

RTL 综合中 FPGA 片上 RAM 工艺映射

李 艳¹, 张东晓², 于 芳¹

(1. 中国科学院微电子研究所, 北京 100029; 2. 北京飘石科技有限公司, 北京 100080)

摘 要: RAM(Random-Access-Memory, 随机存储器)是 FPGA(Field Programmable Gate Arrays)片上最重要的宏单元之一, RTL(Register-Transfer-Level)综合对 FPGA 开发中 RAM 的有效利用起至关重要作用. 本文针对 RTL 综合中 RAM 源描述和目标结构多样化带来的技术难题, 提出了一种 RAM 工艺映射方法, 即建立工艺无关的 RAM 统一模型, 在模型基础上通过建模、模式匹配、造价计算、绑定四步实现. 该方法应用于 RTL 综合, 可以将多种 RAM 源描述有效地映射到最佳类型和数量的 FPGA 片上 RAM 资源. 实验数据表明采用该方法实现的 RAM 工艺映射效果和主流 FPGA 综合工具——Synplify 和 XST 相当, 该模块已经集成在自主开发的 RTL 综合工具——Hqsyn 中并实现商用.

关键词: 现场可编程门阵列; 寄存器传输级综合; 片上随即存储器; 工艺映射

中图分类号: TN47

文献标识码: A

文章编号: 0372-2112 (2016)11-2660-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2016.11.014

Technology Mapping of FPGA on-Chip-RAM in RTL Synthesis

LI Yan¹, ZHANG Dong-xiao², YU Fang¹

(1. Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China;

2. Uptops Design Technologies, Inc., Beijing 100080, China)

Abstract: RAM is one of the most important macro-cells of FPGA, and RTL synthesis plays a critical role on the effective use of RAM resources in FPGA development. For the difficulty of multi-resources and multi-targets in RAM technology mapping of RTL synthesis, this paper presents a method of technology mapping for FPGA on-chip RAM. In this method, an unified technology-independent RAM model is proposed, and based on this model, RAM technology mapping is performed through a series of steps, including model set-up, mode-matching, cost calculation, and binding. When applied in RTL synthesis, this method is capable of mapping various styles of RAM RTL descriptions into the most appropriate type and number of FPGA on-chip RAM resources. Experimental result shows that this method achieves comparable RAM mapping results as the mainstream FPGA RTL synthesis tools-Synplify and XST, this technology has been integrated into the self-developed RTL synthesis-Hqsyn and has been applied into the FPGA market.

Key words: FPGA(Field Programmable Gate Arrays); RTL(Register-Transfer-Level) synthesis; on-chip RAM; technology mapping

1 引言

FPGA 广泛地应用在各种电子系统中, 几乎所有系统均涉及大数据量处理和存储, 当前主流 FPGA 厂商如 Altera、Xilinx、Lattice 等公司的 FPGA 芯片都内嵌片上 RAM 资源, 片上 RAM 已经成为大规模 FPGA 至关重要的结构^[1]. FPGA 设计流程包括 RTL 综合、布局布线、码流产生, 其中 RTL 综合作为设计流程中的第一步, 对整个电路实现的性能起着至关重要的作用, 特别在决定 RAM 资源的有效利用方面起关键作用.

至今, 对于 RAM 存储器的综合研究, 在学术界, 研究主要分为三个分支: 第一分支主要研究存储器的高级综合^[2,3], 着眼于高级综合中的多层次存储器架构生成, 主要面向 ASIC(Application Specific Integrated Circuit), 探讨生成何种 RAM 描述, 而没有涉及具体实现; 第二分支主要研究 RTL 级存储器翻译转化成逻辑存储器^[4-8], 没有工艺映射的研究; 第三分支主要研究逻辑存储器映射到基于目标工艺库的存储器, 即存储器的工艺映射^[9-13], 本文的工作属于该分支. 其中^[9-12]研究不同面积规模的逻辑存储器到目标存储器的映射, 不

考虑读写类型等工作模式,^[13]研究的存储器工艺映射涉及不同种类、不同面积规模的目标存储器的映射,但种类只包含端口有无读写功能,实用性不强.本文提出的存储器工艺映射主要面向实用数字电路中的 RAM 存储器,涵盖各种类型和各种面积规模的存储器,并且重点在类型上做系统研究,类型主要包括:(1)分布式 RAM^[14] (Distributed-RAM, DRAM) 和块式 RAM^[14] (Block-RAM, BRAM); (2)单端口^[14] (Single-Port) RAM 和双端口^[14] (Dual-Port) RAM; (3) RAM 端口的读写操作,即只有读、只有写、有读有写; (4)端口工作模式,即读模式、写模式、读写模式; (5)端口管脚类型; (6)RAM 存储规模等.在工业界,FPGA 片上 RAM 的工艺映射已广泛集成于 FPGA RTL 综合工具中,这类技术被国外 Synopsys、Xilinx 等少数几家国外公司所垄断,用户只能获得 RAM RTL 级描述的用户手册信息^[15,16] 具体映射实现的技术实现细节不公开.

