모듈사이에서의 데이터 전송 시 LS 모듈과 데이터 간의 동기화 문제를 해결하고, 비동기 핸드세이크 프로토콜을 생성하는 wrapper는 GALS 시스템의 핵심 회로이다. 또한, 모든 데이터가 wrapper를 통해서 전송되기 때문

본 연구는 정보통신부의 정보통신기초기술 연구지원사업과 IDEC의 지원으로 수행되었음.

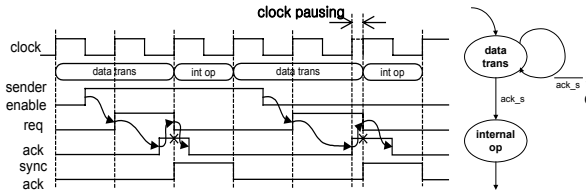


그림 1. UNV 방식의 데이터전송 메커니즘

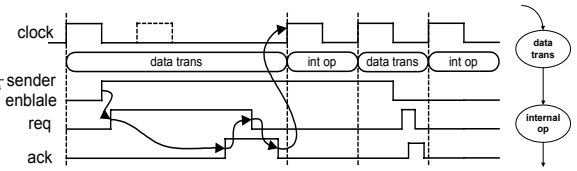


그림 2. KNV 방식의 데이터전송 메커니즘

에 wrapper 회로의 성능은 전체 시스템의 성능에 직접적인 영향을 미치게 되므로 효율적인 GALS 시스템을 구현하기 위해서는 고성능의 wrapper 회로 개발이 시급하다.

본 논문에서는 내부 클록과 외부 데이터 통신의 병렬 수행을 지원하고 개선된 pausible clocking 방법에 기반한 wrapper를 설계한다. 구조상에서 병렬데이터 전송 메커니즘을 지원하고, 구현상에서 트랜지스터 수준의 최적화 설계를 하여, 면적 감소 및 성능 개선을 기대할 수 있다.

2. Pausible clocking 방법

GALS 시스템의 LS 모듈간의 접속장치에 관한 연구는 주로 비동기적으로 발생하는 외부 입력 신호와 내부 클록과의 동기화(synchronization) 실패 문제를 해결하기 위한 인터페이스 회로설계를 중심으로 수행되어왔다. 그 중에서도 링 오실레이션 방식[3]에 기반한 pausible clocking 방법[4][5]이 활발히 연구되고 있다.

이 방법은 입력신호와 내부 클록 신호사이의 준안정 상태(metastable state)가 발생했을 때 내부 클록을 정지시켜 동기화 문제를 해결하는 방식으로, 클록 발생을 억제하는 방법에 따라 UNV (unknown next value) 와 KNV (known next value)의 2가지로 구분할 수 있다.

4-위상(phase) 핸드세이크 프로토콜[6]을 지원하는 wrapper를 장착한 센더(sender) LS 모듈에서의 동작을 예제로 설명하면 다음과 같다. 먼저, 첫번째 방식은 그림 1의 파형도처럼 데이터 전송을 요구하는 *sender enable* 신호에 의해서 내부 클록에 동기화되어 *req* 신호가 발생된 후, *ack* 신호를 다음 클록으로 sampling하여 인지하는 방법이다. 준안정 상태 발생 시에는 클록을 잠시 동안 정지시켜 *ack* 신호를 강제로 sampling 하여 클록에 동기화된 *sync ack* 신호를 LS 모듈로 보내준다. 부분 FSM이 나타내고 있는 것처럼, LS 모듈은 데이터 전송 상태에서 이 *sync ack* 신호를 판별하여 계속 기다릴 것인지 아니면 다음 상태로 천이 할 것인지를 결정하게 된다. 이 방식에서는 *req* 신호 발생 후, 다음 클록 발생 시점에서 외부 ack 신호 값을 알지 못하므로 여기서는 편의상 UNV 방식이라고 부른다.

