

一种基于可重构多 FPGA 的任务调度与任务复制方法^①

郭向金, 尤志强

(湖南大学 软件学院, 长沙 410082)

摘要: 在可重构多现场可编程逻辑门阵列(FPGA)系统中, 任务调度是一个极其重要的研究方向。参照同构与异构计算领域的调度算法, 结合可重构多 FPGA 计算模型的自身特点, 在现有的调度算法的基础上, 将任务复制方法引入到可重构多 FPGA 系统计算领域, 如果任务余图最长路径上的父子节点不在同一 FPGA 上, 通过寻找 FPGA 上的复制空间, 提出的算法将父节点尽可能复制在子节点所在的 FPGA 上, 减小了任务之间的通信开销。实验结果表明, 对于任务调度有向无环图, 提出方法的调度长度优于或等于前人方法的性能下界, 而且, FPGA 利用率有所提高。

关键词: 重构; 多现场可编程逻辑门阵列; 有向无环图; 任务复制; 调度长度

Task Schedule and Task Duplicate Method Based on Reconfigurable Multi-FPGAs

GUO Xiang-Jin, YOU Zhi-Qiang

(School of Software, Hunan University, Changsha 410082, China)

Abstract: Task scheduling is an extremely important research direction in the reconfigurable multi-FPGAs systems. Referencing to the scheduling algorithm in homogeneous and heterogeneous computing, combining its own characteristics of the reconfigurable multi-FPGA computing model, and based on the existing scheduling algorithm, task duplication method is introduced in this paper. If the parent and son nodes on the longest path of the excess task graph are not on the same FPGA, through looking for the Duplicate Space on the FPGA, the proposed algorithm will duplicate the parent node onto the FPGA where the child node is, reducing the cost of communication between tasks. The experimental results show that for a directed acyclic graph of the task scheduling, the makespan of the proposed method is superior to or equal to the performance lower bound of the previous methods, moreover, the utilization of FPGA has been increased.

Key words: reconfigurable; multi-FPGA; directed acyclic graph; task duplicate; makespan

FPGA 容量上的增长速度远落后于 ASIC 芯片规模的增长速度, 只有它的 1/10 到 1/5^[1]。可重构多 FPGA 系统成为提供大规模逻辑容量的一种可行方案。在可重构多 FPGA 系统中, 任务图的任务模块全部映射成硬件任务, 将这些硬件任务配置到 FPGA 上, 以达到高速计算的目的^[2,3]。

任务调度无论对于可重构单 FPGA 还是对于可重构多 FPGA 系统都非常重要, 合理的调度算法有效地对硬件任务进行调度与管理, 减少任务的总执行时间,

提高芯片利用率^[4,5]。肖艳等基于多 FPGA 系统模型(图 1)提出了一种离线的动态关键路径适应度(DCPFD)任务调度算法^[2]。DCPFD 算法通过动态识别任务图 DAG 的关键节点来确定调度顺序, 按适应度规则(即判断在恰当的时间是否存在恰当的配置资源)选择一块 FPGA 并将已调度的任务节点放置在 FPGA 上, 大幅度降低了任务图的调度长度。

本文首先介绍多 FPGA 可重构系统架构模型以及任务模型图, 并简介 DCPFD 调度算法。然后, 在

① 收稿时间:2011-03-21;收到修改稿时间:2011-04-30

DCPFD算法的基础上,通过寻找FPGA上的复制空间,探讨将父节点复制到子节点所在FPGA上的可能性,可行则进行任务复制从而减小任务之间的通信开销。文章将对复制时间的确定以及关键任务的子任务时间前移等问题展开论述。

1 可重构计算系统模型与任务模型

1.1 系统模型与任务模型

可重构计算结构由调度器与若干块FPGA组成^[2],如图1所示。在本文中,对FPGA采用灵活一维可重构模式,每个硬件任务运行在水平方向的一个可重构模块上,各个硬件任务宽度可以不同,垂直方向上在此任务完成前,它的剩余空间都不能被利用^[6]。对任务的放置采用左侧优先放置模型,如果一个任务在开始时刻在一个FPGA上没有空余位置可以放置,那么就需要查看其它FPGA有无足够大的空闲位置。