本文在调研学术理论、RAM 各种类型特点和多家 FPGA 综合工具的基础上,针对 RAM 工艺映射中多源

和多目标带来的技术难题,提出基于 Verilog HDL 语言描述的 RAM 工艺映射技术和实现方案——Rsyn,采用 C++ 语言实现,最终实现多种 RAM 源描述有效地映射到最佳类型和数量的 FPGA 片上 RAM 资源,产生 EDIF (Electronic Design Interchange Format) 格式的电路网表,其正确性在 Xilinx 的多个系列 FPGA 上得到充分验证, Rsyn 已经集成在自主开发的 RTL 综合工具——Hqsyn, 并实现商用。

2 技术难点

RAM 工艺映射的技术难点主要在于 RAM 源描述的多样性、FPGA 片上 RAM 的多模式以及由此造成的多源 RAM 匹配对多目标 RAM 映射困难.

FPGA 片上 RAM (下文 RAM 均指 FPGA 片上 RAM), 主要分为两类: DRAM 和 BRAM, 图 1 所示为 Xilinx Virtex2 FPGA 器件上 RAM 资源的分布情况^[14], DRAM 分布在逻辑单元中, 用逻辑单元内部结构实现, BRAM 是 FPGA 片上专有单元。

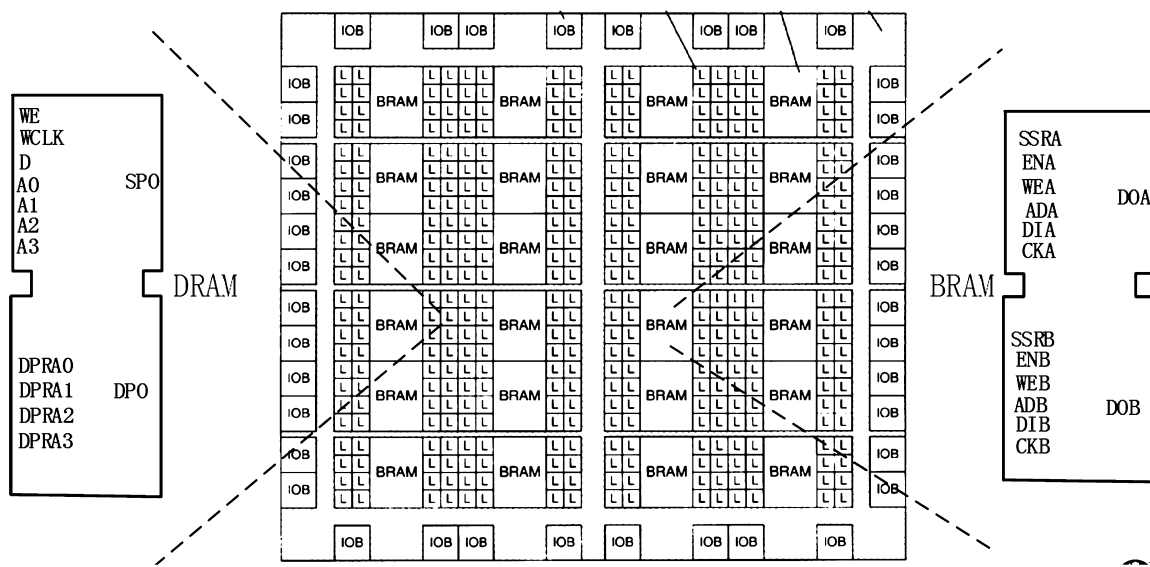


图1 Xilinx的RAM结构位置图和结构图

DRAM 和 BRAM 分别具有不同的特性,包括端口数量、端口有无读写操作、端口工作模式,即读模式、写模式、写读 (read-during-write) 模式,管脚类型 (如图 1 左图中 RAM 的 WE、WCLK、D、A0 等),以及存储规模等,每一种 RAM 都是可以由这几种特征描述。其中最为复杂的是写读模式,用于控制 RAM 处于写入状态时读端口数据与存储单元数据以及写数据之间的关系,主要分为先读 (read-first)、先写 (write-first)、保持 (no-change) 三种模式^[15]。

表 1 两个端口 RAM 读写操作模式表

		端口 A				端口 B			
		无	只读	只写	读写	无	只读	只写	读写
端口 A	无	-	-	-	-	-	✓	-	✓
	只读	-	-	-	-	✓	✓	✓	✓
	只写	-	-	-	-	-	✓	-	-
	读写	-	-	-	✓	✓	✓	-	-
端口 B	无	-	✓	-	✓	-	-	-	-
	只读	✓	✓	✓	✓	-	-	-	-
	只写	-	✓	-	-	-	-	-	-
	读写	✓	✓	-	-	-	-	-	✓

表1是针对端口数为2、只考虑端口读写操作的可能RAM类型,表中显示共有20种RAM结构.如果考虑工作模式、同步异步、不同管脚类型、存储规模,以及交叉端口的写读模式^[17]等情况,则可能的结构类型更多,实现难度会更大.上文介绍目标RAM的多样性和复杂性,同样RAM源描述方面,一种功能结构的RAM可以有多种不同的描述方式,以图2为例,图中上部分是用Verilog HDL语言描述的三种不同RAM源描述,它们功

能等价,均描述了读写模式为“先写”的RAM.

实现上述“先写”RAM源描述的RAM结构也有多种选择,图中下部分的三种结构是针对功能等价的“先写”RAM的三种不同实现方式,而图例中一个“先写”模式的RAM源描述到目标结构的实现就有9种不同的映射匹配目标,如果考虑端口读写操作、同步异步、不同管脚类型、存储规模,以及交叉端口的写读模式,匹配的目标RAM会有更多种结构.

