



(12)发明专利申请

(10)申请公布号 CN 108228966 A

(43)申请公布日 2018.06.29

(21)申请号 201711278509.2

(22)申请日 2017.12.06

(71)申请人 复旦大学

地址 200433 上海市杨浦区邯郸路220号

(72)发明人 陈更生 叶汉辰 倪思宇 黄超

(74)专利代理机构 上海正旦专利代理有限公司

31200

代理人 陆飞 陆尤

(51)Int.Cl.

G06F 17/50(2006.01)

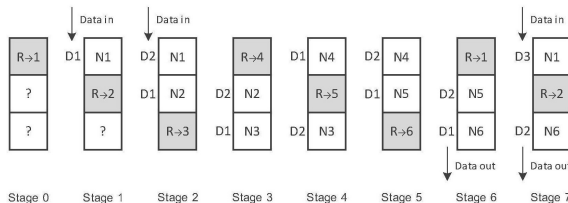
权利要求书2页 说明书6页 附图2页

(54)发明名称

基于FPGA局部动态重构技术的异型流水线设计方法

(57)摘要

本发明属于微电子技术领域,具体为一种基于FPGA局部动态重构技术的异型流水线设计方法。本发明利用FPGA局部动态重构技术,在FPGA上划分出若干个局部可重构区块,于运行时依次动态重构,通过以时间换空间的方式循环复用FPGA可编程逻辑资源,使原本无法部署在单块FPGA上的大规模应用能够完整呈现,很大程度上突破了FPGA硬件资源的限制;此外,将重构块的重构过程作为独立流水级,与用户逻辑的运算过程并行进行,显著地增加了计算的粘性。当流水线饱和时,所有的运算-重构过程以及结果输出、数据输入的过程均处在深度流水状态,达到同等资源消耗下的最优系统吞吐率与可接受的运算延时。本发明具有很强的通用性和扩展性。



CN 108228966 A

1. 一种基于FPGA局部动态重构技术的异型流水线设计方法,其特征在于,具体步骤为:

(1) 生成 n 级用户逻辑单元: N_1, N_2, \dots, N_n ,依次对应FPGA上 m 个可重构区块: $M_1, M_2, \dots, M_k, \dots, M_m$,每个可重构区块对应的用户逻辑单元数量均为 n/m ,其中可重构区块 $M_k (k \in [1, m])$ 对应的用户逻辑单元为: $N_k, N_{k+m}, \dots, N_{k+m \times i}, j \in [0, n/m-1]$;设等待输入的数据为 D_1, D_2, \dots ;

(2) 将 m 个可重构区块作为流水线上的 m 个流水级,每个流水周期依次指定1个区块作为重构块,执行动态重构操作;其余 $(m-1)$ 个区块作为运算块,执行用户逻辑的运算操作;

(3) 在第0个流水周期即上电时,区块 M_1 动态重构用户逻辑单元 N_1 ;

(4) 在第1个流水周期,区块 M_1 已重构为用户逻辑单元 N_1 ,输入数据 D_1 进行运算;同时,区块 M_2 动态重构用户逻辑单元 N_2 ;

(5) 在第 x 个流水周期, $1 \leq x < m-1$,区块 $M_{(x-y-1) \% m+1}$ 已重构为用户逻辑单元 $N_{(x-y-1) \% n+1}$,输入数据 D_{y+1} 或 D_{y+1} 的中间结果进行运算, $0 \leq y < x$;同时,区块 $M_{x \% m+1}$ 动态重构用户逻辑单元 $N_{x \% n+1}$;此时流水线未饱和;

(6) 在第 x 个流水周期, $m-1+t \cdot n \leq x \leq (t+1) \cdot n, t \geq 0$,区块 $M_{(x-y-1) \% m+1}$ 已重构为用户逻辑单元 $N_{(x-y-1) \% n+1}$,输入数据 $D_{y+1+t \cdot (m-1)}$ 或 $D_{y+1+t \cdot (m-1)}$ 的中间结果进行运算, $0 \leq y < m-1$;同时,区块 $M_{x \% m+1}$ 动态重构用户逻辑单元 $N_{x \% n+1}$;此时流水线饱和;

