

# CMOS 디지털 설계를 위한 트랜지스터 크기의 최적화기법 (New Transistor Sizing Algorithms For CMOS Digital Designs)

李尙憲\*, 金敬昊\*\*, 朴松培\*

(Sang Heon Lee, Kyung Ho Kim and Song Bai Park)

## 要約

컴퓨터를 이용하여 CMOS 디지털 회로의 동작속도와 칩 크기를 자동으로 최적화하는 방법으로는 보통 수학적방법 또는 경험적방법이 사용되어져 왔다. 본 논문에서는 앞의 두 가지 최적화방법을 결합하여 다중의 임계경로들에는 수학적방법을, 그 이외의 비임계경로들에는 경험적기법을 사용하는 새로운 최적화방법을 제안하였다. 전체 문제의 차원을 줄이기 위하여 기본 스테이지와 그의 병렬부분과 그들의 상보부분을 포함하는 확장 스테이지라는 기본단위를 사용하였다. 다중임계경로들의 비선형 최적화문제를 풀기 위하여는 augmented Lagrange multiplier 방법이 사용되었다. 비임계경로들에는 최적화된 임계경로들의 지연시간에 영향을 주지 않는 범위 내에서 출력으로부터 입력 쪽으로 점진적으로 진행하며 면적을 최소화하는 기법이 사용되었다. 제안된 기법들은 만족할 만한 결과를 보여주었다.

## Abstract

In the automatic transistor sizing with computer for optimizing delay and the chip area of CMOS digital circuits, conventionally either a mathematical method or a heuristic method has been used. In this paper, we present a new method of transistor sizing, a sort of combination of the above two methods, in which the mathematical method is used for sizing of critical paths and the heuristic method is used for desizing of non-critical paths. In order to reduce the overall problem dimension, a basic block called an extended stage is introduced which includes a basic stage, parallel transistors and complementary part. Optimization for multiple critical paths is formulated as a problem of area minimization subject to delay constraints and is solved by the augmented Lagrange multiplier method. The transistor sizes along non-critical paths are decreased successively without affecting the critical path delay times. The proposed scheme was successfully applied to several test circuits.

\* 正會員, 韓國科學技術院 電氣 및 電子工學科  
(Dept. of Electrical Eng., KAIST)

\*\* 正會員, 三星電子 半導體部門 研究所 CAE 팀

(CAE Team, Semiconductor Business R & D  
Center, Samsung Electronics Company)

接受日字: 1993年 2月 19日

### I. 서론

전자회로의 크기가 커짐에 따라 회로의 설계와 이의 성능을 향상 시키는데 있어서 소프트웨어는 보다 나은 결과를 얻기 위해 거의 필수적인 도구가 되고 있다. 고성능의 MOS 디지털 회로를 설계하는 데에는 빠른 동작속도와 적은 전력소모, 작은 칩 면적 사이의 절충이 필요하게 된다. 그러나 CMOS 회로에서 전력소모는 대개 칩의 면적에 비례하므로 동작속도와 칩 면적사이의 관계만을 고려하는 것이 보통이며, 회로의 지연시간과 칩의 면적 사이의 관계는 비선형적이므로 이들 사이의 절충은 간단히 해결 되지 않는다. 이를 해결하기 위해 연구된 방법들을 살펴보면 크게 네 가지로 나눌 수 있다. 첫째, 비선형 최적화 모듈과 회로 시뮬레이터를 사용하여 반복적으로 문제를 해결하는 방법으로<sup>1)</sup> 설계자인 사람이 회로 시뮬레이터와 경험에 의하여 수행하던 것을 소프트웨어가 대신하는 형태이다. 이 방법은 간단한 회로에 대하여 사람이 수행하는 것보다 정확하게 해결할 수 있으나 회로가 커짐에 따라 기하 급수적으로 시뮬레이션의 기억용량 요구량과 계산시간이 늘어나는 단점이 있

다. 두 번째는 경험적기법을 이용한 것으로<sup>2)</sup> 주어진 조건을 만족할 때까지 회로의 소자 크기를 점차 늘려가며 반복하는 방법이다. 이 방법은 빠르며 큰 회로를 다룰 수 있다는 장점이 있는 반면 최적해를 보장하지는 못한다는 단점이 있다. 세 번째는 회로전체를 단순화한 수식으로 바꾸어 수식에 대하여 비선형 최적화를 취하는 방법으로<sup>3)</sup> 변환된 수식에 대한 최적해를 얻을 수 있고 첫째 방법보다는 큰 회로를 다룰 수 있으나 매우 큰 회로를 다루기 어려운 문제점이 있다. 네 번째는 시뮬레이티드 어닐링을 이용한 방법으로<sup>4)</sup> 소자의 크기를 제조공정에서 조절가능한 최소한의 양 단위로 규격화하여 조합형 최적화문제로 바꾸면 최적해를 구할 수는 있지만 시간이 너무 많이 걸리는 단점이 있다.