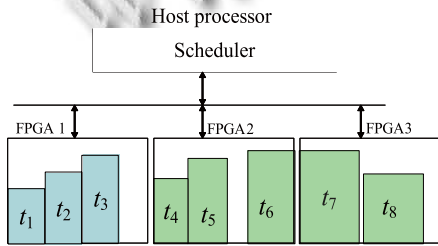


图1 可重构计算系统结构

本文采用有向无环图^[7,8]来描述应用,图中各个任务都是硬件任务,任务之间存在偏序关系,任务图上的有向边代表父子任务节点的通信开销,如图2所示。

硬件任务是指可重构系统中基于FPGA实现的功能模块,是能在可重构器件上执行的电路。硬件任务是任务调度研究的基本单位,一般用向量 $T(w,r,e,d)$ 描述,其中, w , r , e 和 d 分别代表硬件任务的宽度,重构时间,执行时间和终止时间,如表1所示。

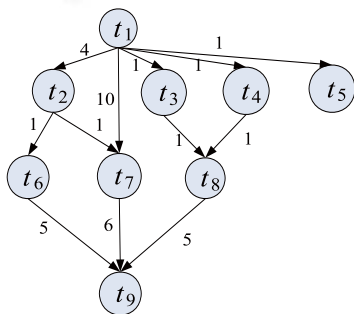


图2 任务图

表1 图1所示各个任务的属性

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
$r+e$	2	3	3	4	5	4	4	4	1
d	2	8	10	8	20	13	12	15	24
w	4	2	3	5	4	6	6	5	3

1.2 动态关键路径适应度算法^[2]

调度器使用动态关键路径(DCP)调度算法为到达的任务分配一块FPGA,并将任务放置于FPGA上一个恰当的位置。DCP调度首先需要识别关键节点,通过计算各个节点的最早开始时间(EST)与最迟开始时间(LST)来识别。如果一个节点的EST与LST相等,则认为是关键节点。该算法每次从关键节点中选择EST最小的节点进行调度。

任务节点 t_i 的EST与LST定义如下:

$$EST_i = \max_{1 \leq k \leq p} \{EST_{i_k} + r_{i_k} + e_{i_k} + c_{i_k}\} \quad (1)$$

$$LST_i = \min_{1 \leq m \leq q} \{LST_m - r_{i_m} - e_{i_m} - c_{i_m}\} \quad (2)$$

其中, p 是 t_i 父节点的数目, q 是 t_i 子节点的数目, i_k 是 t_i 的第 k 个父节点, r_{i_k} 是 i_k 的配置时间, e_{i_k} 是 i_k 的执行时间, c_{i_k} 是任务图中父子任务之间的通信开销。如果 t_i 是入点, 那么它的最早开始时间记为 0。

已知最早开始时间以及节点的配置与执行时间,则各个节点的最早完成时间(EFT)可以计算如下:

$$EFT_i = EST_i + r_{i_k} + e_{i_k} \quad (3)$$

取所有节点中最早完成时间最大值,就是任务整体的调度长度(LDCP)。如果 t_i 是出点,那么它的LST为:

$$LST_i = LDCP - r_{i_k} - e_{i_k} \quad (4)$$

文献[2]定义了适应度(FD)算法来判断一块FPGA是否存在足够大的空闲区域容纳分配而来的任务。FD算法计算任务的实际开始时间减去最早开始时间,当FPGA上没有足够大的空间容纳所分配来的任务时,如果该任务需要放置在此FPGA上,需要等待,这个等待时间记为 dfd 。 dfd 是一个非负的整数,值越小,就说明所调度的FPGA越有能力接受所调度的任务。

文献[2]对图2以及表1所对应的任务模型进行了DCPFD调度与放置,通过DCPFD调度与放置过程后,任务之间的通信开销得到了有效降低。任务经过DCPFD调度与放置过程后,任务余图的最长路径经过

计算为 $17(t_1 \rightarrow t_4 \rightarrow t_8 \rightarrow t_9)$ 。其中, t_1, t_2, t_4, t_9 放置在 FPGA1 上, t_3, t_8 放置在 FPGA2 上, t_5, t_6 放置在 FPGA3 上。其中 FPGA2 上的任务分配如图 3 所示。

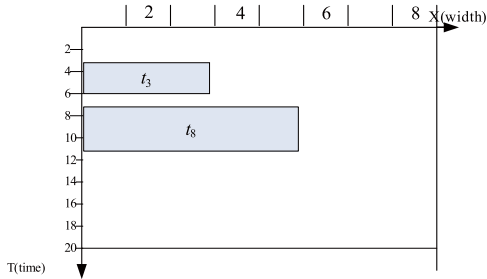


图 3 复制前 FPGA2 上的任务分配

2 复制策略

2.1 任务复制可行性分析

在复制过程开始之前, 需要对复制过程做可能性分析。首先, 自底向上查看任务余图上的关键路径的各个任务节点, 如果它们不在同一块 FPGA 上, 判断子节点所在的 FPGA 能否容纳该节点的父节点, 如果可以, 则进行任务复制。此时, 还需要确定父节点的复制时刻 t 以及子节点的前移时刻 k 。

