

文章编号:1001-2486(2001)02-0071-04

一种性能驱动的时序规划方法*

郭阳,李思昆,杨强

(国防科技大学计算机学院,湖南长沙 410073)

摘要:在超级计算机体系结构设计阶段,有必要进行时序规划,以尽早判断设计方案和性能指标的物理可实现性。提出一种性能驱动的时序规划方法,采用适于组件性能描述的时序性能模型,并按照线网权重分配路径延时余量,算法具有效率高、延时分配合理的优点。

关键词:数字系统;时序规划;并行设计;性能模型;超级计算机

中图分类号: TN391.72 **文献标识码:** A

An Algorithm for Performance-driven Timing-planning

GUO Yang, LI Si-kun, YANG Qiang

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: In the architecture design stage of super computer, it is indispensable to perform timing planning to judge the physical realizability of architecture alternative and performance metrics as early as possible. This paper presents a performance-driven timing-planning method. A timing performance model which is suitable to describe component performance is adopted, and the path delay slack is distributed according to net weight. The algorithm has advantages of high efficiency and rational delay distribution.

Key words: digital system; timing-planning; concurrent design; performance model; super-computer

在超级计算机体系结构设计阶段,设计者一般根据性能指标及设计经验提出一些约束来驱动物理设计^[1]。采用传统设计方法时,只有在物理设计后期才能确定约束是否合理,性能指标能否达到。若发现约束不能满足,则需要重新调整性能指标或体系结构方案,导致设计周期延长。因此,有必要在设计早期进行设计规划,以规划出的约束驱动和控制下游物理设计工具,尽早判断体系结构设计方案和性能指标的物理可实现性,及时改进设计^[2]。

目前,设计规划技术的研究主要分布在两个层次:第一是集成电路层,这方面的研究较多^[3,4],且已有多产品进入市场;第二是整机系统层,这一层次把集成电路、印制电路板、底板看作是组装和互连的单元(组件),电子设备整机是由组件单元在机柜中通过机械组装和电气互连构成的机电一体化的三维物理实体。目前尚未发现支持该层次设计规划的成熟产品。

1 性能驱动的时序规划方法

1.1 性能驱动的时序规划

时序规划的基本方法包括^[3,4](1)在逻辑设计阶段估计互连长度,布图完成后,利用实际线长,判断系统能否满足性能约束;(2)在物理设计过程早期引入时序约束,随布局的进行,使用逐步精确的时序信息;(3)在布图构造过程中,在时序分析和布局算法间来回地传递时序和布图数据;(4)将性能要求作为直接的约束,找到关键路径,在布局过程中赋予较高的权。

性能驱动的时序规划方法以性能模型、性能指标和性能约束作为规划的依据,首先建立复杂电子系统的性能模型,并用性能描述语言(PML)描述;其次,将性能描述与功能描述相结合,产生面向特定设计的性能模型;然后,对该模型进行分析、评价、约束满足求解,规划出面向特定设计的性能约束;最后,将

* 收稿日期:2000-10-27
基金项目:国家863基金资助项目(863-511-42-01)
作者简介:郭阳(1971-),男,助理研究员,博士。

性能约束转化为布局可以接受的电气约束,驱动三维布局工具或其它物理设计工具。性能驱动的时序规划的任务主要包括两方面:一是根据性能指标和性能模型的属性值、属性评价规则、约束规则规划出路径约束;二是将路径约束分配到线网,为三维组装布局产生延时约束文件。

1.2 时序性能模型

根据超级计算机体系结构设计的特点和需求,我们提出时序性能模型,将系统的所有设计对象归结为三类:模块、载体和端口。模块是分层次的,表示组成系统的部件或单元块。模块可以有端口,并可以包含其它的模块和载体。载体用于连接端口,一般用于表示连线、总线以及其它表示信息流动的部件。端口表示模块和载体的输入与输出。模块、载体和端口都可带有属性、属性评价规则及约束规则。时序性能模型中所描述的属性主要与时序特征相关。时序性能描述模型如图1所示。

