

# 用 RC 网络作延迟模型的开关级 定时模拟

胡 易 王 兆 明

(四川固体电路研究所, 四川, 永川) (电子科技大学, 四川成都)

1987年8月20日收到

本文提出一种用 RC 网络作延迟模型进行开关级定时模拟的方案。此方案把开关级定时模拟划分为求解将来状态和计算状态变化的延迟两个步骤来进行。文中讨论了延迟模型的建立及延迟计算中的有关问题。按照所述方案, 开发了一个适用于 MOS VLSI 逻辑模拟及延迟估算的计算机程序 LOMOS。实践表明, LOMOS 模拟出的信号延迟时间同电路模拟程序 SPICEII 相比误差通常在 30% 以内, 模拟速度要快近三个数量级。

**主题词:** 逻辑模拟, 开关级模型, 开关级模拟, MOS 数字集成电路, 超大规模集成电路

## 一、引言

八十年代发展起来的开关级模拟方法克服了门级逻辑模拟在 MOS 数字集成电路中遇到的困难, 能够准确地模拟 MOS 电路的逻辑功能, 目前已在 MOS VLSI 的设计中获得实际应用。开关级模拟有两种模式:

1. 基于单位时延假设、迭代求解下一状态而获得网络稳态响应的稳态模拟方式。这种方式对一些由信号延迟的时间差决定逻辑功能的电路可能出现差错。

2. 考虑了信号时延信息的开关级定时模拟 (Switch-Level Timing Simulation) 方式。它利用了开关级网络中的基本电学参数 (MOS 管的等效导通电阻及节点电容) 估算节点状态变化的时延, 在一定程度上填补了电路模拟和逻辑模拟之间的鸿沟。

现有的开关级定时模拟方案存在着的一些问题是: 对网络的拓扑结构有所限制<sup>[1,2]</sup>; 需预先确定节点状态变化的时延<sup>[3]</sup>; 不能适当地确定“X”态晶体管等效电阻的取值而可能造成状态计算失误<sup>[4,5]</sup>; 估算的时延值与电路模拟方法相比误差太大而使时延数据失去价值<sup>[6]</sup>。

本文提出一种新的开关级定时模拟方案, 其基本思想是把定时模拟划分为将来状态的求解和节点上信号时延的计算两个步骤来进行。先用图论算法模型<sup>[6]</sup>来求解将来状态, 再构造 RC 网络作为延迟模型来确定状态变化的时延。这样, “X”态 MOS 管等效电阻的取值仅对时延有所影响而不会危及状态值的可信度。并且预先求出了将来状态, 使得延迟模型中的参数修正明确了方向, 从而有利于改善延迟估算的精度。此外, 我们选择的算法对网络拓扑没有限制条件。

下面先讨论 RC 网络的信号延迟及计算中的有关问题, 然后叙述我们的开关级定时模拟方案和延迟模型的建立方法, 最后扼要地介绍作者开发的开关级门级混合模拟程序 LOMOS 以及一些模拟结果。

## 二、RC 网络的信号时延及计算

本文中的 RC 网络是指满足下述条件的线性 RC 网络:

1. 每条浮地支路仅含一个电阻元件;
2. 每个节点上有一个接地电容, 不存在接地电阻;
3. 网络中仅有一个一端接地的阶跃电压源  $U_s(t)$ , 其非地节点称为源节点;
4. 网络中物理量按阶跃电压幅度归一化, 即电压无量纲, 电容与电荷量纲相同。记这种 RC 网络为  $N_{RC}$ 。特别地, 若  $v_s(t) = u(t)$  记为  $N_{RC}$ ; 若  $v_s(t) = 1 - u(t)$ , 记为  $N_{RC}^0$ , 其中  $u(t)$  是单位阶跃函数。可以用一个无向图来表示  $N_{RC}$ , 如图 1 所示。