由于父子节点之间具有偏序关系, 父节点必须要在子节点 t_i 的 EST_i 时间点之前完成, 设父节点的复制时刻为 t , 则

$EFT_{ancestor_{t_i}} \leq t \leq EST_i - e_{parent_{t_i}}$ 其中, $EFT_{ancestor_{t_i}}$ 为 t_i 的祖先节点最早完成时间 $EFT_{ancestor_{t_i}} = EST_{ancestor_{t_i}} + r_{ancestor_{t_i}} + e_{ancestor_{t_i}}$, $e_{parent_{t_i}}$ 为任务 t_i 的父节点的执行时间。

递减 t , 经由适应度算法, 寻找 dfd 为 0 时父节点最早的复制时刻。由于是硬件任务, 还涉及到子节点的时间以及位置的前移, 子节点的前移时刻 k , $k \in [0, t + e_{parent_{t_i}} - EST_i]$ 在各个时刻点经由适应度算法判断出最大的前移时刻 k , 置任务余图上 $Ctt_i = Ctt_i - k$ 。其中, Ctt_i 为任务 t_i 与任务 t_j 之间的通信开销。

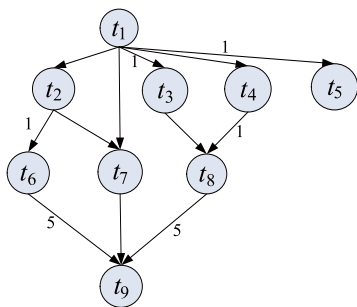


图 4 调度后产生的任务余图

图 4 为图 2 经过 DCPFD 过程之后产生的任务余图, 以图 4 为例。通过识别任务余图的最长路径, 逆序观察最长路径上各个节点所在的 FPGA, 可以看出, 最长路径 ($t_1 \rightarrow t_4 \rightarrow t_8 \rightarrow t_9$) 上的任务不在同一块 FPGA 上, t_1, t_4, t_9 在 FPGA1 上, 而 t_8 在 FPGA2 上。根据文献[3]中的方法, 经过适应度计算 FPGA1 不能容纳 t_8 , 否则整个任务的完成时间将变长。但在 FPGA2 上, t_8 开始执行时间之前只有任务 t_3 执行, 见图 3(b)。经过适应度判断过程可知, $dfd_{t_1, t_2} = dfd_{t_4, t_2} = 0$, 其中, dfd_{t_i, t_j} 表示将任务 t_i 放置在第 f_j 块 FPGA 上所需要的等待时间。FPGA2 完全可以容纳 t_1, t_4 , 这样 t_1, t_4 就可以复制到 FPGA2 上, t_4 的复制时刻 $t \in [2, 3]$, 根据适应度算法判断, t_4 可以在时刻 2 开始复制, 同理 t_1 可以在时刻 0 开始复制。由于 t_1, t_4, t_8 被放置在同一 FPGA 上, 他们之间的通信开销即可记为 0。节点 t_8 的最早开始时间从原来的 7 变为 6, 即将 t_8 前移至 6 时刻这时计算任务余图的最长路径, 发现路径上节点未发生变化, 表明不能继续任务复制过程。如果路径上节点发生变化, 则继续判断复制可能性, 可行则进行任务复制, 递归进行此过程。此例中, 复制完成后, 图的最长路径由 17 变为 16。图 5 为任务复制后 FPGA2 上的任务分配。

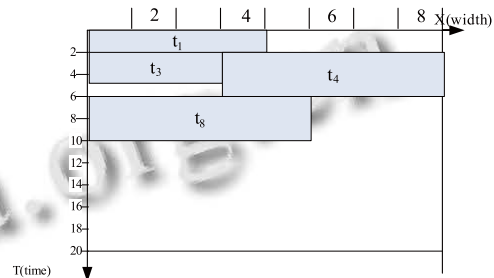


图 5 复制后 FPGA2 上任务分配

2.2 任务复制算法框图

DCPFD 过程结束后, 进行如下操作:

1) 判断任务余图的最短路径上的各个任务是不是都分配在同一 FPGA 上, 如果是, 则退出, 否则进行下面的操作。

2) 自底向上查看最短路径上节点所在的 FPGA, 结合适应度算法判断将父节点复制到此 FPGA 的可能性, 如果不可行, 则退出, 否则将父节点复制到该 FPGA 上。

