

基于 RAM 存储阵列的并行多通道 FIFO 设计

袁亚鹏,倪 伟,郑强强,张多利,宋宇鲲
(合肥工业大学 电子科学与应用物理学院,安徽 合肥 230009)

摘 要: FPGA 中 Block RAM 的块状特性决定了具有读写位宽转换功能的异步 FIFO 容易存在存储资源浪费. 针对这个现象,本文提出一种基于存储阵列的新型并行 FIFO 结构——多通道 FIFO 共享多 RAM(Multi-channel-FIFO sharing Multiple RAM, MFMR). 与通用 FIFO IP 相比,MFMR 以仅增加少量通用 Slice 资源为代价,大幅度降低专有 Block RAM(BRAM)存储资源消耗,成倍提升存储资源空间有效利用率,最大可达通用 FIFO IP 资源利用率的 N 倍,其中 $N=\max(\alpha, 1/\alpha)$, α 是 FIFO 读位宽与写位宽的比值.

关键词: 并行 FIFO;位宽转换;存储阵列;存储浪费

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2018)12-0027-06

Parallel Multi-Channel FIFO Design Based on RAM Storage Array

YUAN Ya-peng, NI Wei, ZHENG Qiang-qiang, ZHANG Duo-li, SONG Yu-kun

(School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China)

Abstract: The feature of block RAM in FPGA determines that asynchronous FIFO with read-write bit-width conversion is easy to waste of storage resources. To solve this problem, a new parallel FIFO architecture, multi-channel FIFO sharing multiple RAM (MFMR), based on Storage array, is proposed in this paper. Compared with the common FIFO IP, MFMR reduces the consumption of proprietary Block RAM (BRAM) storage resource by a small amount of only a few general-purpose slice resources, and multiplies the efficient utilization of storage resources by up to N times the utilization of common FIFO IP resources, in which $N=\max(\alpha, 1/\alpha)$, where α is the ratio between the read-bit width and the write-bit width of the FIFO.

Key words: parallel FIFO; bit-width conversion; storage array; storage waste

1 引言

FIFO 作为基本存储部件之一,在数字集成电路设计中不可或缺. 第一, FIFO 常用于数据缓冲;第二, FIFO 能有效解决数据流跨时钟域问题,确保数据传输安全;第三, FIFO 提供位宽转换功能. FPGA 已经广泛应用于消费电子、移动终端、图像显示、医疗设备等领域,在物联网时代、人工智能浪潮中将发挥更大作用^[1]. 在基于 FPGA 的设计中,合理地使用 FIFO 有助于降低设计复杂度、增加系统稳定性,为此两大 FPGA 巨头 Xilinx 和 Altera 都建议使用 FIFO IP,并且建议使用 Block RAM

(BRAM)资源来构建大容量 FIFO^[2-3],优点是速度更快,缺点是容易导致存储资源浪费.

在 FPGA 芯片中,Block RAM(BRAM)资源块呈区域性纵向分布,以 Xilinx XC7V2000T 器件为例,总 BRAM 资源被分为 11 部分(列),平均每列含 120 个 36 kb BRAM(BRAM36),呈纵向垂直分布,其中单个 BRAM36 由两个 BRAM18 组成^[4]. BRAM18 支持有限的配置模式(深度 $\times$ 宽度): 512×36 、 $1k\times 18$ 、 $2k\times 9$ 、 $4k\times 4$ 、 $8k\times 2$ 、 $16k\times 1$ ^[4-5]. 当使用 BRAM 资源时,BRAM18 的最小配置深度决定了:若所需存储深度不足 512,即使所需宽度为 1,也将至少占用一个 BRAM18;最大配置宽度决定

了:若所需访存宽度恰好超过 36,即使深度为 1,也将至少占用两个 BRAM18. BRAM 资源的这种“整体块状”特性容易导致跨位宽转换 FIFO 的资源消耗很大而利用率很低. 在设计中,BRAM 资源利用率低下的 FIFO 的大量存在将导致两种危害:(1)增加资源消耗,增大设计面积. 由 FPGA 芯片中 BRAM 资源的区域性垂直分布特点可知,在单个布局区域内,映射属于同一 FIFO 的众多 BRAM 资源将变得困难,跨区域布局布线又将导致设计面积的膨胀.(2)降低设计频率,影响系统性能. 构成 FIFO 的大量 RAM 块给布局布线带来压力,可能导致较差时序路径,甚至成为设计瓶颈.