(7) 在第 x 个流水周期, $t \cdot n < x < m-1+t \cdot n, t \geq 1$,区块 $M_{(x-y-1) \% m+1}$ 已重构为用户逻辑单元 $N_{(x-y-1) \% n+1}$,输入数据 $D_{y+1+t \cdot (m-1)}$ 或 $D_{y+1+t \cdot (m-1)}$ 的中间结果进行运算, $0 \leq y < x \% n$;区块 $M_{(x-y-1) \% m+1}$ 已重构为用户逻辑单元 $N_{(x-y-1) \% n+1}$,输入数据 $D_{y+1+(t-1) \cdot (m-1)}$ 或 $D_{y+1+(t-1) \cdot (m-1)}$ 的中间结果进行运算, $x \% n \leq y < m-1$;同时,区块 $M_{x \% m+1}$ 动态重构用户逻辑单元 $N_{x \% n+1}$,此时流水线饱和;

数据在区块 M_1 部署为用户逻辑单元 N_1 时输入流水线,运算结果在区块 M_m 部署为用户逻辑单元 N_n 的流水周期结束时输出流水线;

这里, N 表示用户逻辑单元, n 为生成的用户逻辑单元总数, M 表示FPGA上的可重构区块, m 为可重构区块总数, D 表示等待输入的数据,依次编号为 D_1, D_1, \dots ; x 表示流水线运作过程中的某个流水周期; j, y, t 均为整数变量。

2. 根据权利要求1所述的设计方法,其特征在于,用户逻辑单元总数 n 为FPGA可重构区块总数 m 的整数倍,若不是整数倍,则取余超出部分的用户逻辑从流水线中去除,不计入流水线设计;或填充空闲用户逻辑单元以凑足整数。

3. 根据权利要求2所述的设计方法,其特征在于,可重构区块总数 m 至少为2个。

4. 根据权利要求3所述的设计方法,其特征在于,可重构区块与用户逻辑单元一一对应,在流水线重构过程中对应关系始终保持不变;为实现上述配置,在电路综合时,将所有用户逻辑单元分别综合成为不包含I/O端口的逻辑网表,将不可重构的静态电路综合成为包含I/O端口的逻辑网表;在布局布线时,保持静态电路锁定,对可重构区块载入用户逻辑单元的所有情况分别进行一次布局布线,共计进行 n/m 次布局布线;在第 $(j+1)$ 次布局布线时,可重构区块 M_k 载入用户逻辑单元 $N_{k+m \times j}, k \in [1, m], j \in [0, n/m-1]$ 。

5. 根据权利要求1所述的设计方法,其特征在于:

(1) 流水线上的 m 个流水级并非都直接参与数据处理,而是由 $(m-1)$ 个参与数据处理的运算块和1个辅助的重构块组成,因此该流水线上有效的流水级为 $(m-1)$ 个;

(2) 流水线上的 $(m-1)$ 个有效流水级并非由固定不变的数字电路构成,而是随着流水线

进行,循环地在 $(m-1)$ 个重构完成的可重构区块间切换,因此该流水线上的运算块为等效运算块;

(3)流水线上并非同时进行 n 个用户逻辑单元的运算,而是随着流水线进行,循环重构FPGA上的可重构区块,在若干连续的流水周期分时进行 n 个用户逻辑单元的运算;

(4)流水线上单个流水级的时间并非固定不变的取决于运算时间最长的用户逻辑单元,而是实时变化的,取决于该流水周期中重构块的重构时间及运算块的运算时间中的最大者;当重构区域的重构时间及各用户逻辑单元的运算时间基本保持一致时,近似得到流水级的平均耗时;

(5)当流水线饱和时,并非每个流水周期都输入一个新数据,输出一个结果,而是在 $(m-1)$ 个连续的流水周期连续输入 $(m-1)$ 个数据,称为“组”数据,经过若干流水周期的运算之后,在 $(m-1)$ 个连续的流水周期连续输出该组数据的 $(m-1)$ 个结果;每个流水周期,FPGA上始终有 $(m-1)$ 个数据处于运算状态;相邻两组数据之间的附加输入间隔称为“额外延时”。

基于FPGA局部动态重构技术的异型流水线设计方法

技术领域

[0001] 本发明属于微电子技术领域,具体涉及一种基于FPGA局部动态重构技术的异型流水线设计方法。

背景技术

[0002] 现场可编程逻辑门阵列(Field Programmable Gate Array,FPGA)是一种通用型的可编程逻辑器件,用户可以根据需要对FPGA进行灵活配置,以实现不同的电路功能。然而FPGA的片上逻辑资源(包括LUT、BRAM、DSP等可编程资源)通常是有限的,导致大规模的数字电路难以完整部署在单块FPGA板卡上。

