



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사학위논문

다중 전력 모드 설계를 위한 가변 지연
시간 버퍼 설계 및 제어 회로 분석

Analysis on Adjustable Delay Buffer Design and
Control Circuit for Multiple Power Mode Designs

2014년 2월

서울대학교 대학원

전기·정보 공학부

김 근 호

다중 전력 모드 설계를 위한 가변
지연 시간 버퍼 설계 및 제어 회로
분석

Analysis on Adjustable Delay Buffer Design
and Control Circuit for Multiple Power Mode
Designs

지도교수 김 태 환

이 논문을 공학석사 학위논문으로 제출함

2014년 2월

서울대학교 대학원

전기 정보 공학부

김 근 호

김근호의 공학석사 학위논문을 인준함

2014년 2월

위 원 장 : __ 최 기 영_____

부위원장 : __ 김 태 환_____

위 원 : __ 김 재 하_____

초록

다중 전력 모드 설계를 위한 가변 지연 시간 버퍼 설계 및 제어 회로 분석

서울대학교 대학원
전기·정보 공학부
김근호

디자인의 추세가 다중 전력 모드로 되면서, 공급 전력에 따른 클락 스큐 제한(clock skew constraint)을 만족시키는 것이 어려워졌다. 클락 스큐 제한을 만족시키기 위해서 제안된 것이 가변 지연 시간 버퍼(Adjustable Delay Buffer, ADB)이며, 가변 지연 시간 버퍼는 공급 전력의 변화에 따라 전파 지연(propagation delay)을 조절함으로써 클락 스큐 제한을 만족시켰다. 지금까지 가변 지연 시간 버퍼 관련 논문은 가변 지연 시간 버퍼를 회로에 삽입하는 최적화 알고리즘을 주로 다루었다. 이 논문에서는 가변 지연 시간 버퍼의 종류와 특징을 살펴보고, 가변 지연 시간을 조절하는 컨트롤 박스 로직의 실제 구현 방식을 알아보려 한다. 그리고 다중 전력 모드의 수에 따라 가변 지연 시간 버퍼에 연결되는 컨트롤 신호 라인(control signal line)의 전체 길이의 변화를 알아봄으로써 가변 지연 시간 버퍼의 사용이 회로에서 얼마만큼의 영역을 차지하는지 그 비중을 살펴보고자 한다.

주요어 : 가변 지연 시간 버퍼, 컨트롤 박스 로직, 전파 지연, 다중 전력 모드

학번 : 2012-20743

목차

초록	i
목차	iii
표 목차	iv
그림 목차	v
제 1 장 서론	1
제 2 장 본문	3
2.1 CADB와 IADB의 특성	3
2.2 CADB 컨트롤 박스 로직 구현	14
2.3 ADB 컨트롤 박스에 연결되는 컨트롤 신호 라인의 길이	18
제 3 장 구현	21
제 4 장 결론	22
참고문헌	23
ABSTRACT	24

표 목차

표 2.1 각 벤치마크에서의 ADB 컨트롤 신호 라인(ADB control signal line) 길이와 클럭 신호 라인(clock signal line) 길이 비교	19
--	----

그림 목차

그림 2.1	두 종류의 ADB의 구조[1]. (a) CADB(Capacitor based ADB)의 구조 (b) IADB(Inverter based ADB)의 구조.....4
그림 2.2	스위치 개수에 따른 CADB의 전파 지연 특성의 분석. (a) CADB_1: 캐퍼시터 1개로 이루어진 캐퍼시터 뱅크를 갖는 CADB의 스위치 ON 개수에 따른 전파 지연 그래프 (b) CADB_4: 동일한 크기의 캐퍼시터 4개로 이루어진 캐퍼시터 뱅크를 갖는 CADB의 스위치 ON 개수에 따른 전파 지연 그래프6
그림 2.3	스위치 개수에 따른 CADB의 전파 지연 특성의 분석. (a) CADB_4_45: W값이 45 nm인 스위치 4개로 이루어진 CADB의 전파 지연 그래프 (b) CADB_4_135: W값이 135nm 인 스위치 4개로 이루어진 CADB의 전파 지연 그래프8
그림 2.4	스위치 개수에 따른 CADB의 전파 지연 특성의 분석. (a) CADB_1: W값이 180 nm인 스위치 1개로 이루어진 CADB의 전파 지연 그래프 (b) CADB_4: W값이 45nm 인 스위치 4개로 이루어진 CADB의 전파 지연 그래프10
그림 2.5	IADB의 전파 지연. 동일한 W의 인버터로 구성된 인버터 체인을 갖고 있는 IADB의 스위치 ON 개수에 따른 전파 지연 그래프.11
그림 2.6	식 2.3에서 구한 W 비율을 적용한 IADB의 전파지연13
그림 2.7	트랜지스터의 레이아웃. (a) WL 인 트랜지스터 1개. (b) (W/2)L인 트랜지스터 2개15

그림 2.8	CADB의 캐퍼시터 뱅크가 가지는 캐퍼시터들의 개수와 캐퍼시터의 크기 최적화 알고리즘.....	16
--------	---	----

제 1 장 서론

클락 스큐 제한(clock skew constraint)를 만족시키는 것은 클락 트리 합성(clock tree synthesis)에서 중요한 일이다. 더욱이 디자인의 추세가 다중 전력 모드(multiple power mode)로 가면서, 사용되는 어플리케이션의 변화에 따라 공급 전압이 바뀌었다. 파워 모드가 여러 개가 되면서 클락 트리의 스큐 제어가 어려워졌는데, 이는 동작 중에 클락 시그널 지연 시간(clock signal delay) 또한 동적으로 바뀌기 때문이다. 최근 많은 연구에서는, 다중 전력 모드에서 클락 스큐의 변화에 따른 문제를 효과적으로 해결하기 위해, 지연 시간을 적절하게 동적으로 조절할 수 있는 가변 지연 시간 버퍼(Adjustable Delay Buffer, ADB)가 제안되었다. 그 후, CADB (Capacitor based ADB)와 IADB (Inverter based ADB)라는 두 종류의 가변 지연 시간 버퍼가 소개되었고, 파워 모드 별로 클락 스큐를 조절하기 위해서 클락 스큐의 어느 위치에 가변 지연 시간 버퍼를 적절히 넣을 것인가를 분석하는 알고리즘[1][2]이 제시되었다.

논문 [1], [2] 는 다중 전력 모드 설계에서 클락 스큐 제한을 만족시키기 위해 가변 지연 시간 버퍼를 삽입하는 문제를 다루고 있다. 논문 [1] 은 가변 지연 시간 버퍼 삽입 문제에 대해, 각 파워 모드에 대해 최적화된 선형 시간 알고리즘(linear-time algorithm)을 제안하였다. 논문 [2] 는 논문 [1] 이 각각의 파워 모드에 대한 개별적인 최적화 알고리즘인 점을 개선하여, 모든 파워 모드 전반에 대한 최적화된 가변 지연 시간 버퍼 삽입 알고리즘을 제안하였다.

그런데 가변 지연 시간 버퍼(ADB)와 관련한 기존의 논문들은 다중 전력 모드에서 클락 트리의 클락 스큐 제한을 만족시키기 위한 ADB 삽입 알고리즘에 관한 내용은 다루었지만, 실제로 ADB를 구현했을 시, 각 파워 모드에 따라 ADB 내부의 전파 지연(propagation delay)을 조절하는 ADB 컨트롤 박스의 로직(logic)을 구현하는 것에 대한 본격적인 언급은

소홀하였다.

이 논문에서는 두 타입의 가변 지연 시간 버퍼 IADB와 CADB의 특성을 알아보고, 이를 참조하여 ADB의 컨트롤 박스 로직을 구현하는 알고리즘을 소개하려 한다. 또한, 파워 모드에 따라 ADB의 전파 지연(propagation delay)을 조절하기 위해 컨트롤 박스에 연결되는, 신호 라인(signal line)의 길이를 계산한 뒤, 신호 라인(signal line)이 회로에서 차지하는 비중을 알아보려고 한다.

