

논문 2004-41SD-9-8

비동기 회로의 신호천이 감소를 위한 RZ/NRZ 혼합 2선식 데이터 전송 방식

(RZ/NRZ Mixture mode Data Transmission to reduce Signal Transition
in the Asynchronous Circuits)

이 원 철*, 이 제 훈*, 조 경 록*

(Won-Chul Lee, Je-Hoon Lee, and Kyoung-Rok Cho)

요 약

본 논문에서는 비동기식 회로 설계시 2선식(Dual-Rail) 코드를 사용할 때, 회로 구조에서 갖는 신호 천이를 줄여 소비전력을 감소시키는 RZ/NRZ 혼합 데이터 전송 방식을 제안한다. RZ 방식 2선식 코드는 비동기 회로 구현에 많이 사용되고 있으며, 고정 지연을 사용하는 단선구조와는 달리 데이터를 통하여 신호의 유효성을 판별할 수 있다. 그러나, 단선 구조에 비해 많은 회로 면적과 모든 신호가 Return-to-Zero의 스위칭에 의해 전력 소비를 가져오므로, 신호 천이의 수를 감소시킬 필요가 있다. 본 논문에서는 RZ/NRZ 방식을 제안하여 스위칭을 약 50% 감소시키며 소비전력을 비교한 결과, 기존의 2선식에 비해 약 23% 정도 감소하는 결과를 얻었다.

Abstract

In this paper, we propose a RZ/NRZ mixture data transmission method for the asynchronous circuit design to reduce power consumption. The dual-rail data with RZ decoding scheme is used to design asynchronous circuit, and it is easy to get a completion signal of the data validity from the native data as contrasted with single-rail. However, the dual-rail scheme suffers from large chip area and increasing of power consumption from all signals by the switching of the return-to-zero. We need to diminish number of circuit switching. The proposed RZ/NRZ data transmission reduces a switching activity to about 50% and it shows 23% lower power consumption than the conventional dual-rail coding with RZ's.

Keywords : self-timed circuit, dual-rail data encoding, delay-insensitive,

I. 서 론

비동기식 회로는 시스템 클럭에 의해 전체 모듈이 구동되는 대신 각 모듈의 요구(request)와 확인(acknow

-ledge)의 두 이벤트 신호에 따른 핸드셰이킹 프로토콜을 사용하여 모듈이 구동된다^[1]. 시스템 클럭에 의해 동작 필요 여부와 관계없이 모든 모듈이 구동되는 동기식 시스템과 달리, 이벤트 신호에 의해 동작이 필요한 모듈만이 자기동기식으로 구동되므로 전력 소모량이 감소되고 클럭 신호의 지연과 클럭 스퀴가 발생되지 않는다. 비동기식 회로는 클럭을 사용하지 않는 대신 각 모듈의 동작이 시작됨을 지시하고 완료된 경우 동작이 종료되었음을 인접한 모듈에 지시해야 한다^{[2][3]}.

번들 데이터(bundled-data) 지연 모델의 경우 이미 데이터를 전송할 때 모듈내의 연산에 필요한 지연 시간을 알고 있으므로 단선 인코딩(single-rail encoding)을 사

* 정회원, 충북대학교 정보통신공학과 컴퓨터정보통신 연구소

(Dept. of Computer and Communications Engineering and Research Institute for Computer and Information Communication, Chungbuk Nat'l University)

※ 본 연구는 반도체설계교육센터(IDEC)의 설계 지원과 정통부가 지원하는 IT-SoC 사업단의 연구비 지원으로 수행되었음.

접수일자: 2003년10월23일, 수정완료일: 2004년8월31일

용하여 데이터를 전송하고, 고정된 지연 시간 후 완료 신호를 전송한다. 그러나 DI(delay insensitive) 지연 모델을 적용한 시스템의 경우 데이터의 지연 시간을 알지 못하므로, 완료 신호 검출을 위해 2선식 인코딩 기법으로 데이터를 전송하고 모듈의 마지막 출력단에서 동작을 검출하여 완료 신호를 생성한다.

현재 비동기 회로 인코딩에는 단선식, 2선식(dual-rail)과 더불어 1-of-4 인코딩 방식, 그 외에 1-of-M, N-of-M 인코딩 방식에 대한 연구가 이루어지고 있다⁴⁾⁵⁾⁶⁾. DI를 지향하는 모든 1-of-M 인코딩 방법에서 신호는 모두 RZ 방식을 사용하며, 이는 신호가 강제로 zero로 돌아가기 때문에 CMOS회로에서는 0→1→0 스위칭이 증가하여 전력 소모가 발생하게 된다. 본 논문에서는 RZ 방식과 같은 DI 지연 모델을 구현하면서 RZ, NRZ 방식의 혼합을 통해 기존보다 신호전이를 감소시켜 전력소모를 줄이는 새로운 2선식 방식을 제안한다.