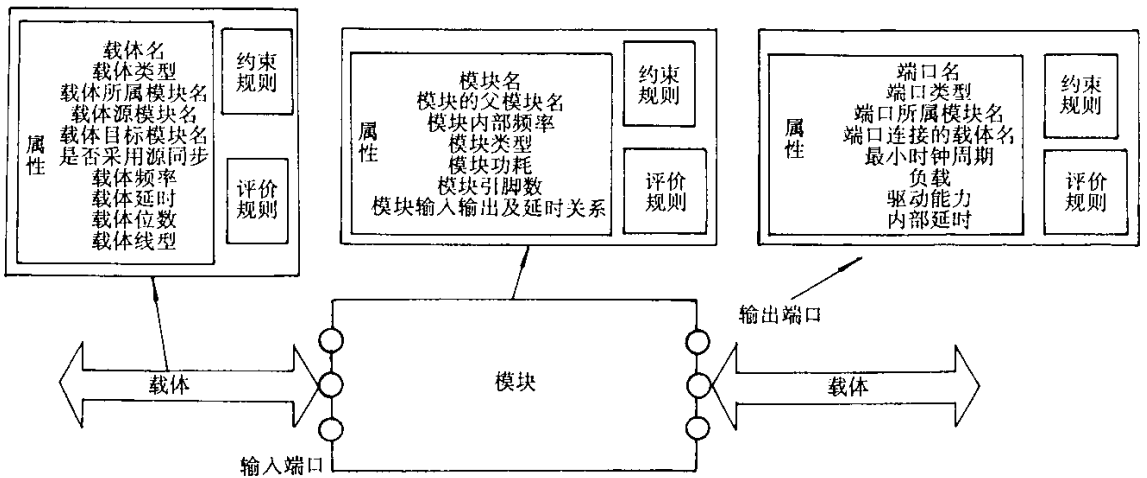


图1 时序性能描述模型

Fig.1 The timing performance model

时序性能模型是进行性能驱动的时序规划的基础。设计人员通过我们开发的 PML 语言和交互建模环境,为组成设计的各模块建立时序性能模型、时序规划时,从时序性能模型描述文件中读取有关模块、端口和载体的属性信息。

1.3 路径约束的产生

根据时序性能模型的描述和用户提出的要求,可计算出路径时序约束。假设 t_1 为源时序模块内输出寄存器站到输出端口的时间, t_2 为时序模块间路径的延时, t_3 为目标时序模块输入端口到模块内输入寄存器站的延时, f 为设计要求的时钟频率。从而有 $t_1 + t_2 + t_3 \leq 1/f$, 即路径时序约束 $t_2 \leq 1/f - t_1 - t_3$ 。同时 t_2 也应小于时序性能模型中由用户直接指定的路径约束 Ψ , 因此 $t_2 = \min\{1/f - t_1 - t_3, \Psi\}$ 。

产生路径约束后,再利用性能驱动的按线网权重分配延时余量的零余量算法,就可以产生线网约束。

2 性能驱动的按线网权重分配延时余量的零余量算法

如果系统的设计电路中的每条短路径大于延时下限,并且每条长路径小于延时长限,则设计不存在时序问题。实际设计中,长路径问题较难克服,最坏情况下需要重新设计电路,它对性能的影响也最大,因为系统的实际速度往往取决于最长路径上的延时。因此,算法主要讨论长路径的时序规划,相应地,主要考虑各线网的时序约束上限问题。时序规划的目标就是在实际布局前将不考虑线网延时得出的路径延时余量,合理地分配给各段线网,以保证系统在考虑线网延时后仍是安全的。

2.1 基本概念与原理

定义1 系统的设计电路的时序关系用有向图 $D_T = (P, A)$ 表示。 P 为顶点集,与模块的端口一一

对应 ; A 为弧集 ,是顶点间互连关系的集合 , $\forall (p_i, p_j) \in A$,弧的权 $d(p_i, p_j)$ 表示端口之间的信号传输延时值 ,弧的方向表示信号的流向。 $A = A^I \cup A^E$,其中 A^I 表示模块内的信号流 , A^E 表示线网的信号流 , $(p_i, p_j) \in A^E$ 称为线网 ,与载体 $c_i \in C$ 相对应。

定义 2 定义在每个端口 - 载体对上的映射 $\pi : P \times C \rightarrow \{1, 0, -1\}$ 为

$$\pi(p, c) = \begin{cases} -1 & \text{如果 } c \text{ 是 } p \text{ 的输入} \\ 1 & \text{如果 } c \text{ 是 } p \text{ 的输出} \\ 0 & \text{否则} \end{cases}$$

