

# 基于种类-位置模型的可重构资源指派方法

张惠臻, 谢维波, 李 蹊, 洪 欣

(华侨大学计算机科学与技术学院, 福建厦门 361021)

**摘 要:** 在基于指令集动态可扩展技术的可重构指令集处理器研究中, 如何有效使用系统的可重构资源, 将很大程度上影响扩展得到的定制指令的功能实现, 进而影响系统性能的优化效果. 本文针对可重构资源的利用问题, 首先设计了一种可重构资源模型, 该模型弱化了可重构资源的功能和数量属性, 主要提供其种类和位置属性, 并能够以此计算资源使用的时间属性. 基于此模型, 本文将图论中的图着色问题进行扩展, 引入多遍着色的思想, 提出了一种针对粗粒度可重构资源的资源指派算法, 该算法将可重构资源的指派等价为一个图多遍着色问题, 通过模型提供的属性参数和限制条件完成指派过程. 实验结果验证了算法的有效性, 并揭示了资源使用中的规律性, 对提高资源利用率和系统性能具有一定的指导意义.

**关键词:** 可重构计算; 资源建模; 资源指派; 定制指令

**中图分类号:** TP302.7      **文献标识码:** A      **文章编号:** 0372-2112 (2015)02-0299-06

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2015.02.015

## Reconfigurable Resource Assignment Based on a Type-Location Model

ZHANG Hui-zhen, XIE Wei-bo, LI Xi, HONG Xin

(School of Computer Science and Technology, Huaqiao University, Xiamen, Fujian 361021, China)

**Abstract:** In the research of reconfigurable instruction set processors based on instruction-set extension, the effectiveness of reconfigurable resources utilization will greatly impact the implementation of function units for custom instructions and furthermore performance improvement of the whole system. For the problem, this paper first designs a resource model, which weakens the functions and amounts of reconfigurable resources and mainly provides their types and locations that can calculate utility time. Based on the model, an assignment algorithm for coarse-grained reconfigurable resources is proposed. The algorithm deals with the problem as a multi-coloring graph, and assigns resources for custom instructions through extending graph coloring algorithm in graph theory. Experimental results prove the correctness and effectiveness of the algorithm, and reveal some interesting rules which have guiding significance to improve resource utilization and system performance.

**Key words:** reconfigurable computing; resource modeling; resource assignment; custom instructions

### 1 引言

可重构计算作为一种新型的时空域计算模式, 利用诸如 FPGA(Field Programmable Gate Arrays)等可编程逻辑器件将通用处理器的软件可编程性和硬件电路的空间并行性结合, 兼具高灵活性和高性能的特点, 是嵌入式和高性能计算领域近年来的研究热点. 随着可重构计算技术的发展, 结合传统专用指令集处理器<sup>[1]</sup>设计方法, 出现了基于指令集动态扩展技术<sup>[2]</sup>的可重构指令集处理器<sup>[3]</sup>. 可重构指令集处理器一般在诸如 FPGA 的单片可重构芯片上嵌入通用 CPU 核, 根据应用特征动态修改其指令集, 扩展实现能够提高处理性能的定制指令,

并利用可重构技术实现相应的功能部件, 更新系统硬件架构, 为应用提供最优执行环境.

指令集的动态扩展, 一般是针对由基本的通用 CPU 指令构成的应用程序, 分析抽取其计算特征生成候选指令, 然后根据系统状态对候选指令进行评估并选择得到扩展的定制指令集. 为使这些定制指令能够在应用处理中有效使用, 将利用系统提供的可重构资源, 动态构建对应的功能部件或协处理核, 并根据重构后的系统状态修改编译工具链. 在这一过程中, 需要对可重构资源抽象建模, 将其以参数形式作为评估候选指令、重构系统硬件和修改编译工具等步骤的算法处理依据. 因此建立一个良好的可重构硬件资源模型, 对可重构指令集处理

器的指令集动态扩展实现具有重要意义.同时,由于系统可重构资源的有限性,如何在资源模型的基础上充分利用可重构资源以实现定制指令的功能,也成为指令集动态扩展能否有效提升系统处理性能的关键.

