面向分组密码的四维度并行处理架构研究

王寿成[1],李功丽[1,2],严迎建[1],徐进辉[1]
(1. 解放军信息工程大学,河南郑州 450001; 2. 河南师范大学计算机信息工程学院,河南新乡 453002)

摘 要: 通过对分组密码算法加密特征的分析,将分组密码算法的并行性划分为分组内同操作并行性、分组内异操作并行性、分组间同操作并行性和分组间异操作并行性等四维度并行性,并根据此提出了基于 Amdahl 定律的分组密码四维度并行处理模型 FDPM. 该模型能够指导分组密码处理架构设计,为架构资源配置和并行性开发提供整体建议. 以 FDPM 为依据,提出了一种面向分组密码的可重构流处理架构 RCSA,该架构能够有效开发分组密码处理的并行性,在提高密码处理性能的同时也能提高资源利用率. 通过算法映射结果分析,证明了 FDPM 模型的正确性与 RCSA 架构的高效性.

关键词: 分组密码; Amdahl 定律; 四维度并行处理; 流体系结构; 加速比
中图分类号: TP309.7 **文献标识码:** A **文章编号:** 0372-2112 (2017)10-2457-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2017.10.021

Four Dimensions Parallel Processing Architecture for Block Cipher

WANG Shou-cheng[1], LI Gong-li[1,2], YAN Ying-jian[1], XU Jin-hui[1]
(1. *PLA Information Engineering University, Zhengzhou, Henan* 450001, *China*;
2. *College of Computer & Information Engineering, Henan Normal University, Xinxiang, Henan* 453002, *China*)

Abstract: According to the characteristics of block cipher algorithm, the parallelism of block cipher algorithm can be divided into Four Dimensions Parallelism (FDP), which consists of same operational parallelism in a block, different operational parallelism in a block, same operational parallelism among multiple blocks and different operational parallelism among multiple blocks. Four Dimensions Parallel Processing Model (FDPM) for block cipher based on Amdahl's law was proposed, which can guide the design of block cipher processing architecture and provide the overall suggestions for resource allocation and parallelism development. And then Reconfigurable block Cipher Stream Architecture (RCSA) based on FDPM was proposed, which can effectively develop parallelism of block cipher and improve processing performance and the utilization rate of resources. The experimental results prove the veracity of FDPM and efficiency of RCSA.

Key words: block cipher; Amdahl's law; four dimensions parallel processing; stream architecture; speedup

1 引言

作为保障网络与信息安全的有效手段,分组密码算法在数据加密、数字签名/认证及密钥管理等方面广泛应用,其高效实现直接影响网络与信息通信系统的性能. 并行计算作为实现高性能计算的重要技术,通过开发分组密码算法的并行性使加密操作同时执行,能够实现算法加密的并行加速. 因此针对分组密码算法并行处理的研究越来越多,如何充分挖掘分组密码算法固有的并行性,如何设计分组密码高效的并行处理架构,已经成为了信息安全领域的重要研究问题.

国内外针对分组密码算法的并行处理进行了一系列研究. Michael G 等提出的可重构密码多核处理器 MCCP[1] 集成了 4 个 32 bit 的处理核心和 1 个 8 bit 的任务调度器,多核间可实现密码处理的分组间并行,能够有效提升算法 ECB 加密性能. 孟涛等提出了一种基于 VLIW 的 4 路并行可重构分簇式分组密码处理架构 RCBCP[2],该架构可灵活重构为 4 个 32 bit、2 个 64 bit 和 1 个 128 bit 数据通路,能够实现不同密码算法的分组内或分组间并行,其加密性能可达到数百 Mbps. Gokhan S 等提出的可重构阵列结构处理器 Cryptoraptor[3],集成了 80 个可重构功能单元 PE,能够有效开发

收稿日期:2016-04-28;修回日期:2016-10-18;责任编辑:李勇锋
基金项目:国家自然科学基金(No.61404175)

分组密码算法处理的并行性,对 AES、DES、KASUMI 算法的实现性能达到了 6.4 Gbps、2.67 Gbps 和 1.0 Gbps. 此外,文献[4,5]也提出了一些分组密码处理架构,但上述架构均不能充分开发分组密码算法的并行性,并且普遍存在资源利用率较低的情况.

