

同时非均匀分段和非统一缓冲器插入算法^①

马 鸿^② 何庆延* 彭思龙

(中国科学院自动化研究所国家专用集成电路设计工程研究中心 北京 100080)

(* Takumi 科技公司 圣克拉拉 加利福尼亚 94085)

摘 要 提出了非均匀分段和非统一缓冲器插入算法,针对长互连线驱动大负载的问题,采用逐级增大的缓冲器驱动模型并调节不同缓冲器之间线段的长度,解决了缓冲器插入获得最小延时、减少插入数目和降低计算量的问题,给出了缓冲器插入延时的准确估计。该算法充分考虑了连线的电阻和电容效应,以及芯片上不能随便插入缓冲器的块对缓冲器插入算法的影响。实验表明,该算法可以很好地解决长线网驱动大负载问题,在布图布局阶段采用该算法可以对长互连线驱动大负载的情况给出准确的估计,保证设计和时序收敛,并节省布线资源。

关键词 非均匀分段,缓冲器链,电容效应,梯度算法,Elmore 延时

0 引 言

在电路的时序性能优化技术中,调整驱动门尺寸、连线尺寸和插入缓冲器^[1,2]被广泛使用。其中,缓冲器插入对减小传输延时,提高信号完整性以及减小传输线效应都十分有效^[2,4]。缓冲器插入模型主要分为两类:统一的插入和非统一的插入。统一的缓冲器插入在线网上插入大小相同或相似的缓冲器^[5,6],然而,当负载电容远远大于缓冲器的输入电容时,最后一级缓冲器的驱动能力不足,该技术就变得不是很有用了。更好的做法是采用一串缓冲器尺寸逐级递增的缓冲器链来驱动负载,从而得到最优延时,该技术被称为非统一缓冲器插入。传统的非统一缓冲器插入技术只考虑缓冲器和负载,没有考虑连线的影响,一般选择的缓冲器尺寸比为 4^[1]。在深亚微米工艺中,由于连线延时占据了延时的主要部分,不考虑连线影响的非统一缓冲器插入技术就不够准确了。多篇论文讨论了该问题^[7-10],Chu 和 Wong 提出了同时缓冲器插入和线宽调整技术^[7,8],该方法考虑了连线的电阻和电容效应,但只在离散域对缓冲器尺寸和连线宽度进行调整。Chang 等采用更为精确的模型^[9],连续调整缓冲器尺寸和连线宽度并解决驱动大负载问题,但由于布线通道资源非常珍贵,同时增大连线宽度会增大连

线的电容,对信号完整性造成不利影响,所以在布线资源比较紧张以及对信号完整性要求比较高的情况下,这种方法是不可行的^[11-13]。

片上系统(system on a chip, SOC)设计技术的不断发展促使 IP 技术和层次化设计技术被广泛采用,这简化了系统设计,但给缓冲器插入带来了很大困难,它导致芯片上不能随便插入缓冲器的块越来越多(尽管这些块内可以走线)。在大的阻挡块内部,连线会很长,同时驱动多个器件,对于驱动它的上一级缓冲器而言这等效为一个很大的负载。后版图互连优化阶段出现了很多缓冲器插入算法^[5-8],然而,如果版图上大多数布线区域被覆盖,缓冲器插入会非常困难,越来越多的缓冲器插入时序估计算法出现在布图布局阶段^[9-12]。

本文提出了非均匀分段和非统一缓冲器插入技术,解决了线网经过大阻挡块时如何获得最优延时并降低计算量的问题。通过同时调整缓冲器宽度和缓冲器间距比来达到延时最优,并大大节约了计算量。在布图布局阶段采用该算法可以准确地计算长互连线驱动大负载的延时,保证设计和时序收敛,并节省布线资源。

1 参数和模型

本文采用的参数基于 SIA'97 的 0.18 μ m 工艺,

① 国家科技支撑计划(2006BAK07B04)资助项目。

② 女,1981 年生,博士生;研究方向:互连优化和版图优化;联系人, E-mail: hong.ma@ia.ac.cn
(收稿日期:2008-08-26)

其数值如表 1 所示。

表 1 SIA'97 的 0.18 μm 工艺参数

参数	描述	单位	值
h	连线的宽度	μm	0.18
r_0	线表面电阻	Ω	0.068
c_0	线单位面积电容	$\text{fF}/\mu\text{m}^2$	0.06
c_f	连线的侧边电容	$\text{fF}/\mu\text{m}$	0.064
T_{in}	缓冲器的本征延时	ps	36.4
c_b	最小缓冲器的输入电容	fF	23.4
r_b	最小缓冲器的输出电阻	Ω	180

