

引用格式:吴皓月,邓军勇,山蕊,等. 向下全互连可重构阵列设计与实现 [J]. 微电子学与计算机,2019,36(10):59-64. [WU H Y,DENG J Y, SHAN R,et al. Design and implementation of a fully interconnected reconfigurable array[J]. Microeletronics & Computer,2019,36(10):59-64.]

向下全互连可重构阵列设计与实现

吴皓月¹,邓军勇¹,山蕊¹,贺飞龙²,刘新闻¹

(1 西安邮电大学 电子工程学院,陕西 西安 710121;2 西安邮电大学 计算机学院,陕西 西安 710121)

摘 要: 可重构电路的灵活性是电路中的绝对优势,但其电路互连资源有限、PE(Processing Element)资源利用率低. 向下全互连可重构阵列结构实现了阵列中每个 PE 与下一行 PE 的全连接,提高了 PE 资源利用率、整体数据处理速度. 本设计使用 modelsim、quartus 分别进行了功能仿真、综合下载,结果表明所设计的可重构数据流处理单元能够进行多种配置的可重构. 通过验证 FFT 算法在向下全互连的可重构阵列上的映射,结果显示可重构数据流处理单元成功实现了预期功能,并且满足其时序等要求. 综合结果显示,4 * 4 的可重构阵列,包括其外围电路,共占用 19429(58%)门逻辑资源,支持 76.07 MHz 的工作频率;单个的可重构数据流处理器占用 2565(8%)门逻辑资源,支持最高 108.96 MHz 的工作频率.

关键词: 向下全互连;动态可重构;FPGA;FFT 算法

中图分类号: TP302

文献标识码: A

文章编号: 1000—7180(2019)10—0059—06

Design and implementation of a fully interconnected reconfigurable array

WU Hao-yue¹,DENG Jun-yong¹, Shan Rui¹,HE Fei-long²,LIU Xin-chuang¹

(1 School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China;

2 School of Computing, Xi'an University of Posts and Telecommunications, Xi'an 710121, China)

Abstract: The flexibility of the reconfigurable circuit is an absolute advantage in the circuit, but its circuit interconnection resources are limited, and the utilization of PE (Processing Element) resources is low. The downward fully interconnected reconfigurable array structure realizes full connection between each PE and the next row of PEs in the array, improving PE resource utilization and overall data processing speed. This design uses modelsim and quartus to perform functional simulation and comprehensive downloading. The results show that the reconfigurable data stream processing unit can be reconfigurable in various configurations. By verifying the mapping of the FFT algorithm on the fully interconnected reconfigurable array, the results show that the reconfigurable data stream processing unit successfully implements the expected functionality and meets its timing requirements. The combined results show that 4 * 4 reconfigurable arrays, including its peripheral circuits, occupy a total of 19,429 (58%) gate logic resources, supporting a working frequency of 76.07 MHz; a single reconfigurable data stream processor occupies 2565 (8%) The gate logic resource supports a working frequency of up to 108.96 MHz.

Key words: full interconnection down; dynamic reconfigurable; FPGA; FFT algorithm

收稿日期: 2019—01—17; 修回日期: 2019—02—24

基金项目: 国家自然科学基金资助项目(61802304,61772417, 61834005,61602377,61634004);陕西省国际科技合作计划项目(2018KW-006);陕西省科技统筹创新工程项目(2016KTZDGY02-04-02);陕西省重点研发计划(2017GY-060)

1 引言

可重构计算被认为是结合传统处理器的高度灵活性的有效解决方案具有 ASIC(专用集成电路)的高处理效率^[1-2]. FPGA(现场可编程门阵列)是最多的广泛使用的可重新配置设备,可动态扩展重新配置,这是实现硬件级多任务的重要支持^[3]. 这种 FPGA 的处理区域通常被分成不同的块. 当新硬件任务到达时,必须为此硬件任务分配空闲块,而其他现有硬件任务不会受到影响. 如果硬件任务完成,则应释放此任务占用的块,并将其重新用作使用其他硬件任务的候选块^[4-5]. 可重构电路的特点是有一定的灵活性,但算法到芯片硬件的直接映射很复杂. 因而,需要研究提出具有可重构的阵列,使得其能高效的实现数据级并行计算^[6-7].

