

可重构系统中硬件任务布局布线算法研究

韩国栋 肖庆辉 张 帆

(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要 可重构计算系统中,二维可重构硬件任务的布局布线问题是影响系统资源利用率的重要因素。在异质化的可重构器件和任务模型基础上,对可重构硬件任务进行了适当分类,并提出一种能够对多类型可重构硬件任务同时布局布线的算法 DRS-TCW。实验表明,该算法能够有效提高可重构器件的资源利用率和任务布线连通率。

关键词 可重构计算系统,硬件任务,关联任务,布局布线

中图法分类号 TP302,TP311 **文献标识码** A

Algorithms of Placing and Routing Hardware Task in Reconfigurable System

HAN Guo-dong XIAO Qing-Hui ZHANG Fan

(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract The problems of placing and routing of 2D reconfigurable hardware tasks are important factors of system resource utilization in reconfigurable computing system. Reconfigurable hardware tasks, which are based on heterogeneous reconfigurable devices and task models, were properly classified, and a new algorithm named of DRS-TCW was proposed, which can place and route multiple reconfigurable hardware tasks simultaneously. Experiment results indicate that the algorithm can improve the resource utilization of reconfigurable devices and tasks' routing ratio.

Keywords Reconfigurable computing system, Hardware task, Correlative task, Placement and route

基于可重构器件,尤其是部分动态可重构 FPGA 的可重构计算,正日益成为新的研究热点^[1]。其中将可重构硬件任务布局到可重构资源上,如何实现硬件任务的高效布局,是影响系统性能关键因素之一。国外已研究的布局算法 LAH^[2]、ALAP^[3]等,仅依据任务大小布局到器件的资源上。国内对此研究较少,文献[4]提出一种保存任务顶点信息的布局算法 Kvit,文献[5]提出一种管理可重构系统空闲资源的算法 TT-KAMER。但这些算法中任务模型为独立同质的,与愈来愈异质化^[6]可重构器件、任务模型以及任务间存在密切通信联系是不相符的。对此本文建立了异质化器件和关联任务模型,并提出能同时对多样性硬件任务进行布局布线的算法 DRS-TCW。

1 可重构器件模型及描述

1.1 可重构器件模型

本文以支持部分动态可重构的 Xilinx Virtex-5 FPGA 为模型,该系列 FPGA 提供了丰富的 BRAM 和 DSP 乘法单元资源。BRAM 一般单列分布,如图 1(a)所示。DSP slice 常紧挨 BRAM 右侧分布,且与 BRAM 共用开关矩阵。可将 BRAM 和 DSP slice 视为一体,以 BRAM 为代表,器件在中间位置提供了一系列的 IO Logic 资源。如图 1(b)所示,依据异质化可重构资源建立了任务模型,用多元素组 $T(w, h, e)$ 来表示, w, h, e 分别表示任务的宽度、高度和执行时间长度,并

根据任务内部使用资源的不同分为以下两种任务:普通硬件任务 GRHT (General Reconfigurable Hardware Task, GRHT),任务内不含有静态资源,仅含有 CLB 和布线资源;含静态资源的硬件任务 SRHT (Special Reconfigurable Hardware Task, SRHT),如含 BRAM、DSP Slice 或 IO 放置位置要求等特殊资源的任务。FPGA 内的布线资源主要由连接盒 CB (Connect Block, CB) 和开关盒 SB (Switch Box, SB) 组成。图 1(c)所示为 FPGA 内 SB 的拓扑结构示意图。

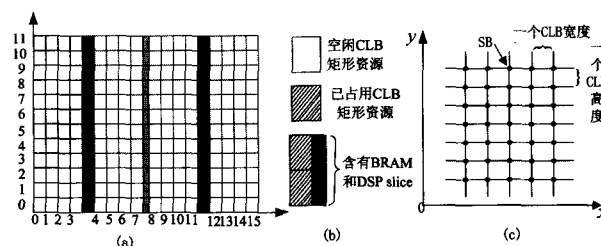


图1 可重构器件和任务模型图以及 SB 拓扑结构图

1.2 可重构器件模型矩阵描述

可重构器件模型由 W 列 H 行逻辑单元 CLB 组成。每一逻辑单元可用坐标 (x, y) 表示,其中 $0 \leq x \leq W-1, 0 \leq y \leq H-1$ 。器件用异质化数据结构描述,如图 2 所示,矩阵大小与器件模型相同,0 表示 CLB 空闲状态,-1 表示 BRAM 未使用状态,-2 表示器件中间的 IO Logic 未使用状态。当器件