제 2 장 본문

2.1 CADB와 IADB의 특성

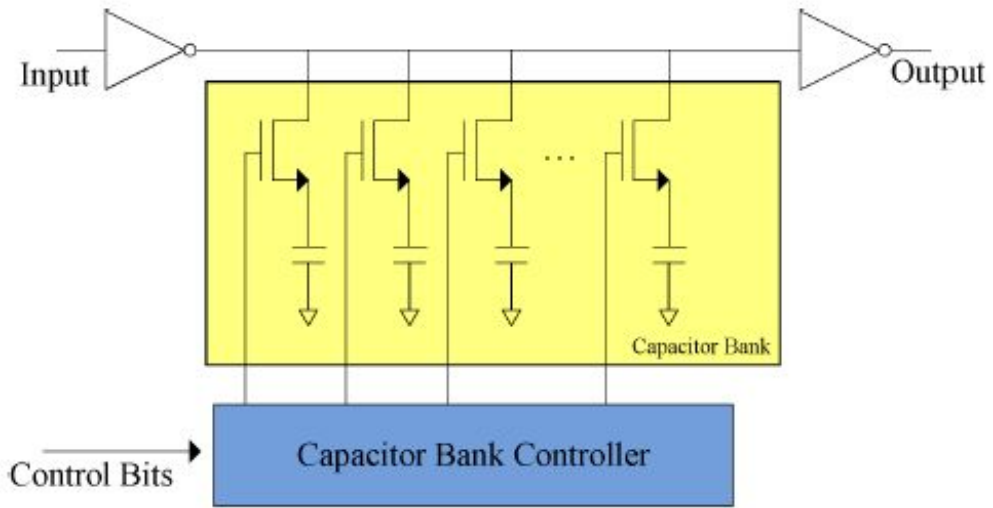
가변 지연 시간 버퍼(Adjustable Delay Buffer, ADB)에는 두가지 종류가 있다. 하나는 캐퍼시터 기반 가변 지연 시간 버퍼(Capacitor based ADB, CADB)이고, 다른 하나는 인버터 기반 가변 지연 시간 버퍼(Inverter based ADB, IADB)이다.

CADB는 캐퍼시터를 보관하는 캐퍼시터 뱅크(Capacitor Bank)를 갖고 있으며, 회로의 전압 모드에 따라 캐퍼시터 뱅크에 있는 캐퍼시터를 활성화/비활성화 시켜 가변 지연 시간 버퍼의 지연 시간을 조절한다. 이때, 캐퍼시터 뱅크 내부의 캐퍼시터의 활성화/비활성화는 컨트롤 박스를 통해 조절하며, 컨트롤 박스는 전압 모드를 입력값으로 받아, 활성화/비활성화 시킬 캐퍼시터의 정보를 출력값으로 내보낸다.

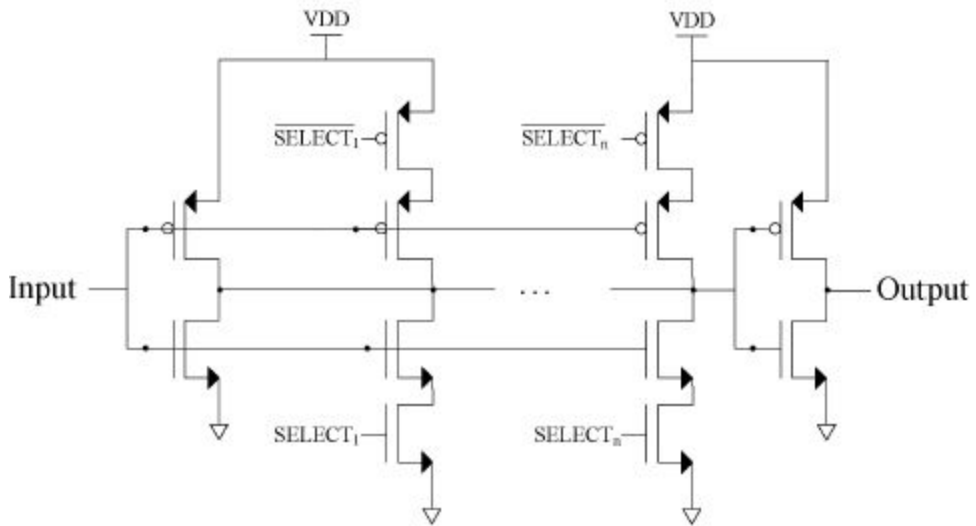
IADB는 입력 인버터 단에 활성화/비활성화 가능한 인버터들이 병렬로 연결된 인버터 체인(Inverter Chain)을 갖고 있으며, 회로의 전압 모드에 따라 인버터 체인에 있는 인버터를 활성화/비활성화 시켜 가변 지연 시간 버퍼의 지연 시간을 조절한다. 이때, 인버터 체인 내부의 인버터의 활성화/비활성화는 컨트롤 박스를 통해 조절하며, 컨트롤 박스는 전압 모드를 입력값으로 받아, 활성화/비활성화 시킬 인버터의 정보를 출력값으로 내보낸다.

CADB와 IADB의 구조는 그림 2.1의 (a), (b) 와 같다.

CADB는 캐퍼시터들을 모아둔 캐퍼시터 뱅크(Capacitor Bank)를 가지고 있다. 캐퍼시터 뱅크 컨트롤러(Capacitor Bank Controller)는 신호 라인(signal line)을 통해서 파워 모드 정보를 컨트롤 비트(control bits)로 받고, 그에 맞춰 캐퍼시터 뱅크 내부의 캐퍼시터들을 활성화/비활성화 시켜서 CADB의 전파 지연(propagation delay)을 변화시킨다.



(a)



(b)

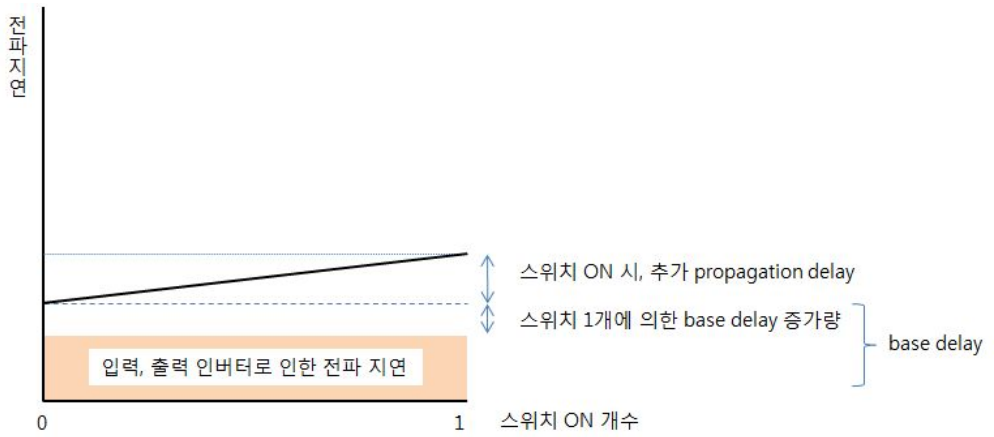
그림 2.1 두 종류의 ADB의 구조[1]. (a) CADB(Capacitor based ADB)의 구조: MOSFET 스위치를 ON/OFF 함으로써 캐피시터를 활성화/비활성화 시켜 전파 지연(propagation delay)을 조절한다. (b) IADB(Inverter based ADB)의 구조: Input단에 병렬 연결된 inverter 들을 활성화/비활성화 시켜 전파 지연(propagation delay)을 조절한다.

IADB는 ON/OFF 스위치를 가진 인버터들이 연결된 인버터 체인 (inverter chain)을 가지고 있다. 그림 2.1-(b)에는 나와 있지 않지만 IADB도 CADB와 마찬가지로 컨트롤 비트(control bits)를 입력으로 받는 인버터 체인 컨트롤러(Inverter Chain Controller)에서 인버터들의 활성화/비활성화를 제어하여 IADB의 전파 지연을 조절한다. 이때, CADB와 IADB는 캐퍼시터 뱅크/인버터 체인을 제어하는 스위치의 ON/OFF 상태에 따라, 전파지연 성향에 있어서 대조적인 결과를 보여준다.