본 논문에서는 두 번째의 경험적기법과 세 번째의 수학적기법의 장점을 취해 다중임계경로에는 수식적 방법을 적용하여 지연시간을 최적화하고 임계경로 이외 부분에는 경험적방법을 적용하여 면적을 최적화하는 새로운 기법을 보였다. 그림 (1)은 전체적인 시스템의 구조이다.

### II. 지연시간 및 면적 모델

임계경로의 최적화문제에는 정확하고 특성이 좋은 지연시간 모델이 필요하다. 모델의 정확도가 떨어지면 최적화결과도 정확하지 못하여 최적화의 의미를 감소시키며, 수학적특성이 좋지 못하면 사용되는 알고리즘에 따라 좋지않은 결과를 낳게 된다. 지연시간 최적화시 수학적특성이 좋은 모델은 연속이고 미분가능하며, 문제를 구성했을 때 convex한 특성을 갖는다. Lumped RC 지연 모델과 분포 RC 지연 모델<sup>5)</sup>이 최적화문제에서 convex한 목적함수의 모양을 갖는다는 것은 잘 알려진 사실이다.<sup>1)</sup><sup>6)</sup> 본 논문에서는 지연시간 모델로 분포 RC 모델을 사용하였다. 또한, 전체 문제의 차원을 줄이기 위하여 일반적인 CMOS 조합논리회로와 그 출력에 연결된 트랜스미션 게이트들을 확장 스테이지<sup>7)</sup>로 정의하여 최적화의 기본단위로 사용하였는데, 그림 (2)는 그 한 예를 나타낸다. 최적화과정 중에는 각 확장 스테이지에 대해 그 내부 소자들의 크기와 비례하는 하나의 지수 S를 설정하여 이 값의 증감으로 소자의 크기를 조절한다.

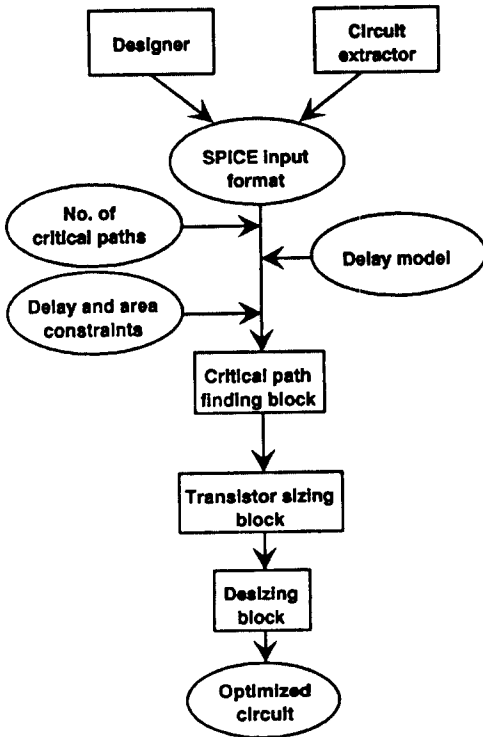


그림 1. 전체적인 시스템의 구성  
Fig. 1. Overall system configuration.

1) 본 논문에서 다루는 지연시간 모델에는 특정 MOS 스위치가 turn on 될 때 지연시간이 가장 큰 경로를 이용한다.  
2) 확장 스테이지라는 용어는 참고자료 [7]에 나오는 부하전하의 이동경로인 "스테이지"를 확장한데서 유래한다.

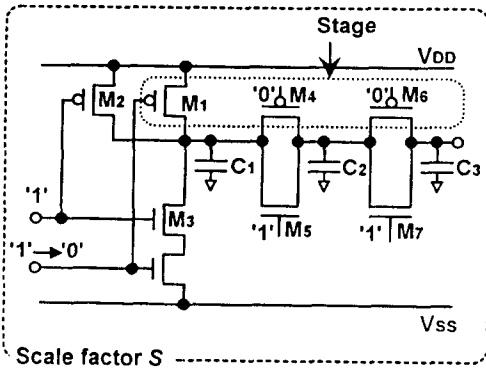


그림 2. 확장 스테이지의 예  
Fig. 2. An example of extended stage.