阵列的互连方式对阵列处理器的整体性能起着关键作用. 因而,出现了阵列的多种互连方式. CMesh^[8]互连结构在 Mesh 互连结构基础上增加了四个路由,每个路由直接连接四个 PE,这种结构的缺点是互连资源有限. DBIM^[9]互连方式在 Mesh 互连基础上增加了隔 PE 间的互连,这种互连方式灵活性提高了,但是整体功耗大. PDNoC^[10]互连结构在 Mesh 结构基础上增加了一个 PE 与斜方位 PE 互连,这种结构简单,但是 PE 资源利用率较低.

本文提出一种向下全互连的可重构阵列,并实现 FFT 算法在可重构阵列上映射.

2 向下全互连可重构阵列

2.1 互连方式设计

可重构处理单元经由数据流驱动可以接收来自内部寄存器周围临近四个处理单元传送来的数据,再经过相应的处理后,发送到内部寄存器的上、下、左、右四个处理单元,并且支持同时多发送,即可将输出结果同时发送给多个输出端口,由六个模块构成:左边数据选择单元、右边数据选择单元、配置单元、扇出 fanount 单元、ALU 和内部寄存器^[9].

数据流处理单元能够完成 +、-、*、\、&、>> 等基本的数据运算,属于轻核处理元,为了更好的利用处理元资源,需要较为丰富的互连资源,以实现处理元之间的高效互连. 本文互连设计如下图 1 所示:

可重构阵列处理器每个 PE 处理元不仅可以与每个 PE 相邻的上下左右 PE 互连,也能实现与相邻下一行任意 PE 互连,这样互连结构大大增加了数据处理速度,也增大了 PE 资源利用率,使得总体网

络结构的数据处理速度和 PE 间互连资源利用率大幅度提升. 同时这种互连结构在传统的 mesh 结构上增加了每个 PE 向下一层任意一个 PE 的传输路径,这种互连结构容易实现,同时不会占用过多的芯片资源,满足设计要求.

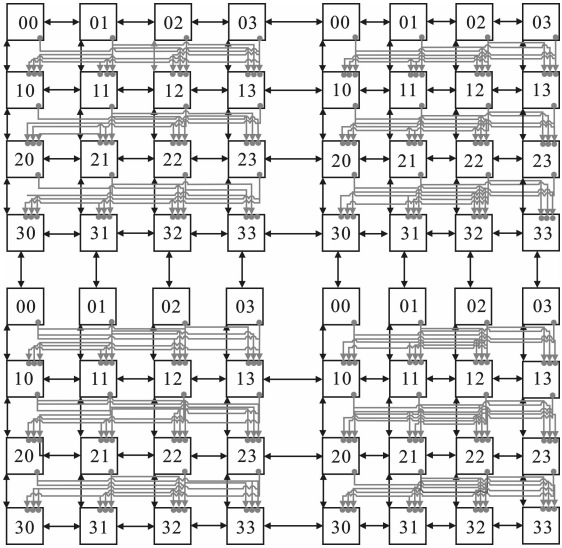


图 1 8*8 可重构阵列结构

2.2 处理单元结构设计

根据 PE 的功能,将 PE 分为以下模块:配置模块,数据选择模块、运算模块、输出模块. 由配置模块控制各模块的具体操作,其他部分分别用于选择输入数据、数据运算、数据输出,共同完成单个处理单元的运算功能. 具体结构如图 2.

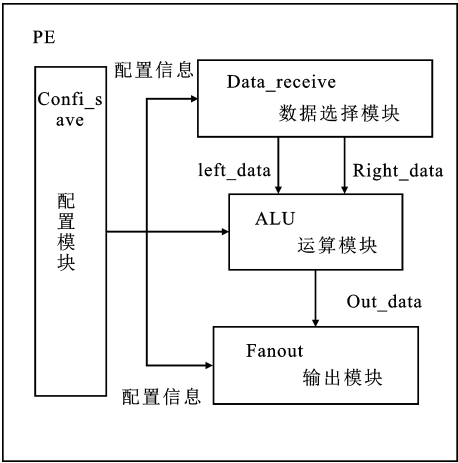


图 2 处理元内部结构

总体结构就是由配置模块产生 op、out_sel、left_sel、right_sel、imme 等配置信息;数据选择模块根据 left_sel、right_sel 选择运算数据,输出 L_O_D、R_O_D 数据至运算模块;运算模块根据 op 对 L_O_D、R_O_D、imme 进行运算,并将结果 O_O_D 输送

