

# 도선의 전송지연을 최소화하기 위한 데이터 초기화에 기반한 단일선 4-위상 프로토콜

<sup>0</sup>정은구<sup>1</sup> 이동익<sup>1</sup> 노명찬<sup>2</sup>  
<sup>1</sup>광주과학기술원 정보통신공학과  
<sup>2</sup>한국전자통신연구원 전자상거래연구부 SCM연구팀  
 (egjung, dilee)@kjist.ac.kr, mcroh@etri.re.kr

## Four-Phase Single-Rail Protocol with Return-to-Zero Data to Reduce Delay in Long Wire

<sup>0</sup>Eun-Gu Jung<sup>1</sup> Dong-Ik Lee<sup>1</sup> and Myung-Chan Roh<sup>2</sup>

<sup>1</sup>Dept. of Information and Communications, Kwangju Institute of Science and Technology (K-JIST)

<sup>2</sup>SCM Team, Dept. of EC, ETRI

### 요 약

VLSI칩 내부에서 길이가 긴 도선을 통해서 데이터를 보낼 때, 누화(crosstalk)의 영향으로 데이터 값에 따라 지연시간이 변한다. 그리고 지연시간의 변화폭도 CMOS공정이 초미세화됨에 따라서 더욱 커진다. 본 논문에서는 지연시간을 줄이기 위해서, 데이터 코딩을 이용하여 가장 긴 지연시간을 피하는 데이터 초기화에 기반한 단일선 4-위상 프로토콜을 제안한다. 제안된 프로토콜을 긴 도선에 적용함으로써 도선에서의 최대 지연시간을 최대 49% 감소시킨다.

### 1. 개요

CMOS공정이 초미세화됨에 따라서 VLSI칩 내부에서 도선의 면적에 기인한 커패시턴스에 대한 인접한 도선과의 결합 커패시턴스의 비율이 증가한다. 이 비율이 증가함에 따라, 도선에서의 누화(crosstalk)의 영향도 커진다. 따라서, 도선을 통해서 데이터가 전송될 때, 도선에서 데이터 값의 변화에 의한 지연시간의 변화폭도 증가한다. 특히, 버스와 같은 긴 도선을 사용하는 곳에서는 이 값의 변화가 시스템의 성능에 영향을 준다. 이 지연시간을 줄이는 방법은 버퍼를 삽입하거나[1], 데이터 코딩을 사용해서 가장 긴 지연시간을 가지는 데이터 값의 변화를 피하는 것이다 [2,3]. 동기식 시스템에서 코딩 방법을 이용할 때, 추가적인 도선을 사용하거나 두개의 클럭 사이클을 이용해서 구현할 수 있다. 반면에 단일선 4-위상 프로토콜(four-phase single-rail protocol)을 가진 비동기 시스템에서는 추가적인 도선을 사용하지 않고 작은 프로토콜 비용으로 코딩이 가능하다. 본 논문에서는 데이터가 항상 초기화상태에서 변화도록 코딩한 새로운 단일선 4-위상 프로토콜을 제안한다.

### 2. 도선에서의 지연시간

도선에서 데이터 값의 변화에 의한 지연시간의 변화는 표1과 같다 [2]. 여기서  $\mu$ 는 기술변수(technology parameter) 이고,  $\lambda$ 는 도선의 면적에 기인한 커패시턴스(line-to-ground capacitance: CLG\_G)에 대한 인접한 도선과의 결합 커패시턴스(line-to-line capacitance: CLL\_G)의 비율이다. - 는 데이터의 값의 변화가 없음을 의미하고, ↑는 데이터 값이 0에서 1로 변함을 의미하고, ↓는 데이터 값이 1에서 0으로 변함을 의미한다. 그리고 도선  $k-1$ 와  $k+1$ 는 도선  $k$ 에 인접한 도선을 의미한다. 도선  $k$ 에서의 지연시간은 도선  $k, k-1$ 와  $k+1$ 에서의 신호변화에 의해서 결정된다. 그래서, 도선을 통해서 데이터를 전송할 때 가지는 최대