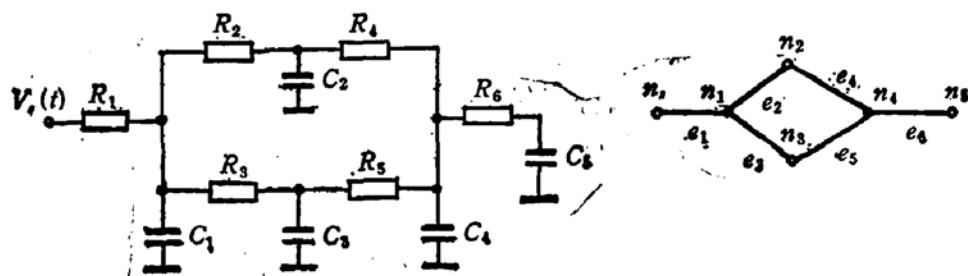


图 1 一个 RC 网络及其图

设  $N_{RC}$  中节点  $n_i$  的单位阶跃响应为  $y_i(t)$ , 定义  $n_i$  上的信号延迟时间为

$$t_i = \int_0^\infty [1 - y_i(t)] dt \quad (1)$$

其意义可由图 2 说明。

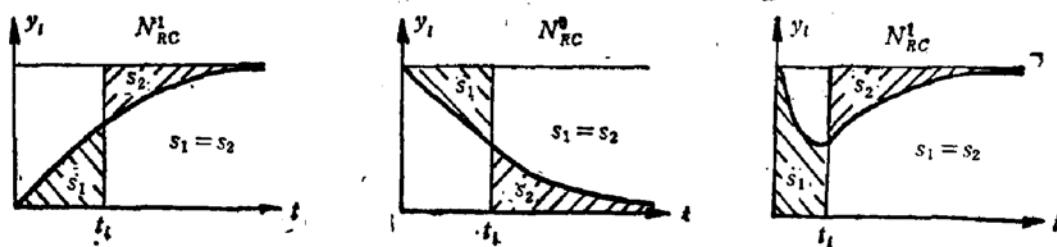


图 2 RC 网络信号时延的定义

对于零初始状态的 RC 树形网络, 式(1)定义的延迟位于波形界法所确定的延迟上界与下界之间。由于这一定义同零状态单位阶跃响应的拉氏变换和初始状态存在着密切联系, 可以导出一套系统地计算  $t_i$  的方法, [4] 中对此作了详细的论述。我们将[4]建立的计算 RC 网络信号时延的方法称为 L-M 方法, 其主要步骤如下:

1. 将一般拓扑结构的 RC 网络分解成若干个 RC 树形网络；
2. 对各个 RC 树形网络作时延计算；
3. 用 LRD(Load ReDistribution) 算法调整负载电容在各树形网络上的分布，计算电容分布变化后的时延，直到各等价节点(见下节)上的时延值相等。

对后两个步骤，[4]中给出了明确的算法。下节就第一个步骤给出我们的解决方法。

### 三、RC 网络树形分解算法

RC 网络树形分解是指将  $N_{RC}$  的某些节点分裂(split)，成为 RC 树网络集合  $N'_{RC}$ ，即

$$N_{RC} \rightarrow N'_{RC} = \{N'^0_{RC}, N'^1_{RC}, \dots, N'^m_{RC}\} \quad (2)$$

各 RC 树的根都对应着  $N_{RC}$  的源节点  $n_s$ 。设节点  $n_i$  被分裂成  $i$  个节点  $n(i, k)$ ， $k = k_1, \dots, k_i$ ，称其为  $n_i$  的等价节点。各等价节点的电容须满足

$$\sum_{k=k_1}^{k_i} c_{(i,k)} = c_i \quad (3)$$

其中  $c_{(i,k)}$ 、 $c_i$  分别是  $n(i,k)$ 、 $n_i$  的电容。

为了保证 LRD 算法的收敛，分解时还要求

$$R_{s(i,v)} = 0, k, v = k_1, \dots, k_i, k \neq v \quad (4)$$

其中  $R_{s(i,v)}$  是  $n(i,k)$  和  $n(i,v)$  到  $n_s$  的公共路径上的电阻之和。一般情况下，这势必造成某些边的分裂。我们可以将被分裂的边视为若干边并联而成。设边  $e_i$  分裂成  $m$  条边  $e_{(i,k)}$ ， $k = k_1, \dots, k_m$ ，于是应有

$$\sum_{k=k_1}^{k_i} \frac{1}{R_{(i,k)}} = \frac{1}{R_i} \quad (5)$$

其中  $R_{(i,k)}$ 、 $R_i$  分别是  $e_{(i,k)}$ 、 $e_i$  的电阻。

按照下面的 TDS (Tree Decomposition Scheme) 算法，可以将  $N_{RC}$  分解成符合上述要求的 RC 树集合。TDS 的基本思想是从  $n_s$  出发作 DFS (Depth-First Search，深度优先搜索)<sup>[7]</sup>，将  $N_{RC}$  分解成 DFS 树和若干棵无分支树的集合。以下给出 TDS 的伪码，为简洁起见，略去了涉及到式(3)和式(5)的部分。