두번째 방식은 그림 2와 같이 *sender enable* 신호에 의해 *req* 신호가 발생되면, 클록을 정지시키고 외부 핸드세이크 프로토콜이 종료되었을 때 클록을 발생시키는 방법으로 애초에 클록과 *ack* 신호의 준 안정 상태를 피할 수 있다. *req* 신호 발생 후에 정지된 클록이 다시 발생하는 시점에서는 핸드세이크 프로토콜이 끝난 상태이므로 항상 *ack* 신호는 '0'을 유지하게 된다. 그래서, 두번째 방식을 UNV 방식에 대응하여 KNV 방식으로 명명한다.

UNV 방식을 기본으로 한 wrapper인 PCC (pausible clocking control) 회로 [7]에서는 처음으로 pausible clocking 방법에 여러 입력 신호들을 중재하는 기능을 구현하였다. 그러나, 준안정 상태를 검출할 수 있는 회로가 필요하고, 중재회로(arbiter)가 특정시간에 오직 하나의 입력만을 처리하므로 여러 개의 입력신호를 동시에 처리하지 못하는 구조이다. 또한, 핸드세이크 프로토콜이 종료되기를 기다리는 동안에 클록은 계속 발생되므로 불필요한 전력을 소모한다는 단점을 지니고 있다.

KNV 방식으로 분류 될 수 있는 [8]의 D-type wrapper는 리시버의 응답 시간이 길어지게 되면, 센더에서는 무조건 그 만큼의 시간동안 클록을 정지시켜야 하므로, LS 모듈의 내부 연산과 데이터의 전송이 병렬적으로 수행되지 못함으로써, 전체 시스템의 성능을 저하시킬 수 있는 단점을 갖는다.

두 방식의 단점을 보완하는 기술로 KNV 방식을 기반으로 LS 모듈의 데이터 전송 상태 이후의 다음 상태가 데이터 전송과는 무관한 내부 동작 상태이면, 클록을 정지시키지 않으므로써 내부 클록과 외부 핸드세이크 프로토콜을 부분적으로 분리시킨

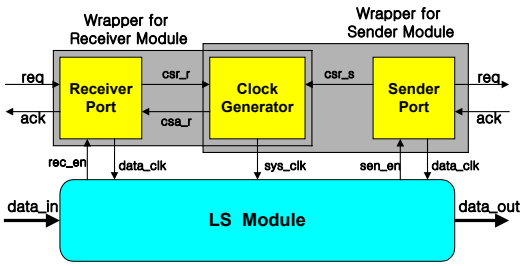


그림 3. 제안된 wrapper의 구조

decoupled형 wrapper가 연구되었다[9]. 그러나, 내부 AFSM(asynchronous finite state machine)을 사용한 제어 회로의 제약 조건을 위해 부가적인 회로가 필요하고, 복잡한 지연소자 조건을 계산해야 한다. 또한, 게이트 수준에서 표준셀로만 설계되었기 때문에 성능상 최적화되지 못했다.

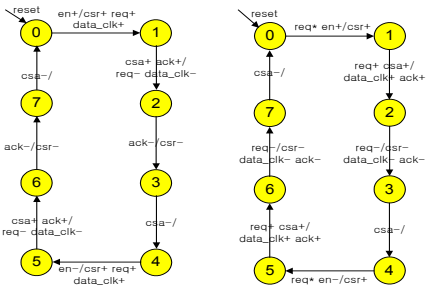
본 논문에서는 pausible clocking 방법을 사용하는 decoupled 형 wrapper를 기반으로 단순하면서도 견고한 구조의 wrapper를 제안하고, 이를 비동기 합성 툴을 이용하여 트랜지스터 수준에서 최적화하여 설계한다.