定义 3 定义在一个模块内部每个输入端口 - 输出端口对上的映射 $\sigma : P \times P \rightarrow \{1, 0, -1\}$ 为

$$\sigma(p_i, p_j) = \begin{cases} -1 & \text{如果 } p_i \text{ 是 } p_j \text{ 的输出} \\ 1 & \text{如果 } p_i \text{ 是 } p_j \text{ 的输入} \\ 0 & \text{否则} \end{cases}$$

定义 4 假设 $d(p)$ 表示端口 $p(p \in P)$ 的输出载体 ,则 p 的扇入端口集定义为 $\pi^-(p) = \{p' \mid \pi(p, d(p')) = -1\} \cup \{p' \mid \sigma(p, p') = -1\}$;扇出端口集定义为 $\pi^+(p) = \{p' \mid \pi(p', d(p)) = -1\} \cup \{p' \mid \sigma(p', p) = -1\}$ 。

定义 5 路径 ρ 定义为端口序列 $\langle p_1, \dots, p_k \rangle$, $p_1 \in I$ (原始输入) , $p_k \in G$ (原始输出) , $p_i \in \pi^+(p_{i-1})$, $i = 2, \dots, k$ 。

定义 6 设目标网络的延时限为 RT ,端口 $p \in P$ 的需要到达时间 $t_r(p)$ 定义如下 :

- (1) 若端口 p 只扇出到原始输出端 G ,则 $t_r(p) = RT$;
- (2) 若端口 p 扇出到 $\pi^+(p)$,则 $t_r(p) = \min_{z \in \pi^+(p)} t_r(z) - d(pz)$;

定义 7 给定电路原始输入端的信号到达时间 AT ,端口 $p \in P$ 的实际到达时间 $t_a(p)$ 定义如下 :

- (1) 若原始输入 I 连接到端口 p ,则 $t_a(p) = AT$;
- (2) 若 $z \in \pi^-(p)$ 为端口 p 扇入单元 ,则 $t_a(p) = d(zp) + \max_{z \in \pi^-(p)} t_a(z)$;

对每个系统输出 $g \in G$,可以计算延时余量 $s(g) = t_r(g) - t_a(g)$ 。

定义 8 如果对每个输出 $g \in G$, $s(g) \geq 0$,则称系统是安全的。

相应地 ,每个端口 p 和每条路径 ρ 都可以计算出延时余量 $s(p) = t_r(p) - t_a(p)$, $s(\rho) = t_r(p_k) - t_a(p_1) - \sum_{1 \leq i \leq k} d(p_i p_{i+1})$ 。

定理 1 对设计电路中的任意路径 $\rho = \langle p_1, \dots, p_k \rangle$, $s(p_j) \leq s(\rho)$, $1 \leq j \leq k$ 。

证明 :由定义 6 和定义 7 可知

$$t_r(p_j) \leq t_r(p_k) - \sum_{i \geq j} d(p_i p_{i+1}) \text{ 且 } t_a(p_j) \geq t_a(p_1) + \sum_{i < j} d(p_i p_{i+1})$$

因此 , $t_r(p_j) - t_a(p_j) \leq t_r(p_k) - t_a(p_1) + \sum_{1 \leq i \leq k} d(p_i p_{i+1})$,即 $s(p_j) \leq s(\rho)$ 。

定理 2 对任意端口 $p \in P$,存在一条路径 ρ_x ,使得 $s(p) = s(\rho_x)$ 。

证明 :

考虑路径 $\rho_x = \langle p_1, \dots, p_j = p, \dots, p_k \rangle$,使得

(1) 对于 $i \geq j$, $t_r(p_i) = t_r(p_{i+1}) - d(p_i p_{i+1})$;(2) 对于 $i \leq j$, $t_a(p_i) = t_a(p_{i-1}) + d(p_{i-1} p_i)$ 。

由 $t_r(p)$ 和 $t_a(p)$ 的定义可知 ,这样一条路径是肯定存在的。

由此可推出 $t_r(p) = t_r(p_k) - \sum_{i \geq j} d(p_i p_{i+1})$,且 $t_a(p) = t_a(p_1) + \sum_{i \leq j} d(p_{i-1} p_i)$ 。