[0003] FPGA的生产厂商(包括Xilinx和Intel等)均提供了适用于各自芯片的局部动态重构技术,即允许FPGA上一部分电路在运行时保持不变(称为静态电路),同时对FPGA上若干局部可重构区域中的一个或几个进行动态重构,从而在运行时改变电路的功能(称为可重构电路)。局部动态重构的运行过程既可以由主机进行控制,也可以由FPGA上静态电路中的数字逻辑进行控制。利用局部动态重构技术有望突破FPGA资源限制的瓶颈。

[0004] 目前关于FPGA可重构计算的研究工作有很多,但大多局限在粗粒度重构的范畴,基于局部动态重构技术的细粒度重构研究较少,尚没有在FPGA上部署大规模数字电路的高效的完整解决方案。

发明内容

[0005] 本发明的目的在于提出一种计算粘性强,对资源利用率高,性能优异,易于扩展的基于FPGA局部动态重构技术的异型流水线设计方法。

[0006] 符号约定: N 表示用户逻辑单元, n 为生成的用户逻辑单元总数,各用户逻辑单元编号为 N_1, N_2, \dots, N_n ; M 表示FPGA上的可重构区块, m 为可重构区块总数,各可重构区块编号为 M_1, M_2, \dots, M_m , k 为某个具体的可重构区块编号($k \in [1, m]$); D 表示等待输入的数据,依次编号为 D_1, D_2, \dots ; x 表示流水线运作过程中的某个流水周期; j, y, t 均为整数变量;

[0007] 本发明提出的基于FPGA局部动态重构技术的异型流水线设计方法,具体步骤为:

[0008] (1)生成 n 级用户逻辑单元: N_1, N_2, \dots, N_n ,依次对应FPGA上 m 个可重构区块: $M_1, M_2, \dots, M_k, \dots, M_m$,每个可重构区块对应的用户逻辑单元数量均为 n/m ,其中可重构区块 M_k ($k \in [1, m]$)对应的用户逻辑单元为: $N_k, N_{k+m}, \dots, N_{k+m \times j}$, $j \in [0, n/m-1]$; 设等待输入的数据为 D_1, D_2, \dots ;

[0009] (2)将 m 个可重构区块作为流水线上的 m 个流水级,每个流水周期依次指定1个区块作为重构块,执行动态重构操作;其余 $(m-1)$ 个区块作为运算块,执行用户逻辑的运算操作;

[0010] (3)在第0个流水周期(上电时),区块 M_1 动态重构用户逻辑单元 N_1 ;

[0011] (4)在第1个流水周期,区块 M_1 已重构为用户逻辑单元 N_1 ,输入数据 D_1 进行运算;同时,区块 M_2 动态重构用户逻辑单元 N_2 ;

[0012] (5)在第 x 个流水周期($1 \leq x < m-1$),区块 $M_{(x-y-1) \% m+1}$ ($0 \leq y < x$)已重构为用户逻辑单元 $N_{(x-y-1) \% m+1}$,输入数据 D_{y+1} 或 D_{y+1} 的中间结果进行运算;同时,区块 $M_{x \% m+1}$ 动态重构用户逻辑

单元 $N_{x \% n + 1}$ 。此时流水线未饱和；

[0013] (6) 在第 x 个流水周期 ($m-1+t \cdot n \leq x \leq (t+1) \cdot n, t \geq 0$)，区块 $M_{(x-y-1) \% m + 1}$ ($0 \leq y < m-1$) 已重构为用户逻辑单元 $N_{(x-y-1) \% n + 1}$ ，输入数据 $D_{y+1+t \cdot (m-1)}$ 或 $D_{y+1+t \cdot (m-1)}$ 的中间结果进行运算；同时，区块 $M_{x \% m + 1}$ 动态重构用户逻辑单元 $N_{x \% n + 1}$ 。此时流水线饱和；

[0014] (7) 在第 x 个流水周期 ($t \cdot n < x < m-1+t \cdot n, t \geq 1$)，区块 $M_{(x-y-1) \% m + 1}$ ($0 \leq y < x \% n$) 已重构为用户逻辑单元 $N_{(x-y-1) \% n + 1}$ ，输入数据 $D_{y+1+t \cdot (m-1)}$ 或 $D_{y+1+t \cdot (m-1)}$ 的中间结果进行运算；区块 $M_{(x-y-1) \% m + 1}$ ($x \% n \leq y < m-1$) 已重构为用户逻辑单元 $N_{(x-y-1) \% n + 1}$ ，输入数据 $D_{y+1+(t-1) \cdot (m-1)}$ 或 $D_{y+1+(t-1) \cdot (m-1)}$ 的中间结果进行运算；同时，区块 $M_{x \% m + 1}$ 动态重构用户逻辑单元 $N_{x \% n + 1}$ 。此时流水线饱和；

