

논문 2004-41SD-2-9

RLC 연결선의 버퍼 삽입 방법

(A Buffer Insertion Method for RLC Interconnects)

김 보 겸*, 김 승 용*, 김 석 윤**

(Bo Kyeom Kim, Seung Yong Kim, and Seok Yoon Kim)

요 약

본 논문은 인덕턴스 성분을 포함한 단일 도선 및 트리 구조 RLC 연결선의 버퍼 삽입 방법을 제시한다. 이를 위해 먼저 CMOS 버퍼가 구동하는 단일 RLC 도선에 대한 시간 지연의 대수식을 제시한다. 이 수식은 현재의 서브마이크로미터 공정을 위한 n -th power law 기반에서 유도되었으며, 다양한 RLC 부하를 가지고 실험해 본 결과, 실제 SPICE 시뮬레이션 결과에 비해 최대 9% 오차를 갖는 것으로 나타났다. 본 논문은 이 지연 시간 수식을 바탕으로 단일 도선 RLC 연결선을 여러 개로 나누는 버퍼 삽입에 관한 수식과 RLC 트리 연결선의 시간 지연을 최적화하기 위해 삽입될 버퍼의 사이즈를 결정하는 알고리즘을 제시한다. 제시된 버퍼 삽입 알고리즘은 $0.25\mu\text{m}$ CMOS 공정의 트리 연결선에 적용하였으며, HSPICE 결과를 이용하여 정확도를 검증하였다.

Abstract

This paper presents a buffer insertion method for RLC-class interconnect structured as a single line or a tree. First, a closed form expression for the interconnect delay of a CMOS buffer driving single RLC line is represented. This expression has been derived by the n -th power law for deep submicrometer technology and occurs to be within 9 percentage of maximal relative error in accuracy compared with the results of HSPICE simulation for various RLC loads. This paper proposes a closed form expression based on this for the buffer insertion of single RLC lines and the buffer sizing algorithms for RLC tree interconnects to optimize path delays. The proposed buffer insertion algorithms are applied to insert buffers for several interconnect trees with a $0.25\mu\text{m}$ CMOS technology and the results are compared against those of HSPICE.

Keywords : Terms - inductance, interconnect, buffer insertion, tree

I. 서 론

공정 기술이 발달되고, 다이(die) 면적이 증가함에 따라, VLSI(Very Large Scale Integration) 회로 내 전체 시간 지연 중에서 연결선 지연이 차지하는 비중이 게이트 지연에 비해 증가되고 있다^{[1][3]}. 그러나 연결선 시간 지연을 알기 위하여 게이트에 실제 연결선을 그대로 연결하여 시뮬레이션을 한다면 많은 시간이 걸린다. 그러므로 정확도를 잃지 않으면서도 효율적으로

연결선 시간 지연을 해석할 수 있는 연결선 모형이 필요하다. 가장 간단한 연결선 모형화 방법은 전체 연결선의 총 커패시턴스로 모형화하는 것이다. 이 모형화 방법은 연결선의 저항 성분이, 논리 소자가 연결선 커패시턴스의 일부를 보지 못하도록 만드는 저항 차폐 효과(resistance shielding effect)를 무시하여 실제 시간 지연보다 큰 지연 값을 계산해 내는 결과를 초래한다. 이러한 오차는 논리 소자의 스위칭 속도가 빨라지고 내부 저항이 줄어드는 반면 연결선의 저항은 커지는 기술 발전 경향에 비추어 논리 회로 시간 지연 값 계산에서 상당한 요소가 되고 있다. 그리하여 증가되는 연결선의 저항 성분을 고려한 RC 연결선 모형화가

* 학생회원, ** 정회원, 숭실대학교 컴퓨터학과

(Soongsil university, Graduate school, Computing of School)

※ 본 연구는 숭실대학교 교내연구비 지원으로 이루어 졌음

접수일자 : 2002년 11월 18일, 수정완료일 : 2004년 2월 13일

도입되었다. 현재는 논리 소자의 스위칭 속도가 더욱더 빨라지고 동작 주파수 역시 높아지고 있다. 게다가 연결선의 저항을 줄이기 위한 새로운 물질, 커패시턴스를 줄이기 위한 새로운 유전체가 사용됨으로써 상대적으로 인덕턴스의 영향이 중요시되고 있다^[4-17] 이러한 환경에서 RC가 아닌 RLC 연결선 모형이 필요하다.

