复旦大学

硕士学位论文

FPGA布线算法的研究

姓名：王怡

申请学位级别：硕士

专业：微电子学与固体电子学

指导教师：童家榕

20070520

摘要

现场可编程门阵列（FPGA）是一种可实现多层次逻辑器件。基于 SRAM 的 FPGA 结构由逻辑单元阵列来实现所需要的逻辑函数。FPGA 中，互连线资源是预先定制的，这些资源是由各种长度的可分割金属线，缓冲器和 MOS 管实现的，所以相对于 ASIC 中互连线所占用的面积更大。为了节省芯片面积，一般都采用单个 MOS 晶体管来连接逻辑资源。MOS 晶体管的导通电阻可以达到千欧量级，可分割金属线段的电阻相对于 MOS 管来说是可以忽略的，然而它和地之间的电容达到了 0.1pf [1]。为了评估 FPGA 的性能，用 HSPICE 仿真模型虽可以获得非常精确的结果，但是基于此模型需要花费太多的时间。这在基于时序驱动的工艺映射和布局布线以及静态时序分析中都是不可行的。于是，非常迫切地需要一种快速而精确的模型。

FPGA 中连接盒、开关盒都是由 MOS 管组成的。FPGA 中的时延很大部分取决于互连，而 MOS 传输晶体管在互连中又占了很大的比重。所以对于 MOS 管的建模对 FPGA 时延估算有很大的影响意义。对于 MOS 管，Muhammad [15]采用导通电阻来代替 MOS 管，然后用 Elmore [3]时延和 Rubinstein [4]时延模型估算互连时延。Elmore 时延用电路的一阶矩来近似信号到达最大值 50%时的时延，而 Rubinstein 也是通过计算电路的一阶矩估算时延的上下边界来估算电路的时延，然而他们都是用来计算 RC 互连时延。传输管是非线性器件，所以没有一个固定的电阻，这就造成了 Elmore 时延和 Rubinstein 时延模型的过于近似的估算，对整体评估 FPGA 的性能带来负面因素。

本论文提出快速而精确的现场可编程门阵列 FPGA 中的互连资源 MOS 传输管时延模型。首先从阶跃信号推导出适合 50%时延的等效电阻模型，然后在斜坡输入的时候，给出斜坡输入时的时延模型，并且给出等效电容的计算方法。结果验证了我们精确的时延模型在时间上的开销少的性能。

在岛型 FPGA 中，单个传输管能够被用来作为互连线和互连线之间的连接，或者互连线和管脚之间的连接，如 VPR 把互连线和管脚作为布线资源，管脚只能单独作为输入或者输出管脚，以致于它们不是一个线网的起点就是线网的终点。而这恰恰忽略了管脚实际在物理上可以作为互连线来使用的情况（VPR 认为 dogleg 现象本身对性能提高不多）。本论文通过对 dogleg 现象进行了探索，并验证了在使用 SUBSET 开关盒的情况下，dogleg 能提高 FPGA 的布通率。

关键词：FPGA、时序驱动、延时模型、斜坡、布通率

中图分类号：**TN432**

Abstract

Field-Programmable gate array (FPGA) can implement hierarchical logic. The FPGA structure based on SRAM implements desired functions by logic arrays. The interconnection resource is pre-fabricated in FPGAs, comprised of metal wire segments, buffers and MOS transistors, so it takes more areas compared with ASICs. In order to save chip area, it is common to use individual MOS transistor to connect logic resources. The "turn-on" resistance of MOS transistor is up to the order of $K\Omega$. Thus the resistance of segmented wire is negligible comparatively, however, the wire capacitance with the ground reaches 0.1pF[1]. If the HSPICE simulation method is used to evaluate the performance of a FPGA design, the result is very precise, but too much time is needed, which is impractical in time-driven placement, routing or static timing analysis. Hence, a quick and precise model is in demand. In FPGAs, the connection boxes and switch boxes are comprised of MOS transistors, which is of great importance in interconnections that in a large extent determines the delays in FPGAs. Therefore modeling MOS transistor delay will greatly contribute to the delay estimation in FPGAs. As for MOS transistor, Muhammad[2] first represents the pass transistor with a switch-on resistor and then uses Elmore[2] model and Rubinstein[3] model to estimate interconnection delay. "Elmore delay" is the delay time for the first-order approximation of the signal reching 50% of its maximum value and Rubinstein found out the circuit delay by estimating its upper and lower bounds. However, they are just available for calculating RC interconnection delays and thus not suitable in FPGAs ,because pass transistors are non-linear devices with uncertain resistance value.

This thesis proposes FPGA interconnection pass transistor delay model. First, the equivalent resistance delay model is presented based on 50% timing delay for the pulse input. The equivalent capacitance delay model is thereafter proposed for the slope input. The corresponding effective capacitance delay calculation method is also given. The experimental results show the efficiency and accuracy of the proposed delay model for FPGA interconnection.

In island-style FPGA, single pass transistor can be used as connection between wire and wire, or between wire and pin. VPR tool considers wire and pin as the routing resources, and considers pin either as an ipin or an opin that is either the source or sink of a net. It does not consider that pin have a property as wire, which can be used

to conduct signal or transit the signal to other routing resource. This thesis investigates the dogleg phenomenon, and experimental results show that under SUBSET switch, dogleg can enhance the routability of FPGA by 11%.

论文独创性声明

本论文是我个人在导师指导下进行的研究工作及取得的研究成果。论文中除了特别加以标注和致谢的地方外，不包含其他人或其它机构已经发表或撰写过的研究成果。其他同志对本研究的启发和所做的贡献均已在论文中作了明确的声明并表示了谢意。

作者签名：王怡　　日期：2007.6.19

论文使用授权声明

本人完全了解复旦大学有关保留、使用学位论文的规定，即：学校有权保留送交论文的复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。保密的论文在解密后遵守此规定。

作者签名：王怡　　导师签名：[illegible]　　日期：2007.6.19

第一章 引言

1.1 可编程逻辑器件背景知识

当今世界正在掀起一股数字化浪潮，其中起至关重要作用的是数字集成电路，尽管 VLSI 设计自动化技术的飞速发展，然而由于制造芯片的周期很长，所以 ASIC 芯片推向市场的时间仍然很长，这就催生了一个新的市场，可编程逻辑器件是近年来才发展起来的一种新型集成电路，由用户根据自己的需要对其进行编程，确定芯片的功能。最早的可编程逻辑器件是 1970 年出现的 PROM，70 年代中期出现了可编程逻辑阵列（PLA－Programmable Logic Array）器件，由于编程复杂，开发起来有一定难度，因而没有得到广泛的应用。70 年代末出现了早期的基于与或阵列的可编程逻辑器件（简称 PLD）。这种可编程器件的结构简单，虽然具有高速的性能，但是因为规模小，难以实现复杂的逻辑功能。20 世纪 80 年代中期，Altera[41]公司推出了一种新型的可编程逻辑器件（EPLD－Erasable Programmable Logic Device）,它采用了 CMOS 和 UVEPROM 工艺制作，可以用紫外线擦除并重复编程。集成度比 PLD 高的多，设计也更加灵活，但内部的互连能力比较弱。1985 年 Xilinx[40]公司推出了现场可编程门阵列（FPGA－Field Programmable Gate Array）器件，它是一种新型的高密度 PLD，采用了 CMOS－SRAM 工艺制作，内部由许多独立的可编程逻辑模块组成，逻辑快之间可以灵活地相互连接，具有密度高、编程速度快、设计灵活和可再配置设计能力等许多优点。其后，各家公司又相继推出了各种功能强劲地 FPGA 系列产品。

一般来讲，目前地大规模可编程器件按其结构特征被分为 FPGA 和 CPLD（Complex Programmable Logic Device）两类。总体结构上，FPGA 是逻辑门级的可编程，CPLD 是逻辑块级的可编程。然而，在各自的发展过程中它们相互取长补短，界限越来越模糊。

现场可编程门阵列（FPGA）是一种可由用户根据需要自行配置的高密度专用集成电路，它将定制的 VLSI 电路的单片逻辑集成优点和可编程器件设计灵活、实现方便、产品上市快捷的长处结合起来，已经成为一类标准的产品。FPGA 的基本逻辑构造单元是可编程逻辑块，按照逻辑功能块的大小不同，可将 FPGA 分为粗颗粒结构和细颗粒结构，粗颗粒的 FPGA 逻辑块规模大，功能强，然而有些块的资源不能充分利用。细颗粒的 FPGA 逻辑功能一般较小，功能块的资源可以被完全利用，缺点是通常需要大量的连线和可编程开关，因而速度相对较慢。按照互连结构不同，可将其分为分段互连型和连续互连型。分段互连型结构是 FPGA 的主流，这种 FPGA 内有不同长度的多种金属线，各金属线段之间通过开关矩阵或反熔丝编程连接这种连线结构走线灵活，布通率高，但是内部延时与布

局布线的具体处理过程有关，在设计完成前无法预测延时。连续互连型 FPGA 是利用相同长度的金属线，通常是贯穿整个芯片的长线来实现互连，连接与距离远近无关。因而布线延时是固定和可预测的。

FPGA 之所以流行，关键之处在于仅仅通过合适的编程，它就能够实现任意电路。对于其他的电路实现方法，譬如标准单元和 MPGA（掩模可编程门阵列），就要求对每一个设计都必须制造一个新的 VLSI 芯片。相对于这些定制技术，使用标准 FPGA 有两个重要的优点：降低一次性费用（NRE）和缩短上市时间。

要在 MPGA 和标准单元上实现一个电路，人们要把完整的设计结果送到硅代工厂，代工厂生产一个芯片使所设计的电路得到精确的（且唯一的）实现。生产第一块芯片的一次性费用一般在 10 到 25 万美元；这笔费用包括为设计制作光刻和掩模板的费用，以及新设计通过整个工艺流程的费用。相反，我们只要对 FPGA（一个标准的器件）编程就可以实现用户所需的功能，这样就不需要一次性费用。这使得 FPGA 成为中小量产规模电路设计最廉价的实现方法。

上市时间是 FPGA 的另一项关键优势。整个生产过程一般需要 6 至 8 周。如果生产出来的芯片中发现了问题，它们必须被扔掉，然后又得等 6 至 8 周来生产修改过的设计。相反，FPGA 编程能在数秒内完成，如果芯片在系统级测试时发现问题，通过重新编程，就可以在几分钟内被解决。因此，FPGA 的快速上市时间满足了当今生产周期变短的要求，成为其压倒性的优势。

然而，FPGA 也为可编程特性付出了代价。在 MPGAs 和标准单元中，电路是用金属线互联的，FPGA 却一定要通过可编程开关来连接电路。这些开关相对于金属线有更高的电阻，并且在互连中加入了明显的电容，从而降低了电路的速度。另外这些开关和金属连线相比占用了更多的面积，因此，实现同一个电路，FPGA 要比 MPGA 的面积大很多。在同等工艺条件下，同一个电路在 FPGA 上实现，其面积一般是在 MPGA 上实现的 10 倍，而速度大概是在 MPGA 上实现的 1/3[1]。对于高量产设计， FPGA 需要更大的芯片尺寸，会比 MPGA 昂贵，而且 FPGA 的低速度使得其在高速电路领域没有用武之地。FPGA 和 MPGA 的这些差异要求研究新型 FPGA 结构，从而弥补速度和密度上的不足。此外，由于 FPGA 市场竞争激烈，FPGA 的生产商正不停地寻找更好的结构以获得速度和密度上的优势。

1.2 FPGA 的基本硬件结构

FPGA 结构模型如图 1 所示。FPGA 结构上主要包括了可编程逻辑块（CLB—Configurable Logic Block）、输入输出单元（IOB—I/O Block）和互连资源（IR—Interconnect Resource）。当然现在的 FPGA 还包括了 RAM 块等专用的宏模块，扩展了 FPGA 的性能。常见的 FPGA 结构主要包括 3 种类型：查找表结构、多

路开关结构、多级与非门结构[1]。

(1) 查找表型 FPGA 结构

查找表型 FPGA 的可编程逻辑块是查找表，由查找表构成函数发生器，通过查找表来实现逻辑函数。查找表的物理结构是静态存储器（SRAM）。M 个输入项的逻辑函数可以由一个 2^M 位容量的 SRAM 实现，函数值存放在 SRAM 中，SRAM 的地址线起输入线的作用，地址即输入变量值，SRAM 的输出为逻辑函数值，由连线开关实现与其他功能块的连接。

(2) 多路开关型 FPGA 结构

在多路开关型 FPGA 中，可编程逻辑块是可配置的多路开关。利用多路开关的特性对多路开关的输入和选择信号进行配置，得到固定电平或输入信号上，从而实现不同的逻辑功能。

(3) 多级与非门型 FPGA 结构

采用多级与非门结构的器件是 Altera 公司的 FPGA。Altera 公司的与非门结构基于一个与－或－异或逻辑块，这种基本电路可以用一个触发器和一个多路开关来扩充。多路开关选择组合逻辑输出、寄存器输出或锁存器输出。异或门用于增强逻辑块的功能，当异或门输入端分离时，它的作用相当于或门，可以形成更大的或函数，用来实现其他算术功能。Altera 公司的 FPGA 的多级与非门结构同 PLD 的与或阵列很类似，它是以线与形式实现与逻辑的。在多级与非门结构中线与门可编程，同时起着逻辑连接和布线的作用，而在其他 FPGA 结构中，逻辑和布线是分开的。

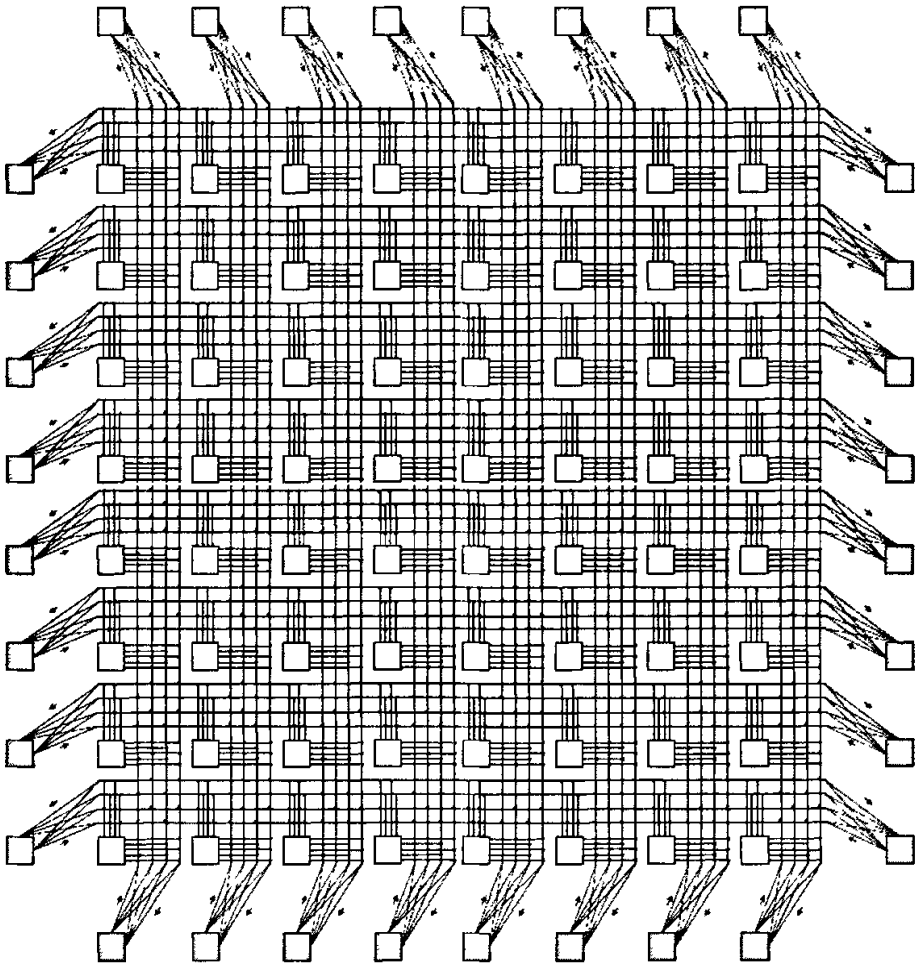

图 1 FPGA 结构模型图

1.3 可编程逻辑器件的 CAD 系统

可编程逻辑器件的设计流程[31]一般包括设计输入（包括原理图的输入、硬件描述语言输入、状态图输入等）、逻辑综合、功能模拟、划分、映射、装箱、布局布线、时序分析、位流生成、编程下载等过程。早期的 FPGA 公司都采用第三方的软件来实现设计输入和模拟工作，现在的 FPGA 设计软件是一个集成的开发环境。用户不必去进行繁琐的文件操作[2]。基本的 CAD 系统如图 2 所示：

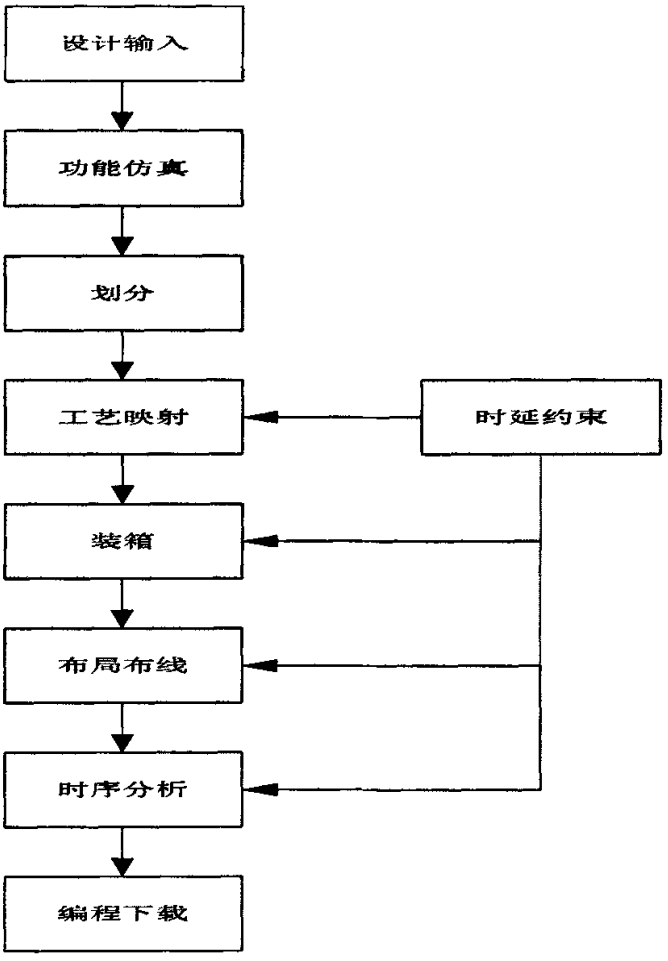

图 2 FPGA 的 CAD 系统

其中

设计输入模块：为电路设计输入模块，用户可以通过电路图，HDL（硬件描述语言）语言和状态机等输入电路设计。有些厂家的软件可以通过 EDIF 格式文件和元件库映射来完成第三方软件到可编程逻辑器件软件的衔接。元件库中有许多功能块（如加法器、乘法器、比较器、计数器、译码器等），应尽量利用这些现成的功能模块。