[0015] 数据总是在区块 M_1 部署为用户逻辑单元 N_1 时输入流水线，运算结果总是在区块 M_m 部署为用户逻辑单元 N_n 的流水周期结束时输出流水线。

[0016] 这里符号 $a \% b$ 是取余运算符，如 $5 \% 3 = 2$ 。

[0017] 本发明中，用户逻辑单元总数 n 为FPGA可重构区块总数 m 的整数倍，若不是整数倍，则取余超出部分的用户逻辑从流水线中去除，不计入流水线设计；或填充空闲用户逻辑单元以凑足整数。

[0018] 本发明中，可重构区块总数 m 至少为2个。

[0019] 本发明中，可重构区块与用户逻辑单元一一对应，在流水线重构过程中对应关系始终保持不变。

[0020] 为实现上述配置，在电路综合时，将所有用户逻辑单元分别综合成为不包含IO端口的逻辑网表，将不可重构的静态电路综合成为包含IO端口的逻辑网表；在布局布线时，保持静态电路锁定，对可重构区块载入用户逻辑单元的所有情况分别进行一次布局布线，共计进行 n/m 次布局布线。在第 $(j+1)$ 次布局布线时，可重构区块 M_k 载入用户逻辑单元 $N_{k+m \times j}$ ， $k \in [1, m]$ ， $j \in [0, n/m-1]$ 。

[0021] 本发明根据FPGA资源约束、流水线时序约束及本发明的异型流水线设计方法约束进行流水级划分：

[0022] I. 所有用户逻辑单元占用FPGA片上Slice、DSP、BRAM的资源总和(参考综合或布局布线之后的资源报告)应小于FPGA可用逻辑资源总数的一半，且不能占用FPGA非可重构部分资源。所有流水级的资源占用情况应基本保持一致。

[0023] II. 各流水级用户逻辑单元的运算时间及重构区块的重构时间应基本保持一致。

[0024] III. 异型流水线设计方法要求用户逻辑单元总数 n 为FPGA可重构区块总数 m 的整数倍，若不是整数倍，则取余超出部分的用户逻辑从流水线中去除，不计入流水线设计；或填充空闲用户逻辑单元以凑足整数。

[0025] 本发明根据FPGA可编程逻辑资源和IO管脚的分布及本发明的异型流水线设计方法约束进行可重构区块划分：

[0026] I. 可重构区块可用Slice、DSP、BRAM资源总数应超出对应用户逻辑单元资源占用量的10%以上。可重构区块应为不可重构静态电路保留足够的Slice、DSP、BRAM、PCIe、IOB、ICAP、BSCAN及其他资源。可重构区块的划分应满足FPGA生产厂商的所有设计约束要求。

[0027] II. 异型流水线设计方法要求可重构区域应至少为两个。

[0028] 本发明所提出的流水线的结构及运作模式有别于传统流水线，而称之为“异型”流

水线,区别至少包含以下几点:

[0029] (1) 流水线上的 m 个流水级并非都直接参与数据处理,而是由 $(m-1)$ 个参与数据处理的运算块和1个辅助的重构块组成,因此该流水线上有效的流水级为 $(m-1)$ 个;

[0030] (2) 流水线上的 $(m-1)$ 个有效流水级并非由固定不变的数字电路构成,而是随着流水线进行,循环地在 $(m-1)$ 个重构完成的可重构区块间切换,因此该流水线上的运算块为等效运算块;

[0031] (3) 流水线上并非同时进行 n 个用户逻辑单元的运算,而是随着流水线进行,循环重构FPGA上的可重构区块,在若干连续的流水周期分时进行 n 个用户逻辑单元的运算;

[0032] (4) 流水线上单个流水级的时间并非固定不变的取决于运算时间最长的用户逻辑单元,而是实时变化的,取决于该流水周期中重构块的重构时间及运算块的运算时间中的最大者;当重构区域的重构时间及各用户逻辑单元的运算时间基本保持一致时,可以近似得到流水级的平均耗时;