为了计算方便,本文采用 Elmore 模型计算延时^[11-14]。缓冲器和连线的模型如图 1 所示,采用开关级 RC 电路模拟缓冲器并同时考虑缓冲器本征延时,其中 r_b 是缓冲器输出电阻, c_b 是缓冲器输入电容,其本征延时为 k_b ,采用 π 模型模拟连线段,并考虑连线的侧边寄生电容($c_e = c_0 h + c_f$)。

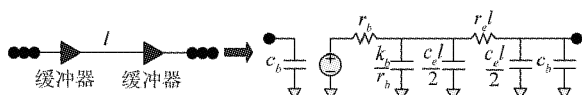


图 1 缓冲器的开关级 RC 模型和连线段的 π 模型

本文主要解决线网通过大阻挡块时,缓冲器插入备选位置和缓冲器尺寸的选择问题。为了简化讨论,构造如图 2 所示的线网模型。图 2 中,线网穿过阻挡块 A(宽度为 L_A),经过距离 L 后又穿越阻挡块 B(宽度为 L_B),两个阻挡块之间长度为 L 的连线区域可以自由插入缓冲器。按照传统的均匀分段模型,需要在该线网上均匀插入尺寸大小相同的缓冲器来达到最优延时。然而,当负载电容远大于缓冲器输入电容时,最后一级缓冲器驱动能力不足,造成时序性能下降。更好的做法是采用一串缓冲器尺寸逐渐递增的缓冲器链来驱动大负载,从而得到最优的延时。

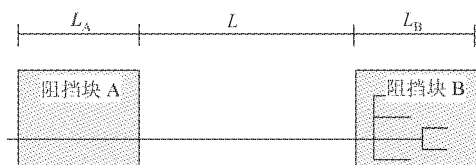


图 2 线网穿越大阻挡块模型

同时缓冲器插入和线宽调整技术^[9]在均匀分段的基础上通过调整缓冲器链中缓冲器大小和连线宽度来计算最优延时。虽然该技术更好地解决了大负载的驱动问题,但由于布线通道资源紧张(特别是在

比较低的布线层),增大连线宽度也必然增大线电容,从而对信号完整性造成不利影响,所以在布线资源比较紧张和对信号完整性要求比较高的情况下,这种方法是不可行的。

2 同时非均匀分段和非统一缓冲器插入技术

为了解决上述问题,本文提出了非均匀分段和非统一缓冲器插入技术,通过同时调整缓冲器宽度和互连线段长度,达到延时最优并同时减小了计算量。如图 3 所示,为了保证穿越阻挡块 A 中线段 L_A 的负载最小,最小尺寸的解耦缓冲器需要在位置 a 放置。定义穿越阻挡块 B 的线段 L_B 的电容及其驱动负载为 C_L ,需要在位置 c 放置大尺寸缓冲器。这样,b,c 之间的互连线采用尺寸逐级增大的缓冲器链来驱动大小为 C_L 的负载。

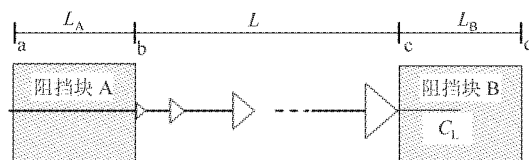


图 3 线网穿越大阻挡块时缓冲器插入方式

通过选择最佳的缓冲器宽度比 β 和线段长度比 ω 来获得最小延时和最小缓冲器数。其中,缓冲器宽度比指第 $i+1$ 个缓冲器宽度和第 i 个缓冲器宽度的比值,线段长度比指第 $i+1$ 段和第 i 段的线段长之比。如图 4 所示,我们在长度为 L 的线段上插入 N 个缓冲器,缓冲器的宽度比为 β ,线段长度比为 ω ,负载电容为 C_L ,源端缓冲器为最小宽度的缓冲器。

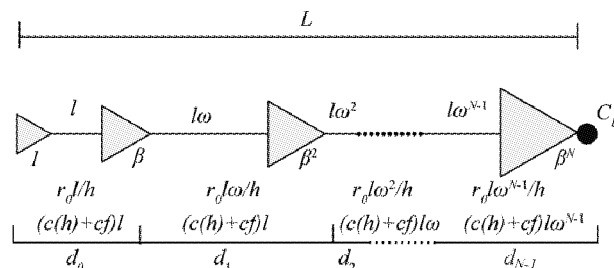


图 4 同时缓冲器宽度和线长调整技术模型

算法表述如下:

算法:

同时非均匀分段和非统一缓冲器插入算法。

输入:

连线长度 L , 驱动电阻 r_b (单位缓冲器), 负载 C_L , 连线单位面积电容 c_0 , 连线单位长度侧边电容 c_f , 连线表面电阻 r_0 , 最小缓冲器输入电容 c_b , 最小缓冲器输出电阻 r_b , 缓冲器插入的个数 N 和阻挡块的宽度 L_B 。

目标:

确定缓冲器的宽度比 β 和线段长度比 ω , 使得连线 L 上的延时最短。

由图 4 可以得到, 每段连线的延时为

$$\begin{aligned} d_i &= k_b + r_b(c_0 h l \omega^i + c_f l \omega^i + c_b \beta^{i+1}) / \beta^i \\ &\quad + r_0 l \omega^i (c_b \beta^{i+1} + (c_0 h l \omega^i + c_f l \omega^i) / 2) / h \\ d_N &= k_b + r_b C_L / \beta^{N-1} \end{aligned} \quad (1)$$

其中, $0 < i < N$, 因此总延时 $D_N(\beta, \omega)$ 可以表示为

$$\begin{aligned} D_N(\beta, \omega) &= k_b + r_b C_L / \beta^{N-1} + \sum_{i=0}^{N-2} k_b \\ &\quad + r_b(c_0 h l \omega^i + c_f l \omega^i + c_b \beta^{i+1}) / \beta^i \\ &\quad + r_0 l \omega^i (c_b \beta^{i+1} + (c_0 h l \omega^i + c_f l \omega^i) / 2) / h \end{aligned} \quad (2)$$

定理: 上述公式中, 延时 $D_N(\beta, \omega)$ 是缓冲器宽度比 β 和线段长度比 ω 的凸函数。

证明如下: 将公式(2)对缓冲器宽度比 β 求二次偏导数可以得到

$$\begin{aligned} \frac{\partial^2 D_N(\beta, \omega)}{\partial \beta^2} &= N(N-1) r_b C_L / \beta^{N+1} \\ &\quad + \sum_{i=0}^{N-1} i(i+1) r_b (c_0 h l \omega^i + c_f l \omega^i) / \beta^{i+2} \\ &\quad + i(i+1) r_0 l c_b \beta^{i-1} / h \end{aligned} \quad (3)$$

同样将公式(2)对线段的长度比 ω 求二次偏导数可以得到

$$\begin{aligned} \frac{\partial^2 D_N(\beta, \omega)}{\partial \omega^2} &= \sum_{i=0}^{N-1} r_b i(i-1) (c_0 h l + c_f l) \omega^{i-2} / \beta^i \\ &\quad + i(i-1) r_0 l c_b \beta^{i+1} \omega^{i-2} \\ &\quad + i(2i-1) (r_0 l^2 c_0 h + r_0 l^2 c_f) \omega^{2i-2} / h \end{aligned} \quad (4)$$

将公式(2)对 β 和 ω 求导可得到

$$\begin{aligned} \frac{\partial^2 D_N(\beta, \omega)}{\partial \beta \partial \omega} &= \sum_{i=0}^{N-1} -i^2 r_b (c_0 h l \omega^{i-1} + c_f l \omega^{i-1}) / \beta^{i+1} \\ &\quad + i(i+1) r_0 l \omega^{i-1} c_b \beta^i \end{aligned} \quad (5)$$

不难得到, $\frac{\partial^2 D_N(\beta, \omega)}{\partial \beta^2}$, $\frac{\partial^2 D_N(\beta, \omega)}{\partial \omega^2}$ 和 $\frac{\partial^2 D_N(\beta, \omega)}{\partial \beta^2} + \frac{\partial^2 D_N(\beta, \omega)}{\partial \omega^2} - \frac{\partial^2 D_N(\beta, \omega)}{\partial \beta \partial \omega}$ 都大于 0, 在插入 N 个缓冲器时, 延时 $D_N(\beta, \omega)$ 是 β 和 ω 的

凸函数。所以其局部最优解即为全局最优解。问题可以采用任何有效的搜索算法(如梯度搜索算法等)来获得最优解。图 5 是公式(2)在 ($N=4$, $L=1\text{cm}$, $C_L=10\text{pf}$) 情况下的一个示例, 很明显, 延时 $D_N(\beta, \omega)$ 是一个凸函数。