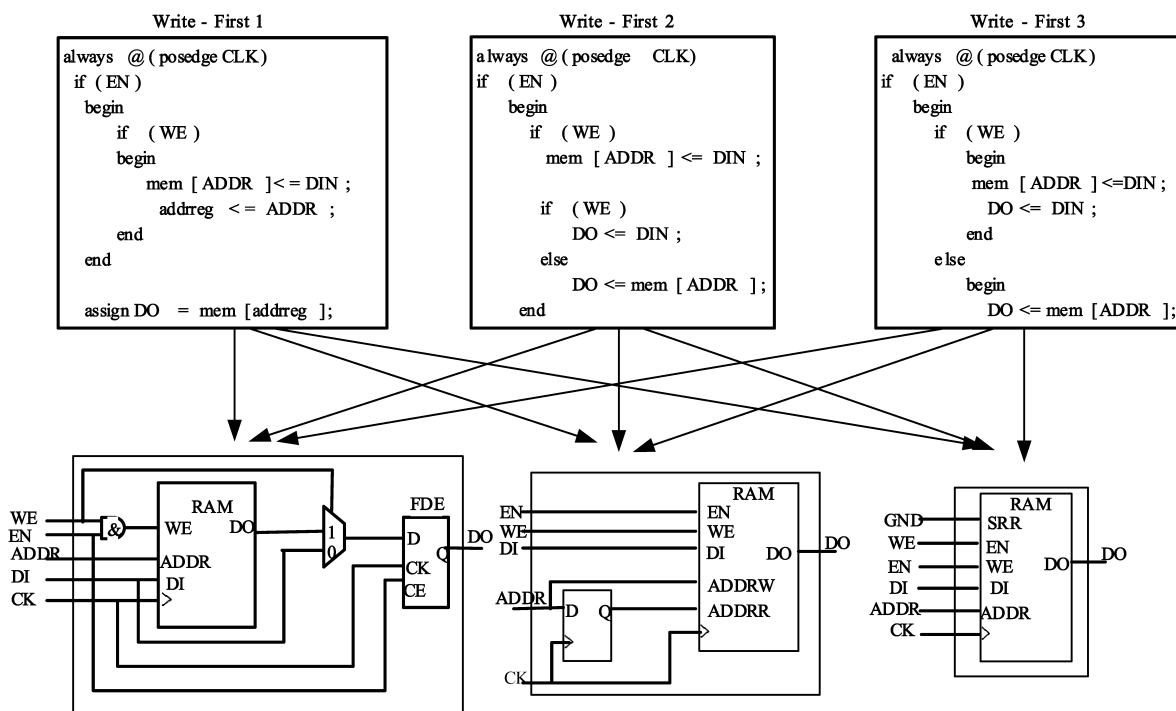


图2 功能相同RAM源描述和目标RAM结构的对应图

3 RAM的工艺映射

针对第2节提出的技术难点,提出解决方法:建立工艺无关的RAM统一模型.该模型统合了RAM的各种特征,以便于实现包含建模、模式匹配、造价计算、绑定等步骤的RAM工艺映射.

3.1 工艺无关的RAM模型

各种电子应用系统的RAM结构可以抽象成包括Memory Array, Write block, Read block, Write control, Read control, Write address, Read address, Write data 和 Read data 模块的电路,基于此,对于多样性RAM源描述,可建立图3所示的识别数据结构模型,实现去多样性目的.该数据结构会转化成图4工艺映射统一结构模型.图4模型与工艺无关,是RAM工艺映射实现的关键,涵盖常用RAM结构.统一模型中RAM的特征通过表2中六个关键性特征属性描述,即端口模式、管脚类型、地址模式、写模式、读模式和写读模式,各种源和

目标RAM结构都从该六种属性描述.

源RAM经过Rsyn的识别建模,可以转化成统一模型的数据结构,目标RAM(库单元中的RAM)描述也基于统一模型的数据结构,这样保存了源RAM和目标RAM转化成的数据结构都是统一模型的子集.不同种源和目标RAM就可以按照特征配置,有相同的RAM标准模型结构,使得复杂的RAM工艺映射成为可能.

表2 RAM工艺映射模板特征表

RAM模板特征	特征分类
端口模式	单端口,伪双端口,全双端口,多端口
管脚类型	地址,数据,时钟,读使能,写使能,clock使能
地址模式	是否寄存,即同步,还是异步
写模式	是否寄存,即同步,异步
读模式	是否寄存,即同步,异步
写读模式	先读,先写,保持

引入该模型使得源RAM和目标RAM的多样、多模式统一化,进而使RAM工艺映射难题得以解决.由于该模型与工艺无关,基于不同系列FPGA器件RAM

的工艺映射,只需把器件 RAM 结构按照图 4 统一模型结构描述库文件,就可以实现面向各种不同 FPGA 的移植.以 Xilinx Virtex、Virtex2 系列 FPGA 的 RAM 映射为例. Virtex2 系列 FPGA 的 RAM 与 Virtex 的 RAM 有相同的端口模式、管脚类型、地址模式、写模式、读写模式,