到稿日期:2010-12-15 返修日期:2011-03-22 本文受河南省自然科学基金资助项目(092300410035)资助。

韩国栋(1964—),男,博士,副教授,硕士生导师,主要研究方向为计算机网络体系结构、可重构计算;肖庆辉(1984—),男,硕士生,主要研究方向为可重构计算;张帆(1981—),博士生,主要研究方向为高性能计算与 SoC 设计。

放置任务后,占用 CLB 加 1,占用 BRAM 加 2,且因 BRAM 分布特殊性,要同时对同列 4 个 BRAM 加 2,占用 IO Logic 加 3。任务运行完毕,对所释放资源的数据状态进行相反操作即可。

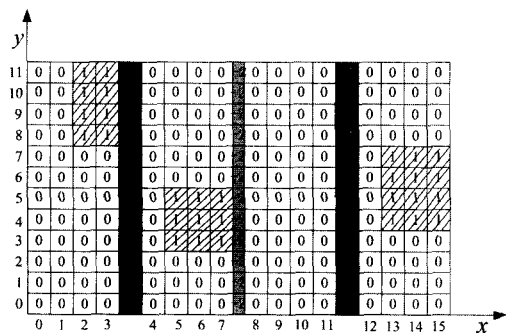


图2 可重构资源模型的矩阵描述

2 任务布局布线分析

2.1 任务间关系分析

依据任务间是否存在通信联系,将任务进行两种分类。

独立任务:任务为一个计算整体,任务间不存在任何通信联系。如图 3(a)所示,每个任务均为独立的计算整体,节点 V2 表示该任务为 SRHT 型任务。

关联任务(correlative task):任务间存在通信联系,需要各个任务相互配合执行才能完成一个计算整体。称关联任务整体为父亲任务,其所包含的任务称为子任务。

关联任务可用有向无环图 DAG (Directed Acyeling Graph,DAG)描述,如图 3(b)所示,DAG 可表示为 $G=(V, E)$,其中 V 为 DAG 中的节点集,每个节点对应一个任务 T_i ; E 表示 DAG 中任务间有通信关系的边的集合, e_{ij} 表示 T_i 的输出为 T_j 的输入。子任务布局成功与否对于父亲任务整体是同等重要的。

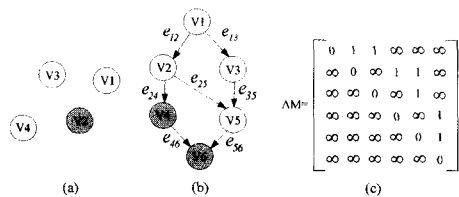


图3 任务关系图和关联任务的矩阵描述

DAG 用邻接矩阵 (Adjacency Matrix, AM) 描述,如图 3 (b)可用图 3(c)所示的 AM 表示。父亲任务截止时间不受限时,其子任务没必要同时布局到器件上,将这种关联任务归为 DAG1 任务;若父亲任务需将该父亲任务所有子任务同时布局到器件上,即子任务间须共同存在于可重构器件内部,将这种关联任务归为 DAG2 任务。

2.2 布局分析

硬件任务布局至关重要,一个很差的布局结果不可能通过良好的布线算法达到较好的系统性能,故布局的重要目标就是为了满足后续布线需要^[7]。布局 GRHT 独立任务时,应预留有限的静态资源位置,增加后续 SRHT 任务布局成功的概率。关联任务中不含 SRHT 任务时,可先放置难布局的任务,如面积较大者等。对于 DAG1 任务,需先依据子任务间