#### procedure TDS

“ $N: N_{RC}$  的节点集合； $E: N_{RC}$  的边集合”

“ $AE: N_{RC}$  中节点的关联边集合； $n1, n2$ : 边的端点”

“ $SONS: N'^0_{RC}$  中节点的子节点集合； $son: N'^k_{RC}$  中节点的子节点； $k > 0$ ”

“ $father: N'^k_{RC}$  中节点的父节点； $ENS: N'^0_{RC}$  中节点的等价节点集合”

“ $Path$ : 存贮  $n_s$  到  $n_i$  路径的数组； $mark$ : 搜索标志”

“ $e1, e2: N'^k_{RC}$  中节点的关联边， $e1$  连向父节点  $e2$  连向子节点  $k > 0$ ”

procedure SEARCH ( $e_i, n_i, p, k$ )

begin “SEARCH”

$e_i \cdot mark := true;$

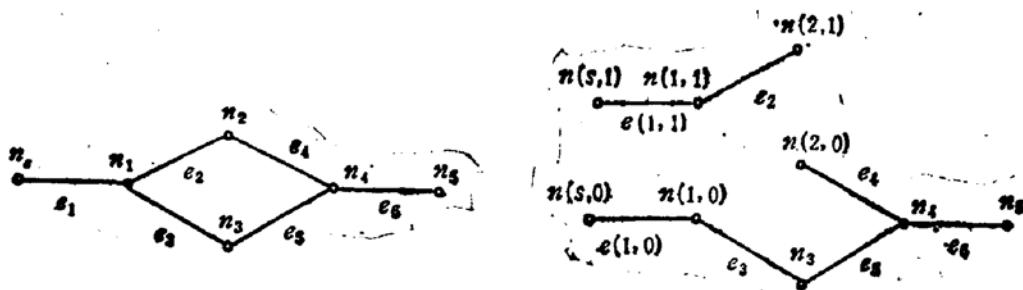
```

if  $n_i = e_i \cdot n_1$  then  $n_s := e_i \cdot n_2$  else  $n_s := e_i \cdot n_1$ ;
if  $n_s \cdot \text{mark} = \text{false}$  then
begin "forward"
   $n_s \cdot \text{mark} := \text{true}$ ;  $n_s \cdot \text{father} := n_i$ ;
   $n_i \cdot \text{SONS} := n_i \cdot \text{SONS} \cup n_s$ ;
   $\text{Path}(p) := e_i$ ;  $\text{Path}(p+1) := n_s$ ;  $p := p+2$ ;
  for  $e_j \in n_s \cdot \text{AE}$  do
    if  $e_j \cdot \text{mark} = \text{false}$  then  $\text{SEARCH}(e_j, n_s, p, k)$ ;
     $p := p-2$ ;
end "forward"
else begin "decomposition"
   $k := k+1$ ;  $j := 2$ ;  $fath := n_{(i,k)}$ ;
  while  $\text{Path}(j) \neq n_s$  do begin
     $e_p := \text{Path}(j-1)$ ;  $n_p := \text{Path}(j)$ ;  $j := j+2$ ;
     $n_p \cdot \text{ENS} := n_p \cdot \text{ENS} \cup n_{(p,k)}$ ;  $n_{(p,k)} \cdot \text{father} := fath$ ;
     $fath \cdot \text{son} := n_{(p,k)}$ ;  $fath \cdot e2 := e_{(p,k)}$ ;
     $n_{(p,k)} \cdot e1 := e_{(p,k)}$ ;  $fath := n_{(p,k)}$ ;
  end "while"
   $E := E - e_i$ ; "remove  $e_i$  from E"
   $n_{(s,k)} \cdot \text{son} := n_{(i,k)}$ ;  $n_{(i,k)} \cdot \text{father} := n_{(s,k)}$ ;
   $n_{(s,k)} \cdot e1 := e_i$ ;  $n_{(i,k)} \cdot e2 := e_i$ ;
end "decomposition"
end "SEARCH"
begin "TDS"
   $k := 0$ ;  $p := 1$ ;
  initialize; "set all SONS and ENS to nil, mark to false"
   $n_s \cdot \text{mark} := \text{true}$ ;
  for  $e_i \in n_s \cdot \text{AE}$  do
    if  $e_i \cdot \text{mark} = \text{false}$  then  $\text{SEARCH}(e_i, n_s, p, k)$ ;
end "TDS"