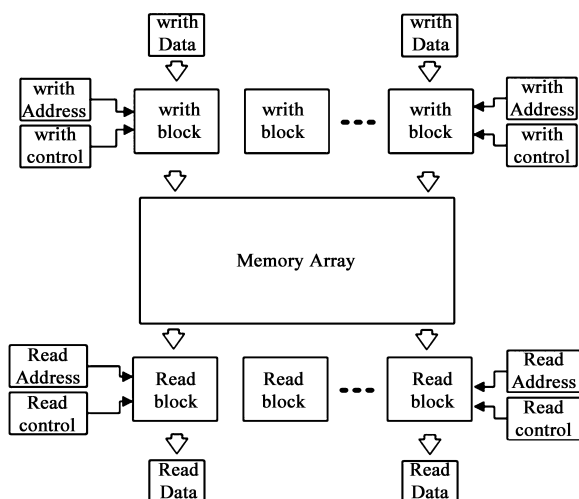


图3 工艺无关的RAM识别数据模型

唯一不同的是 Virtex 的 RAM 只有先写工作模式,而 Virtex2 的 RAM 有先写、先读、保持三种工作模式,这点在工艺映射库文件中,通过读写工作模式属性设定.这样识别出的 RAM 结构按照其各 RAM 特征属性,便可以映射到库中 RAM.

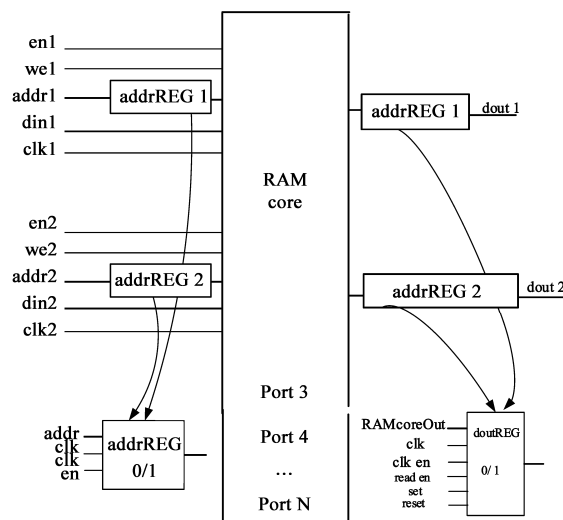


图4 工艺无关的RAM统一结构模型图

3.2 RAM 工艺映射实现

工艺映射过程主要分为建模、模式匹配、造价计算、绑定等四个步骤,下面依次介绍四个步骤,其中建模是整个过程的核心,也是本文的介绍重点.

3.2.1 建模

建模是通过分析源 RAM (srcRAM) 结构和相关逻辑特征,建立 RAM 统一模型的过程,主要包括预处理、端口压缩、规模判断、建模和后处理五个步骤,算法实现的伪代码如下:

```

1  if (PortNum(srcRAM) > 1) {
2      for each port p of srcRAM {
3          compress(p, srcRAM);
4      }
5  }
6  checkRAMbuildbyScale(srcRAM);
7  for each port p of srcRAM {
8      if(port p is write-first)
9          doWriteFirstModel(p, srcRAM);
10     else if(port p is read-first)
11         doReadFirstModel(p, srcRAM);
12     else if(port p is no-change)
13         doNochangeModel(p, srcRAM);
14     else continue;
15 }
16 postHandling();

```

其中 1~4 行是预处理,1~3 行是端口压缩,4 行是规模判断,5~12 建模,13 是后处理.

1~3 行端口压缩,在工艺映射前,从 Verilog HDL 中识别出的 RAM 存在多个端口情况,其 RAM 功能可能只是单端口或者是双端口 RAM,为此可以按照地址控制和读写操作特点,合并多端口为单端口或双端口,这样可以达到 RAM 工艺映射、提高 RAM 资源利用率的效果.

第 4 行是规模判断,判断映射目标选择 DRAM 还是 BRAM 的过程,目的是为了使源 RAM 映射到最佳类型和数量的 FPGA 片上 RAM 资源.由于多个 DRAM 加额外逻辑可以实现和 BRAM 一样的功能,因此源 RAM 既可以映射到 BRAM,也可映射到 DRAM 加辅助逻辑,但是这两种映射方案在电路性能和片上资源利用率方面均有所不同,因此需要合理选择映射目标,以避免大规模 RAM 采用 DRAM 造成 FPGA 片上逻辑资源浪费,性能降低;又或者小规模 RAM 采用 BRAM 造成 BRAM 资源浪费的问题.具体实现主要是筛选出可以用 BRAM 实现的大规模 RAM,以目标器件 FPGA 中 BRAM 的存储空间(tgtSize)作为筛选标准,当 srcRAM 的存储空间小于 tgtSize 时,用 DRAM 实现,不经过下面的建模处理;反之,经过下面建模处理,用 BRAM 实现.

其中 5~12 行是建模的核心算法,依次遍历源 RAM 的每一个端口,按照 RAM 的三种工作模式——write-first、read-first、no-change 作为建模处理的三个主分支,而后考虑管脚类型、地址模式、写模式、读模式的特

征,把源 RAM 和可能相关逻辑单元重构为统一模型结构.以图 2 中提到多目标映射的单端口同步读且 write-first 工作模式为例说明的建模原理,设定三个例子中 RAM 的存储规模都是 4096X4(地址深度 X 数据位宽),如图 4,第一步先从直译出的电路网表中识别出某种模式的 RAM 以及相关逻辑单元.图 2 中虽然 write-first2

和 write-first3 的描述不同,但是直接翻译出的电路结构相同,所以三个描述归为两种.然后建模,源 RAM 和相关逻辑单元结构,从端口模式、管脚类型、地址模式、写模式、读模式和写读模式六个方面,通过结构重组、配置成 RAM 统一模型,如图 5,把源 RAM 和相关的寄存器、与门、选择器逻辑单元重构为右图结构.