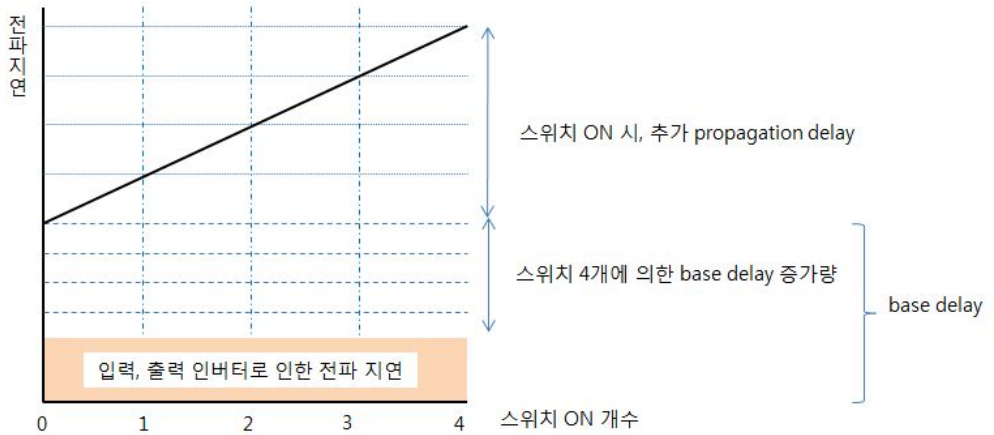
CADB의 전파 지연은 입력단과 출력단에 존재하는 인버터에 의한 전파 지연, 캐퍼시터 뱅크에 존재하는 스위치와 캐퍼시터에 의한 전파 지연의 합으로 구해진다. 입력단과 출력단의 인버터에 의한 전파 지연은 스위치 ON/OFF 여부에 무관하게 항상 동일한 값을 유지하므로, 우리는 스위치 ON/OFF 여부에 따라 스위치와 캐퍼시터의 특성이 전파 지연에 미치는 영향을 알아보기로 하겠다.

ADB의 캐퍼시터 뱅크/인버터 체인을 제어하는 스위치가 모두 OFF 상태일 때의 전파 지연을 기본 지연(base delay)이라 정의하자. 또한 캐퍼시터 1개로 구성된 캐퍼시터 뱅크를 가진 CADB는 CADB_1이라 하고, 캐퍼시터 4개로 구성된 캐퍼시터 뱅크를 가진 CADB를 CADB_4 라고 부르기로 하자. CADB 구조상, 스위치 개수와 캐퍼시터 개수는 일치한다. 이때, 동일한 W값을 갖는 스위치와, 균일한 크기의 캐퍼시터로 이루어진 CADB_1과 CADB_4 의 스위치 ON 개수에 의한 전파지연 그래프 경향을 살펴보면 그림 2.2와 같다.

그림 2.2에서 알 수 있듯이, 캐퍼시터 뱅크 내부에 존재하는 스위치의 개수가 늘어날수록, 기본 지연(base delay)가 선형 비례로 증가하고, ON 상태인 스위치가 많아질수록, 전파 지연이 선형 비례로 증가한다. 이 때, 스위치와 캐퍼시터가 CADB의 전파 지연에 끼치는 영향은 각각 다르다. 스위치의 존재는 OFF 상태일 때의 기본 지연의 값을 증가시키는 요인이며, 스위치가 ON 상태일 때, 스위치의 W값에 비례하여 전파 지연을 증가시킨다. 스위치에 연결된 캐퍼시터는 스위치가 OFF 상태일 때는 기본 지연과 전파 지연에 전혀 영향을 끼치지 않지만, 스위치가 ON 상태일 때는 전파 지연의 증가에 영향을 미친다. 하지만 CADB에서 실질적으로 전파



(a)



(b)

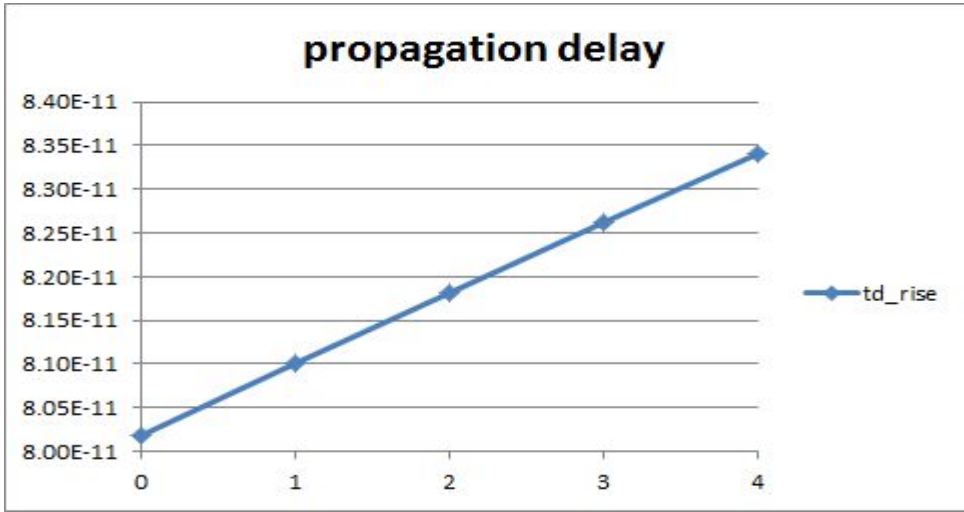
그림 2.2 스위치 개수에 따른 CADB의 전파 지연 특성의 분석. (a) CADB_1: 캐퍼시터 1개로 이루어진 캐퍼시터 뱅크를 갖는 CADB의 스위치 ON 개수에 따른 전파 지연 그래프 (b) CADB_4: 동일한 크기의 캐퍼시터 4개로 이루어진 캐퍼시터 뱅크를 갖는 CADB의 스위치 ON 개수에 따른 전파 지연 그래프. 그래프에서 알 수 있듯이, 캐퍼시터 뱅크에 캐퍼시터의 개수가 증가할수록, 전파 지연의 변화폭이 커지지만, 동시에 캐퍼시터 제어 스위치에 의한 기본 지연(base delay) 또한 증가했음을 알 수 있다.

지연의 증가에 크게 영향을 미치는 것은 캐퍼시터의 크기보다 스위치의 W값의 크기라 할 수 있다. 스위치의 W값과 캐퍼시터의 크기를 각각 변경해주면서 실험을 진행하였을 때, 전파 지연은 스위치의 W값에 비례하여 선형으로 증가한 반면, 캐퍼시터의 크기 변화에는 그 증가폭이 작았다. 캐퍼시터의 크기가 2배, 3배 변화하는 동안 전파 지연의 증가량은 2배, 3배가 아닌, 1.1배 ~ 1.3배 정도의 증가량만을 보였으며, 캐퍼시터의 크기 증가만으로는 특정 한도 이상으로 전파 지연을 증가시킬 수 없었다. 따라서 캐퍼시터의 사이즈를 증가시키는 것 보다 스위치의 W값을 증가시키는 것이 훨씬 효율적인 CADB 설계라고 할 수 있다.

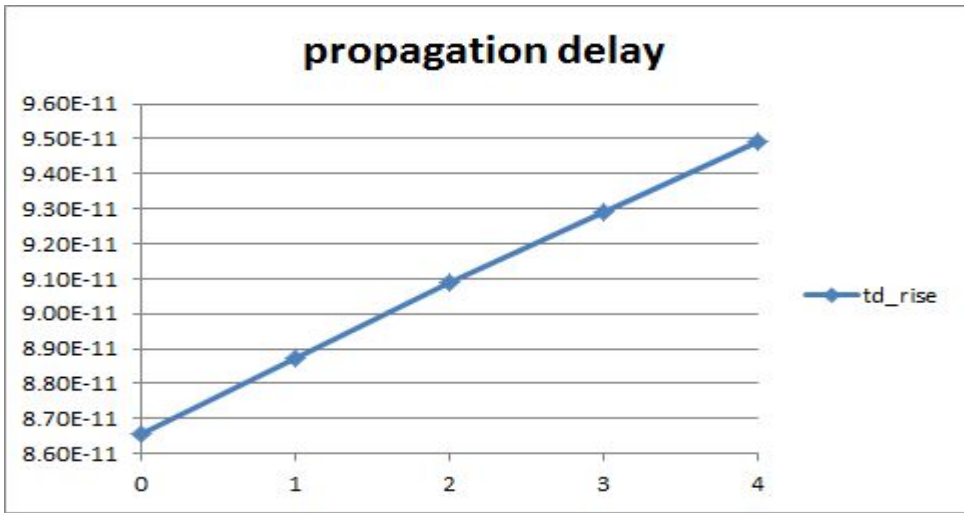
정리하자면 스위치 OFF 상태에서는 캐퍼시터의 크기는 전파지연에 영향을 끼치지 않으며, 스위치의 개수에 따라 기본 지연이 증가하고, 전파지연의 크기를 조절하려면 캐퍼시터의 크기를 조절하는 것 보다 스위치의 W값을 조절하는 것이 훨씬 효율적이라고 할 수 있겠다.

