## 映射优化的CGRA算子调度研究

### 研究目的

通过优化数据流图（DFG）的算子调度，提升DFG在可重构计算阵列上的映射能力。

### 背景介绍

1. **可重构计算阵列**

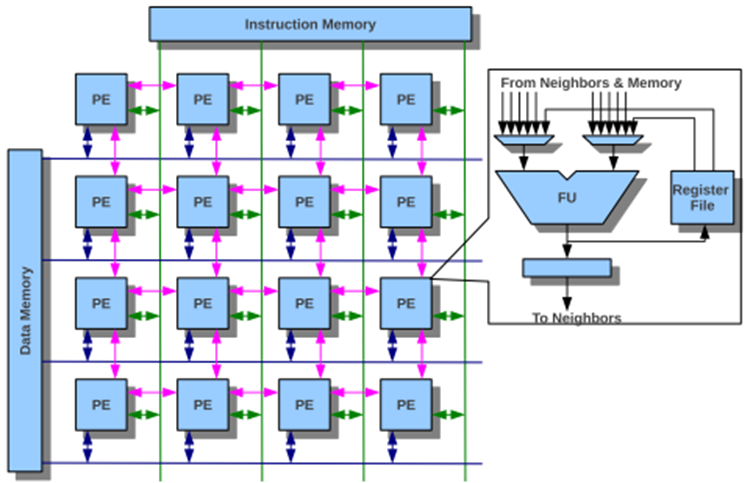


图1 可重构计算阵列

如上图所示，CGRA包含数据存储器、配置指令存储器以及处理单元（PE）阵列（PE array，PEA）。阵列大小一般为 22，4 和 88。每个PE内部包含一个功能单元（FU）和一个寄存器堆（RF）。PE之间可以有灵活的可配置的互连关系，每个PE中FU的两个输入可以来自于上下左右四个PE的输出。但是每个PE内部的RF只能被PE内部的FU访问。CGRA中FU的功能以及PE之间的互连关系可以动态重构，以实现不同的数据通路。