为充分挖掘分组密码算法处理的并行性并提高资源利用率,将分组密码的并行性划分为四维度并行性,并结合 Amdahl 定律定量分析了各维度并行性的执行时间和加速比. 根据此提出了基于 Amdahl 定律的分组密码四维度并行处理模型 FDPM 和基于 FDPM 的可重构分组密码流处理架构 RCSA,并分别进行了 AES、DES、IDEA、SAFER + 、SMS4 和 FOX64 算法的映射和性能分析,全面评估 RCSA 的算法适配能力和算法实现性能. 此外,本文还进行了 AES 算法在 RCSA 各规模结构下的面积能效比计算和比较,并提出了 FDP 并行性的开发优先顺序.

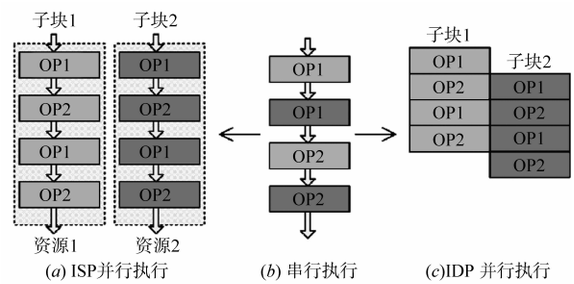
2 四维度并行处理模型 FDPM

算法程序可分解为串行执行部分和多种并行度的并行执行部分,采用并行架构对并行执行部分加速能够提升系统性能. Amdahl 定律指出"系统采用并行处理技术后所能获得的性能提升受限于系统中并行化部分所占比例",并使用加速比(Speedup,Sp)来衡量并行架构的并行处理能力和加速性能. Amdahl 定律如下:

$$\mathrm{Sp}=\frac{t_{\text{并行加速前}}}{t_{\text{并行加速后}}} \tag{1}$$

文献[6]和文献[7]指出分组密码算法处理具有两个方向的并行性,分别是分组内并行性(Parallelism In Block,IP)和分组间并行性(Parallelism Among Blocks,AP). 通过对大量分组密码算法的结构和特征进行分析发现,分组密码算法处理的并行性可以进一步划分,如图 1、图 2 所示,IP 并行性可以划分为组内同操作并行性(Same Operational Parallelism In Block,ISP)和组内异操作并行性(Different Operational Parallelism In Block,IDP),AP 并行性可以划分为组间同操作并行性(Same Operational Parallelism Among Blocks,ASP)和组间异操作并行性(Different Operational Parallelism Among Blocks,ADP). 本文将上述四种并行性合称为分组密码算法处理的四维度并行性(Four Dimensions Parallelism,FDP). 下文将结合 Amdahl 定律分别定量分析各维度并行性的执行时间和加速比.

IP 并行性旨在开发分组内可并行执行部分的并行性. 如图 1(*a*),ISP 通过扩展可并行执行资源来实现,理论上只要资源足够就可以通过 ISP 完全开发 IP 并行性. 如图 1(*c*),当 ISP 未完全开发 IP 并行性时,IDP 可以通过指令级并行来进一步开发 IP 并行性,此时不需要扩展资源,主要的实现方式为软件流水技术[8].