버퍼 삽입은 저항 성분이 많은, 길이가 긴 연결선을 구동하기 위해 널리 사용되고 있는 설계 방법이다^{[8]-[10]} RC 연결선에서 시간 지연은 길이의 제곱에 비례하므로, 버퍼를 삽입함으로써 하나의 연결선을 여러 개의 섹션으로 나누는 것은 전체 연결선 시간 지연을 줄이는데 효과적인 방법이다. 삽입된 버퍼는 커패시턴스를 디커플링(decoupling)하여 연결선 시간 지연을 줄인다^[10]. 기존의 버퍼 삽입에 관한 논문들은 주로 RC 연결선에 대해 다루었다. 하지만 공정 기술이 발달되어짐에 따라 연결선에서 인덕턴스의 영향이 더욱 중요시되고 있기 때문에 RLC 연결선에 대한 버퍼 삽입 연구의 필요성이 증대되고 있다. 일반적으로 VLSI 회로 내 연결선은 단일 도선 보다는 트리(tree) 구조로 되어 있다. 그러므로 본 논문에서는 단일 RLC 도선 및 RLC 트리 연결선에서의 버퍼 삽입 방법을 제안하고자 한다. 제안된 버퍼 삽입 알고리즘을 RLC 트리 연결선에 대해 실험하였으며, 결과를 통해 연결선의 인덕턴스 영향이 커지면 버퍼 삽입으로 인한 시간 지연 감소와 반복 회수가 줄어들음을 알 수 있다. 논문에서 제시된 5개의 알고리즘은 버퍼 삽입 위치가 같으므로 동일한 연결선 시간 지연을 갖는다. 그러나 bottom-up 순서로 버퍼를 삽입하는 알고리즘 II, V가 다른 알고리즘에 비해 반복 회수가 적다.

본 논문의 구성은 다음과 같다. 서론에 이어 II절에서는 단일 RLC 도선에서 버퍼를 삽입했을 때의 시간 지연과 연결선 시간 지연을 최소화하는 버퍼의 사이즈와 개수를 수식으로 표현한다. III절에서는 기존의 RLC 트리 연결선에 대한 버퍼 삽입 알고리즘과 본 논문에서 제안한 알고리즘들을 소개하고, IV절에서는 III절에 제시되어 있는 5가지 알고리즘에 대한 시뮬레이션 결과를 보인다. V절 결론으로 본 논문을 맺는다.

II. 단일 RLC 도선에서의 버퍼 삽입

1. CMOS 버퍼가 구동하는 단일 RLC 도선의 시간 지연

그림 1은 CMOS 버퍼가 구동하는 단일 RLC 도선 모형이다. R_t 그리고 L_t , C_t 는 순서대로 연결선의 전체 저항, 전체 인덕턴스, 전체 커패시턴스이다. $R_t = RL$, $L_t = LL$, $C_t = Cl$ 이며, R 그리고 L , C 는 연결선의 단위 길이당 저항, 단위 길이당 인덕턴스, 단위 길이당 커패시턴스이다. 연결선의 길이는 l 이고, C_L 은 출력단에 달린 부하(load) 커패시턴스이다. 입력 전압 V_{in} 은 입력 신호로서 상승 계단 신호(rising step signal)이다.

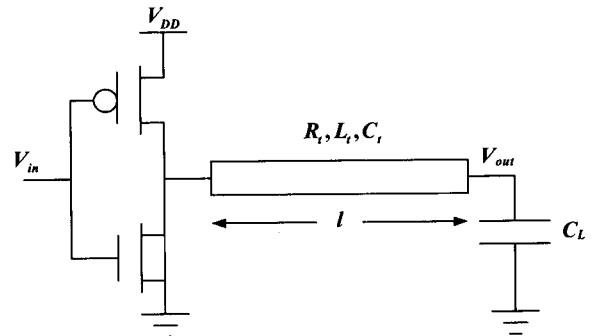


그림 1. 단일 RLC 도선을 구동하는 CMOS 버퍼
Fig. 1. A CMOS buffer driving an RLC interconnect line.

본 논문에서는 상승 입력 신호를 사용한다고 가정하고, 단일 RLC 도선의 시간 지연을 유도하기 위해 NMOS 변수(parameter)를 사용한다. 만약 하강 입력 신호가 쓰여진다면 NMOS 대신 PMOS 변수를 이용하면 된다. CMOS 소자를 n -th power law^[11]에 따라 모형화하면, 포화 영역(saturation region)일 때 드레인과 소스 사이에 흐르는 전류의 양은

$$I_{DS} = P_c \frac{W_d}{L_d} (V_{GS} - V_T)^n \quad (1)$$

이고, 선형 영역(linear region)일 때 흐르는 전류값은

$$I_{DS} = \frac{P_c}{P_v} \frac{W_d}{L_d} (V_{GS} - V_T)^{\frac{n}{2}} V_{DS} = \frac{V_{DS}}{R_r} \quad (2)$$

이다. P_c , P_v 는 공정 상수로서 각각 포화 영역, 선형 영역일 때 CMOS 소자에 흐르는 전류의 특성을 나타낸다. W_d , L_d 는 각각 CMOS 소자의 기하학적인 폭(width)과 길이(length)이다. V_T 는 문턱 전압(threshold voltage)이고, n 은 n -th power law를 사용해서 구해지는 1부터 2사이의 상수 값이다. n -th power law에서 CMOS의 I-V 특성 곡선은 단지 선형 영역에서의 저항 R_r 에 의해 근사된다. R_r 만을 가지고 전체 출력 전압에 대해 선형화하는

것은 CMOS 소자가 포화 영역에서 작동할 때를 무시하기 때문에 무리가 있다.