3) 将子节点前移。重复上面过程, 直至不能复制,

过程终止。

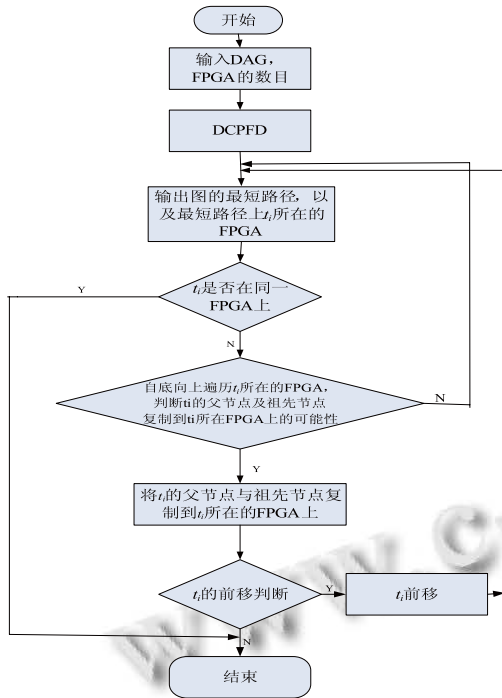


图 6 任务复制算法 (DCPFDD) 流程图

3 实验结果

3.1 调度长度比率

调度长度比率 (SLR) 是调度算法的主要性能指标。由于任务图具有不同特性, 因此有必要对调度长度进行标准化, 从而对调度算法的性能进行客观比较, SLR 的分母是图中关键路径上任务的执行时间的和, 分子是实际调度长度。显然, $SLR \geq 1$, 而且 SLR 越小, 表明该算法的调度长度越短, 性能越好^[8]。

$$SLR = makespan / (\sum_{t_i \in CP_{min}} (e_{t_i} + r_{t_i})) \quad (5)$$

其中, makespan 是任务图调度的最短路径。

3.2 加速比

加速比 (speed) 指的是同一个任务在多 FPGA 系统和单 FPGA 系统中单任务串行运行消耗的时间比率用来衡量并行系统的性能和效果, 加速比可用如下公式得到:

$$speed = \sum_{i=1}^n (e_{t_i} + r_{t_i}) / makespan \quad (6)$$

下面, 我们用随机生成的任务图对复制算法前后进行性能比较, 我们通过改变如下三个参数来产生任务图:

1) 任务节点个数 $V, V \in \{20, 40, 60, 80, 100\}$;

2) 任务级数 $L, 0.2V \leq L \leq 0.8V$;

3) 通信与计算时间比率 (CCR)。CCR 是任务图中节点间通信量的平均值与节点计算量的平均值的比率。CCR 反映了任务通信时间与计算时间的大致比例关系, $CCR \in \{0.5, 1.0, 2.0\}$ 。

对于任务的宽度, 如同文献[2], 我们设为 [0, 6] 的随机值, FPGA 的固定宽度设为 8。我们对 V 中每个任务图随机 L 与 CCR 产生 1000 张任务图。

图 7 与图 8 表示的为不同数目任务节点经过任务复制后调度长度比率 SLR 以及加速比 speed 的变化, 可以看出, 经过任务复制过程后, 调度长度变短, 而加速比变大。其中, 图 7 的纵坐标为 SLR, 图 8 的纵坐标为 speed。

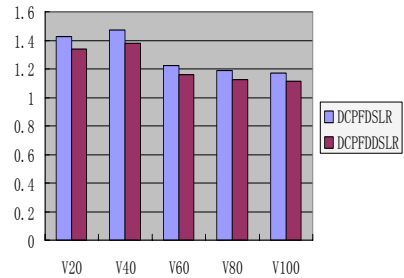


图 7 复制后 SLR 的变化

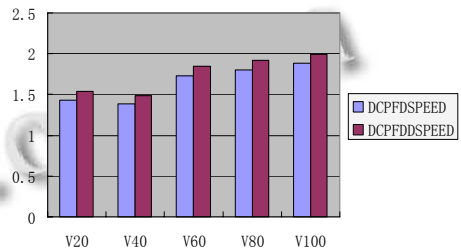


图 8 复制前后加速比变化

4 结语

本文提出一种任务调度与复制算法, 该算法通过递归计算任务图的最长路径, 判决任务复制的可能性, 将父任务节点复制到子任务节点所在的 FPGA 上执行, 从而减少父子任务节点之间的通信开销, 有效地降低任务整体的完成时间。任务调度与任务放置共同决定可重构计算的性能, 在任务放置方面, 本文采取的方法相对简单, 这是以后的一个重点研究方向。