[0033] (5) 当流水线饱和时,并非每个流水周期都输入一个新数据,输出一个结果,而是在 $(m-1)$ 个连续的流水周期连续输入 $(m-1)$ 个数据,称为“组”数据,经过若干流水周期的运算之后,在 $(m-1)$ 个连续的流水周期连续输出该组数据的 $(m-1)$ 个结果;每个流水周期,FPGA上始终有 $(m-1)$ 个数据处于运算状态;相邻两组数据之间的附加输入间隔称为“额外延时”;

[0034] 本发明所提出的异型流水线中,除前 $(m-2)$ 个流水周期在填充流水线外,流水线始终饱和,所有的运算-重构过程以及结果输出、数据输入的过程均处在深度流水状态;设系统总运算量(n 级用户逻辑单元运算量之和)为 W ,单个流水级的平均耗时为 T ,则系统吞吐率为 $W/(\frac{n}{m-1} \cdot T)$;对于单个数据,其运算延时(从输入流水线到输出流水线)恒定为 $N \cdot T$;对于不同组的数据,异型流水线相比传统流水线,数据的运算延时还要加上额外延时,令数据的组号 $G = \lfloor D_k/(m-1) \rfloor, k \in [1, \infty)$,则额外延时为 $(n-m+1) \cdot (G-1) \cdot T$ 。

[0035] 本发明利用FPGA局部动态重构技术,在FPGA上划分出若干个局部可重构区块,于运行时依次动态重构,通过以时间换空间的方式循环复用FPGA可编程逻辑资源,使原本无法部署在单块FPGA上的大规模应用能够完整呈现,很大程度上突破了FPGA硬件资源的限制。

[0036] 本发明针对FPGA局部动态重构的细粒度重构形式进一步提出了有别于传统流水线的“异型”流水线工作模式,将重构块的重构过程作为独立流水级,与用户逻辑的运算过程并行进行,显著地增加了计算的粘性。当流水线饱和时,所有的运算-重构过程以及结果输出、数据输入的过程均处在深度流水状态,达到同等资源消耗下的最优系统吞吐率与可接受的额外延时。

[0037] 本发明适用于所有可进行流水线优化的大规模应用,并可适应应用规模的成倍扩大,具有很强的通用性和扩展性。

附图说明

[0038] 图1为异型流水线重构次序及数据流向示意图。

[0039] 图2为传统流水线与异型流水线时空图对比示意图。

[0040] 图3为异型流水线工程实现步骤示意图。

具体实施方式

[0041] 以下结合一个具体的实施例,对本发明作进一步的阐述。实施例仅用于对本发明做说明而不是对本发明的限制:

[0042] 实施例:用户逻辑单元总数为6个,FPGA片上划分的可重构区块总数为3个,即令 $n=6, m=3$ 。可以得到图1所示的流水线重构次序及数据流向图。图中共有3个可重构区块,从上到下依次为 M_1, M_2, M_3 ,对应6个用户逻辑单元: N_1, N_2, \dots, N_6 。

[0043] 根据本发明提出的基于FPGA局部动态重构技术的异型流水线设计方法,每个流水周期的操作如下:

[0044] (1) 在第0个流水周期(上电时),区块 M_1 动态重构为用户逻辑单元 N_1 ;

[0045] (2) 在第1个流水周期,区块 M_1 已重构为用户逻辑单元 N_1 ,输入数据 D_1 进行运算;同时,区块 M_2 动态重构为用户逻辑单元 N_2 。此时流水线未饱和;

[0046] (3) 在第2个流水周期,区块 M_2 已重构为用户逻辑单元 N_2 ,输入数据 D_1 的中间结果进行运算;同时,区块 M_1 输入数据 D_2 进行运算;同时,区块 M_3 动态重构为用户逻辑单元 N_3 。此时流水线饱和;

[0047] (4) 在第3个流水周期,区块 M_3 已重构为用户逻辑单元 N_3 ,输入数据 D_1 的中间结果进行运算;同时,区块 M_2 输入数据 D_2 的中间结果进行运算;同时,区块 M_1 动态重构为用户逻辑单元 N_4 。此时流水线饱和;

[0048] (5) 在第4个流水周期,区块 M_1 已重构为用户逻辑单元 N_4 ,输入数据 D_1 的中间结果进行运算;同时,区块 M_3 输入数据 D_2 的中间结果进行运算;同时,区块 M_2 动态重构为用户逻辑单元 N_5 。此时流水线饱和;