本文通过对现有商用 FPGA 硬件结构的分析,针对指令集动态扩展中的可重构资源建模和利用问题,提出一种基于资源的种类和位置的可重构资源模型 RTL-M(Resource Type and Location Model).该模型弱化了可重构资源的功能和数量信息,而使用资源种类和位置信息,并能够以此计算资源使用时的时间信息.同时,基于此模型,借鉴相关图论思想,研究了一种针对粗粒度可重构资源的图多遍着色指派算法.该算法将定制指令的资源指派等价为一个图多遍着色问题,通过有效提取 RTL-M 模型提供的属性参数和系统限制条件完成资源指派过程,有益于提高资源利用率和系统性能.

## 2 相关工作

作为一种最普遍的通用可重构硬件,FPGA 在包括可重构计算系统在内的各种嵌入式领域广泛使用.目前,市场上主流的商用 FPGA 主要有 Xilinx 公司的 Virtex 系列和 Altera 公司的 Stratix 系列<sup>[4,5]</sup>.从顶层结构看,二者都采用一种“岛式”结构,由布线资源在水平和垂直方向上构成可编程布线通道,将内部的可编程逻辑资源和分布在四周的 I/O 控制块分割包围.布线通道中不同类型的布线资源负责完成所有资源间的数据传输.

Virtex 系列 FPGA<sup>[4]</sup>中的基本逻辑资源是可配置逻辑块 CLB(Configurable Logic Block),同时片上集成了专用存储器和专用乘法器等特殊资源.所有逻辑资源都通过局部互连的开关矩阵连接到布线通道.水平和垂直方向的布线通道拥有相同的结构,由几种具有不同长度和传输延时的布线资源组成.而 Stratix 系列 FPGA<sup>[5]</sup>中的基本逻辑资源是逻辑阵列块 LAB(Logic Array Block),片上特殊逻辑资源为专用存储器和 DSP 模块(由专用乘法器和累加器构成).LAB 通过局部互连资源连接到布线通道.同 Virtex 系列 FPGA 不同,Stratix 系列 FPGA 中水平和垂直方向的布线通道拥有不同的结构.

学术界和工业界存在很多对 FPGA 进行抽象建模的研究工作,传统的具有代表性的 FPGA 模型主要有 VPR<sup>[6]</sup>、UBO<sup>[7]</sup>和 HF/THF<sup>[8]</sup>.VPR 是一个基于细粒度岛式结构的结构级模型;UBO 和 HF/THF 是功能级模型,主要描述器件上各种资源的功能,对布线资源描述得比较简单.这三个模型都没有描述片上资源的数量和位置,比较适合于硬件综合、布局布线算法以及新的可重构硬件体系结构研究,而无法提供编译过程相关的循环体映射布局算法、指令集扩展实现以及软硬件代

码划分等问题所需的底层 FPGA 的硬件结构参数.Li 等人<sup>[9]</sup>的工作是近年来在可重构资源建模上较新颖的研究成果.他们提出了一种利用 XML 语言对粗、细粒度可重构资源实现一致化建模的方法,通过对资源的多层划分打包和构建描述库实现模型化.不过其方法的适用性需进一步验证.

基于各自不同的可重构资源建模方法,针对软硬件任务的性能、功耗、面积等不同优化目标,不少学者开展了相关的研究工作.Ito<sup>[10]</sup>在粗粒度的可重构功能单元模型上,为降低可重构功能单元上运行的任务相互间的通信延迟,对任务执行时间进行了静态分析,研究了如何针对任务操作类型和时间完成相邻可重构功能单元的动态重构并进行指派的方法,以达到快速处理的目的.Eguro<sup>[11]</sup>针对领域专用的 FPGA,就如何利用其专用性的可重构功能单元和结构化的通信资源以定制面向不同领域应用最优系统的问题,从性能、吞吐量和面积等因素考虑了三种可重构资源的分配算法.通过比较分析,Eguro 认为其改进的基于面积约束的分配算法在面向特定应用领域时是一个较好的选择.Plaxton<sup>[12]</sup>、Gavrilovska<sup>[13]</sup>等都扩展 FPGA 的虚拟化思想,研究了资源共享化的指派策略.只是这类方法受限于 FPGA 资源虚拟后的有效性,并且会对每个处理任务指派同样数量的资源,这就无法有效适应任务的异构性.Shafique<sup>[14]</sup>等将少数者博弈的思想应用于可重构资源的分配过程,提出了一种在多核处理架构下实现资源公平分配的方法,不过其系统开销是需要注意的地方.Li 等<sup>[15,16]</sup>在其研究的部分可重构的异构多核片上系统设计方法中,介绍了一种具有一定通用性的保证任务吞吐量的资源最小化分配策略,并提出了其与启发式算法结合使用以适应较复杂应用任务的可能.