至输出模块;输出模块根据 out_sel,将数据输出至 R、D、N、S、W、E 等端口。

3 FFT 算法映射

3.1 FFT 算法

快速傅里叶变换 (Fast Fourier Transform, FFT), 是一种 DFT 的高效算法, 根据 DFT 的奇偶性改进获得的。离散傅里叶变换虽然在理论上阐述了如何利用数字化方法进行信号的分析, 但 DFT 的计算效率很低, 无法满足实际需求, 因此 DFT 的应用并不广泛, 直到 20 世纪信号处理的快速算法有了新的突破, 及快速傅里叶变换算法的出现^[5]。

FFT 算法可分为按时间抽取、按频率抽取两种算法, 文章采用按时间抽取的基 2 FFT 算法, 基本原理如下^[6]:

在该算法中, 要求序列 $x(n)$ 的长度为 2 幂次方, 即满足 $N = 2L$ (L 为整数), 先将序列 $x(n)$ 按 n 的奇偶分成两组, 再根据其周期性, 得到:

$$X_1\left(\frac{N}{2} + K\right) = X_1(K) \quad (1)$$

$$X_2\left(\frac{N}{2} + K\right) = X_2(K) \quad (2)$$

所以, $X(K)$ 的前半部分和后半部分分别可以表示为 ($k = 0, 1, \dots, \frac{N}{2} - 1$):

$$X(K) = X_1(K) + W_N^K X_2(K) \quad (3)$$

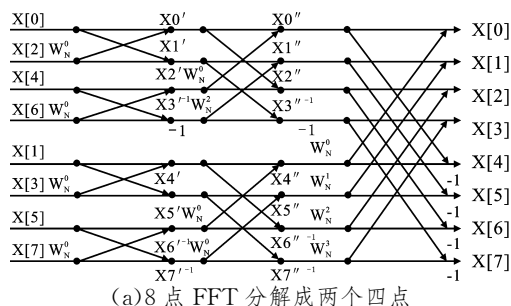
$$X(K) = X_1(K) - W_N^K X_2(K) \quad (4)$$

对于 8 点的 DFT, 可以将 8 点 DFT 分解成两个 4 点 DFT 变换而成, 如图 3 中(a); 按照这一方法一直分解, 直接得到最简的分解表达式, 如图 3 中(b)所示, 给出了最终的分解结构。这就是 DFT 的快速实现方法, 即 FFT。

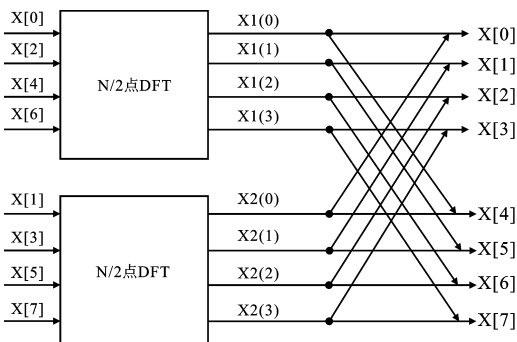
3.2 FFT 算法 8 点可重构阵列结构设计

8 点 FFT 算法的映射类似并且基于 4 点 FFT 的映射。8 点 FFT 的映射分两次进行, 第一次完成 X_0'' 、 X_1'' 、 X_2'' 、 X_3'' 、 X_4'' 、 X_5'' 、 X_6'' 、 X_7'' 的运算, 包括其实数部分和虚数部分, 其计算方式类似于 4 点 FFT 实部的映射, 不再重复其具体映射过程, 在 8×8 可重构阵列上的映射如图 4(a) 所示; 第二次则由第一次的运算结果, 完成 $X[0]$ 、 $X[1]$ 、 $X[2]$ 、 $X[3]$ 、 $X[4]$ 、 $X[5]$ 、 $X[6]$ 、 $X[7]$ 的计算, 映射如图 4(b) 所示, 图(b)中 11 代表操作运算 $\sqrt{2}/2$ 。