1. **时间扩展的CGRA（TEC）**

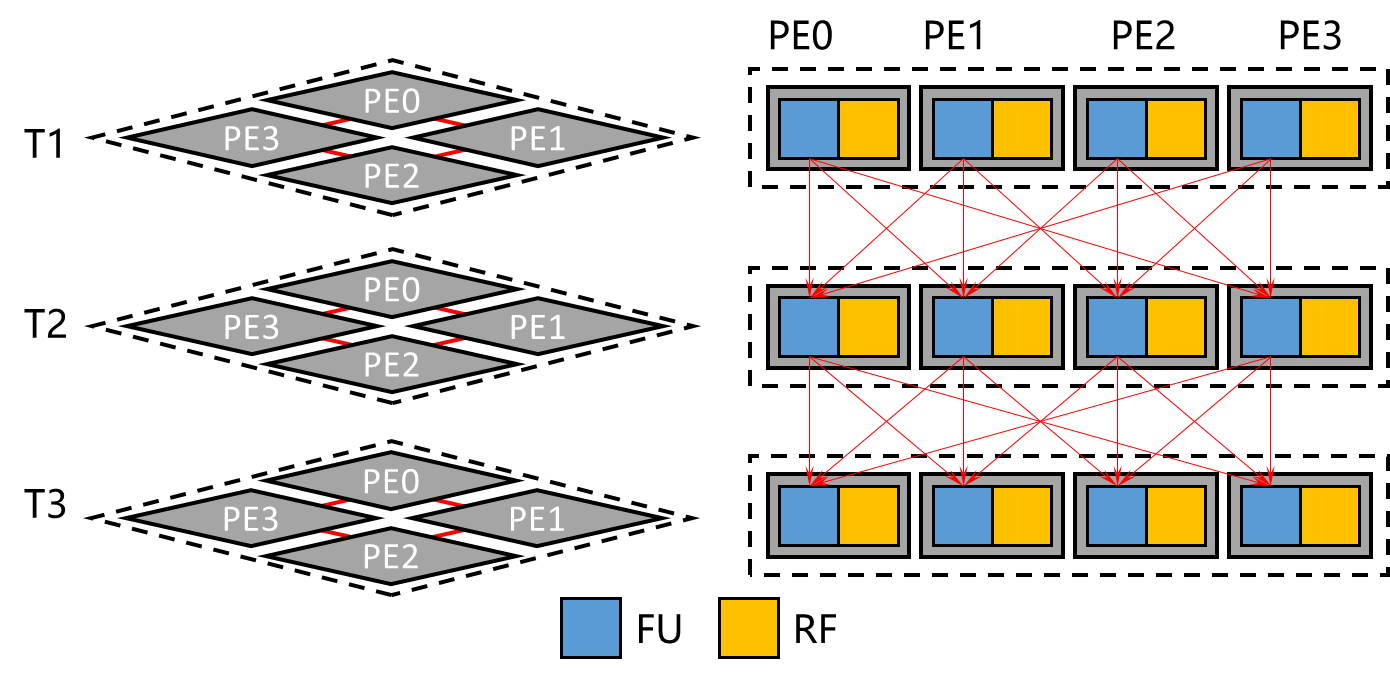


图2 时间扩展的CGRA

由于CGRA具有动态重构的能力，每个时钟周期PE的功能及其互连关系都可以动态重构。因此，DFG的映射问题实际上变成了一个将算子映射一个时间扩展的计算阵列上面。如图2左边所示，将一个22 的PEA时间扩展成时间高度为3的TEC。如果将TEC中的每一层资源进行平铺，就变成图2右部的资源图。蓝色的FU可以连接到邻居的FU。

1. **循环软件流水**

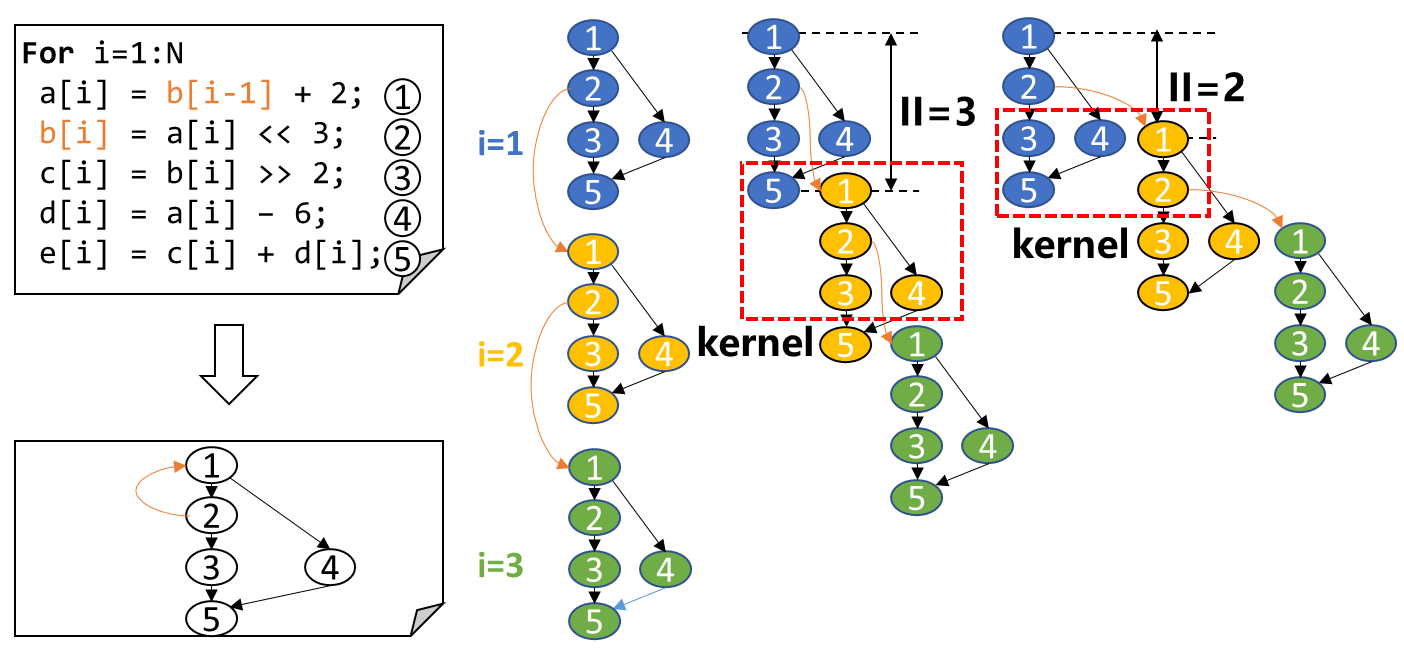
****

图3 循环软件流水示意图

如图3所示循环核心可以转换成带循环依赖的数据流图（DDG），包含迭代内部依赖（黑色箭头）和迭代间依赖（黄色箭头）。相邻循环迭代启动的时钟周期间隔称为启动间隔（Initiation Interval，II）。为了提升循环并行性，可以在不违背依赖约束的情况下缩小II。如图3所示，循环启动间隔依次缩短为II=3和II=2。循环流水之后，最小的重复单元称之为kernel，如图三中的红色虚线框所示。