3. wrapper의 기본 구조

본 논문에서 언급하는 wrapper는 4-위상 핸드세이크 bundled 데이터 방식과 *active-output-passive-input* 형태의 푸쉬 채널 [6]를 가정하고, 구조가 간단한 링 오실레이션 방식을 사용하여 내부 클록을 발생시킨다. 프로토콜 상에서 성능을 높이기 위해 wrapper의 센더 부분에서 내부 클록과 외부 데이터 통신이 서로 병렬 수행 가능한 데이터 전송 메커니즘을 지원한다. 또한, wrapper의 구조를 표준셀이 아닌 비동기 셀을 사용하여 최적화 함으로써 성능과 면적의 개선을 꾀한다

3.1. wrapper의 상위 구조

그림 3에서 보는 바와 같이, 기본적으로 제안된 wrapper의 구조는 크게 출력 데이터의 전송을 수행하는 센더 모듈(**Sender Module**)과, 들어오는 입력 데이터를 저장하기 위한 리시버 모듈(**Receiver Module**)로 나뉘어 진다. 다시, 각각의 모듈들은 내부 클록 제어용 신호들을 생성하고 외부와의 통신을 수행하는 포트(Port) 블록과, pausible 클록을 발생시키는 클록 발생기(Clock Generator) 블록을 가지고 있다. 동기 모듈은 오직 하나의 클록에 의해 작동되므로, 다중 포트를 갖는 경우에 각 포트 블록들은 클록 발생기 블록을 공유하게 된다.



(a) (b)

그림 4. normal 형태의 센더 포트 AFSM (a)
리시버 포트 AFSM (b)

동기 모듈에서 데이터를 받아야 하는 경우나, 데이터를 출력해야 하는 시점에서 내부 클록에 동기화된 신호(*rec_en, sen_en*)를 wrapper에 보내면, Port 블록에서는 제어 신호(*csr:clock stop req*)를 생성하여 내부 클록을 정지시킨다. 동시에, 데이터의 저장과 출력을 위해 동기 모듈 맨 끝단의 래치를 구동하는 신호(*data_clk*)를 만든 후에, 핸드세이크 프로토콜을 수행한다. 외부와의 통신이 끝나게 되면, 각 포트 블록에서는 클록을 정지시킨 신호와 래치 구동 신호를 해제하여 내부 클록을 다시 작동시키고, 다음 데이터 전송을 준비하게 된다.

본 논문에서는 [9]에서처럼, 전통적인 KNV 방식을 따르는 normal 형의 센더 포트 및 리시버 포트와 개선된 구조의 decoupled 형 센더 포트를 설계한다. 다음에서 각각의 포트에 대해서 설명하고 클록 발생기에 대해서는 4장에서 설명한다.

3.2. normal 형의 센더/리시버 포트

그림 4는 AFSM으로 기술된 normal 형의 센더와 리시버 포트를 나타낸다. 모두 데이터를 전송하는 핸드세이크 프로토콜 수행 동안은 *csr* 신호에 의해서 클록이 정지되어 있음을 알 수 있다.

AFSM은 비동기 제어 회로의 스펙을 기술하는 방법중의 하나로 동기식 Mealy형 동기 순서 회로의 동작과 유사하다. 본 논문에서 사용된 AFSM은 burst-mode 방식 [10]에 기반하여, 주어진 상태에서 모든 입력 신호의 변화가 만족되면, 출력 신호를 발생시키고 내부 신호가 안정된 후에 다음 상태로 천이하게 된다.

decoupled 형의 리시버 포트는 normal 형의 그것과 구조상 동일하지만, 센더 포트는 [9]에서 보다 훨씬 더 간단한 구조를 갖는다.

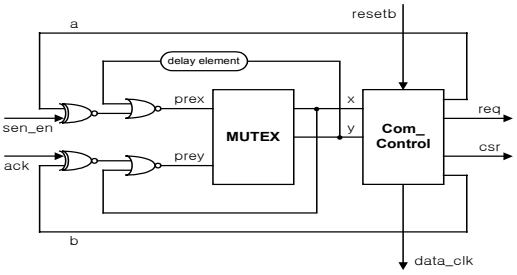


그림 5. decoupled 형태의 센더 포트 블록

3.3. 개선된 decoupled 형 센더 포트

전통적인 KNV 방식의 센더 모듈에서는 4위상 핸드세이크 프로토콜을 활성화 시키는 *sen_en* 신호 발생시 내부 클록을 정지시켜 (1)과 같은 입출력 신호의 순서를 유지시킨다.

sen_en+->req+->ack+->req-->ack-->sen_en-->req+...