(*a*) ISP并行执行 (*b*) 串行执行 (*c*)IDP 并行执行

图1 IP并行执行示意

假设密码算法程序共有 w 个操作,每个操作的执行时间为 Δ,则其串行执行的时间 $t_s=w\Delta$. 若只开发 ISP,设第 i 个操作的并行度为 α_i($\alpha_i\leqslant R$,R 为可并行执行资源数),则 ISP 并行执行的时间为:

$$t_{\mathrm{ISP}}=\sum_{i=1}^{w}\frac{\Delta}{\alpha_i} \tag{2}$$

根据 Amdahl 定律,ISP 的加速比为:

$$\mathrm{Sp}_{\mathrm{ISP}}=\frac{t_s}{t_{\mathrm{ISP}}}=\frac{w\Delta}{\sum_{i=1}^{w}\frac{\Delta}{\alpha_i}}=\frac{w}{\sum_{i=1}^{w}\frac{1}{\alpha_i}} \tag{3}$$

若只开发 IDP,设第 i 个操作的并行度为 β_i($\beta_i\leqslant N_{op}$,N_{op} 为可并行执行异操作数),则 IDP 并行执行的时间为:

$$t_{\mathrm{IDP}}=\sum_{i=1}^{w}\frac{\Delta}{\beta_i} \tag{4}$$

根据 Amdahl 定律,IDP 的加速比为:

$$\mathrm{Sp}_{\mathrm{IDP}}=\frac{t_s}{t_{\mathrm{IDP}}}=\frac{w\Delta}{\sum_{i=1}^{w}\frac{\Delta}{\beta_i}}=\frac{w}{\sum_{i=1}^{w}\frac{1}{\beta_i}} \tag{5}$$

若同时开发 ISP 和 IDP 两种并行性,此时第 i 个操作的并行度为 $\alpha_i\beta_i$,则 IP 并行执行的时间为:

$$t_{\mathrm{IP}}=\sum_{i=1}^{w}\frac{\Delta}{\alpha_i\beta_i} \tag{6}$$

根据 Amdahl 定律,IP 加速比为:

$$\mathrm{Sp}_{\mathrm{IP}}=\frac{t_s}{t_{\mathrm{IP}}}=\frac{w\Delta}{\sum_{i=1}^{w}\frac{\Delta}{\alpha_i\beta_i}}=\frac{w}{\sum_{i=1}^{w}\frac{1}{\alpha_i\beta_i}} \tag{7}$$

AP 并行性负责开发 ECB 加密模式下或交错技术下的分组间并行性,主要通过扩展可同时执行分组数实现性能提升. 如图 2(*a*),ASP 通过扩展可并行执行资源来实现多个分组的并行执行,其并行度取决于资源数量. 如图 2(*c*),若开发 IDP 后,软件流水线仍有空闲,则可以通过 ADP 进一步开发 AP 并行性,此时不需要扩展资源,能够实现在提高资源利用率的同时有效提升系统性能.

假设单密码分组的执行时间为 t_b. 若只开发 ASP,设分组并行度为 μ($\mu=R$,R 为可并行执行资源数),则

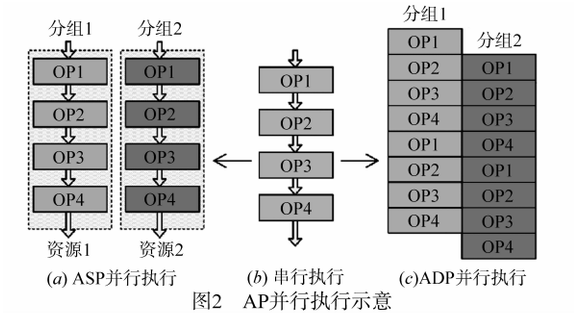


图2　AP并行执行示意

执行 μ 个分组的串行执行时间和 ASP 并行执行的时间分别为:

$$t_s=\mu t_b, t_{\mathrm{ASP}}=t_b \tag{8}$$

根据 Amdahl 定律,ASP 的加速比为:

$$\mathrm{Sp}_{\mathrm{ASP}}=\frac{t_s}{t_{\mathrm{ASP}}}=\frac{\mu t_b}{t_b}=\mu \tag{9}$$

若只开发 ADP,设分组并行度为 $\delta(\delta\leqslant N_{op}$,$N_{op}$ 为可并行执行操作数),则执行 δ 个分组的串行执行时间和 ADP 并行执行的时间分别为:

$$t_s=\delta t_b, t_{\mathrm{ADP}}=t_b+[(\delta-1)+\rho]\Delta \tag{10}$$

其中 ρ 指在开发 ADP 时软件流水调度多个分组产生的填充时间. 根据 Amdahl 定律,ADP 的加速比为:

$$\mathrm{Sp}_{\mathrm{ADP}}=\frac{t_s}{t_{\mathrm{ADP}}}=\frac{\delta t_b}{t_b+[(\delta-1)+\rho]\Delta} \tag{11}$$