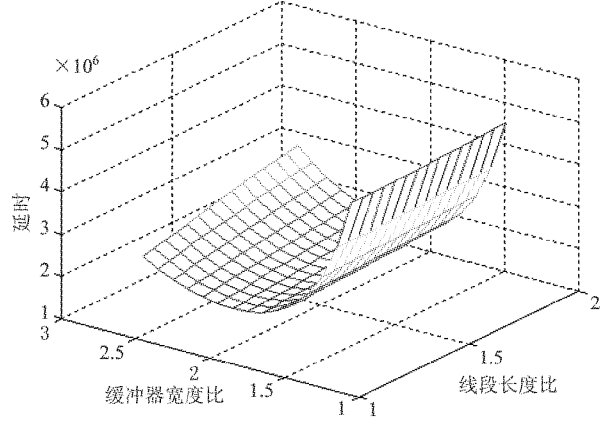


图 5 缓冲器宽度比和线段长度比对延时的影响

由于芯片密度的约束, 缓冲器不能任意放置, 考虑缓冲器插入的备选位置, 将得到的每一级分段线长均匀分为 k 份。如图 6 所示, 本文算法中, $k=5$, 实际上更大的 k 值对延时提高没有太大的贡献, 而且会大大增加计算量。

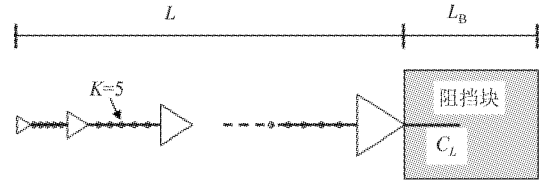


图 6 缓冲器插入位置

3 算法实现

非均匀分段和非统一缓冲器插入算法解决了大阻挡块内部不能插入缓冲器所致的大负载驱动问题。算法首先判断 L_B 内部负载的大小, 将 C_L 同经验阈值 K_1 进行比较, 当阻挡块小于 K_1 时, 采用传统的分段和缓冲器插入算法进行计算; 当阻挡块大于 K_1 时, 采用本文算法进行缓冲器插入进行计算; 第二步利用梯度搜索算法, 求出最优的缓冲器宽度比 β , 线段长度比 ω 和分段个数 N ; 第三步根据 β , ω 和 N 求出各个缓冲器插入点的位置 P_i , 相应的待插入缓冲器的宽度, 以及每一级分段长度 W_i ; 考虑到在 P_i 上可能无法插入缓冲器, 因此进一步将 W_i 均匀划分为 K_2 (一般取 K_2 等于 5) 个子线段, 并计算出 P_i 的备选位置 CP_{ij} ; 第五步在最优位置上插入相应

宽度的缓冲器,如果在该最优位置上没有插入空间,则在备选位置上插入缓冲器;第六步返回插入缓冲器后的线网。图 7 是该算法的伪代码。

```

非均匀分段和非统一缓冲器插入算法(non-uniform wire segmenting and buffer insertion/sizing algorithm),简称 NWSBI
输入:一个经过阻挡块为  $L_B$  的两端线网 NET_P
缓冲器库 BUFFER
输出:插入缓冲器的线网 NET_R
1. if (  $C_L < K_1$  )
    use uniform buffer insert algorithm;
else
    use NWSBI algorithm;
2. compute the stage ratio of the wire length  $\omega$ , the stage ratio of buffer size  $\beta$ , and the number of wire segments  $N$ ;
3. compute the width of buffer  $B_i$ , the position  $P_i$  for buffer  $B_i$  to be inserted and the length of each wire segment  $W_i$ ;
4. segment each  $W_i$  into  $K_2$  shorter wire segments with equal length and compute the candidate buffer insertion position  $CP_{ij}$ ;
5. for  $i$  from 1 to  $N-1$ 
    If ( $B_i$  can be inserted on  $P_i$ )
        insert  $B_i$  on  $P_i$ 
    else
        for  $j$  from 1 to  $K_2$ 
            if ( $B_i$  can be inserted on  $CP_{ij}$ )
                insert  $B_i$  on  $CP_{ij}$ 
6. return NET_R;