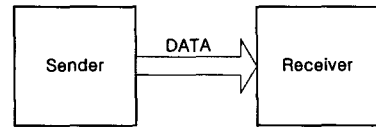
본 논문의 구성은 다음과 같다. II장에서는 일반적인 단선식, 2선식 방식과 각각의 방식에서 사용하는 RZ, NRZ방식에 대해 설명하고 III장에서는 새로운 2선식 방식에 대해 제안하고 이를 위한 파이프라인 구조와 더불어 비동기 가산기 블록을 통해 더 나은 신호 전이 감소를 위해 추가적으로 고려해 보고 각각의 신호전이에 대해 비교해 본다. IV장에서는 제안된 방식의 검증을 위해 가산기 블록에 제안한 방식을 적용한 시뮬레이션을 통해 소비전력을 비교한다. V장에서는 시뮬레이션을 통해 얻어진 결과를 통해 결론을 맺는다.

II. 데이터 전송 채널 방식

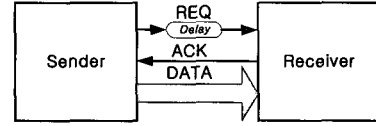
1. 단선식 채널과 다선식 채널

그림 1(a)는 비동기 회로의 bundle 데이터 전송을 위한 채널모델의 개념을 나타낸다. 비동기 회로는 데이터 전송을 위해 보내는 송신단 (Sender)블록과 데이터를 받는 수신단 (receiver)블록을 가지고 있다.

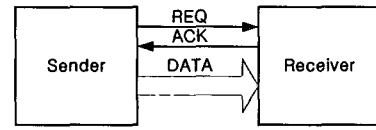
데이터는 송신단에서 수신단으로 요구신호와 함께 데이터를 보내고, 수신단은 송신단에서 온 요구신호에 의해 데이터를 받아들이고, 확인신호(Acknowledge)신호를 수신단으로 보내 데이터의 전송여부를 알려주어 데이터 전송 프로토콜을 완료한다. 그림1(b)는 번들 데이터 (Bundled data) 전송 방식으로 단선식이라고 하며, 1비트 데이터를 한개의 선으로 표현하여 데이터를 전송하고, 요구신호는 데이터를 전송하는데 필요한 지연이 고려되어 송신된다. 그림 1(c)는 1비트의 데이터를 2개의



(a) 데이터 전송 기본 모델



(b) 단선식 데이터 전송



(c) 2선식 데이터 전송

그림 1. 데이터 전송 방식

Fig. 1. The data transmission method.

선으로 표현하여 하나의 선은 DATA를 다른 하나의 선은 \overline{DATA} 를 나타내도록 하며, 그 외의 데이터에 대해서는 유효하지 않은 데이터로 판단하는 방식으로, 데이터를 통해 값이 유효한지, 아닌지를 알 수 있다. 따라서 송신단에서 보낸 데이터를 수신단이 받아 데이터의 유효성을 판단할 수 있다. 그러나 데이터를 전송하지 않을 때는 유효하지 않은 값으로 변환시켜야 하므로, 데이터를 보낼 때와 보내지 않을 때 항상 데이터를 보내는 단선방식에 비해 더 많은 신호의 변화가 요구된다.

데이터를 전송하는데 있어서 단선식과 2선식방식은 데이터 신호선의 성질에 따라 다른 전송 방식을 갖는다. 단선 방식은 데이터의 요청(Request)신호에 전송 지연에 대한 고려가 되어 있으므로, 연속적인 데이터를 전송하는 NRZ(Non-return-to-Zero)방식으로 데이터를 전송한다. 2선식 방식은 데이터의 값을 통해 유효한 값인지 판별할 수 있으므로, 데이터의 유효성을 검출하는 완료 검출기 (Completion detector) 블록을 출력 데이터에 사용하여 검출한 신호를 요청신호로 대신한다. 그러므로 요청신호가 있을 때는 입력 데이터를 유효한 값으로, 그렇지 않을 때는 입력 데이터를 유효하지 않은 값이 되게 하는 RZ(Return-to-Zero)방식으로 블록내의 지연에 따른 올바른 동작을 수행하게 된다. 따라서 데이터 전송 방식에 따라 데이터를 표현하는 방법이 다르다. 단선식 방식과 2선식 방식에서 데이터를 표현하는 방법은 표1에 나타내었다.

단선 방식에서 기능 블록의 데이터 지연은 미리 추정된 고정지연 블록에 의해 주어지므로 입력되는 비트값을

표 1. 단선식과 2선식 데이터 인코딩
Table 1. The single-rail and dual-rail data encoding.

데이터 비트	코드	데이터 비트	코드
0	0	00	Invalid
		01	0
		10	1
1	1	11	Not use

(a) 단선(single-rail) 방식 (b) 2선식(dual-rail) 방식

그대로 받아들인다. 단선 방식의 데이터 표현 방식은 표 1(a)에 나타내었다. 그러나 2선식 방식에서는 데이터에 의해 판단되므로, 유효한 데이터 값 중 1은 "01"으로, 0은 "10"으로 표현하고 유효하지 않은 상태는 "00"으로 표현한다. 그리고 "11"의 경우는 사용하지 않는다. 2선식 방식에서의 데이터 표현 방식은 표1(b)에 나타내었다.