第一次 8×8 可重构阵列计算出 X'' 值后, 作为第二次映射的输入值, 并跟据图 4(b) 的映射方式开始计算 $X[*]$ 值。以 $X[2]$ 、 $X[6]$ 的计算为例, 说明 8 点



(a) 8 点 FFT 分解成两个四点



(b) 8 点 FFT 分解成 4 个两点

图 3 8 点 FFT 分解方案

FFT 在 8×8 可重构阵列的映射过程。将 X_2'' _R 输入到 PE 8_4 中, 经过 PE 7_4、PE 7_5 传输至 PE 6_6、PE 7_6 中, 以准备进入下一步的计算; 将 X_6'' _I 由 PE 8_5 传输到中 PE 6_6、PE 7_6 中; 通过 PE 6_6、PE 7_6 对 X_2'' _R 和 X_6'' _I 分别进行加、减运算, 并得到 $X[2]$ 的实部及 $X[6]$ 的实部。将 X_2'' _I 输入到 PE 8_6, 并经过 PE 7_7 传输至 PE 7_8、PE 8_8, 以准备进入下一步运算; X_6'' _R 经过 PE 8_7 传输至 PE 7_8、PE 8_8; 在 PE 7_8、PE 8_8 中, 分别进行 X_2'' _I 和 X_6'' _R 的减运算、加运算, 分别得到 $X[2]$ 的虚部及 $X[6]$ 的虚部。由此, 完成 $X[2]$ 、 $X[6]$ 的映射及运算。

3.3 仿真与结果分析

4×4 可重构数据流处理阵列是 8 点 FFT 算法的映射, 需要在 8×8 的可重构阵列上分两次完成, 以分别完成 8 点 FFT 的中间值 X'' 、最终数据 $X[*]$ 的计算。

首先, 设计 8 点 FFT 算法的 8 组输入数据, $x[0]$ 、 $x[1]$ 、 $x[2]$ 、 $x[3]$ 、 $x[4]$ 、 $x[5]$ 、 $x[6]$ 、 $x[7]$, 然后根据 FFT 算法公式计算中间值 X_0'' 、 X_1'' 、 X_2'' 、 X_3'' 、 X_4'' 、 X_5'' 、 X_6'' 和 X_7'' , 最后计算出 $x[*]$ 的值。8 点 FFT 仿真分为两个部分, 第一次按照图 4(a) 配置信息, 计算出中间值 X'' , 每个值得仿真数据分为实部输出、虚部输出; 第二次, 按照图 4(b) 配置数据, 计算出 8 点 FFT 最终计算结果 $x[*]$, 每个值分为实部、虚部分别输出。