## 3 RTL-M 资源模型

通过对现有商用 FPGA 硬件结构的分析,本文提出一种根据资源的种类和分布位置构建的粗粒度 FPGA 硬件模型 RTL-M.该模型弱化对计算资源功能和互连资源数量的描述,主要提供其种类和位置信息,并以此估算资源使用时的硬件配置时间和执行时间,从而能为诸如指令集扩展的可重构实现等研究工作提供需要的 FPGA 硬件参数.

### 3.1 资源种类和位置

在 RTL-M 中,FPGA 资源被归纳为三种:计算资源、接口资源和互连资源:(1)计算资源:主要负责完成逻辑计算过程,包括了通用计算资源和多种特殊计算资源.每个计算资源具有一定的面积(宽度×高度),并规定特殊计算资源的宽度和高度都是通用计算资源的整数倍;(2)接口资源:主要负责 FPGA 片上资源与片外资

源的信息交互,每个接口资源控制着某个位置区域中,同一行或列的计算资源所对应的一个或多个芯片引脚;(3)互连资源:负责在各计算资源之间,以及计算资源和接口资源之间传送数据。

上述三种资源散布在 FPGA 芯片的不同位置.其中计算资源处于 FPGA 内部,占据较大的面积;接口资源位于外围,构成了 FPGA 的边界;而互连资源则位于计算资源四周及计算资源和接口资源之间.互连资源将 FPGA 芯片分割为一个二维的方格阵列.每个计算资源占用一个或多个方格,位于相邻位置的同种计算资源聚合成一个二维的矩形区域,而接口资源区域则处于不同边界位置,被看作一维线性区域.图 1 给出了一个根据资源种类和位置划分了资源区域的 FPGA 硬件结构示例。

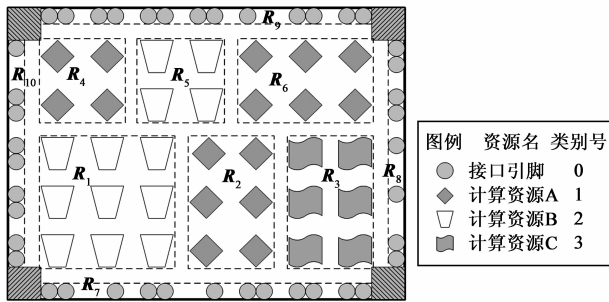


图1 可重构资源种类与位置示例

### 3.2 资源属性

RTL-M 中计算资源和接口资源的属性包括作为资源种类标识的唯一类别号和其他便于实现资源配置的相关属性. RTL-M 中规定接口资源的类别号为 0,通用计算资源的类别号为 1.在对 FPGA 片上资源进行了行列编号的基础上,按区域聚合的各种资源的属性定义主要如下:

(1)计算资源区域,主要包括了种类、宽度、高度和位置四个属性.(a)种类(TYPE):使用该区域中包含的计算资源的类别号表示;(b)宽度(WIDTH)和高度(HEIGHT):表示该区域包含的资源阵列的列数和行数;(c)位置(LOCATION):由该区域中左下角的计算资源的列号和行号决定,以形如<列号,行号>的元组表示.规定 FPGA 芯片左下角的计算资源为<1,1>.