2. NRZ vs. RZ

단선식 구조에서 사용하는 NRZ방식의 경우 기능 블록의 지연은 추정에 의하지만, 2선식 구조에서는 RZ방식의 데이터를 사용하여 한다. 그림 2는 NRZ 방식을 사용하여 데이터를 보내는 단선식에서 이전단의 핸드셰이킹 시점에서 다음단의 핸드셰이킹까지의 신호천이를 나타낸다.

그림 2에서 상태변화는 이전 단의 데이터의 저장시점에서, 핸드셰이킹(Hs1) 후 고정된 지연블럭에 의해 지연된 요청신호를 통하여 데이터의 유효한 상태로 간주하여, 다음단 파이프라인에서 저장할 수 있도록 핸드셰이킹(Hs2) 함으로 이루어진다. 따라서 고정된 지연 블록은 식(1)과 같이 기능블럭에 의해 입력신호의 출력 때까지의 지연시간(t_{Func_delay})보다 파이프라인의 데이터 저장 후부터 다음단 요청신호까지의 핸드셰이킹 지연시간(t_{Hs1})과 요청신호의 지연을 위한 고정된 지연블럭에 의한 지연시간(t_{Fix_delay})의 합보다 더 작아야 한다.

$$t_{Hs1} + t_{Fix_delay} > t_{Func_delay} \quad (1)$$

NRZ에 의한 현재단의 데이터 저장시부터 다음단의 데이터 저장까지의 시간은 식(2)와 같이 나타낼 수 있다.

$$T_{NRZ} = t_{Hs1} + t_{Fix_delay} + t_{Hs2} \quad (2)$$

그림 3은 RZ방식 Dual-Rail 신호천이를 나타낸다. 이 방식 역시 요청신호가 있을 때에만 유효한 입력 데이터로 하고, 요청신호가 없을 때에는 유효하지 않은 값(Zero)의 상태로 만들어 출력 데이터가 유효한 값을 가질때 이를 검출하여 요청신호를 보내는 방식이다. 따라

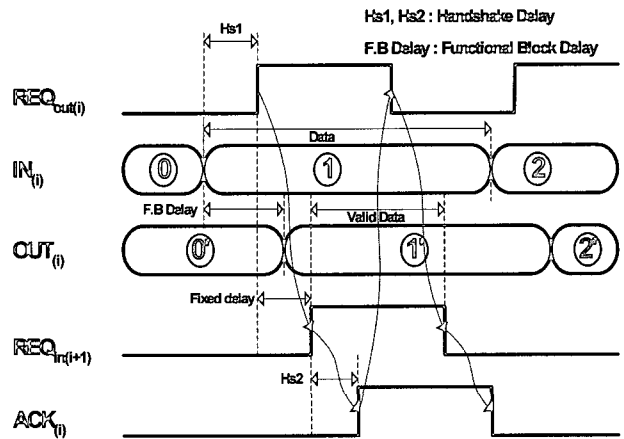


그림 2. NRZ방식 Single-Rail 신호천이
Fig. 2. The Single-Rail signal transition of NRZ method.

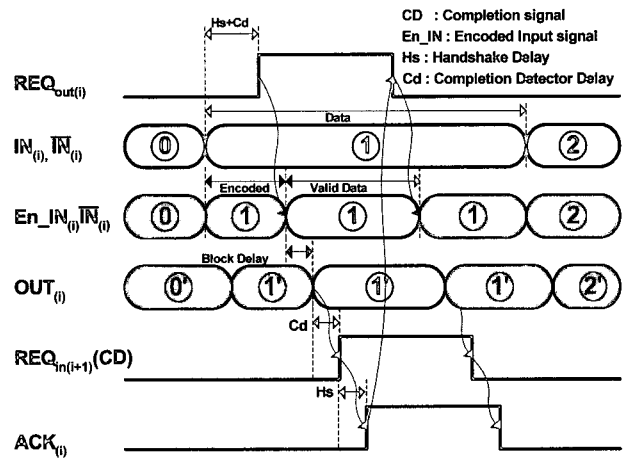


그림 3. Dual-Rail 방식의 신호천이
Fig. 3. The signal transition of Dual-Rail method.

서 데이터를 표현하기 위해서는 신호선 당 2번의 신호천이가 요구되며, 출력 데이터는 기능 블럭에 의해 지연을 갖은 후 유효한 값이 되기 때문에 올바른 동작을 위한 요청 신호를 검출해 낼 수 있다.

