

ED MOS 논리 LSI의 지연시간 모델링과 다지연 논리 시뮬레이터

(Delay Time Modeling for ED MOS Logic LSI and Multiple Delay Logic Simulator)

金 敬 昊*, 全 永 鉉*, 李 昌 雨*, 朴 松 培*

(Kyung Ho Kim, Young Hyun Jun, Chang Woo Lee and Song Bai Park)

要 約

본 논문은 ED MOS 논리게이트의 정확한 지연시간 모델링과 그 다지연 논리시뮬레이터의 응용을 다룬 것이다.

ED MOS 논리게이트에 대하여 제안된 지연 모델은 부하조건 뿐 아니라 입력파형의 기울기도 고려한 것이다. 지연시간을 물리적 기준레벨에 대하여 출력부하를 충전 / 방전 시키는데 걸리는 시간으로 정의하여 상승 / 하강 지연시간을 전압천이의 각 끝점에서 최적화된 가중 부하전류를 사용하여 explicit한 공식으로 유도하였다.

위에서 기술한 모델에 근거를 둔 다지연모델을 사용하는 논리 시뮬레이터를 개발하였고 새로운 모델과 타이밍검증 방법을 지연시간의 정확도와 실행시간에 대해서 평가하였다.

Abstract

This paper is concerned with an accurate delay time modeling of the ED MOS Logic gates and its application to the multiple delay logic simulator.

The proposed delay model of the ED MOS logic gate takes account of the effects of not only the loading conditions but also the slope of the input waveform. Defining delay as the time spent by the current imbalance of the active inverter to charge and discharge the output load, with respect to physical reference levels, rise and fall model delay times are obtained in an explicit formulation, using optimally weighted imbalance currents at the end points of the voltage transition.

A logic simulator which uses multiple rise/fall delays based on the model as described in the above has been developed.

The new delay model and timing verification method are evaluated with respect to delay accuracy and execution time.

I. 서 론

반도체 기술의 급격한 진보로 집적도가 증가함에 따

라 소프트웨어 시뮬레이션은 디자인 검증을 위한 필수적인 방법이 되었다. 이러한 시뮬레이터에는 회로시뮬레이터, 시간시뮬레이터, 논리시뮬레이터, 스위치레벨 시뮬레이터, 혼합시뮬레이터 등이 있으며 정확도와 효율간에 타협을 하게 된다.

회로시뮬레이터와 시간시뮬레이터는 정확한 해석을

*正會員, 韓國科學技術院 電氣 및 電子工學科

(Dept. of Elec. Eng., KAIST)

接受日字: 1987年 2月 25日

하지만 시간과 저장용량 때문에 회로의 크기가 수백 개에서 수천개의 게이트 이하로 제한된다.

논리시뮬레이터는 수 만개의 게이트로 이루어진 회로까지 해석이 가능하지만 MOS LSI 회로의 게이트 지연시간이 부하커패시턴스 뿐만 아니라, 입력과형에 의해서도 큰 영향을 받기 때문에 관련된 지연모델이 정확한 해석을 하는데는 불충분하다.

한편, 스위치레벨 시뮬레이터는 MOS의 다양한 구조를 서술하기에는 용이하지만, 지연시간을 도입하기가 어려우며, 혼합시뮬레이터는 각 모드드 간에 정보를 주고 받을 때 정확도가 상실 된다.

상승/하강 지연시간의 정확한 모델링이 VISI 디자인 검증에서는 중요하다.

스텝응답 방법에 근거를 둔 TAU모델^[1]은 인버터의 평균 상승/하강 시간을 이용한 것으로 입력과형을 고려하지 않고, 부하조건도 단순히 fan-out을 정수배 해 줌으로써 지연시간의 부정확한 예측을 했으며 입력과형과 부하조건을 고려하여 지연시간을 계산한 TOKUDA 모델^[2]은 충전/방전 전류를 각각 I_{pu} 와 I_{pd} 의 정수배로 단순화 함으로써 실제상황과 잘 부합되지 않았다.

한편 최근의 연구^[3]에서는 충전/방전 전류를 I_{pu} 와 I_{pd} 의 차에 해당하는 전류로 계산함으로써 지연시간 예측이 향상되었다.

디지털회로의 스위칭속도의 수식화는 시간에 의존하는 입력과형을 고려하면 해의 존재를 보장할 수 없는 implicit한 방정식이 된다.