그림 2.2에서는 동일한 W값을 가지는 스위치, 균일한 크기의 캐퍼시터를 갖는 CADB의 특성에 대해 알아보았고, 이 경우 ON 상태의 스위치 개수에 비례하여 전파지연이 선형적으로 증가함을 알 수 있었다. 하지만 다중 전력 모드에서 삽입된 CADB에 요구하는 전파 지연 값의 변화가 선형적이기를 바라는 것은 지나치게 낙관적인 전망이다. CADB가 실제 회로에서 유용하게 활용되기 위해서는 ON 상태의 스위치 개수에 비례하는 전파 지연을 갖는 것이 아니라, 각 전력 모드에 맞게 다양한 범위의 전파 지연 값을 가질 수 있어야 한다. 본 논문에서 전파 지연의 증가폭을 조절할 수 있는 가장 적합한 방법은 각 스위치의 W값을 조절하는 것이다.

그림 2.2를 참고하여 앞에서 언급했듯이, 동일한 W 값을 갖는 스위치가 ON 상태가 될 때마다 CADB의 전파 지연은 동일한 증가폭을 갖는다. 하지만 스위치의 W값이 증가하면, 거기에 비례하여 CADB의 기본 지연(base delay) 또한 증가하는 문제가 있다. 캐퍼시터의 경우, 스위치와 달리, 스위치 OFF 상태에서 기본 지연이 증가하지 않는 장점이 있다. 하지만 캐퍼시터의 크기 조절로는 전파 지연의 변화 폭을 유의미하게 증가시키지 못한다는 단점이 있다. 결국 원하는 크기의 전파 지연을 얻기 위해서는 스위치의 W값을 조절하는 것이 불가피하다.



(a)



(b)

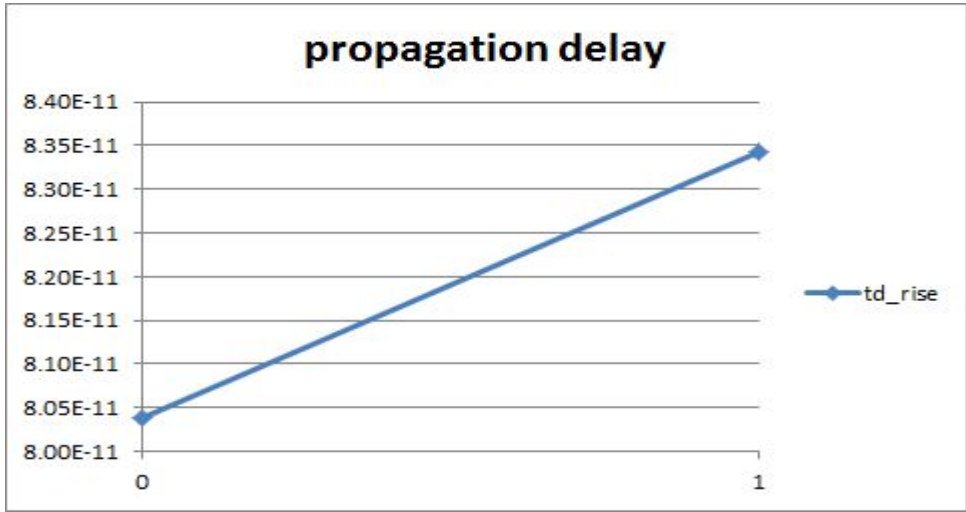
그림 2.3 스위치 개수에 따른 CADB의 전파 지연 특성의 분석. (a) CADB_4_45: W값이 45 nm인 스위치 4개로 이루어진 CADB의 전파 지연 그래프 (b) CADB_4_135: W값이 135nm 인 스위치 4개로 이루어진 CADB의 전파 지연 그래프. (b)의 스위치의 W값은 (a)의 스위치의 W 값의 3배이며, 거기에 비례하여 전파지연의 증가폭은 (b)가 (a)의 3배이다.

그림 2.3의 (a), (b)는 각각 스위치 4개로 이루어진 CADB의 스위치 ON 개수에 따른 전파지연 그래프이다. 2.3-(a)와 2.3-(b)의 차이점이라면, CADB의 캐퍼시터 뱅크 내부의 스위치의 W값이 2.3-(a)의 경우엔 $W = 45\text{nm}$, 2.3-(b)의 경우엔 $W = 135\text{nm}$ 로, 2.3-(b)의 스위치의 W값이 2.3-(a)의 스위치의 W 값보다 3배 크다는 점이다. 실험 결과, CADB의 기본 지연은 각 스위치의 W값의 크기에 비례하여 증가했고, 전파 지연의 변화폭은 ON 상태인 스위치의 W값의 크기에 비례하여 증가했다.

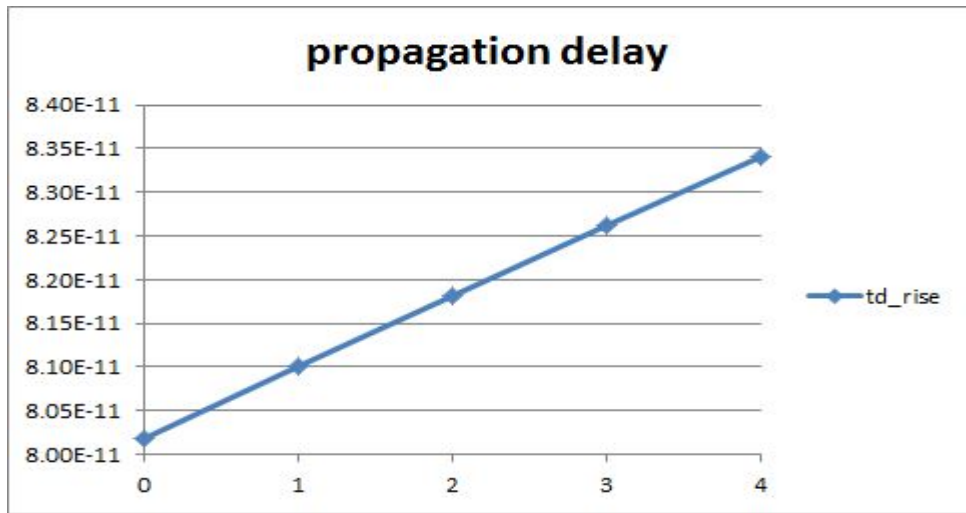
그림 2.3에서는 스위치의 개수가 동일하게 4개인 CADB_4 간의 전파 지연을 비교하였다. 그런데 다중 전력 모드의 개수가 다를 경우엔 캐퍼시터 뱅크 내부의 스위치와 캐퍼시터 개수가 달라진다. 이 경우, CADB_4에서 측정했던 스위치 W값의 변화에 따른 기본 지연, 전파 지연의 증가폭 데이터를 CADB_1 이나 CADB_6 등에 동일하게 적용시킬 수 있을 것인지를 확인해야 한다. 만약 전력 모드의 개수에 따라 CADB의 전파 지연 조절에 필요한 스위치 W값이 달라진다면, CADB 회로 설계는 난항을 겪을 것이기 때문이다.