RZ 방식을 통한 상태 변화는 이전 단의 데이터의 저장 시점에서 기능블럭에 입력되는 데이터는 유효하지 않은 상태(Invalid)를 갖으며, 핸드셰이킹 시간(Hs1)과 완료검출(Cd)에 의해 요청신호(REQout)가 발생하고 나서부터 저장된 데이터는 유효값을 갖게 된다.

RZ 방식에 의한 현재단의 데이터의 저장부터 다음단의 데이터 저장까지의 시간은 식(3)과 같이 나타낼 수 있다.

$$T_{RZ} = t_{Hs1} + 2 t_{Cd} + t_{Encoding} + t_{Block_delay} + t_{Hs2} \quad (3)$$

따라서 RZ 방식은 NRZ 방식에 비해 블럭 지연과 핸드셰이킹 지연 외에도 몇 가지의 지연이 더 추가되지

만, 블록내의 지연을 그대로 반영할 수 있기 때문에 블록지연에 민감한 회로를 구성할 수 있다.

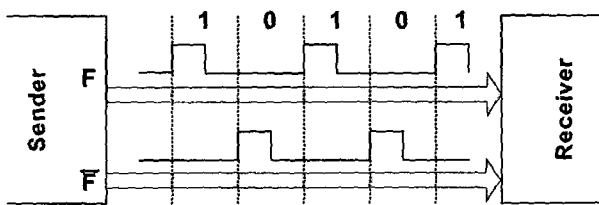
III. RZ/NRZ 전송 방식과 파이프라인 구조

1. 제안된 RZ/NRZ 전송구조

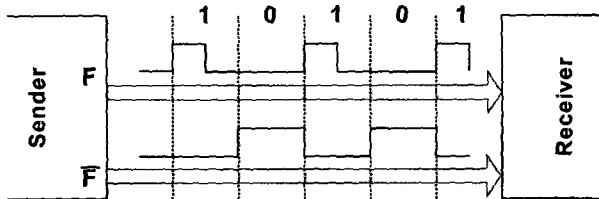
DI모델을 사용하는 2선식구조는 요청신호가 있을 때, 하나는 유효한 데이터 값을 나타내고, 다른 하나는 그 값의 반대의 값을 나타낸다. 연산의 지연을 판단하기 위해서는 데이터가 유효한 때와 유효하지 않은 때로 분리되어야 지연에 의한 데이터의 유효한 때를 판단할 수 있다. 따라서 2선식 구조에는 RZ가 사용되어지나, 이 방식은 NRZ방식에 비해 2배의 신호천이를 요구하게 된다.

신호의 유효상태만을 판단할 때 2선식 구조에서 한쪽의 데이터 선에 대해서는 RZ방식을 사용하고, 다른쪽의 데이터 선에 대해서는 NRZ 방식을 사용해도 유효상태의 판단은 가능하다. 이것은 2선의 값이 반전된 유효상태에서 한쪽의 값이 모두 0값이 되었을 때, 한가지 경우를 제외하고 모두 올바른 반전값이 아니므로 유효상태로 판단되지 않기 때문에 유효상태와 유효하지 않은 상태의 구분이 가능하다. 따라서 기존의 RZ방식을 사용하는것과 마찬가지로 데이터 전송을 올바르게 수행할 수 있다. 그리고 RZ 방식을 사용할 때 발생하는 신호천이의 수를 효율적으로 줄일 수 있으며, 소비전력도 감소시킬 수 있다. 그러나 2선식 신호가 상보적으로 스위칭되지 않으므로 EMI 특성은 저하될 수 있다.

그림 4는 2선식 RZ방식과 제안된 RZ/NRZ 혼합방식간의 데이터 전송 방식에 따른 데이터의 표현에 대해



a) 일반적인 2선식 RZ 방식



b) 제안된 2선식 RZ/NRZ 혼합 방식

그림 4. 전송 방식내의 데이터

Fig. 4. The data in transmission method.

나타낸다. 그림 4(a)의 RZ방식은 F와 F를 통한 데이터의 전송 후 0으로 복귀함으로써 하나의 데이터에 대한 주기를 갖는다. 그림 4(b)의 제안된 RZ/NRZ 혼합방식은 한쪽은 RZ방식으로 다른 한쪽은 NRZ방식을 사용함으로써 좀더 효율적인 신호천이를 갖는 인코딩 스킴을 얻는데에 있다.