功能仿真模块：为了对用户设计进行仿真，检查功能的正确性。

划分模块[28]：当用户的电路规模很大时，以致于超过了单块可编程逻辑芯片的容量，这是通过划分模块对电路进行分割，形成小块的电路，这样，单块的可编程逻辑芯片能够容纳。

工艺映射模块：将电路映射倒 FPGA 芯片的具体可编程逻辑结构中。优化逻辑单元数目的映射将尽量减少使用逻辑单元的数目；优化时延特性的映射将尽量减少信号经过的逻辑单元的数目。生成节点对应文件反映用户输入设计中的节点归属

于哪个 LC 中，时序模块由此寻找时序约束文件中指示的电路节点。
时延约束模块：用户输入时序约束信息。
装箱模块[29][30]：将 LC 打包成 CLUSTER（簇），形成层级逻辑结构。
布局模块[33][34]：将各CLUSTER和IO PAD 放到适当的位置。一般情况下，由于可编程逻辑器件布线资源的有限性，布局阶段的优化目标首先是布通率，通道密度的均匀性。但是实用中还有一些其它问题要考虑。例如用户对某些路径的时延提出限制，或指定一些线网的重要程度，或要求管脚锁定。这样布局算法就要综合考虑。自然，各种约束条件和布通率之间是存在矛盾的，怎样兼顾各方面的要求又能最大程度的提高布通率是一个很值得研究的方向。
布线模块[35][36][37]：实现CLUSTER之间的连线。布线要求百分之百的布通率，否则就是布线失败。因为可编程逻辑器件布线资源的特殊性——布线资源都已确定，布线实际上只是确定哪些编程点接通而哪些不接通。由于可编程逻辑器件布线资源的有限性和多样性，其布线算法也有特殊性，与普通的布线算法有很大的不同，相应的也有许多值得研究的问题。布线完成以后，各路径的时延值才能最后确定。这时，需要进行时序验证， 以检查电路在有时延的情况下工作是否正常。如正常，进入下一步编程下载；如不正常，需仔细检查各相关点，找出问题所在，用适当的方法解决（如加入时延约束、关键线网、调整线网的重要程度、指定某些逻辑单元的位置，或人工干预， 拆除某些线网、移动某些逻辑单元，重新布线）等。这样的修改可能要反复多次。
时序分析模块[38][39]：对电路的关键路径，最大延时等做分析，以及判断电路时序是否符合用户输入的时延约束。
编程下载：当满足时序约束后，系统生成位流文件，下载到芯片中。

1.4 论文的组织结构

论文的第一章主要介绍了可编程逻辑的基本知识，第二章主要介绍了本论文的研究背景，第三章提出了一种适合在 FPGA 时驱布线中可以使用的延时模型，第四章对 dogleg 现象进行了探索，第五章实现了 FDP250K 的布线系统，第六章作为总结和展望。对将来的工作做了一个预测。

第二章 研究背景

2.1 FPGA 电路结构

FDP250K 芯片是复旦大学微电子系 ICCAD 实验室所开发的具有历史意义的 FPGA 芯片，包括了软件系统和硬件系统。

FDP250K 的可编程资源分为可编程逻辑资源、可编程互连资源和可编程 IO 单元。可编程逻辑资源是 FDP250K 芯片结构的基本构成单元，用来实现逻辑功能函数；可编程互连资源将逻辑资源连接成一个整体，完成电路功能；可编程 IO 单元实现设计电路的输入输出功能。FDP250K 总体结构和实际版图分别如图 3 和图 4 所示。

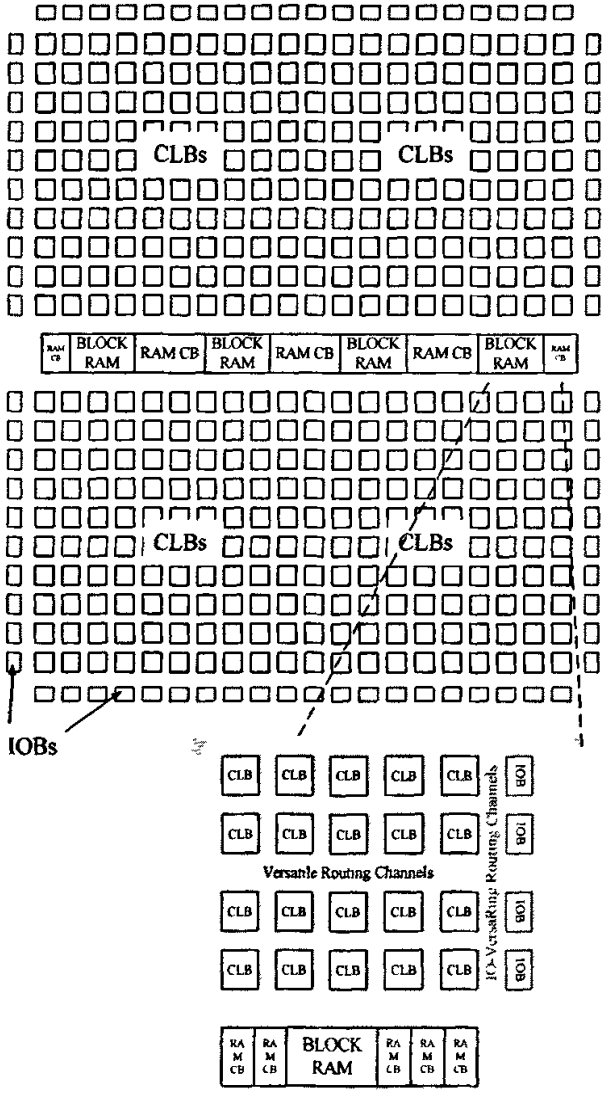

图 3 FDP250K 整体结构图

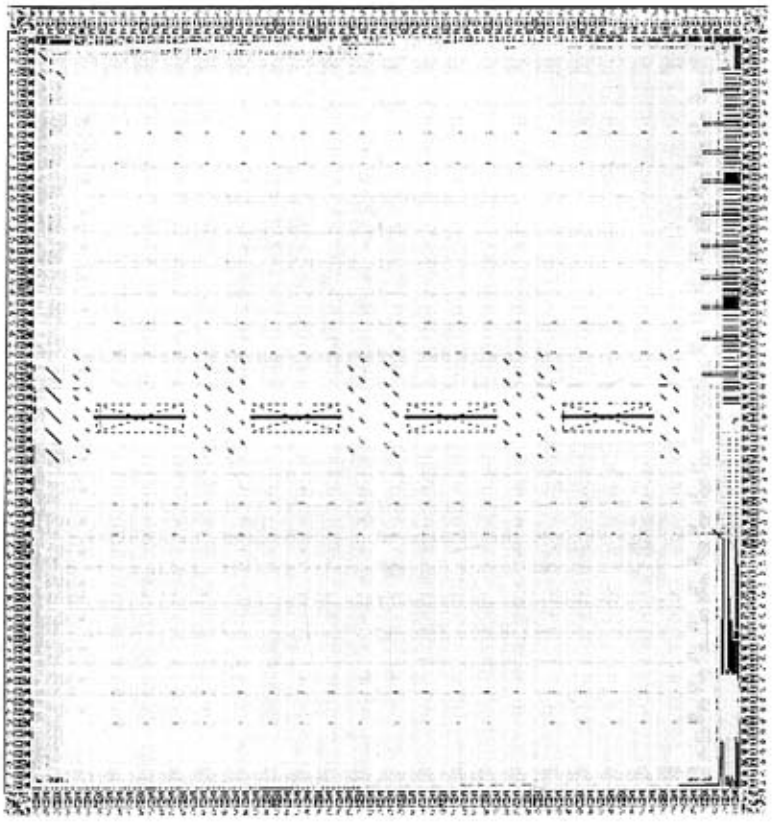

图 4 FDP250K 芯片版图

2.1.1. FPGA 逻辑资源结构

（1） 可编程逻辑单元簇（CLUSTER)

图 5 是一个可编程逻辑单元簇（CLUSTER)内部结构，它由上下两个完全相同的可编程逻辑片(SLICE)以及一个时序控制部件 SCU(Sequential Control Unit)构成。一个 SLICE 包含上下两个 LC 以及两者间的关联部件，可以完成两个独立 4 输入组合逻辑或者一个 5 输入组合逻辑。CLUSTER 内部的 4 个时序单元作为完全独立对称的单元分布在 4 个 LC 中，其可以统一配置成带有异步复位或置位以及使能的 D 触发器（DFF）或者电平锁存器（LATCH)。CLUSTER 的 SCU 处理从互连、专用时钟网络、全局复位网络送来的时序控制信号，从而为 4 个 LC 产生统一的时钟、使能、复/置位以及功能选择信号。

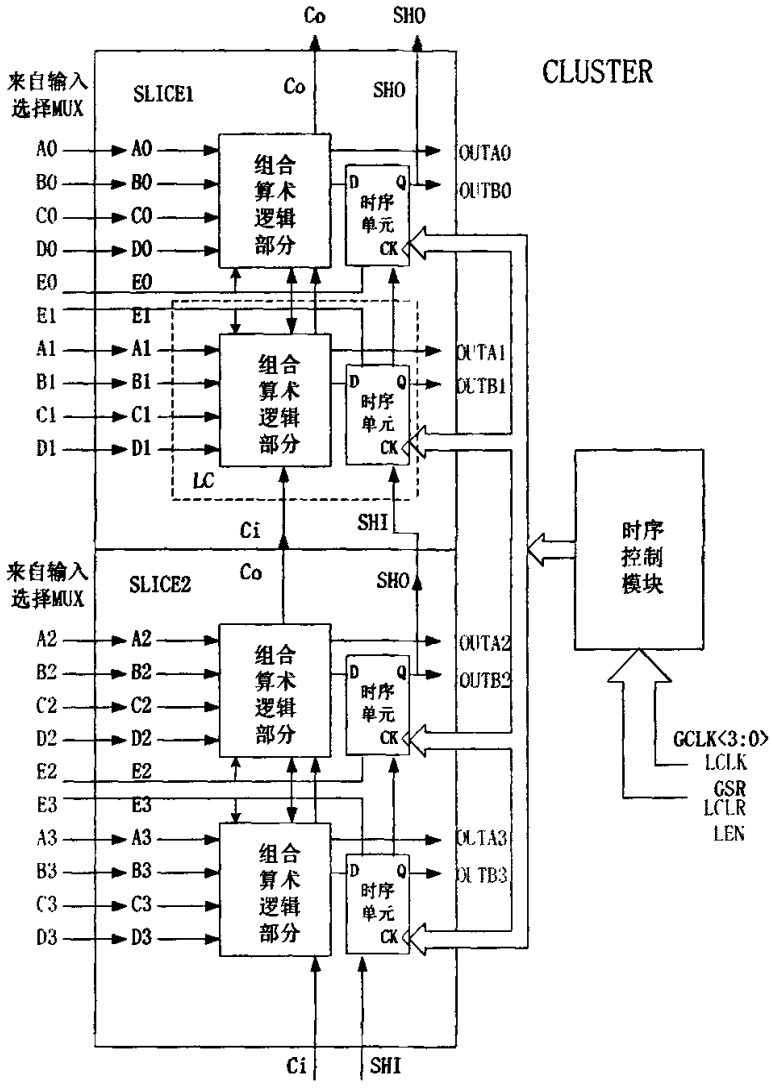

图 5 FDP250K CLUSTER 内部结构

（2） 可编程逻辑单元（LC）

FDP250K 芯片的 CLUSTER 内部 LC 基于 LUT 查询表结构。图 6 是 FDP250K 的一个 SLICE 结构（内部悬空点由编程点控制），由两个相同的 LC 以及两者结合部件组成。不同于常见的基于四输入查询表结构的 LC，FDP250K 一个 LC 是由两个具有相同输入的 3 输入 LUT、多个数据选择器、快速进位链、以及一个可编程控制时序逻辑单元构成。两个独立三输入查询表和数据选择器可完成最高 4 输入任意组合逻辑，也可以产生两个相同输入的任意 3 输入函数。快速进位单元可以结合 1 个 3 输入 LUT 实现加、减和一位乘加算术逻辑功能；时序单元可以对输入端 E 或者组合逻辑输出进行锁存，而且时序单元带有扫描链逻辑，可实现芯片内 LC 的扫描测试。

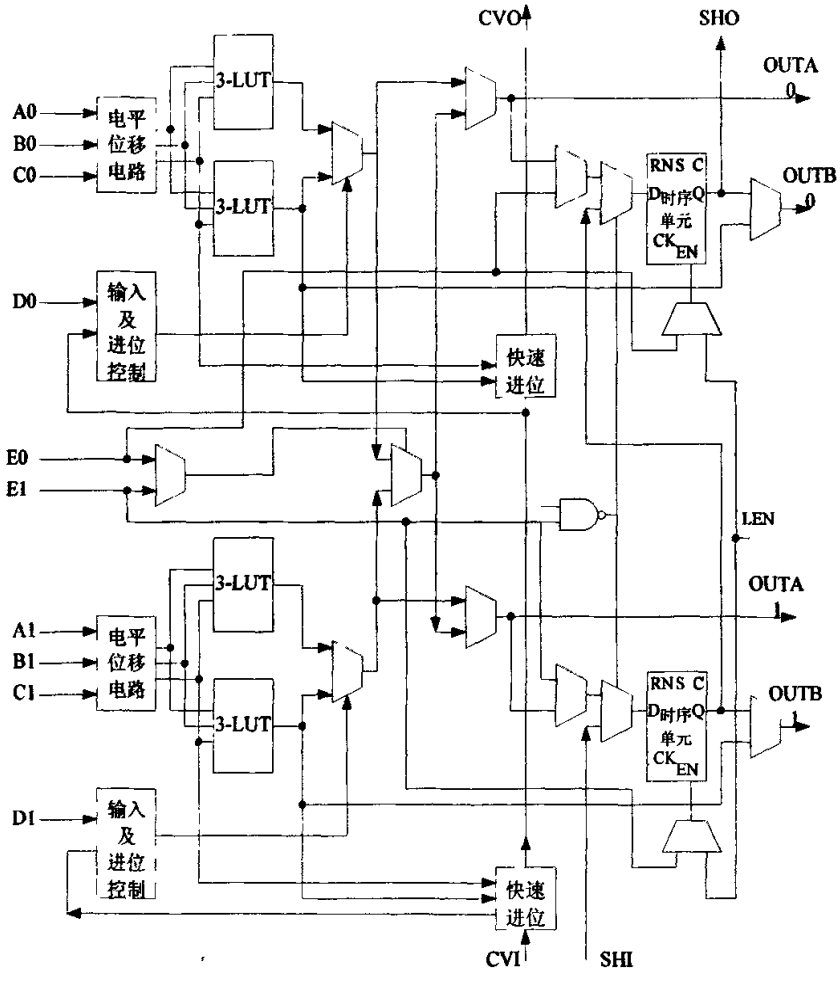

图 6 SLICE 和 LC 结构图

2.1.2. FPGA 布线资源结构

FDP250K 整体芯片采用层次式的互连资源结构，由以下几部分组成：CLUSTER 内部紧凑互连；CLUSTER 外部分段式互连；I/O 互连；专用互连；BLOCK SRAM 互连等，其层次结构框图如所示。

具体而言，每个 CLUSTER 包含 2 个 SLICE 单元（即 4 个 LC 单元）和 1 个时序控制 SEQ 单元，之间由内部紧凑互连进行连接；每个 CLUSTER 包含 12 个输入和 8 个输出，均匀分布在 CLUSTER 四边，即每边含有 3 个输入和 2 个输出。

CLUSTER 外部提供三种分段式互连线资源，水平通道数与竖直通道数相同，其中每个互连通道拥有 2 倍 CLUSTER 线，4 倍 CLUSTER 线，长线；三种分段式互连线共同组成统一的 Wilton 结构的开关盒 SB。在相邻的 SB 之间，上述互连资源以交错扭线方式相连。如图 8 所示。