그림 2.4는 스위치 개수를 변화시키고 스위치의 총 W 값을 동일하게 맞춘 CADB_1과 CADB_4의 전파지연 그래프이다. 그림 2.4-(a)의 $W = 180\text{nm}$ 인 스위치 1개로 구성된 CADB_1과 그림 2.4-(b)에서 $W = 45\text{nm}$ 인 스위치 4개로 구성된 CADB_4는 기본 지연, 전파 지연, 전파 지연 증가폭에서 동일한 값을 나타내었다. 그림 2.4를 통해서 우리는 CADB의 기본 지연과 전파 지연은 스위치의 개수에는 영향을 받지 않고, 스위치의 W값의 총 합에 의해 그 값이 결정됨을 알 수 있다.

정리하자면, CADB는 캐퍼시터 뱅크에서 활성화(ON)상태의 스위치 개수가 증가할수록 전파지연이 증가하며, 캐퍼시터 뱅크 내부의 스위치의 W값의 총합에 선형으로 비례하는 기본 지연과 지연 시간을 가진다고 하겠다.



(a)



(b)

그림 2.4 스위치 개수에 따른 CADB의 전파 지연 특성의 분석. (a) CADB_1: W값이 180 nm인 스위치 1개로 이루어진 CADB의 전파 지연 그래프 (b) CADB_4: W값이 45nm 인 스위치 4개로 이루어진 CADB의 전파 지연 그래프. 기본 지연과 최종 전파지연은 (a)와 (b) 둘다 동일함을 알 수 있다. 이는 스위치의 개수에 무관하게 스위치의 W값의 총합이 같다면, 기본 지연과 전파 지연의 값은 동일하다는 것을 나타낸다.

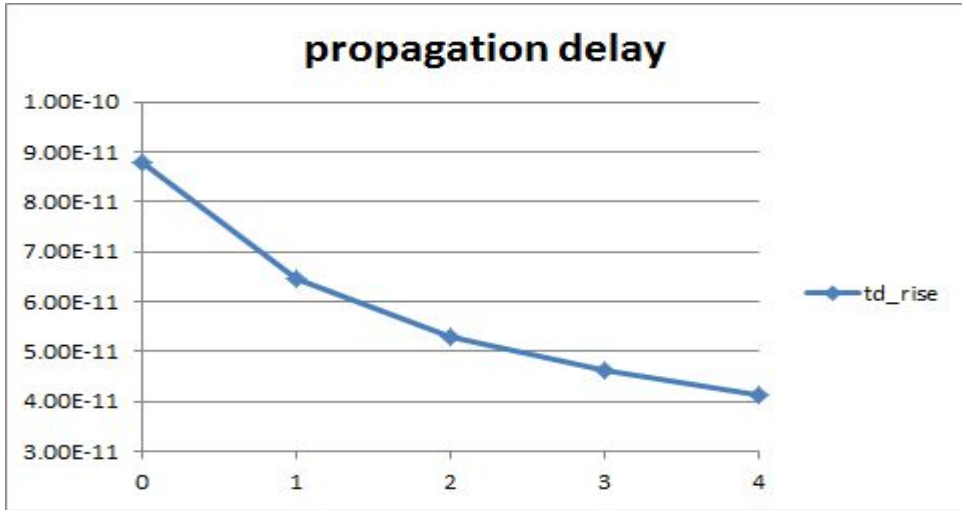


그림 2.5 IADB의 전파 지연. 동일한 W의 인버터로 구성된 인버터 체인을 갖고 있는 IADB의 스위치 ON 개수에 따른 전파 지연 그래프. 반비례하여 감소하는 경향을 보인다.

그림 2.1에서 나타나 있는 IADB의 구조는 2X 인버터, 4X 인버터, 8X 인버터의 구조와 유사하다. IADB는 입력단의 인버터에 스위치를 포함하고 있는 인버터들이 인버터 체인을 구성하여 병렬로 연결된 구조이며, 스위치로 인버터 체인의 인버터를 활성화시킴으로써, driving strength가 강해져 2x 인버터, 3x 인버터, 4x 인버터 등과 유사한 동작을 수행할 수 있다. 여기에서 예상할 수 있는 것은 활성화 된 인버터의 개수가 증가할수록 IADB의 전파지연이 줄어들 것이라는 점이다.

CADB가 스위치 ON 수가 증가할수록 전파 지연이 증가하는 경향을 보인다면, 그림 2.5에서 알 수 있듯이 IADB는 그와 반대로 ON 상태의 스위치의 개수에 따라 전파 지연이 감소하는 경향을 보인다. 그림 2.5의 IADB는 동일한 W값의 인버터로 구성된 인버터 체인을 갖고 있으며, ON 상태인 스위치의 개수가 증가할수록 전파 지연이 줄어드는 경향을 보이고, 균등한 W 값을 갖는 인버터 체인에 대하여 IADB의 전파 지연은 비선형을 보임을 알 수 있다. 또한 그림 2.5에서 보이듯이 스위치가 켜질수록 전파지연의 감소폭이 줄어드는데, 이는 인버터의 개수가 늘어날수록

증가하는 기본 지연이 인버터를 병렬 연결함으로써 줄일 수 있는 전파 지연의 크기보다 크기 때문이며, 스위치의 수를 늘려도 특정값 이하의 전파 지연을 얻을 수 없음을 보여준다.

동일한 W의 인버터 체인을 가진 IADB에서 전파지연이 비선형 감소를 보이는데, IADB의 전파지연이 선형으로 감소하도록 하려면 인버터 체인 내부 인버터들의 W 값을 스위치를 ON 시키는 순서에 맞춰서 적절한 비율로 맞춰주어야 한다.

입력단의 기본 인버터의 W값을 W_0 , 인버터 체인에서 스위치에 연결된 인버터의 W값을 $W_1, W_2, W_3, \dots, W_N$ 으로 두고, 스위치 ON 개수에 따른(0, 1, 2, ..., N) 전파지연(propagation delay)을 $d_0, d_1, d_2, d_3, \dots, d_N$, 스위치는 $W_1, W_2, W_3, \dots, W_N$ 에 연결된 스위치가 차례대로 ON 된다고 하자. 이 때, 인버터의 W값과 전파지연간의 관계는 다음과 같은 비례관계가 성립한다.

$$W_0 \propto d_0 \quad (2.1)$$

$$\sum_{n=0}^N W_n \propto \frac{1}{d_N} \quad (2.2)$$

예를 들어, 입력단의 기본 인버터를 제외하고, 인버터 체인을 구성하고 있는 인버터를 4개라고 하자. 이때 스위치 4개가 전부 OFF 상태일 때의 전파지연 값을 1, 스위치 4개가 전부 ON 상태일 때의 전파지연 값을 0.2라고 하자. 중간에 각각 스위치가 1개씩 켜질 때마다 감소하는 전파지연은 0.2라 둔다. 즉, 스위치가 1개씩 켜질 때마다 전파지연은 1, 0.8, 0.6, 0.4, 0.2 가 된다고 하자.

ON개수	0	1	2	3	4
W	W_0	W_1	W_2	W_3	W_4
d	1	0.8	0.6	0.4	0.2

이 경우, 위의 식 2.1과 2.2를 적용하여 W값을 계산하면 다음과 같은 비례가 나오며 해당 IADB의 전파지연은 그림 2.6과 같다.

$$W_0 : W_1 : W_2 : W_3 : W_4 = 1 : \frac{1}{4} : \frac{5}{12} : \frac{5}{6} : \frac{5}{2} \quad (2.3)$$

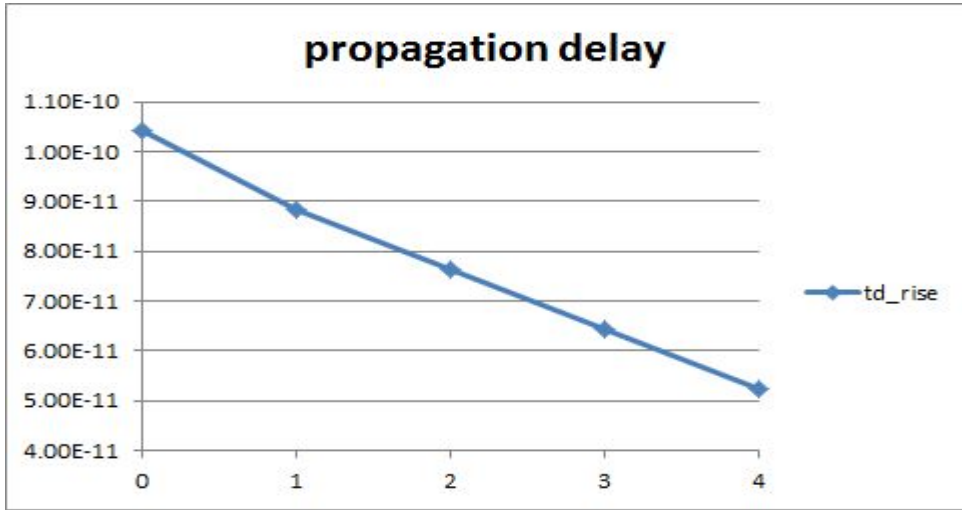


그림 2.6 식 2.3에서 구한 W 비율을 적용한 IADB의 전파지연. 전파지연이 선형으로 감소한다.