그림 5은 RZ와 NRZ를 동시에 사용하는 제안된 데이터 전송 방식의 신호 천이를 나타낸다. 여기서 REQout(i)는 이전단에서 발생한 요청신호를 나타내며, IN(i), IN(i)는 이전단에 저장된 유효한 데이터를 나타낸다. 또한 En_IN(i)와 En_IN(i)는 기능블록의 입력인 인코딩된 데이터이며 OUT_IN(i)와 OUT_IN(i)는 기능블록의 출력으로 나오는데이터이다. 그리고 REQin(i+1)는 기능블록의 출력으로 검출한 요청신호를 나타낸다. 파이프라인에 데이터가 저장되면 기능 블록의 입력 데이터인 IN(i)는 유효한 값이 그대로 NRZ 인코딩 방식으로 구현되며, IN(i)는 요청신호에 의해 유효값이 되는 RZ 인코딩 방식으로 구현된다. 따라서 완료 검출과 핸드셰이킹을 거쳐야 요청신호(REQout)는 '1'이 되며, 입력 데이터를 유효한 값을 갖도록 인코딩 된다. 요청신호가 있는 구간 동안은 파이프라인이 갖는 유효한 입력 데이터가 전송되고, 요청신호가 없는 구간 내에서는 한쪽에만 유효하지 않은 값으로 인코딩 된다. 예외로 IN(i)쪽의 모든 비트의 값이 '1'일때는 유효상태의 판단이 쉽다. 즉 유효하지 않은 상태일 때 IN(i)의 모든 비트가 '0'으로 되어야 하지만 NRZ방식에 의해 변하지 않으므로 완료 검출 블록으로써는 판단할 수 없다. 따라서 이때는 IN(i)도 IN(i)와 마찬가지로 RZ방식을 적용한다.

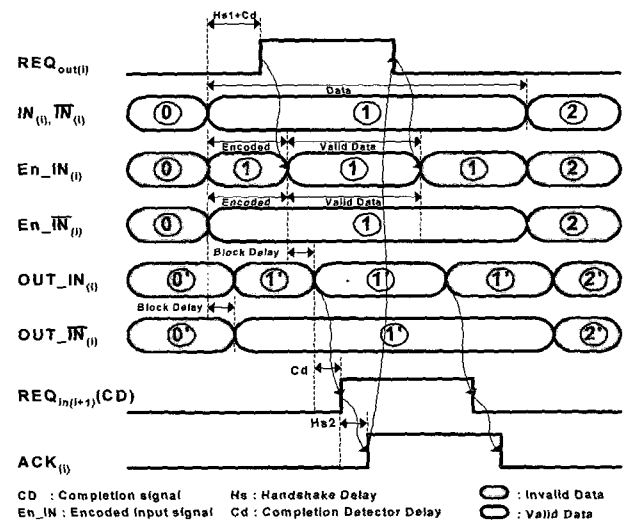


그림 5. 제안된 혼합 데이터 신호 방식

Fig. 5. The signal method of proposed mixture data.

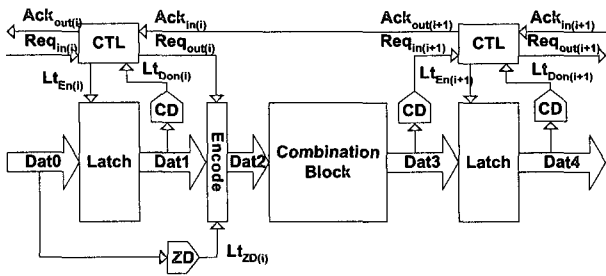


그림 6. 제안된 방식을 위한 스테이지 구조
Fig. 6. The stage structure for proposed method.

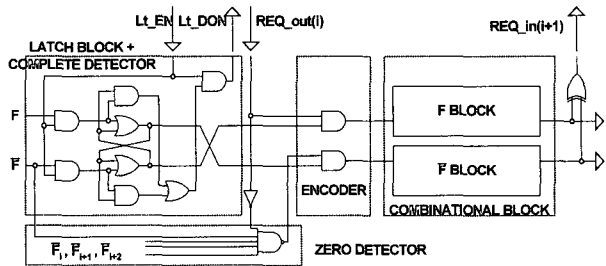


그림 7. 게이트 레벨의 제안된 데이터 전송 구조
Fig. 7. The proposed data transmission structure of a gate level.

2. RZ/NRZ 방식 파이프라인 구조

그림 6는 제안된 NRZ/RZ 혼합 데이터 전송방식의 마이크로 파이프라인 구조로, 기능 블록 사이에 2개의 완료검출 블록(CD)과 예외값 검출 블록(ZD)이 있다. 첫 단의 완료검출(CD) 블록은 래치에 저장을 확인하기 위한 블록으로 핸드셰이킹의 완료 신호(LtDon(i))를 발생하여 핸드셰이킹 컨트롤러에서 요청신호(Reqout(i))를 발생하게 해준다. 발생된 요청신호는 인코더 블록을 통해 RZ/NRZ를 위해 데이터를 변화시킨다. 변화된 데이터는 기능 블록에서 연산이 수행된 후 완료 검출 블록에 의해 결과값의 유효상태가 검출된다.