표 1. 도선의 지연시간

Line $k-1$	Line $k$	Line $k+1$	Delay of line $k$
↑	↑	↑	$\mu$
↓	↓	↓	$\mu$
-	↑	↑	$\mu(1+\lambda)$
-	↓	↓	$\mu(1+\lambda)$
↑	↑	-	$\mu(1+\lambda)$
↓	↓	-	$\mu(1+\lambda)$
-	↑	-	$\mu(1+2\lambda)$
-	↓	-	$\mu(1+2\lambda)$
↑	↑	↓	$\mu(1+2\lambda)$
↑	↓	↓	$\mu(1+2\lambda)$
↓	↑	↑	$\mu(1+2\lambda)$
↓	↓	↑	$\mu(1+2\lambda)$
-	↑	↓	$\mu(1+3\lambda)$
-	↓	↑	$\mu(1+3\lambda)$
↑	↓	-	$\mu(1+3\lambda)$
↓	↑	-	$\mu(1+3\lambda)$
↓	↑	↓	$\mu(1+4\lambda)$
↑	↓	↑	$\mu(1+4\lambda)$

지연시간은  $\mu(1+4\lambda)$ 이며, 이와 같은 최대지연시간은 코딩방법을 이용하여 데이터의 변화를 변경함으로써 줄일 수 있다.

### 3. 데이터 초기화에 기반한 단일선 4-위상 프로토콜

#### 3.1 프로토콜

기존의 단일선 4-위상 프로토콜은 그림1과 같다 [4]. 송신기(sender)에서 유효한 데이터가 존재하면, 데이터와 함께 request신호를 수신기(receiver)로 보낸다. 그러면, 수신기는 데이터를 받고, acknowledge신호를 송신기에 보낸다.



그림 1. 단일선 4-위상 프로토콜

따라서, 데이터는 그림1과 같이 request신호가 상승하고 acknowledge신호가 상승할 때까지만 유효하면 되고, 새로운 request신호가 상승하기 전에 유효한 데이터가 존재하면 된다. 그러므로 acknowledge신호가 상승하고 request신호가 다시 상승하는 구간사이에서는 의미 없는 데이터가 존재하기 때문에, 이 부분을 다른 값을 가지도록 변경해서 도선에서의 지연시간을 감소시킬 수 있다. 데이터 값들의 변화가 램덤하다고 가정하고, 코딩으로 인한 성능의 감소를 최소화 시키면서 도선에서의 지연시간을 최대한으로 감소시키는 방법은 데이터를 초기화하는 것이다. 즉, 의미 없는 구간에 데이터를 0으로 변경해서, 그림2와 같이 데이터가 항상 0에서 변할 수 있도록 해준다.

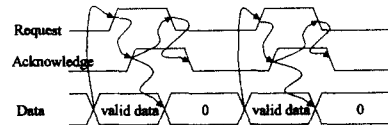


그림 2. 데이터 초기화에 기반한 단일선 4-위상 프로토콜

그러면, 도선에서 데이터를 전송할 때 지연시간은 표1에서 보는 것처럼  $\mu(1+2\lambda)$ 가 된다. 데이터 값을 0으로 바꾸는데 필요한 코딩시간을 무시할 경우, 도선상의 지연시간이  $\mu(1+2\lambda) / \mu(1+4\lambda)=0.52$  배로 감소한다.  $0.25\mu\text{m}$  CMOS공정에서  $\lambda=5.8$ 이다.

3.2 구현

단일선 4-위상 프로토콜을 구현한 것 중에서 완전히 분리된 (fully decoupled) 형태의 회로는 그림 3과 같다 [4]. 이것은 이전 단계에서 데이터를 받기 위한 수신기(receiver)기능과 다음 단계에 저장된 데이터를 보내기 위한 송신기(sender)기능을 가지고 있다. 즉, Rin(request)신호가 들어오면, Lt신호를 발생시켜서 데이터를 D플립플롭에 저장하고, Ain(acknowledge)신호를 발생시킨다. 그리고 저장된 데이터를 다음 단계 전달하기 위해서 Rout(request)신호를 발생시킨다. 여기서 사용된 D 플립플롭은 그림4와 같다 [5].