본 논문에서는 지연시간을 물리적 기준레벨 사이에서 pull-up 전류와 pull-down 전류의 차에 해당하는 전류가 출력부하를 충전/방전하는데 걸리는 시간으로 정의하여, n채널 enhancement-depletion(ED) MOS 게이트의 지연시간에 대하여 fan-in과 fan-out의 영향을 포함한 explicit 한 공식을 얻는다.

또한 이 결과를 적용하여 다자연 논리시뮬레이터를 위한 지연시간 모델링을 제안한다.

디바이스 구조와 관련 지연시간의 실제적인 모델링을 통해 다자연 논리시뮬레이터는 회로시뮬레이터나 시간시뮬레이터에 해당하는 정확도를 얻을 수 있으며, 게이트 레벨에서 시뮬레이션을 수행하므로 실행속도는 논리 시뮬레이터와 동일한 order이다.

II. 지연 시간의 모델링

1. 지연 시간의 정의

그림 1의 정적(static) 전압전달곡선으로부터 입력전압이 V_{il} 또는 V_{ih} 에 도달하기까지는 출력에 상당한 변화가 없다는 것을 알 수 있다. 그래서 이런 점들을 지

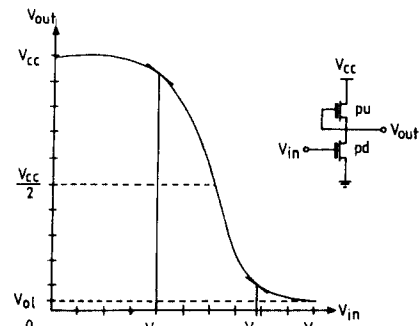


그림 1. Depletion-load 인버터의 정적 전압전달곡선. V_{il} 과 V_{ih} 는 소신호이득이 1이 되는 점들로 출력은 V_{il} - V_{ih} 구간에서만 변한다고 가정

Fig. 1. Static Transfer Curve of Depletion-Load Inverter. V_{il} and V_{ih} are Defined as the Input Levels at Which the Static Small-signal Gain of the Inverter is Unity.

연시간을 결정하기 위한 기준점으로 사용한다.

Dynamic상황에서 이상적인 경우 ($C_{out} = 0$)에 출력 특성은 pullup 트랜지스터와 pull-down 트랜지스터의 전류가 같다는 조건을 만족한다.

하지만 실제적인 경우 ($C_{out} \neq 0$) I_{pu} 와 I_{pd} 의 차에 해당하는 전하가 부하를 충전/방전시키기까지는 출력 변화가 일어나지 않는다. 따라서 static과 dynamic 출력응답 사이에 시간지연을 생기게 한다.

지연시간은 보통 static 인버터 문턱전압에 대하여 정의하거나, 10~20퍼센트 레벨과 80~90퍼센트 레벨 사이를 변화하는데 걸리는 시간으로 정의^[4]된다.

그러나 최근의 연구^[3,5]에서는 MOS구조에서의 지연시간이 입력에 의해 구동되는 레벨까지 출력 노우드를 평행상태에 이르게 하는 시간과 직접 관련되어 있으므로 좀 더 adaptive한 상황을 고려하여 다른 구조에서 계산되는 지연시간을 정의한다.

그림 2에서 나타난 것처럼 상승시간 T_{LH} 는 입력이 V_{ih} 를 통과하는 시간에서 출력이 $V_{cc}/2$ 까지 상승하는데 걸리는 시간이며, 하강시간 T_{HL} 은 입력이 V_{il} 을 통과해서 출력이 $v_{cc}/2$ 까지 하강하는데 걸리는 시간으로 정의한다. 이 정의는 V_{il} - V_{ih} 구간 밖의 전압에 대하여는 출력이 변하지 않는다고 가정했기 때문에 "static" 조건을 포함하며, 입력전압에 의해 출력전압이 $V_{cc} 2$ 까지 도달되는 것이 입력 기술기와 구조의 함수이므로 입력과형의 기술기를 고려한 것이다.