假如同时开发 ASP 和 ADP,则执行 $\mu\times\delta$ 个分组的串行执行时间和 AP 并行执行的时间分别为:

$$t_s=\mu\delta t_b, t_{\mathrm{AP}}=t_b+[(\delta-1)+\rho]\Delta \tag{12}$$

根据 Amdahl 定律,AP 加速比为:

$$\mathrm{Sp}_{\mathrm{AP}}=\frac{t_s}{t_{\mathrm{AP}}}=\frac{\mu\delta t_b}{t_b+[(\delta-1)+\rho]\Delta} \tag{13}$$

综上所述,若同时开发分组密码算法处理的 FDP 并行性,此时单分组执行时间 t_b 为 t_{IP},则执行 $\mu\times\delta$ 个分组的时间为:

$$\begin{aligned}T_{\mathrm{FDP}}&=t_{\mathrm{IP}}+[(\delta-1)+\rho]\Delta\\&=\sum_{i=1}^{w}\frac{\Delta}{\alpha_i\beta_i}+[(\delta-1)+\rho]\Delta\end{aligned} \tag{14}$$

即单个分组的平均执行时间为:

$$t_{\mathrm{FDP}}=\frac{\Delta}{\mu\delta}\left[\sum_{i=1}^{w}\frac{1}{\alpha_i\beta_i}+(\delta-1)+\rho\right] \tag{15}$$

根据 Amdahl 定律,FDP 加速比为:

$$\begin{aligned}\mathrm{Sp}_{\mathrm{FDP}}&=\frac{t_s}{t_{\mathrm{FDP}}}=\frac{w\Delta}{\frac{\Delta}{\mu\delta}\left[\sum_{i=1}^{w}\frac{1}{\alpha_i\beta_i}+(\delta-1)+\rho\right]}\\&=\frac{\mu\delta w}{\sum_{i=1}^{w}\frac{1}{\alpha_i\beta_i}+(\delta-1)+\rho}\end{aligned} \tag{16}$$

式(16)是基于 Amdahl 定律的加速比模型,称之为分组密码的四维度并行处理模型(Four Dimensions Parallel Processing Model,FDPM),式(16)中,w 是密码算法参数,α_i, β_i,μ,δ 和 ρ 是架构的并行参数. 由 FDPM 可知,提高四维并行性的任一并行性都能有效提高分组密码处理性能,但在资源受限的实际应用中,为平衡资源利用与性能提升的关系,需要分析各维度并行性对性能提升的影响而选择其开发优先顺序.

3　可重构流处理架构 RCSA

3.1　总体结构

以上文提出的分组密码四维度并行处理模型 FDPM 为理论依据,借鉴流处理器架构,本文提出了一种面向分组密码的可重构流处理架构(Reconfigurable Block Cipher Stream Architecture,RCSA). RCSA 的总体结构如图 3 所示.