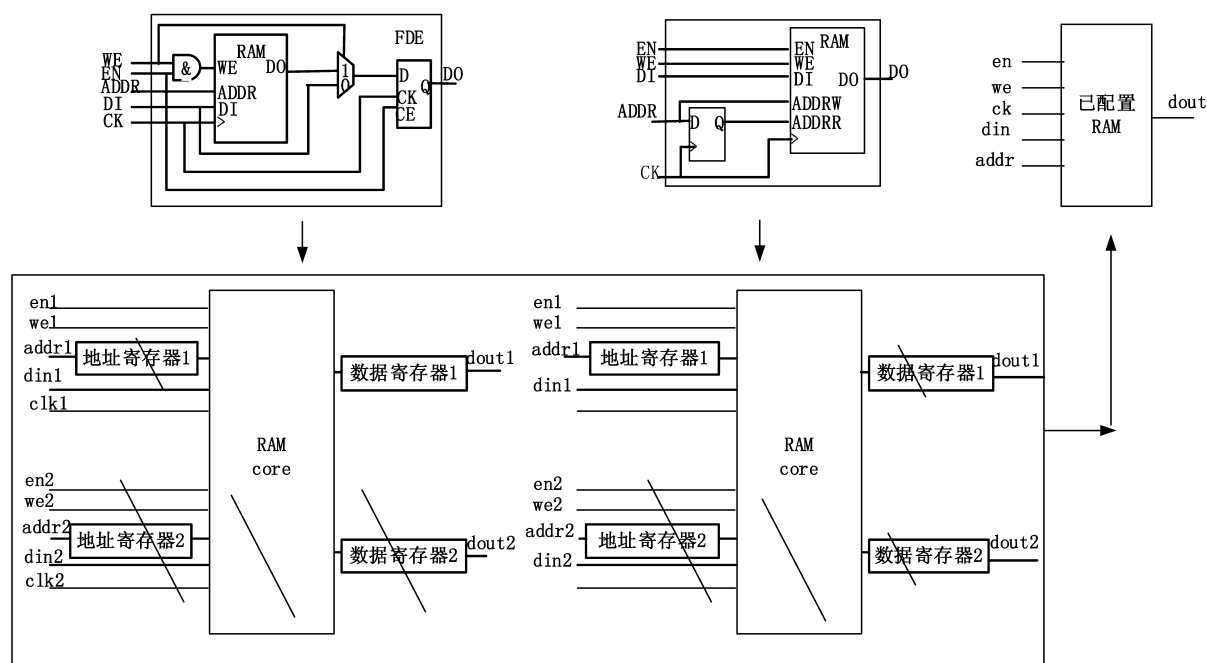


图5 write-first RAM建模的实现原理图

第 13 行的后处理过程主要处理一些不满足第 5 ~ 12 行处理要求,但又不能直接映射到 FPGA 片上 RAM 的情况,比如一个规模小的 RAM,两个端口写同步,读异步,读端口连接寄存器,这类 RAM 和连接的寄存器单元由于 RAM 规模较小,只能放在后处理,即通过统一模型重构,再实现工艺映射.

3.2.2 模式匹配

模式匹配是指配置成 RAM 统一模型结构的源 RAM,按照 RAM 六个特征属性,在工艺映射库中选择属性匹配的 RAM 库单元作为目标 RAM 的过程.工艺映射库中包含某系列 FPGA 片上的各种 RAM 单元,为源 RAM 提供映射目标 RAM 单元,其中映射库中的 RAM 单元都以 RAM 统一模型描述.匹配过程包括:精准匹配、模糊匹配.匹配原理公式如下:

$$\text{Source}(pm, pt, adm, w, r, wr) \leq \text{Target}(pm, pt, adm, w, r, wr);$$

其中 pm 、 pt 、 adm 、 w 、 r 、 wr 分别表示 RAM 的端口模式、管脚类型、地址模式、写模式、读模式、写读模式, $\text{Source}()$ 和 $\text{Target}()$ 分别表示源 RAM 和目标 RAM 模式匹配函数,包含了精准匹配和模糊匹配.精准匹配要求目标

RAM 的管脚类型、地址模式、写模式、读模式、写读模式等特征和源 RAM 的完全一致.模糊匹配则需要目标 RAM 特征和源 RAM 的特征满足包含关系.在匹配的过程中,先执行精准匹配,对于不满足精准匹配的情况,才进行模糊匹配.以图 6 所示为例,先精准匹配,得到单端口、同步读、先写的 RAM4096X1、RAM2048X2、RAM1024X4 可以匹配,单端口、异步读 RAMD16X1 不匹配,再通过模糊匹配得到双端口、同步读、先写的 RAM1024X4 也可以匹配,最后得到在映射工艺库中共有 4 个目标 RAM 满足匹配条件.