확장 스테이지에 대한 지연시간 모델은 일반적인 분포 RC 지연시간 모델을 사용했다. 식(1)은 그림(2)에 있는 확장 스테이지에 대한 지연시간 계산 예이며, 식(2)는 확장 스테이지  $j$ 에 대한 일반적인 지연시간 계산식이다.

$$T = \left\{ \alpha_c \left[ \alpha_c (W_m + W_m) + \alpha_c W_m \right] C_1 \right\} \frac{\alpha_c}{S} \frac{1}{W_m} \quad (1)$$

$$+ \left\{ S \left[ \alpha_c (W_m + W_m) + \alpha_c (W_m + W_m) \right] C_2 \right\} \frac{1}{S} \left( \frac{1}{\alpha_c W_m} + \frac{1}{\alpha_c W_m + \alpha_c W_m} \right)$$

$$+ \left\{ S \left( \alpha_c W_m + \alpha_c W_m \right) + C_3 \right\} \frac{1}{S} \left( \frac{1}{\alpha_c W_m} + \frac{1}{\alpha_c W_m + \alpha_c W_m} + \frac{1}{\alpha_c W_m + \alpha_c W_m} \right)$$

$$T_j = \sum_i \left\{ \left( \sum_{(cc_i)} \alpha_c S W_i + C_i \right) \left( \sum_{(cc_i)} \frac{1}{\alpha_c S W_i} \right) \right\} \quad (2)$$

식(1)과 식(2)에서 ‘·’의 왼쪽은 각 노드에 대한 커패시터 값이고 오른쪽은 VDD에서 각 노드까지의 저항값에 해당한다

- 식(1)과 식(2)에서
- $S$ 는  $j$ 번째 확장 스테이지의 스케일 지수.
- $W_i$ 는  $i$ 번째 트랜지스터의 채널 폭.
- $C_k$ 는  $k$ 번째 노드에서 부하와 배선의 용량.
- $\alpha_c$ 는 커패시터 상수.
- $\alpha_r$ 는 저항상수.

$R_{ko}$ 는 노드  $j$ 에서 경로  $(i, k)$ 와  $(i, o)$ 가 만나는 노드의 사이에 있는 트랜지스터들을 나타낸다.

칩의 크기는 소자의 배치와 크기 그리고 소자간의 배선에 영향을 받으므로 정확한 면적의 모델은 이들 모두를 고려해야 한다. 그러나 배치와 배선을 정확하게 고려하는 것은 소자 크기에 최적한 배치 배선 방법이 완전히 정립되지 않았고, 배치 배선에 많은 시간이 필요하므로

칩의 면적은 대략적으로 구하는 것이 보통이다. 이에 가장 흔히 사용되는 방법중의 하나로, 칩 내부 모든 트랜지스터 채널영역의 합이 칩 면적과 어느 정도 비례 관계를 갖는 경향을 이용하여 채널영역만의 합을 칩 면적에 대한 지표로 삼는 방법이 있다.<sup>[6]</sup> 본 논문에서는 다음과 같이  $j$ 번째 확장 스테이지의 면적을 계산한다.

$$A_j = \sum_i S_i (W_i L_i) \quad (3)$$

여기에서

$S$ 는  $j$ 번째 확장 스테이지의 스케일 지수.

$W_i, L_i$ 는  $i$ 번째 트랜지스터의 채널 폭과 길이이다.

### III. 최적화 알고리즘

최적화를 수행해야 할 회로에는 그 회로가 충족해야 하는 지연시간보다 큰 지연시간을 갖는 임계경로가 여러 개 있을 수 있고, 임계경로들에 대한 최적화의 결과로 임계경로가 아니던 것들이 새로운 임계경로로 등장하는 수가 있으므로 이를 효율적으로 다루기 위해서 여러 개의 임계경로를 동시에 다루도록 하여 최적화한다. 여러 임계경로를 최적화하는 문제는 지연시간에 제한을 두고 회로의 면적을 최소화하는 비선형문제로 바뀌며 이를 일반적인 최적화 소프트웨어에 의해 해결한다. 비임계경로에 있는 트랜지스터들은 임계경로의 최적화가 끝난 뒤 그들의 값에 영향을 주지 않는 범위 내에서 출력 쪽으로부터 입력 쪽으로 진행하며 순차적인 면적최소화를 진행한다.

#### 1. 임계경로의 최적화

최적화부분은 임계경로를 찾는 부분에서 찾아준 특정값 이상의 지연시간을 갖는 경로를 모두 받아 지연시간과 면적에 관한 모양으로 변환한 뒤 소자의 크기와 지연시간에 대한 모든 제한 조건을 설정한다. 이때 설계자는 입·출력 노드의 결정과 최적화하려는 경로의 결정에 관여할 수 있고 각 경로마다 다른 지연시간을 설정할 수도 있다. 제한조건과 목적함수가 완성되면 이들은 비선형문제해결 소프트웨어에 의해 최적화가 진행되는데 이 때 선택한 모든 임계경로가 동시에 최적화된다. 한편, 임계경로에 포함되지 않은 경로가 최적화가 끝난 후 목적인 지연시간보다 큰 값을 갖는 경우가 있으나 거의 모든 경우 그 증가분은 지연시간에 비해 매우 작은 값을 갖는다.<sup>[7]</sup> 이 경로들

3) 최적화는 전체적으로 지연시간을 줄이는 방향으로 작용한다. 초기설계가 어느정도 된 회로의 경우 최적화 후 설정지연시간보다 큰 지연 시간을 갖는 비임계경로가 발생했다면 그 경로의 초기지연시간은 설정지연시간에 매우 근접한 값을 갖는다.