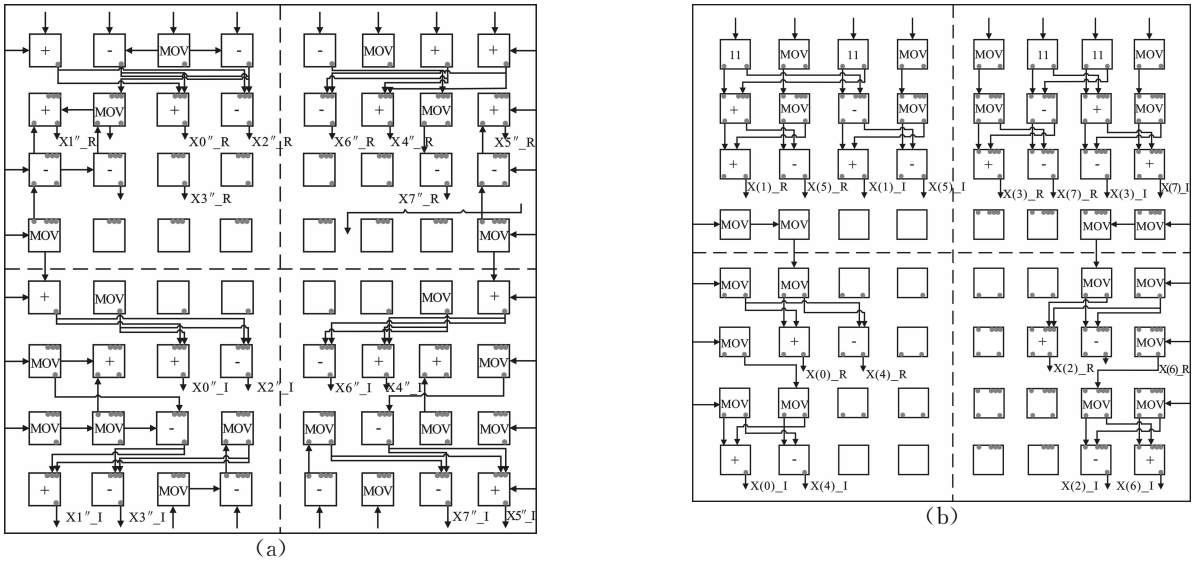
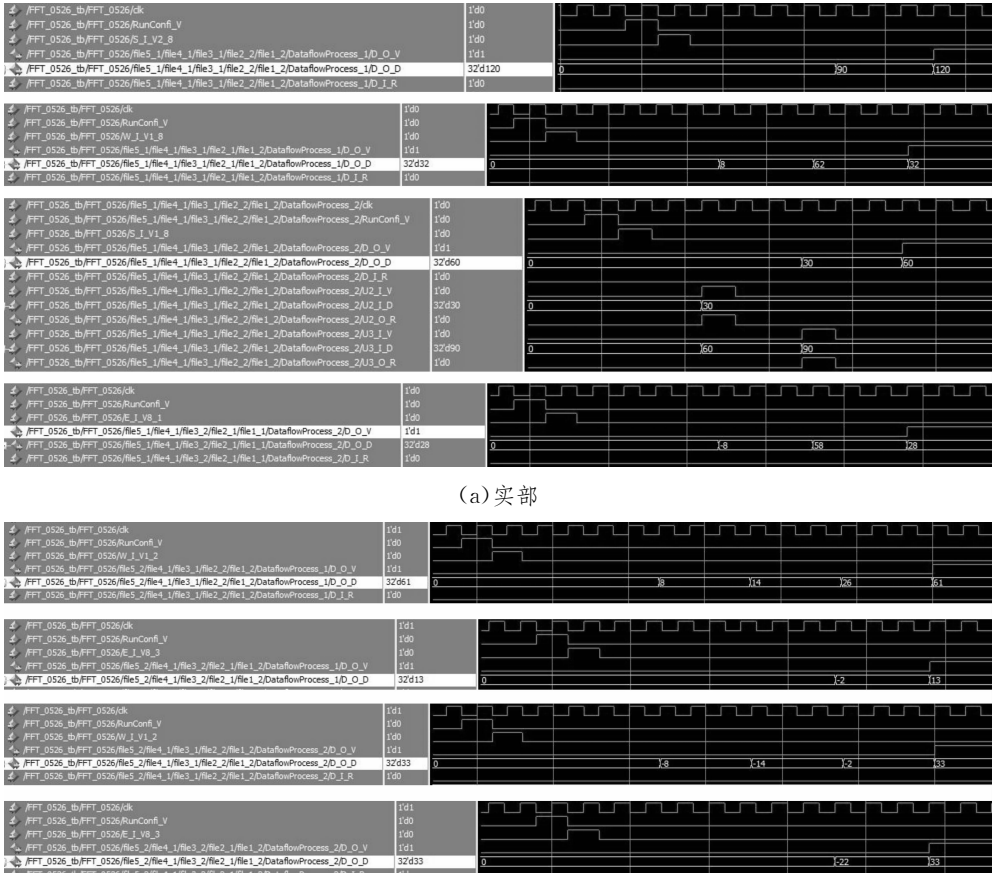


图 4 8×8 FFT 可重构阵列结构

在时序方面,8 点 FFT 计算中,中间值 X'' 的计算,需要 9 至 15 个时钟周;由 X'' 到 $X[*]$ 的计算需要 9 到 12 个时钟周期,所以,在整个 8×8 的可重构阵列上,映射的 8 点 FFT 算法,从数据输入到数据输出需要 27 个时钟周期. 以下为 $X[0]$ 、 $X[1]$ 、 $X[6]$ 、 $X[7]$ 的实部和虚部仿真结果.

