一种应用图论方法管理可重构资源的策略

张宏烈[1,2] 张国印[1] 丛万锁[2] 胡海燕[2]
(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)[1] (齐齐哈尔大学计控学院 齐齐哈尔 161006)[2]

摘 要 可重构硬件资源的管理是可重构操作系统的一个首要任务。提出了一种基于图论技术的管理空闲资源的UPFS算法。其核心思想是将FPGA的空闲区域映射成无向图,在无向图中运用邻接矩阵和方向矢量交角等概念,求解最大回路和通路,最终找到满足条件的最大空闲矩形集。仿真实验表明,UPFS算法与已有算法相比,能有效减少系统资源浪费,降低系统硬件布局时间,是可行的管理策略。

关键词 可重构资源管理,图论技术,最大空闲矩形集
中图法分类号 TP302 **文献标识码** A

One Management Strategy of Reconfigurable Resource Using Graph Theory

ZHANG Hong-lie[1,2] ZHANG Guo-yin[1] CONG Wan-suo[2] HU Hai-yan[2]
(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)[1]
(College of Computer and Control Engineering, Qiqihar University, Qiqihar 161006, China)[2]

Abstract The management of reconfigurable resource is an important task for operating system of reconfigurable system. This paper presented UPFS algorithm for FPGA based on graph theory. The main idea of UPFS is the free space of FPGA mapped to undigraph, and then calculated the biggest loop and chain using adjacency matrix and direction vector inclination in undigraph concept, at last we found a set of the biggest rectangular satisfied the conditions. The simulation results show that, compared with existent algorithms, UPFS is feasible management strategy, which can reduce the waste of system resource and decrease the time of hardware distribution.

Keywords Management of reconfigurable resource, Graph theory, Set of the biggest rectangular

可重构系统通常由一个主处理器和可重构硬件组成。当前主流的可重构硬件是基于SRAM(Static RandomAccess Memory)的FPGA,其配置信息存储在SRAM中,具有可任意次配置、容量大、配置速度快等优点。可重构系统根据应用程序的要求,在FPGA上能够实现配置并执行各种电路,而且不影响当前运行的其他电路,为了表达其动态属性,这种电路被记为硬件任务。由于允许在运行时间内,对FPGA芯片进行部分可重构,因此可重构硬件资源的动态管理及合理分配的问题变得越来越重要。目前在可重构系统尤其是运行时可重构系统的应用中面临的最大困难就是寻找合适的操作系统支持,只有合适的操作系统,才能充分利用可重构系统的软、硬件资源达到系统的最佳性能。

可重构硬件资源的管理是影响动态可重构系统性能的关键因素之一,管理实时变化的可重构硬件资源是可重构操作系统的一个首要任务。可重构硬件资源的管理主要包括任务布局和任务调度两个方面。一个任务被映射到芯片的哪个位置,其布局策略决定了芯片表面碎片的多少,高碎片率能够导致难以想象的情形,尽管有足够的可用空间,但是却无法布局任务。为了避免这样的情况发生,需要一种组织任务布局的服务机制。这项工作主要采用FPGA的任务在线布局算法,其目标是高质量的布局和布局算法的有效实施。

1 相关工作

图1的系统由CPU和部分可重构FPGA组成。FPGA提供了可重构的逻辑资源,此资源是由二维阵列的可重构逻辑块组织的。CPU执行操作系统功能,处理可重构系统资源。在FPGA上布局和执行之前,到达的任务存储在队列中。调度决定接下来队列中哪一个任务应该载入和执行并呼叫布局器找到可行的定位。布局器处理FPGA的空闲空间并试图找到可行的任务布局。每当布局器为任务找到了可行的定位,装载机就将任务装载并执行。任务完成后,释放所有被任务占用的可重构资源[1]。

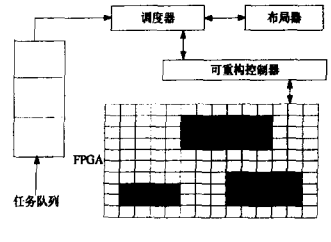


图1 系统模型

到稿日期:2010-01-08 返修日期:2010-03-21 本文受黑龙江省自然科学基金项目(B2007-07)资助。
张宏烈(1966—),女,博士生,教授,主要研究方向为嵌入式系统与FPGA应用、可重构计算等研究,E-mail:zhang_honglie@163.com;张国印(1962—),男,博士,教授,博士生导师,主要研究方向为嵌入式系统、网络与信息安全;丛万锁(1980—),男,硕士生;胡海燕(1984—),女,硕士生,主要研究方向为嵌入式系统与可重构计算。