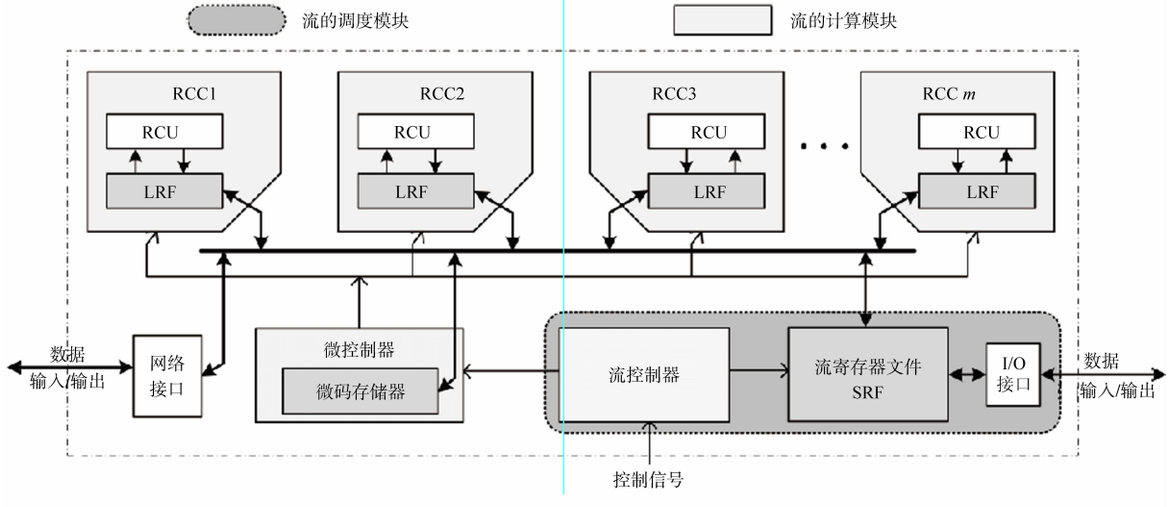


图3　RCSA总体结构

RCSA 将数据流的调度和计算相分离,同时对数据调度指令和密码操作指令的执行模块也进行解耦合,实现了数据调度与密码运算的任务级并行性(Task Level Parallelism,TLP). 基于此,RCSA 的硬件结构可划分为流的调度模块和流的计算模块. 流的调度模块负责控制协调各功能部件执行并负责提供数据流和指令流,流的计算模块专注执行密码运算. 其架构特点如下:

(1)RCSA 将控制系统进行了分层设计,分为流控制器和微控制器. 流控制器是控制逻辑的核心,在不违反相关性前提下尽可能地并行调度数据存储操作及密码运算操作,从而开发存储调度与算法执行间的任务级并行. 微控制器以 SIMD 或 MIMD 的指令发射方式对超长指令字(Very Large Instruction Word,VLIW)进行发射,控制可重构密码运算簇(Reconfigurable Cipher processing Cluster,RCC)完成密码操作.

(2)根据流的特征,RCSA 的数据访问被分解成了多个存储层次. 簇内本地寄存器文件(Local Register File,LRF)与流寄存器文件(Stream Register File,SRF)和片外 SDRAM 构成了三级层次化的存储结构,在簇内用 LRF 来捕捉数据重用局域性,同时 LRF 也是可重构密码流处理单元(Reconfigurable Cipher processing Unit,RCU)的数据源和中间结果暂存器;在中间层用 SRF 来捕捉生产者-消费者的局域性,且只在输入/输出数据的缓存和加载调度时访问 SRF;最外层的 SDRAM 为 RCSA 提供数据流和加/解密微指令.

3.2 密码运算核心

密码运算单元是 RCSA 完成密码操作的核心结构,是实现密码处理 FDP 并行性的硬件基础,其设计依据是四维度并行处理模型 FDPM. 如图 4 所示,RCSA 的密码运算单元由 $m \times n$ 规模的密码流处理 BANK 组成,其中 m 个 RCC 用于支持并行度为 m 的 ASP 并行性开发,簇内 n 个 BANK 能够支持最大并行度 α_{max} 为 n 的 ISP 并行性开发. IDP 和 ADP 通过软件流水技术来实现,其开发程度取决于 ISP 并行度. 基于安全性和实现难度的考虑,密码算法的运算粒度通常是 8 bit、16 bit、32 bit、64 bit 或 128 bit,所以本文将单 BANK 的数据位宽和最小的运算粒度选定为 8 bit. 考虑到分组密码的分组宽度通常为 128 bit,故 n 的最大值为 16. 为方便数据的组织调度,m 和 n 的值均选定为 2 的整数倍.