2. T_{LH} , T_{HL} 의 계산

T_{LH} , T_{HL} 은 부하커패시턴스가 평균 전류와 같은 일정

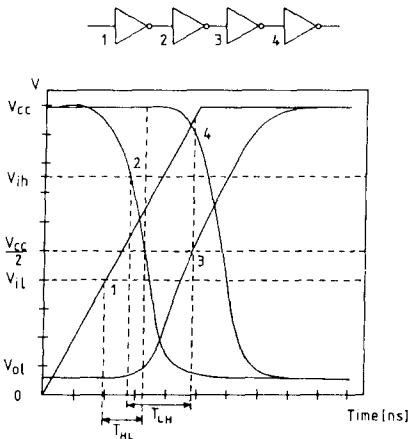


그림 2. 지연시간의 정의를 보여주는 인버터 체인의 Transient 응답
 Fig. 2. Transient Response of a Chain of Inverters with Indication of the Delay Definitions.

전류에 의해 충전 / 방전된다고 가정함으로써 식(1)에 의하여 계산될 수 있다.

$$T_{in,HL} = \frac{C_{out} \cdot \Delta V}{\langle I \rangle_{in,HL}} \quad (1)$$

여기서 C_{out} : 부하 커패시턴스
 ΔV : 출력전압 변화폭

그러나 정확한 평균전류의 계산은 해의 가능성이 없는 복잡한 수식화가 된다. 그러므로 근사적인 평균전류는 전압천이의 두 끝점에서 configuration비에 따라 weighting을 달리 준 pull-up 트랜지스터(TR_{pu})와 pull-down 트랜지스터(TR_{pd})의 전류차에 해당하는 전류를 계산함으로써 구한다.

(1) 상승지연시간 T_{LH} 의 계산

입력전압이 기울기 $1/\tau_{in}$ 을 가지고 V_{cc} 에서 V_{in} 까지 선형 감소한다고 가정한다.

$$V_{in} = V_{cc} (1 - t/\tau_{in}) \quad (2)$$

입력이 V_{cc} 에서 V_{in} 까지 떨어질 때는 그대로 유지된다는 것을 고려하면 상승지연시간은 아래의 관계식을 식(1)에 대입함으로써 구할 수 있다.

$$\langle I \rangle_{LH} = W_1 (I_{pu} - I_{pd})_{in} + W_2 (I_{pu} - I_{pd})_{HL} \quad (3)$$

여기서 $(I_{pu} - I_{pd})_{in}$: V_{in} 에서의 전류차

$$(I_{pu} - I_{pd})_{HL} : \frac{V_{cc}}{2} \text{에서의 전류차}$$

$V_{in} = V_{in}$ 에서 TR_{pu} 는 포화되며 TR_{pd} 는 선형영역에서 동작한다. $V_{out} = V_{cc}/2$ 에서 TR_{pu} 는 포화상태로 유지되나, 상승할 때와 하강할 때의 속도 차이때문에 TR_{pd} 는 출력이 $V_{cc}/2$ 를 통과할 때 차단영역에 있게 된다.

NMOS 트랜지스터에 대하여 식(1), (3)을 적용하면

$$T_{LH} = \frac{\tau_n \left(\frac{V_{cc}}{2} - V_{ol} \right)}{V_{ol} \left[\frac{W_1 \cdot V_{th}^2 (V_{ol}) + W_2 \cdot V_{th}^2 \left(\frac{V_{cc}}{2} \right)}{V_{th}^2 (V_{in})} - 2W_1 \left(\frac{2V_{ol}}{3(V_{cc} - V_{th})} \right)^2 \right]} \quad (4)$$

여기서

$$\frac{1}{R_{eff}} = \mu C_{ox} \left(\frac{W}{L} \right)_{pd} (V_{cc} - V_{th})$$

$$\tau_n = R_{eff} \cdot C_{out}$$

위의 결과식으로부터 상승지연시간이 τ_n 을 통해 출력 부하에 의존하고, 두 트랜지스터의 이용 가능한 전류에 의존함을 알 수 있다.

(2) 하강지연시간 T_{HL} 의 계산

입력레프 $V_{in} = V_{cc}/\tau_{in}$ 를 가정하고, 입력이 V_{in} 에 도달할 때까지는 출력이 V_{cc} 에서 유지된다는 것을 고려하면 하강지연 시간은 아래 식을 식(1)에 대입함으로써 구할 수 있다.

$$\langle I \rangle_{HL} = W_3 (I_{pd} - I_{pu})_{in} + W_4 (I_{pd} - I_{pu})_{HL} \quad (5)$$

여기서

$(I_{pd} - I_{pu})_{in}$: V_{in} 에서의 전류차

$(I_{pd} - I_{pu})_{HL}$: $\frac{V_{cc}}{2}$ 에서의 전류차

$V_{in} = V_{in}$ 에서 TR_{pu} 는 전류가 양이며, TR_{pd} 는 포화된다. $V_{out} = V_{cc}/2$ 에서 TR_{pu} 는 포화 영역에서 동작되며, TR_{pd} 는 선형영역에 있다. 하강지연시간의 V_{cc} 에서 $V_{cc}/2$ 까지 출력이 떨어지는 시간이므로 입력전압은 $V_{in} = V_{in} + V_{cc} \cdot \tau_{HL} / \tau_{in}$ 이다.