[0049] (6) 在第5个流水周期,区块 M_2 已重构为用户逻辑单元 N_5 ,输入数据 D_1 的中间结果进行运算;同时,区块 M_1 输入数据 D_2 的中间结果进行运算;同时,区块 M_3 动态重构为用户逻辑单元 N_6 。此时流水线饱和;

[0050] (7) 在第6个流水周期,区块 M_3 已重构为用户逻辑单元 N_6 ,输入数据 D_1 的中间结果进行运算,并输出数据 D_1 的最终运算结果;同时,区块 M_2 输入数据 D_2 的中间结果进行运算;同时,区块 M_1 动态重构为用户逻辑单元 N_1 。此时流水线饱和。本周期类似第0个流水周期的操作,其后第12,18,24……流水周期的操作均类同,仅参与运算的数据发生变化;

[0051] (8) 在第7个流水周期,区块 M_1 已重构为用户逻辑单元 N_1 ,输入新数据 D_3 进行运算;同时,区块 M_3 输入数据 D_2 的中间结果进行运算,并输出数据 D_2 的最终运算结果;同时,区块 M_2 动态重构为用户逻辑单元 N_2 。此时流水线饱和。本周期类似第1个流水周期的操作,其后第13,19,25……流水周期的操作均类同,仅参与运算的数据发生变化;

[0052] (9) 其后各流水周期操作类推。

[0053] 为实现上述配置的异型流水线,在FPGA工程阶段进行如下设计步骤(这里以Xilinx Vivado套件中的设计流程为例,其中综合、优化、布局布线、生成位流等流程均可以通过Tcl脚本自动执行):

[0054] (1) 设计静态电路,该部分的电路逻辑在运行时保持不变。

[0055] 静态电路中应包含控制异型流水线运行(包含运算、重构和数据输入输出)的支持

模块。同时,根据说明书中对可重构区块划分的约束,在静态电路中手工划分出可重构区块 M_1, M_2, M_3 。可重构区块 M_1, M_2, M_3 对应的逻辑单元设计为不包含实际逻辑电路的黑盒。将静态电路综合成为包含IO端口的dcp (Design Check Point) 文件;

[0056] (2) 设计动态电路,即各用户逻辑单元,该部分电路会在运行时重构到FPGA的可重构区块中。

[0057] 根据说明书中对流水级划分的约束,将应用划分为用户逻辑单元 N_1, N_2, \dots, N_6 ,其中 N_1, N_2, \dots, N_6 的顶层模块命名应与可重构区块 M_1, M_2, M_3 相对应。即 N_1, N_4 对应区块 M_1 ; N_2, N_5 对应区块 M_2 ; N_3, N_6 对应区块 M_3 。对应相同区块的用户逻辑单元与静态电路的接口应保持绝对一致。将6个用户逻辑单元分别综合成为不包含IO端口的dcp文件。

[0058] (3) 优化、布局布线及生成位流文件。

[0059] I. 打开静态电路的综合后dcp文件,读入用户逻辑单元 N_1, N_2, N_3 的综合后dcp文件,将用户逻辑单元 N_1, N_2, N_3 分别填充到可重构区块 M_1, M_2, M_3 对应的黑盒中。称此时的逻辑电路为“配置一”。对配置一执行优化、布局布线和生成位流文件流程,得到配置一的实现后dcp文件、位流文件以及用户逻辑单元 N_1, N_2, N_3 的局部位流文件。

[0060] II. 打开配置一的实现后dcp文件,将用户逻辑单元 N_1, N_2, N_3 从可重构区块 M_1, M_2, M_3 中挖去,使可重构区块 M_1, M_2, M_3 重新回到不包含实际逻辑电路的黑盒状态。将此时的逻辑电路锁定,使其中的静态电路不会在之后的优化、布局布线流程中被改变,得到静态电路[锁定的]实现后的dcp文件。

[0061] III. 打开静态电路[锁定的]实现后的dcp文件,读入用户逻辑单元 N_4, N_5, N_6 的综合后dcp文件,使用户逻辑单元 N_4, N_5, N_6 分别填充到可重构区块 M_1, M_2, M_3 对应的黑盒中。称此时的逻辑电路为“配置二”。对配置二执行优化、布局布线和生成位流文件流程,得到配置二的位流文件以及用户逻辑单元 N_4, N_5, N_6 的局部位流文件。