IADB를 실제 구현한다면, 각 전력 모드에 따른 요구 전파 지연값을 이용하여 인버터 체인의 W값들을 거기에 맞춰 변화시킬 필요가 있다. 하지만 식 2.2에서 보이듯이, 인버터 체인에 연결된 인버터들이 각각 독립적이지 않아, 각 전력 모드에 따른 전파지연을 맞추기 위해선 다른 인버터들의 W값을 참조해야 한다. 또한 W의 합과 전파지연의 관계가 반비례 관계이기 때문에, 전파지연 $d = 0.01$ 과 같이 매우 작은 경우엔 인버터의 W값이 비현실적으로 큰 값으로 설정될 수도 있다.

앞에서 살펴본 바와 같이, CADB는 활성화되는 스위치의 W값의 총 합에 비례하여 전파 지연이 선형으로 증가하는 경향을 보인다. 반면, IADB는 활성화되는 인버터의 개수가 늘어날수록 점점 전파 지연이 줄어드는 경향을 보인다. 본 논문은 CADB의 전파 지연이 활성화된 스위치의 W값의 총 합에 선형으로 비례하여 증가한다는 특성에 주목하였다. CADB의 이러한 직관적인 특성은 캐퍼시터 뱅크의 컨트롤 박스 로직(control box logic)을 용이하게 구현할 수 있게 한다. 따라서 우리는 CADB의 캐퍼시터 뱅크를 제어할 컨트롤 박스 로직을 구현하는 알고리즘을 다루고자 한다.

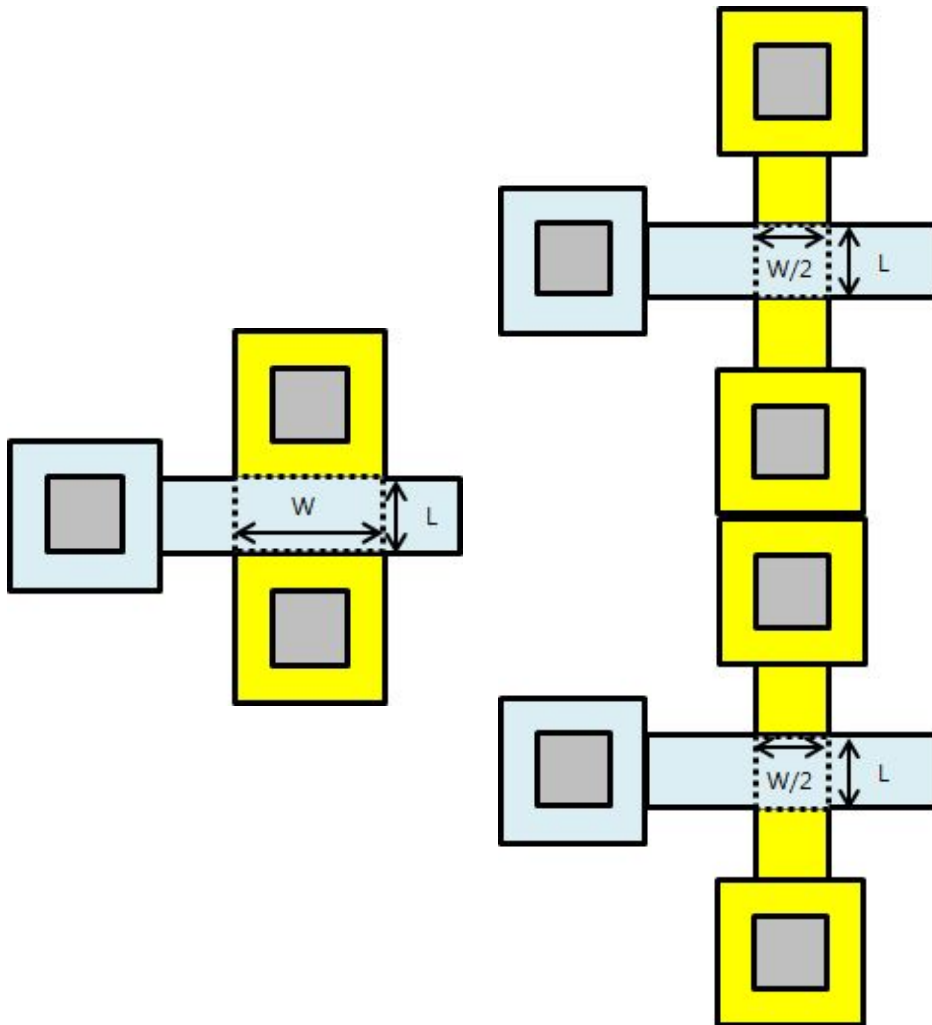
2.2 CADB 컨트롤 박스 로직 구현

이 절에 한해서 캐퍼시터 बैं크 내부의 스위치와 스위치에 연결된 캐퍼시터를 한데 묶어서 캐퍼시터로 부르기로 한다. 이는 로직 구현 과정에서 스위치의 W값을 언급하다가 캐퍼시터 बैं크라는 개념에 혼란을 주는 것을 막기 위함이다. 차후 언급되는 캐퍼시터의 사이즈라는 용어는 스위치의 W값과 같은 의미로 이해하여 주길 바란다.

CADB의 컨트롤 박스 로직을 구현하기에 앞서 생각해야 할 문제가 있다. 그것은 캐퍼시터 बैं크에 있는 캐퍼시터의 사이즈와 개수에 대한 것이다. 만약, 파워 모드가 4 종류가 있고, 각 모드에서 필요한 캐퍼시터 사이즈e 값이 [0, 30, 100, 70] 이라고 하자. 이때 캐퍼시터 बैं크에 사이즈가 각각 30, 100, 70 인 캐퍼시터가 들어있어서, 모드에 따라서 각각 1개의 캐퍼시터를 활성화해준다고 한다면, 이는 불필요하게 회로의 공간을 낭비하는 알고리즘이라고 볼 수 있다. 왜냐하면 사이즈가 각각 30과 70인 캐퍼시터 2개를 사용하면 사이즈가 100인 캐퍼시터의 역할을 해낼 수 있기 때문이다.

혹시나 누군가가 사이즈가 10인 캐퍼시터 10개를 캐퍼시터 बैं크에 놓고 사용하려 한다면 이것 또한 공간의 낭비를 가져온다. 수치상으로 드러난 캐퍼시터 사이즈의 총합은 100으로, 최적화 된 것처럼 보이지만, 그림 3에서 알 수 있듯이, 캐퍼시터의 개수가 많아지는 것은 그 자체로 공간을 낭비하는 행위이다. 트랜지스터가 하나 증가할 때마다 게이트, 소스, 드레인 생성으로 공간을 소모하기 때문이다. 캐퍼시터의 사이즈는 그림 2.7-(a)의 W로 이해할 수 있다. 그림 2.7-(a) 를 사이즈가 20인 캐퍼시터 1개로 가정하고, 그림 2.7-(b) 를 사이즈가 10인 캐퍼시터 2개로 가정한다면 둘 중 회로에서 공간을 적게 차지하는 것은 사이즈가 20인 캐퍼시터 1개임은 자명하다. 따라서 캐퍼시터 बैं크 내의 캐퍼시터들은 그 수가 적을수록 공간을 적게 차지하므로 좋다.