为研究提高资源利用率的方法,本文使用参数模型 $FIFO[x,y,s,t]$ 来描述一个写宽度 x bit,写深度 y ,读宽度 s bit,读深度 t 的 FIFO. 若记 $\alpha = s/x$ 为读 / 写位宽变换比,则该参数模型可记作 $FIFO[x,y,\alpha]$. 根据 α 值将 FIFO 分为三类:位宽一致型(Same Write-width and Same Read-width, SWSR)、位宽扩展型(Little Write-width and Big Read-width, LWBR)、位宽缩减型(Big Write-width and Little Read-width, BWLR),依次对应 $\alpha = 1, \alpha > 1, \alpha < 1$. 一般地,EDA 工具提供的 FIFO IP 要求 $\alpha = 2^n$,其中 $n \in \mathbb{Z}$.

目前业界内对提升 FIFO 的资源利用率的研究偏少,且主要面向 SWSR FIFO^[6-7],无法适用于 LWBR 或 BWLR FIFO. 本文针对 LWBR 与 BWLR FIFO,从通用 FIFO 存储组织方式出发,分析资源利用率低下的根本原因,提出一种多通道 FIFO 共享多 RAM (Multi-channel-FIFO sharing Multi-RAM, MFMR)的存储组织结构,以及适用于该结构的并行访问策略,能显著降低资源消耗,有效提升资源利用率,最大可达通用 FIFO IP 资源利用率的 N 倍,其中 $N = \max(\alpha, 1/\alpha)$.

2 优化的 FIFO 存储组织方式

2.1 通用 FIFO 存储组织方式

以 Xilinx 公司器件为例,通用 IP 定制 LWBR FIFO[32,512,4]的存储组织方式如图 1 所示,该 FIFO 由 4 个并联的配置模式为 36×512 (宽度 $\times$ 深度)的 BRAM18 组成. 其中,每个 BRAM18 占有 FIFO 有效存储总容量的 25%. 对于单片 BRAM,实际使用的地址段为 0~127,而地址段 128~511(右图灰色阴影部分)无法继续被其他 FIFO 使用,这种现象称为“地址屏蔽”. 对于地址段 0~127 内的单个

地址空间,可存储 36 位数据,实际占用 32 位,余下 4 位闲置,这种现象称为“宽度屏蔽”. 该分析也适用于 Altera 器件.

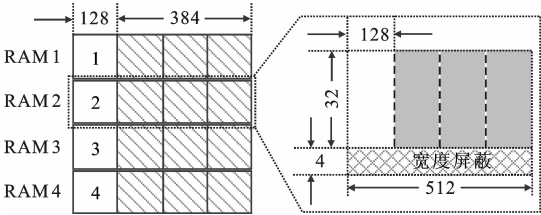


图 1 通用 FIFO 存储组织方式

图 1 所示 FIFO 具有 75% 的深度屏蔽率和 11.1% 的宽度屏蔽率,资源利用率仅为 22.2%。“深度屏蔽”和“宽度屏蔽”是 FIFO 存储资源浪费严重的根本原因. 深度屏蔽、宽度屏蔽与空间利用率之间有固定关系: $\beta = (1-d) \times (1-w)$,其中 β, d, w 分别指空间资源利用率、深度屏蔽率、宽度屏蔽率. 根据关系式可知,提高资源利用率就是要降低深度屏蔽率或降低宽度屏蔽率.

2.2 MFMR 存储组织策略

为解决资源利用率低下的问题,本文提出一种基于存储阵列的存储组织策略,如图 2 所示,即将图 1 中被屏蔽的地址段充当其他 FIFO 的存储空间,下面以剩余地址段三等分为例来介绍.