关系对任务进行分组,再布局。如图 4(a)所示,第一步可将子任务 T1、T2 和 T3 划为 C1 组,对 C1 中 3 个子任务进行连续布局;第二步布局 C2 中子任务 T5;第三步布局 C3 中子任务 T4,最后布局 C4 中的子任务 T6。

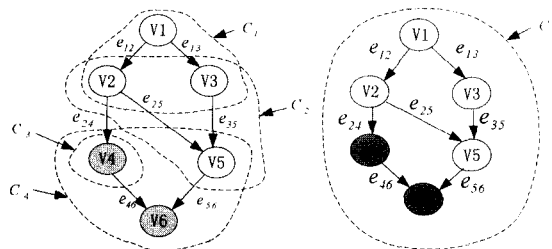


图4 关联任务 DAG1 型和 DAG2 型

布局 DAG2 任务时,如图 4(b)所示,先放置父亲任务中的 SRHT 型子任务;其次,依据 DAG2 中任务间关系再逐个布局,第一个子任务 V1 选择位置时,在相对方向上与任务 V4 和 V6 成垂直或水平趋势,以便于 V2 和 V3 位置选择;第三,依据 V1 和 V4 布局位置确定 V2 的位置,使 V2 与 V1、V4 成垂直或水平趋势,再确定 V3 和 V5 的位置。

关联 SRHT 型任务放置原则:在多个可选放置位置中,尽可能使得任务间呈垂直或水平趋势放置。

关联 GRHT 型任务放置原则:任务位于 DAG1 时,优先放置占用面积较大的任务,依据通信关系按下述一般放置方法再对其他任务逐个进行布局;任务位于 DAG2 时,依据已布局好的 SRHT 型任务位置,优先放置占用面积较大者,再按下述一般放置方法进行其他任务布局。在布局一般 GRHT 型任务时有:

(1)放置单指向任务时,在距离优先条件下,选使任务间呈垂直或水平趋势的位置放置。如图 5(a)所示,对 T2 单指向 T4 进行放置,T2 可放置在位置 s3,使 T2 和 T4 呈垂直趋势放置。但 s1 较 s3 距离 T4 小,即 $l_1 > d > l_2$,其中 l_1 、 d 、 l_2 分别为 T4 与 s2、T4 与 s3 和 T4 与 s1 的中心距离,故选 s1 放置。

(2)放置多指向任务时,如图 4(b)中 T6,在多个可放置位置中需将 T6 放置靠近与 T6 前驱任务已确定的方向线上,如图 5(b)所示,T6 放置 s2 较放置 s1 时有 $l_1 + l_2 > l_1' + l_2' > d$,即可放位置 s2 更靠近 T4 和 T5 确定的方向线,故选 s2 放置。

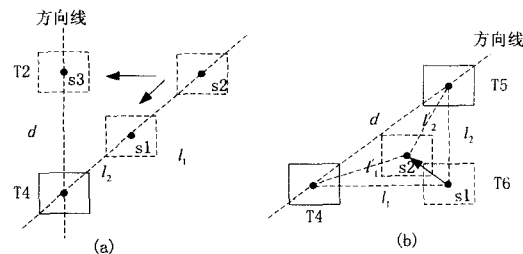


图5 GRHT 型任务布局位置选择示意图

2.3 布线分析

任务放置会与其周围 SB 相邻,任务通信时需利用其相邻的 SB 进行布线。又 CLB 为左输入右输出,便于问题分析,描述了任务的输入区域和输出区域:输入区域 (Input Region, IR),即任务放置后其左侧相邻两端 SB 之间的布线区域 (见

图 6);输出区域(Output Region,OR)即任务放置后其右侧相邻两端 SB 间的布线区域(见图 6)。

布线连通时,布线进入 IR 和 OR 的范围即可。每条布线起点到终点进行具有方向性连通的搜索,即连通方向搜索(Direction of Routability Search,DRS),在 SRHT 型任务布局后再布局 GRHT 型任务,并由 GRHT 任务确定的指引连通 SRHT 和 GRHT 任务间通信连接的搜索方向,表现为该方向上 SB 跳转概率值。