```

过程 SEARCH 开始时,由节点  $n_i$  通过边  $e_i$  搜索到  $n_s$ 。如果  $e_i$  不是回退边,作必要的记录后继续往下搜索;如果  $e_i$  是回退边,则对  $N_{RC}$  进行分解,生成第  $k$  个无分支的树网络  $N_{RC}^k$ ,在这部分伪码中,带有一对下标的节点和边都是新生成的。TDS 完成之后,  $N_{RC}$  变为 DFS 树,即  $N_{RC}^{t0}$ 。我们看到,等价节点都分布在不同的树中,式(4)自然得到满足。例如,对图 1 的  $N_{RC}$ ,执行 TDS 后得到的结果见图 3。

节点分裂时,节点电容的分配只要满足式(3)即可,因为在作 LRD 时还要对等价节点上的电容进行调整,使其时延相等;边分裂时电阻的分配除了要满足式(5)外,还有一个很显然的要求,即  $R_{(i,k)} \neq \infty$ , 执行 LRD 时边电阻不再变化。

图 3  $N_{RC}$  树形分解结果

用 TDS 作树形分解所得到的  $N'_{RC}^k$  ( $k > 0$ ) 是无分支树, 这可以用来进一步改善 LRD 算法的效率。由于  $N'_{RC}^0$  的节点同  $N_{RC}$  一一对应, 我们就可将 LRD 迭代时的扫描范围缩小, 仅对  $N'_{RC}^0$  扫描。对一个节点扫描时同时处理该节点的等价节点上负载电容的调整和时延的计算。因为  $N'_{RC}^k$  ( $k > 0$ ) 中不存在兄弟节点<sup>[6]</sup>, 也就没有负载调整时的交叉影响, 使得负载调整对时延影响的传递非常方便。在编写程序中我们按照 TDS 产生的网络结构改进了 LRD 算法。

#### 四、RC 网络延迟模型建立方法

开关级模拟算法的基本线索是在网络划分基础上的活动路径跟踪。作开关级定时模拟时, 对一个子网络的计算归结为两点:

1. 确定各存贮节点的将来状态;
2. 确定各存贮节点到达将来状态所需要的时间, 即信号传播的时延。

将两个问题放在一起处理难免顾此失彼, 我们采用了分而治之的策略。对前一问题以 Bryant 的图论算法模型为基础<sup>[6]</sup>、将节点状态定义加以扩展(附加一个表示连通状况的标志)来求解将来状态。这是一种比较可靠的模型, 它妥善地考虑了“X”态 MOS 管对网络响应的影响。后一问题用前面所述的 RC 网络来计算。模拟调度算法采用变步长时钟的时间匹配算法, 时钟步长由所记录到的事件发生的最早时间确定。所要考虑的关键问题是如何将两个问题联系起来。下面先引入两个定理。

设  $G_{sw} = \langle N, E \rangle$  是时间  $t^i$  时一个子网络的开关图<sup>[6]</sup>,  $G_{swc} = \langle N_c, E_c \rangle$  是  $G_{sw}$  的一个连通子图。令  $N_c = N_{ci} \cup N_{cs}$ , 其中  $N_{ci}$  是输入节点集合,  $N_{cs}$  是存贮节点集合。 $y_i$  表示  $n_i$  的当前状态,  $y'_i$  表示  $n_i$  的将来状态。