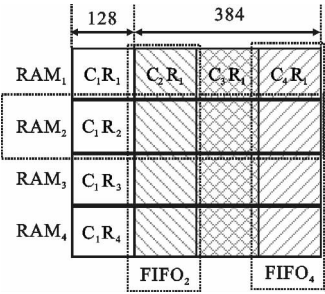


图 2 MFMR 存储组织策略

图 2 中由 4 片 BRAM18 构成的存储阵列被划分为 16 个存储块(段),构成一个存储表格,其中块 $C_xR_y (1 \leq x \leq 4, 1 \leq y \leq 4)$ 属于 $FIFO_x$ 和 RAM_y . 此时 $d=0, w=11.1\%, \beta=88.8\%$,资源有效利用率较通用 FIFO IP 提升 3 倍.

2.3 并行访存策略

基于 MFMR 存储组织策略,本文提出一种并行访存策略. 根据图 2 中数据与存储空间的对对应关系,将整个并行访问过程分为 4 个状态步骤($S_1 \sim S_4$)循环进行,如图 3 所示.

约定:“活动块”指存储表格中的某个存储块在

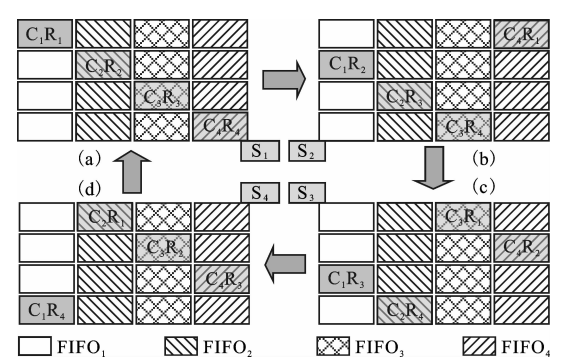


图 3 四路并行访存策略实例

当前时刻允许被 FIFO 访问(深色标记),坐标 C_xR_y ($1 \leq x \leq 4, 1 \leq y \leq 4$) 指该活动块来自 RAM_y 的第 x 段,允许被 $FIFO_x$ 访问. 各状态步骤包含的活动块如表 1 所示.

表 1 四路并行访存状态编码

状态	编码	活动块
S_1	0001	$C_1R_1/C_2R_2/C_3R_3/C_4R_4$
S_2	0010	$C_1R_2/C_2R_3/C_3R_4/C_4R_1$
S_3	0100	$C_1R_3/C_2R_4/C_3R_1/C_4R_2$
S_4	1000	$C_1R_4/C_2R_1/C_3R_2/C_4R_3$

记对于 $C_{i1}R_{j1}$ 与 $C_{i2}R_{j2}$, 若 $i_1 \neq i_2$ 与 $j_1 \neq j_2$ 同时满足, 则称 $C_{i1}R_{j1}$ 与 $C_{i2}R_{j2}$ “不相关”, 否则即为“相关”; 记 $\psi(s)$ 为状态 s 拥有的活动块集合, 其中 $s \in \{S_1, S_2, S_3, S_4\}$, 如 $\psi(S_1) = \{C_1R_1, C_2R_2, C_3R_3, C_4R_4\}$; 若集合 $\psi(s)$ 中所有元素之间均不相关, 则称集合 $\psi(s)$ 为“不相关集合”, 否则即为“相关集合”. 基于图 3, 并行访存策略可概括为两条基本原则:

原则 1 $\forall s \in \{S_1, S_2, S_3, S_4\}, \psi(s)$ 为不相关集合;

原则 2 $\psi(S_1), \psi(S_2), \psi(S_3), \psi(S_4)$ 交集为空.

原则 1 是并行访问的充要条件, 禁止由于多个访问请求同时到达同一个存储体从而造成的访问冲突. 原则 2 是并行访问持续进行的充要条件, 确保对于任意一个 FIFO 的外部数据流能够写到各自正确的存储位置, 避免数据写污染.

3 MFMR 方案架构

基于 MFMR 存储组织策略与并行访存策略, 本文提出了多通道 FIFO 架构(MFMR 架构), 该方案以多片 BRAM 组成的存储阵列为中心, 结合外围控制逻辑, 实现多个 LWBR 或 BWLR 类型 FIFO 的绑定封装. 下面以 LWBR FIFO 为例做详细说明.