目前处理任务布局问题的研究细分为两类：离线和在线方案。在线方案的布局需要保证任务在任何时间到达都能执行，为了达到这个目的，必须管理 FPGA 上的空闲面积。管理空闲面积最简明的方法是标记已使用的或未使用的 CLB (Configurable Logic Block))来核实一个正在到达的任务的所有可能定位。对于 $H \times W$ 个 CLB 规模的 FPGA 来说，有 $H \times W$ 范围的可能性布局。为了减少潜在的大量的定位数量，提高布局效率，Barzargan 等人在文献[2]中提出了基于 FPGA 空间的空闲矩形布局方案。从空闲矩形的模型出发，目前管理 FPGA 中的空闲资源主要有 3 种方式：不重叠空闲矩形列表、最大空闲矩形列表和顶点列表[3]。

Bazargan 提出的算法就是通过维护不重叠空闲矩形列表来管理可重构芯片的资源[2]，布局算法布局质量较高，但是这个方法必须为 n 个布局的任务处理复杂度为 $O(n^2)$ 的矩形并且任务的插入和删除很难执行。文献[4]使用最大空闲矩形列表管理空闲区域，与不重叠空闲矩形列表相比，其资源利用率较高，但是寻找可重构芯片中最大空闲矩形的完全集却很耗时。Handa 提出阶梯算法来寻找最大空闲矩形的完全集[5]，随后 Welder[1] 和 Amandine[6] 对阶梯算法进行了改进和变形，但都存在时间复杂度高的问题。Jin Cui[7] 提出基本扫描线算法，实验结果表明，其平均运行时间优于阶梯算法，但基本扫描线算法仍有很多不足，主要是冗余计算和重复计算的问题。

针对上述方法存在的问题，本文另辟蹊径，考虑将无向图运用到可重构资源管理模型中，提出了用图论技术解决最大空闲矩形搜索问题的无向图划分自由空间 UPFS(Undigraph partition free space)算法。该方法将 FPGA 中空闲资源形成的不规则区域映射成无向图，然后利用邻接矩阵和方向矢量交角，找到无向图的最大回路，并在此基础上给出快速寻找最大空闲矩形的方法。该方法运用图论中回路和通路等概念划分空闲区域，数据结构简单，实现容易，大大降低了平均运行时间和计算复杂度。

2 空闲区域向无向图的映射方法

定义 1(无向图[8]) 一个无向图 G 是一个有序二元组 $G=(V,E)$，V 是顶点的有穷集合，E 是边的集合。

例如，图 2 所示的无向图为 $G=(V,E)$，$V=\{V_1,V_2,V_3,V_4,V_5,V_6,V_7\}$，$E=\{\langle V_1,V_2\rangle,\langle V_2,V_3\rangle,\langle V_3,V_5\rangle,\langle V_4,V_1\rangle,\langle V_5,V_4\rangle,\langle V_6,V_3\rangle,\langle V_6,V_7\rangle,\langle V_7,V_2\rangle\}$。

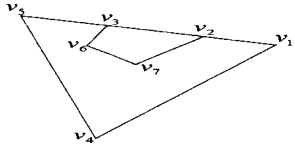


图 2 无向图

定义 2(通路[8]) 设 $G=(V,E)$ 是一个无向图，A,B 是 V 中的顶点，则 G 中从 A 到 B 的一条通路是一个边的序列 $E_1,E_2,\cdots,E_k(k\geqslant 1)$，该序列满足下列条件：

(1)$A\in E_1$，$B\in E_k$；

(2)对于所有 $1\leqslant i<k$，$E_i\cap E_{i+1}\neq\phi$。

引论 1 若从 A 到 B 经过 $A_1,A_2,\cdots,A_K$，K 个顶点，则从 A 到 B 的通路可以表示为 $C=(A_1,A_2,\cdots,A_K)$。