**定理 1** 若  $\exists n_i \in N_{cs}$ ,  $y'_i = "1"$  且  $n_i$  与输入节点连通, 则  $\exists n_i \in N_{ci}$ ,  $y_i = "1"$ 。

**定理 2** 若  $\exists n_i \in N_{cs}$ ,  $y'_i = "0"$  且  $n_i$  与输入节点连通, 则  $\exists n_i \in N_{ci}$ ,  $y_i = "0"$ 。

我们已对这两个定理作了严格证明, 此处从略。

由  $G_{swc}$  确定了  $N_{cs}$  中诸节点的将来状态后, 就可建立  $G_{swc}$  所对应的开关级网络的延迟模型。任取  $n_i \in N_{cs}$ , 若  $n_i$  与输入节点不连通(由扩展的节点状态直接判断), 则将来状态是由电荷存贮或静态电荷分配造成, 不必作详细的时延计算; 否则, 有下面四种情况:

**情况 1**  $\forall n_i \in N_{CS}, y'_i \neq "0"$  且  $\exists n_i \in N_{CS}, y'_i = "1"$ ;

**情况 2**  $\forall n_i \in N_{CS}, y'_i \neq "1"$  且  $\exists u_i \in N_{CS}, y'_i = "0"$ ;

**情况 3**  $\exists n_i \in N_{CS}, y'_i = "1"$  且  $\exists n_k \in N_{CS}, y'_k = "0"$ ;

**情况 4**  $\forall n_i \in N_{CS}, y'_i = "X"$ .

延迟模型的建立方法如下(方法 1 针对情况 1, 依此类推):

**方法 1** 构造  $N_{CI}^1 = \{n_i | n_i \in N_{CI}, y_i \neq "0"\}$  代替  $N_{CI}$ , 即在  $G_{SWC}$  中删除“0”状态的输入节点, 然后将  $N_{CI}$  合并成一个“1”状态的节点  $n_s^1$ , 这样  $G_{SWC}$  变成一个以  $n_s^1$  为源节点的 RC 网络图  $N_{RC}^1$ , 其中各边电阻等于  $G_{SWC}$  中相应边所代表的 MOS 管的等效电阻, 节点电容等于  $G_{SWC}$  中各节点的有效电容。由  $N_{RC}^1$  计算出  $G_{SWC}$  中各节点的时延。

**方法 2** 构造  $N_{CI}^0 = \{n_i | n_i \in N_{CI}, y_i \neq "1"\}$  代替  $N_{CI}$ , 将  $N_{CI}^0$  合并成一个“0”状态节点  $n_s^0$ , 得到以  $n_s^0$  为源节点的  $N_{RC}^0$ 。其它同方法 1。

**方法 3** 分别按方法 1 和方法 2 构造 RC 网络  $N_{RC}^1$  和  $N_{RC}^0$ 。记由  $N_{RC}^1$  计算的时延为  $t_{di}^1$ , 由  $N_{RC}^0$  得到的时延为  $t_{di}^0$ 。节点  $n_i \in N_{CS}$  的时延  $t_{di}$  用下式确定

$$t_{di} = \begin{cases} t_{di}^1 & y'_i = "1" \text{ 或 } y'_i = "X", y_i = "0" \\ t_{di}^0 & y'_i = "0" \text{ 或 } y'_i = "X", y_i = "1" \\ \max\{t_{di}^1, t_{di}^0\} & \text{其它情况} \end{cases} \quad (6)$$

**方法 4** 分别按方法 1 和方法 2 构造  $N_{RC}^1$  和  $N_{RC}^0$ 。若  $N_{CI}^1 = \phi$  ( $N_{CI}^0 = \phi$ ), 则对  $\forall n_i \in N_{CS}$ , 令  $t_{di}^1 = 0$  ( $t_{di}^0 = 0$ ), 这里  $\phi$  表示空集。然后对建立成功的 RC 网络作延迟计算, 按下式确定  $n_i$  的时延

$$t_{di} = \begin{cases} t_{di}^1 & y_i = "0", t_{di}^1 \neq 0 \\ t_{di}^0 & y_i = "1", t_{di}^0 \neq 0 \\ \max\{t_{di}^1, t_{di}^0\} & \text{其它情况} \end{cases} \quad (7)$$

定理 1 和定理 2 保证了对前两种情况能构造出一个 RC 网络来计算延迟; 对第三种情况能构造两个 RC 网络; 最后一种情况至少能构造出一个 RC 网络。因此, 各种情况下都能确定出相应的时延。