식(1), (5)를 적용하면

$$T_{HL} = \frac{\tau_n \cdot \tau_{in}}{\sqrt{2A} W_4} \cdot \frac{(W_3 B + W_4 C)}{2A W_4 (\beta)} \tau_{LH} \quad (6)$$

여기서

$$V_{th}^{*2} \triangleq \frac{1}{2} \cdot (V_{th}^2 (V_{ol}) + V_{th}^2 \left(\frac{V_{cc}}{2} \right))$$

$$A \triangleq \frac{V_{cc}}{2(V_{cc} - V_{th})}$$

$$B \triangleq \frac{V_{ol}}{V_{cc}} \left(\left(\frac{V_{th}^{*2}}{V_{th}^2 (\beta + 1)} - 1 \right) \right)$$

$$C \triangleq \frac{V_{th}^{*2}}{V_{th}^2} V_{ol} \sqrt{\frac{\beta}{\beta + 1}} - \frac{V_{cc}}{8(V_{cc} - V_{th})}$$

위의 결과식에서 하강지연시간은 τ_n 을 통해 출력 부하에 의존하고 τ_{in} 을 통해 fan-in에 영향을 받고 B, C를 통해 configuration비에 관련됨을 알 수 있다.

3. 가중계수의 최적화

상승지연시간 T_{LH} 와 하강지연시간 T_{HL} 을 구할 때 전압천이의 구간에서 흐르는 전류를 각 끝점에서 구하여 β 에 따른 가중계수(weighting factor)가 어떤 파라미터들의 어떤 함수 형식인가를 곱해줌으로써 보상할

수 있다고 가정 하였다. 가중계수의 의존성을 찾아내기 위하여 전압 천이의 두 끝점 중에서 하나에 0.5를 곱하고 다른쪽의 가중계수를 계산하는 실험적인 방법에 의해 가중계수를 다음과 같은 형식으로 가정하였다.

$$\begin{aligned}
 W_1 &= K_1(C) \cdot \beta^{-K_2}(C) \\
 W_2 &= K_3(C) \cdot \beta^{-K_4}(C) \\
 W_3 &= K_5(C) \cdot \beta^{-K_6}(C) \\
 W_4 &= K_7(C) \cdot \beta^{-K_8}(C)
 \end{aligned} \tag{7}$$

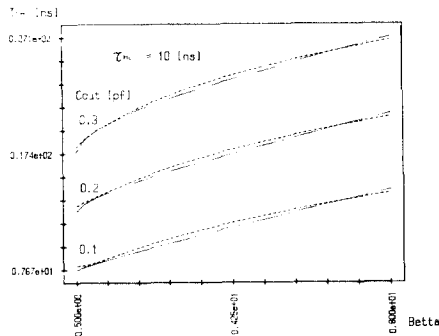
여기서 C는 C_{out} 이다.

Fletcher-Powell 최소화 방법을 사용하여 상승오차의 합

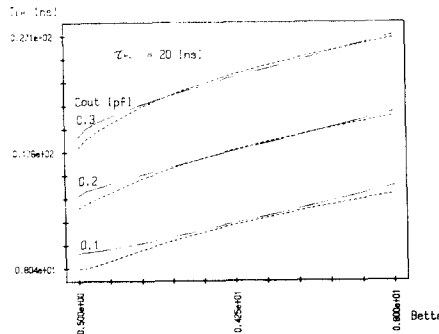
$$\sum_n \epsilon_n^2(\beta) \cong \sum_n (t_n - t_n^*)^2 / t_n^2$$

이 가장 적게 되도록 (K_1, K_2, K_3, K_4), (K_5, K_6, K_7, K_8)을 구하였다. 위에서 t_n, t_n^* 는 각각 SPICE 및 제안된 모델에서 계산된 지연시간이다.