(2)接口资源区域,主要包括长度、分布和位置三个属性.(a)长度(LENGTH):即该区域中包含的接口资源的个数;(b)分布(DISTRIBUTION):以一维向量表示,其中含有 LENGTH 个元素,每个元素代表相应接口资源控制的引脚个数;(c)位置(LOCATION):接口资源区域都处于 FPGA 四个边界,对应下、右、上、左四个位置,取值分别为 1,2,3,4.

(3)互连资源,只考虑其长度和传输延迟两个属

性.(a)长度(LENGTH):根据连接的两个资源的位置信息进行计算;(b)传输延时(DELAY):主要根据该资源的长度属性进行估算。

一般情况下,RTL-M 中主要描述计算资源区域和接口资源区域,假定互连资源数量无限,其属性可通过前两类资源的属性推算而得.图 1 是一个具有三种计算资源的 FPGA 芯片示例,其中互连资源位于图中空白处.其对应的资源模型属性参数如表 1 所示。

表 1 资源属性示例

| 区域       | 种类 | 宽度/长度 | 高度/分布           | 位置    |
|----------|----|-------|-----------------|-------|
| $R_1$    | 2  | 3     | 3               | (1,1) |
| $R_2$    | 1  | 2     | 3               | (4,1) |
| $R_3$    | 3  | 2     | 3               | (6,1) |
| $R_4$    | 1  | 2     | 2               | (1,4) |
| $R_5$    | 2  | 2     | 2               | (3,4) |
| $R_6$    | 1  | 3     | 2               | (5,4) |
| $R_7$    | 0  | 7     | <2,1,2,1,2,2,2> | 1     |
| $R_8$    | 0  | 5     | <2,1,1,2,2>     | 2     |
| $R_9$    | 0  | 7     | <2,2,2,2,1,2,2> | 3     |
| $R_{10}$ | 0  | 5     | <2,2,2,2,1>     | 4     |

## 4 RTL-M 的可重构资源指派算法

### 4.1 可重构资源的粗粒度指派原则

在上述 RTL-M 资源模型中,根据种类和位置划分的资源区域构成了一种粗粒度的可重构资源,以它们为基本组成单元,通过组合和互连等方式形成各个定制指令的功能部件.资源指派算法以这些可重构资源的相关资源属性作为算法参数,结合定制指令的资源需求和执行时间等约束条件,完成定制指令指派.所有可重构资源在指派过程中,被视为一个一维数组.指派的结果是一个可重构资源的多覆盖图.图 2 给出了一个资源指派结果的示例。

| 图例    | 指令 | $R_1$ | $R_2$ | $R_3$ | $R_4$ | $R_5$ | $R_6$ | $R_7$ | $R_8$ | $R_9$ | $R_{10}$ |
|-------|----|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|
| $I_1$ | 1  | 0     | 0     | 1     | 0     | 0     | 1     | 0     | 0     | 0     | 0        |
| $I_2$ | 0  | 1     | 0     | 0     | 0     | 1     | 0     | 1     | 0     | 0     | 0        |
| $I_3$ | 1  | 1     | 1     | 0     | 0     | 1     | 0     | 1     | 0     | 1     | 0        |
| $I_4$ | 0  | 0     | 0     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0        |

图2 可重构资源指派结果示例

定制指令的可重构资源指派过程遵循如下基本原则:(1)若两条定制指令不会同时执行,则可将某个可重构资源同时指派给这两条指令;(2)若两条定制指令会同时执行,则它们不能同时占用同一个可重构资源.该资源优先指派给较早开始执行的定制指令。

## 4.2 可重构资源指派的图多遍着色算法

可重构资源构成的功能部件,可能被指派给多条具有相同功能的定制指令实例.从编译器的角度看,这些功能部件可以被视作类似寄存器的硬件资源,通过复用传统编译程序中的寄存器指派算法就能完成其指派.但是,功能部件由一个或多个可重构资源组成,其数目和结构具有可变性,且可能存在某些可重构资源的重叠使用.因此,为了能够有效提高资源的利用率,需要从定制指令与可重构资源之间的关系来看待资源的指派.而如图 2 所示,这种指派关系具有多对多映射的特点.相比于传统寄存器指派问题被等价于一个图着色问题,可重构资源的指派问题被视为更复杂的图多遍着色问题,其定义如下.