보다 정확하게 CMOS 버퍼가 구동하는 단일 RLC 도선의 시간 지연을 구하기 위해서는 다음과 같은 두 가지 경우로 나누어서 고려해 보아야 한다. 입력 신호는 가정한대로 상승 계단 신호이다. 첫 번째는 NMOS 소자가 전체적으로 포화 영역에서만 작동한다고 볼 때, 두 번째는 NMOS 소자가 선형 영역에서만 작동한다고 볼 때이다. 첫 번째의 경우 NMOS 소자에 포화 전류가 흐르기 때문에 C_t , C_L 이 방전(discharging)되며, 인덕턴스와 저항은 시간 지연에 영향을 거의 끼치지 않으므로

$$t_{pdsat} = \frac{V_{DD}}{2} \frac{L_n(C_t + C_L)}{W_n P_{Cn} (V_{DD} - V_{Tn})^{n_n}} \quad (3)$$

로 나타낼 수 있다. 두 번째의 경우 버퍼는 R_{tr} 값을 가진

저항으로 대체될 수 있고, 시간 지연은 곡선일치법을 사용하여

$$t_{pdlin} = (e^{-2.9\zeta^{1.35}} + 1.48\zeta) / \omega_n \quad (4)$$

로 나타낼 수 있다. 사용된 변수들의 값은

$$\omega_n = \frac{1}{\sqrt{L_t(C_t + C_L)}} \quad (5)$$

$$\zeta = \frac{R_t}{2} \sqrt{\frac{C_t}{L_t} \frac{R_T + C_T + R_T C_T + 0.5}{\sqrt{1 + C_T}}} \quad (6)$$

$$R_T = \frac{R_{tr}}{R_t}, C_T = \frac{C_L}{C_t} \quad (7)$$

이다. R_T , C_T 는 연결선 기생 임피던스(parasitic impedance)에 대한 버퍼 기생 임피던스의 중요성 정도를 나타낸다^[3].

표 1. 다양한 연결선 기생 임피던스와 부하 커패시턴스에 대한 식 (8)과 SPICE시뮬레이션 결과 비교
Table 1. SPICE simulation results as compared to (8) with various interconnect parasitic impedances and load capacitances.

R _t kΩ	L _t nH	C _T = 0.1			C _T = 0.5			C _T = 1		
		SPICE	(8)	오차율	SPICE	(8)	오차율	SPICE	(8)	오차율
0.5	1000	1045	1062	1.6%	1199	1236	3.1%	1406	1436	2.1%
	100	364	394	8.2%	555	555	0.0%	794	776	2.3%
	10	339	336	0.9%	525	526	0.2%	750	762	1.6%
1	1000	1065	1064	0.1%	1334	1295	2.9%	1706	1617	5.2%
	100	572	569	0.5%	912	897	1.6%	1313	1318	0.4%
	10	567	558	1.6%	895	895	0.0%	1294	1317	1.8%
2	1000	1149	1212	5.5%	1751	1734	1.0%	2543	2468	2.9%
	100	1026	1002	2.3%	1642	1635	0.4%	2392	2427	1.5%
	10	1022	1002	2.0%	1635	1635	0.0%	2383	2427	1.8%

CMOS 소자가 두 영역에서 작동하므로 t_{pdsat} , t_{pdlin} 중 하나의 값만을 가지고 CMOS 버퍼가 구동하는 단일 RLC 도선의 시간 지연을 구하는 것은 실제 시뮬레이션 결과와 많은 차이가 있다. 그리하여 t_{pdsat} , t_{pdlin} 두 개의 값을 사용하면 이보다 정확하게 연결선 지연을 예측할 수 있다^[12]. 곡선일치법을 사용하여 CMOS 버퍼가 구동하는 단일 RLC 도선의 시간 지연을 나타내면

$$t_{pd} = t_{pdlin} + t_{pdsat} \exp(-1.1 \frac{t_{pdlin}}{t_{pdsat}}) \quad (8)$$

이다. 식 (8)의 t_{pd} 와 SPICE 시뮬레이션을 통한 연결선

시간 지연이 표 1에 있다. 다양한 값의 연결선 기생 임피던스와 부하 커패시턴스를 가지고 시뮬레이션을 해 본 결과 식 (8)이 높은 정확도를 가짐을 알 수 있다. 0.25μm CMOS 트랜지스터 모델을 사용하였으며, 현재 공정에서 대부분의 연결선은 0부터 1사이의 R_T , C_T 값을 가진다. 이 실험에서 연결선은 100개의 RLC Π 로 모형화하였다.

2. 단일 RLC 도선의 시간 지연 최소화

RC 연결선의 시간 지연은 연결선의 길이의 제곱에 비례한다. 연결선의 길이를 줄이기 위해 버퍼를 삽입하여 여러 개의 섹션으로 나누어 연결선의 시간 지연을 줄이

게 된다^{[8]-[10]}. 마찬가지로 RLC 연결선의 경우도 버퍼를 삽입함으로써 연결선 시간 지연을 줄일 수 있다. 그림 2는 버퍼를 삽입하여 k 개의 섹션으로 나누어진 일반적인 RLC 연결선을 보여준다. 모든 버퍼의 사이즈는 최소 크기 버퍼의 h 배다. R_0 는 최소 크기 버퍼의 출력 저항이고, R_{tr} 은 크기 h 인 버퍼의 출력 임피던스로서 값은 R_0/h 이다. C_0 은 최소 크기 버퍼의 입력 커패시턴스이고, C_L 은 크기 h 인 버퍼의 입력 커패시턴스로서 값은 hC_0 이다. 버퍼가 삽입된 연결선의 전체 시간 지연은 $t_{pdtotal}$ 이다. $t_{pdtotal}$ 이 최소일 때 h, k 의 값은 식 (9)와 같은 두 개의 편미분 방정식을 풀어 구한다.