下图 5 为 modelsim 仿真平台对可重构阵列处理器中 8 点 FFT 映射仿真结果,FFT 算法映射分为实部和虚部,下图 5 中(a)为部分实部 $X0''$ 、 $X1''$ 、 $X2''$ 和 $X3''$ 映射仿真结果,图 5 中(b)为部分虚部 $X0''$ 、 $X1''$ 、 $X2''$ 和 $X3''$ 映射仿真结果.



(b) 虚部

图 5 FFT 算法 8 点映射仿真结果

经过仿真验证,仿真数据正确,即阵列处理器中8点FFT的算法映射、 8×8 的可重构阵列的互连设计功能与预想设计结果一致。

4 FFT 算法 FPGA 实现

4.1 验证方案

4×4 的可重构数据流处理阵列只是运算的核心部分,为了进行其验证工作,还需要设计其外围电路,包括分频模块、状态模块、配置输入模块、数据输入、数据存储等,其结构如图6所示.由clk模块进行分频,控制时钟,并传输至每个模块;confi_in模块进行配置信息的输入,将对应的配置信息输送到每个数据流处理单元;由于4点FFT的运算需要 4×4 阵列分两次进行,所以state模块负责检测 4×4 阵列的运行状态。

图6中将状态数据传输至数据输入、数据存储模块;data_in模块,根据 4×4 阵列的运行状态,需要计算的数据输入至阵列中;data_save模块,同样需要根据stat输送的状态信息,将 4×4 阵列的计算结果存储至对应的地址中,并且data_save模块接收到的数据来判断FFT算法的映射是否正确。

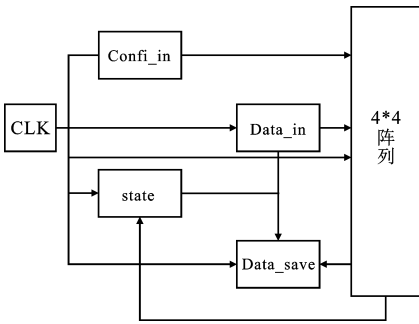


图6 外围电路结构设计

4.2 实验结果

4×4 可重构阵列的输出结果是通过读ram中的数据 and FPGA 应输出的数据进行对比,来验证功能正确性.表1中的数据 and ram中读取出来的数据120、61、32、13、33和28数据进行对比,结果相同,即本文设计的 4×4 可重构阵列及4点FFT的映射是成功的。

4.3 性能分析

4×4 可重构阵列的综合,是基于Altera DE2开发板完成的,综合所用的芯片为Cyclone II系列,电路综合面积为19429逻辑资源,利用率为58%.引脚资源占用率7%,共占用31个引脚, 4×4 的可重构阵列最高工作频率为76.07MHZ,如表2所示。

表3显示参数相对应资源占用率,表4为FFT算法FPGA实现的频率比较。

表1 FPGA 输出数据

原数据	实部 1/ 虚部 0	高 1/低 0 (16 位)	输出的 16it 数据	s 地址 描述
120	1	0	00000000,01111000	0000
	1	1	00000000,00000000	0001
61	0	0	00000000,00111101	0010
	0	1	00000000,00000000	0011
32	1	0	00000000,00100000	0100
	1	1	00000000,00000000	0101
13	0	0	00000000,00001101	0110
	0	1	00000000,00000000	0111
60	1	0	00000000,00111100	1000
	1	1	00000000,00000000	1001
33	0	0	00000000,00100001	1010
	0	1	00000000,00000000	1011
28	1	0	00000000,00011100	1100
	1	1	00000000,00000000	1101

表2 FFT 算法阵列 FPGA 综合报告

Parameter	Value
Total logic elements	19429/33216
Total combinational functions	15826/33216
Dedicated logical registers	6716/33216
Total registers	6716
Total pins	31/475
Total memory bits	1248/483840