3.1 LWBR MFMR 基本架构

图 4 所示的是 4 片 RAM 拼接 4 个 LWBR FIFO[32,512,4]的架构. 图中数字 $x=(1,2,3,4)$ 对应指该对象属于 $FIFO_x$.

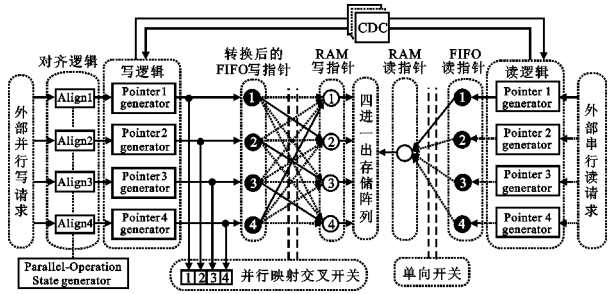


图 4 LWBR MFMR 架构方案示意

图 4 所示架构对外结构表现为, 包括四路输入端口(写端口 1~4), 四路输出端口(读端口 1~4), 读写端口允许工作在不同时钟域. 四路写端口之间相互独立, 支持任意并行持续写操作, 即四路写端口允许同时发起持续写请求(图中四路 FIFO 都在写访问). 四路读端口之间相互对立, 仅支持串行读访问, 即禁止多个读端口同时发起读请求(图中仅有 FIFO1 正在读访问). 四路读端口可选择复用读数据输出端口, 也可在输出端通过插入输出数据保持寄存器的方式为每路读端口生成读数据输出端口, 其中写、读数据位宽分别为 32、128.

图 4 所示架构对内结构表现为, 包括存储阵列、状态发生器、写对齐逻辑、读/写逻辑、读/写指针映射控制逻辑、跨时钟域(Clock Domain Cross, CDC)逻辑. 存储阵列是一个符合 MFMR 存储组织策略规范的“四入一出”存储表格, 允许在同一时刻写入 4 个写数据、读出 1 个读数据. 状态发生器为并行访存操作提供时序基准, 为对齐模块提供状态参考. 写对齐逻辑基于时序基准对外部数据流做时序调整, 为并行写访问提供安全保障, 避免访问冲突. 写指针映射控制逻辑负责 FIFO 写指针到 RAM 写指针的并行映射, 以及并行写数据流的重定向. 读指针映射控制逻辑是一个多路选择器, 负责 FIFO 读指针到 RAM 读指针的串行映射. CDC 逻辑保证地址指针跨时钟域的安全同步.

3.2 访存请求对齐

记“接入状态”为某个 FIFO 下次允许写访问的状态, 如初始化后 $FIFO_1 \sim FIFO_4$ 的接入状态依次为 S_1, S_4, S_3, S_2 . 在系统初始化后, 如果 FIFO1 的首次外部写请求在 S_3 时刻到来, 则将访问失败, 因为 C_1R_1 不是 S_3 状态的活动块, 不满足并行访存原

则. 为此, 本文提出访存请求对齐策略: 根据当前存储阵列活动块的状态信息, 判断是否阻塞请求序列, 若时序不满足对齐策略, 在请求序列前插入空白周期, 控制阻塞时长, 使得到达存储阵列时的请求序列符合并行访存策略的时序要求.

如图 5 所示, 访存请求对齐逻辑的核心是一个由多个缓冲单元构成的序列阻塞器, 主要包含: 时钟端、阻塞开关、序列输入端、地址输入端、序列输出端. 输出端与内部标号为地址输入端口值的缓冲单元保持连通. 对齐过程如下:

- (1) 外部访问序列 (数据流) 到来时被直接送入序列阻塞器, 在缓冲单元间向右流动, 同时增加地址线值, 输出端口 Q 动态跟踪序列头部;
- (2) 同步监听状态监测器, 若检测到“接入状态”到来, 则固定地址端口值 (此时输出序列 Q 满足并行访存策略要求), 输出对齐后的请求序列;
- (3) 更新“接入状态”, 准备下次序列对齐.