은 다음에 나올 면적최소화 부분에서 해결한다.  
이 최적화문제는 다음과 같은 형태로 기술된다.

Minimize Total area  
Subject to Delay constraints  
Area constraints  
Transistor size constraints

이를 수식적으로 기술하면 다음과 같이 표현된다.

$$\begin{aligned} & \text{Minimize} \quad f(z) \\ & \text{Subject to} \quad g_j(z) \leq 0, \text{ where } j = 1, \dots, m \\ & \quad \quad \quad z \in R^n \end{aligned} \quad (4)$$

식(4)에서  $z$ 는 변수들로 트랜지스터의 크기, 최대 지연시간 등이며,  $f(z)$ 는 목적함수로 전체면적이 되고,  $g_j(z)$ 는 지연시간, 면적, 트랜지스터 크기 등 만족해야할 조건이다. 본 논문에서는 식(4)를 augmented Lagrangian algorithm<sup>[8]</sup>으로 해를 찾는다. 다음은 augmented Lagrangian 함수와 최적화를 위한 algorithm에 대한 설명이다.

Augmented Lagrangian 함수는 기본적으로 Lagrangian 함수에 penalty 항을 추가한 형태를 갖는다. Lagrangian 함수는

$$L(z, \lambda) = f(z) + \lambda^T g(z) \quad (5)$$

와 같이 정의 되는데  $f(z)$ ,  $g_j(z)$ 는 식(4)와 의미가 같으며  $\lambda$ 는 쌍대(dual) 변수로  $g_j(z)$ 에 대한 여분치(marginal cost)를 나타낸다. 모든  $g_j(z)$ 가 등식 제약조건(equality constraint)을 갖는다고 하면 식(4)는 다음 식의 해를 찾으면 구할 수 있다.

$$\max_{\lambda} [\min_z L(z, \lambda)] \quad (6)$$

식(6)은 제약조건이 없는 최적화문제이긴 하지만 최대치와 최소치를 같이 구해야하는 문제이므로 복잡하다. 따라서 식(6)을 쌍대 함수  $\phi(\lambda) = \max_z L(z, \lambda)$ 를 이용하여 쌍대변수만의 함수로 바꾸면 다음과 같이 된다.

$$\max_{\lambda} [\phi(\lambda)] \quad (7)$$

Augmented Lagrangian 함수는 식(5)의 Lagrangian 함수에 penalty 항을 추가하는데 2차 penalty 항을 고려하면

$$L(z, \lambda) = f(z) + \lambda^T g(z) + \frac{1}{2} c g(z)^T g(z) \quad (8)$$

와 같은 모양을 갖으며  $c$ 는  $L(z, \lambda)$ 에 대한 penalty 항의 기여정도를 나타내는 변수이다. 식(8)을 식(6)과 같은 모양으로 바꾸면 다음의 식이 된다.

$$\max_{\lambda} [\min_z L_A(z, \lambda)] \quad (9)$$

식(9)는 식(6)보다 penalty 항 때문에 빠른 수렴 특성을 갖는다.

$L_A(z, \lambda)$ 가 최소가 되기 위한 필요조건은

$$\nabla_z L_A(z, \lambda) = \nabla_z f(z) + \lambda \nabla_z g(z) + c g(z) = 0 \quad (10)$$

과 같이 되고 쌍대함수는  $\phi(\lambda) = \min_z L_A(z, \lambda)$ 가 되어 기술기는 다음 식과 같이 된다

$$\begin{aligned} \nabla_{\lambda} \phi(\lambda) &= \nabla_z L_A(z, \lambda) \nabla_z z(\lambda) + g(z(\lambda)) \\ &= g(z(\lambda)) \end{aligned} \quad (11)$$

식(6)에서 식(11)까지는 등식 제약조건을 가정했을 경우인데 실제의 경우 거의 모든 문제는 부등식 제약조건이 들어가므로 이를 고려하기 위해 부가변수(slack variable)  $v_j$ 를 사용하면 부등식 제약조건  $g_j(z)$ 는 다음과 같이 바꿀 수 있다.