그림 7은 게이트 레벨로 표현된 1비트 데이터 회로이다. 핸드셰이킹 제어 블록에 이전 스테이지로부터 요청신호를 받으면 래치에 인에이블(Lt_En)신호가 발생되어 이전 단의 데이터는 저장되어진다. 데이터의 저장이 완료 되면 완료 신호(Lt_Don)가 발생하고, 다음 스테이지를 위한 요청신호를 발생하게 된다. 2선식 데이터의 유효판단은 보수관계의 유무에 의해서 이루어진다. 하지만 제안된 RZ/NRZ 구조에서 RZ방식이 적용된 데이터에 '1'이 포함되어 있지 않으면 유효판단이 이루어지지 않는다. 따라서 예외값인 유효값 '0'의 F의 입력에 해당하는 모든 비트의 '1'을 검출하여, F에 대해 유효값 '0'일 때는 RZ방식을 그 외의 값에 대해서는 NRZ방식을 수행하여 기능 블록에 대한 올바른 유효 데이터 판

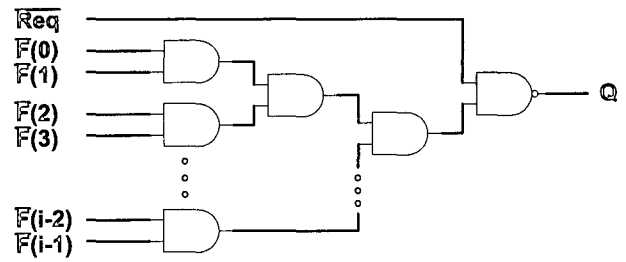


그림 8. 유효 '0'값 판단을 위한 검출 구조
Fig. 8. The detection structure for a valid value.

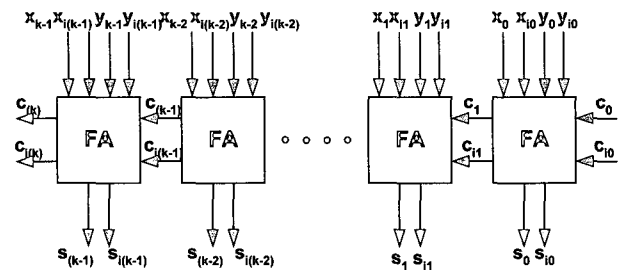


그림 9. 비동기 Ripple Carry Adder
Fig. 9. Asynchronous Ripple Carry Adder.

단이 가능하게 한다.

그림 8은 유효값 '0'에 대해 검출하기 위한 블록 구조이다. 모든 비트의 반전 신호 F에 대해서 각각 AND조합에 의해 검출된 신호와 핸드셰이킹 제어 블록에서 받은 요청신호를 통하여 유효값 '0'일 때 출력 Q는 '0'으로, 그 외의 값에 대해서는 '1'을 발생시킨다.

3. 비동기 가산기 블록

그림 9는 각 비트간 출력에 대한 경로가 다른 ripple carry adder(RCA)이다.

RCA의 임계경로는 첫단에서 발생된 캐리가 마지막 단까지 전달될 때이다. 이에 비트별 NRZ와 RZ를 적용시에, 첫 번째 비트와 캐리비트에는 RZ 인코딩 방식을, 그 외의 다른 입력 비트에 대해서는 NRZ 인코딩 방식을 적용할 수 있으며, 출력값의 완료검출로 유효 상태를 올바르게 판단할 수 있다. 따라서 임계경로에 따른 비트별 고려에 따른, NRZ방식의 적용을 통해 더 나은 스위칭 감소 결과를 얻을 수 있다.

4. 인코딩 방식의 신호천이 비교

다음의 표2는 인코딩 방식에 따라 RZ와 NRZ방식과 제안된 RZ/NRZ 혼합방식을 적용한 특성 비교를 나타낸다. 2선식 인코딩 방식에서는 F, F를 각각 RZ, NRZ방식을 서로 사용한 것과 앞서 RCA에서와 같이 임계경로를 고려하여 해당 비트는 RZ로 그 외의 비트는

표 2. 각 인코딩 방식의 비교

Table 2. The comparison of an encoding method.

	Area wire / bit	Energy Transition / bit
Single-Rail	1	0.5 (평균)
Dual-Rail (RZ)	2	2
Dual-Rail (NRZ)	2	1
Dual-Rail (RZ/NRZ)	2	1.5 (평균)
Dual-Rail (RZ/NRZ) (critical path 고려)	2	$1 + \frac{1}{M}$
1-of-4 (RZ)	2	1
1-of-M (RZ)	$\log_2 M$	$\frac{2}{\log_2 M}$
1-of-M (RZ/NRZ)	$\log_2 M$	$\frac{1.5}{\log_2 M}$

* M : 단위 인코딩 비트수

NRZ방식을 사용하는 방법 등을 기존의 RZ, NRZ방식과 비교하였다.

단선방식에서 사용되는 NRZ방식은 데이터 전송 구간동안 1개의 선을 통해 1번의 천이로 데이터를 표현할 수 있으며, 2선식에서 사용되는 RZ방식은 2개의 선을 통해 라인당 2번의 신호천이까지 일어날 수 있으며, 제안된 방식은 한쪽은 RZ방식을 사용하여 2번의 천이 다른 한쪽은 NRZ방식을 사용하여 한번의 천이가 있으므로, 평균 1.5번의 신호 천이를 갖는다. 또한 임계 경로를 고려한다면, RZ의 적용비트를 최소화하여 신호천이를 더 감소시킬 수 있다.