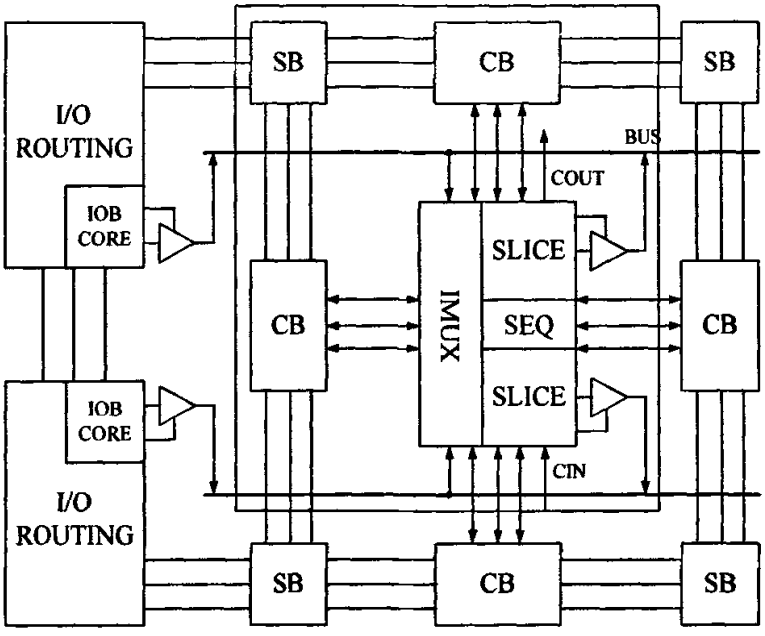

图 7 层次式可编程互连资源结构示意图

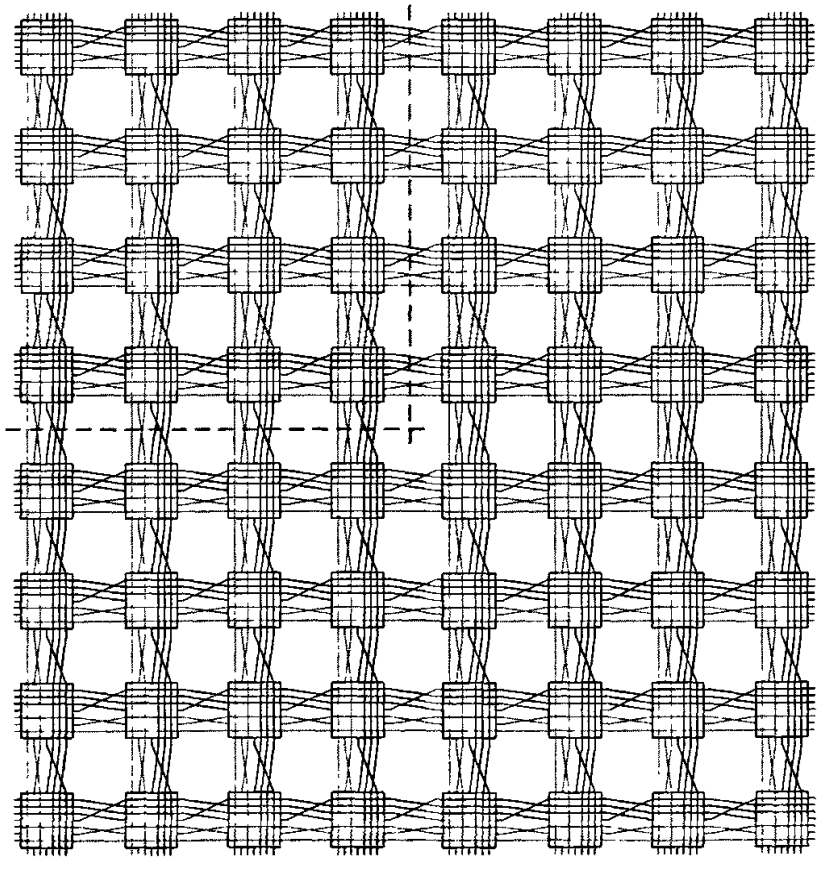

图 8 互连结构示意图

I/O 互连提供内部互连通道与 IOB 输入输出控制逻辑之间的连接以及 I/O 专用互连环路。I/O 互连提供内部互连通道与 IOB 输入输出控制逻辑之间的连接，左右两侧的 IOB 同与之最近的一个水平互连通道相连，上下两侧的 IOB 同与之最近的一个竖直互连通道相连，其具体分布如图 9 所示。IO 互连中包括了 2 倍 CLUSTER 线、4 倍 CLUSTER 线、长线，水平与竖直通道数相同，此外，内部互连通道与 I/O 专用互连资源之间亦有连接。

I/O 专用互连线以交错扭线方式相连，从而在芯片四周形成专用互连环路为所有的 IOB 提供信号连接与交换，如图 9 所示。图中，蓝线代表 I/O 4 倍 CLUSTER 线，每 4 个 IOB 为一连接循环；红线代表 I/O 2 倍 CLUSTER 线，每 2 个 IOB 为一连接循环；绿线代表 I/O 长线。连线跨过 IOB 则代表此连线在 IOB 中不断，反之则代表断开。

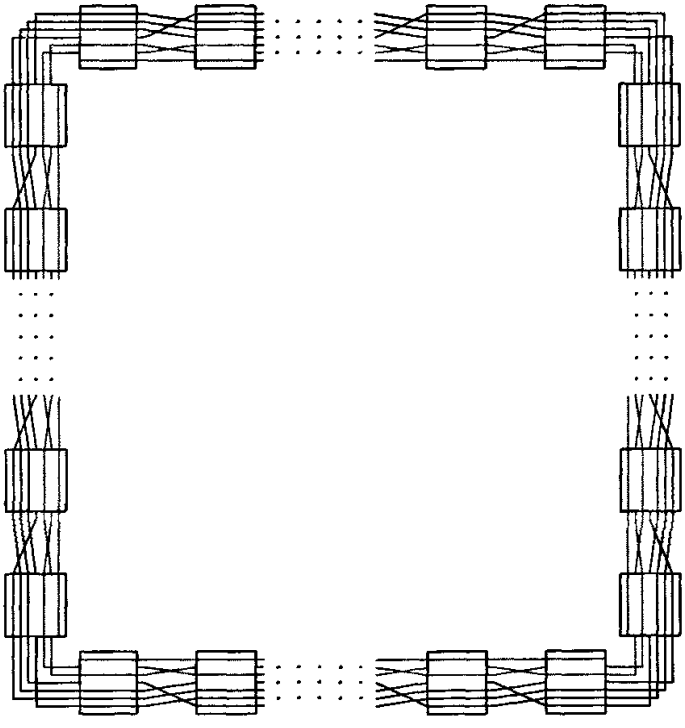

图 9 I/O 专用互连环路示意图

水平与竖直 CLUSTER 之间分别提供从左至右、从上至下的两条快速输入输出的级联逻辑，竖直 CLUSTER 之间还提供自下而上的快速进位链与移位链逻辑，芯片内还有水平总线，均匀分布在水平通道中。

整体芯片中，水平与竖直 CLUSTER 之间分别提供了从左至右、从上至下的两条快速输入输出的级联逻辑，即 CLUSTER 的组合时序输出端（OUTB）只需通过一个 MUX 即可与其右侧或下侧 CLUSTER 的组合输入端相连，从而为逻辑阵列中相邻的 CLUSTER 提供快速的信号连接，如图 10 所示。

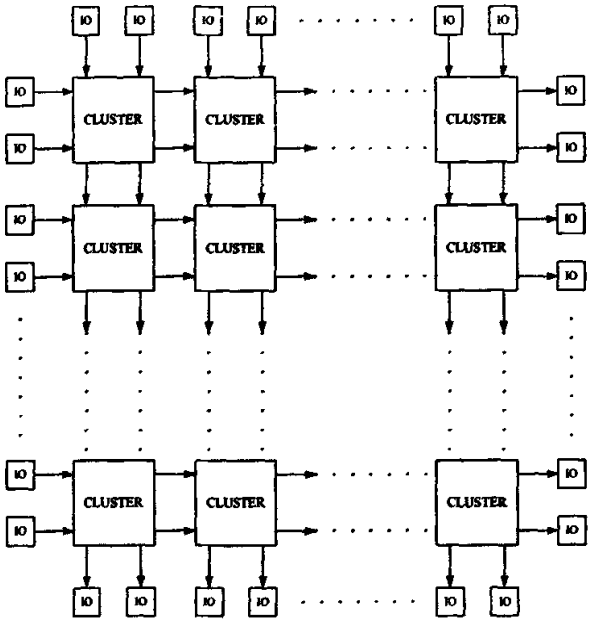

图 10 I/O 水平与竖直快速连接示意图

芯片中竖直 CLUSTER 列提供两条逻辑链路，一条是快速进位链，用以产生高速进位信号，优化算术逻辑的关键路径；另一条是寄存器移位链，由于相邻寄存器输入输出间只经过缓冲器和两级 MUX，因此可以实现高速的数据移位采集，满足串并转换接口等高速电路的需要，同时还可以实现扫描链功能，提高芯片的可测性。其连接关系如图 11 所示

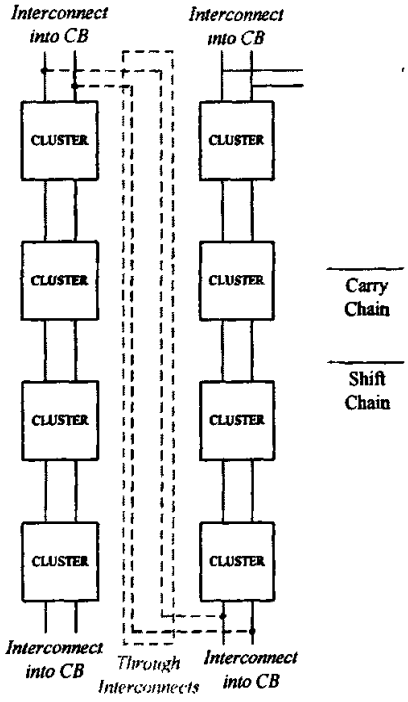

图 11 竖直快速进位链与移位链示意图

整体芯片中还包括了水平总线，均匀分布在水平通道中，即每行 CLUSTER 各拥有上下两条总线。每个 IOB 的输出端也可以驱动上下两条总线。总线的具体连接关系如图 12 所示。

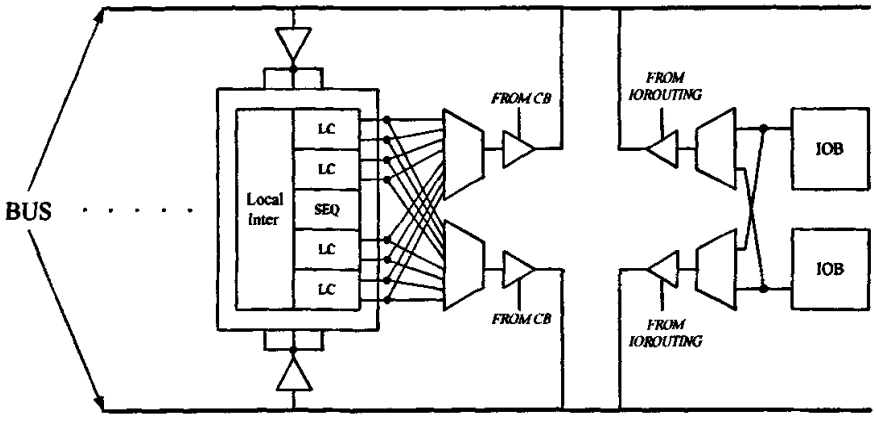

图 12 总线连接示意图

2.1.3. FPGA 可编程 I/O 单元结构

芯片结构中的可编程 IO 单元结构如图 13 所示，核心是一个双向三态 IO，输入输出信号通过可编程开关连接到可编程连线资源上，三态缓冲器的使能信号既可以通过编程 SRAM 控制也可以连接到连线资源上。

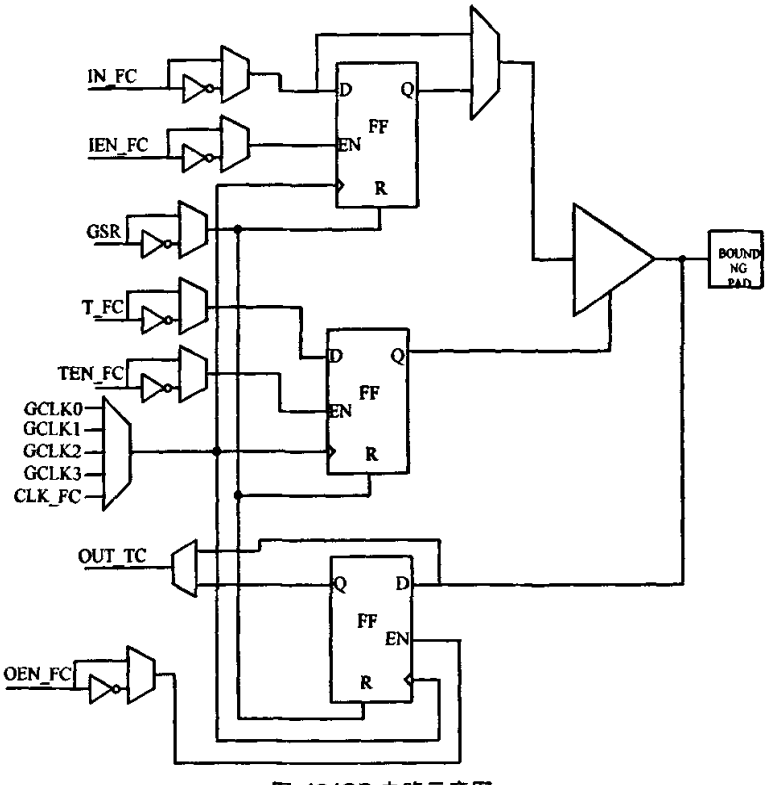

图 13 IOB 电路示意图

这种可编程 IO 可以实现输入、输出、三态输出、双向四种 IO 功能。而且在输入输出信号上都配有寄存器，可以实现对 IO 信号的锁存。寄存器的时钟可以连接到连线资源，也可以连接到专用的时钟网络，寄存器的异步复位信号能够连接到连线资源也能够从芯片的全局清零信号而来。

2.2 FPGA 布线方法

FPGA 和 ASIC 的布线有很大的区别，FPGA 中的布线资源已经定制了，所以需要选择其中的部分布线资源来形成连接，和 ASIC 布线时资源可自由分配不同，这种走线是在有限的资源下进行的，所以可编程逻辑器件的布线算法强烈地依赖可编程逻辑器件的结构尤其是布线资源结构，并且布通率是布线开始就必须考虑的问题[3] [4]。

当电路中所有的逻辑单元的位置选定后，FPGA 的布线器就来决定打通哪些可编程开关，以连接电路需要的所有的逻辑单元块输入和输出管脚。在 FPGA 布线中，通常用一个有向图描述 FPGA 的布线结构。在布线资源图中，每一根连线和每一个逻辑管脚成为一个结点，而可能的连线关系成为边。一些过去的研究已经用一些无向图描述 FPGA，但是无向图无法描述信号的流向，比如当电路中很多输出处都有三态缓冲器，如果用无向图就无法描述信号的输出方向，或者电路中存在多路选择器的时候，信号就是单向流动的。所以用无向图已经无法满足 FPGA 布线的需要，这就需要一个有向图来描述。

FPGA 布线算法可分为两大类：基于通道的布线算法和基于线网的布线算法。

基于通道的布线算法分全局布线和详细布线两步。全局布线借鉴标准元胞和门阵列设计方法中已较为成熟的总体布线算法，优化目标偏重布线的均匀性，避免局部走线过密导致布不通。详细布线算法和标准元胞、门阵列的算法有很大不同。因为可编程逻辑器件布线资源和可编程开关资源已被限定，详细布线实际上是为每条连线找出一组可编程开关，使这组开关连通它们所接的布线段，构成所需连线。可编程逻辑器件详细布线是跨通道的，以线网为单位进行。选择构成连线段时为每一条可能用到的线段计算代价函数，代价函数反映其它线网对被考察线段的需用程度，然后在众多可能的构成方案中选取线段代价总和最低的一组线段。最优解的求解是 NP 完全问题，所以必须采用启发式算法降低时间复杂度[5] 。

基于线网的布线算法[6][7]以线网为单位，采用迷宫算法布线。迷宫算法的优点是对每一条线网而言，能求得当时的（针对线网长度的）最优解，且如果存在通路，一定能找到解；缺点是耗时太大。所以一般 VLSI 设计中总是最后采用迷宫算法来解决其它算法不能解决而遗留下来的问题。可编程逻辑器件阵列不大，所以迷宫算法得以广泛应用。在 FPGA 布线中不仅要考虑线网的长度，更要考虑

众多线网对有限的布线资源的需求程度。为了避免过多地占用 FPGA 中有限的互连资源，也为了缩短布线之间的延时，通常希望这条路径要尽量地短。然而一个电路设计中总是存在很多的线网，每条线网都会占用某些特定的互连资源，当线网比较少的时候，一般问题不大，但是现在的 FPGA 容量都达到了百万门级的规模，而所应用的电路规模也是相当的大，所以某些线网会彼此争夺布线资源，所以在局部会形成拥挤，这就产生了竞争，如何给线网分配布线资源，需要一个合理的仲裁机制。绝大部分 FPGA 的布线器有一些避免拥挤的策略以解决布线资源竞争的问题。。此外，基于线网的布线算法都无法避免布线顺序对布线质量的影响，可以单独或并用以下方法提高性能：一、布线之前为线网排序，精心设计排序算法；二、应用"拆线重布"算法，通过几个循环的重布使结果收敛于较优解。

VPR (Versatile Place and Route) [8]在布线阶段采用了改进型迷宫布线算法。VPR 在普通迷宫算法中另外加入了用于指导布线的资源拥挤度预测，以提高布通率。在具体操作上它分两步完成布线，首先不管布线资源的重用，以每条线网的最小代价为目标对所有线网进行布线，计算所有布线资源的占用率，每个布线资源的占用率作为拥挤度预测指示而影响下一次迭代时使用此布线资源的费用。第二步使用拆线重布的办法进行迭代重布线。

当然，布线器的目标还包括了通过使用短的路径来确保电路的速度。这就是时延驱动的布线方法[13]。因为 FPGA 中的大部分延时都是在 FPGA 的布线之中，所以时延驱动的布线对 FPGA 的性能相当重要。[9][10][11]中的那些布线器使用了时序分析，以确定哪些线网位于，或者接近于关键路径上，所有这些布线器的内核均使用派生的迷宫布线器方法[12]来连接每一个线网的终端。迷宫布线器本质上是运行 Dijkstra 算法，从而在布线资源图中找到线网起始端和终止端之间的最短路径（最低的整体成本）。所有这些算法执行多次布线迭代，在迭代的过程中，有些线网会被拆除并重新布到不同的布线资源上，以解决布线资源的竞争或者改进电路的速度。[10]和[11]使用时序分析，仅仅是帮助确定好的线网拆线重布——如果用一条更快的路径对这些线网重新布线，这些线网就能改进电路速度。然而，不管这些线网在时序方面的关键程度如何，拆线重布这些线网仅仅改变线网的序列，它们都是由同一个迷宫布线算法进行布线的。拥挤度和时延兼顾的 PathFinder（路径搜索器）[9]使用了一个更为复杂的技巧，即每一条连线的拥挤度和时延的折衷是由该连线在时序方面的关键程度所控制。换句话说，一条时序关键的连线将被一条时延最少的路径进行布线，即使这条路径是拥挤的，而一个在时序方面并不关键的线网将占用一条更长的不拥挤的路径。这个算法产生了极好的结果。

2.3 FPGA 互连资源延时模型

现场可编程门阵列（FPGA）是一种可实现多层次逻辑器件。基于 SRAM 的 FPGA 结构由逻辑单元阵列来实现所需要的逻辑函数。FPGA 中，互连资源是预先定制的，各种长度的可分割金属线，缓冲器，MOS 管，所以相对于 ASIC 中互连所占用的面积更大，为了节省芯片面积，一般都采用单个 MOS 晶体管来连接逻辑资源，MOS 晶体管的导通电阻达到了千欧这样一个量级，可分割金属线段的电阻相对于 MOS 管来所是可以忽略的，然而它和地之间的电容达到了 0.1pf [14]。为了评估 FPGA 的性能，用 HSPICE 仿真可以获得非常精确的结果，但是需要花费太多的时间。这在时驱工艺映射，时驱布局布线，静态时序分析中是不可行的。这就需要一种快速而精确的模型。