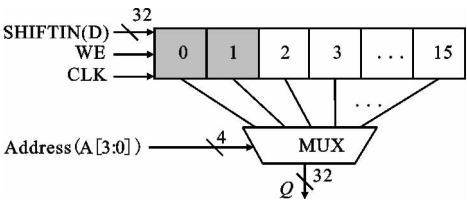
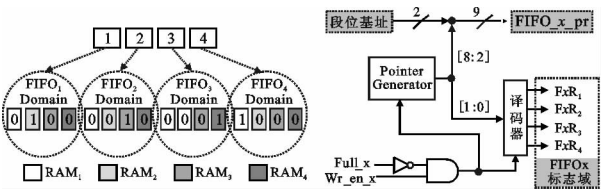


图 5 访存请求对齐结构

3.3 地址空间映射规则

在图 4 中, 写指针映射控制逻辑负责将 FIFO 指针空间映射到 RAM 指针空间, 以及并行数据流重定向. 由于单个 FIFO 的数据流与地址流保持同步, 因此写指针映射逻辑的核心就是地址控制. MFMR 存储组织策略决定了 MFMR FIFO 空间与 RAM 空间呈“存储交叉”关系, 因此, 并行 FIFO 的指针空间到多个 RAM 的指针空间的映射也存在“映射交叉”, 这种交叉关系通过并行映射交叉开关控制, 如图 6(a) 所示.



(a) 并行映射交叉开关 (b) 并行写指针转换

图 6 并行映射交叉开关及并行写指针转换

并行映射交叉开关由多位标志位组成, 位数由 MFMR 架构中的 FIFO 个数与 RAM 个数的乘积决定, 如图 4 架构中为 16 位. 标志位在 FIFO 域内

连续; 在 RAM 域内不连续. 标志位高有效, 表示所在域 FIFO 指针映射到所在域的 RAM 指针空间. 图 6 中的标志位的值指出了图 4 架构中的映射状态, 即 $FIFO_1 \sim FIFO_4$ 指针同时分别映射到 $RAM_2, RAM_3, RAM_4, RAM_1$ 指针空间. 地址映射要求遵从图 6(b) 给出的地址空间映射规则, 其中 $FxR_1 \sim FxR_4$ 组成图 6(a) 中的 $FIFO_x$ 映射标志域.

3.4 并行满标志的产生

并行访存在结构上通过数据流对齐逻辑保证写访问符合并行访存策略原则, 这将导致外部写访问 (对齐前) 与内部写访问 (对齐后) 存在若干周期的相位差, 使得满标志判断不及时, 存在“FIFO 已满但外部仍在发起写请求”的可能. 为避免写溢出, 建议使用可编程满标志, 即: 当存储空间距离真正满还有部分空闲空间时, 拉高满标志.

3.5 BWLR MFMR 基本架构

BWLR MFMR 架构如图 7 所示, 由于读数据流和读地址流的时序关系严格对应, 读一侧没有显示出读数据的流动走向. 与图 4 中结构相比, BWLR FIFO 的主要区别在于二者在并行访存侧, 对齐逻辑与并行指针生成逻辑的前后关系是相反的, 即图 4 的对齐逻辑在写逻辑前, 图 7 的对齐逻辑在读逻辑之后. 而且图 4 中对齐逻辑的关注对象是待写数据流; 而图 7 中的对齐逻辑的关注对象是读地址流. 这主要是处于“读安全”的考虑. 由于图 4 结构带来的对齐延时将导致“空标志”产生不及时, 容易引发“空后读”, 而图 7 结构能够立即生成“空标志”, 规避“空后读”, 确保并行读操作的安全可靠.