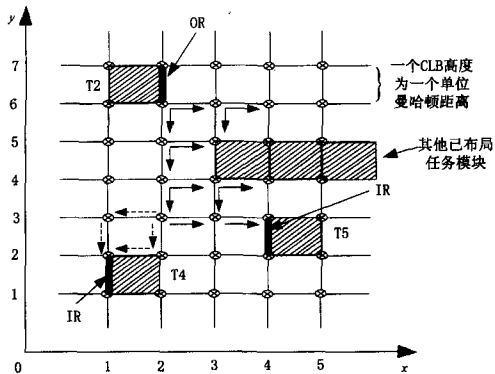


图 6 布线路径搜索示意图

因 SB 可在 4 个方向上进行跳转,本文构建了 SB 专用的数据结构,表示如下:

$$SB(x, y) : [DRS-up \quad DRS-right \quad DRS-down \quad DRS-left]$$

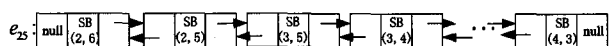
SB 数据结构中从左至右分别表示 SB 可跳转的 4 个方向,0 表示该方向可用,小数或 1 表示该方向的跳转概率值,Z 表示该方向不可用。

DRS 方法:设 T2、T4 和 T5 布局位置如图 6 所示,可先进行 T2 至 T5 的连通。对于 T2,其 OR 可由 SB(2,6) 开始进行连通方向搜索。对于 T5,可由其 IR 中的 SB(4,3) 接收来自 SB(2,6) 的连通连接。因 SB(4,3) 在 SB(2,6) 右下方,两者间用曼哈顿距离(Manhattan Distance, MD)表示,即 x 方向 MD 为 $MDX=2$, y 方向 MD 为 $MDY=3$ 。对于 SB(2,6),其到 SB(4,3) 进行下一跳转的概率值可按如下计算:

$$DRS-down = \frac{\beta \times MDY}{\alpha \times MDX + \beta \times MDY} \times 100\% \quad (1)$$

$$DRS-right = \frac{\alpha \times MDX}{\alpha \times MDX + \beta \times MDY} \times 100\% \quad (2)$$

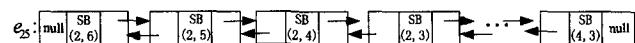
式(1)和式(2)在布线初始计算时 $\alpha=\beta=1$,计算可得 SB(2,6) 到 SB(4,3) 在 y 方向跳至 SB(2,5) 的概率 $DRS-down=0.6$ 。同理,从 x 方向跳至 SB(3,6) 的概率 $DRS-down=0.4$ 。SB 每次进行下一跳时,从跳转概率中选取概率值最大的方向进行跳转,即 SB(2,6) 跳至 SB(2,5),之后重新计算由 SB(2,5) 至 SB(4,3) 的 MDX 和 MDY ,可得 $MDX=MDY=2$,即 $DRS-down=DRS-right=0.5$ 。任选其中一个进行跳转,假设下一个跳为 SB(3,5),到达 SB(3,5) 后重新计算由 SB(3,5) 至 SB(4,3) 的 MDX 和 MDY 以及对应的 $DRS-down$ 和 $DRS-right$ 。如此循环下去,直到达至 SB(4,3),用双向链表(Double Linked List)描述上述搜索过程可表示如下:



布线搜索过程中 4 种特殊情形及解决方法描述如下。

情形 1:上述布线搜索至 SB(3,4) 时,若该 SB 已占用,则

布线搜索返回至该 SB 上一跳 SB(3,5),并对 SB(3,4) 数据状态进行全 Z 标记,对 SB(3,5) 的 y 方向进行不可用 Z 状态标记,然后由 SB(3,5) 在 x 方向搜索;跳至 SB(4,5) 时,计算 MD 得出该 SB 到达目的 SB 应在负 y 方向跳转。若该 SB 的布线区域已用,则由 SB(4,5) 继续向右跳至 SB(5,5);到达 SB(5,5) 后,若该 SB 所对应向下布线区域已占用,而且布线搜索至该 SB 时经过的 MD 等同于 SB(2,5) 至 SB(4,3) 的 MDX 和 MDY 之和,继续右跳, x 方向可能连通,但后续负 x 方向的左回跳会导致路径过长。任务运行完,其周围有正使用的布线通道,空闲资源不能与其他空闲资源合并,碎片化了空闲资源。对此,布线搜索至 SB(5,5) 时就沿原布线路径回跳 3 次至 SB(2,5),并对该路径上的 SB 进行 Z 标记。为了避免上述 x 方向搜索失败的重复出现,在重新计算 SB(2,5) 跳转概率时,对 MDX 和 MDY 进行加权计算,即 $\alpha=1, \beta=3$ 。其中 $\beta=3$ 是上述 x 方向由 SB(5,5) 回跳至 SB(2,5) 的次数,因而在 SB(2,5) 有 $DRS-down=0.75, DRS-right=0.25$,即下一跳为 SB(2,4),且有 $DRS-down=0.6, DRS-right=0.4$,故继续向下跳至 SB(2,3)。上述布线搜索过程用双向链表表示如下:



情形 2:式(1)和式(2)在计算 SB 跳转概率时,会出现概率为 1 且该方向不可连通的情形。此时可将出现概率 1 设为 0.9,0.1 为前后两个 SB 间顺延方向的跳转概率。如图 6 中,可设 SB(4,5) 至 SB(5,5) 的方向概率为 0.1。

情形 3:当需将相同信号连通至另一模块 IR,且该信号已成功连通至其他任务时,可计算成功连通路经上 SB 至目的 SB 的 MDX 和 MDY 之和,从中选取 MD 之和最小的 SB 作为源 SB,再按上文所述 DRS 布线方法进行布线。连通则已,反之按照情形 4 的解决方法查看搜索路径 CLB 使用状态,使用 CLB 内 Slice 资源进行布线,否则布线失败。

情形 4:若布线至目的 SB 不可连通,且距离目的 SB 不大于两个 MD 时,查看该路径周围 CLB 的使用状态,利用 CLB 内 Slice 资源进行布线。连通则已,否则布线失败。

3 DRS-TCW 布局布线算法

为满足任务的布局布线需求,本文吸取文献[8]中任务队列设置布局窗口的思想,在任务等待队列上设置任务分类窗口(Task Classify Window, TCW),主要对关联任务进行布线可连通性的布局,在任务先到先服务(First Come First Service)的简单调度方法下进行任务布局布线仿真,故将本文算法命名为 DRS-TCW 算法,步骤如下:

Step1 按照 FCFS,任务排队,TCW 检查等待队列并进行分类,放入等待子队列。

Step2 将独立任务转到 Step3,若任务为 DAG1 转到 Step4,为 DAG2 转到 Step5。

Step3 若任务为 GRHT 型转到 Step6,若任务为 SRHT 型转到 Step7。

Step4 DAG1 任务,按上述 2.2 节中划分方法进行组划分,若待布局任务组中含有 SRHT 型任务转到 Step7,将其余 GRHT 型任务转到 Step6,然后进行布线转到 Step9。若前驱任务运行完,则检查分组是否布局完,没有则转到 Step8。

Step5 DAG2 任务,若任务为 SRHT 型转到 Step7,若任务为 GRHT 型转到 Step6,然后进行布线转到 Step9,最后转到 Step10。

Step6 对于独立 GRHT 任务,在避免使用静态资源前提下寻找合适位置放置,然后转到 Step10;关联 GRHT 任务按上述 2.3 节中的该类型任务放置原则进行放置,并返回。

Step7 对于独立 SRHT 型任务,寻找合适且靠近角落的位置放置,然后转到 Step10;对于关联 SRHT 型任务,按上述 2.3 节中的该类型任务放置原则进行放置,并返回。