## 五、延迟模型参数的确定

RC 网络延迟模型中的电阻、电容和初始电荷分别对应开关级网络中 MOS 管的等效导通电阻、节点电容和节点当前状态, 确定方法如下:

### 1. 电阻 $R_i$

设 MOS 管导通等效电阻  $R_{ON}$  为常量, 近似认为

$$R_{ON} = \frac{L}{W} r \quad (8)$$

其中,  $r$  是每方沟道的导通等效电阻,  $L$ 、 $W$  分别是沟道的长和宽。根据  $\tau$  模型<sup>[9]</sup>, 用电路模拟程序对标准倒相器链模拟后经简单计算得到  $r$ 。在工艺条件不变时,  $r$  为定值。

一般情况下,  $R_i$  取值为

$$R_i = \begin{cases} R_{ONi} & \text{MOS 管为“1”态} \\ K_x \cdot R_{ONi} & \text{MOS 管为“X”态} \end{cases} \quad (9)$$

其中  $K_x$  是对 “X” 态管等效电阻的修正系数, 建议取为 4 左右。

若要进一步改善时延估算的精度, 可根据漏、源节点的将来状态判断出传播的信号, 对  $R_{ON}$  乘上一个修正系数来作为“1”态 MOS 管的等效电阻<sup>[10]</sup>。在 NMOS 电路中, 对传递管驱动的 MOS 管等效电阻也可以加以修正。

## 2. 电容 $C_i$

电容  $C_i$  等于实际网络中的节点电容, 即

$$C_i = \sum_j S_{gi} C_g + S_d C_d + S_p C_p + S_m C_m \quad (10)$$

其中,  $S_{gi}$  是以  $n_i$  为栅节点的第  $i$  个 MOS 管的栅区面积;  $S_d$ 、 $S_p$ 、 $S_m$  分别是  $n_i$  的扩散、多晶和金属层面积;  $C_g$ 、 $C_d$ 、 $C_p$ 、 $C_m$  分别是栅区、扩散区、多晶层和金属层的单位面积电容。

单位面积电容由一定的工艺参数计算获得, 各单位面积则直接来源于掩膜图。

## 3. 初始电荷 $Q_i$

当节点  $n_i$  处于非过渡区(已到达前一次计算出来的将来状态)时,  $C_i$  上的初始电荷为

$$Q_i = \begin{cases} C_i & y_i = "1" \\ \frac{1}{2} C_i & y_i = "X" \\ 0 & y_i = "0" \end{cases} \quad (11)$$

若  $n_i$  处于过渡区, 则在上式基础上用线性插值估计。

## 4. 有效电容 $C_{ei}$

在对  $N_{RC}$  作延迟计算时, 我们用有效电容来作为  $N_{RC}$  的节点对地电容, 使对  $N_{RC}^1$  和  $N_{RC}^0$  的延迟计算都归结为对零初始电荷条件下  $N_{RC}^1$  的延迟计算。有效电容的定义为

$$C_{ei} = \begin{cases} C_i - Q_i & N_{RC} = N_{RC}^1 \\ Q_i & N_{RC} = N_{RC}^0 \end{cases} \quad (12)$$

顺便指出, 以上确定参数的方法是不考虑器件非线性效应和输入波形影响的近似方法。开关级定时模拟的目的是快速估算电路的时延, 上述方法已能满足实用要求。

## 六、LOMOS 及实验结果

LOMOS (Logic simulator for MOS digital circuits) 是针对 MOS 数字集成电路的特点设计的开关级门级混合模拟程序, 其开关级定时模拟运用了本文所述之方案。LOMOS 的输入是自由格式语言编写的电路和信号文件, 以交互方式运行, 输出结果为逻辑波形图。它能够模拟出开关级门级混合网络的逻辑功能和信号延迟, 还可以模拟互连线上的时延, 特别适合于 MOS 数字集成电路的功能模拟和时延估算。LOMOS 由 4700 余条 Fortran 77 语句和少量的汇编语句组成, 运行在操作系统为 UNIX 的 Dual 83/20