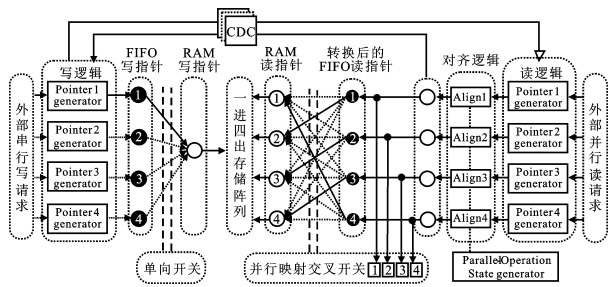


图 7 BWLR MFMR 架构示意图

4 实验分析

本文中设计将多通道 FIFO 封装成 IP 且支持读写位宽参数化. 为了与通用 FIFO IP 做资源对比, 以 xilinx xc7v2000 系列芯片为平台采集 MFMR 设计的总资源消耗数据, 并对数据做归一化处理, 即单个 FIFO 的平均资源消耗. 在不同 α 情况下, 归一

化后的 MFMR 的资源消耗量与写宽度(x)之间的关系如图 8 所示(只显示部分采样点的数据),对应的 BRAM 资源消耗情况如表 2 所示.

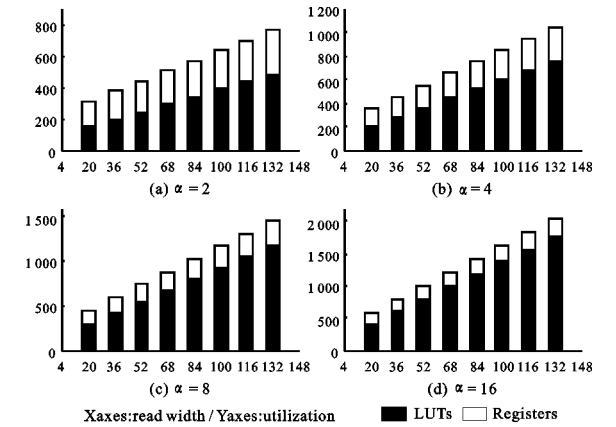


图 8 归一化后 MFMR 的 Slice 资源消耗

表 2 归一化后 MFMR 的 BRAM 资源消耗

写宽度	BRAM18 块数
20~36	1
37~72	2
73~108	3
109~144	4

本文采用最小二乘法对典型参数下的资源消耗增长趋势进行拟合. LUTs 资源消耗量与 x 呈线性增长关系,增长速度由 α 决定,如图 9 所示;Registers 资源消耗与 x 也呈线性增长关系,但增长速度与 α 无关,如图 10 所示. 据此,MFMR 消耗的资源可预测.

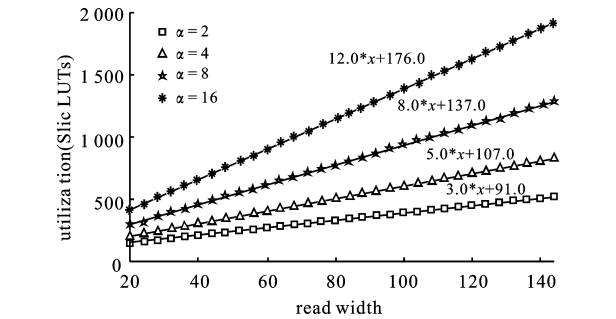


图 9 MFMR 的 LUTs 资源增长拟合

考虑如下应用场景:片上网络中的 8 个运算核心同时写访问片外缓存,其中结果数据有效位宽为 64 bit,DDR4 IP 用户接口数据位宽为 512 bit. 由于运算核心与用户接口位于不同时钟域,因此 8 路数据流需要通过 FIFO[64,512,8]完成跨时钟域的数据位宽转换. 表 3 给出了分别采用 FIFO IP 方案与

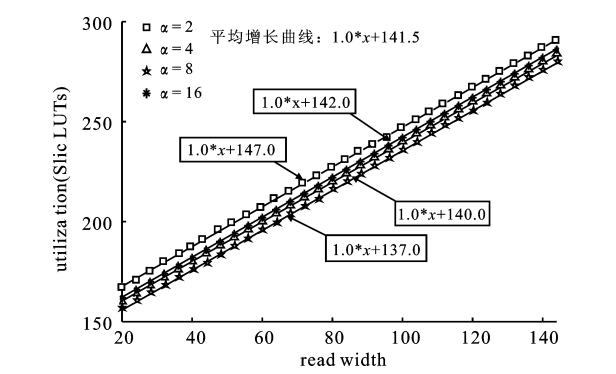


图 10 MFMR 的 Registers 资源增长拟合

MFMR 方案实现上述 FIFO 的资源消耗情况.