$$\frac{\partial t_{pdtotal}(h, k)}{\partial h} = 0, \quad \frac{\partial t_{pdtotal}(h, k)}{\partial k} = 0 \quad (9)$$

RC 연결선에서의 h, k 의 값은

$$h(RC) = \sqrt{\frac{R_0 C_t}{R_t C_0}}, \quad k(RC) = \sqrt{\frac{R_t C_t}{2R_0 C_0}} \quad (10)$$

이다. 식 (10)은 Bakoglu^[13]가 제시한 수식과 같다. RLC 연결선에서 식 (9)에서 계산된 h, k 의 값을 곡선일치법을 사용하여 다시 표현하면

$$h(RLC) = \sqrt{\frac{R_0 C_t}{R_t C_0}} \frac{1}{[1 + 0.16(T_{L/R})^3]^{0.24}} \quad (11)$$

$$k(RLC) = \sqrt{\frac{R_t C_t}{2R_0 C_0}} \frac{1}{[1 + 0.18(T_{L/R})^3]^{0.3}} \quad (12)$$

이다.

$$T_{L/R} = \sqrt{\frac{L_t/R_t}{R_0 C_0}} \quad (13)$$

식 (11), 식 (12)에서 구해진 h, k 의 근사값과 식 (9)에서 구해진 원래의 h, k 값으로 각각 버퍼를 삽입하여 연결선의 시간 지연을 비교할 경우, 두 방법 사이의 차이는 0.05% 미만에 불과하다^[14].

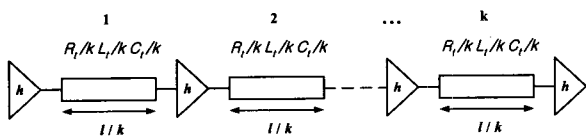


그림 2. 연결선 지연 최소화를 위해 버퍼가 삽입된 단일 RLC 도선

Fig. 2. Buffers inserted in an RLC line to minimize interconnect delay.

III. RLC 트리 연결선에서의 버퍼 삽입

트리 연결선은 단일 도선에 비해 복잡한 구조를 가지고 있다. 그러므로 단일 도선의 경우처럼 수식을 이용하여 간단하게 버퍼를 삽입하여 연결선 시간 지연을 줄이기는 어렵다. 트리 구조의 복잡성 때문에 연결선의 시간 지연을 최소로 하는 버퍼의 사이즈나 개수를 대수식 형태로 나타내기 힘들기 때문이다. 본 절에서는 RLC 트리 연결선의 시간 지연을 줄일 수 있는 5가지 알고리즘을 제시한다. 알고리즘 I은 Ismail^[15]에 의해 소개되었고, 본 논문에서는 이외에 일반적인 트리 탐색 방법을 고려한 4가지 알고리즘을 추가로 제시하고 성능 분석을 하고자 한다.

1. 알고리즘 I

그림 3은 n 개의 도선으로 구성된 일반적인 트리 연결선이고, 알고리즘을 사용할 때 버퍼가 삽입될 위치를 함께 보여준다. 세모로 그려진 부분이 버퍼가 삽입될 지점이다. 버퍼가 삽입이 되는 지점은 커패시턴스의 디커플링 효과를 최대로 하기 위해 도선의 시작 부분에 위치한다^[10]. 각 지점에 삽입이 가능한 버퍼의 종류는 B 개이다.

RLC 트리 연결선의 버퍼 삽입 문제는 연결선의 루트(root) 지점부터 측정하고자 하는 임의의 노드까지의 시간 지연을 최소화하기 위하여 m 개의 지점에서 버퍼들의 사이즈(h_j)를 결정하는 것이다($1 \leq j \leq m$). $h_j = 0$ 은 노드 j 에 버퍼가 삽입이 되지 않는 것을 의미한다. 버퍼 사이즈의 범위는 $1 \leq h_j \leq h_{max}$ 인 연속된 정수 배이다. h_{max} 는 삽입 가능한 버퍼의 최대 사이즈이다.

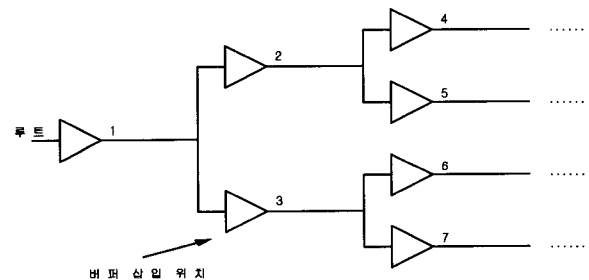


그림 3. 버퍼 삽입 위치를 나타내는 트리 연결선

Fig. 3. A tree interconnect representing buffer insertion positions.