微机上。

我们对一些电路用 LOMOS 作了开关级定时模拟，并同 SPICE2 的模拟结果进行了比较。现将部分数据摘录整理成图 4 和表 1。

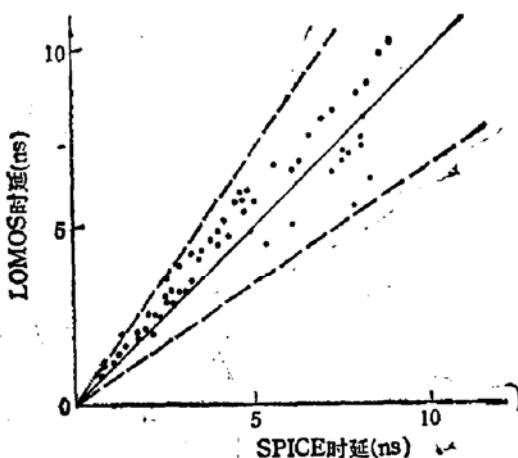


图 4 LOMOS 的时延误差示意图

表 1 LOMOS 与 SPICE 的速度比较

电路名称	MOS 管数目	CPU <sub>LOMOS</sub>	CPU <sub>SPICE</sub>	5 * CPU <sub>SPICE</sub> CPU <sub>LOMOS</sub>
组合逻辑电路	39	0.5 秒	57 秒	570
1 位传递逻辑全加器	26	0.5 秒	38.1 秒	381
4 位传递逻辑全加器	104	0.9 秒	167.5 秒	932

注: SPICE 在 VAX11/780 上运行。经测试, VAX11/780 比 Dual 83/20 快 6—8 倍。表中对 CPU<sub>SPICE</sub>乘 5 只是为了作粗略的估评。

## 七、结语

本文提出一种用开关级网络的图论算法模型来确定将来状态、用 RC 网络作时延计算的开关级定时模拟方案, 给出了 RC 网络的树形分解算法, 从而解决了 RC 网络延迟计算中的一个实际问题, 并且详细讨论了在开关图的基础上建立 RC 网络延迟模型的方法。用这种开关级定时模拟方案得到的时延与电路模拟程序 SPICE 相比误差通常在 30% 以内, 速度要快 2—3 个数量级, 能够满足 MOS VLSI 逻辑功能验证和时延性能估算的要求。

作者在此感谢顾德仁教授对本文工作的支持与帮助。

## 参 考 文 献

- [1] C. J. Terman, IEEE ICCT'83, 437—440 (1983).
- [2] V. B. Rao and T. N. Trick, IEEE ISCAS'85, 229—232 (1985).
- [3] M. Kawai and J. P. Hayes, 21st DAC, 2—9 (1984).

- [4] T. M. Lin and C. A. Mead, *IEEE Trans. on CAD*, **CAD-3**, 331—349 (1984).
- [5] M. H. Heydemann, G. D. Hachtel and M. R. Lightner, *IEEE ICCC'82*, 46—49 (1982).
- [6] R. E. Bryant, *IEEE Trans. on Computers*, **C-33**, 160—177 (1984).
- [7] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks, and Algorithms*, 449—457, John Wiley & Sons, Inc. (1981).
- [8] 左孝凌、李为盛、刘永才, 离散数学, 329, 上海科技文献出版社(1982).
- [9] C. A. Mead and L. A. Conway, *Introduction to VLSI systems*, Addison Wesley (1980).
- [10] J. K. Ousterhout, *IEEE 21st DAC*, 542—548 (1984).

## Switch-Level Timing Simulation Using RC Network as Delay Model

Hu Yi

(Sichuan Institute of Solid-State Circuits, Sichuan, Yongchuan)

Wang Zhaoming

(Chengdu Institute of Radio Engineering, Chengdu)

### Abstract

A new method for switch-level timing simulation is proposed. We suggest that the timing simulation should be performed by two steps. First, the future state is evaluated. Then, some RC networks are constructed to calculate signal delay. The issues on implementation of signal delay calculation and construction of RC network delay model are discussed. A computer program LOMOS (LOgic simulator for MOS digital circuits) is developed. Experiments show that LOMOS runs two to three orders of magnitude faster than SPICE with delay errors falling within 30% usually.

**KEY WORDS:** Logic simulation, Switch-level model, Switchlevel simulation, VLSI, MOS digital integrated circuits