表 3 两种方案的资源消耗量

	LUT	Register	BRAM18
IP	416	952	120
MFMR	5 240	1 644	16
Change ⁽¹⁾	+4 824	+692	-104

注 1:Change 为 MFMR 消耗量与 IP 消耗量的差.

在 FPGA 中通用 slice 资源非常丰富,BRAM 资源相对稀缺,使用少量 slice 资源换取大量 BRAM 资源是合理的,表 4 给出了 Xilinx 7 系列不同等级典型芯片在实现 MFMR 方案时的资源增量的占比情况^[8].

表 4 MFMR 资源增量占比(xilinx 7 series)

Device	LUT _{s(1)}	FF _{s(1)}	BRAM ₁₈₍₂₎
XC7A35T	23.19%	1.66%	104.00% ⁽³⁾
XC7A50T	14.8%	1.06%	69.33%
XC7A200T	3.58%	0.26%	14.25%
XC7K70T	11.77%	0.84%	38.52%
XC7K480T	1.62%	0.12%	5.45%
XC7VX330T	2.36%	0.17%	6.93%
XC7V2000T	0.39%	0.03%	4.02%

注 1:相比 IP 方案,MFMR 增加的资源量;注 2:相比 IP 方案,MFMR 节省的 BRAM18 块数;注 3:此时 IP 方案无法实现,但 MFMR 可以实现.

5 结束语

本文设计的多通道 FIFO 共享多 RAM 的并行 FIFO 解决方案,能够以少量 Slice LUTs 与 Slice Registers 等通用资源换取大量 Block RAM 专有资源,从根本上解决了传统 FIFO IP 的存储资源空间利用率低下的问题. 结合 MFMR 的特点,可以应用于具有“多路数据流采集汇总”或“数据分流”特点的场景,如 FPGA 相关设计中并发访问片外缓存.

参考文献：

[1]

王伟. 物联网时代的 FPGA 发展之路[J]. 电子技术应用, 2017, 43(2):9.

[2]

Xilinx Inc. 7 Series FPGAs memory resources user guide. UG473 (v1.5):13-14[M/OL]. https://www.xilinx.com/support/documentation/suer_guides/ug473_7Series_Memory_Resources.pdf.

[3]

Altera Inc. SCFIFO and DCFIFO IP cores user guide. UG-MFNALT_FIFO[M/OL]. [2017-11-06]. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_fifo.pdf.

[4]

Xilinx Inc. 7 Series FPGAs configurable logic block user guide. UG474 (v1.3):10-12[M/OL]. https://www.xilinx.com/support/documentation/suer_guides/ug474_7Series_CLB.pdf.

[5]

Altera Inc. Embedded memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide. UG-01068[M/OL]. [2017-11-06]. https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug_ram_rom.pdf.

[6]

刘炎华, 孙玲, 赖宗声. 一种用于片上网络的异步共享 FIFO 设计[J]. 微电子学与计算机, 2013, 30(7):159-162.

[7]

Gong Z, Chen T, Zou F, et al. Implementation of multi-channel FIFO in one blockRAM with parallel access to one port[J]. Journal of Computers, 2014, 9(5):1193-1200.

[8]

Xilinx Inc. 7 Series FPGAs Overview. DS180 (v2.5):2-5[M/OL]. https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf.

作者简介：

袁亚鹏 男,(1992-),硕士研究生. 研究方向为集成电路设计与测试. E-mail:jsangypeng@qq.com.

倪伟 男,(1977-),博士,副教授,硕士生导师. 研究方向为大规模数字集成电路设计、可重构计算和多核与片上网络等.

郑强强 男,(1991-),硕士研究生. 研究方向为 SoC 设计.

张多利 男,(1976-),博士,研究员,硕士生导师. 研究方向为多核处理器体系结构与设计方法、多媒体专用集成电路设计实现等.

宋宇鲲 男,(1975-),博士,副研究员. 研究方向为面向数据高密和计算密集应用的 SoC/MPSO 体系结构与实现.