定义 3(回路[8]) 设 $G=(V,E)$ 是一个无向图，G 中与某一顶点 V_i 相关联的边数(自回路算作 2 条)称为顶点 V_i 的度数，记为 $TD(V_i)$。通常称度数为 1 的顶点为悬挂顶点，与它关联的边为悬挂边。在由相关顶点和边构成的子图中，每个顶点的度数都为 2 的连通图称为回路，记为 L_i，例如图 2 所示 $L1=(V_2,V_3,V_6,V_7,V_2)$ 是图 G 中的回路。G 中能包容全部顶点和边的回路称为最大回路。如图 2 所示，$L2=(V_1,V_2,V_3,V_5,V_4,V_1)$ 是最大回路。

定义 4(方向矢量交角) 设 $G=(V,E)$ 是一个无向图，如图 3 所示，当沿边 $\langle V_i,V_j\rangle$ 搜索到 V_j 时，将与 V_j 相关的各边按搜索方向矢量化，求出搜索边矢量 V_jV_n 相对于当前路径边矢量的交角，即为方向矢量交角。

在图 3 中，当 V_i 点沿边 $\langle V_i,V_j\rangle$ 搜索到 V_j 点后，将边 $\langle V_j,V_k\rangle$，$\langle V_j,V_m\rangle$，$\langle V_j,V_n\rangle$ 按搜索方向矢量化，如图 4 所示，则搜索边矢量相对于 V_iV_j 的方向交角为 $\alpha_1,\alpha_2,\alpha_3$。方向交角有正有负，图 4 中 α_1,α_2 为正，α_3 为负。特殊情况下，当搜索边矢量 V_jV_n 与当前路径边矢量垂直或水平时，其方向交角为特殊角，如图 4 所示，可以得出 $\alpha_1=+90°$，$\alpha_2=0°$，$\alpha_3=-90°$。

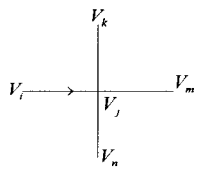


图 3 矢量化前无向图

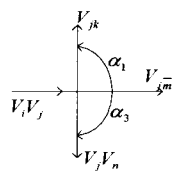


图 4 按搜索方向矢量化

定义 5(邻接矩阵[8]) 设 $G=(V,E)$ 是一个无向图，G 中诸顶点间邻接关系用 $M=[M_{ij}]_{v\times v}$ 矩阵来描述，其中元素 M_{ij} 定义如下，M 称为邻接矩阵。

$$M_{(i,j)}=\begin{cases}1, & \langle V_i,V_j\rangle\in(G)\\ 0, & \langle V_i,V_j\rangle\notin(G)\end{cases}$$

对于无向图 G 而言，M 是对称方阵，且主对角线元素全为 0。图 2 所示无向图的邻接矩阵 $M1$ 为：

$$M1=\begin{bmatrix}0&1&0&1&0&0&0\\1&0&1&0&0&0&1\\0&1&0&0&1&1&0\\1&0&0&0&1&0&0\\0&0&1&1&0&0&0\\0&0&1&0&0&0&1\\0&1&0&0&0&1&0\end{bmatrix}$$

定义 6(FPGA 区域模型[4]) 基于 FPGA 的可重构芯片可看作是由 W 行 H 列基本配置逻辑单元 CLB 构成的二维阵列，W,H 分别是可重构芯片单行、单列具备 CLB 的个数。所以可以将 FPGA 抽象成为宽为 W、高为 H 的矩形区域模型，包含 $W\times H$ 个 CLB。

如图 5 所示，在该矩形区域模型中，每个单元的位置编号假设为 (i,j)，其中 $1\leqslant i\leqslant W$，$1\leqslant j\leqslant H$；每个单元的使用状态有两种：空闲状态和占用状态，假设建立一个二维数组 $M[W][H]$ 来保存每个单元的使用状态，那么有如下定义：

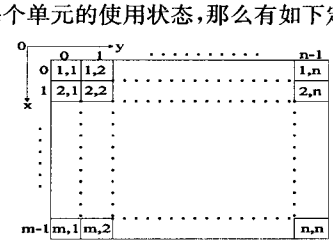


图 5 FPGA 区域模型

$$M[i][j]=\begin{cases}1, & (i,j)\text{为空闲状}\\ 0, & (i,j)\text{为占用状态}\end{cases}$$