그림 4에서 알고리즘 I을 도식화하였다. t 는 루트 노드부터 임의의 노드까지의 연결선 시간 지연이고, $t(h_j)$ 는 j 번째 버퍼 삽입 위치에 h_j 사이즈의 버퍼를 삽입했을 때

의 연결선 시간 지연이다. 알고리즘은 모든 버퍼 삽입 지점에 버퍼가 삽입이 되지 않은 초기 상태에서 시작한다. 버퍼 삽입 위치 1에서 사이즈가 다른 B 개의 버퍼를 삽입하여 연결선의 시간 지연을 구한다. 이 때 1번 지점을 제외한 다른 버퍼의 사이즈 h_2, \dots, h_m 은 0으로 한다. 결과를 비교하여 연결선 시간 지연을 최소로 하는 버퍼의 사이즈를 h_1 으로 정한다. 만약 $h_1 = 0$ 이면 버퍼를 삽입하지 않는다. 다음 과정은 1번 위치에 이전 단계에서 구한 사이즈가 h_1 인 버퍼를 달고 h_2, \dots, h_m 은 0으로 한다. 2번 위치에 마찬가지로 B 개의 버퍼를 삽입하여 연결선 시간 지연을 최소로 하는 버퍼의 사이즈 h_2 를 찾는다. 이와 같은 과정을 모든 m 개의 버퍼가 삽입될 지점에서 반복한다. 각 과정에서 버퍼의 사이즈를 찾을 때, 나머지 위치에 버퍼 사이즈는 지금까지 구한 값을 유지한다. 모든 m 개의 과정에서 수행이 끝났을 때 반복 회수(i)가 하나씩 증가한다. 첫 번째 반복이 끝나면 연결선의 시간 지연을 최소로 하는 버퍼 사이즈 h_1, h_2, \dots, h_m 을 찾기 위한 두 번째 반복을 한다. 두 번째 반복이 수행될 때 각 버퍼 삽입 지점의 버퍼의 사이즈는 첫 번째 반복이 끝났을 때까지 구한 버퍼의 사이즈다. 그리하여 두 번째 반복을 할 때 각 삽입 위치에서의 부하 캐패시턴스와 구동 저항이 첫 번째 반복일 때에 비해 연결선 시간 지연을 줄이므로, 보다 시간 지연이 적은 버퍼의 사이즈를 찾을 수 있다. 이러한 반복은 모든 위치의 버퍼 사이즈가 이전 반복 때의 결과와 다르지 않을 때까지 한다.

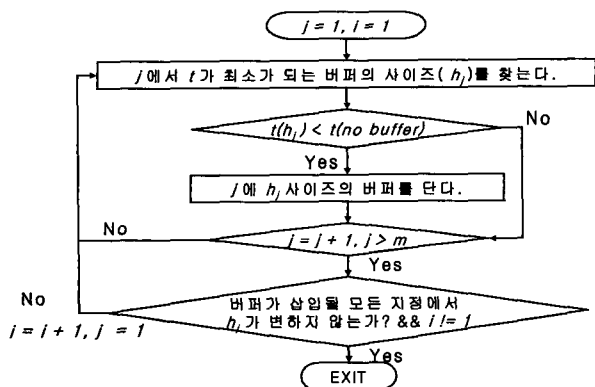


그림 4. 버퍼 삽입을 위한 알고리즘 I
Fig. 4. Algorithm I for buffer insertion.

알고리즘은 여러 번 반복한다. 알고리즘 1회 수행은 m 개의 버퍼 삽입 지점에서 구하고자 하는 트리 연결선의 시간 지연을 최소로 하는 버퍼의 사이즈를 찾는 것이다. 각 버퍼 삽입 위치에서는 삽입 가능한 버퍼의 개수가 B

이기 때문에 연결선 시간 지연을 B 번 구한다. 그리하여 1회 수행의 시간 복잡도는

$$\Theta(\text{반복}) = O(m \cdot B) \cdot O(\text{시간 지연 계산}) \quad (14)$$

이다. 알고리즘 I을 그림 3에 적용할 때 버퍼 삽입 순서는 1 - 2 - 3 - 4 - 5 - 6 - 7 이 된다.

2. 알고리즘 II

알고리즘 II는 그림 5와 같다. 알고리즘 I은 너비 우선 탐색(Breadth First Search) 방법을 이용한 버퍼 삽입 방법이다. 너비 우선 탐색 방법은 다음 레벨의 노드를 조사하기 앞서 트리의 같은 레벨이 있는 레벨의 노드를 먼저 조사하는 방법이다. 이것은 자식의 자식들이 고려되기 전에 현 노드의 바로 밑에 있는 자식 노드가 조사된다는 것을 의미한다.

알고리즘 II는 이에 반하여 역방향으로 버퍼를 삽입하는데, 우선 자식 노드가 없는 단말(terminal) 노드부터 조사한다. 다음으로 단말 노드의 부모 노드들을 조사하고, 그 다음 단말 노드의 부모의 부모 노드들을 조사한다. 이런 과정을 루트 노드까지 진행한다. 알고리즘 II를 그림 3에 적용할 때 버퍼 삽입 순서는 7 - 6 - 5 - 4 - 3 - 2 - 1 이 된다.

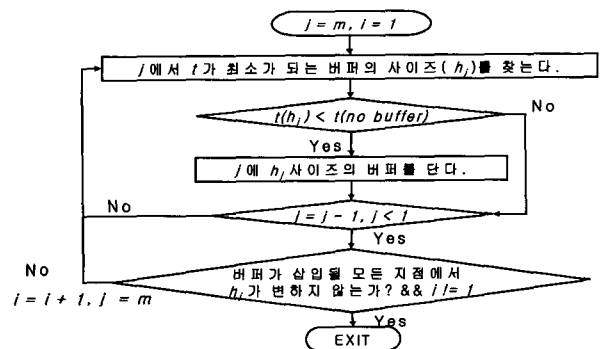


그림 5. 버퍼 삽입을 위한 알고리즘 II
Fig. 5. Algorithm II for buffer insertion.