FPGA 中连接盒、开关盒都是由 MOS 管组成的。FPGA 中的时延很大部分取决于互连，而 MOS 传输晶体管在互连中又占了很大的作用。所以对于 MOS 管的建模对 FPGA 时延估算有很大的影响。对于 MOS 管，Muhammad [15]采用导通电阻来代替 MOS 管。然后用 Elmore [16]时延和 Rubinstein[17]时延模型估算互连时延。Elmore[23]时延用电路的一阶矩来近似信号到达最大值 50%时的时延，而 Rubinstein 也是通过计算电路的一阶矩估算时延的上下边界来估算电路的时延，然而他们都是用来计算 RC 互连时延，因为传输管是非线性器件，所以没有一个固定的电阻，所以这样的近似过于粗糙[21]。

Muhammad[15]提出的用在 FPGA 时延计算中的 MOS 管时延模型如所示，

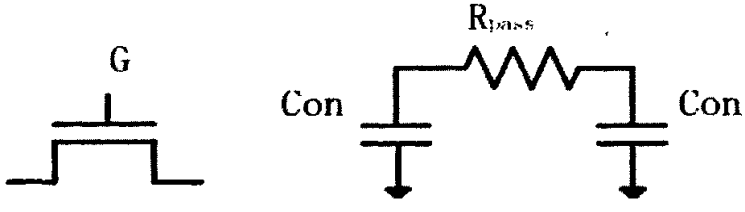

图 14 MOS 管的等效电阻模型

R 为传输管的导通电阻，Muhammad[15]中提到导通电阻在不同栅源电压时有不同的电阻值，取最大导通电阻时，再用 Rubinstein 分析时延模型会获得更好的精度。Fig.1 中电容为

$$C_{on} = C_{drain}(C_{source}) = C_{diffusion} + \frac{1}{2} \times C_{gate}$$

其中：

$$C_{diffusion} = C_{ja} \times (ab) + C_{jsw} \times (2a + 2b)$$

其中的 C_{ja} 是结面电容，C_{jsw} 为结侧壁电容，a 和 b 是扩散区的宽和长。

$$C_{gate} = \frac{\varepsilon_0 \times \varepsilon_{SiO_2}}{T_{ox}} \times A$$

其中 ε_0 是介电常数，ε_{SiO_2} 是二氧化硅的介电常数，A 是栅的有效面积，T_{ox} 是栅和衬底之间的氧化层的厚度。因为导通电阻随着源漏电压的增加而单调非线性增加，为了简单，选择最大的导通电阻作为模型中的电阻值。这种模型由于对传输晶体管的导通电阻采用固定电阻值，所以必然带来非常大的误差，在时延分析中是不够精确的。

第三章 延时模型

3.1 简介

岛型 FPGA 中连接盒、开关盒基本都是由 MOS 管组成的。FPGA 中的时延很大部分在于互连，其中又以 MOS 传输晶体管占了很大的作用。所以对于 MOS 管的建模对 FPGA 时延估算有很大的影响。先前的一些模型要么是太简单，精度不够，要么就是太复杂，不能达到计算速度要求。我们提出了一种新的延时模型[22]。

3.2 在阶跃响应下的等效电阻模型

首先对如图 15 所示的电路进行分析，假定 G 端信号为 Vdd，而 $E(t)$为一个阶跃信号，当时间大于零后，电压为 Vdd。小于零时刻，电压为 0。其中的电容 C 包含晶体管本身源端对地电容和负载电容

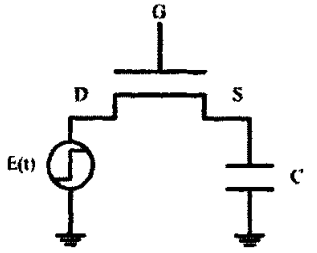

图 15 阶跃输入下的 MOS-C 电路

在阶跃信号激励下，D 端电压和源端相同，所以在任意时刻

$$V_{DS} \geq V_{GS} - V_T \tag{1}$$

V_{DS} 是漏源电压，V_{GS} 是栅源电压，V_T 为阈值电压，传输管处于饱和状态，忽略一些二级效应，此时 D 端的电流可以用如下公式[18]来表示

$$I_D = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{GS} - V_T)^2 \tag{2}$$

$$V_T = V_{TH0} + \gamma(\sqrt{2\phi_F + V_{SB}} - \sqrt{2\phi_F}) \tag{3}$$

其中 μ_n 是电子迁移率，C_{ox} 是单位面积的栅氧化层电容，W 是氧化层的有效宽度，L 是沟道的有效长度。V_{TH0} 是没有体效应时的阈值电压，γ 是体效应系数，$2\phi_F$ 是表面势，V_{SB} 是源和衬底之间的电压。当衬底接地后，$V_{SB} = V_S$。D 端电流对电容进行充电，所以

$$I_D = C\frac{dV_S}{dt} \tag{4}$$

这样把(2),(3)代入(4),得到:

$$C\frac{dV_S}{dt}=\frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{GS}-V_T)^2$$
$$=\frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_G-V_{TH0}-$$
$$\gamma(\sqrt{2\phi_F+V_S}-\sqrt{2\phi_F})-V_S)^2$$
$$=\frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_G-V_{TH0}-$$
$$(1+\frac{\gamma}{\sqrt{2\phi_F+V_S}+\sqrt{2\phi_F}})V_S)^2 \tag{5}$$

令:

$$\alpha=\mu_n C_{ox}\frac{W}{L}$$
$$V_G=V_G-V_{TH0}$$
$$f(V_S)=(1+\frac{\gamma}{\sqrt{2\phi_F+V_S}+\sqrt{2\phi_F}})V_S=\beta V_S$$

其中:

$$\beta=1+\frac{\gamma}{\sqrt{2\phi_F+V_S}+\sqrt{2\phi_F}}$$

我们发现这个函数接近线性，可以用一阶函数曲线拟合来获得β。如果为了方便，可以用最大输出电压的一半来代替β中的V_S。

将上述α，β，V_Ω代入(5)式:

$$C\frac{dV_S}{dt}=\frac{1}{2}\alpha(V_\Omega-\beta V_S)^2 \tag{6}$$

$$V_S=\frac{V_\Omega}{\beta}(1-\frac{\frac{2C}{\alpha\beta V_\Omega}}{t+\frac{2C}{\alpha\beta V_\Omega}}) \tag{7}$$

从这个等式可以得到当V_S达到最大输出值50%时的确切的时延。

$$D_{50\%}=\frac{2C}{\alpha\beta V_\Omega} \tag{8}$$

如果为了用一阶RC电路来代替传输管，为了获得50%点的精确时延,见图 16

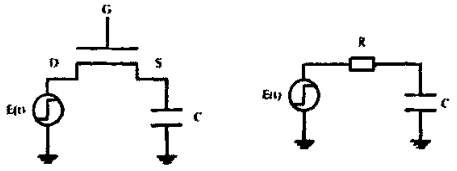

图 16 MOS-C 电路的一阶 RC 等效

(a)阶跃信号下的 MOS-C 电路 (b)一阶 RC 模型

一阶 RC 的响应为

$$V = E(t)(1-e^{-\frac{t}{RC}}) \tag{9}$$

它的 50%时延为

$$d_{50\%} = \ln 2 \cdot RC \tag{10}$$

为了获得 50%时延时的等效电阻，只需要让(8)和(10)相等。这样等效电阻为

$$R = \frac{2}{\ln 2 \cdot \alpha\beta V_{\Omega}} \tag{11}$$

这表示为了获得 50%确切时延，可以用一个电阻 R 来近似。如果用 Elmore 时延尺度来衡量时延，那么只需要：

$$R_{elmore} = \frac{2}{\alpha\beta V_{\Omega}} \tag{12}$$

就能得到比较精确的时延。

3.3 在斜坡输入时的时延模型

因为传输管上的时延比较大，信号经过传输管以后，输出信号不可能是阶跃信号，而变成了一个慢慢爬升的信号。一个传输管的输出信号会变成另外一个传输管的输入信号，此时可以把信号当成一个斜坡输入信号[24]，见图 17 中的 $V_{in}(t)$，假设信号

$$V_{in}(t) = \begin{cases} k \cdot t & 0 < t < t_0 \\ V_{Max} & t_0 < t \end{cases} \tag{13}$$

其中 k 为斜率，V_{Max} 是前级传输管的输出最大电压，$V_{Max} = V_G - V_T$，t_0 为 $V_{in}(t)$ 上升到 V_{Max} 时的时间。

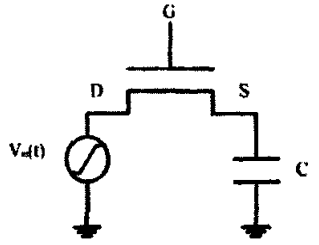

图 17 斜坡输入时的 MOS-C 电路

因为如果第一级传输管的输入电压为 VDD,那么经过这个传输管以后的输出电压最大值要损失一个阈值电压，所以为了考虑一般性，假设输入电压最大值比 VDD 小一个阈值电压。经过一个传输管以后的信号电压爬升比较慢，此时V_{DS}比较小，在晶体管的传输特性曲线中处于线性区，此时不能用饱和区的电流公式，所以电流方程为

$$C\frac{dV_S}{dt}=\mu_n C_{ox}\frac{W}{L}((V_{GS}-V_T)V_{DS}-\frac{1}{2}V_{DS}^2) \tag{14}$$

注意我们只关心输出电压 50%时的时延，在输出电压 50%时延附件，V_{DS}比较小，在 50%电压以后接近饱和区时，V_{DS}逐渐增加的比较快，所以在 50%电压以前，可以忽略平方项，这样公式(13)变成

$$\begin{aligned}C\frac{dV_S}{dt}&=\mu_n C_{ox}\frac{W}{L}(V_{GS}-V_T)V_{DS}\\&=\mu_n C_{ox}\frac{W}{L}(V_{GS}-V_T)(V_D-V_S)\end{aligned} \tag{15}$$

这样的假设是合理的，当输入信号的斜率很小时，V_D上升很慢，此时V_S有足够的时间随着V_D变化，则V_{DS}很小，如果输入信号的斜率非常小的时候，以至于V_D上升到最大电压的时候，V_S和V_D重合,这时候可以看作没有时延。

$$\begin{aligned}&C\frac{dV_S}{dt}=\mu_n C_{ox}\frac{W}{L}(V_G-V_{TH0}-\\&\gamma(\sqrt{2\phi_F+V_S}-\sqrt{2\phi_F})-V_S)(V_D-V_S)\\&=\alpha(V_\Omega-\beta V_S)(kt-V_S)\end{aligned} \tag{16}$$

假设输出也是一个线性函数，它的导数用它的电压和时间的比来近似

$$C\frac{V_S}{t}=\alpha(V_\Omega-\beta V_S)(kt-V_S) \tag{17}$$

这样：

$$V_S=\frac{V_\Omega+\beta kt+\frac{C}{\alpha t}-\sqrt{(V_\Omega+\beta kt+\frac{C}{\alpha t})^2-4\beta V_\Omega kt}}{2\beta} \tag{18}$$

为了求 50%时的时延，那么只需要V_S等于最大输出电压的一半时的时间，当V_S最大时，传输管将截至，所以：

$$V_S = V_G - V_T \tag{19}$$

把(3)代入(19)，得到

$$V_S = V_G - V_{TH0} - \gamma(\sqrt{2\phi_t + V_S} - \sqrt{2\phi_F}) \tag{20}$$

从等式(6),(20)可以得到最大输出电压为

$$V_{SMax} \approx \frac{V_\Omega}{\beta} \tag{21}$$

为了得到 50%时的时延，用最大电压的一半来求得 50%输出电压时得时间

$$t_{50\%delay} = \sqrt{\frac{V_\Omega^2}{16\beta^2 k^2} + \frac{C}{\alpha\beta k}} + \frac{V_\Omega}{4\beta k} \tag{22}$$

而输出信号定义为

$$V_S(t) = \begin{cases} p \cdot t & 0 < t < t_1 \\ V_{Max} & t_1 < t \end{cases} \tag{23}$$

其中，t_1 为 V_S 上升到 V_{Max} 时的时间，把斜率 p 定义为 50%输出电压和 50%时延时的比值

$$\begin{aligned} p &= \frac{\frac{V_\Omega}{2\beta}}{\sqrt{\frac{V_\Omega^2}{16k^2} + \frac{C}{k\alpha}} + \frac{V_\Omega}{4\beta}} \\ &= \frac{2V_\Omega k}{V_\Omega + \sqrt{V_\Omega^2 + \frac{16\beta k C}{\alpha}}} \end{aligned} \tag{24}$$

这样，输出信号在 $t_1 < t$ 时为

$$\begin{aligned} V_S &= \frac{\frac{V_\Omega}{2\beta}}{\sqrt{\frac{V_\Omega^2}{16k^2} + \frac{C}{kq}} + \frac{V_\Omega}{4\alpha}} \\ &= \frac{2V_\Omega k}{V_\Omega + \sqrt{V_\Omega^2 + \frac{16\beta k C}{\alpha}}} t \end{aligned} \tag{25}$$

3.4 在斜坡输入下的等效负载电容

在 RC 电路中，Elmore 时延没有考虑电阻屏蔽效应，只是用下级电容之和来表示等效电容，而[19][20]计算等效电容时,已经考虑到电阻屏蔽效应，对于 MOS-C 电路来说，图 18 所示，由于电阻不是固定的，所以不能用他们的方法来

计算等效电容。当 MOS-C 作为负载时，可以用一个等效电容来匹配。

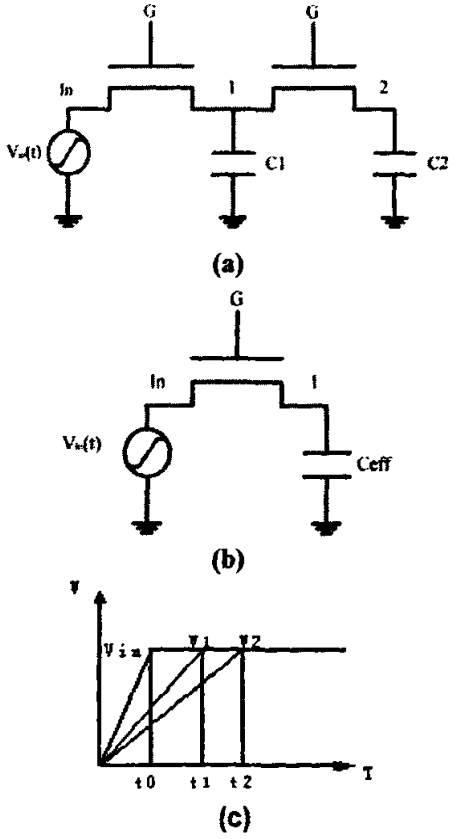

图 18 两级 MOS-C 电路

(a) MOS-C 电路作为负载 (b)等效电容模型 (c) 输入波形 Vin 和输出波形 V1，V2

假设输入信号的线性段为 $V_{in}=kt$，中间节点 1 的信号为 $V_1=p_1t$，而输出节点 $V_2=p_2t$，这样根据电路节点方程

$$C_1\frac{dV_1}{dt}+C_2\frac{dV_2}{dt}=\mu_nC_{ox}\frac{W}{L}(V_G-V_{TH0}-\frac{\gamma V_1}{\sqrt{2\phi_F+V_1}+\sqrt{2\phi_F}}-V_1)(V_{in}-V_1) \tag{26}$$

$$C_2\frac{dV_2}{dt}=\mu_nC_{ox}\frac{W}{L}(V_G-V_{TH0}-\frac{\gamma V_2}{\sqrt{2\phi_F+V_2}+\sqrt{2\phi_F}}-V_2)(V_1-V_2) \tag{27}$$

由(27)可得

$$C_2p=\alpha(V_\Omega-\beta V_2)(p_1t-V_2) \tag{28}$$

可以得到

$$V_2 = p_2 t = \frac{2V_\Omega p_1}{V_\Omega + \sqrt{V_\Omega^2 + \frac{16\beta p_1 C_2}{\alpha}}} t \tag{29}$$

由(26)可以得到

$$C_1 p_1 + C_2 p_2 = \alpha(V_\Omega - \beta V_1)(kt - V_1) \tag{30}$$

见图 18 两级 MOS-C 电路图 18(b)，它的电路节点方程为

$$C_{eff} p_1 = \alpha(V_\Omega - \beta V_1)(kt - V_1) \tag{31}$$

这样为了获得等效电容，让(30),(31)相等

$$\begin{aligned} C_{eff} &= C_1 + C_2 \frac{p_2}{p_1} \\ &= C_1 + C_2 \frac{2V_\Omega}{V_\Omega + \sqrt{V_\Omega^2 + \frac{16\beta p_1 C_2}{\alpha}}} \end{aligned} \tag{32}$$

然而其中还有一个参数 p_1 未知，让 k 来近似 p_1，这样的近似是合理的，当 k 比较小的时候，输出斜率 p_1 和 k 非常接近。

3.5 实验结果

我们的程序在 CPU 2GHz，512M 内存 PC 上运行，估算时延用的是 Matlab 程序，仿真用 HSPICE

（1）在 SMIC 0.18um 工艺库下对阶跃信号下的等效电阻模型进行仿真。VDD 为 1.8V，仿真结果见图 19，V.SPICE 为用 SPICE 对 MOS 管仿真的结果，对 MOS 管仿真时的阶跃信号的高电平为 1.8V。V.Resti.是我们获得等效电阻后仿真得到的结果，由于有阈值电压损失，MOS 管输出不能达到 1.8V，对等效电阻模型进行仿真时，以传输管稳定输出电压值作为阶跃信号的输入高电平值，这样输出稳定时的电压和 MOS 管输出稳定电压相同，等效电阻波形信号达到 50%最大输出值时时延为 22.76ps 而对 MOS 管仿真时测得的时延为 23.79ps,误差为 4.33%