$$g_{j(z)} \leq 0 \equiv \hat{g}_j(z, v_j) = g_j(z) v_j^2 = 0 \quad (12)$$

식(12)는 부가 변수가 문제의 차원을 높이는데 이는 수식적인 방법으로 차원을 낮출 수 있다.<sup>[10]</sup> 다음 식은 그 결과의  $\circ(\lambda)$ 와  $\nabla_{\lambda} \circ(\lambda)$ 이다.

$$\phi(\lambda) \min \left[ f(z) + \sum_j \begin{cases} \lambda_j g_j(z) + \frac{1}{2} c g_j(z)^2, & \text{if } g_j(z) \geq -\frac{\lambda_j}{c} \\ \frac{1}{2c} \lambda_j^2, & \text{if } g_j(z) < -\frac{\lambda_j}{c} \end{cases} \right] \quad (13)$$

$$\nabla_{\lambda} \phi(\lambda) = \sum_j \begin{cases} g_j(z), & \text{if } g_j(z) \geq -\frac{\lambda_j}{c} \\ -\frac{\lambda_j}{c}, & \text{if } g_j(z) < -\frac{\lambda_j}{c} \end{cases} \quad (14)$$

이상 설명한 augmented Lagrangian 방법을 이용한 최적화 algorithm은 그림 3과 같다.

## 2. 비임계경로의 최적화

임계경로의 최적화가 끝난 후 선택한 모든 임계경로들은 설계자가 지정한 지연시간을 갖는다. 설계자가 설정한 입·출력 노드들 사이에 있는 최적화된 임

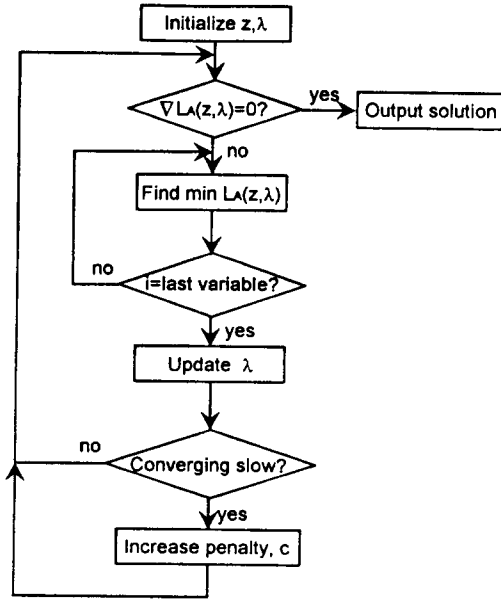


그림 3. Augmented Lagrangian 알고리즘  
Fig. 3. Augmented Lagrangian algorithm.

계경로를 제외한 나머지 경로들인 비임계경로들은 대개 지연시간에 어느 정도 여유를 가지고 있다. MOS 회로에서는 일반적으로 소자의 크기를 작게 하면 지연시간이 증가되고 그 역도 성립되므로, 지연시간 여유를 줄임으로써 소자의 크기 즉, 칩의 면적을 줄일 수 있다. 이 과정을 비임계경로의 최적화 또는 비임계경로의 면적최소화라 하는데 본 절에서는 비임계경로에 대한 면적최소화를 예를 들어 설명하고 알고리즘을 보이도록 한다. 한편, 임계경로의 최적화 후 지연시간이 설정치 보다 크게 된 경로들도 면적최소화 후 설정한 지연시간에 맞게 크기가 조절된다.

그림 4는 비임계경로 면적최소화의 간단한 예이다. 각 부호의 의미는 다음과 같다.

- 실선 임계경로
- 점선 비임계경로
- $T_n$  최적화 이 전의 노드  $n$ 까지의 지연 시간
- $T_n'$  임계 경로 최적화 후의 노드  $n$ 까지의 지연 시간
- $T_n''$  비 임계 경로 최적화 후의 노드  $n$ 까지의 지연 시간
- $I_1, I_2$  입력 노드
- $O_1, O_2$  출력 노드
- $T_M$  설계자가 제시한 출력 노드까지의 지연 시간

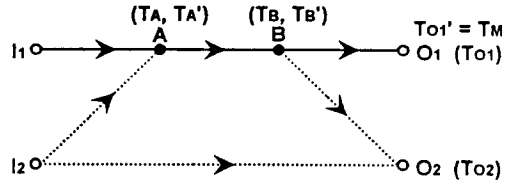


그림 4. 면적최적화의 기본 기본적인 경우  
Fig. 4. A simple graph.

먼저, 임계경로는 최적화가 수행된 후이므로  $T_A'$  과  $T_B'$  은 면적최소화시 변하지 않는 값으로 가정한다. 경로 결정부는 비임계경로의 출력 노드 중 하나를 선택하여 입력 쪽으로 거슬러 가며 면적최소화를 수행한다. 그림 4에 대한 자세한 면적최소화는 다음과 같이 수행된다.