← clock stop → (1)

그러나, 병렬 데이터 전송 메커니즘을 위한 센더 모듈에서는 핸드세이크 프로토콜을 시작할 때마다 내부 클록을 정지시키지 않으므로, 처음 *sen_en* 신호에 의해 활성화된 *req+* 신호를 보낸 후에, 프로토콜 종료를 의미하는 *ack-* 신호를 인지하기 전까지 다시 *sen_en* 신호 발생을 허용하므로, 다음과 같은 3가지 패스를 수행할 수 있다.

sen_en+->req+->ack+->req-->ack-->sen_en-->req+...

← clock no stop → (2)

sen_en+->req+->sen_en-->ack+->req-->ack-->req+...

← clock stop → (3)

sen_en+->req+->ack+->req-->sen_en-->ack-->req+...

clock stop ← → (4)

(2)는 *sen_en* 신호로부터 생성된 핸드세이크 프로토콜이 종료 후에 다시 *sen_en* 신호가 발생되어, 다음 핸드세이크 프로토콜을 수행하는 경우이다. (3)과 (4) 처럼, 핸드세이크 프로토콜 종료 전에 또 다른 sen_*en* 신호가 발생되었을 때는, 클록을 정지시키고 핸드세이크 프로토콜이 끝날 때까지 기다린다. *ack-* 신호를 확인하면, 나중에 발생한 *sen_en* 신호에 대한 *req+* 신호를 내보내어 핸드세이크 프로토콜을 다시 시작시킨 후, 클록을 활성화하여 동기 모듈의 동작을 시작시킨다.

기본적으로 병렬 데이터 전송 메커니즘에서는 *ack* 신호와 *sen_en* 신호는 서로 독립적으로 발생되는 것을 허용하므로, 위의 3가지 패스를 올바르게 실행하기 위해서는

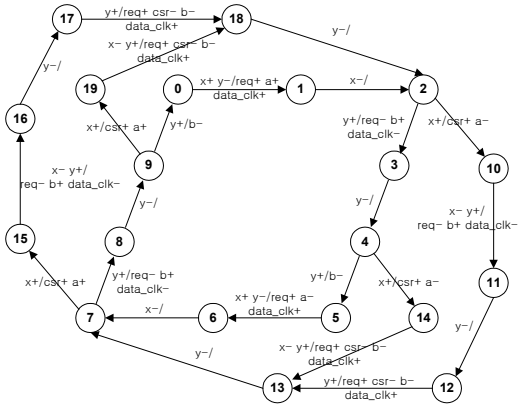


그림 6. **Com_Control** 블록의 AFSM

두 신호사이에 준 안정상태가 발생하지 않아야 한다. 다시 말해서, (*sen_en+*, *ack+*), (*sen_en+*, *ack-*), (*sen_en-*, *ack+*), (*sen_en-*, *ack-*)의 4개의 입력신호 순서쌍에 대해서 두 신호가 짧은 시간 안에 동시에 발생하면 준 안정 상태가 발생할 수 있고, 이때 순서를 결정해 주지 못하면 회로 내부는 오동작을 할 수 있다.

하나의 MUTEX 블록 [3]에, 중재한 결과를 피드백시킨 신호와 각각 *sen_en* 신호, 또는 *ack* 신호를 XOR 연산으로 조합하여 MUTEX 블록의 입력 신호로 사용하면, 4가지 순서쌍 모두에 대해 중재 기능을 수행할 수가 있다. 그림 5는 하나의 MUTEX 블록으로 4 가지 순서쌍의 중재기능을 포함한 센더 포트 블록의 다이어그램을 나타내고 있다. 독립적으로 생성되는 *sen_en* 신호와 *ack* 신호의 변화에 대하여, MUTEX 블록의 입력 신호인 *prex*, *prey* 신호를 logical 1로 만들기 위해, 이전의 *sen_en*, *ack* 신호를 배타적으로 중재한 결과인 *x*, *y* 신호를 **Com_Control** 블록에서 재생성하여 피드백시킨 *a*, *b* 신호를 사용한다.