그리고 1-of-M과 같은 M비트단위로 one-hot code를 사용하는 인코딩 방식에서도 적절히 적용할 수 있다.

IV. 시뮬레이션 결과

본 논문에서 제안된 데이터 전송 방식의 신호천이 감소에 대한 소비전력을 검증하기 위해, 가산기(RCA) 블록을 사용하여 RZ와 NRZ/RZ 혼합 데이터 방식에 따른 신호천이의 수를 비교하였다. 그리고 경로 지연에 따른 비트별 혼합방식의 적용에서 RZ방식의 적용은 캐리 입력인 C_0 , C_{i0} 와 각 입력의 LSB에 대해서 적용하였고, 그 외의 비트에 대해서는 NRZ방식을 사용하였다. 여기서 제안된 RZ/NRZ 혼합방식은 RZ/NRZ(1)으로, 임계경로를 고려한 비트별 RZ/NRZ 혼합 방식은 RZ/NRZ(2)으

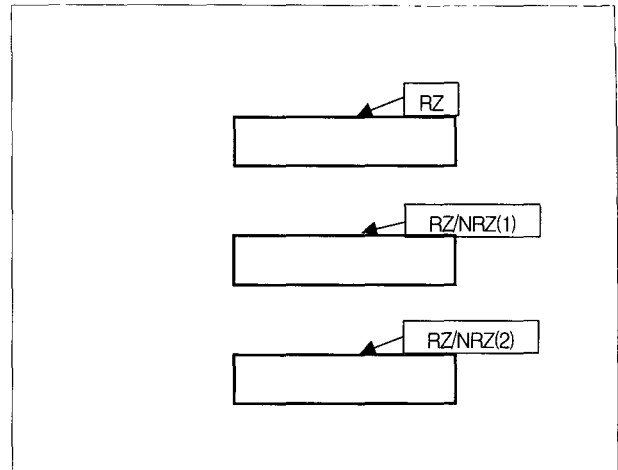


그림 10. RZ방식과 NRZ 혼합방식을 적용한 ADDER

Fig. 10. The adder to apply RZ method and NRZ mixed method.

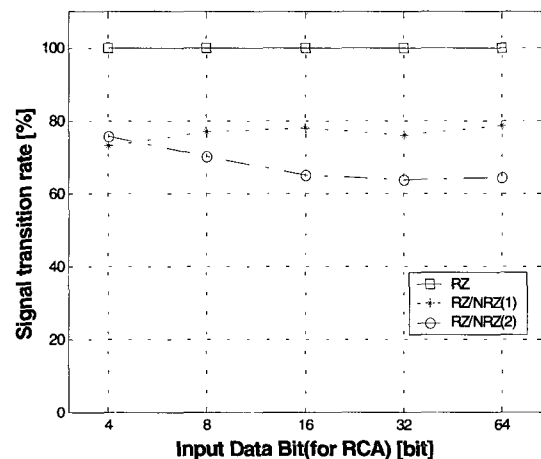


그림 11. RCA의 데이터 전송 방식에 따른 기능블록내의 신호천이 비교

Fig. 11. The signal transition comparison according to the data transmission of RCA.

로 하였다.

신호천이 측정을 위해 VHDL로 각 방식에 대한 회로를 설계하여 게이트 레벨로 합성하였고, Hynix 0.35um ASIC공정의 spice 모델을 적용하여 컴파일한 후 nanosim을 통해 시뮬레이션 결과를 확인하였다. 입력 데이터가 16비트일 때의 가산기 신호천이 비교를 위한 시뮬레이션 파형은 그림 10과 같다. 입력 데이터는 랜덤 데이터를 동일하게 기존의 RZ방식과 제안된 2가지의 인코딩 스킴에 적용하였다. 각 인코딩 방식에서 요청신호(req1_out)에 따라 출력 데이터의 유효 판단이 완료신호(req2_in)를 통해 올바르게 수행됨을 볼 수 있다. 따라서 제안된 인코딩 방식도 RZ방식과 같이 고정 지연의 요구 없이 블록 지연에 민감한 동작이 가능하

표 3. 데이터 인코딩 방식별 회로 면적 및 신호전이 비교

Table 3. The circuit area and signal transition comparison for data encoding.