因此 , $t_r(p) - t_a(p) = t_r(p_k) - t_a(p_1) - \sum_{1 < j \leq k} d(p_{i-1} p_i)$,即 $s(p) = s(\rho_x)$ 。

推论 给定一个安全电路 , $p_i, p_j \in P$,线网 $c_i = (p_i, p_j) \in A^E$,一个线网延时赋值 $\delta(p_i p_{i+1})$ 是安全的 ,当且仅当 $\delta(p_i p_{i+1}) \leq s(p_i)$ 。

定理 3 给定一个安全电路 ,一组线网延时赋值 $\delta(X)$ 是安全的 , $X \subseteq A^E$,当且仅当

$$\sum_{(p_i, p_{i+1}) \in X} \alpha(p_i, p_{i+1}) \leq \min_{(p_i, p_{i+1}) \in X} \alpha(p_i).$$

2.2 算法实现思路

(1) 按线网权重分配延时余量

线网 c 的权重定义为 $W(c) = LF_c^* AcL_c$, 其中 AcL_c 为端口的外部负载电容, LF 为负载因子, LF_c 和 AcL_c 都可通过时序性能模型得到。路径 ρ 中线网 c 的延时分配值可按下式计算, $\alpha(c) = W(c) \times s(\rho) / W(\rho)$ 。假设 ψ_c 为通过线网 c 的所有路径, 为使线网的时序界限满足通过线网的所有路径的时序界限要求, 线网 c 的延时赋值应取最小值 $u_c = \min_{\rho \in \psi_c} \alpha(c)$ 。

定义 9 路径的分数定义为路径余量 $s(\rho)$ 与权值 $u(\rho)$ 的比, $\alpha(\rho) = s(\rho) / u(\rho)$ 。

定义 10 电路图 D_T 中的路径 ρ 覆盖线网 c , 当且仅当该路径是通过线网 c 的所有路径中分数最小的。

(2) 通过零余量路径集和零余量线网集加速算法收敛

具体实现时, 设置一个零余量线网集 $Zero-net$, 迭代过程中余量为 0 的路径中的所有线网都加入 $Zero-net$ 中。计算路径权重时, 不考虑路径中的零余量线网。同时, 设置一个零余量路径集 $Zero-path$, 余量为 0 的路径加入零余量路径集中, 以后迭代时无需再考虑。

(3) 算法步骤

根据上述思想, 我们实现了约束限制下按线网权重分配延时余量的零余量算法, 其主要步骤为:

I. 从网表文件和时序性能模型描述文件读取初始数据信息, 设置变量初值;

II. 建立设计的电路图模型, 找出所有路径, 计算路径权重;

III. 当零余量路径数小于路径总数且零余量线网数小于线网总数, 循环

(1) 计算每条路径的延时余量和权重;

(2) 调整零余量路径集和零余量线网集;

(3) 找到每条线网的覆盖并计算延时赋值;

(4) 根据性能约束调整延时赋值和零余量线网集。

3 结论

我们将性能驱动的组件级时序规划方法应用于某超级计算机路由板的设计, 该路由板由两块自行设计的芯片 (NR 和 NI) 和一些组合逻辑控制电路组成, NR 和 NI 都包含近百个输入、输出管脚。系统体系结构设计时, NR、NI 并未完成芯片设计, 只知道逻辑功能、基本的互连关系和初步估计的一些延时条件。我们按照时序性能模型对路由板、NI 和 NR 进行建模, 描述时序性能要求和时序性能参数, 并计算得出组件级路径约束和线网延时约束, 用约束驱动布局, 大大减少了迭代次数。

性能驱动的时序规划方法应用于由多块 PCB 构成的系统, 也取得了较好的结果。

参考文献:

- [1] Kai Hwang, Zhiwei Xu. Scalable Parallel Computing[M]. New York: McGraw Hill Companies, 1997.
- [2] 李思昆, 郭阳, 杨强. 基于虚拟原型的数字系统并行设计方法研究[J]. 软件学报, 1998, 9(suppl): 89-92.
- [3] 洪先龙, 严晓浪, 乔长阁. 超大规模集成电路版图理论与算法[M]. 北京: 科学出版社, 1998.
- [4] Narayanan V, Lapotin D. PEPPER - A Timing Driven Early Floorplanner[C]. IEEE International Conference on Computer - Aided - Design[C], San Jose, 1995: 230-235.