1. **CGRA上的模调度**

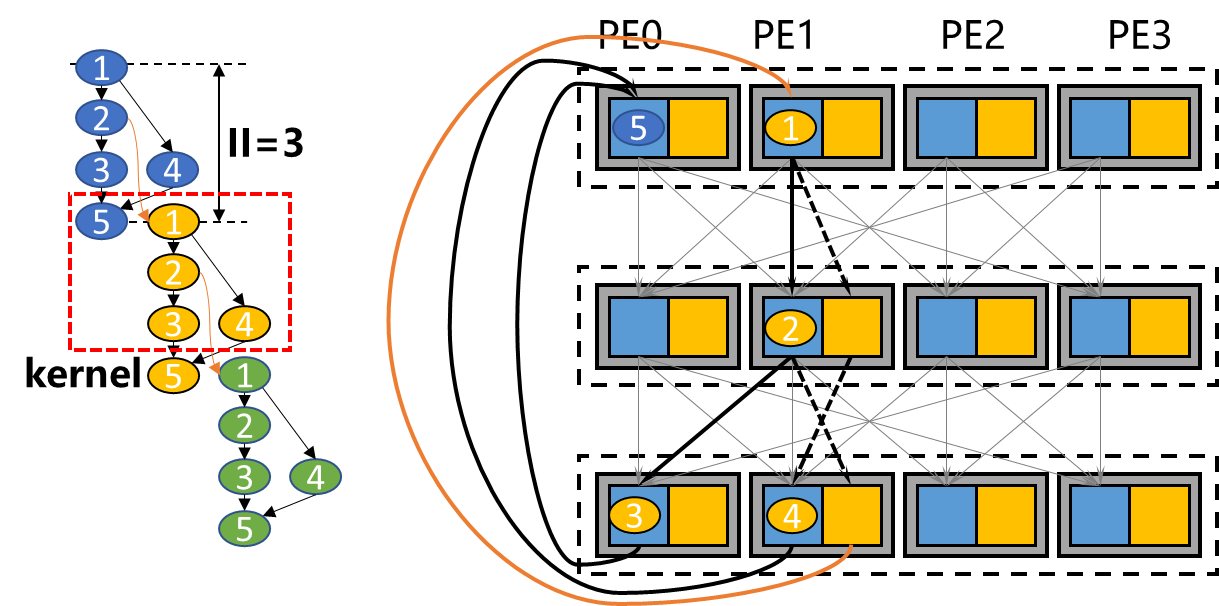


图4 CGRA上的模调度

如图4所示，经过循环流水之后所形成的kernel，可以映射到时间高度等于II的TEC上面去，这样就实现了CGRA上的循环模调度。映射的过程要满足几个约束条件：

（a）依赖约束：这个可以在软件流水和算子调度的时候进行满足。

（b）资源约束：放在TEC同一层的算子的数量不能超过PEA的PE的总数。

（c）长依赖约束：由于长依赖约定只能通过PE内部的RF传输，且RF只能别自己的PE访问，则长依赖的源节点和目标节点算子只能放在同一个PE上。如图4所示，迭代间依赖2->1属于一条长依赖，则2算子和1算子只能放在同一个PE上（PE1）。