비트	인코딩 방식	회로 면적 (#Tr)				신호전이 수			
		#기능블록	#예외값 검출 블록	#인코더 블록	#완료 검출 블록	#기능 블록	#예외값 검출 블록	#인코더 블록	#완료 검출 블록
16	RZ	1160	-	396	228	58789	-	32370	21920
	RZ/NRZ(1)	1160	98	398	228	45768	3036	18358	27850
	RZ/NRZ(2)	1160	-	36	28	38378	-	2632	20681

다. 하지만 제안된 방식은 추가의 회로를 요구하므로 회로 면적이 증가된다. 표3은 16비트 RCA에서의 각 방식에 따른 회로 면적과 신호전이의 비교를 나타낸다. 기존 RZ방식에 비해 제안된 방식은 예외값 검출 블록의 추가로 인해 스테이지 면적이 약 2%정도 증가되었다. 그러나 신호전이 수에 있어서는 예외값 검출 블록에 의해 추가된 신호전이 수보다 변형된 인코더 블록과 기능 블록에 의해 감소된 신호전이의 수가 더 많아 전체적으로는 감소된 신호전이를 나타내었다.

그림 11은 가산기(RCA) 블록을 사용하여 각 전송 방식에 따른 신호전이 수의 비교를 나타낸다. 4, 8, 16, 32 비트의 입력 데이터를 갖는 가산기 블록을 사용하여 각각의 경우에 대한 블록내의 신호 전이 수를 전송 방식별로 나타내었다. 제안된 방식은 입력 값에 의해 감소되는 신호전이의 수가 변하므로, 랜덤한 데이터 200개를 통해 시뮬레이션 하였다. 2선식에 RZ를 모두 적용하였을 때 보다 RZ와 NRZ를 혼합하여 적용한 RZ/NRZ(1)은 평균 23%정도의 신호전이 수가 감소되었으며, 임계 경로를 고려한 비트별 적용한 RZ/NRZ(2)은 평균 32%정도의 신호전이 수가 감소되었다. 이 신호전이 수의 감소는 입력 데이터 조건에 따라 최대 50%까지 NRZ방식의 신호전이에 대해 근접하게 감소시킬 수 있다.

서 제안된 RZ/NRZ 데이터 전송 방식은 비동기 회로의 이점 중 하나인 EMI면에서는 특성이 기존보다 낮아질 수 있지만, 전력 면에서는 저 전력을 구현할 수 있는 데이터 전송 방식임이 검증되었다.

참 고 문 헌

- [1] S. Huack, "Asynchronous design methodologies: an overview," *Proc. the IEEE*, vol. 83, no. 1, pp. 69-93, Jan 1995.
- [2] V. Akella, N. H. Vaidya, G. R. Redinbo, "Limitations of VLSI implementations of delay-insensitive codes," *IEEE Proc.26th Int. Symp. on Fault-Tolerant Computing*, pp. 208-217, Jun 1996,
- [3] T. Verhoeff, "Delay-insensitive code an overview," *Distributed Computing*, vol. 3, pp. 1-8, 1988
- [4] W. J. Bainbridge, and S. B. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding" *Async' 2001*, pp. 118-126, April 2001.
- [5] D. W. Lloyd, and J. D. Garside, "A practical comparison of asynchronous design styles", *Async' 2001*, pp. 36-45, April 2001.
- [6] W.J. Bainbridge, et. al. "Delay-Insensitive, Point-to-Point Interconnect using m-of-n Codes"; *ASYNC' 2003*, Vancouver, Canada, pp. 132-140, May 2003,

V. 결 론

본 논문은 2선식 데이터 전송 방식에서 회로 내에서 RZ/NRZ를 혼용하여 신호전이 수를 최대 50%까지 감소시키고 소비전력을 줄이는 새로운 데이터 전송 방식을 제안한다. 제안된 알고리즘의 비동기식 파이프라인 구조와 예외 값 처리를 위한 블록을 설계하였다. 그리고 기존 RZ방식의 2선식 데이터 인코딩 방식과 제안된 2선식 인코딩 방식과 비교하기 위해 가산기 블록을 사용하여 입력 데이터의 비트크기에 따른 전력 소비를 비교한 결과 평균 23%의 감소를 확인할 수 있었다. 따라

저 자 소 개



이 원 철(학생회원)
 2002년 2월 충북대학교
 정보통신공학과 학사.
 2004년 2월 충북대학교 정보통신
 공학과 석사.
 <주관심분야: 메모리 소자 설계,
 비동기 회로 설계, 고속 마이크로
 프로세서 설계>



이 제 훈(정회원)
 1998년 2월 충북대학교
 정보통신공학과 학사.
 2001년 2월 충북대학교
 정보통신공학과 석사.
 2003년 2월 충북대학교 정보통신
 공학과 박사과정수료.
 <주관심분야: 홈 네트워킹, 통신용 ASIC 설계,
 고속 마이크로프로세서 설계>



조 경 록(정회원)
 1977년 경북대학교 전자공학과
 공학사.
 1989년 일본 동경대학교
 전자공학과 공학석사.
 1992년 일본 동경대학교
 전자공학과 공학박사.
 1979년~1986년 (주)금성사 TV연구소 선임연구원.
 1992년~현재 충북대학교 정보통신공학과 부교수.
 <주관심분야: VLSI 시스템설계, 통신 시스템용
 LSI 개발, 고속 마이크로프로세서 설계.>