RCC 主要由 LRF 和 RCU 组成,LRF 和 RCU 采用分块式组织的方式,被划分为 n 个 BANK. 单个 BANK 的数据位宽和运算粒度为 8 bit,通过指令广播的方式,n 个 BANK 可以将数据组织成 $8n$ bit,在单周期内高效灵活实现了数据组织. RCU 是 RCSA 中密码操作执行的基本单元,是软件流水技术实现的硬件支撑,是实现

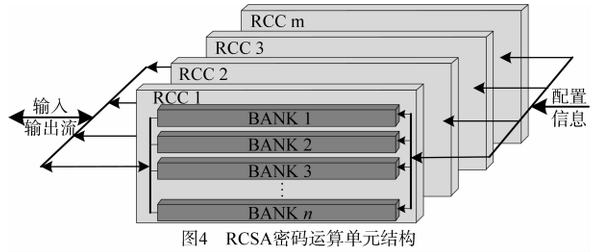


图4 RCSA密码运算单元结构

IDP 和 ADP 并行性的核心结构. 单 BANK 的基本运算粒度为 8 bit,经 BANK 间共享拼接使用,RCU 可完成 8-8 为基础扩展的 S 盒替代操作与 $8n$ bit 内的移位操作、$GF(2^n)$上的矩阵乘法、算术乘法操作、算术模加/减操作和逻辑运算等. 此外,n 个 BANK 共享一个 128 bit 的比特置换单元,用于完成 128 bit 内的置换操作和大位宽移位操作,同时还作为通信单元完成 BANK 间数据交互.

4 算法映射与性能评测

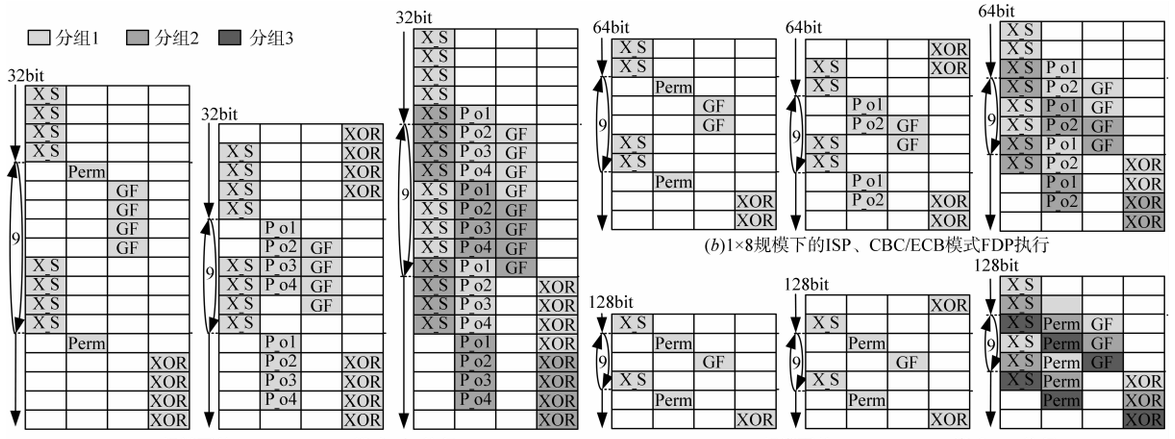
4.1 AES 算法映射与模型验证

对 AES、DES、IDEA 和 SMS4 等 40 余种公开分组密码算法的映射结果表明,RCSA 能够从四个并行维度高效并行处理密码算法. 本文以 AES 算法在 1×4、1×8 和 1×16 规模架构下映射过程为例,通过分析算法在不同维度并行性的不同并行度下的执行过程和结果,分析算法的 CBC/ECB 加密性能,来验证 FDPM 模型的正确性.