对于一个有向图  $G(V, E)$ ,如果  $G$  满足如下三个条件,则认为  $G$  是  $K$  可多着色的:

- ① 对于  $\forall v \in V$ , 满足  $\Phi(C(v)) = W(v)$ .
- ② 对于  $\forall e: a \rightarrow b \in E$ , 有  $T(a) \geq T(b)$ .
- ③ 对于  $\forall e: a \rightarrow b \in E$ , 若  $T(a) = T(b)$ , 满足  $C(a) \cap C(b) = \emptyset$ .

其中,  $V$  为  $G$  的顶点集合,每个顶点  $v$  对应一条定制指令;  $E$  为  $G$  的边集合,边  $e: a \rightarrow b$  表示  $a, b$  两个顶点所对应的定制指令之间存在执行冲突关系,且  $a$  指令先于  $b$  指令开始执行.  $K = \{0, 1, \dots, k-1\}$  为颜色集合,其颜色元素与可用于指派的资源一一对应.  $C(v)$  为顶点  $v$  的着色结果,是  $K$  的一个子集,表示相应的可重构资源被指派给  $v$  对应的定制指令.  $\Phi(C(v))$  是对所有指派给顶点  $v$  的可重构资源的权值计算函数,考虑了各可重构资源的类型、位置和面积等因素.  $W(v)$  为权值函数,指示了顶点  $v$  对应的定制指令所需的资源权重,其值通过资源模型属性和定制指令的编译信息进行估算.  $T(v)$  指示顶点  $v$  在第几遍时被着色.

针对该问题,实现了一个考虑系统资源约束的图多遍着色算法 GMTCA (Graph Multi-Times Coloring Algorithm),其伪代码如算法 1 所示.算法输入为根据编译信息构造的定制指令执行冲突关系图  $G$ ,输出为如图 2 所示的二维着色数组  $C$  及着色遍数数组  $T$ .

算法 1 资源指派的图多遍着色算法伪码

```
* Graph Multi-Coloring Algorithm * /
Multi-Coloring( G )
{
    Times = 0; Clear( T ); Clear( C ); // Initialization
    W ← Priority( G, ResourceModel ); // Calculate the weigh of each v in G
    while( constraints( G ) = OK ) {
        Candidateset ← PrepareNodes( G );
        Times ← Times + 1; // Set the current times for v in CandidateSet
        while( candidateNodeSet! = NULL ) {
```

```
In 0 Set ← InZero( candidateSet ); // Select the in-degreis 0
v ← HighestPriority( In 0 Set ); // and highest priority v in G
if( Colorable( v ) = TRUE ) // Color v in Multi-Coloring Graph
    Color( C, V ); T( v ) ← Times; ModifyConstraints( G, V );
} // if
// Modify the available color sets and in-degress of neighbors of v ;
ModifyNeighbor( Candidateset, v );
Remove( CandidateSet, v )
} // while
} // while
Unallocated( G ); // Handle the unsuccessfully allocated v in G
}
```

具体步骤描述如下:

(1) 根据资源模型属性和相关编译信息,计算  $G$  中各个顶点的权值.高频率、小延迟和大面积的顶点,在提高性能和资源使用率上具有优势,其权值较高.

(2) 将所有未着色顶点列入待着色顶点集合,初始化各顶点可用颜色集合为  $K$ .

(3) 选择入度为 0 的权值最高的待着色顶点  $v$ ,若其可着色(即满足  $W(v)$ ),则设置其着色遍数信息  $T(v)$ ,并在数组  $C$  中着色(资源对应的数组元素置为 1),然后将  $v$  移出待着色顶点集合,修改其相邻顶点的入度和可用颜色集合;若  $v$  不可着色,直接将其移出待着色顶点集合并修改相邻顶点属性.