3.3.1. Com_Control 블록

MUTEX 블록의 중재 기능을 위한 *a, b* 신호를 발생시키는 역할 이외에도, **Com_Control** 블록은 데이터 전송에 필요한 *req* 신호와 *data_clk* 신호를 생성하여 외부 LS 모듈과 통신하고, *csr* 신호를 사용하여 클록 정지 유무를 결정하는 기능을 수행한다. 그림 6은 **Com_Control** 블록의 AFSM 기술을 나타내고 있다. 초기 상태 ⓪에서 ⑤까지가 위에서 설명한 패스 (2)에 해당되고, ⓪에서 ⑩을 거쳐 ⑬까지의 패스가 (3)에 해당된다. (4)의 패스는 ⓪에서 ③을 거쳐, ⑭, ⑬를 거친 패스이다.

일반적으로 AFSM을 올바르게 동작시키기 위해서는, 다음과 같은 두 개의 설계 제약조건을 만족시켜야 한다[10].

■ *fundamental-mode environmental constraint* : 두 신호 중 하나의 신호에 의해 발생하는 출력 신호가 안정화되기 전에 다른 나머지 신호가 발생되지 않아야 한다.

■ *distinguishability constraint* : 주어진 한 상태에서 두개 이상의 상태로 분기되는 곳에서의 입력 신호들은 그 중 오직 한 개의 상태만으로 천이될 수 있도록 입력 신호들이 발생되어야 한다.

MUTEX 블록에서 *sen_en*과 *ack* 신호를 중재하여 *x, y* 신호를 서로 배타적으로 발생시키므로 *distinguishability constraint* 를 만족시킬 수가 있다. 그러나, 그림 6의 AFSM에서 상태천이 ⓪→①→②의 예처럼, 상태 ① 이후 내부 회로가 안정화 되기 전에 *sen_en-*에 의해 다시 *x+* 신호가 발생할 수 있으며, 마찬가지로, *req+*에 대응하는 *ack+* 신호가 상태 ② 이전에 발생하면 *x-* 이후의 내부회로 안정화 시간 중에 *y+*이 발생가능하다. 이 두 가지 경우 모두에 대하여 회로 동작의 안정성을 보장할 수 없게 된다. *fundamental-mode environmental constraint*를 만족시키기 위해서는 *sen_en+*와 *sen_en-* 신호 사이의 시간, 다시 말하면, 이들 두 신호는 내부 클록과 동기화되어 발생하므로 최소 클록 주기가 존재해야 한다. 마찬가지로 *req* 신호와 이에 따르는 외부 *ack* 신호 사이에도의 최소한의 지연시간이 필요하다.

4. wrapper의 설계 및 시뮬레이션

그림 7은 그림 3과 같은 2개의 포트를 갖는 wrapper에서의 클록 발생기를 나타내고 있다. 홀수개의 인버터로 구성된 링 오실레이터로 내부 클록(*sys_clk*)을 생성시킨다. 클록 정지 시에 각 포트에서 발생되는 신호들(*csr_s, csr_r*)이 동시에 발생할 때에도 모두 클록에 동기화되어 발생하므로, 내부 신호의 해저드 없이 OR 연산을 통해서 한꺼번에 처리 할 수 있도록 설계하였다.