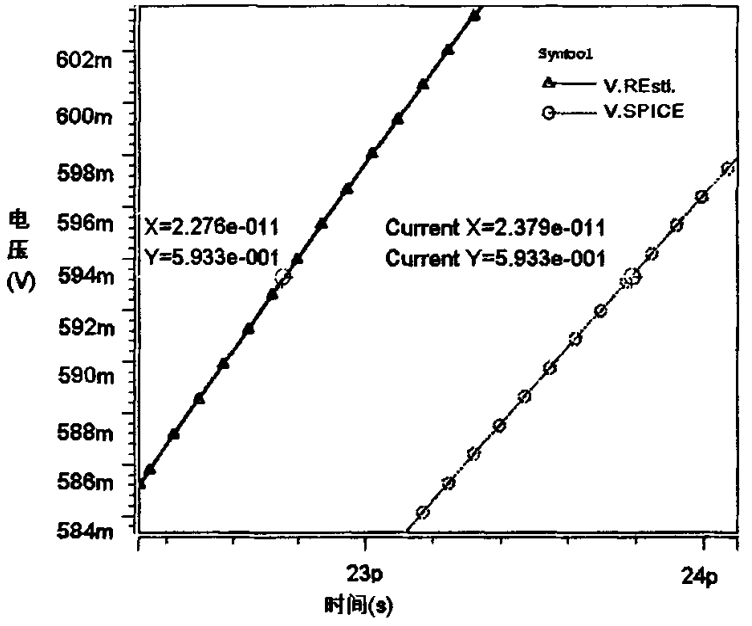

图 19 阶跃信号下等效电阻输出结果

（2）在斜坡输入时的结果和 SPICE 仿真比较结果见

，输入电压为 1.2V,接近实际 MOS 管输出，上升时间从 700ps 到 1.2ns，负载电容为 20fF，接近 800um 长线的电容，FDT 是我们估算方法，MEsti.是 Muhammad[15]中用到的等效电阻模型，然而 Muhammad [15]中没有提供单个传输管的时延数据，我们按照他们提供的方法，用电阻的最大值来估算。由于 Muhammad [15]提到的输入信号是阶跃信号，这里不同信号上升时间比较的时候，把他们的时延值当成一个值。从结果看，仿真得到的时延随着上升时间的上升而逐渐缩小，而我们的估算结果也随着减小，和 SPICE 比较后的误差逐渐减小，误差来自对 MOS 管线性区电流中的平方项忽略，随着上升时间的不断增大，源漏之间的电压随之不断减小，忽略平方项导致的误差自然会逐渐减小。而用 Muhammad [15]的模型仿真得到的时延在斜坡输入是的误差很大，最大达到了 42.82%，并且误差随着上升时间的增大而增大，并且我们的方法算到的输出斜率误差在 2.08%以内，表明对后级的影响较小，所以我们的模型更加精确

上升时间		700ps	800ps	900ps	1000ps	1100ps	1200ps
时延 (ps)	SPICE	18.0	17.8	17.6	17.5	17.4	17.0
	FDT	19.68	19.11	18.50	17.86	17.20	16.52
	MEsti.	24.28	24.28	24.28	24.28	24.28	24.28
相对 SPICE 时延误差	FDT	9.33%	7. 36%	5.11%	4.44%	1.15%	2.82%
	MEsti.	34.89%	36.40%	37.95%	38.74%	39.54%	42.82%
输出斜率 (V/ns)	SPICE	1.63	1.44	1.28	1.16	1.06	9.7
	FDT	1.60	1.41	1.26	1.14	1.04	9.58
相对 SPICE 斜率误差	FDT	1. 84%	2. 08%	1.56%	1.72%	1.89%	1.24%

表 1 斜坡输入时比较结果

（3）在一个 MOS-C 作为负载的时候，同样也是在斜坡输入的情况下，用等效电容来计算节点 1 (Fig.5)的时延，见，FDT 为我们估算的结果，时延误差最大为 13.23%，而输出斜率最大达到 4.96%，随着上升时间的增大，逐渐缩小。并且基本随着上升时间的增大，逐渐缩小。

上升时间		700ps	800ps	900ps	1000ps	1100ps	1200ps
时延 (ps)	SPICE	35.3	35.3	35.2	35.2	35.2	35.2
	FDT	39.97	39.95	39.81	39.56	39.22	38.83
相对 SPICE 误差		13.23%	13.17%	13.10%	12.07%	11.42%	10.31%
输出斜率 (V/ns)	SPICE	1.56e9	1.41e9	1.25e9	1.14e9	1.04e9	9.50e8
	FDT	1.51e9	1.34e9	1.21e9	1.10e9	1.01e9	9.25e8
对 SPICE 误差		3.21%	4.96%	3.20%	3.51%	2.88%	2.63%

表 2 斜坡输入时等效负载电容比较结果

（4）用我们的 FPGA 结构对测试电路布局布线后生成的网表进行时延估算，这里只对电路的最长互连线进行比较，结果见表 3。假设开关盒里缓冲器在三种方式下的时延相同。因为缓冲器在电路中时延占的比重较大，所以相对来讲，测得的误差要比前面测得的误差小。用我们的方法（FDT）估算所耗费的时间是 Muhammad 方法的 2 倍左右，而比 SPICE 相比快了 40～50 倍。而我们的方法估算时延比 Muhammad 的方法精度提高了 1～2 倍。

测试电路	最长互连时延(ps)			仿真时间(s)			相对 SPICE 误差	
	SPICE	Elmore	FDT	SPICE	Elmore	FDT	Elmore	FDT
miller_2ip	793.79	905.61	848.44	2.09	0.0156	0.0469	14.09%	6.88%
cm150a_orig	1167.11	1385.91	1249.14	4.62	0.0468	0.0938	18.75%	7.03%
rnd4-2	897.20	1043.46	950.39	2.97	0.0313	0.0781	16.3%	5.93%
ieee_adder_2ip	643.78	771.00	683.78	2.39	0.0312	0.0625	19.76%	6.21%
cm82a	752.07	863.02	782.96	2.37	0.0312	0.0625	14.75%	4.11%

表 3 电路网表测试结果

3.6 本章小结

本章提出了一种 FPGA 互连资源 MOS 管斜坡输入时分析时延模型，并且提出了计算 MOS 管级连时的等效电容的计算方法，通过实验数据的比较，我们的模型计算快速，并且足够精确。可以被用在 FPGA 时驱布局，时驱布线中。

第四章 Dogleg 现象探索

4.1 Dogleg 现象

在岛型 FPGA 中，单个传输管能够被用来作为线和线之间的连接，或者线和管脚之间的连接。然而许多 FPGA 制造商都选用缓冲器和 MUX 来作为 CB 结构中的连接器件。通道中的连接线都是预先制造的。它们可以用来连接源端管脚到终端管脚，如图 20 所示。图中的每个通道里都有 4 个线轨。交叉处的节点表示一个传输管。信号从 OUT 管脚输出，通过三个传输管到达 IN 管脚，其中一个在开关盒中，另外两个在连接盒中，中间 CLB 管脚上的传输管不在信号通路上，所以不用考虑。传统的布线算法例如 VPR[13]把线和管脚作为布线资源，PIN 只能作为输入或者输出管脚，它们不是一个线网的起点就是线网的终点。这样的假设忽略了管脚实际在物理上可以作为互连线来使用的情况。（VPR[13]认为 dogleg 现象本身对性能提高不多，Dogleg 这个术语来源与通道布线中的启发式算法，Deutsch[27]的曲干通道布线算法中引入了 dog-leg 技术，很大程度上解决了线网之间的垂直约束问题，扩大了左边算法的应用范围。）

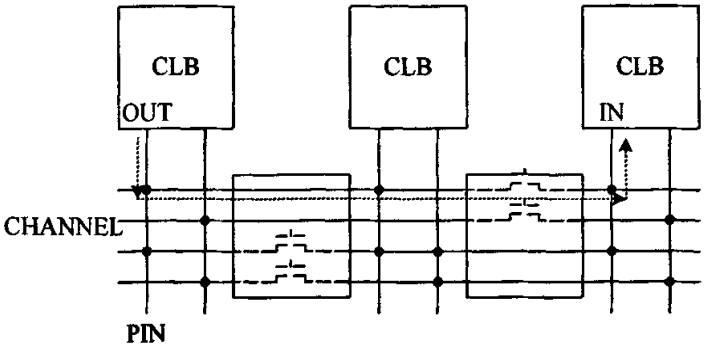

图 20 通道中的布线

当我们把管脚也作为线的时候，如图 21 所示，有一个三端线网，OUT 管脚是驱动源，其他两个是接收端，第一个接收端 IN_A（SINK）能通过第一和第三线轨到达，但是第二个接收端 IN_B 不能直接到达，从 OUT 管脚不能直接连接到 IN_B 管脚。在 VPR 中，通过绕线来到达 IN_B 管脚，但是这样会损失很多布线资源。

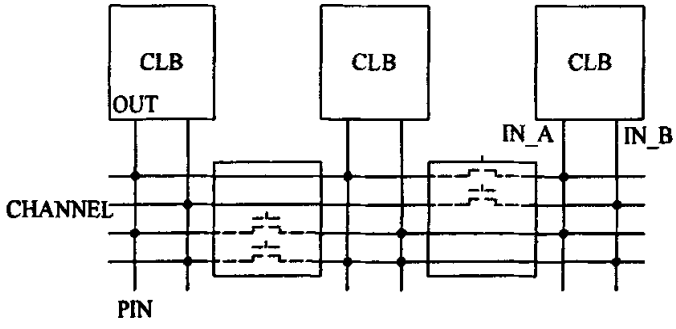

图 21 互连资源图

如果我们认为管脚具有和互连线一样的连接特性，我们可以在线轨之间切换。如图 22 所示，虚线表示从驱动端到接收端的通路。在中间的 CLB 管脚处，通路二分，从某种意义上来讲，连接盒和开关盒有相同的属性，都能用来切换通路。这一点和 VPR 不同。这种情况下，我们节省了很多布线资源。当然这也带来了另外的问题——中间的 CLB 不能被占用。这意味着我们不能在这个 CLB 上放置任何逻辑，否则会导致信号冲突。

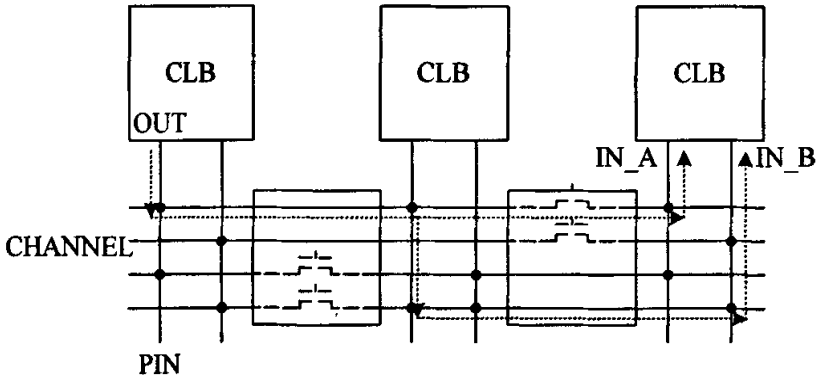

图 22 管脚作为互连资源后的路径图

4.2 动态构建布线资源

所谓动态构建布线资源图，就是在构建布线资源图的时候，不再只依托硬件结构，而是根据用户网表的特性来构建布线资源图。这样，构建的布线资源图不再是唯一的。

4.2.1. 布线资源图

VPR[8]使用布线资源图来描述 FPGA，这比任何参数化来得更为通用，因为这样可以描述任意的连接性。并且由于布线资源的连接信息是在包含在图里的，所以也就可以更快地确定连接信息，例如某一个给定的布线线段能够连接到其他互连线。我们也用布线资源图来描述 FPGA

在布线资源图里，互联线段和逻辑单元的管脚都是布线资源节点，每个开关则成为两个节点间的有向边（对于单向开关，如缓冲器）或一对有向边（对于双

向开关，如传输管））。图 23 举例说明了对应的一部分包含一个两输入和一个输出查找表的 FPGA 的布线资源图。

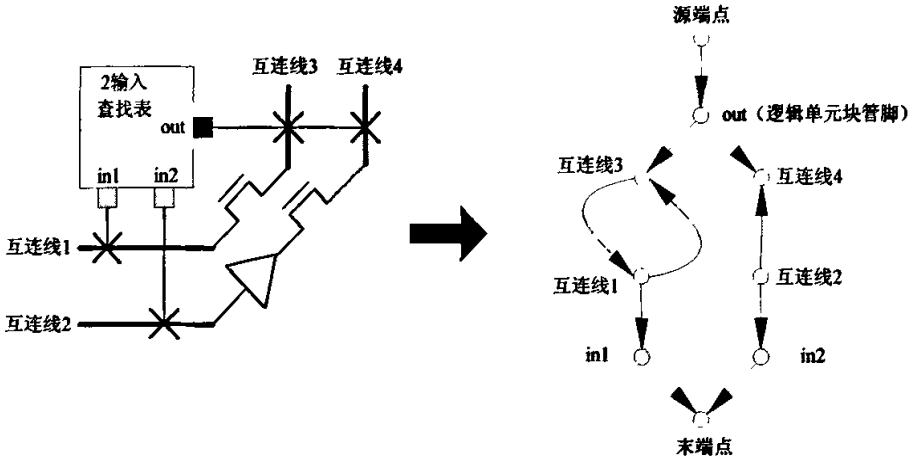

图 23 逻辑单元硬件连线资源和对应的布线资源图

逻辑等效的管脚在 FPGA 中经常出现：例如，查找表的所有输入管脚是逻辑上等价的。这意味着，布线器可以通过利用查找表的任意一个输入管脚来完成给定的连接；改变查找表内的配置信息可以修正因布线器引起的查找表管脚的重排序的差异。这里值得指出的是我们这里说的只是查找表的所有输入管脚，并不是指所有查找表能连到的外部逻辑单元的管脚逻辑等效，在外部的逻辑单元的输入管脚层面上讲，如果逻辑单元管脚和查找表的输入管脚的连通度比较低的时候，输入管脚不一定能逻辑等效。我们对这种逻辑等效性的建模是在所有线网的起始处添加一个源端点（source），在所有线网的终结处添加一个末端点（sink）。所有逻辑等价输出管脚只拥有一个源端点，从源端点到这些输出管脚之间有边相连。类似地，所有逻辑等价输出管脚只拥有一个末端点，从这些输入管脚到末端点之间有边相连。

4.2.2. 动态构建布线资源图

我们实现了一个动态的方法来构建布线资源。由于布线连接关系包含在布线资源图中，对于固定硬件结构的 FPGA 芯片来讲，只有一种固定的布线资源图，传统的布线算法例如 VPR 考虑每个芯片只有一个有效的布线资源图。所以 VPR 根据一种结构生成一种布线资源图，只有在更换芯片结构后，才会更换布线资源图，在这个布线资源图中，包括了各种互联资源点和互联之间的固定连接关系。然而我们认为布线资源图应该取决于用户的网表，不同的网表，布线资源点所扮演的角色也不一样，所以不同的网表可以有不同的布线资源图。VPR 中的逻辑单元的管脚只能作为布线资源点，不能作为互联资源来对待，然而在特定的情况

下，逻辑单元管脚是可以作为互联资源来利用的。如图 24 所示，在采用 dogleg 技术情况下，假设 out 管脚没有被逻辑占用，可以作为互连，那么这个管脚 out 就能作为互联资源使用，而不再作为逻辑单元的管脚，这时，在互连线 1 和 2 之间存在了一条通路，不用占用很多的互连线就能实现互连线 1，2 之间的连接。

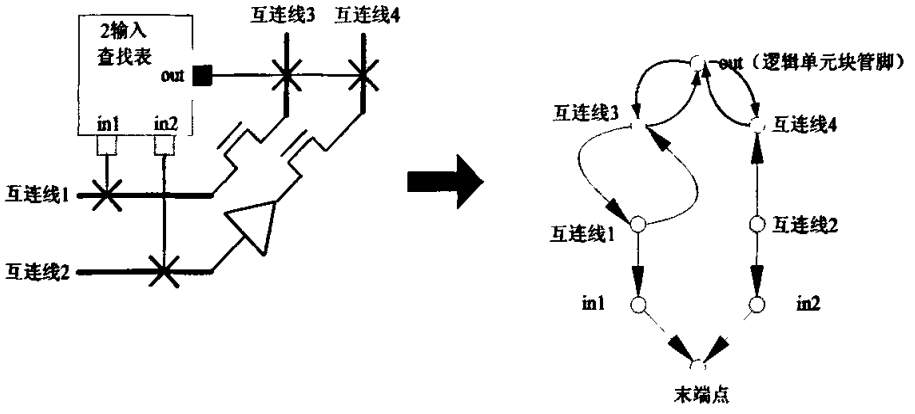

图 24 利用 dogleg 后逻辑单元硬件连线资源和对应的布线资源图

VPR 在知道芯片结构后就开始构建布线资源图，这种构建方法只能得到单一固定的布线资源图，我们改变了 VPR 的流程，在读取了布局信息后，再构建布线资源图，这样我们就可以知道哪些 CLB 已经被占用，所有没有被占用的 CLB 的管脚都可以被用来作为互连资源。如果有很多没有用到的 CLB，那么就有很多管脚可以被用来当成布线资源。

VPR 中定义了以下几种布线资源类型

SOURCE、

SINK、

OPIN、

IPIN、

CHANX、

CHANY

其中 SOURCE、SINK 是虚拟的点，SOURCE 表示源点，SINK 表示终点，而 OPIN 表示输出管脚，IPIN 是输入管脚，OPIN 和 IPIN 都是逻辑单元的管脚，CHANX 和 CHANY 表示互连线，CHANX 表示水平通道上的互连线，CHANY 表示垂直通道上的互连线。

我们增加了一种新的布线资源类型 PINWIRE，它的地位类似于 CHANX 和 CHANY，占用到的 CLB 中的管脚不是 OPIN 就是 IPIN，而没有用到的 CLB 中的管

脚的类型不再是 OPIN 或者 IPIN，而是 PINWIRE 这种互连类型，这时的管脚已经不再是作为逻辑单元的输入管脚或者输出管脚，而是作为互连通道中的互连线。当 CLB 没有被占用的时候，就会有很多的管脚可以用来作为互连资源，这样，互连中的布线资源就会增多，布通率也可以随之提高。

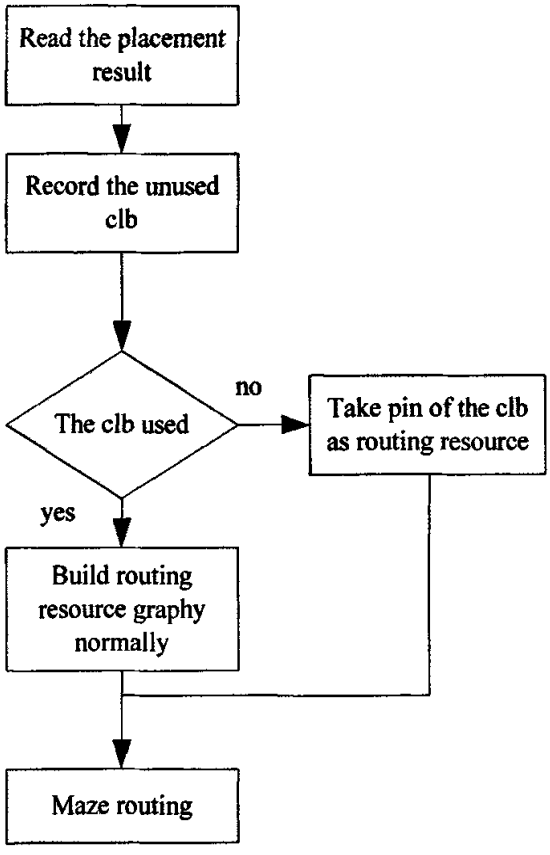

图 25 算法流图

该流程中如图 25，我们首先会读取布局信息，错误！书签自引用无效。从布局信息中我们就能了解到 clb 的利用情况，如果这个 clb 被占用了，那么 clb 的管脚当成正常的布线资源来使用，如果 clb 没有被占用，那么我们可以把它的管脚当成是互连资源来使用。最后和 VPR 一样，进行迷宫布线。