### 研究问题A（难度系数0.9）

**假设**：资源类型只有一种（即PE资源），PE的数量为16，每个 PE 支持给定 DFG 中的任意算子，给定目标II。

**目标**：通过算子调度最小化 PE 的使用数量（即调度后kernel最宽的地方的宽度）

**约束：**

依赖约束：保证依赖的算子的先后顺序

资源约束：TEC 中同一层的算子数量不超过 PE 的数量

依赖长度约束：没有依赖长度等于II或II倍数的依赖

**建模**：可以用 ILP 建模，也可以用其他启发式算法。

### 研究问题B（难度系数1）

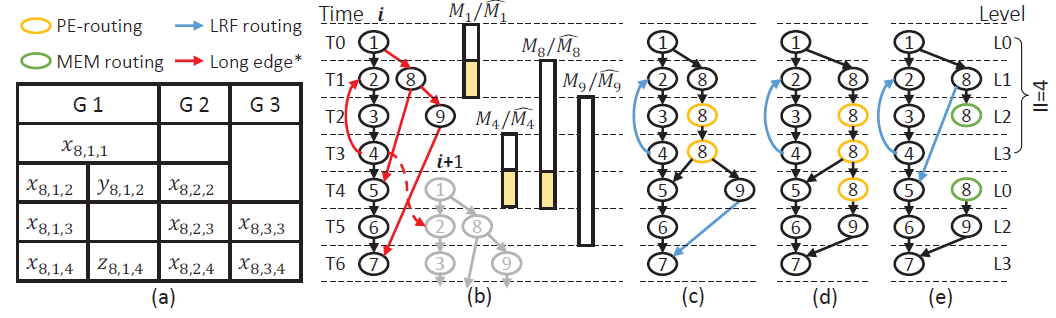


图5 增强算子调度

在研究问题A的基础上，扩展调度算法不仅支持算子机动性的调整，还支持长依赖的路由节点插入。目标函数如下：

（1）

权重系数，Npe表示最大资源使用量（即调度后DFG最大的宽度），N­ins表示插入的路由算子的数量。

### 项目评估

1. **研究报告 20 分**

* 背景描述
* 问题定义：目标+约束
* 问题求解
* 结果及讨论

至少给10个DFG（算子数量10-100），分析调度结果和编译时间

* 小组分工

1. **代码实现 20 分**

* 书写规范
* 功能验证

助教检查时任意给定算子数量小于100的 DFG，看能否产生正确调度

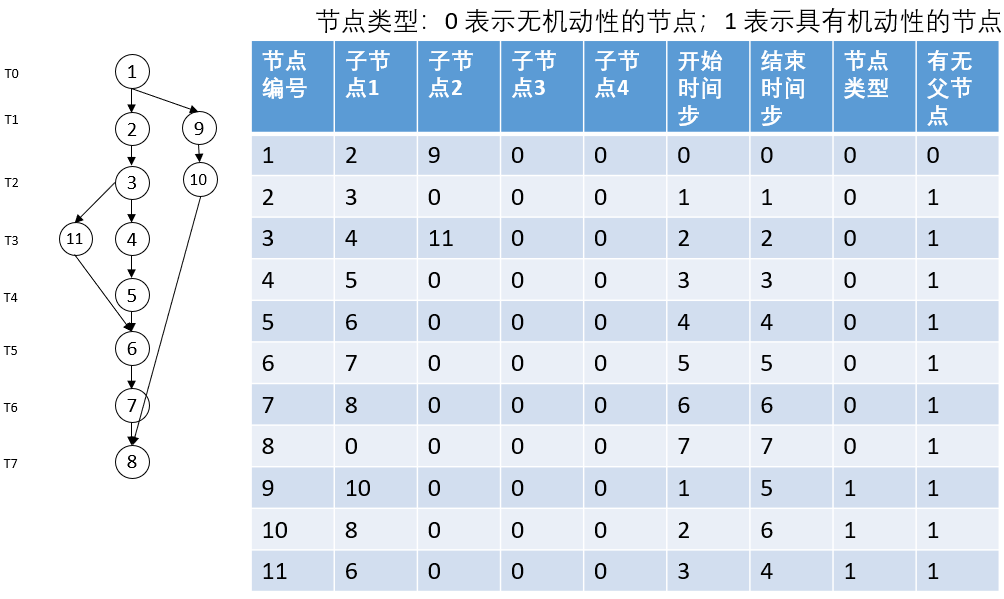
* 性能验证

助教检查时给定算子数量为100的DFG，看编译时间多长

### 附录

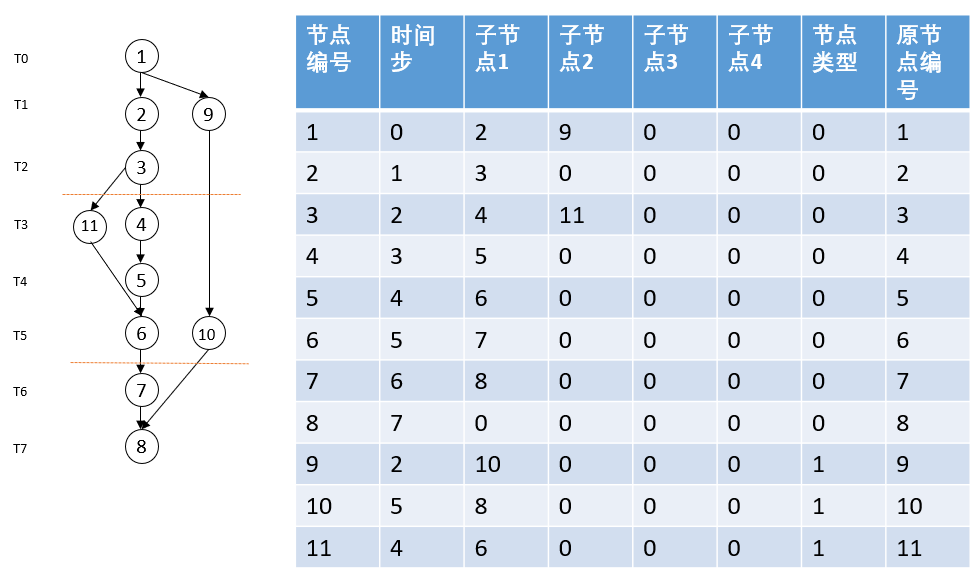
1. **输入格式：**

不同节点以换行符分割；节点内部不同字段以逗号分割



1. **输出格式：**

不同节点以换行符分割；节点内部不同字段以逗号分割



1. **整数线性编程参考**

PuLP 2.0: https://pypi.org/project/PuLP/