3.2절의 normal 형 센더 포트와 리시버 포트, 그리고 개선된 구조의 decoupled 형 센더 포트 내의 AFSM 합성은 burst-mode 기반 합성 툴인 3D [10]를 이용하였다. 3D는 각각의 출력신호에 대해서 set, reset 식

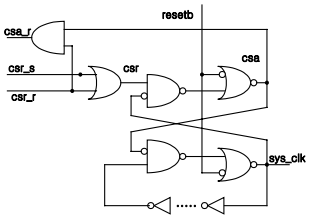


그림 7. 클록 발생기

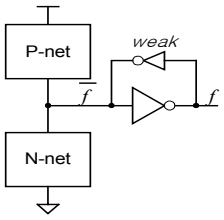


그림 8. gC 회로

을 발생시킨다. 그림 8과 같은 gC (generalized C-element) [11]를 사용하여, 각각 set, reset 식에 대응하여 N-net와 P-net를 통해 설계하였다. 본 논문에서 제시하는 개선된 구조의 decoupled 형 센더 포트를 위해, gC를 이용한 구현은 하이닉스 0.35㎛ 공정의 라이브러리를 사용하여 트랜지스터 수준(adv_gC라 명함)에서 설계하였다. 또한, 성능 및 면적 비교를 위해서, 2-레벨 AND-OR 게이트 형태로 3D의 합성결과를 추출하여 이를 아남 0.25㎛ 공정을 사용한 표준셀의 게이트 레벨(adv_AOG라 명함) 및 0.35㎛ 공정의 트랜지스터 수준(adv_AOT라 명함)에서 설계하였다.

3장에서 언급하였던 센더 포트의 설계 제약조건은 내부 클록의 경우에는 최소 클록 주기는 8ns, 외부 핸드세이크 프로토콜의 최소 지연시간은 약 1.56ns로 0.35㎛ 공정을 고려할 때, 적절한 제약조건으로 생각할 수 있다.

같은 decoupled 형태의 데이터 전송 메커니즘을 수행하지만, [9]에서 제안한 것에 비해 개선된 구조로 설계된 센더 포트의 아키텍쳐상의 성능 향상 기여도를 측정하기 위해, 동일한 one-to-one 환경에서 게이트 레벨 시뮬레이션을 수행하였다. 표 1은 [9]에서 제시한 센더 포트(dec_AOG)와 개선된 구조의 게이트 레벨 센더 포트(adv_AOG) 각각에 대하여 throughput을 측정한 뒤, normal 형의 센더 포트에 비해서 성능 향상 정도를 표시하고 있다.

시뮬레이션 환경의 전송 데이터 수에 따라서 adv_AOG를 장착했을 때 더 좋은 성능 향상 정도를 보였다. 이는 본 논문에서 설계한 센더 포트는 AFSM 제약조건을 만족시켜 주기 위해 부가적으로 필요한 회로와 지연소자가 필요없고, 비동기셀인 MUTEX 블록을 사용하여 더 최적화된 구조로 설계되었기 때문이다.

표준셀로 게이트 수준에서 설계한 것(dec_AOG, adv_AOG)에 비해, gC를 사용하여 트랜지스터 수준에서 설계한 것(adv_gC)

표 1. 개선된 센더 포트의 성능향상 비교(%)

전송 데이터수	10	25	50	100	250	500	1000
dec_AOG	8.5	10.3	9.9	9.7	9.9	9.0	8.3
adv_AOG	11.1	11.4	11.0	11.0	11.0	10.3	9.4

dec_AOG : [9]에서 제시한 센더 포트
adv_AOG : 개선된 센더 포트

의 구현상의 이득 정도를 표 2에서 나타내고 있다.

같은 스펙을 수행하는 센더 포트에 대해서 [9]에서 제안된 구조의 게이트 레벨 구현(280.5)보다 개선된 구조의 게이트 레벨 구현(233.5)이 게이트 수가 더 적었고(16.8%감소), 이를 트랜지스터로 구현(852)하는 것 보다 gC를 이용한 구현방법(356)이 면적측면에서 개선되었다(58.2%감소).