그림 3은 상승지연시간을 최적화하여 출력부하에 관



(a)



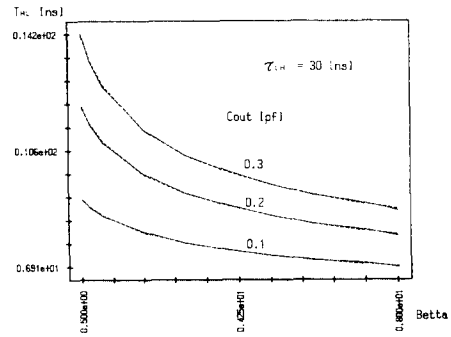
(b)

그림 3. 부하커패시턴스에 따른 상승지연의 최적화 (a) 가장 좋은 경우 (b) 가장 나쁜 경우

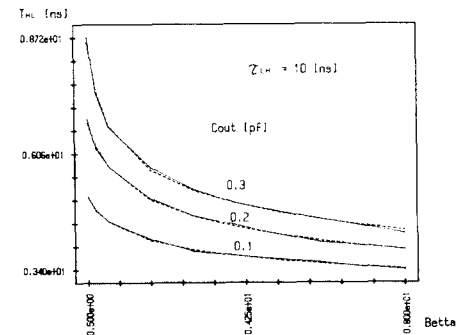
Fig. 3. Rising Delay Optimization for Several Load Capacitances.

(a) Best Case.
(b) Worst Case.

하여 가장 좋은 경우와 가장 나쁜 경우를 그린 것이며, 그림 4는 하강지연시간을 최적화하여 부하에 관해서 가장 좋은 경우와 가장 나쁜 경우를 보여준다. 상승지연의 경우 평균 2.5% 정도의 실제값과의 차이가 있었으며, 하강지연의 경우는 1.8% 이내에서 최적화 되었다.



(a)



(b)

그림 4. 부하 커패시턴스에 따른 하강 지연의 최적화 (a) 가장 좋은 경우 (b) 가장 나쁜 경우

Fig. 4. Falling Delay Optimization for Several Load Capacitances.

(a) Best Case.
(b) Worst Case.

4. 모델 지연시간과 SPICE시뮬레이션 값과의 비교 앞 절에서의 상승지연과 하강지연에 관한 모델을 검증하기 위하여 식(4), (6)에 의해 계산된 지연시간과 SPICE 시뮬레이션에서 나온 결과를 인버터체인(2단, 5단, 10단)과 링오실레이터에 대하여 실험하였다. K 모델은 본 논문에서 제안한 모델이며, F모델은 전압 천이의 두 끝점에서의 전류에 0.5를 곱하여 지연시간을 계산한 모델이다.³⁾

새로운 모델은 상승지연의 경우 SPICE와의 평균 차이가 3.8% 이내이고, 하강지연의 경우에도 5.6% 이내에서 제한되는 반면, F모델은 상승지연의 경우 21.9%였고, 하강지연의 경우 25.7%로서 K모델이 지연시간을 정확히 예측함을 알 수 있다. 최악의 경우에도 K모델은 T_{LH} 와 T_{HL} 의 경우 각각 6.1%, 10.1% 인데 반해, F모델은 39.3%, 44.9%의 오차가 발생했다. T_{HL} 의 경우 T_{LH} 보다 최적화 오차가 적었는데도 불구하고 SPICE와의 차이가 더 나는 것은 T_{LH} 가 출력 부하만의 함수인데 반해, T_{HL} 은 출력부하와 입력과형에 모두 의존하기 때문이다.

그림 5에 T_{LH} 와 T_{HL} 에 대하여 새로운 모델과 SPICE 시뮬레이션 결과를 동시에 그려서 비교 하였다.

III. 다지연 논리 시뮬레이터

논리 설계자가 회로를 설계한 후에, 그 회로가 정확히 동작하는지 각 소자의 지연시간에 의해 시간상의 문제로 인한 오동작의 가능성이 있는지 하드웨어로 제작하기 전에 확인하는데 사용되는 것이 논리시뮬레이터이며, 정확한 지연시간 모델을 사용하여 논리소자의 각 입력에서 출력으로의 변화를 구별하여 각 입력마다 다른 상승/하강 지연시간값을 가지고 논리적인 오퍼레이션을 수행하는 것이 다지연 논리시뮬레이터^{16,7)}이다.