也就是说若 $M[i][j]=1$ 表示编号为(i,j)的 CLB 单元为空闲状态，否则被占用。

定义 7(空闲连通区域) 设存在 $W\times H$ 的矩形区域，对于任意空闲单元(i,j)，若存在多个空闲单元连续存放的情况，则所形成的区域称为空闲连通区域。图 6 所示空白区域为空闲连通区域。

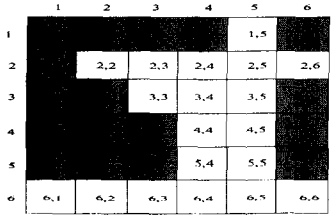


图 6 空闲连通区域

空闲连通区域中间包含已被占用的区域称为空心连通区域，否则为实心连通区域。一般来说，当有一个硬件任务需要在空闲区域布局时，除了需要考虑适合的面积之外，在布局上，位置基本都是以某一边界为准布局，这样有利于减少生成更多的碎片。因此，本文以实心连通区域为研究对象。

若已经有空闲区域，用以下方法可把空闲区域映射成一个无向图。

(1)将空闲 CLB 块映射为无向图的一个点；

(2)将两个 CLB 块的相邻关系映射为无向图的一个边；

(3)只考虑空闲区域边界的所有 CLB 块，空闲区域内部不予考虑。

可以得知空闲连通区域与无向图之间存在映射关系。得到的无向图具有形状不变的性质，即连通区域的规则形状映射到无向图上，其形状不变，以图 6 为例，其对应的无向图如图 7 所示。用 C_{ij} 来表示坐标为(i,j)的 CLB 块对应的顶点。显然，根据图的定义 $G=(V,E)$，图 7 所表示的无向图 G 中，

$V=\{C_{15},C_{22},C_{23},C_{24},C_{25},C_{26},C_{33},C_{34},C_{35},C_{44},C_{45},C_{54},C_{55},C_{61},C_{62},C_{63},C_{64},C_{65},C_{66}\}$

$E=\{\langle C_{15},C_{25}\rangle,\langle C_{22},C_{23}\rangle,\langle C_{23},C_{24}\rangle,\langle C_{24},C_{25}\rangle,\langle C_{25},C_{26}\rangle,\langle C_{23},C_{33}\rangle,\langle C_{24},C_{34}\rangle,\langle C_{25},C_{35}\rangle,\langle C_{33},C_{34}\rangle,\langle C_{34},C_{44}\rangle,\langle C_{35},C_{45}\rangle,\langle C_{44},C_{45}\rangle,\langle C_{44},C_{54}\rangle,\langle C_{45},C_{55}\rangle,\langle C_{54},C_{55}\rangle,\langle C_{54},C_{64}\rangle,\langle C_{55},C_{65}\rangle,\langle C_{61},C_{62}\rangle,\langle C_{62},C_{63}\rangle,\langle C_{63},C_{64}\rangle,\langle C_{64},C_{65}\rangle,\langle C_{65},C_{66}\rangle\}$

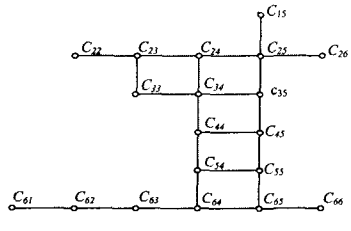


图 7 空闲区域映射的无向图

结论：空闲区域映射的无向图是一个由若干条水平线和垂直线构成的规则形状连通图。

3 空闲区间 UPFS 划分算法

将 FPGA 的空闲区域映射成对应的一个无向图时，寻找空闲矩形集的问题就转化成寻找无向图中满足条件的回路和通路问题，实现过程如下所述。

3.1 无向图简化

对于通过映射关系得到的无向图如图 7 所示，需要找到其中的最大回路，搜索步骤如下。

3.1.1 无向图的预处理

在由相关顶点和边构成的子图中，每个顶点的度数都为 2 的连通图称为回路。因此，图 7 中必须将度数为 1 的悬挂顶点以及与它关联的悬挂边裁剪掉，连续进行裁剪后，得到图 8 所示的无向图。