또한, 입력 신호인 *sen_en*과 *ack*로부터 출력 신호 *req*가 발생되는 시간을 측정한 내부 latency도 [9]에서 제시된 센더 포트보다 개선된 구조의 센더 포트가 5.9% 개선되었다. 그리고, 이를 트랜지스터 수준에서 구현한 방식보다 gC로 구현한 방식이 6.6% 개선되어 내부 지연시간이 적게 소모됨을 알 수 있었다.

adv_AOG 및 adv_AOT의 결과를 통해 구현되지 않은 [9]의 decoupled 방식의 트랜지스터 수준 설계 데이터 결과를 환산하고, 이를 개선된 구조의 gC 버전과 비교하였을 때, 면적 측면에서는 65.2%, 내부 latency 측면에서는 12.1%의 감소 효과를 얻을 수 있었다.

5. 결론

본 논문에서 제시한 wrapper는, 내부 클록과 외부 프로토콜의 동기화 문제를 해결하면서, 서로 병렬적으로 수행될 수 있는 데이터 전송 메커니즘을 지원하므로 종래의 pausible clocking 방식보다 프로토콜상에서 고성능의 이점을 갖는다. 또한, 병렬 데이터 전송 메커니즘을 위해 비동기 셀을 사용한 개선된 wrapper의 구조로 보다 간단한 구조이면서도 성능향상을 도모할 수 있다. 설계 시에는 비동기 합성 툴을 사용하여 기존의 표준셀의 게이트 레벨 방식보다 최적화된 트랜지스터 레벨에서 설계하여 65.2%의 면적 감소와 12.1%의 latency의 단축을 얻었다.

표 2. 설계 방식에 따른 면적 및 latency 비교

	Gate-level(0.25㎛)		Tr-level(0.35㎛)	
	Area	latency	area	latency
dec_AOG	280.5	1.02	*1023	*1.62
adv_AOG	233.5	0.96	--	--
adv_AOT	--	--	852	1.52
adv_gC	--	--	356	1.42
개선정도	16.8%	5.9%	58.2%	6.6%

adv_AOG : 2 레벨 AND-OR 게이트 수준 구현
adv_AOT : 2 레벨 AND-OR Tr 수준 구현
adv_gC : gC 트랜지스터 구현
area : 게이트 수(NAND=1기준), 트랜지스터 수
* : 환산 데이터임

참고문헌

[1] *Hunt, M.; Rowson, J.A.,* "Blocking in a system on a chip," IEEE Spectrum, vol. 33, issue. 11, pp. 35-41, Nov. 1996.

[2] International Technology Roadmap for Semiconductors, Semiconductor Industry Association, 2001.

[3] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980.

[4] M. Pechoucek, "Anomalous Response Times of Input Synchronizers," IEEE Trans. Comput., vol. C-25, no. 2 , Feb. 1976.

[5] D. M. Chapiro, "Globally-Asynchronous Locally-Synchronous Systems," Ph.D. thesis, Stanford University, Oct. 1984.

[6] Furber, S.B. and Day, P., "Four-phase micro pipeline latch control circuits," *IEEE Trans. VLSI Systems,* vol. 4, no. 2, pp. 247 -253, Jun. 1996.

[7] K. Y. Yun and A. E. Dooply, "Pausible clocking based heterogeneous systems," *IEEE Trans. VLSI Systems*, vol. 7, no. 4, pp. 482-487, Dec. 1999.

[8] Jens Muttersbach, et al., "Practical Design of Globally Asynchronous Locally Synchronous Systems," in *Proc. International Symposium on Asynchronous circuits and Systems*, pp. 52-59, Apr. 2000.

[9] 오명훈, 최병수, 이동익, "전역적 비동기, 지역적 동기 시스템용 부분 분할 비동기식 접속장치", *2001 SOC Design Conference*, pp. 348-355, 2001년 11월.

[10] K. Y. Yun, "Synthesis of Asynchronous Controllers for Heterogeneous Systems," Ph.D. thesis, Stanford University, Aug. 1994.

[11] K. Y. Yun, "Automatic synthesis of extended burst-mode circuits using generalized C-elements," in Proc. Design Automation Conference, pp. 290-295, 1996.