AES 算法的操作可分解为 10 个 128 bit 的置换操作(P)、36 个 32 bit 的有限域乘法操作(GF)、160 个 8 bit 的绑定前异或的 S 盒操作(X_S)和 16 个 8 bit 的异或操作,即 AES 算法在 ECB 加密模式下 w_{ECB} 为 222. 其中置换操作为串行部分,有限域乘法操作的最大并行度为 4,S 盒和异或操作的最大并行度为 16. 在 CBC 加密模式下还要多 16 个 8 bit 的初始异或操作,即 w_{CBC} 为 238. 在 RCSA 架构中,ISP 和 ASP 通过硬件资源来实现,其中 RCC 用于开发 ASP,其数量等于 ASP 并行度 μ;BANK 用于开发 ISP,其值等于 ISP 最大并行度 α_{max}. IDP 和 ADP 通过软件流水技术来实现.

在 1×4 规模下,IDP 最大并行度 β_{max} 为 2,ADP 并行度 δ 为 2. 如图 5(a)所示,通过软件流水技术开发数据子块的 IDP 和分组间的 ADP,从而加速密码分组的执行速度,在 ECB 模式下为开发分组间的 ADP,单分组执行过程需要增加较多填充时间. 在 1×8 规模下,IDP 最大并行度 β_{max} 为 3,ADP 并行度 δ 为 2. 如图 5(b)所示,在 CBC 加密模式下仅能开发 IDP,此时 β_{max} 为 2,即最多有 2 个不同操作并行. 此时不需要插入太多填充时间就能够开发分组间的 ADP. 在 1×16 规模下,IDP 最大并行度 β_{max} 为 3,ADP 并行度 δ 为 3. 如图 5(c)所

示,此时 α_{max} 为 16,即 ISP 已完全开发 IP 并行性,故 1 × 16 规模无法开发 IDP. 值得注意的是图 5 中 FDP 执行时置换操作表示按数据分块输出结果,并不代表置换操作并行执行.



(a)1×4规模下的ISP、CBC/ECB模式FDP执行　(b)1×8规模下的ISP、CBC/ECB模式FDP执行　(c)1×16规模下的ISP、CBC/ECB模式FDP执行

图5　AES算法映射结果

设每个操作执行的时间 Δ 为 1,则 AES 算法在完全串行执行时的 ECB 加密时间 t_{ECB} 为 222,CBC 加密时间 t_{CBC} 为 238. 根据图 5 的算法映射结果,可计算出 AES 算法在 1 ×4、1 ×8 和 1 ×16 规模架构下的并行执行时间,AES 算法在各规模下的执行时间如表 1 所示.

表 1　AES 算法映射的执行时间

规模	ISP 执行			FDP 执行		
	CBC	ECB	δ	CBC	ECB	δ
1 ×4	94	90	1	64	85	2
1 ×8	52	50	1	42	43	2
1 ×16	31	30	1	31	32	3

从表 1 看出,仅通过开发算法的 ISP,就能有效提升其处理性能. 在 1 ×16 规模下,16 个 BANK 可以完全开发算法 IP 并行性,使得 AES 算法的 CBC 加密速度达到最大. 此外,通过 FDP 进一步开发算法并行性,在减小分组执行时间的同时还能扩展可并行执行分组数,使得 AES 算法的 CBC 加密性能和 ECB 加密性能同时得到提升. 根据 Amdahl 定律和 FDPM 模型,可以计算出 FDPM 的理论 FDP 加速比和 RCSA 的实际 FDP 加速比,具体结果如表 2 所示.

表 2　FDPM 验证结果

规模	α_{max}	β_{max}	μ	δ	ρ	FDPM		RCSA	
						Sp_{CBC}	Sp_{ECB}	Sp_{CBC}	Sp_{ECB}
1 ×4	4	2	1	2	20	3.72	5.22	3.72	5.22
1 ×8	8	3	1	2	1	5.67	10.33	5.67	10.33
1 ×16	16	3	1	3	0	7.68	20.81	7.68	20.81

从表 2 中可以看出,RCSA 的实际 FDP 加速比与 FDPM 的理论 FDP 加速比完全一致,从而证明了 FDPM 模型的正确性与 RCSA 架构的合理性. 通过开发算法的 FDP 并行性,1 ×4、1 ×8 和 1 ×16 规模架构可以将算法执行效率较完全串行执行方式提升 3.72 ~ 20.81 倍,有效加速了密码算法执行速度.