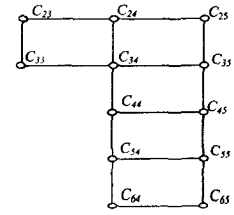


图 8 预处理后的无向图

3.1.2 建立无向图邻接矩阵

根据图 8 建立邻接矩阵 $M2$，由于邻接矩阵是对称方阵，存储时只需存储矩阵的上三角矩阵。

$$M2=\begin{array}{c}\\ C_{23}\\ C_{24}\\ C_{25}\\ C_{33}\\ C_{34}\\ C_{35}\\ C_{44}\\ C_{45}\\ C_{54}\\ C_{55}\\ C_{64}\\ C_{65}\end{array}\begin{bmatrix} C_{23} & C_{24} & C_{25} & C_{33} & C_{34} & C_{35} & C_{44} & C_{45} & C_{54} & C_{55} & C_{64} & C_{65}\\ 0&2&0&2&0&0&0&0&0&0&0&0\\ 2&0&2&0&2&0&0&0&0&0&0&0\\ 0&2&0&0&0&2&0&0&0&0&0&0\\ 2&0&0&0&2&0&0&0&0&0&0&0\\ 0&2&0&2&0&2&2&0&0&0&0&0\\ 0&0&2&0&2&0&0&2&0&0&0&0\\ 0&0&0&0&2&0&0&2&2&0&0&0\\ 0&0&0&0&0&2&2&0&0&2&0&0\\ 0&0&0&0&0&0&2&0&0&2&2&0\\ 0&0&0&0&0&0&0&2&2&0&0&2\\ 0&0&0&0&0&0&0&0&2&0&0&2\\ 0&0&0&0&0&0&0&0&0&2&2&0\end{bmatrix}$$

3.1.3 搜索最大回路

对于连通的无向图而言，最大回路只有一个。引入方向矢量交角，可以在无向图中很快找到最大回路。具体方法是按回路的逆时针方向进行搜索，那么搜索到某顶点时，方向矢量交角最小的边即为所选择的搜索边。

(1)确定搜索出发点和起始边。无向图上全部顶点均有相应坐标，可以确定 X 坐标为最小或最大以及 Y 坐标为最小或最大的顶点必然是最大回路中的顶点。搜索出 X 坐标最小的顶点即最左边的某一顶点作为搜索的出发点，在图 8 中是 C_{23} 顶点。根据邻接矩阵，求该点邻接边相对于 X 轴的方向矢量交角，找出方向矢量交角最小的边$\langle C_{23},C_{33}\rangle$作为搜索起始边，该边必然为最大回路中的一条边，将邻接矩阵 $M2$ 中的对应元素 $M2_{41}$ 减一，C_{33} 作为下一次搜索的起始顶点。

(2)搜索最大回路中与当前搜索边相邻的边。根据邻接矩阵 $M2$ 和顶点坐标，求与顶点 C_{33} 的邻接边$\langle C_{33},C_{34}\rangle$相对于当前搜索边$\langle C_{23},C_{33}\rangle$的方向矢量交角，找出方向矢量交角最小的边$\langle C_{33},C_{34}\rangle$，该边也一定是最大回路的一边，将邻接矩阵 $M2$ 中对应元素 $M2_{54}$ 减 1。将搜索出的边$\langle C_{33},C_{34}\rangle$作为下次搜索的起始边，该边的终点 C_{34} 作为下次搜索的起点。

(3)重复步骤(2)，直到搜索边的终点回到出发点为止，邻接矩阵演化为 $M3$，所形成的回路即为最大回路，如图 9 所示，即 $Lm=(C_{23},C_{24},C_{25},C_{35},C_{45},C_{55},C_{65},C_{64},C_{54},,C_{44},C_{34},$

C_{33},C_{23})。

$M3=$

	C_{23}	C_{24}	C_{25}	C_{33}	C_{34}	C_{35}	C_{44}	C_{45}	C_{54}	C_{55}	C_{64}	C_{65}
C_{23}	0	1	0	1	0	0	0	0	0	0	0	0
C_{24}	1	0	1	0	2	0	0	0	0	0	0	0
C_{25}	0	1	0	0	0	1	0	0	0	0	0	0
C_{33}	1	0	0	0	1	0	0	0	0	0	0	0
C_{34}	0	2	0	1	0	2	1	0	0	0	0	0
C_{35}	0	0	1	0	2	0	0	1	0	0	0	0
C_{44}	0	0	0	0	1	0	0	2	1	0	0	0
C_{45}	0	0	0	0	0	1	2	0	0	1	0	0
C_{54}	0	0	0	0	0	0	1	0	0	2	1	0
C_{55}	0	0	0	0	0	0	0	1	2	0	0	1
C_{64}	0	0	0	0	0	0	0	0	1	0	0	1
C_{65}	0	0	0	0	0	0	0	0	0	1	1	0