(4) 若待着色顶点集合不为空,则返回(3);否则,判断当前总的资源使用和重构开销等约束条件是否满足:若满足,则返回(2),继续下一遍着色;若不满足,则结束着色.着色结束时,由于约束条件的关系,可能存在未被着色的顶点,将给出提示并作一定的替换处理.

## 5 实验分析

### 5.1 实验环境

为检验 RTL-M 和 GMTCA 的可行性与有效性,进行了相关的实验分析.首先,对 Xilinx XC2V40 型 FPGA(封装规格为 fg456)进行了硬件结构建模,利用 XML 语言完成模型描述供指派算法使用.模型中的参数通过 Xilinx 公司的集成开发工具集 ISE 10.1 获得.利用 ISE 中的 FloorPlanner 工具得到所有资源区域的种类、高度、宽度、位置等属性,利用 Time Analyzer 和专门设计的测试程序计算互连资源的传输延迟参数等,结合 Xilinx 公司提供的相关数据手册得到各种资源的配置数据大小.其次,以 gcc 3.4.0 为基础实现了一个面向 PowerPC 指令集的交叉编译环境,并移植 SimpleScalar 模拟器以执行 PowerPC 指令代码.测试程序选用标准测试集 NetBench 中的 crc 和 md5 两个程序,利用相关工作<sup>[17]</sup>进行定制指令的生成选择,结合编译器提供的相关信息,人

工分析获得测试程序的定制指令执行关系图。

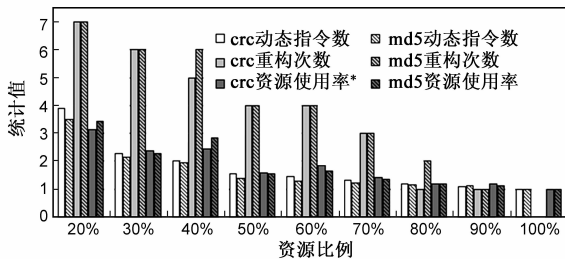
## 5.2 实验结果

分析测试程序得到的定制指令情况如表 2 所示。

表 2 定制指令分析结果

| 测试程序 | 定制指令类型 | 定制指令数目 | 最大使用的资源量 |
|------|--------|--------|----------|
| crc  | 15     | 1020   | 5880     |
| md5  | 9      | 2754   | 32410    |

实验中首先假定可重构资源充足,计算出理想情况下刚好能够满足所有定制指令都完成资源指派的最小资源数  $R_M$ 。由于实际情况中可重构资源的数目有限,多数情况下算法可能无法一次性完成所有定制指令的资源指派,需进行多遍指派从而产生运行时重构。通过限制可重构资源数目进行实验,结果如图 3 所示。其中横轴为资源数,选择了  $30\% R_M$ 、 $50\% R_M$ 、 $80\% R_M$  和  $R_M$  四种情况;纵轴给出了三种实验结果。动态指令数以理想情况下程序指令数目的倍数来表示,资源使用率由成功指派给定制指令的资源数除以当前资源数与重构次数的乘积来求得(为了更便于图形显示和统一纵轴坐标数值,图中使用其倒数数值表示)。



\*为统一纵轴坐标值,将资源使用率的计算结果用倒数数值的形式表示

图3 不同资源情况下的指派结果

运行时重构的系统开销是比较大的,实际应用中应限制系统重构次数。在资源数目和重构次数都受限的条件下,可能会出现算法无能完成对某些定制指令的资源指派的情况。图 4 给出在不同重构条件下,统计得到的不能完成资源指派的定制指令在所有待指派指令中的比例。图例中的取值  $m$  表示单次指派可用的资源数,用理想资源数的百分比表示;横轴坐标  $n$  表示允许的重构次数。从图中曲线可以看出,不可指派的情况随着资源数目和重构次数的变化而呈现一定的规律:

(1)当  $m$  相同而  $n$  不同时,随着  $n$  增大,不可指派情况减少,但当  $n$  增大到一定值后,这种减少趋势变缓直至不再变化。

(2)当  $n$  相同而  $m$  不同时,随着  $m$  增大,不可指派情况减少,当  $m$  达到一定值后将完全消失。

(3)在一定的资源比例范围内,若  $m$  与  $n$  的乘积  $s$  相近,则不可指派的定制指令的比例也相近。

进一步分析发现,上述现象主要因为程序中存在

少数资源需求较大的定制指令,在  $m$  较小时,无论  $n$  如何增加,都无法完成其指派。 $m$  越小,这类相对大的定制指令就越多。而随着  $m$  增大,这类指令的需求得以满足,则未指派的定制指令的比例明显降低。同时,对于(3),实验发现的比例区间为  $[30\%, 80\%]$ 。可以认为在这一范围内,每重构一次,可用的资源数近似增加一倍, $s$  就表示了指派过程中的总可用资源数目。 $s$  值相近,则系统完成定制指令指派的能力也就相近。

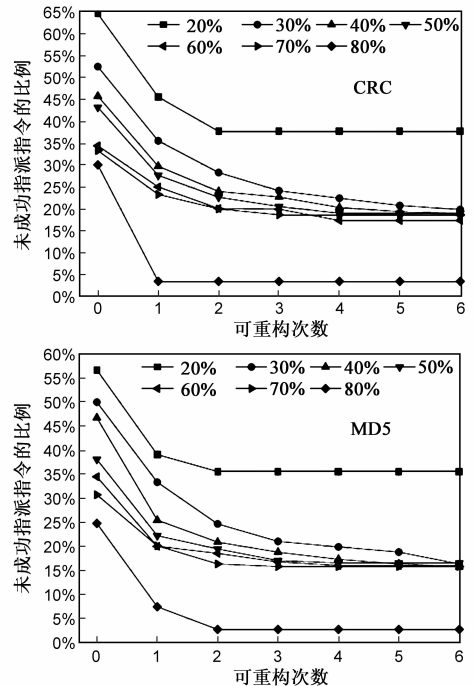


图4 不同资源条件下的定制指令指派比例

## 6 结束语

可重构计算系统中的可重构资源利用,因其影响着系统性能、功耗等多方面的关键指标而倍受关注。通过对常用可重构资源特性的分析,本文详细介绍了一个主要描述资源种类和位置信息的模型,提出了一种基于该模型的资源指派算法。该算法利用图多遍着色的思想实现可重构资源的指派使用。相关实验结果表明了所做工作的有效性,对提高系统可重构资源使用率和性能具有指导意义。后续工作拟对资源模型中的资源进行细粒度分割,并扩展指派算法,实现对混合粒度可重构资源指派使用的支持。

### 参考文献

- [1] Jain M K, Balakrishnan M, Kumar A. ASIP design methodologies: survey and issues [A]. Proceeding of 14th International Conference on VLSI Design [C]. Los Alamitos, CA: IEEE Computer Society, 2001. 76-81.