- (1) 경로 (B,  $O_2$ )의 트랜지스터는  $O_2$  부터 시작하여 각 확장 스테이지의 지연시간이  $(T_{O1}' - T_B) / (T_{O2} - T_B)$  배가 되도록 크기를 조절한다.
- (2) 경로 ( $I_2, O_2$ )의 트랜지스터는 지연시간이  $T_M / T_{O2}$  배가 되도록  $O_2$ 부터 시작하여 크기를 조절한다.
- (3) 경로 ( $I_2, A$ )의 트랜지스터는 임계경로의 지연시간 최적화시 노드 A에서의 부하가 달라진 것을 고려하여 노드 A를 포함하는 확장 스테이지부터  $T_A' / T_A$  배로 지연시간을 맞춘다

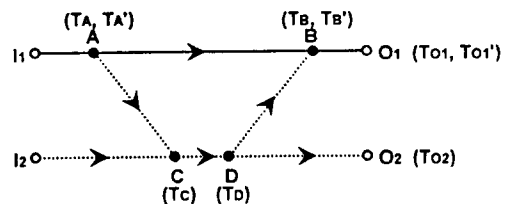


그림 5. 면적최적화의 복잡한 경우  
Fig. 5. A more complicated graph.

그림 5는 그림4 보다 일반적인 경우의 예를 보인 것인데 사용된 부호들의 의미는 그림 4와 같다. 이 경우에는 다음과 같이 면적최소화를 수행한다.

- (1) 경로 ( $O_2, D$ )내 각 확장 스테이지의 지연시간이  $(T_M - T_D) / (T_{O2} - T_D)$ 배가 되도록  $O_2$ 부터 트랜지스터 크기를 조절한다.
- (2) 경로 ( $D, B$ )의 각 트랜지스터는 B부터 시작하여 각 확장 스테이지의 지연시간이  $(T_B' - T_D) / (T_B - T_D)$ 배가 되도록 조절된다.
- (3) 경로 ( $D, C$ )의 트랜지스터는 노드 D에서의

용량변화를 고려하여 지연시간 ( $TD - TC$ )를 유지하도록 크기가 조절된다.

(4) 경로 ( $C, A$ )내 각 확장 스테이지의 지연시간이  $(T_c - T_A)/(T_c - T_A)$ 배가 되도록  $C$ 노드부터 크기를 조절한다.

(5) 경로 ( $C, B$ )의 확장 스테이지가 지연시간  $T_c$ 를 유지하도록  $C$ 노드부터 크기를 조절한다

앞의 예와 같이 주어진 지연시간을 갖는 확장 스테이지의 스케일 지수를 결정하는 문제에는 비선형등식을 풀어야 하는데 여기에는 Brent 알고리즘<sup>9)</sup>을 사용했다.

이상 설명한 내용은 그림 6과 같이 표현된다.

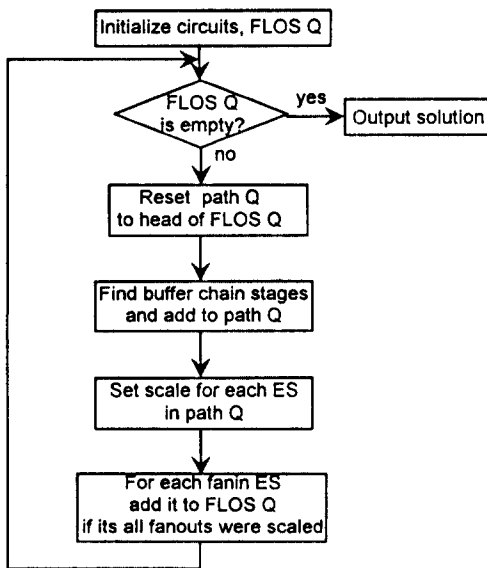


그림 6. 면적최소화 알고리즘  
Fig. 6. Desizing algorithm.

위의 방법을 C 언어와 유사한 알고리즘으로 기술하면 다음과 같다.

```

desize(circuit_list, FLOS_queue)
/* FLOS : An extended stage whose load is fixed */
{
while(FLOS_queue is not empty)
{
/* ES : an Extended Stage */
/* fanin_X : a fanin Extended Stage of X */
/* fanout_X : a fanout Extended Stage of X */
/* X->sized : is the size of X fixed ? */
ES = take_out(FLOS_queue);
  