(下转第 117 页)

其中五个电容均取 10 μ F。串口只用三根线,5 端为公共端连接系统地,2 端和 3 端则分别连接接收和发送端。该电路完成 TTL 电平与 RS232 电平转换。DB9 接口通过交叉串口线连到机上,以完成硬件的串行通信。其线路原理如图 7 所示。

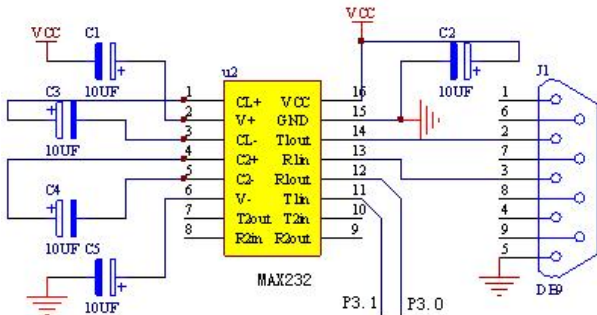


图 7 串行接口电路原理图

此外辅助模块主要包括时钟模块和液晶显示模块。DS1302 为系统提供所需要的时间参数,OCMJ4X8C 型大液晶显示系统时间、系统名称等辅助信息,其线路原理不一一赘述。

3 结语

本文提出的电梯控制仿真系统既能直观透彻地反映电梯的运行规律,达到事半功倍的效果,又能将新型计算机控制技术应用于电梯技术的教学和科研中,

为其提供先进的技术手段。本系统已经应用于现代电梯控制技术的教学中,取得了良好的教学效果。同时,本系统可以为深入开发电梯控制系统的关键技术提供实际有效的平台。

参考文献

- 1 岳庆来.电梯现代智能控制技术.北京:机械工业出版社,2009.4-28.
- 2 叶安丽.电梯控制技术(第 2 版).北京:机械工业出版社,2008.25-40.
- 3 刘剑,朱德文,梁质林.电梯电气设计.北京:中国电力出版社,2006.127-140.
- 4 陈继文,杨红娟,范文利,等.电梯控制集成仿真系统.山东建筑大学学报,2008,23(5):420-423.
- 5 宗群,罗欣宇,王中海.虚拟电梯系统的开发与应用.制造业自动化,2002,24(8):56-59.
- 6 石云.基于 PLC 的电梯控制系统的设计与实现.工业控制计算机,2009,22(4):5-6.
- 7 马殷元,姚闯.基于状态图的电梯控制建模及其 PLC 实现.计算机工程,2009,35(16):221-223.
- 8 文畅,黄琴.可编程控制器的电梯控制设计及监控研究.长江大学学报(自然科学版),2008,5(4):328-330.
- 9 钟晓.PLC 及变频器在电梯控制系统中的应用.装备制造技术,2009,(8):84-85.

(上接第 94 页)

参考文献

- 1 归敏单,吴锡生.HyperLynx 在多 FPGA 系统设计中的应用.自动化技术与应用,2008,40(2):102-104.
- 2 Xiao Y, Duan ZH. Dynamic Critical-Path Based on Fit Degree Scheduling for Reconfigurable Multi-FPGAs. IEEE International Conference on Application of Concurrency to System Design. Xi'an. 2008.27-32.
- 3 Pan Z, Wells BE. Hardware Supported Task Scheduling on dynamically reconfigurable SOC architectures IEEE Trans. on Very Large Scale Integration Systems. 2008. 1465-1474.
- 4 梁樑,周学功,王颖,彭澄廉.采用预配置策略的可重构混合任务调度算法.计算机辅助设计与图形学学报,2007,7:635-641.
- 5 Steiger MPLTC, Walder H. Online scheduling and placement of real-time tasks to partially reconfigurable devices. RTSS, 2003.224-235.
- 6 Saha P, Ghazawi TEI. Extending embedded computing scheduling Algorithms for reconfigurable computing systems. 3rd Southern Conference on Programmable Logic. 2007. 87-92.
- 7 Lam Y, Coutinho JGF, Luk W, Leong PHW. Mapping and scheduling with task clustering for heterogeneous computing systems. International Conference on Field Programmable Logic and Applications. 2008. 275-280.
- 8 Sim JE, Wong WF, Teich J. optimal placement-aware Trace-based Scheduling of hardware reconfigurations for FPGA accelerators. 17th IEEE Symposium on Field Programmable Custom Computing Machines. 2009. 279-282.