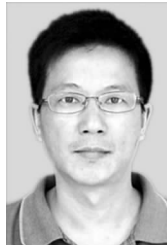
- [2] Galuzzi C, Bertels K. The instruction-set extension problem: a survey[J]. ACM Transactions on Reconfigurable Technology and Systems, 2011, 4(2): 1 – 28.
- [3] Barat F, Lauwereins R. Reconfigurable instruction set processors: a survey[A]. Proceedings of 11th International Workshop on Rapid System Prototyping [C]. Los Alamitos, CA: IEEE Computer Society, 2000. 168 – 173.
- [4] Vasicek Z, Sekanina L. An evolvable hardware system in Xilinx Virtex II Pro FPGA [J]. International Journal of Innovative Computing and Applications, 2007, 1(1): 63 – 73.
- [5] Lewis D, et al. The Stratix II logic and routing architecture [A]. Proceedings of 13th International Symposium on Field-Programmable Gate Arrays [C]. New York: ACM, 2005. 14 – 20.
- [6] Li X, Yang H, Zhong H. Use of VPR in design of FPGA architecture [A]. Proceedings of 8th International Conference on Solid-State and Integrated Circuit Technology [C]. Shanghai: IEEE Press, 2006. 1880–1882.
- [7] Lagadec L, Pottier B. Object-oriented meta tools for reconfigurable architectures [A]. Proceedings of 2000 International Conference on Modeling, Signal Processing and Control [C]. Bellingham WA: SPIE, 2000. 69 – 79.
- [8] Bossuet L, Gogniat G, et al. A modeling method for reconfigurable architectures [A]. Proceedings of 2nd International Workshop on System-on-Chip for Real-Time Applications [C]. Alberta, Canada: IEEE Computer Society, 2002. 170 – 179.
- [9] Li Z, et al. A modeling and mapping method for coarse/fine mixed-grained reconfigurable architecture [A]. Proceedings of 11th International Conference on Solid-State and Integrated Circuit Technology [C]. Xi'an: IEEE Press, 2012. 1 – 4.
- [10] Ito K. A scheduling and allocation method to reduce data transfer time by dynamic reconfiguration [A]. Proceedings of 2000 Asia and South Pacific Design Automation Conference [C]. New York: ACM, 2000. 323–328.
- [11] Eguro K. Resource allocation for coarse-grain FPGA development [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 24(10): 1572 – 1581.
- [12] Plaxton C G, Yu S, et al. Reconfigurable resource scheduling [A]. Proceedings of 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures [C]. New York: ACM, 2006. 93 – 102.
- [13] Cavrilovska A, Kumar S, et al. High-performance hypervisor architectures: virtualization in HPC systems [A]. Proceedings of 1st Workshop on System-Level Virtualization for High Performance Computing [C]. New York: ACM, 2007. 1 – 8.
- [14] Shafiq M, Bauer L, et al. Minority-game-based resource allocation for run-time reconfigurable multi-core processors [A]. Proceedings of 2011 Design, Automation & Test in Europe [C]. New York: ACM, 2011. 1 – 6.
- [15] Li J, Das A, Kumar A. A design flow for partially reconfigurable heterogeneous multi-processor platforms [A]. Proceedings of 23rd IEEE International Symposium on Rapid System Prototyping [C]. Tampere, Finland: IEEE Reliability Society, 2012. 170 – 176.
- [16] 齐骥, 李曦, 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.  
Qi Ji, Li Xi, et al. Algorithms of resource management for reconfigurable systems based on hardware task vertexes [J]. Acta Electronica Sinica, 34(11): 2094 – 2098. (in Chinese)
- [17] 周学海, 等. 基于差分进化和贪心策略的自定义指令选择算法研究 [J]. 电子学报, 2009, 37(2): 372 – 376.  
Zhou Xue-hai, et al. Study on differential evolution and greedy strategy based custom instruction selection algorithms [J]. Acta Electronica Sinica, 2009, 37(2): 372 – 376. (in Chinese)

#### 作者简介



**张惠臻** 男, 1983 年 10 月出生, 福建龙岩人. 分别于 2005 年和 2010 年在中国科学技术大学获工学学士和工学博士学位, 现为华侨大学计算机学院讲师, 主要从事嵌入式软硬件协同设计、可重构计算、编译性能优化等方面的研究工作.

E-mail: zhanghz1006@gmail.com



**谢维波(通信作者)** 男, 1964 年 10 月出生, 福建泉州人. 分别于 1988 年和 2010 年在哈尔滨工业大学和华侨大学获工学硕士和工学博士学位. 现为华侨大学计算机学院教授、硕士生导师, 主要从事以智能信息与信号处理为基础的人工智能研究和以计算机硬件、底层系统软件为基础的逻辑系统及集成研究.

E-mail: xwblxf@hqu.edu.cn



**李 曦** 女, 1992 年 12 月出生, 湖北荆门市人. 华侨大学计算机学院网络工程系 2010 级本科生, 主要从事嵌入式系统设计与应用方面的研究.

**洪 欣** 女, 1977 年 10 月出生, 福建莆田人. 分别于 2001 年和 2004 年在华侨大学获工学学士和工学硕士学位. 现为华侨大学计算机学院副教授, 主要从事软件工程、大数据等方面的研究工作.

E-mail: hxpotato@163.com