3. 알고리즘 III ; Inorder Tree Insertion Algorithm

알고리즘 III은 일반적인 트리 탐색법 중에서 inorder 트리 탐색법을 이용한 버퍼 삽입 알고리즘이다. inorder 트리 탐색법은 왼쪽 서브 트리를 먼저 순회하고 루트 노드를 순회 그리고 오른쪽 서브 트리 순으로 순회하는 방식으로 트리의 최하위의 왼쪽 노드에서부터 시작된다. inorder 트리 삽입 알고리즘을 그림 3에 적용할 때 버퍼 삽입 순서는 4 - 2 - 5 - 1 - 6 - 3 - 7 이 된다.

4. 알고리즘 IV ; Preorder Tree Insertion Algorithm

알고리즘 IV은 일반적인 트리 탐색법 중에서 preorder 트리 탐색법을 이용한 버퍼 삽입 알고리즘이다. preorder 트리 탐색법은 루트 노드를 먼저 탐색하고 왼쪽 서브 트리를 탐색 그리고, 오른쪽 서브 트리를 탐색하는 방식으로 루트 노드가 가장 먼저 탐색된다. preorder 트리 삽입 알고리즘을 그림 3에 적용할 때 버퍼 삽입 순서는 1 - 2 - 4 - 5 - 3 - 6 - 7 이 된다.

5. 알고리즘 V ; Postorder Tree Insertion Algorithm

알고리즘 V은 일반적인 트리 탐색법 중에서 postorder 트리 탐색법을 이용한 버퍼 삽입 알고리즘이다. post-order 트리 탐색법은 왼쪽 서브 트리를 먼저 순회하고 다음 오른쪽 서브 트리를 그리고, 루트 노드를 순회하는 방식으로 트리의 최하위 왼쪽 노드에서 시작된다. post-order 트리 삽입 알고리즘을 그림 3에 적용할 때 버퍼 삽입 순서는 4 - 5 - 2 - 6 - 7 - 3 - 1 이 된다.

IV. 시뮬레이션 결과

III절에서 소개한 5가지 버퍼 삽입 알고리즘을 그림 6과 같은 레벨 3의 이진 RLC 트리 연결선에 대해서 실험해 보았다. 이진 트리는 분지수가 2인 트리인데, 모든 일반적 트리 구조는 임피던스가 0인 도선을 삽입하여 이진 트리로 만들 수 있고, 반대로 이진 트리 구조에 임피던스 0인 도선 삽입으로 삼진 이상의 트리 구조를 만들 수 있다. 따라서 본 논문의 이진 트리 연결선에 대한 실험은 모든 트리 연결선에 대해 일반성을 가지고 있다. 시뮬레이션에서 사용한 HSPICE 모형 파라미터는 TSMC사 0.25 μm 공정의 level 49 MOS모형이다.

회로의 입력 지점에서 노드 7번까지의 연결선 시간 지연 및 필요한 버퍼의 사이즈, 반복 회수 결과를 알아본다. 5가지 방법 모두 반복법이기 때문에 어떤 방법이 효율적인가를 결정할 때 반복 회수는 중요한 요소로 고려된다. 버퍼의 사이즈는 NMOS 트랜지스터의 폭이고, PMOS 트랜지스터의 폭은 NMOS의 3.5배이다. NMOS와 PMOS의 길이는 0.24 μm 이다. X 표시는 해당 번호 위치에 버퍼가 삽입되지 않음을 의미한다.

0.25 μm CMOS 공정에서 실험을 통해 추출된 연결선 파라미터 값(R, L, C)이 [4]에 제시되어 있다. 도선의 폭이 0.9 μm 일 때, $R = 494(\Omega/\text{cm})$, $L = 4.75(\text{nH}/\text{cm})$, $C = 1.73(\text{pF}/\text{cm})$ 이다. 폭이 51.9 μm 인 경우 $R = 8(\Omega/\text{cm})$, $L =$

1.60(nH/cm), $C = 27.3(\text{pF}/\text{cm})$ 이다.

단일 도선의 총 $R = 494\Omega$, $L = 4.75\text{nH}$, $C = 1.73\text{pF}$ 값을 가질 때 시뮬레이션 결과는 표 2에 있다. 버퍼 삽입 알고리즘을 사용하기 전 연결선의 입력 단부터 노드 7번까지의 시간 지연은 6.704876ns이다.

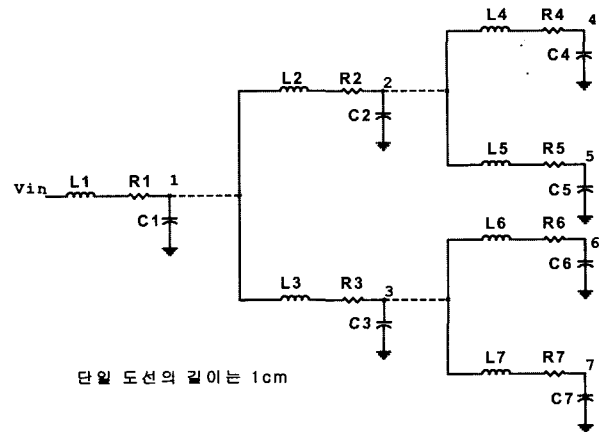


그림 6. 일반적인 RLC 트리 연결선의 예
Fig. 6. An example of general RLC tree interconnects.