Step8 对于分组中的后继任务,在不影响 DAG2 任务连续布局前提下,将后继任务发给图 8 中任务加载选择部件进行布局,并返回到 Step4。

Step9 按照 2.4 节中布线方法进行布线搜索,并返回。

Step10 布局布线成功,输出其位置、布线双向链表,反之输出任务序号,并释放布线搜索中所标记的双向链表没使用到的 SB 状态。

DRS-TCW 算法如图 7 所示。TCW 对任务分类并放入等待子队列,在 FCFS 下选择部件选择 T1 进行加载和配置,之后进行下一任务布局。对于 DAG1 任务对一组任务进行连续布局布线,此时加载选择部件不再接收独立或 DAG2 任务。布局一组后,再进行独立和 DAG2 任务的布局,对 DAG2 须保持布局连续性,即连续进行加载选择和配置。

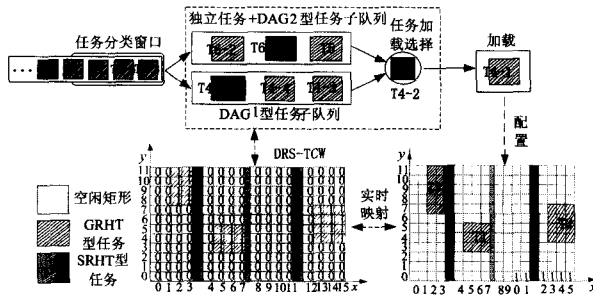


图 7 DRS-TCW 布局布线算法

以图 7 为例,对于 DAG1 任务,已布局 T4-1、T4-2 和 T4-3 这一组 C1 之后,可对 T5 和 T6-1 布局。若资源状态实时映射发现 DAG1 分组 C1 中 T4-1 运行完毕,则可将后继任务 T4-4 在不影响 DAG2 连续布局前提下,发给任务选择部件进行加载。

4 算法仿真

4.1 仿真环境及实验模型

为验证上述布局布线算法的性能,采用 Matlab 仿真工具构建了混合可重构硬件任务布局布线仿真环境 MRHTPRS (Mixed Reconfigurable Hardware Task Placement and Routing Simulator)。计算机硬件配置为:CPU 为 Intel Pentium 2.60GHz,内存为 1.0G,系统 Windows XP。器件按 xc5v1x30 规模定义。仿真环境呈 3 种混合的硬件任务,初步设定 3 种任务比例分别为 75%、15%和 10%,GRHT 和 SRHT 型任务比重为 60%和 40%,并随机分配到 3 种任务中。独立任务宽度和高度在[5,20]内生成,DAG1 和 DAG2 任务宽度和高度在[5,15]生成,其中子任务数在[3,6]之间,SRHT 任务所含 BRAM 数在[1,3]之间, e 在[20,100]生成,单位为 10ms,且

DAG2 任务 e 在选择后保持一致。设定实验条件下生成 5 组任务,每组 1000 个,实验结果的平均值作为该条件下的考查对象。

4.2 评价准则

本实验对 DRS-TCW 算法和 FF(First Fit)算法在资源利用率和算法开销上进行了性能比较,并分析了 DAG 任务中 SRHT 和 GRHT 型任务比重不同时的布线连通率。

因任务具有多样性,本文从两个角度分析了资源利用率,可用如下两个公式表示:

$$\eta = \frac{1}{W \times H \times E_{T(i) \in S_P}} \sum x(T(i)) \times y(T(i)) \times e(T(i)) \quad (3)$$

$$\eta_e = \frac{1}{W \times H \times E_{T(i) \in S_{PR}}} \sum x(T(i)) \times y(T(i)) \times e(T(i)) \quad (4)$$

式中, W 为器件宽度, H 为器件高度, E 为首个任务的起始加载时刻到最后一个任务运行完的时间长度, S_P 表示成功布局到器件上的任务集, S_{PR} 表示 S_P 中运行完并完成整个应用功能的任务。 η 越大,表示器件资源利用率越高; η_e 越大且与 η 差值越小,表示资源有效利用率越大。DRS-TCW 算法含有任务的布线部分,连通率可用下列公式表示:

$$\mu_1 = \frac{\sum_{i \in DAG1} e_i' + \sum_{j \in DAG2} e_j'}{\sum_{i \in DAG1} e_i + \sum_{j \in DAG2} e_j} \times 1 \quad (5)$$

$$\mu_2 = \frac{\sum_{i \in DAG1} e_i'' + \sum_{j \in DAG2} e_j'''}{\sum_{i \in DAG1} e_i' + \sum_{j \in DAG2} e_j} \times 1 \quad (6)$$