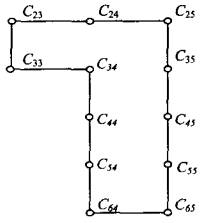


图 9　最大回路

3.2　最大空闲矩形的 UPFS 算法

定义 8（最大空闲矩形）　不能被其它任何一个空闲矩形所完全覆盖的空闲矩形为最大空闲矩形。

定义 9（水平通路）　设 $G=(V,E)$ 是一个无向图，A_1，A_2，…，A_K 是 V 中的 K 个顶点，则 G 中从 A_1 到 A_K 的一条通路是水平通路，它应满足下列条件：对于所有 $1\leqslant i\leqslant k$，A_i 行向量取值相等。在无向图（见图 7）中，下面两个通路表示的就是水平通路。$C=(C_{22},C_{23},C_{24},C_{25},C_{26})$，$C=(C_{33},C_{34},C_{35})$。

定义 10（垂直通路）　设 $G=(V,E)$ 是一个无向图，A_1，A_2，…，A_K 是 V 中的 K 个顶点，则 G 中从 A_1 到 A_K 的一条通路是垂直通路，它应满足下列条件：对于所有 $1\leqslant i\leqslant k$，A_i 列向量取值相等。在无向图（见图 7）中，下面两个通路表示的就是垂直通路。

$C=(C_{24},C_{34},C_{44},C_{54},C_{64})$

$C=(C_{15},C_{25},C_{35},C_{45},C_{55},C_{65})$

定理 1　水平通路构成一个最大空闲矩形，当且仅当：水平通路中经过的所有顶点不完全是最大回路中的顶点。

例如对图 7 来说，水平通路 $C=(C_{22},C_{23},C_{24},C_{25},C_{26})$ 能够构成一个最大空闲矩形，因为其中的顶点 C_{22} 与 $C_{26}\notin Lm$。水平通路 $C=(C_{33},C_{34},C_{35})$ 不能构成最大空闲矩形，因为其中的顶点全部是最大回路 Lm 中的顶点。

定理 2　垂直通路构成一个最大空闲矩形，当且仅当：垂直通路中经过的所有顶点都不完全是最大回路中的顶点。

例如对图 7，垂直通路 $C=(C_{15},C_{25},C_{35},C_{45},C_{55},C_{65})$ 能够构成一个最大空闲矩形，因为其中的顶点 $C_{15}\notin Lm$。垂直通路 $C=(C_{24},C_{34},C_{44},C_{54},C_{64})$ 不能构成最大空闲矩形，因为其中的顶点也全部是最大回路 Lm 中的顶点。

定理 3　回路可以构造若干个部分重叠的最大空闲矩形，当且仅当：回路为最大回路。最大回路中满足条件的部分回路对应最大空闲矩形。

例如在图 9 中就可以构造两个部分回路 $L1=(C_{23},C_{24},C_{25},C_{35},C_{34},C_{33},C_{23})$，$L2=(C_{24},C_{34},C_{44},C_{54},C_{64},C_{65},C_{55},C_{45},C_{35},C_{25},C_{24})$，对应两个最大矩形。

由上可见，对于一个基于 FPGA 的空闲资源区域，采用 UPFS 无向图方法也可以划分空闲区域，快速找到最大空闲矩形的集合，并且集合中空闲矩形具有部分重叠，以保证尽可能获得最大矩形，UPFS 方法没有冗余计算和重复计算的问题。

如果一个新的已到达的任务不止适合一个空闲矩形时，就必须用一个适应策略来选择一个矩形，当前比较成熟的算法如最佳适应算法（BF）、最差适应算法（WF）、最佳最精确的适应算法（BFEF）、最差最精确算法（WFEF）等都可以作为一种策略。