表3 FPGA4 点 FFT 的资源评估

Parameter	Value
Area	58%
Total combinational functions	48%
Dedicated logical registers	20%
Total pins	7%
Total memory bits	<1%

表4 FFT 算法 FPGA 频率比较

Compare	Frequency(Mhz)
Radix-4[11]	44.05
R2MDC[12]	46.65
2-P R2MDC[12]	41.37
RADIX-2[13]	55.66
本文	76.07

5 结束语

本文基于FPGA的动态可重构技术,详细设计了电路结构.电路设计完成之后用Modelsim SE-64 10.2进行设计电路的功能仿真,之后利用Quartus II 13.0进行该设计的综合、布局布线等工作,并且

基于 Altera DE2 开发板 (Cyclone II 系列的 EP2C35F672C6 芯片) 下载, 进行板级验证. 单个可重构数据流处理单元的综合中, 共占用 2565 (8%) 门逻辑资源, 支持最高 108.96 MHz 的工作频率. 整个验证结果表明, 所设计的可重构数据流处理单元成功实现了预期功能, 并且满足其时序要求.

参考文献:

- [1] HSIUNG P A, LIN C S, LIAO C F. Perfecto: A systemc-based design-space exploration framework for dynamically reconfigurable architectures [J]. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 2008, 1(3): 17.
- [2] CAI X, HU W, MA T, et al. A hybrid scheduling algorithm for reconfigurable processor architecture[C]//2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, 2018: 745-749.
- [3] DALES M. Managing a reconfigurable processor in a general purpose workstation environment[C]//2003 Design, Automation and Test in Europe Conference and Exhibition. IEEE, 2003: 980-985.
- [4] WANG G, STITT G, LAM H, et al. A framework for core-level modeling and design of reconfigurable computing algorithms[C]//Proceedings of the Third International Workshop on High-Performance Reconfigurable Computing Technology and Applications. ACM, 2009: 29-38.
- [5] STITT G, VAHID F. Thread warping: a framework for dynamic synthesis of thread accelerators[C]//Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis. ACM, 2007: 93-98.
- [6] YIN S, OUYANG P, YANG J, et al. An ultra-high energy-efficient reconfigurable processor for deep neural networks with binary/ternary weights in 28nm CMOS[C]//2018 IEEE Symposium on VLSI Circuits. IEEE, 2018: 37-38.
- [7] ROUHANI B D, MIRHOSEINI A, KOUSHANFAR F. Rise: An automated framework for real-time intelligent video surveillance on fpga[J]. *ACM Transactions on Embedded Computing Systems (TECS)*, 2017, 16(5s): 158.
- [8] KIM J, BALFOUR J, DALLY W. Flattened butterfly topology for on-chip networks[C]//Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2007: 172-182.
- [9] SABBAGHI-NADOOSHAN R, PATOOGHY A. Analytical performance modeling of de Bruijn inspired mesh-based network-on-chips[J]. *Microprocessors and Microsystems*, 2015, 39(1): 27-36.
- [10] XU T C, LEPPANEN V, LOLJEBERG P, et al. PD-NOC: Partially diagonal network - on - chip for high efficiency multicore systems [J]. *Concurrency and Computation: Practice and Experience*, 2015, 27(4): 1054-1067.
- [11] MANUEL B R, KONGUVEL E, KANNAN M. An area efficient high speed optimized FFT algorithm[C]//2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN). IEEE, 2017: 1-5.
- [12] MOOKHERJEE S, DEBRUNNER L, DEBRUNNER V. A low power radix-2 FFT accelerator for FPGA [C]//2015 49th Asilomar Conference on Signals, Systems and Computers. IEEE, 2015: 447-451.
- [13] DAIL M, GIBSON R M, AMIRA A, et al. An efficient MIMO-OFDM radix-2 single-path delay feedback FFT implementation on FPGA[C]//2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). IEEE, 2015: 1-7.

作者简介:

吴皓月 女, (1995-), 硕士研究生. 研究方向为微电子学与固体电子学. E-mail: why1484132896@163.com

邓军勇 男, (1981-), 博士, 副教授. 研究方向为集成电路设计.

山蕊 女, (1986-), 博士, 副教授. 研究方向为集成电路设计.