3.2.3 造价计算与绑定

如图 6 所示,由于经模式匹配后,映射库中仍存在 4 种可以匹配源 RAM 的目标 RAM 单元,因此需要进一步找到最优匹配的目标 RAM 单元,以达到 FPGA 片上 RAM 资源的高利用率.实现方法:采用造价计算算法,计算出源 RAM 映射到每一个匹配目标 RAM 需要付出的造价值,经比较后,选取需要付出最小造价的目标 RAM 作为最优映射目标.造价计算公式如下:

$$\text{cost} = T(\text{area}) * (\text{sWid}/\text{tWid}) * (\text{sDep}/\text{tDep}) + f(\text{MUX}) + f(\text{FF});$$

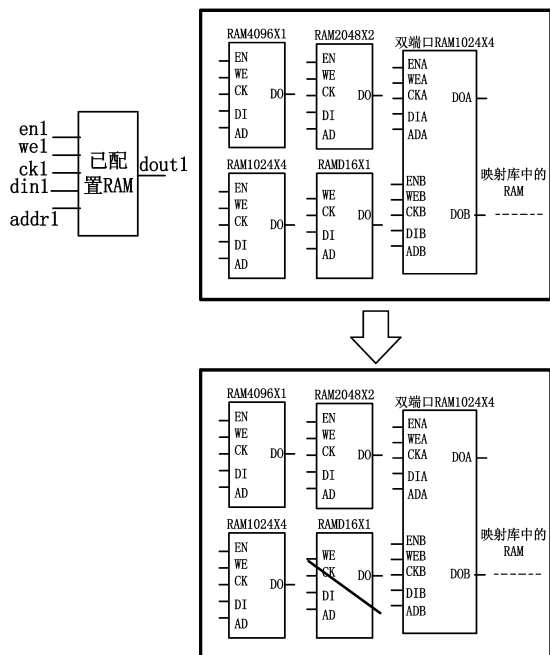


图6 RAM工艺映射的模式匹配原理图

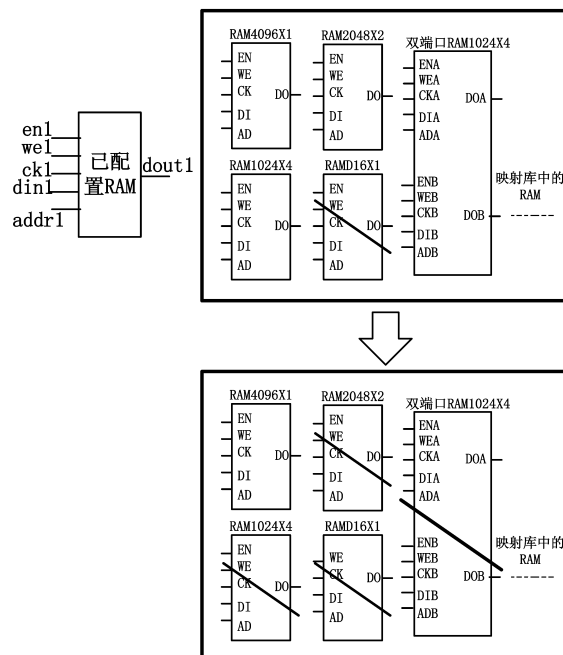


图7 RAM工艺映射的造价计算决策原理图

公式中, $T(\text{area})$ 表示采用目标 RAM 实现所需要占用 FPGA 的面积; $s\text{Wid}$, $s\text{Dep}$, $t\text{Wid}$, $t\text{Dep}$ 分别代表源 RAM 与目标 RAM 的数据宽度和存储空间数量; $f(\text{MUX})$ 和 $f(\text{FF})$ 两个函数表示造价值的补偿因子; $f(\text{MUX})$ 是针对源 RAM 深度大于目标 RAM 而言的, 这类 RAM 需要经过拆分才可以映射到目标 RAM 中, 因拆分产生的地址逻辑 MUX (Multiplexer, 选择器) 需要计入考虑; $f(\text{FF})$ 是针对同步读的源 RAM 映射到异步读的目标 RAM 而言的, 这种情况下需要增加新的 FF (Flip-Flop, 触发器) 实现源 RAM 的同步读功能, 所增加的 FF 面积同样需计入考虑。

图 7 中, 源 RAM 为 RAM4096X4, 匹配中的 4 个 cost 值分别为:

$$\text{cost}(\text{RAM4096X1}) = 4096 * (4/1) * (4096/4096) = 16384$$

$$\text{cost}(\text{RAM2048X2}) = 4096 * (4/2) * (4096/2048) + 16(f(\text{MUX})) = 16400$$

$$\text{cost}(\text{RAM1024X4}) = 4096 * (4/4) * (4096/1024) + 32(f(\text{MUX})) = 16416$$

$$\text{cost}(\text{双端口 RAM1024X4}) = 2 * (4096 * (4/4) * (4096/1024) + 32(f(\text{MUX}))) = 32382$$

经造价计算后, RAM4096X1 被选取为最优映射目标。

绑定指将源 RAM 的连接和属性绑定到最优目标 RAM 单元上的过程, 如果需要分裂的, 则按照地址和数据宽度分裂源 RAM, 以保证既可以映射到目标 RAM 中, 又可以达到对 RAM 资源使用率最小。如图 8 所示, 需要把源 RAM 分裂成 4 个, 分别绑定到 4 个并联的最