4.3 结果分析

1. 首先，我们使用了类似于 VPR 求最小通道宽度的方法，让芯片大小刚好能容纳用户的电路设计。开关盒类型是 SUBSET。我们利用 VPR 的布局器产生相同的布局结果作为布线器的输入。然后利用 VPR 布线器和我们的布线器来获得每个电路网表需要的最小的布线轨道数目。从上看，和 VPR 相比，我们的方法能平均提高 11%。在某些测试电路中，我们的方法能得到 16%的提高。因为每个通道中所需要的最小线轨数降低了，所以所在 SUBSET 开关盒情况下布通率提

高了。

	SUBSET(最小芯片面积)		
	VPR [13]	FDR	改进百分比
Alu4	35	32	9%
apex2	42	38	10%
apex4	40	36	10%
bigkey	24	21	13%
clma	50	47	6%
des	25	22	12%
diffeq	31	27	13%
dsip	25	21	16%
frisc	45	41	9%
misex3	37	33	11%
pdc	61	52	15%
s5378	26	25	4%
seq	42	36	14%
spla	52	47	10%
table3	33	29	12%
tseng	28	25	11%
average	-	-	11%

表 4 SUBSET 开关最小面积下的布通率比较

2．当我们选择 WILTON 开关盒类型的时候，我们的方法（FDR）和 VPR 相比平均能提高 2%。因为 WILTON 开关盒的灵活度已经足够高了，所以结果提高的不多。结果如所示。

	WILTON (minimum chip size)		
	VPR [13]	FDR	改进百分比
alu4	30	30	0%
apex2	37	36	3%
apex4	34	34	0%
bigkey	20	20	0%

clma	43	43	0%
des	22	22	0%
diffeq	26	25	4%
dsip	21	20	5%
frisc	39	39	0%
misex3	32	31	3%
pdc	49	49	0%
s5378	24	23	4%
seq	35	34	3%
spla	44	43	2%
table3	30	29	3%
tseng	24	24	0%
average	-	-	2%

表 5 WILTON 开关最小面积下的布通率比较

3. 当芯片的大小固定为最小芯片大小 2.25 倍的时候（1.5 倍长和 1.5 倍宽）来容纳网表电路，开关盒为 WILTON 类型时，结果能提高 3%。当面积扩大以后，因为用到的 CLB 数目是固定的，但是没有用到的 CLB 数目增加了，所以作为布线资源的管脚也增加了，我们能获得一个更好的结果。其中有一些网表不能提高布通率，因为他们的布通率已经达到了一个高点，所以很难再提高。结果如表 6 所示。

	WILTON （2.25 倍芯片面积）		
	VPR [13]	FDP	改进百分比
alu4	28	28	0%
apex2	34	31	9%
apex4	32	31	3%
bigkey	20	19	5%
clma	41	39	5%
des	21	21	0%
diffeq	24	24	0%
dsip	20	20	0%
frisc	34	34	0%
misex3	30	27	10%
pdc	45	44	2%
s5378	22	21	5%

seq	31	30	3%
spla	41	41	0%
table3	24	24	0%
tseng	21	21	0%
average	-	-	3%

表 6 WILTON 开关 2.25 倍最小面积下的布通率比较

4.4 本章小结

本章对 dogleg 现象就行了探索，在 SUBSET 开关盒情况下，用 dogleg 能比不用 dogleg 平均提高 11%的布通率，然而在 WILTON 条件下，改变不是很多，这是因为 WILTON 开关的灵活度已经比较高，所以提高的幅度相对较小。

第五章 FDP250K 布线系统

5.1 FDP250K 布线系统简介

FDP250K 软件系统是针对 FDP250K 芯片的软件辅助开发系统。布线模块是 FDP250K 软件系统中的关键部分。布线模块的目的在于，在已经布局好的设计网表结果的基础上，利用 FDP250K 芯片中的可编程连线资源来实现布线。布线结果的电路延时信息是由后续的时序分析模块得到。

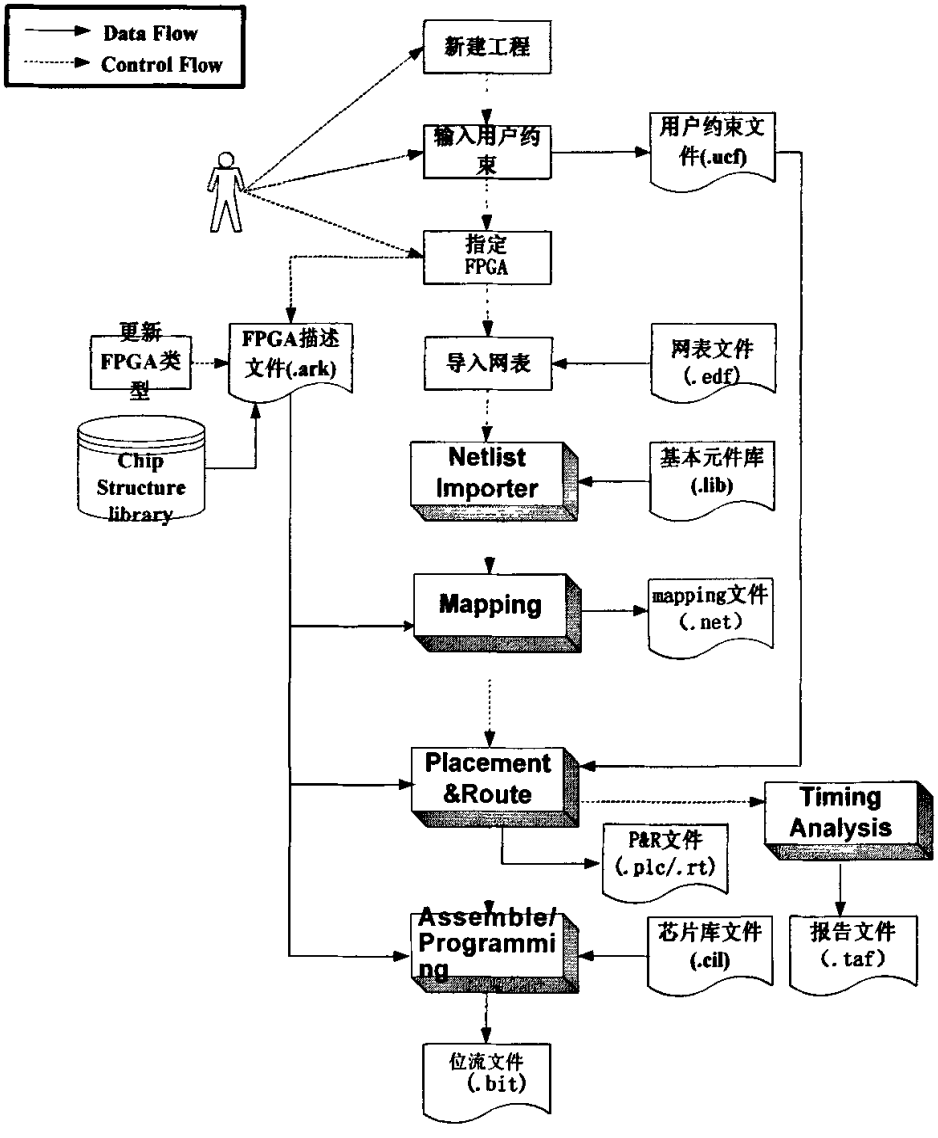

图 26 FDP250K 软件系统流图

设计的前端输入由 Verilog 等硬件描述语言输入，可以调用我们设计单元库中预先定义的宏模块，这些宏模块是经过时延优化以后的 IP 模块，具有结构合

理，延时小等优点，当设计生成的网表综合以后，经过网表转换模块转成我们自己内部定义的网表文件格式，内部网表经过 Mapping[25]模块进行工艺映射，得到映射后的网表。由于我们的硬件结构是一个层次化的结构，所以还需要经过 Packing[26]模块，在该模块中，逻辑单元 (LC)被装箱成逻辑单元簇（CLUSTER），形成的网表再经过 P&R 模块进行布局、布线，经过时序分析模块的分析，验证时序功能的正确性，最后经过位流生成模块生成位流文件，在该文件中包含了 FPGA 芯片的详细配置信息，可以通过编程下载模块下载到具体的芯片中，实现设计的数字电路系统。本论文主要讨论这个 CAD 系统中的布线系统。

5.2 FDP250K 布线系统流程

FDP250K 布线系统流程如图 27 FDP250K 布线模块流程图所示

1. 开始
2. 读取硬件信息（arch 文件）和网表文件（net 文件，plc 文件,macro.lib(包括 RAM 和宏的信息)）

   命令行分解

   解析命令行，得到文件路径、文件名称、布线方式等基本信息

   读取库文件

   读取宏模块信息

   读取芯片结构文件

   读取芯片结构文件，获得芯片的结构信息，包括芯片的整体描述和详细描述，如管脚信息和管脚的连接关系，布线信息以及开关信息等。芯片结构文件描述了整个芯片的硬件信息。

   读取工艺映射后文件

   读取工艺映射后的结果文件，获得 IO、CLUSTER、RAM 等的线网的连接关系，工艺映射后文件包含了所有线网的连接关系。

   读取布局文件

   读取 IO、CLUSTER、RAM、宏模块、时钟等的位置信息
3. 构建布线资源图，首先要给所有的布线资源分配布线资源点，每个布线资源点都有相应的名字和编号，按照位置顺序来构建 CLUSTER 和 IO 的布线资源点，给 IO 和 CLUSTER 中的每个管脚分配布线资源点，以及 IO 和 CLUSTER 中的 class 数目，需要注意的是，当这个 CLUSTER 被宏模块占用了以后，这个 CLUSTER 中的所有输入管脚就不能再逻辑等效，这和正常使用的 CLUSTER 不同。给 IO 和 CLUSTER 分配完布线资源以后，给时钟中心（CLUSTER CENTER）分配布线资源点，时钟中心的布线资源点分配和 IO 类似。RAM 的布线资源分配也和 IO 类似。然后给布线通道分配布线资源点。

最后给所有的线网标注分配的布线资源点。获得该线网的边界盒（bounding box）范围。

4. 给线网的端点分配节点，这里面运用了虚拟节点（SOURCE、SINK）作为线网的起始点和终点，给其也分配布线资源点，并给每一个线网进行编号。
5. 建立连接关系，

   (A)首先按照位置顺序给 IO 和 CLUSTER 构建连接关系，采用从左到右，从下到上的顺序，根据管脚的输入输出属性来区分，如果是输入管脚，那么通道中的连线连接到这个管脚，这个管脚再连接到虚拟的 SINK 点终结。如果这个是输出管脚，那么从虚拟的 SOURCE 点出发，连接到管脚，再连接到外部通道中的互联线。给 CLUSTER 构建连接关系的时候，如果这个 CLUSTER 被宏模块占用，那么输入管脚不在逻辑等效。而是每个输入管脚都有一个后继 SINK 点，每个输出管脚都有一个前驱 SOURCE 点。

   (B)然后再给时钟中心构建布线资源

   (C)给 RAM 构建布线资源图，RAM 的构建比较特殊，根据使用到的 RAM 情况来构建。RAM 如果有级联的时候，两块构成级联的 RAM 的相同管脚的 SOURCE 和 SINK 是相同的。

   (D)构建通道，包括 x 方向连接到 y 方向，y 方向连接到 x 方向，x 方向连接到 x 方向，y 方向连接到 y 方向.
6. 运用迷宫算法进行布线，给所有的线网都标识级数，标识为最高优先级的线网是不用布线的，这类线网是全局线网，例如全局时钟等都是属于这类线网的，这类线网都有专属的布线资源。

   标识为 0 级（时钟 PAD---走全局时钟----走局部时钟---CLUSTER、IO<同步>，时钟 PAD---只走局部时钟---CLUSTER、IO<同步>）

   时钟线网标识为 1 级 （普通 IO----走 V_QU11----走局部时钟--- CLUSTER、IO<同步>，普通 IO、时钟 PAD - ----走互联--- CLUSTER、IO<同步>，CLUSTER 输出- ----走互联--- CLUSTER、IO<异步>，CLUSTER 输出----走 V_QU11----走全局时钟--- CLUSTER、IO<异步>）

   普通线网标识为 2 级，布线时优先布时钟线网。然后再布普通线网。采取宽度优先的办法一层层向外扩张直到找到目标节点，在若干条都到达目标节点的路径中选取所需要代价最小的路径作为最优路径。(一般两个相邻节点间的代价为 1，但若目标节点是输入 Pin 的话,该段代价为 0.95)
7. 把布线的最终结果输出到布线文件中。

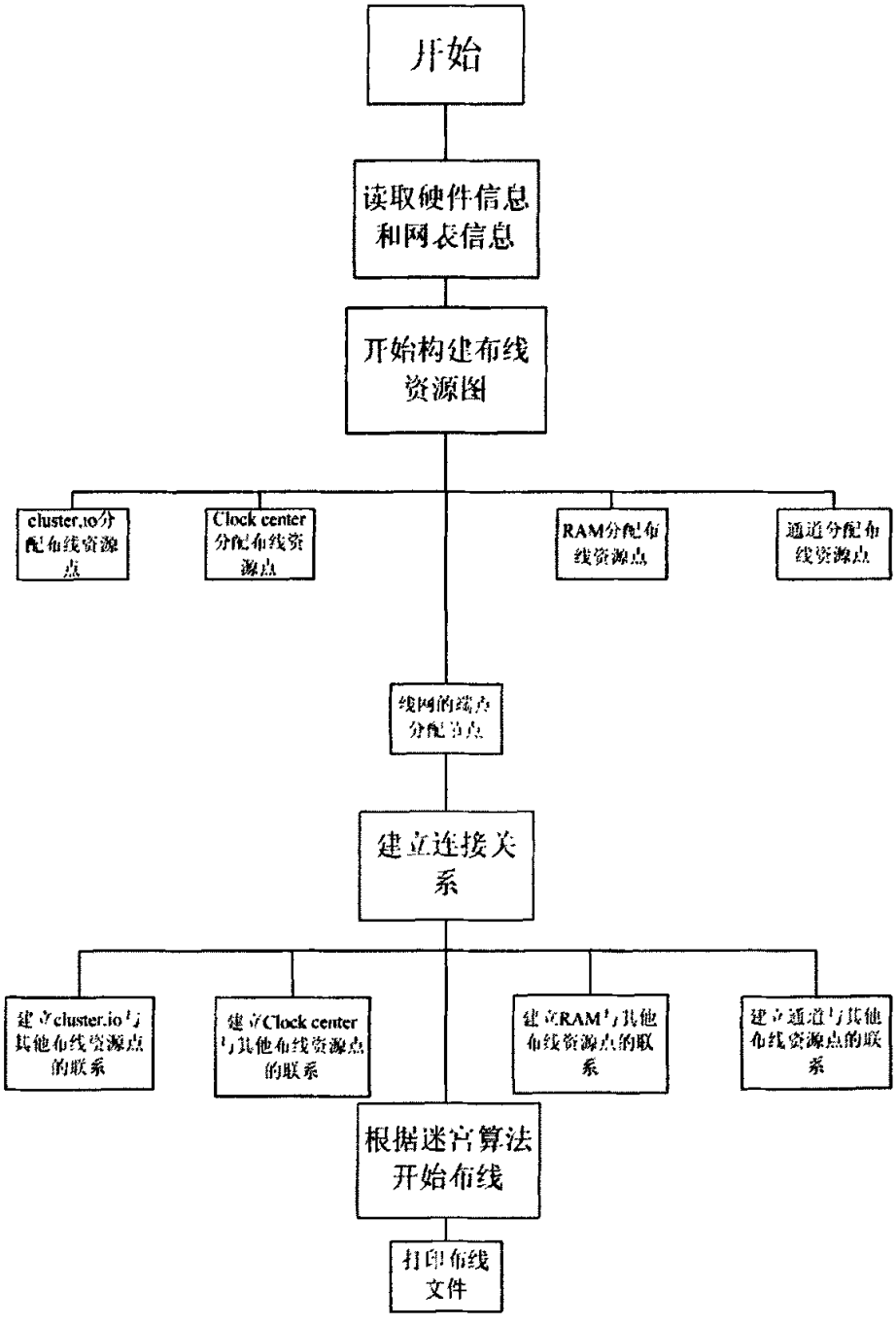

图 27 FDP250K 布线模块流程图

5.3 构建布线资源图

5.3.1. 概述

布线过程的后续操作均是在布线资源图上执行，如图 28 所示。此时布线过程就转化为在布线资源图上采用 Dijkstra 算法在有向图上寻找有效路径的过程。

布线资源图的构建分三部分。

第一部分：根据硬件结构文件对芯片的尺寸大小，逻辑单元的管脚种类和管脚数以及布线轨道数的描述，计算总共的节点数。

第二部分：根据硬件结构文件对布线资源的具体连接关系的描述，分配数据结构。

第三部分：在有了布线资源的具体连接关系数据结构后，根据芯片的尺寸大小对逻辑坐标进行一次遍历，分配节点之间的连接关系。

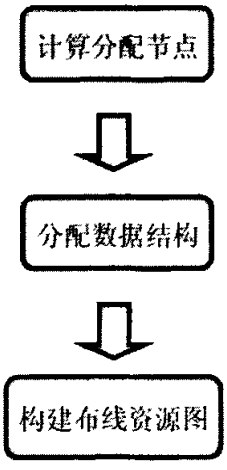

图 28 布线资源图的构建

5.3.2. FDP250K 布线资源图

在构建布线资源图的过程中，每一个硬件资源，包括逻辑单元的输入输出端口，每一条连线，都对应于一个节点。而每一个编程开关则对应于有向边。这样布线模块所用到的硬件资源都已经包括在布线资源图中。