4　仿真

为了测试 UPFS 算法的性能，我们开发了一个 FPGA 仿真环境并进行了仿真实验，仿真环境最小时间单位为 10ms，模拟可重构芯片的规模相当于 Xilinx Virtex XCV1000 FPGA，大小为 96×64 个 CLB 块。

仿真期间，硬件任务根据先入先出的顺序到达可重构芯片，布局器根据 FF 适应策略为到达的硬件任务选择合适的最大空闲矩形。当一个任务被放置到 FPGA 中，或者从 FPGA 中删除时，该 FPGA 的最大空闲矩形集合更新一次。在这个仿真实验中，任务生成器随机产生 20000 个硬件任务存储在队列里，硬件任务的宽度和长度范围是 2 到 8 个 CLB。分别采用 UPFS（Undigraph partition free space）算法、L（Scan line）算法仿真，结果如表 1 所列，UPFS 的平均执行时间接近 SL 算法的 1/2。应该看到，由于扫描线算法有冗余计算的问题，随着可重构芯片规模的增长，UPFS 的优势将更明显。

表 1　UPFS 算法与 SL 算法执行时间对比

单位：10^4ms	UPFS	SL
1	2.3819	4.7178
2	2.8612	5.2962
3	2.5312	4.8215
平均	2.5914	4.9452

定义 11（任务拒绝率，Task Rejection Radio，TRR）　指芯片拒绝的任务和到达芯片的任务总数之比[9]。

定义 12（执行时间，Perform Time，PT）　指放置完所有到达芯片的任务所用的时间[9]。

定义 13（有效耗损时间，Valid Consuming Time，VCT）　是 TRR 和 PT 之积[9]。

当任务拒绝率高而且执行时间长时，有效耗损时间就长。分别触发不同数目的硬件任务，测试 UPFS 算法和 SLA 算法的 VCT 值，由图 10 可知，UPFS 算法的 VCT 是 SLA 算法的 60%左右，性能更优。

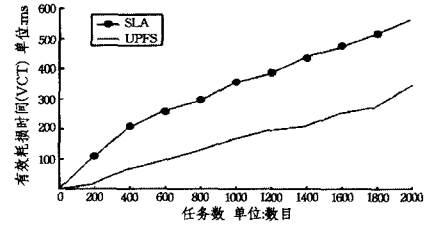


图 10　UPFS 算法和 SLA 算法的 VCT 值

结束语　本文提出的基于图论技术的最大空闲矩形寻找方法，从一个新的角度来解决可重构资源的管理问题，它把空

闲区域映射到无向图上，将寻找最大空闲矩形问题转化为求解最大回路和通路问题，使空闲区域划分过程大大简化。仿真结果表明，应用图论技术解决问题的思想和方法是可行的。将图论技术应用于可重构资源管理领域，目前在国内还未见过报道，本文在这个方面进行探索，初步的结果比较理想，在最大矩形判断方法和计算复杂度等问题上还可以继续做深入研究。

参 考 文 献

[1] Walder H, Steiger C, Platzner M. Fast Online Task Placement on FPGAs: Free Space Partitioning and 2D Hashing[J]. IEEE Trans on Computers, 2003, 53(11): 178-185

[2] Bazargan K, Kastner R, Sarrafzadeh M. Fast Template Placement for Reconfigurable Computing Systems[J]. IEEE Design &Test of Computers, 2000, 17(1): 68-83

[3] 齐骥，李曦，胡楠，等. 基于硬件任务顶点的可重构系统资源管理算法[J]. 电子学报，2006，34(11)：2094-2098

[4] 李涛，杨愚鲁. 可重构资源管理及硬件任务布局的算法研究[J]. 计算机研究与发展，2008，45(2)：375-382

[5] Handa M, Vemuri R. An efficient algorithm for finding empty space for online FPGA placement[C]// The 15th Int'l Conf on Field Programmable Logic and Applications. Tampere, Finland, 2005: 960-965

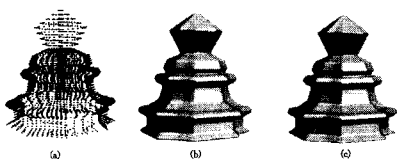
[6] Ahmadinia A, Bobda C, Bednara M, et al. A New Approach for On-line Placement on Reconfigurable Devices[C]// Int'l Conf on Architecture of Computing System. Augsburg, Germany, 2004: 134-140