式中, μ_1 为 DAG 任务中成功布线的边数与该组中 DAG 任务总边数之比, e_i 表示 DAG 任务所有的边, e_i' 表示 e_i 中成功布线的边, μ_1 越大表示连通率越高; e_i'' 表示 e_i' 中实际被用于任务间通信的边,即 μ_2 表示 e_i' 中有效应用于任务通信的布线利用率, μ_2 越大且与 μ_1 差值越小表示连通的布线的利用率越高,反之布线利用率低。

4.3 仿真结果及分析

图 8 所示为 GRHT 和 SRHT 型任务比重不同时的器件利用率。DRS-TCW 的器件利用率略高于 FF,GRHT 型任务比重加大时,DRS-TCW 算法优势增加, η 高于 FF,最大时有 7.2%,最小时有 1.3%,主要是 DRS-TCW 将独立任务和 DAG 任务分别进行布局,放置独立任务时避免占用静态资源,对 DAG 任务进行了方向性位置布局。

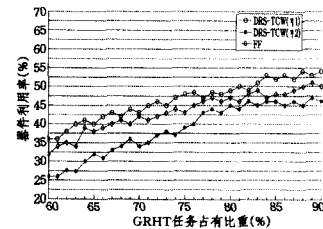


图 8 任务类型不同时的器件利用率

随着 GRHT 任务的比重加大, η 略低于 FF,最大时差 9%,最小时差 1.25%,即器件有效利用率受 GRHT 和 SRHT 任务比重、DAG 任务布线影响。当 GRHT 任务比重加大时, η 趋向于 FF 算法,受静态资源约束变小。因实用中任务异质化,即 SRHT 任务占有比重越来越大。为能够很好地布局以上两种类型任务和 DAG 任务,当 GRHT 任务比重约为 76%、器件利用率为 47%时进一步分析独立任务、DAG1 和

DAG2 任务比重不同时的布线连通率和算法开销时间,将这 3 种任务按表 1 所列比重分为 10 组,分别进行布线连通率和算法开销的对比仿真。

表 1 独立任务、DAG1 和 DAG2 任务比重不同时的分组

任务类型	分组	1	2	3	4	5	6	7	8	9	10
独立任务		75%	72%	69%	66%	63%	60%	57%	54%	51%	49%
DAG1		15%	17%	19%	21%	23%	25%	27%	29%	31%	33%
DAG2		10%	11%	12%	13%	14%	15%	16%	17%	18%	19%

从图 9 中可得,随着 DAG1 和 DAG2 任务比重的加大,DAG 任务的布线连通率逐渐减小,算法的开销时间不断增加,最大时由 96% 减到 74%,即 DAG 任务比重对 μ_1 的影响很大。布线数目增多,导致 μ_1 减小; μ_2 由最大时 91% 减到 58%,随 DAG 任务比重的加大减小,可得出 μ_2 更易受 DAG 任务比重的影响。主要是由于布线过程中,尤其是 DAG2 任务,布线必须全部一次性连通,否则视为无效布线。DRS-TCW 开销时间 t_1 远大于 FF 开销时间 t_2 ,并随分组序号呈增加的态势。这主要是由于 DAG 任务增加,对任务进行相对位置方向的搜索耗时加大。对于在线布局布线算法,需在任务分组、连通率和开销时间之间取一平衡点。第 4 组即 3 种任务分别占 66%、21% 和 13% 时,连通率 μ_1 和 μ_2 较高,算法开销时间较少,且 DAG 任务比重较大,为 34%。

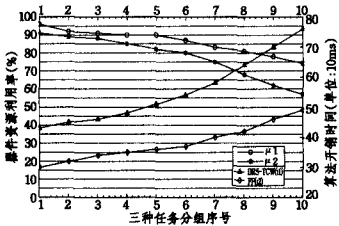


图 9 任务比重不同时的布线连通率和算法开销时间对比

(上接第 277 页)