图 29 列出一个 2 输入查询表及其连线资源和对应的布线资源图的例子。

在布线资源图中每一个节点都有各自的编号而这个编号的值是唯一的，也就是说只要知道节点的编号的值，就可以找到是哪一条布线轨道或逻辑单元的输入输出管脚。在另一方面，只要知道布线轨道或逻辑单元的输入输出管脚，就可以找到是哪一个节点。

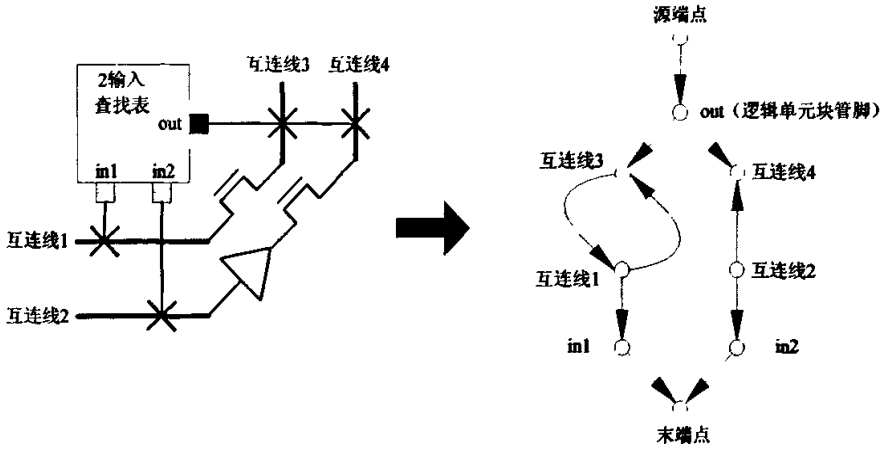

图 29 逻辑单元硬件连线资源和对应的布线资源图

在 FDP250K 中，CLUSTER 的硬件连线资源和对应的布线资源图的例子如图 30 所示

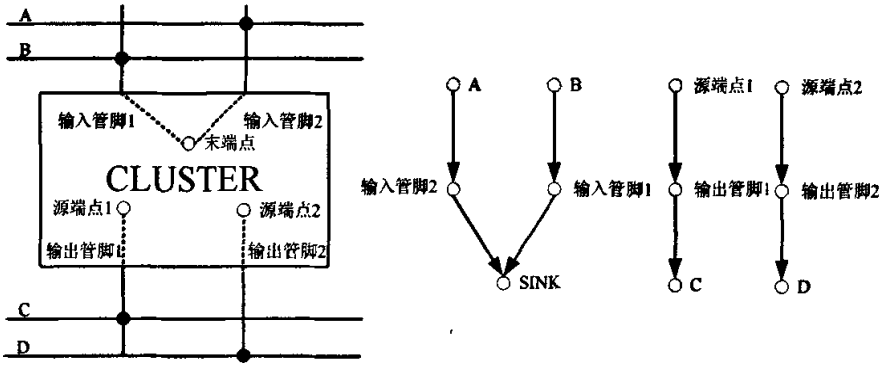

图 30 CLUSTER 的硬件连线资源和对应的布线资源图

在图中两个输入 IPIN 最后进入同一 CLUSTER，两个 PIN 的逻辑地位相同（CLASS 相同）时可以逻辑等效为拥有同一个虚拟的 SINK 点。

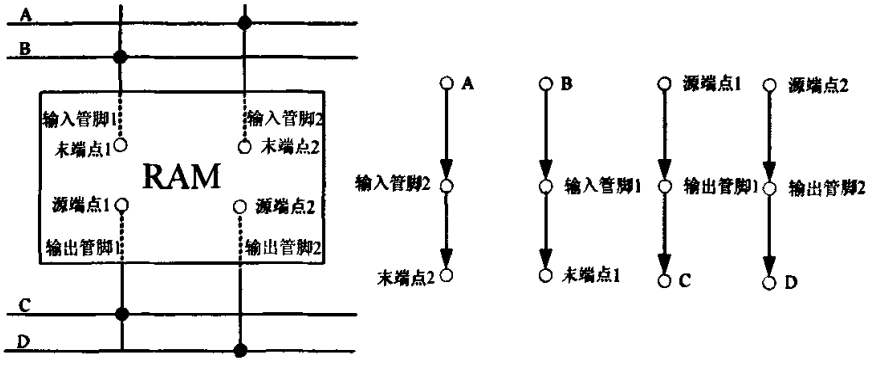

图 31 单块 RAM 的硬件连线资源和对应的布线资源图

FDP250K 芯片中内嵌了 RAM 模块，总容量为 16K（bits），分成 4 块，每块

4K（实际上在设计时每块 RAM 加入 1 位奇偶校验位，使用的是 512*9bits）。VPR 中没有对 RAM 的支持，我们把 RAM 当成和 CLUSTER 类似的元器件，在图 31 中 2 个输入 IPIN 最后进入同一 RAM，RAM 的 PIN 脚的 CLASS 均不同，所以拥有两个不同虚拟的 SINK 点。

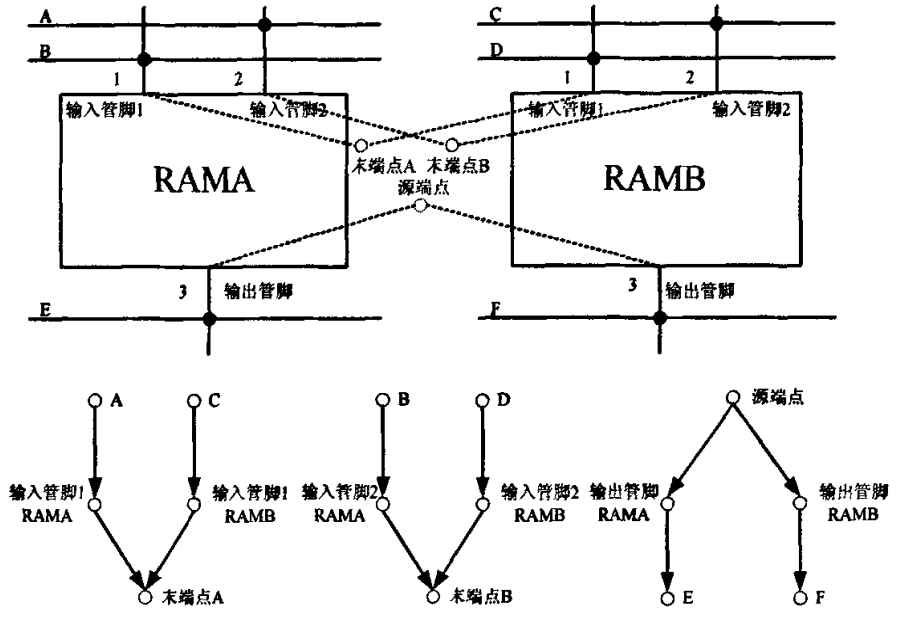

图 32 级联 RAM 的硬件连线资源和对应的布线资源图

在图 32 中 4 个输入 IPIN 最后进入 2 块级联的 RAM 时，同一逻辑位置上的 PIN 脚等价为同一虚拟点，可以看到同时进入 1 位置的 2 个 IPIN 拥有同一 SINK，同时进入 2 位置的 2 个 IPIN 拥有同一 SINK，同时从 3 位置发出的的 2 个 OPIN 拥有同一 SOURCE。

宏是由若干个 CLUSTER 所构成的矩形区域，特别要注意的一点是，当线网进入宏的 PIN 脚时，宏的每一个 PIN 脚都是逻辑不等效的，每一个线网从宏的 PIN 脚输入或者输出时都对应了一个不同的 SOURCE 或者 SINK。如图 33 所示宏模块的布线时采用动态构建的方法，宏模块本身是有已经构建好的一系列 CLUSTER 构成的，它们之间的相对位置已经固定，它们之间的互连也已经确定，所以不能象对待普通 CLUSTER 那样进行布线，普通的 CLUSTER 可以有逻辑等效这样的概念，而宏的 CLUSTER 不能包含逻辑等效，它们的端口逻辑是固定的，所以在构建布线资源图的时候，不能考虑宏的 CLUSTER 的管脚锁定问题。

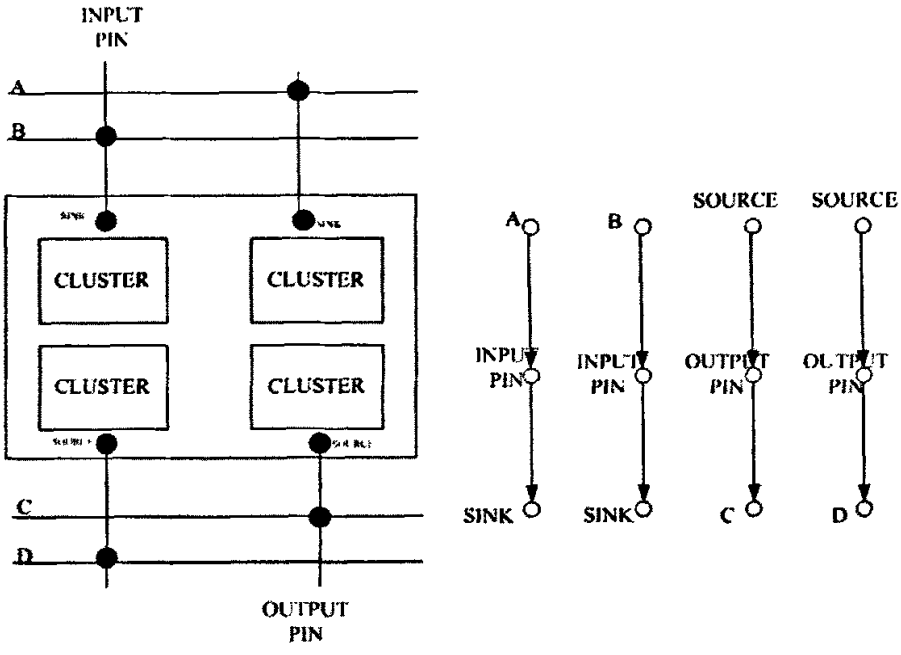

图 33 宏的硬件连线资源和对应的布线资源图

5.3.3. Cluster Connection Box (CCB)的构建

每个 CLUSTER 四周有 4 个 CB 负责将其四周的信号引入布线通道中。，其摆放位置如图 34 所示。最上面一行蓝色的 CB 负责连接第一行 CLUSTER 上边的信号；最下面一行绿色的 CB 负责连接第四行 CLUSTER 下边的信号；最左面一列粉色的 CB 负责左边第一列 CLUSTER 左边的信号；最右边一列红色的 CB 负责最右边一列 CLUSTER 右边的信号；黑色的 CB 负责将 CLUSTER 左右两侧的信号连入竖直布线通道；浅蓝色的 CB 负责将 CLUSTER 上下的信号连入水平布线通道。

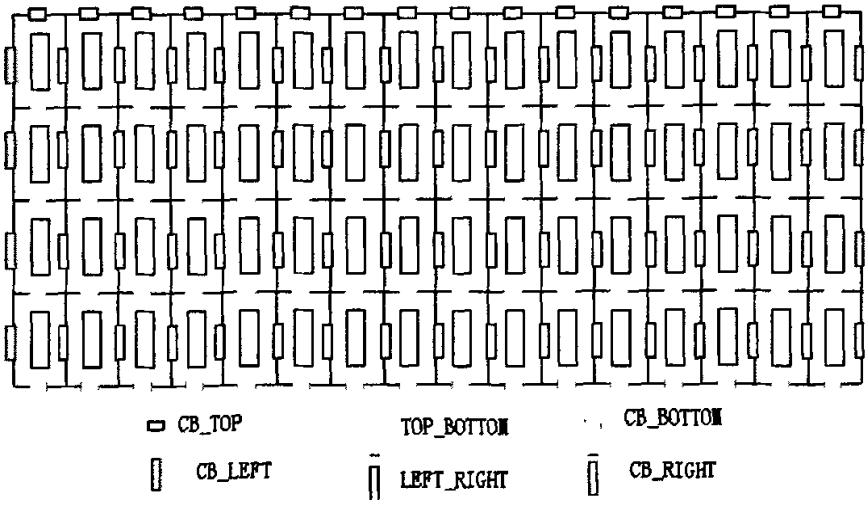

图 34 CCB 的构建

在分配 CLUSTER 上的管脚的连接关系时，还必须由管脚的方位信息及管脚的编号。这些都由芯片的结构文件给定且与芯片的硬件相对应。

CCB 的构建在于建立 CLUSTER 中的 OPIN 与 TRACK 的连接关系。也就是说，CLUSTER 中的 OPIN 是连接到哪几根 TRACK 上的。还有 TRACK 与 CLUSTER 中的 IPIN 的连接关系。也就是说，哪几根 TRACK 是连接到 CLUSTER 中的 IPIN 的。

5.3.4. 布线资源的坐标定义

芯片中有很多类似于图 29 的布线资源图。为了可以有效的合理的重复利用图 29 对布线资源的描述，芯片被划分成四大模块：CLUSTER, CHANX, CHANY, IO。每一个 CLUSTER, CHANX, CHANY, IO 都给定了与之对应的逻辑坐标。坐标表示方法如图 35 所示。坐标的原点和轨道的起始点都在左下角。也就是说，在水平方向的通道中，最左边的通道的 x 坐标为 0；在垂直方向的通道中，最下边的通道的 y 坐标为 0。在 CHANX 中，轨道 0 是最下面的轨道；在 CHANY 中，轨道 0 是最左边的轨道。

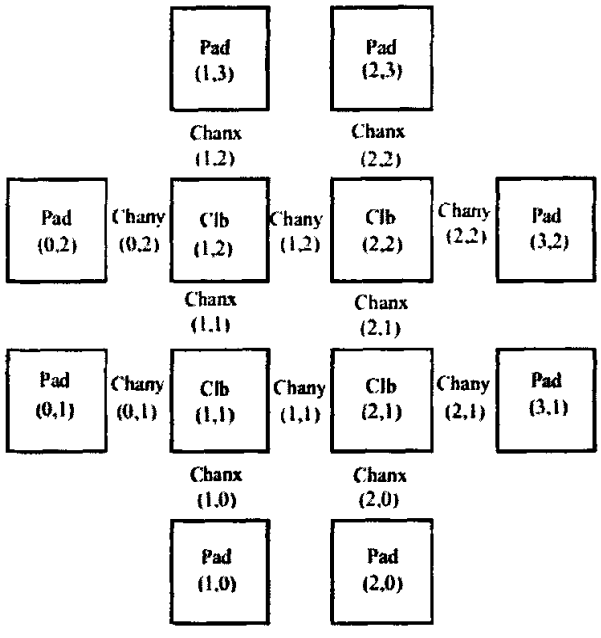

图 35 布线资源的坐标表示

5.4 迷宫算法

在分配好布线资源以及建立好它们之间的连接关系以后，我们运用迷宫布线算法来连接每一根线网的端点。一个迷宫布线算法本质上包含 Dijkstra 算法的运行，从而在一张布线资源图中找到在一根线网起始端和终止端之间的最短路径（最低的整体花费）。所有这些算法执行多次布线迭代，在迭代中，一些或者所

有线网被拆线重布到不同的路径上，以解决布线资源的竞争或者改进电路的速度。

Dijkstra 算法是典型最短路径算法，用于计算一个节点到其他所有节点的最短路径。主要特点是以起始点为中心向外层层扩展，直到扩展到终点为止。Dijkstra 算法能得出最短路径的最优解，但由于它遍历计算的节点很多，所以效率低。Dijkstra 一般的表述通常有两种方式，一种用永久和临时标号方式，一种是用 OPEN，CLOSE 表方式。

Dijkstra 算法的基本过程为：

创建两个表，OPEN，CLOSE。

OPEN 表保存所有已生成而未考察的节点，CLOSED 表中记录已访问过的节点。

1. 访问路网中里离起始点最近且没有被检查过的点，把这个点放入 OPEN 组中等待检查。
2. 从 OPEN 表中找出距起始点最近的点，找出这个点的所有子节点，把这个点放到 CLOSE 表中。
3. 遍历考察这个点的子节点。求出这些子节点距起始点的距离值，放子节点到 OPEN 表中。
4. 重复 2，3，步。直到 OPEN 表为空，或找到目标点。

5.5 路径探索算法

VPR 算法是一个经典的 FPGA 布线算法，然而这是一个学术上才能使用的算法，对于我们的芯片来讲，包含了很多的模块，例如 RAM 和宏模块，以及一些专用的互连资源，比如总线等，而且线型也不一样，在上述的例子中，每个节点到相邻节点（即每一条边）都会有其相应的代价，我们借助 VPR 路径探索算法，以及它的代价因子，对于 VPR 中的线型，由于 VPR 中不同的线型的代价一样，对于我们的芯片来讲不合适，所以我们的每一个节点下面都有其累积代价。每一条边的代价如下表所示：

布线资源，n	Base Cost,b(n)
二倍线	1
四倍线	1.1
长线	1.2
CLUSTER 输出管脚	1
CLUSTER 输入管脚	0.95
源节点（SOURCE）	1
终端（SINK）	0

逻辑单元 1 输入引脚和漏的 b(n)值被设为小于 1 以节省 CPU 运行时间。由于迷宫布线算法常常是在碰到对应于线网其中一个终端的 SINK 时把一个终端连接起来，这样在迷宫扩展时 SINK 更容易达到，从而节省 CPU 的运行时间。换句话说，我们希望尽量使用逻辑单元的输入引脚和 SINK，迷宫布线算法会检查某

个布线资源线段之后的逻辑单元是否包含一个线网的 SINK，这样在找到 SINK 之前就不需要扩展到更多的布线资源。为了达到这个目标，逻辑单元输入引脚的基本代价略小于 1，SINK 的基本代价为 0。由于拥挤度不会出现在 SINK，所以令所有 SINK 的值为 0 不会导致布线失败。

而每个节点下面的累积代价是基于考虑拥挤度的路径探索算法（Pathfinder negotiated congestion algorithm)；我们将每个结点的代价定义如下，当连接布线资源 n 到布线资源 m 时，使用布线资源 n 的代价为：

$$Cost(n)=b(n)*h(n)*p(n)$$

b(n)是结点 n 的基本开销，h(n)是结点 n 的历史拥挤度；在每一次结点 n 被重用并且留给布线器“拥挤记忆”的布线迭代发生之后，h(n)被增加。p(n)是结点 n 的当前拥挤度开销；如果用这个结点对当前连线布线不会造成任何的重用，p(n)是 1，并且随着结点重用的数目的增加而增加。p(n)也是一个已经执行过的布线迭代的次数的函数。在早期的迭代中，p(n)伴随着当前结点 n 的重用缓慢增长；在以后的迭代中，p(n)随着结点 n 的重用非常快地增加。