```

```

initialize(path_queue, ES);
end_delay = absolute delay of ES output node;
while(no of fanin_ES == 1 && no
      of fanout_fanin_ES == 1)
{
put_into(path_queue, fanin_ES);
ES->sized = YES;
ES = fanin_ES;
}
start_delay = absolute maximum delay
              among input node of ES;
path_delay = delay of extended stage
             path in path_queue;
ratio=(end_delay - start_delay)
      /path_delay;
foreach(ES in path_queue)
  set_scale(take_out(path_queue), ratio);
foreach(fanin_ES)
  if(all fanout_fanin_ES->sized == YES)
    put_into(FLOS_queue, fanin_ES);
}
  
```

#### IV. 테스트 결과 및 고찰

본 논문의 시스템은 SUN4/370 workstation의 UNIX환경 하에서 C언어를 사용하여 구현하였다. 테스트에 사용한 회로의 지연시간계산에는 분포 RC 지연 모델을 사용하였고, 면적계산에는 각 소자의 채널 영역의 합을 이용하였다. 또한 다중임계경로에는 5개의 가장 긴 지연시간을 갖는 경로를 이용하였다. 표 (1)은 다중임계경로의 최적화결과이며 area는 최적화된 다중임계경로내 모든 트랜지스터들의 채널영역합이다. 본 논문에서 제시한 알고리즘은 각각의 임계경로에 다른 지연시간을 할당하여 최적화할 수 있으나 테스트에서는 설계자가 요구하는 다중임계경로의 지연시간을 모두 같게 설정하였다. 임계경로들의 최적화결과는 전체적으로 지연시간이 줄어든 대신 면적은 증가함을 보여 주었다. 이는 표 (1)의 8-bit adder에서 임계지연시간을 다르게 주며 면적의 변화를 보인 그림 (7)에서도 알 수 있다. 표 (2)는 표 (1)에 사용한 회로의 비임계경로에 대한 최적화결과이고 area는 비임계경로내 모든 트랜지스터 채널 영역의 합이다. 면적최적화결과 비임계경로의 면적이 약 10%에서 50% 정도까지 줄어드는 것을 알 수 있으며 이는 초기설계의 정도와 최적화시 주어지는 임계

경로의 수에 따라 달라진다. 또한, 표 (1)과 표 (2)의 CPU time을 비교해 보면, 비임계경로의 트랜지스터 수가 임계경로의 트랜지스터 수 보다 여러 배 많음에도 불구하고 계산시간은 임계경로의 최적화에 대략 10배 정도 더 많은 시간이 소요됨을 알 수 있다. 이는 수식적방법이 경험적인 방법보다 매우 많은 시간을 필요로 한다는 것을 보여주는 것으로, 1) 설계시 보다 중요한 임계경로의 지연시간을 맞추기 위해 전역적 최적화가 되는 방법을 사용하면서도 2) 모든 경로를 한번에 다루어 발생하는 시간부담을 줄이자는 본 논문의 방향과 일치함을 알 수 있다.

표 1. 다중임계경로에서 최적화 수행결과(단위: Area [ $\mu m^2$ ], delay [nsec], CPU time [sec])

Table 1. Results of area/delay tradeoffs using transistor sizing in multiple critical paths. (Areas in [ $\mu m^2$ ] and delays in [nsec], and CPU times in [sec].)

Test circuit (# of MOS)	# of sized MOS	Before sizing		After sizing		CPU time
		Area	Delay	Area	Delay	
8-bit adder (400)	134	3919	57.5	4701	50.0	17.3
ALU (472)	90	2205	30.1	2848	24.0	2.5
I/O register (642)	80	333	10.2	393	7.0	0.9
MICOM ALU (808)	168	2872	27.1	3631	24.0	25.7
4-bit parallel multiplier (896)	140	1590	17.1	2082	16.0	32.1

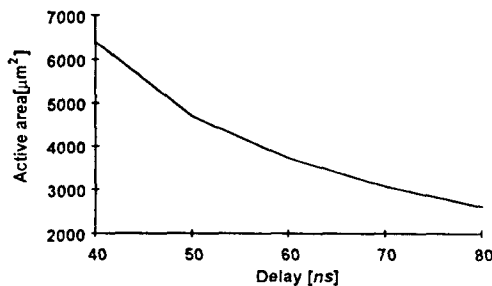


그림 7. 다중임계경로의 지연시간에 따른 면적의 변화 (8bit-adder)

Fig. 7. Variation of the area with the multiple critical path delay (8bit-adder).

표 2. 비임계경로에서의 최적화 수행결과(단위: Area [ $\mu m^2$ ], CPU time [sec])

Table 2. Results of area reduction using transistor desizing in non-critical paths. (Areas in [ $\mu m^2$ ] and CPU time in [sec].)