动估计之前实时地对每一帧进行基于 IAM 的灵活宏块尺寸判别,根据不同的码率和重建质量的要求,可以灵活地调整最大和最小宏块尺寸及 IAM 阈值 T ,以保证解码端能够得到较好的边信息,适用于实际视频编码系统。实验结果表明,针对具有不同运动特性的视频序列,本文提出的两种算法使得 DVC 系统的率失真性能提升了 0.3~1.3dB。进一步的工作将考虑如何使帧内插技术更好地适应大 GOP 序列组的处理,提高 DVC 系统的压缩效率。

参考文献

[1] 向友君,雷娜,余卫宇,等.运动估计算法匹配准则研究[J].计算机科学,2009,36(9):278-280
[2] 陆寄远,朝红阳.一种快捷的高精度运动估计方法[J].计算机科学,2010,37(2):282-285
[3] Slepian S, Wolf J K. Noiseless Coding of Correlated Information Sources[J]. IEEE Transactions on Information Theory, 1973, IT-19:471-480
[4] Wyner A, Ziv J. The Rate-Distortion Function for Source Coding

结束语 本文提出的 DRS-TCW 算法有效解决了多种任务的布局布线问题,取得了较高的器件利用率和连通率。但任务布线难免会碎片化资源,而且任务布局问题是与其调度策略紧密结合的,好的调度算法结合好的布局布线算法,能进一步提高器件的资源利用率。所以未来工作将重点研究任务的调度算法和支持任务布局的碎片量化问题,以进一步提高可重构系统的整体性能。

参考文献

[1] Steiger C, Walder H, Platzer M. Operating systems for reconfigurable embedded platforms: Online scheduling of real-time tasks [J]. IEEE Trans Computer, 2004, 53(11):1393-1407
[2] Tsbero J, Septien J, Mecha H, et al. Task Placement Heuristic Based on 3D-Adjacency and Look-ahead in Reconfigurable Systems[C]//IEEE. 2006:396-401
[3] Redaelli F, Santambrogio M D, Rana V, et al. Scheduling and 2D Placement Heuristics for Partially Reconfigurable Systems[C]//IEEE. 2009:223-230
[4] 龚育昌.部分可重构系统布局的一种新算法[J].中国科技大学学报,2007,37(9)
[5] 李涛.可重构资源及硬件任务布局算法研究[J].计算机研究与发展,2008,45(2):375-382
[6] Hagemeyer J, Kettelhoit B, Koester M, et al. Design Homogeneous Communication Infrastructures for Partially Reconfigurable FPGAs[A]//Proceedings of the International Conference on Reconfigurable Systems and Algorithms[C]. Las Vegas, 2007:238-247
[7] Sharma A, Hauck S, Ebeling C. Architecture-adaptive Routability-driven Placement for Fpgas[C]//IEEE. 2005:0-7803-9362-7
[8] 李德华.异质可重构硬件任务在线布局算法研究[D].2010

with Side Information at the Decoder[J]. IEEE Transactions on Information Theory, 1976, IT-22:1-10
[5] Aaron A, Zhang R, Girod B. Wyner-Ziv coding of motion video [C]//Proceedings of the Asilomar Conference on Signals and Systems. 2002:240-244
[6] Girod B, Aaron A, Rane S, et al. Distributed video coding[J]. IEEE Special Issue on Advances in Video Coding and Delivery, 2005, 93(1):71-83
[7] Ascenso J, Prerira F. Advanced side information creation techniques and framework for Wyner-Ziv video coding[J]. Journal of Visual Communication and Image Representation, 2008, 19(8):600-613
[8] Ascenso J, Brites C, Prerira F. Improving Frame Interpolation with Spatial Motion Smoothing for Pixel Domain Distributed Video Coding[C]//5th EURASIP Conference on Speech Image Processing, Multimedia Communication and Service, 2005
[9] Saha S, Vemuri R. How Do Image Statistics Impact Lossy Coding Performance[C]//Proceedings of the International Conference on Information Technology, Coding and Computing, Las Vegas, NV, USA: IEEE Computer Society, 2000:42-47