```

图 7 NWSBI 程序

4 实验结果

本文算法用 C++ 实现,机器为 1.4Ghz、512M 内

存。我们的试验测试线网来自中科院自动化所国家专用集成电路中心设计的一款 SOC。试验选取了具有代表性的 15 个驱动大负载的长线网,线网长度在 $5000\mu\text{m} \sim 10200\mu\text{m}$ 之间,负载电容在 5pF 到 20pF 之间。连线表面电阻为 $0.068\Omega/\mu\text{m}$,连线单位面积电容和侧边电容分别为 $0.06\text{fF}/\mu\text{m}^2$ 和 $0.064\text{fF}/\mu\text{m}^2$ 。缓冲器库中包括 20 个缓冲器,缓冲器输入电容在 23.4fF ~ 468fF 之间,缓冲器输出电阻在 $9\Omega \sim 180\Omega$ 之间。表 2 将本文算法与传统分段算法在分段点数目和延时两个方面进行了比较。可见,相比传统算法,本文描述的算法显著地降低了分段点数目,从而减少了运算时间并节约了芯片面积和功耗。缓冲器插入后的试验结果表明,本算法给出的延时比传统算法的延时降低了大约 1 倍。

本文描述的算法由于在对线网分段的过程中同时确定了缓冲器尺寸,所以在进行缓冲器插入时,其运行时间和分段点的数目成正比。而传统的均匀分段算法得到的分段点中不包含缓冲器尺寸信息,需要通过动态规划算法进行缓冲器库的搜索,其运行时间同分段点数目和缓冲器数目的平方成正比。同时,非均匀分段方法随着缓冲器驱动能力的不断增加,逐步增大缓冲器的间距,一方面保证了最优缓冲器插入解,同时又节省了分段点数目。图 8(a)为本文算法和传统算法在相同分段点数目下,缓冲器插入算法的运行时间的比较。图 8(b)为本文算法和传统算法在获得相同缓冲器插入解(延时相同)的情况下,分段点的数目比较。可以发现,本文描述的算法在驱动大负载时大大降低了分段点的数目并节省了设计时间。

表 2 缓冲器插入的级数和延时

线网	$L(\mu\text{m})$	$C_L(\text{pf})$	本算法分段点数目	传统算法分段点数目	本算法延时(ps)	传统算法延时(ps)
net1	10125	20.3	40	307	332	818
net2	10038	11.2	35	164	364	770
net3	9902	4.9	35	65	318	692
net4	9119	20.1	35	305	329	767
net5	9035	9.7	35	141	287	685
net6	8903	5.1	35	69	318	650
net7	8112	19.6	35	299	326	717
net8	8010	10.3	30	152	363	679
net9	8035	4.7	30	64	318	610
net10	6967	20.8	30	281	333	670
net11	7000	9.9	30	147	287	610
net12	7013	5.2	25	73	280	573
net13	5947	10.2	25	153	310	529
net14	6130	5.1	25	72	243	496
net15	5082	4.9	20	70	241	447

随着芯片上缓冲器插入数量的增加,在芯片物理设计互连驱动的布图布局阶段,必须考虑缓冲器块的规划,该算法可以用来评估不同布局模型的时序代价。在布图布局阶段的时序分析中,不需要实际插入缓冲器,嵌入该算法也可以给出有效的时序分析解。芯片设计中大模块的摆放使得理想的缓冲器插入解不能实现,如果不考虑大模块对时序优化的影响,采用传统模型进行计算,必然会给出过于保守的估计,影响设计的时序收敛。由本文的实验结果表明,该算法给出的长线网驱动大负载的时序优化解对于设计规划和时序分析是充分和有效的。

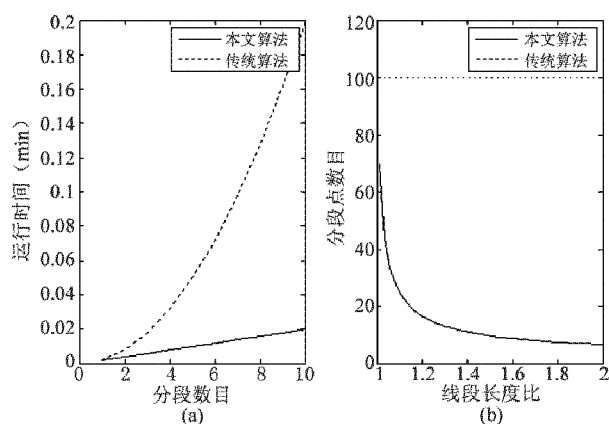


图8 缓冲器插入时间比较(a)和分段点数目比较(b)