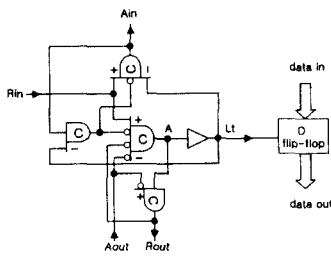


그림 3. 완전히 분리된 형태의 컨트롤 회로

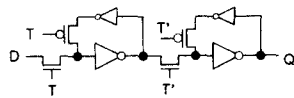


그림 4. D 플립플롭

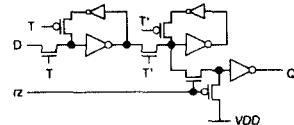


그림 5. 데이터 초기화를 가진 D 플립플롭

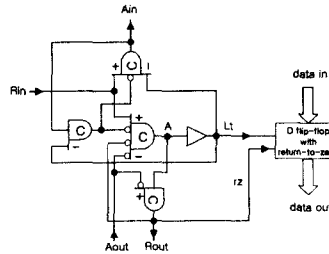


그림 6. 데이터 초기화에 기반한 단일선 4-위상 프로토콜 컨트롤 회로

본 논문에서 제안된 데이터 초기화에 기반한 단일선 4-위상 프로토콜을 구현하기 위해서, D 플립플롭의 구조를 그림5와 같이 변경했다. 그림5 같이 rz입력 신호에 의해서 Q값은 D 플립플롭에 저장된 값이 나가거나, 혹은 0이 나간다. 그리고 그림2에서와 같이 유효한 데이터가 있으면 request신호를 발생시키므로, rz입력 신호를 송신기의 request신호에 의해서 제어되도록 한다. 즉, Rout신호를 사용해서 D 플립플롭의 출력을 제어한다. 제안된 프로토콜의 회로도도 그림 6과 같다.

3.3 성능 측정

제안된 프로토콜의 성능을 알아보기 위해서, 그림 7와 같은 형태로 구성했다. 송신기(sender)는 데이터 초기화 기능이 있는 프로토콜과 그런 기능이 없는 프로토콜을 각각 사용해서 실험하였다. 이때, 수신기(receiver)는 항상 데이터 초기화 기능이 없는 프로토콜을 사용했다.

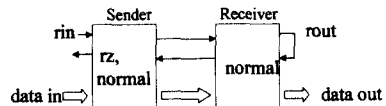


그림 7. 성능 측정을 위한 구조

여기서 normal은 그림 3의 회로를 의미하고, rz는 그림 6의 회로를 의미한다. 송신기와 수신기 사이의 거리는 0이고, 32bit 데이터 버스로 가정된 경우에 대한  $0.25\mu\text{m}$  CMOS공정에서 HSPICE 시뮬레이션 결과는 표2와 같다.

표 2. 프로토콜의 성능

	normal(ns)	rz(ns)	diff(ns)
rin+ → rout+ (latency)	0.897	0.960	0.063
rout+ → rout+ (throughput)	1.251	1.281	0.030

여기서 diff는 normal과 rz사이의 차를 나타낸다. 데이터 초기화에 기반한 프로토콜을 사용함으로써 프로토콜 동작시간이 0.063ns 증가하였고, 이것은 D 플립플롭에서 rz입력신호에 의해서 출력을 발생시키는데 필요한 시간을 의미한다.

4. 누화로 인한 도선지연

4.1 도선의 폭과 간격

버스와 같이 길이가 긴 도선을 사용할 경우, 도선에서의 지연시간을 최소화하기 위해 도선의 저항과 커패시턴스가 작은 값을 갖도록 도선의 폭과 도선과 도선사이의 간격을 결정해야 한다. 특히, CLL\_G값이 누화에 의한 영향에 큰 역할을 하기 때문에, CLL\_G값이 가능한 작아야 한다. 0.25 $\mu$ m CMOS공정에서 도선과 도선 사이의 거리에 따라서 그림8과 같은 CLL\_G값의 변화를 볼 수 있다. 거리가 1s에서 2s로 변했을 때가 2s에서 3s로, 3s에서 4s등으로 변할 때보다 CLL\_G값의 가장 큰 폭의 감소가 발생한다. 여기서 s는 최소 간격을 의미한다.

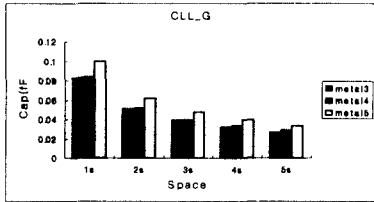


그림 8. 도선 간격에 따른 CLL\_G값

도선과 도선사이의 간격을 2s로 정했을 때, 도선의 폭의 변화에 따른 10mm도선에서의 지연시간은 그림 9와 같다. 1w에서 2w로 변했을 때가 2w에서 3w, 3w에서 4w등으로 변할 때보다 가장 큰 지연 시간의 감소를 나타낸다. 여기서 w는 도선의 최소 폭이다.