Test circuit	# of MOS	Area of non-critical paths		CPU time
		Before desizing	After desizing	
8-bit adder	400	1858	956	1.8
ALU	472	251	226	0.3
I/O register	642	4567	2227	8.3
4-bit parallel multiplier	896	919	798	2.9

V. 결론

본 논문에서는 CMOS 회로의 임계경로 최적화를 위한 수학적인 해석과 경험적인 기법을 합한 새로운 최적화기법을 제안하였다. 제안된 기법의 장점은 다중임계경로의 적용으로 발생가능성이 있는 부가적인 임계경로를 방지할 수 있고 지연시간이 중요시되는 부분에는 정확한 수학적해석을 사용하여 문제의 소지를 줄였으며 비임계경로에 효율적인 알고리즘을 사용하여 큰 회로를 빠르게 처리할 수 있었다는 점이다. 전체적인 면적과 지연시간의 변화비율은 지연시간 최적화와 면적최적화에 들어간 소자의 수, 주어진 지연시간, 회로의 입력에서 출력까지의 확장 스테이지 레벨, 전체경로 수에 의해 큰 변화를 보였다. 테스트 과정에서 임계경로 내에 약 700개의 확장 스테이지를 포함한 회로까지 최적화할 수 있었다.

參考文獻

[1] W. Nye, D. C. Riley, A. Sangiovanni-Vincentelli, and A. L. Tits, "DELIGHT. SPICE: An Optimization-based System for the Design of Integrated Circuits," *IEEE Trans. CAD*, vol. 7, no. 4, pp. 501-519, Apr. 1988.

[2] N. P. Jouppi, "Timing Analysis and Performance Improvement of MOS

- VLSI Designs." *IEEE Trans. CAD*, vol. CAD-6, no. 4, pp. 650-665, July 1987.
- [ 3 ] E. T. Lewis. "Optimization of Device Area and Overall Delay for CMOS VLSI designs." *IEEE Proc.* vol. 72, no. 6, pp. 670-689, June 1984.
- [ 4 ] J. D. Pincus and A. M. Despain. "Delay Reduction Using Simulated Annealing." in *Proc. 23rd DAC*, pp. 690-695, 1986.
- [ 5 ] J. P. Fishburn and A. E. Dunlop. "TILOS: A Posynomial Programming Approach to Transistor Sizing." in *IEEE Dig. 1985 Int. Conf. on CAD*, Santa Clara, CA, pp. 326-328, Nov. 1985.
- [ 6 ] P. Penfield and J. Rubinstein. "Signal Delay in RC Tree Networks." in *Proc. 18th DAC*, pp. 613-617, Jun. 1981.
- [ 7 ] J. K. Ousterhout. "A switch-level timing verifier for digital MOS VLSI." *IEEE Trans. Computer-Aided Design*, vol. CAD-4, no.3, pp. 336-349, July 1985.
- [ 8 ] D. Marple and A. E. Gamal, *Optimal Selection of Transistor Sizes in Digital VLSI circuits*, Advances in Research in VLSI, P. Losleben, Ed. Cambridge, MA:MIT Press, 1987.
- [ 9 ] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling, *Numerical recipes in C*, Cambridge university press, p.299 1988.
- [10] D. Luenberger. *Linear and Nonlinear programming*, Second Edition, Addison wesley, Massachusetts, 1984.



## 著者紹介



李尙憲(正會員)

1964年 12月 4日生. 1987年 2月  
충남대학교 전자공학과 졸업.  
1989年 8月 한국과학기술원 전기  
및 전자공학과 (공학석사). 1989  
年 9月 ~ 현재 한국과학기술원  
전기및 전자공학과 박사과정.  
1989年 9月 ~ 현재 삼성전자 반도체(주) 연구원. 주  
관심 분야는 VLSI Performance Optimization,  
Delay modeling, 아날로그 설계자동화 등임.



金敬昊(正會員)

1961年 2月 3日生. 1984年 2月  
연세대학교 전자공학과 졸업(학  
사). 1987年 2月 한국과학기술원  
전기및 전자공학과 (공학석사).  
1991年 2月 한국과학기술원 전기  
및 전자공학과 (공학박사). 1983  
年 ~ 현재 삼성전자 ASIC 및 CAE 선임 연구원.  
주관심 분야는 Deep Submicron 모델링 및 시뮬레  
이션, 아날로그 자동화 및 논리합성, 메모리 테스트,  
교주파 해석 등임.

朴松培(正會員)

1924年生. 1951年 서울대학교 전자공학과 졸업 공학  
사학위 취득. 1962年 및 1968年 미국 Minesota대학  
전기공학과에서 각각 석사학위 및 박사학위 취득.  
1955年 ~ 1968年 서울대학교 전자공학과 조교수.  
1968年 ~ 1971年 미국 Oregon 주립대학 전기공학  
과 조교수 1973年 ~ 현재 한국과학기술원 전기 및  
전자 공학과 교수. 주관심 분야는 초음파 영상과  
Computer Aided Design 등임.