[0062] 执行完上述步骤后可以得到配置一、配置二的位流文件,以及用户逻辑单元 $N_1, N_2, N_3, N_4, N_5, N_6$ 的局部位流文件。实现步骤参考图3的示意。

[0063] (4) 烧录与运行。

[0064] 将配置一或配置二的位流文件烧录进FPGA芯片可以得到配置一或配置二的完整电路,通过FPGA制造商提供的局部动态重构接口,将任一用户逻辑单元 $N_1, N_2, N_3, N_4, N_5, N_6$ 的局部位流文件烧录进FPGA芯片,即可将对应的可重构区块动态重构为该用户逻辑单元的电路,完成相应运算任务。

[0065] 图2为 $n=6, m=3$ 时基于FPGA局部动态重构技术的异型流水线与传统流水线(等效 $n=6, m=6$)的对比图。在基于FPGA局部动态重构技术的异型流水线中,3个可重构区块在每个流水周期依次指定1个区块作为重构块,执行动态重构操作(在图中并未显示);其余2个区块作为运算块,执行用户逻辑的运算操作。由于运算块并非固定,而是随着流水线循环地在可重构区块间切换,因此与传统流水线的固定运算块的组织形式不同,异型流水线由2个等效运算块组成,每个子运算周期(图中橙、蓝、绿线框出的部分)完成2个输入数据(为一组)的运算,单个数据的运算延时恒为6个流水周期,系统吞吐率是传统流水线的1/3。

[0066] 从图2中可以看到,采用FPGA局部动态可重构技术以后,仅用3个区块(2个用于运算,1个用于动态重构)就可以完成原本需要6个区块的硬件资源消耗才能完成的工作量,使原本无法部署在单块FPGA板卡上的大规模应用能够完整呈现,很大程度上突破了FPGA硬件

资源的限制。然而,由于数据以组为单位输入流水线进行运算,下一组的数据需要等待上一组数据运算完成才能输入流水线,因此增加了额外的延时,如第3、4个输入数据(第2组数据)增加了4个流水周期的延时,第5、6个输入数据(第3组数据)增加了8个流水周期的延时,以此类推。即该异型流水线通过以时间换空间的方式扩展了FPGA可编程逻辑资源。

[0067] 本流水线设计方法将重构块的重构过程作为独立流水级,与用户逻辑的运算过程并行进行,显著地增加了计算的粘性。除开始阶段填充流水线的过程外,此后所有运算-重构过程以及结果输出、数据输入的过程均处在深度流水状态,达到了同等资源消耗下最优的系统吞吐率与可接受的系统延时。本发明适用于所有可进行流水线优化的大规模应用,并可适应应用规模的成倍扩大,具有很强的通用性和扩展性。