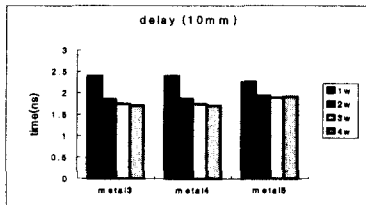


그림 9. 도선 폭에 따른 지연시간

4.2 도선에서의 지연시간 측정

그림 10와 같이 도선에서 지연시간을 측정하기 위해서 driver, wire model[2], load로 구성한다. 그리고 wire model에서 도선의 저항값과 커패시턴스값은 도선에 의해서 차지하는 면적과 지연시간을 고려해서 도선의 폭과 간격의 각각 최소의 2배로 했을 때의 값을 사용한다.



그림 10. 도선의 지연시간 측정을 위한 구조

0.25 $\mu$ m CMOS공정에서의 HSPICE시뮬레이션 결과는 표3과 같다. 여기서  $\mu(1+4\lambda)$ 는 표1에서와 같이 도선에서 나타날 수 있는 최대 지연시간을 의미이며, 또한 기존의 단일선 4-위상 프로토콜을 사용할 때 도선에서 나타날 수 있는 최대 지연시간이다.  $\mu(1+2\lambda)$ 는 제안된 프로토콜을 도선에 적용했을 경우 도선에서 나타날 수 있는 최대 지연시간을 의미한다. 그리고 diff는

표3. 도선의 지연시간

dist.	$\mu(1+2\lambda)$ (ns)	$\mu(1+4\lambda)$ (ns)	diff. (ns)
1mm	0.194	0.258	0.062
2mm	0.245	0.383	0.138
3mm	0.299	0.514	0.215
4mm	0.359	0.653	0.294
5mm	0.426	0.804	0.378
6mm	0.499	0.967	0.468
7mm	0.579	1.145	0.566
8mm	0.665	1.338	0.673
9mm	0.759	1.546	0.787
10mm	0.860	1.771	0.911

$\mu(1+4\lambda)$ 와  $\mu(1+2\lambda)$  사이의 차이를 의미한다. 도선의 길이가 1mm일 때 지연시간의 차이가 0.062ns이고, 이것은 표2와 같이 기존의 단일선 4-위상 프로토콜에 비해서 제안된 프로토콜을 사용함으로써 추가되는 시간보다 약간 작다. 따라서, 도선의 길이가 1mm보다 크면 기존의 프로토콜에 비해서 제안된 프로토콜을 이용함으로써 추가되는 시간이 상쇄된다. 그리고 도선의 길이가 10mm일 때, 제안된 프로토콜을 적용할 때의 최대 지연시간이 기존의 프로토콜을 적용할 때의 최대 지연시간의 0.49배로 감소한다.

5. 결론

긴 도선에서 누화의 영향으로 지연시간이 데이터 값에 의존하기 때문에, 데이터 코딩을 통해서 가장 긴 지연시간을 가지는 데이터 변화를 피해서 보다 작은 지연시간을 가지도록 할 수 있다. 이를 위해서, 데이터 초기화에 기반한 단일선 4-위상 프로토콜을 제안했고, 제안한 방법을 길이가 긴 도선에 적용함으로써 도선에서의 지연시간을 최대 48% 줄일 수 있었다. 제안된 프로토콜은 SoC(system-on-a-chip)를 위한 고성능 비동기 버스 시스템을 설계하는데 적용 가능하다.

참고문헌

- [1] Nekili, M. and Savaria, Y.: "Optimal methods of driving interconnections in VLSI circuits", *Proc. ISCAS*, pp.21 - 24,1992
- [2] Sotiriadis, P.P.and Chandrakasan, A.: "Reducing bus delay in submicron technology using coding", *Proc. ASP-DAC*, pp.109 -114,2001
- [3] Joon-Seo Yim and Chong-Min Kyung: "Reducing cross-coupling among interconnect wires in deep-submicron datapath design", *Proc. DAC*, pp.485 -490,1999
- [4] Furber, S.B. and Day, P.: "Four-phase micropipeline latch control circuits", *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 4, pp.247-253, June 1996
- [5] Hossain, R., Wronski, L.D. and Albicki, A.: "Low power design using double edge triggered flip-flops", *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol. 2, pp.261-265, June 1994