[7] Cui Jin, Deng Qing-xu, He Xiu-qiang, et al. An efficient algorithm for online management of 2D area of partially econfigurable FPGAs[C]// Proceeding of the conference on Design, automation and test. California, 2007: 129-134

[8] 张先迪，李正良. 图论及其应用[M]. 北京：高等教育出版社，2005：46-52

[9] 龚育昌，齐骥. 一种支持动态可重构系统的布局碎片量化方法[J]. 小型微型计算机系统，2007，28(5)：944-947

(上接第 254 页)



(a) 输入的散乱数据；(b) 用传统水平集方法重构的曲面；(c) 使用我们的方法的重构结果

图 4

结束语 本文对给定的散乱数据进行了曲面重构。为了使重构曲面能够保持棱角特征，我们在极小化能量模型(1)中增加了一个各向异性扩散因子。通过变分方法将极小化问题转化为一个弱形式的 L^2-梯度流来求解。利用三次 B-样条基函数将梯度流转化为矩阵形式的方程组。为了解决矩阵 A 存储的问题，我们又设计利用 Schimidt 正交化过程，将矩阵 A 转化为一个单位矩阵，从而大大节省了存储空间，提高了算法效率。试验结果表明我们的模型能很好地保持重构物体的棱角特征。

参 考 文 献

[1] Xu G, Zhang Q, Liu D. Smooth surface reconstruction from noisy scattered data-variational level set methods [J]. Journal of Computer-Aided Design and Computer Graphics, 2007, 19(7): 840-848

[2] Peng D, Merriman B, Osher S, et al. A pde-based fast local level set method [J]. Journal of Computational Physics, 1999, 155: 410-438

[3] Zhao H K, Osher S, Merriman B, et al. Implicit nonparametric shape reconstruction from unorganized points using a variational level set method [J]. Computer Vision and Image Unders-tanding, 2000, 80(3): 295-319

[4] Zhao H K, Osher S, Fedkiw R. Fast surface reconstruction using the level set method [C]// Proceedings of the IEEE Workshop on Variational and Level Set Methods in Computer Vision. 2001: 194-201

[5] Yang Z, Deng J, Chen F. Fitting unorganized point clouds with active implicit B-spline curves [J]. Visual Computer, 2005, 21(8-10): 831-839

[6] 徐国良. 计算几何中的几何偏微分方程方法[M]. 北京：科学出版社，2008，10：284-285

[7] Osher S, Sethian J. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations [J]. Journal of Computational Physics, 1988, 79: 12-49

[8] Osher S, Fedkiw R. Level Set Methods and Dynamic Implicit Surfaces. Vol. 153 [M]. Applied Mathe-matical Science. New York: Springer-Verlag, 2003

[9] Sethian J A. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science (second ed) [M]. Cambridge Monographs on Applied and Computational Mathematical. Cambridge: Cambridge University Press, 1999

[10] Xu G. Adaptive and smooth surface construction by triangular A-patches [M]. AMS/IP Studies in Advanced Mathematics, 2003, 34: 213-225

[11] Bajaj C L, Chen J, Xu G. Modeling with cubic A-patches [J]. ACM Transactions on Graphics, 1995, 14(2): 103-133

[12] Zhao H K, Chan T, Merriman B, et al. A variational level set approach to multiphase motion [J]. Journal of Computational Physics, 1996, 127: 179-195

[13] Preußer T, Rumpf M. An adaptive finite element method for large scale image processing[J]. Journal of Visual Communication and Image Representation, 2000: 183-195

[14] Kobbelt L, Botsch M, Schwanecke U, et al. Feature sensitive surface extraction from volume data[C]// Proceedings of the 28th annual conference on Computer graphics and interactive techniques. New York, USA, ACM, 2001: 57-66

[15] Bajaj C L, Xu G. Anisotropic diffusion of surface and functions on surfaces[J]. ACM Transactions on Graphics, 2003, 22(1): 4-32

[16] Abramowitz M, Stegun I A. Handbook of Mathematical Functions with Formulas, Graphics, and Mathematical Tables[M]. America Dover Publications, Inc., 1972

[17] Xu G, Shi Y. Progressive computation and numerical tables of generalized Gaussian quadrature formulas [J]. Journal on Numerical Methods and the Computer Application, 2006, 27(1): 9-23