표 3은 단일 도선의 총 $R = 8\Omega$, $L = 1.60\text{nH}$, $C = 27.3\text{pF}$ 값을 가질 때 시뮬레이션 결과이다. 버퍼 삽입 알고리즘을 사용하기 전 연결선의 입력 단부터 노드 7번까지의 시간 지연은 1.777991ns이다.

5가지 알고리즘 모두 버퍼의 삽입 위치가 동일하므로 연결선 시간 지연은 같다. 그러나 버퍼의 사이즈와 반복 회수 면에서는 약간의 차이를 보인다. 연결선 시간 지연은 저항과 커패시턴스의 곱에 비례한다. 그리하여 첫 번째 실험에서와 같이 연결선의 저항 성분이 큰 경우 버퍼를 삽입함으로써 많은 시간 지연 효과를 볼 수 있다. 첫 번째 실험의 경우 알고리즘을 적용하기 전 시간 지연의 45.6%(3.058251/6.704876)에 불과하다. 이에 반해 연결선의 저항 성분이 상대적으로 작은 두 번째 실험의 경우는 57.8%(1.028263/1.777991)이다. 연결선의 저항 성분이 작으면 상대적으로 인덕턴스의 영향이 커지기 때문에 시간 지연이 저항 성분이 클 때에 비해 많이 줄어들지는 않는다. 그리하여 필요한 버퍼의 개수도 줄어들고 또한 반복 회수도 줄어든다. 0.25 μm 공정에서 추출된 다양한 연결선 기생 임피던스를 가지고 트리 연결선의 레벨을 높여 실험을 해 본 결과 연결선의 인덕턴스 성분이 큰 경우 상대적으로 시간 지연이 적게 줄어드는 반면 반복 회수는 적었다.

버퍼의 사이즈는 부하 단의 임피던스에 의해 결정된다. 그리하여 반복 회수는 루트에서부터 삽입을 하는 알고리즘 I 보다는 서브 트리의 삽입 위치부터 버퍼를 삽

입하는 방법이 반복 수행 회수가 적었다. 따라서 트리의 단말 노드부터 역방향으로 진행되는 너비 우선 탐색하는

방법을 응용한 알고리즘Ⅱ와 postorder 삽입 방법(알고리즘Ⅴ)이 반복 회수가 적다.

표 2. 실험 결과(R = 494Ω, L = 4.75nH, C = 1.73pF)

Table 2. Experimental results(R = 494Ω, L = 4.75nH, C = 1.73pF)

버퍼 위치	버퍼 사이즈(NMOS 폭)				
	알고리즘 I	알고리즘Ⅱ	알고리즘Ⅲ	알고리즘Ⅳ	알고리즘Ⅴ
1	X	X	X	X	X
2	0.36μm	0.36μm	0.36μm	0.36μm	0.36μm
3	5.64μm	5.64μm	5.64μm	5.64μm	5.64μm
4	3.6μm	4.32μm	3.6μm	3.6μm	3.96μm
5	4.32μm	3.24μm	4.32μm	4.32μm	0.36μm
6	0.36μm	0.36μm	0.36μm	0.36μm	0.36μm
7	5.64μm	5.64μm	5.64μm	5.64μm	5.64μm
시간 지연(ns)	3.058251	3.058251	3.058251	3.058251	3.058251
반복 회수	4회	3회	4회	4회	3회

표 3. 실험 결과(R = 8Ω, L = 1.60nH, C = 27.3pF)

Table 3. Experimental results(R = 8Ω, L = 1.60nH, C = 27.3pF)

버퍼 위치	버퍼 사이즈(NMOS 폭)				
	알고리즘 I	알고리즘Ⅱ	알고리즘Ⅲ	알고리즘Ⅳ	알고리즘Ⅴ
1	X	X	X	X	X
2	0.36μm	0.36μm	0.36μm	0.36μm	0.36μm
3	X	X	X	X	X
4	X	0.36μm	0.36μm	X	0.36μm
5	X	0.36μm	X	X	0.36μm
6	0.36μm	0.36μm	0.36μm	0.36μm	0.36μm
7	X	X	X	X	X
시간 지연(ns)	1.028263	1.028263	1.028263	1.028263	1.028263
반복 회수	2회	2회	2회	2회	2회

V. 결 론

반도체 공정 기술이 발달됨에 따라 회로 내 전체 시간 지연에서 연결선 지연이 차지하는 비중이 커지고 있다. 그에 따라 연결선 지연을 예측할 수 있는 효율적인 모형화 방법이 필요한데, 인덕턴스의 영향을 무시할 수 없으므로 연결선은 RLC-class로 모형화하게 된다.