优匹配——RAM4096X1。

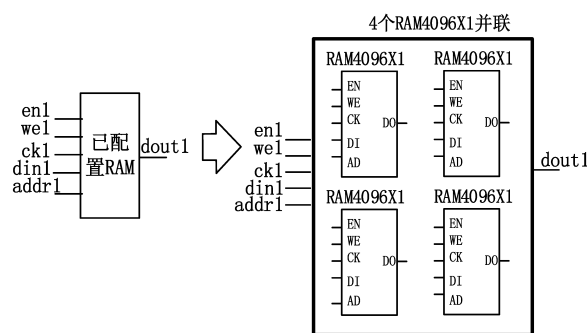


图8 RAM工艺映射的绑定原理图

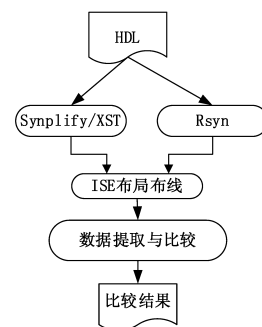


图9 实现测试平台图

4 验证分析

验证在 Linux 系统下进行, 同时嵌入到图 9 的自动化测试平台中, 最终输出比较结果, 如表 3。对比实验的

目标器件为 Xilinx Virtex 和 Virtex2^[14] 系列芯片,是项目支持的器件.对比实验的测试用例为 OC Verilog^[18] 测试集中抽取的 10 个用例,均采用 Verilog HDL 语言描述,含有不同结构的 RAM 描述,LUT 数占用规模在几千到几万不等.本文提出的工艺映射方法实现为 Rsyn,参与对比的综合工具分别为 Synplify-pro2012 和 Xilinx 的 XST10.1(其中 XST10.1 是 Xilinx ISE 软件系列中能支持 Virtex 及 Virtex2 器件的最高版本).

表 3 Rsyn 与 Synplify 和 XST 的 RAM 工艺映射效果对比列表

测试例		结果分析		
		Synplify-pro	XST	Rsyn
1	pipelined_fft_64	4 BRAM	4 BRAM	4 BRAM
2	reed_solomon_codec_generator	1 BRAM	1 BRAM	1 BRAM
3	reed_solomon_decoder	4 BRAM	11 BRAM *	4 BRAM
4	aquarius_perf	36 BRAM	36 BRAM	36 BRAM
5	Spymaster	2 BRAM	2 BRAM	2 BRAM
6	vga_lcd	2 BRAM	2 BRAM	2 BRAM
7	tiny_tate_bilinear_pairing_911	33 BRAM	34 BRAM *	33 BRAM
8	Ethmac	3 BRAM	3 BRAM	3 BRAM
9	sdc_fifo	1 BRAM	1 BRAM	1 BRAM
10	opb_vga_char_display_nodac	3 BRAM	3 BRAM	3 BRAM

表 3 显示 Synplify、XST、Rsyn 综合处理 RAM 的效果对比情况,Rsyn 实现 RAM 工艺映射和 Synplify 效果相同;10 个和 XST 实现 RAM 工艺映射效果相同,其中对于 reed_solomon_decoder 和 tiny_tate_bilinear_pairing_911,XST 分别比 Rsyn 多产生 7 个和 1 个 RAMB,是用来实现 ROM 功能的.

表 4 是面向 Virtex 和 Virtex2 系列 FPGA 的 RAM 工艺映射结果对比表.由于 Virtex 的 BRAM 是 4Kb 规模,且只有先写工作模式,而 Virtex2 的 BRAM 是 16Kb 规模、有先写、先读、保持等工作模式,因此实现的结果有差别,如表 4 所示,reed_solomon_codec_generator、reed_solomon_decoder、aquarius_perf、spimaster 和 ethmac 中的存储模块,在 virtex 器件中用 DRAM 实现;而在 Virtex2 中用 BRAM 实现.sdc_fifo 中的存储模块,在 virtex 器件不支持;而在 Virtex2 中,可以用 BRAM 实现.本文的 RAM 工艺映射技术可以适用到不同厂商、不同系列 FPGA 中的 RAM,如 Altera 和 Xilinx,Virtex、Virtex2、Virtex7 等,只需要更新某 FPGA 中 RAM 各种工作模式在工艺映射库中的描述.

表 4 面向 Virtex 和 Virtex2 FPGA 器件的 RAM 工艺映射对比表

测试例		产生 RAM 数量(个)	
		Virtex	Virtex2
1	pipelined_fft_64	8 BRAM	4 BRAM
2	reed_solomon_codec_generator	DRAM	1 BRAM
3	reed_solomon_decoder	DRAM	4 BRAM
4	aquarius_perf	DRAM	36 BRAM
5	Spymaster	DRAM	2 BRAM
6	vga_lcd	5 BRAM	2 BRAM
7	tiny_tate_bilinear_pairing_911	75 BRAM	33 BRAM
8	Ethmac	DRAM	3 BRAM
9	sdc_fifo	Virtex 不支持 *	1 BRAM
10	opb_vga_char_display_nodac	12 BRAM	3 BRAM