需要指出的是，对带有宏的例子时进行布线时，与常规的不带宏的例子步骤上略有区别，具体如下：

1. 读取后端库，记录下宏中所要占用的线网
2. 通过 map 函数在所建立的布线资源点中去除掉宏内线网所占用的布线资源点
3. 进行普通布线
4. 布线文件输出包括普通布线线网布线信息和宏内的线网信息（程序可自动判断是否运用到宏，如没有用到，则不输出宏内的线网信息）

5.6 布线结果

我们的芯片结构如图 36 所示

我们跑了 24 位乘法器，结果如图 37 所示：

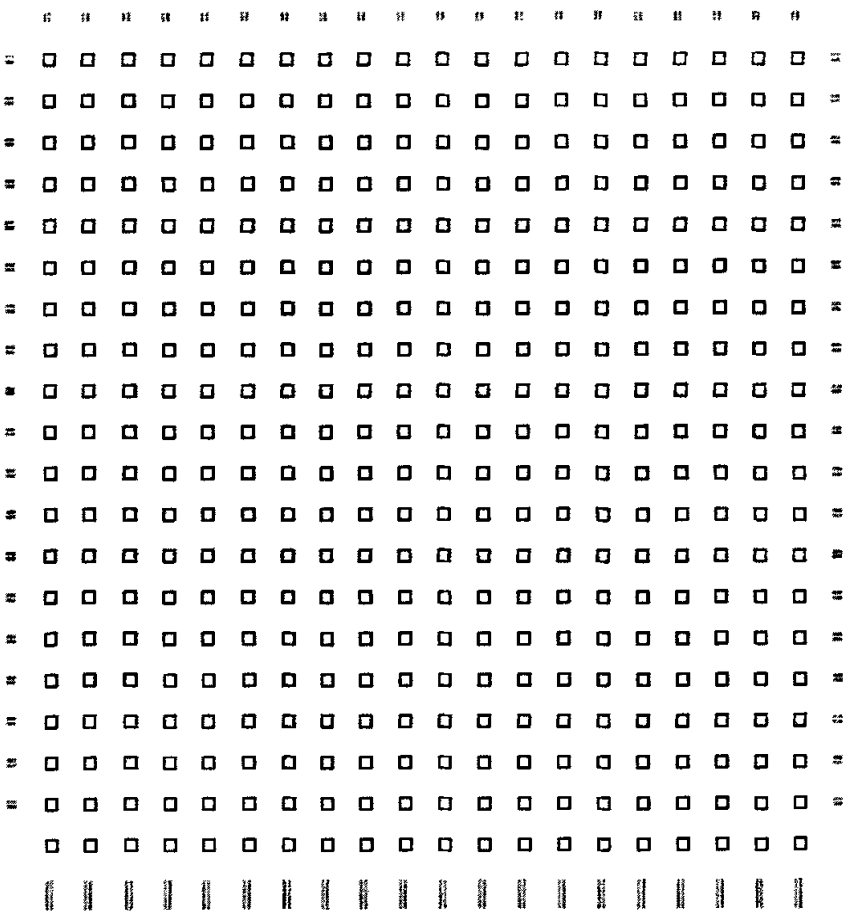

图 36 FDP250K 芯片结构图

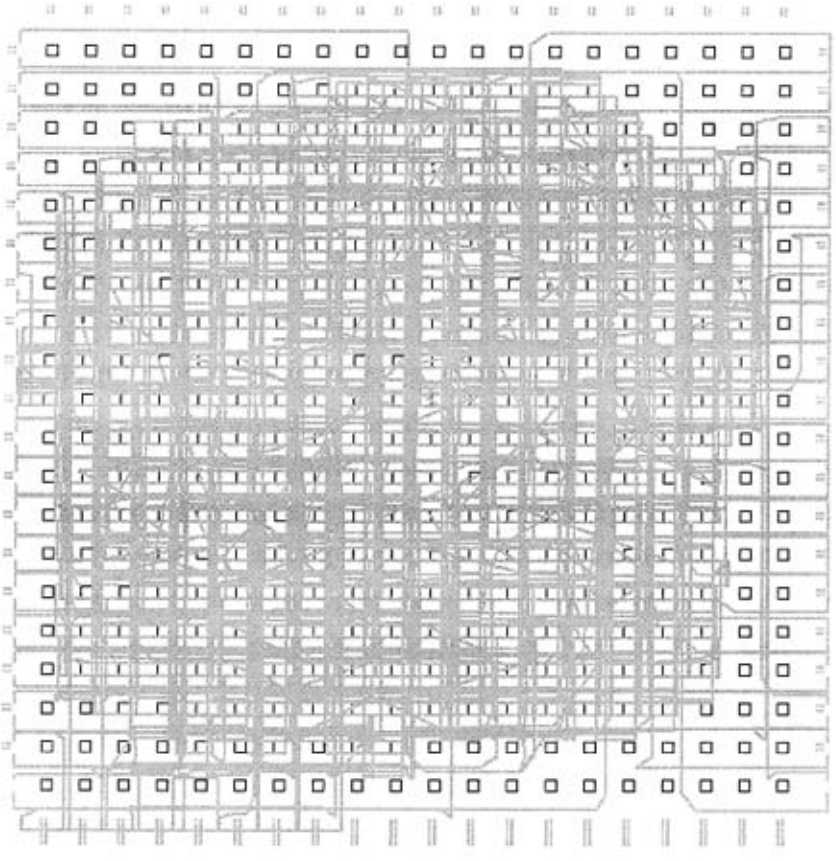

图 37 24 位乘法器布线结果

5.7 本章小结

本章实现了 FDP250K 的布线系统，该布线系统能支持时钟，RAM，BUS，以及宏模块的布线。并给出了布线结果的例子。

第六章 总结与展望

首先本论文根据 FPGA 的特性进行了分析，对以前的相关研究进行了总结，针对 FPGA 互连的特性，提出了互连延时模型。该模型对 FPGA 中的单个传输管进行了建模，能够得到解析表达式。实验结果表明了该模型的有效性和实用性

本论文主要完成了以下几个工作

1、提出了一种 FPGA 互连资源 MOS 管斜坡输入时分析时延模型，并且提出了计算 MOS 管级连时的等效电容的计算方法，和迭代算法例如 SPICE 比较，我们的算法快速，并且足够精确。可以被用在 FPGA 时驱布局，时驱布线中。

2、对 dogleg 现象就行了探索，发现在 SUBSET 开关盒情况下，用 dogleg 能比不用 dogleg 提高 11%的布通率，然而在 WILTON 条件下，改变不是很多，这是因为 WILTON 开关的灵活度已经比较高，所以提高的幅度相对较小。

3、建立了 FDP250K 的布线系统，能够支持总线的布线，时钟的布线以及 RAM 的布线，还有 IO 环的布线。并且支持 RAM 的级联，包括了字扩展和位扩展。通过动态构建布线资源图来处理硬宏模块。

对于将来的工作，我作了如下展望

1、本文提出的 MOS 管模型虽然相对以前的模型有了很大的改进，然而仍然比较复杂，精度虽然提高了不少，但相对 SPICE 来讲，仍然有很大的提高余地，如果能够降低复杂度，提高精度，那么时序驱动的布线能使性能提升很多。

2、对于 FDP250K 布线系统还有待提高的地方，和商用 FPGA 设计软件相比，如何提高布线速度是一个重要的方面。

3、另外区域布线也是一个急待实现的方面。

4、在 FPGA 的实际系统中，测试也是非常重要的一环，而 FPGA 互联的测试在整个 FPGA 测试中占了很大的部分，FPGA 布线的测试不仅能找到互联中的错误，也能发现电路时序方面的错误，所以做好 FPGA 布线测试对于 FPGA 系统来讲就显得尤为重要了，在 FPGA 布线中如何进行方便的测试也是一个重要的研究方面，比如增量布线，当部分线已经布好以后，需要接着往下布线，我们就需要增量布线，保留已经布好的部分，然后从已经布好的部分开始布线。

参考文献

[1] 冯涛，王程. 可编程逻辑器件开发技术——MAX+plusII 入门与提高[M].北京：人民邮电出版社，2002：2.

[2] 王波. 可编程 IP 核布图方法研究[D].上海：上海复旦大学，2004

[3] Dwight D. Hill, "A CAD System for the Design of Field Programmable Gate Arrays", Proc. of the 28th Design Automation Conference, June 1991, pp. 187-192.

[4] Guy G. Lemieux, Stephen D. Brown, "A Detailed Routing Algorithm for Allocating Wire Segments in Field-Programmable Gate Arrays", ACM/SIGDA Physical Design Workshop, 1993, pp. 215-226

[5] 洪先龙，严小浪，乔长阁，《超大规模集成电路布图理论与算法》

[6] Amit Chowdhary, etc. "Detailed Routing of Multi-Terminal Nets in FPGAs", 7th International Conference on VLSI Design, 1994

[7] Yachyang, etc. "Routing for Symmetric FPGAs and FPICs", Proc. of ICCAD, 1993

[8] Vaughn Betz, Jonathan Rose, "VPR: A New Packing, Placement and Routing Tool for FPGA Research", Int. Workshop on Field-Programmable Logic and Applications, 1997, pp.213-222

[9] C. Ebeling, L. McMurchie, S. A. Hauck and S. Burns, "Placement and Routing Tools for the Triptych FPGA,", IEEE Trans. On VLSI, Dec. 1995, pp. 473-482.

[10] Y.- L. Lee, A. Wu, "A Performance and Routability Driven FPGA Router for FPGAs Considering Path Delays," DAC, 1995, pp. 557-561.

[11] J. Frankle, "Iterative and Adaptive Slack Allocation for Performance-Driven Layout and FPGA Routing," DAC, 1992, pp. 536-542

[12] C. Y. Lee, "An Algorithm for Path Connections and its Applications," IRE Trans. Electron. Comput., Vol.EC =10, 1961,pp. 346-365.

[13] Vaughn Betz, Jonathan Rose, Alexander Marquardt. "ARCHITECTURE AND CAD FOR DEEP-SUBMICRON FPGAs," KLUWER ACADEMIC PUBLISHERS, 1999.

[14] Zhou Feng, etc, An Analytical Delay Model for SRAM-Based FPGA Interconnections, Design Automation Conference, 1999. Proceedings of the ASP-DAC'99. 18-21 Jan. 1999 pp. 101-104 vol.1

[15] Muhammad Khellah, Stephen Brown, and Zvonko Vranesic, Modelling Routing

Delays in SRAM-Based FPGAs, Proc. Canadian Conf. on VLSI, 1993, pp. 1013-1029

[16] W. C. Elmore, The transient response of damped linear networks with particular regard to wide-band amplifiers, J. Appl. Phys. 19(1948), pp.55-63.

[17] J. Rubinstein, etc, Signal delays in RC tree networks, IEEE Trans. Computer-Aided Des.CAD-2(1983), pp. 202-211.

[18] Behzad Razavi, Design of Analog CMOS Integrated Circuits, McGraw- Hill, 2001.

[19] Muzhou Shao, Martin D. F. Wong, and Huijing Cao, Explicit Gate Delay Model for Timing Evaluation, ACM, pp. 32-38, 2003.

[20] Andrew B. Kahng, Kei Masuko, and Sudhakar Muddu, Analytical Delay Models for VLSI Interconnects Under Ramp Input, International ICCAD Conf. (1996), pp. 30-37.

[21] Wang Yi, Wang Lingli, Han Ruonan, and Tong Jiarong, A Delay Model for SRAM-Based FPGA Interconnections, International Midwest Symposium on Circuits and Systems. (2006)

[22] 王怡，王伶俐，韩若楠，唐璞山，童家榕等，基于 SRAM 的 FPGA 延时模型，电路与系统学报，2006

[23] Rohini Gupta, Bogdan Tutuianu, and Lawrence T. Pileggi, The Elmore Delay as a Bound for RC Trees with Generalized Input Signals, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 16, NO. 1, JANUARY 1997.

[24] Chandramouli V. Kashyap, Charles J. Alpert, Frank (Ying) Liu, and Anirudh Devgan, Closed-Form Expressions for Extending Step Delay and Slew Metrics to Ramp Inputs for *RC* Trees, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 23, NO. 4, APRIL 2004.

[25] Chang Woo Kang, Ali Iranli, and Massoud Pedram, Technology Mapping and Packing for Coarse-Grained, Anti-fuse Based FPGAs, ASPDAC, 2004, pp 209-211.

[26] E. Bozorgzadeh S. Ogrenci-Memik &I. Sarrafzadeh, RPack: Routability-Driven packing for cluster-based FPGAs, ASPDAC, 2001, pp 629-634.

[27] D.N.Deutch, A "Dogleg" Channel router, Proc, 13th DAC p425.1976.

[28] 胡云，王伶俐，唐璞山，童家榕. 基于概率增益的电路划分算法，电子与信息学报

[29] 胡云，王伶俐，唐璞山，童家榕. 基于布通率的 FPGA 装箱算法，计算机图形学与辅助设计学报，第 19 卷，第一期，108～113 页，2007

[30]A. Marquardt et al., "Using cluster-based logic blocks and timing-driven packing to improve fpga speed and density," in Proc. International Symposium on FPGAs, pp. 37–46, 1999.

[31]Dwight D. Hill, "A CAD System for the Design of Field Programmable Gate Arrays", Proc. of the 28th Design Automation Conference, June 1991, pp. 187-192.

[32]S. Kirkpatrick, C. Gelatf and M. Vecclu, "Optimization by Simulated Annealing," Science, May 1983, pp. 671-680.

[33]M. Wang, X. Yang, and M. Sarrafzadeh. "Congestion Minimization During Placement". IEEE Transactions on Computer Aided Design, 19(10):1140–1148, 2000.

[34]M. Wang, X. Yang, and M. Sarrafzadeh. "Dragon2000: Fast Standard-cell Placement for Large Circuits". In International Conference on Computer-Aided Design, pages 260–263. IEEE, 2000.

[35]Seokjin Lee and Martin D. F. Wong. "Timing-Driven Routing for FPGAs Based on Lagrangian Relaxation". IEEE Transactions on Computer Aided Design, VOL. 22, NO. 4, 2003

[36]Gi-Joon Nam, Karem A. Sakallah, and Rob A. Rutenbar , "A New FPGA Detailed Routing Approach Via Search-Based Boolean Satisfiability" IEEE Transactions on Computer Aided Design, VOL. 21, NO. 6, JUNE 2002

[37]Hasan Arslan and Shantanu Dutt, "ROAD : An Order-Impervious Optimal Detailed Router for FPGAs", ICCD, 2003

[38]S. Sun, D. Du and H Chen, "Efficient Timing Analysis for CMOS Circuits Considering Data Dependent Delays," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, No. 6, June 1998, pp. 546-552.

[39]D. Blaauw, V. Zolotov, S. Sundareswaran, C. Oh, and R. Panda, "Slope Propagation in Static Timing Analysis," 2000 IEEE/ACM Intl. Conf. On Computer-Aided Design, 2000 pp. 338-343.

[40]http://www.xilinx.com/

[41]http://www.altera.com/

致 谢

光阴似箭，寒暑三易，在复旦短暂的硕士生涯即将结束，这期间，有痛苦，也有欢乐，有平淡，也有兴奋。值论文结束之际，我首先要衷心的感谢我的导师童家榕教授。是童老师前瞻性的学术眼光，才使我有幸涉足 FPGA 这一领域。在我攻读硕士学位的三年期间，他给我确定了正确的研究方向，并且指导我完成了多项科研项目。我的学习和研究任务的顺利完成，得益于童老师精心的指导和诚挚的鼓励。童老师积极创新的开拓精神，严谨治学的科学态度和真诚热情的待人接物，也将一直激励着我在科学的道路上积极地探索。

我要特别感谢王伶俐副教授。王老师给予我很多的指导和帮助。王老师渊博的学识，一丝不苟的科研精神和宽以待人的作风将使我终生受益。另外，感谢杨萌助理研究员，平时杨萌对于我的研究提出了非常宝贵的意见，经常和我一起讨论。感谢唐璞山教授，黄均鼐教授，来金梅副教授对我在学业方面的支持。同时也感谢谭道珍老师对我生活中的关怀和帮助。

我还要感谢陈利光博士，在我三年科研的过程中，他多次和我一起研究、讨论，使我的研究工作受益非浅。还有一同求学的胡欣同学，张军营同学，熊伟同学、魏萌、杨铭同学等，也提供了很多不可或缺的帮助。

我要感谢 CAD 实验室所有才华横溢的研究生以及已经毕业离开实验室的师兄师姐。大家共同创造了开放的、学术氛围浓厚的工作环境，使我从中获益良多。

我要特别要感谢所有参与过 FDP250K 课题组的伙伴们，包括软件组的胡云、胡子夏、邢红、张小颖、代莉、王侃文、谈君、章淳和硬件组的陈利光、王元、王亚滨、侯慧、王健等同学，本文完成的工作中凝聚着所有课题组成员付出的心血和汗水。

最后，谨以此文献给我的父母，感谢他们长久以来的教导以及对我学业上的支持和鼓励。

攻读硕士学位期间发表论文的清单

[1] Wang Yi, Wang Lingli, Han Ruonan, and Tong Jiarong, A Delay Model for SRAM-Based FPGA Interconnections, International Midwest Symposium on Circuits and Systems. (2006)

[2] 王怡，王伶俐，韩若楠，唐璞山，童家榕等，基于 SRAM 的 FPGA 延时模型，电路与系统学报（已录用）

FPGA布线算法的研究

作者：王怡
学位授予单位：复旦大学
被引用次数：2次

参考文献(1条)

1. 王波 可编程IP核布图方法研究[学位论文]硕士 2004

本文读者也读过(10条)

1. 孙虎 FPGA自动布局布线算法[学位论文]2008
2. 胡欣 FPGA布线研究与实现[学位论文]2007
3. 赵刚 FPGA结构和布局布线算法研究[学位论文]2008
4. 陈苑锋 FPGA评估系统布局布线模块设计[学位论文]2006
5. 李国平 FPGA低功耗布局布线算法的研究与改进[学位论文]2007
6. 刘战 几种用于FPGA的新型有效混合布线算法[学位论文]2007
7. 杨铭 FPGA布局算法研究和软件实现[学位论文]2007
8. 徐新民 FPGA资源动态重构及低功耗研究[学位论文]2007
9. 施小祥 动态可重构FPGA的布局布线算法研究[学位论文]2007
10. 王石 基于FPGA芯片的功能仿真平台构建及静态时序分析[学位论文]2008

引证文献(1条)

1. 沈静静 用于FPGA的新型混合布线算法的研究[学位论文]硕士 2009