결론적으로, 우리가 해야 할 것은 캐퍼시터 बैं크가 모든 파워 모드의

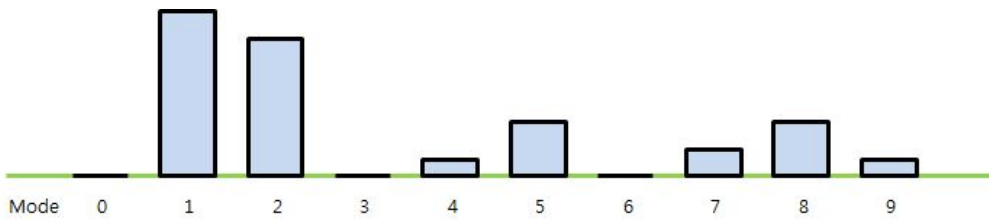


(a)

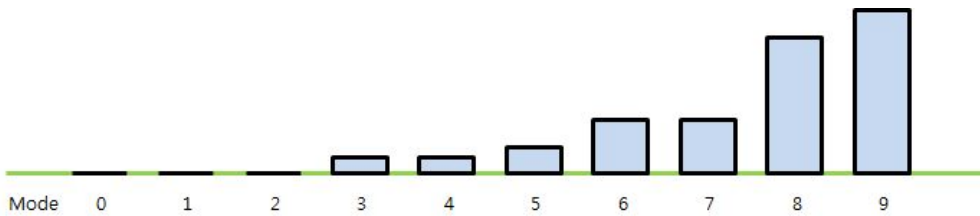
(b)

그림 2.7 트랜지스터의 레이아웃. (a) WL 인 트랜지스터 1개. (b) $(W/2)L$ 인 트랜지스터 2개. $(W/2)L$ 인 트랜지스터 2개를 사용해서 WL 인 트랜지스터 1개와 같은 효과를 낸다. 이 경우, 회로에서 차지하는 영역의 면적이 넓어지므로, 특별한 경우가 아니면 회로 설계에서는 (b) 보다 회로 면적을 적게 차지하는 (a)의 구조를 선호한다.

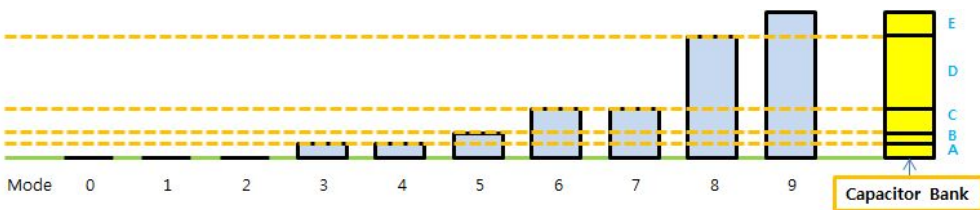
요구에 대응하는 캐퍼시터들을 제공할 수 있도록 하며, 캐퍼시터 뱅크에 포함된 캐퍼시터들의 사이즈와 개수를 최적화 하는 control box logic 알고리즘을 구현하는 것이다. 그리고 그 알고리즘은 그림 2.8에 나타나 있다.



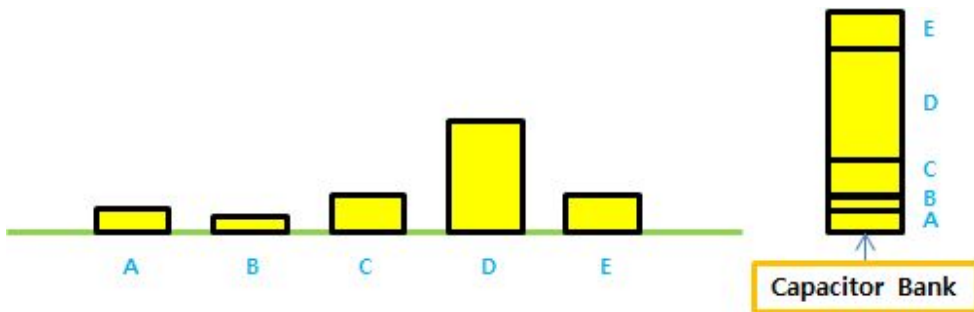
(a)



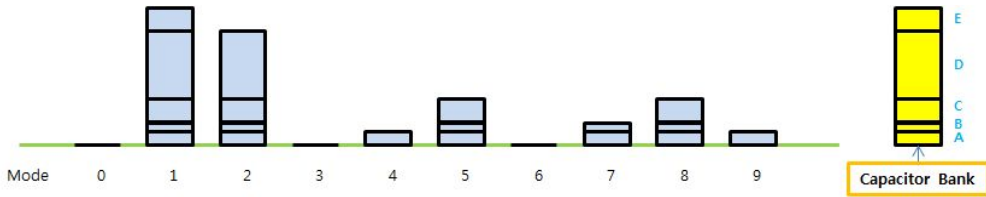
(b)



(c)



(d)



(e)

Mode	CAP
0	0
1	A+B+C+D+E
2	A+B+C+D
3	0
4	A
5	A+B+C
6	0
7	A+B
8	A+B+C
9	A

Mode	E	D	C	B	A
0	0	0	0	0	0
1	1	1	1	1	1
2	0	1	1	1	1
3	0	0	0	0	0
4	0	0	0	0	1
5	0	0	1	1	1
6	0	0	0	0	0
7	0	0	0	1	1
8	0	0	1	1	1
9	0	0	0	0	1

input	Output
0000	00000
0001	11111
0010	01111
0011	00000
0100	00001
0101	00111
0110	00000
0111	00011
1000	00111
1001	00001

input	Output
0-01	00100
010-	00001
1000	00111
0001	11010
-001	00001
0010	01111
01-1	00011

Cap Bank On/Off

2진 변환

Espresso 알고리즘

(f)

그림 2.8 CADB의 캐피터터 뱅크가 가지는 캐피터터들의 개수와 캐피터터의 크기 최적화 알고리즘.

그림 2.8은 CADB control box logic 구현 과정을 나타낸 것이다. 첫째, Mode별로 필요한 캐피터터 값을 가져온다(그림 2.8-a). 둘째, Mode별로 필요한 캐피터터 값을 오름차순 또는 내림차순 정렬한다. 이는 동일한 크기의 필요 캐피터터를 쉽게 찾기 위함이다(그림 2.8-b). 셋째, 인접한 필요 캐피터터들의 차이를 계산하여 캐피터터 뱅크를 만든다(그림 2.8-c). 예를 들어, 모드별로 필요한 캐피터터의 사이즈가 [0, 100, 80, 0, 10, 50, 0, 20, 50, 10] 이고(그림 2.8-a), 오름차순 정렬(그림 2.8-b)시 [0, 0, 0, 10, 10, 20, 50, 80, 100] 이라고 하자. 이때 중복되는 캐피터터들을 제거하면 캐피터터들의 사이즈 값은 [0, 10, 20, 50, 80, 100]이 남는다. 이제 여기서 인접한 캐피터터들의 차이를 계산하면 [10-0, 20-10, 50-20, 80-50, 100-80] 이 되어(그림 2.8-c), 캐피터터 뱅크에 들어갈 캐피터터 사이즈는 [10, 10, 30, 30, 20](그림 2.8-d) 이 된다. 넷째, 캐피터터 뱅크

안의 캐퍼시터들을 합성하여 각 파워 모드에서 필요한 캐퍼시터 값을 만족시키는 로직을 짠다(그림 2.8-e, f). 예를 들면, 그림 2.8-(e) 에서 모드 1에서 필요한 캐퍼시터 크기 100을 만족시키려면 캐퍼시터 बैं크 [10, 10, 30, 30, 20] 안의 캐퍼시터들을 전부 더해야 하므로 $10 + 10 + 30 + 30 + 20 = 100$ 이 된다. 만약 모드 5 에서 필요한 캐퍼시터 크기 50 을 만족시키려면 캐퍼시터 बैं크에서 10, 10, 30을 가져와 $10 + 10 + 30 = 50$ 을 만들면 된다. 모드 0, 3, 6 은 필요 캐퍼시터 크기가 0 이므로 캐퍼시터 बैं크를 사용하지 않는다.