1. 다지연 상승/하강 모델

다입력게이트는 각 입력노우드가 다른 커패시턴스를 가지므로 어떤 입력 노우드가 논리적으로 active 한가에 따라서 상승/하강 지연 시간값을 여러개 가지는 것이 타당하다. 그러므로 게이트모델은 각 입력 노우드에 따라 다지연 상승/하강 지연을 가진다.

게이트모델은 두가지의 기본적인 연산 즉 지연시간 연산, 함수 연산으로 계산된다. 두가지 연산의 배열순서에 따라 두개의 게이트모델 즉 입력측 지연모델, 출력측 지연모델이 가능하다.

입력측 지연모델

$$Z = F(D(X_1, X_2, \dots, X_n)) \\ = F(D_1(X_1), D_2(X_2), \dots, D_n(X_n))$$

출력측 지연모델

$$Z = D(F(X_1, X_2, \dots, X_n)) \\ \text{where } (X_1, X_2, \dots, X_n) : \text{입력}$$

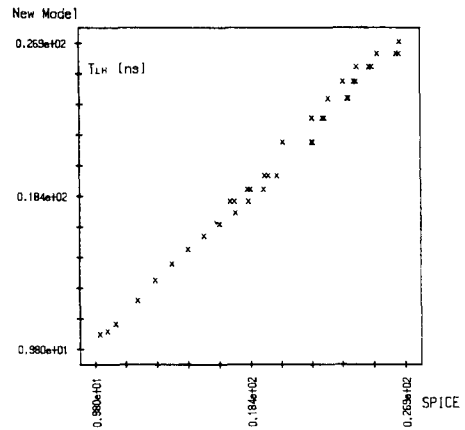
여기서

Z : 출력

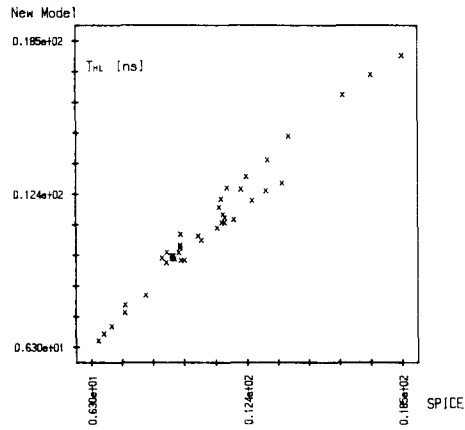
F : 함수

D : 지연 연산자

입력측 지연모델은 상세한 지연시간을 다루기 위해 복잡한 과정을 필요로 하며, 지나치게 까다로운 결과



(a)



(b)

그림 5. 새로운 모델과 SPICE의 지연시간 비교. 가장 이상적인 경우 기울기 1인 축상에 데이터가 모이게 되며, 위에서 (a)는 상승 지연의 경우이고 (b)는 하강 지연의 경우임

Fig. 5. Delay Comparison the New Model and SPICE. In the Ideal Case Data Will Lie Along the Unit Slope Axis.

(a) Rising Delay.

(b) Falling Delay.

를 보여주므로 출력 지연모델을 선택한다. 지연시간 연산시에 지연시간은 어떤 입력이 출력에 영향을 미쳤는가에 따라 택해지며 그 법칙은 다음중의 하나에 속한다.

- (1) 출력변화와 반대로 천이한 입력을 택한다.
- (2) 출력변화와 같은 천이를 일으킨 입력을 택한다.
- (3) 임의의 천이를 일으킨 입력을 택한다.
- (4) 제어입력(clock, reset등)을 택한다.

더우기 겹치는 입력이 출력천이를 일으킬 때에는 지

연시간은 표 1 과 같은 게이트 타입에 특별한 코우드에 따라 계산된다.

다음은 다지연 상승 / 하강 모델의 특징을 요약한 것이다.