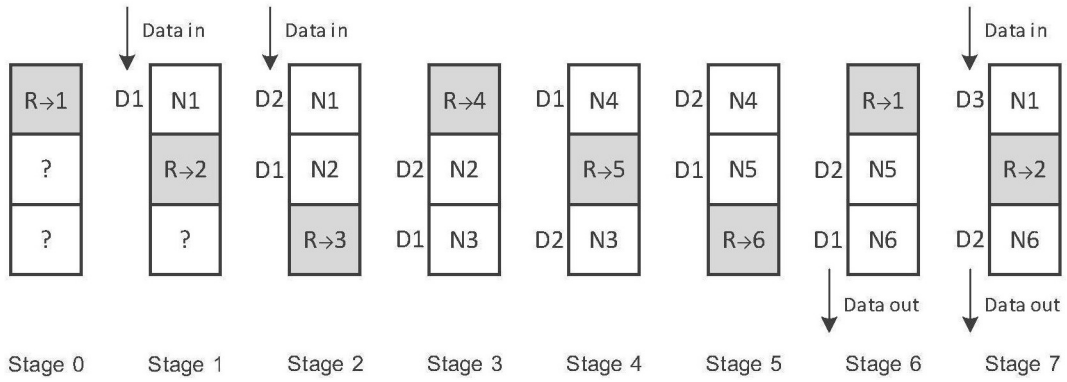
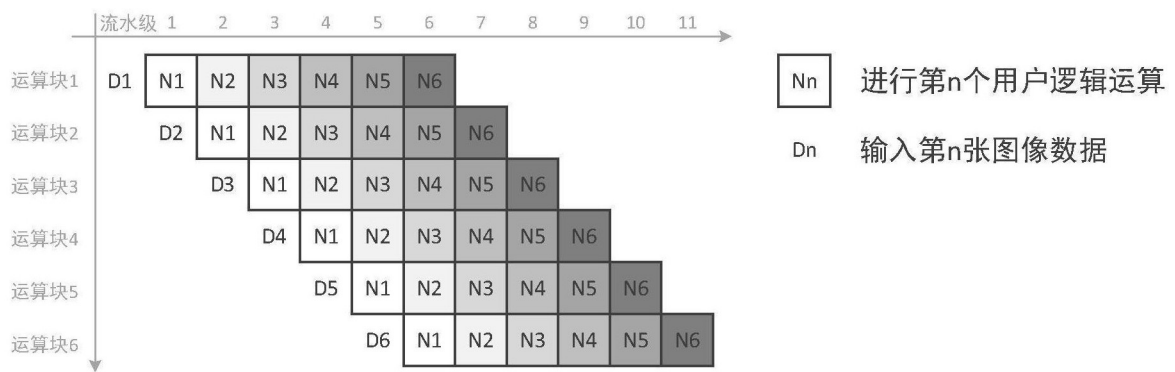
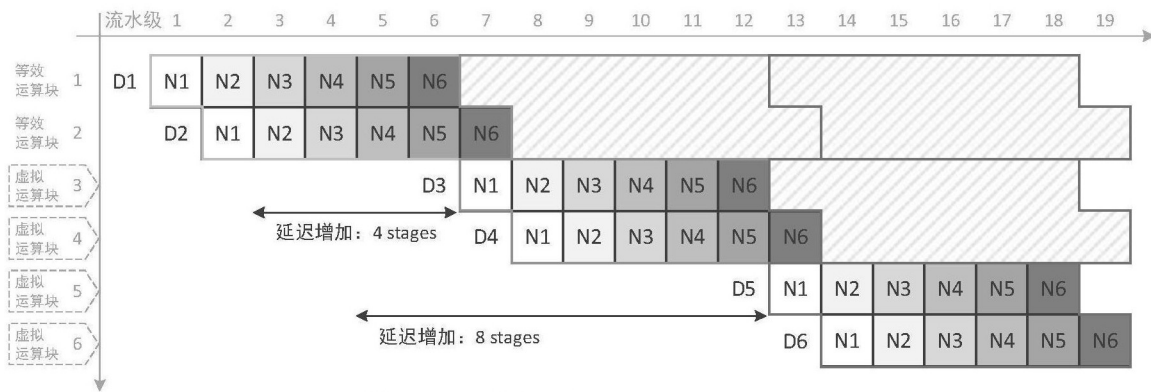


图1



(a) 传统流水线时-空图 (n=6, m=6)



(b) 基于FPGA局部动态重构技术的“异型”流水线时-空图 (n=6, m=3)

图2

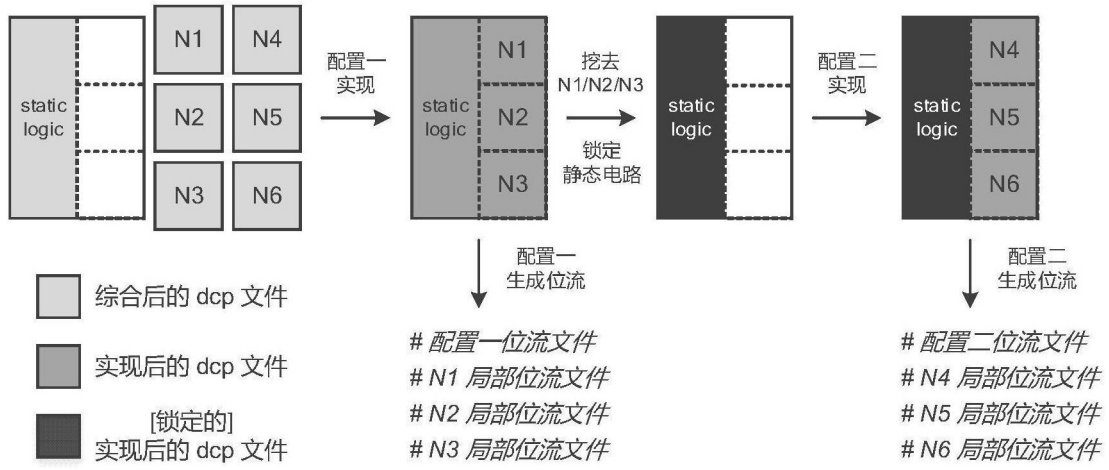


图3