2.3 ADB 컨트롤 박스에 연결되는 컨트롤 신호 라인의 길이

2.1에서 언급했듯이, 그림 2.1-(a)처럼, ADB들이 다중 전력 모드에 따라 ADB의 전파 지연(propagation delay)을 변경하기 위해서는 캐퍼시터 बैं크 컨트롤러(컨트롤 박스)에 컨트롤 비트(Control bits)로 전력 모드 정보를 전송할 신호 라인(signal line)이 반드시 필요하다. 그리고 회로 내에 ADB의 개수가 많을수록 ADB의 컨트롤 박스에 연결되는 컨트롤 신호 라인(control signal line)의 길이도 길어지게 되므로, ADB의 컨트롤 신호 라인의 길이를 최소화 할 필요가 있다. 회로 상에 있는 ADB들에 연결되는 컨트롤 신호 라인의 길이는 Rectilinear Steiner Minimal Tree 알고리즘[3]을 사용해 최적화 할 수 있고, ISPD'09 CTS Contest 벤치마크에 ADB 배치 알고리즘[2]을 적용한 뒤, ADB에 연결된 control signal line의 최적화된 길이는 표 2.1과 같았다.

표에서 보듯이 ADB 신호 total wirelength는 스큐 바운드(skew bound)에 따라 다르지만 대략 클럭 트리 total wirelength의 10~20%에 해당하는 길이를 가졌다. 단, 표에서 계산한 ADB 신호 라인은 전력 모드가 2 가지 이하일 때를 가정한 것이다. 즉, ADB 하나당 신호 라인이 한 개만 들어가는 상황을 가정한 것이다. 만약 전력 모드가 3~4가지 라면,

표 2.1: 각 벤치마크에서의 ADB 컨트롤 신호 라인(ADB control signal line) 길이와 클럭 신호 라인(clock signal line) 길이 비교. TWL은 total wirelength의 약자이다.

벤치마크	TWL (um)	Skew bound	ADBsignal TWL(um)	ADB/TWL (%)	ADB 개수
ispd09f11	178228.24	skew=30	36583.11	20.53	22
		skew=40	31685.08	17.78	18
		skew=50	23296.42	13.07	12
ispd09f12	155361.99	skew=30	33326.89	21.45	25
		skew=40	26422.55	17.01	16
		skew=50	20262.86	13.04	12
ispd09f21	187578.09	skew=30	36253.16	19.33	22
		skew=40	28784.71	15.35	17
		skew=50	19975.56	10.65	10
ispd09f22	104109.06	skew=30	9785.38	9.4	6
		skew=40	5296.34	5.09	2
		skew=50	0	0	0
ispd09f31	384095.32	skew=30	70072.91	18.24	33
		skew=40	46558.25	12.12	14
		skew=50	22271.38	5.8	8
ispd09f32	292904.3	skew=30	41904.06	14.31	20
		skew=40	30045.92	10.26	13
		skew=50	22197.04	7.58	7
ispd09f33	296295.7	skew=30	42188.85	14.24	20
		skew=40	32709.48	11.04	12
		skew=50	30102.75	10.16	9
ispd09f34	234850.26	skew=30	40210.58	17.12	23
		skew=40	23301.32	9.92	10
		skew=50	22413.15	9.54	8

ADB에 들어가는 신호 라인은 두개가 될 것이며, 신호 라인의 total wirelength는 2배로 늘어날 것이다. 즉, 다중 전력 모드의 개수가 N이라면, ADB 신호 라인의 total wirelength는 $\lceil \log_2(N) \rceil$ 에 비례하여 증가한다.

제 3 장 구현

2.1 의 CADB와 IADB의 특성 분석에는 HSPICE를 사용하였고 그림 2.1에서 나온 CADB, IADB 구조에 따라 구현하였다. ON 시킨 스위치의 개수, 스위치와 인버터의 W값, 입력 전압의 크기, 캐퍼시터 뱅크/인버터 체인 내부에 존재하는 캐퍼시터/인버터 개수를 변화시켜 가면서 CADB와 IADB의 전파 지연을 측정하였다.

2.2 에서 CADB 컨트롤 박스는 컨트롤 신호 비트(control signal bits)를 입력으로 받아서 캐퍼시터 뱅크 의 어느 캐퍼시터들을 활성화 할지를 출력으로 알려준다. 이것이 가능하기 위해선, 우선 각 전력 모드별로 필요한 전파 지연(propagation delay) 정보를 각 전력 모드 별로 필요한 캐퍼시터 사이즈 정보로 변환한다. 그 다음, 이 캐퍼시터 사이즈 정보를 입력으로 받아 캐퍼시터 뱅크에 들어갈 캐퍼시터 사이즈와 개수 정보를 알고리즘에 따라 구하고, 각 파워 모드별로 캐퍼시터 뱅크의 어느 캐퍼시터들을 활성화 시킬지에 대한 정보도 구해 출력으로 내보낸다. 그림 2.8-f에서 왼쪽에서 세 번째에 있는 테이블이 컨트롤 신호 비트를 입력으로 하고, 활성화된 캐퍼시터들의 정보를 출력으로 갖는 lookup table이다. 마지막으로 구현한 알고리즘으로 얻은 lookup table을 로직으로 구현하기 위해 2-level logic optimization 툴인 ESPRESSO[4]를 사용하였고, 그 결과가 그림4-f의 네 번째 표이다.

2.3에서 ADB의 control signal line의 길이를 최소화 하는 Rectilinear Steiner Minimal Tree 구조는 FLUTE[1] 알고리즘을 사용해 구하였다.

제 4 장 결론

이 논문은 다중 전력 모드에 사용되는 가변 지연 시간 버퍼인 CADB와 IADB의 특징을 살펴보았으며, CADB의 캐퍼시터 뱅크 컨트롤 박스로직 구현 알고리즘에 대한 이슈를 최초로 다루었다. 또한 제시한 알고리즘을 통해 다중 전력 모드에 대한 캐퍼시터 뱅크 제어 lookup table을 구현했고, 캐퍼시터 뱅크를 최적화함으로써, 회로 상에 불필요한 공간의 낭비를 줄였다. ADB 컨트롤 박스에 연결되는 ADB 컨트롤 신호 라인의 길이를 구해본 결과 회로상에서의 비중이 적지 않아 컨트롤 신호 라인의 길이를 최적화 할 필요가 있음을 보였다.

참고문헌

- [1] Kyoung-Hwan Lim, Deokjin Joo and Taewhan Kim, “An Optimal Allocation Algorithm of Adjustable Delay Buffers and Practical Extensions for Clock Skew Optimization in Multiple Power Mode Designs,” *IEEE*, March 2013
- [2] Juyeon Kim, Deokjin Joo and Taewhan Kim, “An Optimal Algorithm of Adjustable Delay Buffer Insertion for Solving Clock Skew Variation Problem,” *DAC*, 2013
- [3] Chris Chu and Yiu-Chung Wong, “FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design,” *IEEE*, JANUARY 2008
- [4] ESPRESSO, <http://www.sontrak.com/downloads.html> [URL]

ABSTRACT

Analysis on Adjustable Delay Buffer Design and Control Circuit for Multiple Power Mode Designs

Satisfying the clock skew constraint is an important task in the clock tree synthesis. Furthermore, as the design trend moves to that of multiple voltage modes, in which the supply voltage changes as the execution of applications varies, meeting the clock skew constraint for all power modes becomes much hard since the clock signal delay also varies dynamically during the execution. Recently, many works have shown that adjustable delay buffer (ADB) whose delay is adjustable dynamically can effectively solve the clock skew variation problem in the multiple power modes.

In this paper, we characterize the CADB (capacitor based ADB), propose the algorithm which optimizes the capacitor of ADB and realizes the ADB control box logic, and see the optimal RSMT(Rectilinear Steiner Minimal Tree) of ADBs' signal line.

Keywords : adjustable delay buffers, control box logic, propagation delay, multiple power mode

Student Number : 2012-20743