- (1) 각 입력에서 출력에 영향을 줄 때 지정되는 상승 / 하강 시간을 저장하는 지연 매트릭스
- (2) 출력측 지연모델
- (3) 어떤 입력이 출력에 영향을 미치는가를 찾아내는 입력선택 과정
- (4) 겹치는 입력변화의 경우에 대한 지연 계산과정

표 1. 지연계산 코우드
Table 1. Delay Calculation Code.

	rising	falling
Buffer	-	-
AND	Max	Min
NAND	Min	Max
OR	Min	Max
NOR	Max	Min
NOT	-	-
Exclusive OR	Max	Max
Exclusive NOR	Max	Max

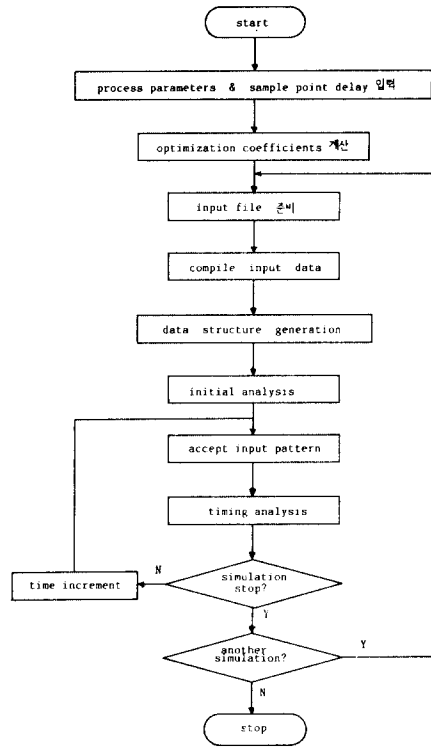


그림 6. KMULTIPLE의 전체 시뮬레이션 흐름도
Fig. 6. Global Flowchart of KMULTIPLE.

2. 시뮬레이션 흐름도

그림 6 은 본 연구에서 개발된 다지연 논리시뮬레이터 KMULTIPLE의 입력파일에서부터 각각의 process parameter 를 사용하여 부하전류의 가중계수를 계산하는 과정과 다지연 논리시뮬레이션을 하는 과정을 간략하게 나타낸 흐름도이며, 그림 7 은 다지연 논리시뮬레이터에서 입력과형과 부하조건에 따라 각 gate 의 delay 를 계산하는 과정 및 시간 scheduling 루틴을 나타내는 흐름도이다.

3. 시뮬레이션 예제

여기에서의 모든 시뮬레이션은 VAX 11/750 UNIX 4.2 BSD(no floating point accelerator) 컴퓨터를 사용한 결과이다. Adder, Binary-to-octal decoder, D-Flop, ALU에 대하여 시뮬레이션을 해보았으며, KMULTIPLE의 실행속도는 지연시간을 계산하는데 걸리는 시간과 입력을 구별하여 지연시간을 따로 지정하는데 걸리는 시간이 전체적인 시간과 비교하여 작기 때문에 보통의 논리시뮬레이터에 가까우며 그 누적 시간의 정확도는 표 2 에서 보여준 것처럼 13% 이내에서 제한됨으로써 믿음직한 결과를 얻었음을 알 수 있다.

그림 8 은 KMULTIPLE의 실행속도가 논리시뮬레이터에 가깝다는 것을 증명하기 위하여 CPU 시간을 그

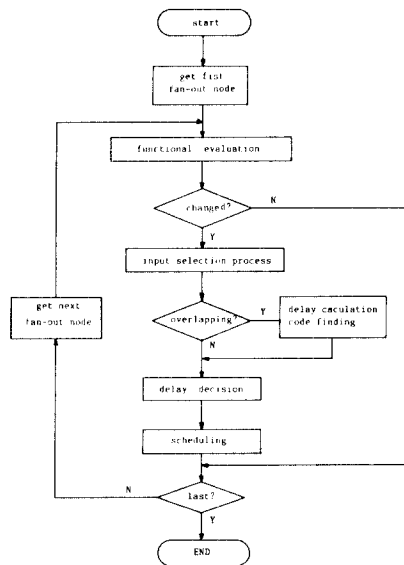


그림 7. EVALUATION 루틴의 흐름도
Fig. 7. Flowchart of EVALUATION Routine.

래프로 표시한 것으로 점선은 시뮬레이션은 실패했으나 extrapolation으로 근사시킨 것을 나타낸다.

표 2. 시뮬레이션 결과 요약
Table 2. Summary of Simulation Results.