5 总结

针对实用电子系统存储器及其描述多样性带来的 RTL 综合实现困难,且关键技术垄断在几家国外公司而不公开的现状,本文提出并系统介绍了 RTL 综合中 FPGA 片上 RAM 工艺映射方法,建立了一套工艺无关的 RAM 统一模型,把各种源和目标 RAM 按照端口模式、管脚类型、地址模式、写模式、读模式和读写模式配置到统一 RAM 模型上,解决了多种模式的源 RAM 映射到多种模式目标 RAM 的难题,该模块已经集成在自主开发的 Hqsyn 综合工具中,并已经商用,具有较高实用价值.实验结果验证了本方法的有效性,即在产生 RAM 性能方面,可以和主流综合工具——Synplify 和 XST 处于接近水平.本文验证是通过 Xilinx,Virtex、Virtex2,系列 FPGA 提到的 RAM 工艺映射技术可以适用到不同厂商、不同系列 FPGA 中的 RAM,如 Altera 和 Xilinx,Virtex、Virtex2、Virtex7 等,只需要更新某 FPGA 中各种工作模式的 RAM 在工艺映射库中的描述即可.

参考文献

- [1] Steven J E. Wilton. Heterogeneous technology mapping for area reduction in FPGA's with embedded memory arrays [A]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems [C]. San Jose CA US: IEEE Trans, 2000. 19:56-68.
- [2] M Milford, J McAllister. Valved dataflow for FPGA memory hierarchy synthesis [A]. IEEE International Conference on Acoustics, Speech and Signal Processing [C]. Kyoto Japan: IEEE Press, 2012. 1645-1648.
- [3] Wang Yuxin, Zhang Peng, Cheng Xu, Jason Cong. An integrated and automated memory optimization flow for FPGA behavioral synthesis [A]. Asia and South Pacific Design Automation Conference-ASP-DAC [C]. Sydney, Australia: IEEE Press, 2012. 257-262.
- [4] T Kim, Liu C. Utilization of multiport memories for hierarchical data streams [A]. ACM/IEEE Design Automation

- Conference[C]. Dallas, Texas, US; IEEE Trans, 1993. 298 – 302.
- [5] L Ramachandran, D Gajski, V Chaiyakul. An algorithm for array variable clustering [A]. European Design and Test Conference[C]. Paris France; EDAC, 1994. 262 – 266.
- [6] P Marwedel, B Landwehr. Exploitation of component information in a RAM-based architectural synthesis system [A]. Logic and Architectural Synthesis, G. Saucier (Ed.) [D]. Chapman & Hall Britain, 1995. 1 – 11.
- [7] P Lippens, J Van Meerbergen, W Verhaegh, A Van Der Werf. Allocation of multiport memories for hierarchical data streams [A]. Proceedings of the IEEE International Conference on Computer-Aided Design [C]. Santa Clara, CA US; IEEE Trans, 1993. 728 – 735.
- [8] F Balasa, F Catthoor, H De Man. Data-driven memory allocation for multi-dimensional signal processing systems [A]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems [C]. San Jose CA US; IEEE Trans, 1994. 31 – 34.
- [9] D Karchmer, J Rose. Definition and solution of the memory packing problem for field-programmable systems [A]. IEEE/ACM International Conference On Computer-aided Design [C]. San Jose CA US; IEEE Trans, 1994. 20 – 26.
- [10] H Schmit, D Thomas. Array mapping in behavioral synthesis [A]. Proceedings of the 8th international symposium on System synthesis [C]. Cannes, France; ACM, 1995. 90 – 95.
- [11] 周海峰, 林争辉. RTL 级综合中存储器工艺映射算法的研究 [J]. 微电子报, 2001, 31 (6): 410 – 413.
Zhou Haifeng, Lin Zhenghui. Memory mapping algorithms in synthesis [J]. Microelectronics, 2001, 31 (6): 410 – 413. (in Chinese)
- [12] S Bakshi, D Gajski. A memory selection algorithm for high-performance pipelines [A]. Proceedings of the conference on European design automation [C]. Los Alamitos, CA, US; IEEE Computer Society, 1995. 124 – 129.
- [13] P K Jha, N D Dutt. Library mapping for memory [A]. Proceedings of 1997 European Design and Test Conference [C]. Washington, DC, US; IEEE Computer Society, 1997. 288.
- [14] Xilinx, Inc. Datasheet: Virtex-II and Virtex-IIe FPGA Family [M]. San Jose, CA, US, 2008.
- [15] Synopsis, Inc. Datasheet: Synplify Reference Manual [M]. Mountain View, CA, U. S. 2012.
- [16] Xilinx, Inc. Datasheet: XST Reference Manual [M]. San Jose, CA, US, 2008.
- [17] Altera, Inc. User Guide: Internal Memory (RAM and ROM) [M]. San Jose, CA, US, 2013.
- [18] Verilog hdl and VHDL opencores [M/OL]. <http://opencores.org/projects>. 2010 – 08.

作者简介



李 艳 女, 1981 年出生于内蒙古巴彦淖尔市, 博士研究生, 助理研究员, 主要研究领域为 FPGA 架构和 FPGA EDA 的设计。
E-mail: liyan_ic@163.com



张东晓 男, 1972 年出生, 北京人, 博士, FPGA EDA 领域专家, 主要研究领域为 EDA、VHDL/Verilog Hdl 语言、高级综合和相关算法; 2007 年作为 CEO, 联合创办北京飘石科技有限公司, 从事 FPGA 软件开发与服务, 提供通用平台化 FPGA 软件解决方案, 目前已经被国内多家 FPGA 单位所使用。
E-mail: zdx@uptops.com.cn