본 논문에서는 단일 도선 및 트리 구조 RLC 연결선의 시간 지연을 줄이기 위한 버퍼 삽입 방법을 제시하며, 단

일 RLC 도선의 경우 대수식을 이용하여 수월하게 버퍼를 삽입하여 시간 지연을 줄일 수가 있었다. 그러나 RLC 트리 연결선일 경우 구조가 복잡하여 단순한 수식을 사용한 방법을 찾기가 어렵기 때문에 RLC 트리 연결선의 시간 지연을 줄이기 위한 축차적 알고리즘 5개를 제시하였다.

기존에 제안된 알고리즘 I 과 본 논문에서 제시한 4가지 알고리즘을 사용하여 결과를 비교하였다. 시뮬레이션 결과를 볼 때, 5가지 알고리즘 모두 버퍼 삽입 위치가 동일하므로 시간 지연은 같다. 그렇지만 반복 회수를 조사

해보면 기존의 알고리즘 I 보다 bottom-up 순서로 버퍼를 삽입하는 알고리즘 II 또는 V가 적다. 따라서 본 논문에서 제시한 알고리즘을 사용하여 RLC 트리 연결선의 버퍼를 삽입할 때 동일한 시간 지연을 보장하면서 보다 빨리 버퍼의 삽입 여부 및 크기를 결정할 수 있다.

참 고 문 헌

- [1] J. M. Rabaey, *Digital Integrated Circuits, A Design Perspective*. Englewood Cliffs, NJ : Prentice-Hall, 1996.
- [2] S. Bothra, B. Rogers, M. Kellam, and C. M. Osburn, "Analysis of the effects of scaling on interconnect delay in ulsi circuits," *IEEE Trans. Electron Devices*, vol. 40, pp. 591-597, Mar. 1993.
- [3] T. Sakurai, "Approximation of wiring delay in MOSFET LSI," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 418-426, Aug. 1983.
- [4] A. Deutch *et al.*, "When are transmission-line effects important for on-chip interconnections?," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1836-1846, Oct. 1997.
- [5] Y. Massoud, S. Magors, T. Bustami, and J. White, "Layout techniques for minimizing on-chip interconnect self inductance," in *Proc. IEEE/ACM Design Automat. Conf.*, June 1998, pp. 566-571.
- [6] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of merit to characterize the importance of on-chip inductance," in *Proc. IEEE/ACM Design Automat. Conf.*, June 1998, pp. 560-565.
- [7] L. T. Pillage, "Coping with RC(L) interconnect design headaches," in *Proc. IEEE/ACE Int. Conf. Comput.-Aided Design*, Sept. 1995, pp. 246-253.
- [8] H. B. Bakoglu and J. D. Meindl, "Optimal interconnection circuits for VLSI," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 903-909, May. 1985.
- [9] V. Adler and E. G. Friedman, "Repeater design to reduce delay and power in resistive interconnect," in *Proc. IEEE Int. Symp. Circuits Systems*, June 1997, pp. 2148-2151.
- [10] C. J. Alpert and A. Devgan, "Wire segmenting for improved buffer insertion," in *Proc. IEEE/ACM Design Automat. Conf.*, June 1997, pp. 649-654.
- [11] T. Sakurai, A. R. Newton, "A Simple MOSFET Model for Circuit Analysis," *IEEE Transactions on Electron Devices*, vol. 38, No. 4, pp. 887-894, Apr. 1991.
- [12] Y. I. Ismail and E. G. Friedman, "Optimum repeater insertion based on a CMOS Delay model for on-chip RLC interconnect," in *Proc. IEEE ASIC Conf.*, Sept. 1998, pp. 369-373.
- [13] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI* Addison-Wesley Publishing Company, 1990.
- [14] Y. I. Ismail and E. G. Friedman, "Effects of inductance on the propagation delay and repeater insertion in VLSI circuits," in *Proc. ACM/IEEE Design Automat. Conf.*, June 1999, pp. 721-724.
- [15] Y. I. Ismail and E. G. Friedman, and J. L. Neves, "Repeater Insertion in Tree Structured Inductive Interconnect," *IEEE Trans. Circuits and Systems II*, vol. 48, NO. 5, May 2001.

저 자 소 개



김 보 겸(학생회원)
 2001년 숭실대학교 컴퓨터학과 졸업,
 2003년 숭실대학교 대학원 컴퓨터학
 과 석사과정
 <주관심분야: 설계 자동화, VLSI
 회로해석 및 설계>



김 승 용(학생회원)
 1994년 부산대학교 전자공학과 졸업,
 1999년 숭실대학교 정보과학대 학원
 전자계산기공학과 석사과정
 1994년~2000년 LG이노텍 연구소
 근무

2000년~현재 숭실대학교 컴퓨터 학과 박사과정.
 <주관심분야: 설계 자동화, VLSI 회로해석 및 설계>



김 석 윤(정회원)
 1980년 서울대 전기공학과 졸업,
 1990년 University of Texas at
 Austin 전기, 컴퓨터 학과 석사.
 1983년 University of Texas at
 Austin 전기, 컴퓨터학과 박사.

1982년~1987년 한국전자통신연구소 연구원.
 1993년~1995년 Motorola Inc., Senior Staff Engineer.
 1995년~현재 숭실대 컴퓨터학부 교수.
 <주관심분야: 설계자동화, VLSI 회로해석 및 설계, 통
 신시스템>