Test Circuit	Simulation Time S		Accumulated Delay nS	
	SPICE	KMULTIPLE	SPICE	KMULTIPLE
Adder	324.82	4.47	67.8	60
Decoder	336.38	7.68	28.8	28
D-FE	167.05	4.50	19.5	22
A. L. U.	1693.52	20.97	11.9	13

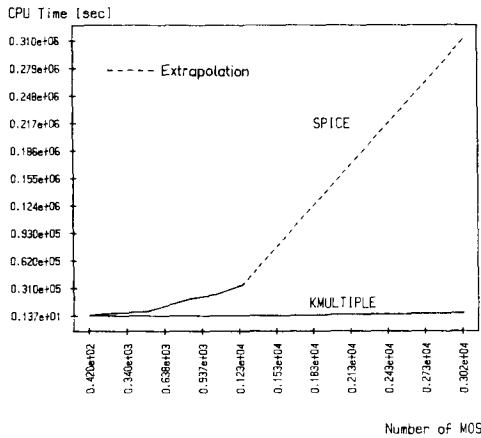


그림 8. SPICE와 KMULTIPLE의 시간비교
Fig. 8. CPU Time Comparison of SPICE and KMULTIPLE.

IV. 결 론

본 논문에서는 부하조건 ED MOS 논리게이트의 지연시간의 모델이 제안되었다. 상승지연시간은 게이트의 출력커패시터스의 함수로 근사되며, 하강지연시간은 출력부하와 전단계이트의 출력부하의 함수인 입력파형에도 의존한다. 정확한 지연시간의 계산은 논리소자의 출력 노우드에서 각 입력노우드의 지연소자를 고려함에 의해 얻어진다.

제안된 모델은 베타의 넓은 범위에 대하여 적용 가능하며, SPICE와 비교하여 정확도가 10% 이내에서 유지됨을 알 수 있었다.

Transmission gate의 경우에는 동기모우드와 비동기모우드로 나누어서 전단 게이트와 후단 게이트를 포함하여 하나의 element로 취급하는 방법을 취한다.

새로운 지연모델을 적용한 다자연 논리시뮬레이터는 VAX 11/750 상에서 표준 PASCAL을 사용하여 구현되었으며 여러개의 입력에서 출력으로의 지연이 정확히 시뮬레이션되어 LSI 설계 검증을 위한 수단을 제

공해 준다.

제안된 모델의 정확도가 SPICE와 비교하여 분석되었고, 다자연 논리시뮬레이터가 여러가지의 실제적 회로에 대하여 실험되고, 그 실행시간이 보통의 논리시뮬레이터에 가깝다는 것이 입증되었다.

자동적으로 critical path를 찾아 시뮬레이션 해주고 test pattern generation을 해주는 기능이 첨가된다면 더욱 강력한 시뮬레이션 tool이 될 것이다.

參 考 文 獻

- [1] C. Mead and L. Conway, *Introduction to VLSI Systems*. Reading, MA : Addison Wesley, 1980.
- [2] T. Tokuda, K. Okazaki, K. Sakashita, I. Ohkura and T. Enomoto, "Delay time modeling for ED MOS Logic LSI", *IEEE Trans. Computer-Aided Des.*, vol. CAD-2, pp. 129-134, July 1983.
- [3] D. Auvergne, G. Cambon, "Delay-time evaluation in ED MOS Logic LSI.", *IEEE J. Solid-state Circuits*, vol. sc-21, pp. 337-343, 1986.
- [4] L.A. Glasser and L.P. Hoyje, "Delay and power optimization in VLSI circuits", in *Proc ACM IEE 21th DAC*, pp. 529-535, June 1984.
- [5] D.A. Hodges and H.G. Jakson, *Analysis and Design of Digital Intergrated Circuits*, New York: Mcgraw-Hill, 1983.
- [6] H.N. Nham and A.K. Bose, *A multiple delay Simulator for MOS LSI Circuits*, Proc. of 17th DAC, pp. 279-285, June 1980.
- [7] Kaoru Okazaki, Tomoko Moriya, Toshiniko Yahara, "A multiple media delay simulator for MOS LSI circuits.", in *Proc. ACM IEEE 20th DAC*, pp. 279-285, 1983.
- [8] R.J. Bayruns, R.L. Johnston, D.L. Fraser and S.C. Fang, "Delay analysis of Si NMOS Gbit/s logic circuits.", *IEEE J. Solid-State Circuits*, vol. sc-19, pp. 755-764, 1984.
- [9] C.A. Mead and M. Rem, "Minimum propagation delays in VLSI. " *IEEE J. Soid-State Circuits*, vol. sc-17, pp. 773-775, 1982.
- [10] T.M. Lin and C.A. Mead, "Signal delay in general RC Networks," *IEEE Trans. Computer-Aided Des.*, vol. CAD-3, pp. 331-349, October 1984. *