5 结论

SOC 芯片设计技术的不断发展,促使 IP 技术和层次化设计技术被广泛采用,这简化了系统设计,但给缓冲器插入带来了很大的困难,因为其导致了芯片上不能随便插入缓冲器的块越来越多,即长导线驱动大负载的线网增多。本文提出一种“非均匀分段和非统一缓冲器插入”算法,解决了线网经过大阻挡块时获得最优延时并降低计算量的问题。该算法采用尺寸逐级增大的缓冲器链来驱动大负载,同时通过调节缓冲器间距比来达到延时最优。实验表明,该算法可以很好地解决线网经过较大阻挡块时驱动大负载的问题,在布图和布局阶段嵌入该算法,可以给出更为准确的时序优化解,同时节省布线资源,降低功耗。

参考文献

[1] Rabaey J M. Digital Integrated Circuits: A Design Perspective. 2nd edition. New Jersey: Prentice Hall/Pearson Education, 2003. 446-464

[2] Osler P J. Placement driven synthesis case studies on two sets of two chips: hierarchical and flat. In: Proceedings of the international symposium on Physical design, Phoenix, Arizona, USA, 2004. 190-197

[3] Cong J, Kong T, Pan D Z. Buffer block planning for interconnect planning and prediction. *IEEE Trans on VLSI Systems*, 2001, 9(6):929-937

[4] Dragan F F, Kahng A B, Mandoiu I, et al. Provably good global buffering using an available buffer block plan. In: Proceedings of the IEEE/ACM Conference on Computer-Aided Design, San Jose, California, USA, 2000. 104-109

[5] Alpert C J, Devgan A. Wire segmenting for improved buffer insertion. In: Proceedings of the 34th Automation Conference, Anaheim, California, USA, 1997. 588-593

[6] Van Ginneken L P P P. Buffer placement in distributed RC-tree networks for minimal Elmore delay. In: Proceedings of the IEEE International Symposium on Circuits and Systems, New Orleans, Louisiana, USA, 1990. 865-868

[7] Chu C C N, Wong D F. Closed form solutions to simultaneous buffer insertion/sizing and wire sizing. *ACM Trans on Design Automation of Electronic Systems*, 2001, 6(3):343-371

[8] Chu C C N, Wong D F. A quadratic programming approach to simultaneous buffer insertion and wire sizing. *IEEE Trans on Computer-aided Design of Integrated Circuits and Systems*, 1999, 18 (6):787-798

[9] Chang N C Y, Chang Y W, Jiang I H R. Formulae for performance optimization and their applications to interconnect-driven floorplanning. In: Proceedings of the International Symposium on Quality Electronic Design, San Jose, California, USA, 2002. 523-528

[10] Kannan L N, Suaris P R, Fang H G. A methodology and algorithms for post-placement delay optimization. In: Proceedings of the 31st annual conference on Design automation, New York, NY, USA, 1994. 327-332

[11] Dhar S, Franklin M A. Optimum buffer circuits for driving long uniform lines. *IEEE Journal of Solid-State Circuits*, 1991, 26(1): 32-40

[12] Banerjee K, Mehrotra A. A power-optimal repeater insertion methodology for global interconnects in nanometer designs. *IEEE Transactions on Electron Devices*, 2002, 49(11):2001-2007

[13] Liu X, Peng Y, Papaefthymiou M C. Practical repeater insertion for low power: what repeater library do we need? *IEEE Transactions on Computer-Aided Design of Integrated Circuit and Systems*, 2006, 25 (5): 917-924

[14] Elmore W C. The transient response of damped linear networks with particular regard to wide band amplifiers. *Journal of Applied Physics*, 1948, 19(1): 55-63

Simultaneous non-uniform wire segmenting and buffer insertion

Ma Hong, Ho Chingyen^{*}, Peng Silong

(National ASIC Design Engineering Center, Institute of Automation, Chinese Academy of Sciences, Beijing 100080)

(* Takumi Technology Corp., Santa Clara, CA 94085)

Abstract

The paper proposes a non-uniform wire segmenting and buffer insertion algorithm, which can effectively solve the problem of driving long wires with heavy load. The algorithm considers the resistance effect and the capacitance effect of the interconnects, adopts a geometrically progressive buffer chain to drive the large load, and computes the delay using the adjustable wire segments at the same time. It can yield the accurate prediction of the delay of buffer insertion while considering the impact of blockages on the layout. The experimental results show that this algorithm can optimally solve the large-load-driving problem and can be efficiently adopted in the floorplan and placement stage which can guarantee timing closure and design convergence while saving the routing resources.

Key words: non-uniform wire segmenting, buffer chain, capacitance effect, gradient method, Elmore delay