4.2　性能评测与分析

本文采用 Verilog 语言对 RCSA 进行了描述,并利用综合工具基于 65nm CMOS 工艺标准单元库逻辑综合获取了硬件资源代价信息. 根据综合结果,RCSA 的工作频率可达到 500MHz. 由于没有统一的评估标准来衡量密码处理架构的性能,本文选择文献中性能分析最为广泛的 AES 算法进行对比. 在进行处理器面积、分组并行数、频率等性能指标对比的同时,还分别进行了 AES 算法在 CBC 加密模式和 ECB 加密模式下的吞吐率对比,以及相应的面积能效比(单位面积的性能)分析. 各规模结构和相关文献架构的面积、吞吐率等性能指标如表 3 所示.

从表 3 可以看出 RCSA 各规模结构都具有较高的 CBC/ECB 加密性能. 当簇内 BANK 数量 n 固定时,各规模的 CBC 加密性能一致,且随着 n 的增加,其 CBC 加密性能也得到提升. 在 ECB 加密模式时,m 和 n 的增加都能够提升 RCSA 的 ECB 加密性能. 与其它文献相比较,RCSA 各规模架构均具有一定的性能优势,且面积能效比较高. 从表 3 吞吐率分析可知,在 BANK 总数相等的情况下,单簇内 BANK 数量 n 越大,相应规模架构的加密性能越高. 在资源受限的实际应用中,为提高处理器的加密性能,应当通过扩展单簇内 BANK 数量 n 优先开发 ISP 并行性.

为更全面地评估 RCSA 的适配能力和实现性能,本文分析了 SP、Feistel 和 L-M 三种结构在 RCSA 上的映射过程,选取的代表算法为 AES、DES、IDEA、SAFER + 、SMS4 和 FOX64 算法,并分别在各规模结构下进行了 CBC/ECB 加密性能的评测. 如图 6 所示,不同算法的性能变化规律相似,规模总数越大,其加密性能越高;相

同规模总数时,簇内 BANK 数量 n 越大,其加密性能越高.

由图6可知,在BANK规模总数确定的情况下,1×4、1×8、1×16和2×16四种规模的性能和资源利用率较高,在实际应用开发中,应当优先考虑这四种规模结构.此外,为平衡资源利用与性能提升的关系,在FDP的实际开发应用中,各维度并行性的开发优先如下:ISP>IDP>ADP>ASP.

表3 性能指标分析对比

架构	是否重构	工艺(nm)	面积(mm^2)	分组并行数	频率(MHz)	CBC加密模式		ECB加密模式	
						吞吐率(Gbps)	面积能效比(Gbps/mm^2)	吞吐率(Gbps)	面积能效比(Gbps/mm^2)
1×4	Y	65	0.68	1	500	1.00	1.47	1.51	2.21
2×4	Y	65	1.25	2	500	1.00	0.80	3.01	2.41
4×4	Y	65	2.40	4	500	1.00	0.42	6.02	2.51
8×4	Y	65	4.69	8	500	1.00	0.21	12.05	2.59
1×8	Y	65	1.20	2	500	1.52	1.27	2.97	2.48
2×8	Y	65	2.29	4	500	1.52	0.66	5.95	2.60
4×8	Y	65	4.46	8	500	1.52	0.34	11.91	2.67
1×16	Y	65	2.23	3	500	2.06	0.92	6.00	2.69
2×16	Y	65	4.35	6	500	2.06	0.47	12.00	2.76
文献4	Y	65	4.28	4	400	—	—	2.16	0.50
文献5	N	65	6.63	1	1210	1.02	0.15	—	—
文献6	Y	180	14.9	4	180	—	—	0.79	0.05
文献7	Y	130	8.75	8	238	—	—	2.87	0.33
文献9	Y	180	13.38	6	243.9	0.47	0.04	2.8	0.21
文献10	Y	65	50	16	100	—	—	2.33	0.05
文献11	N	130	5.3	1	250	0.41	0.08	—	—

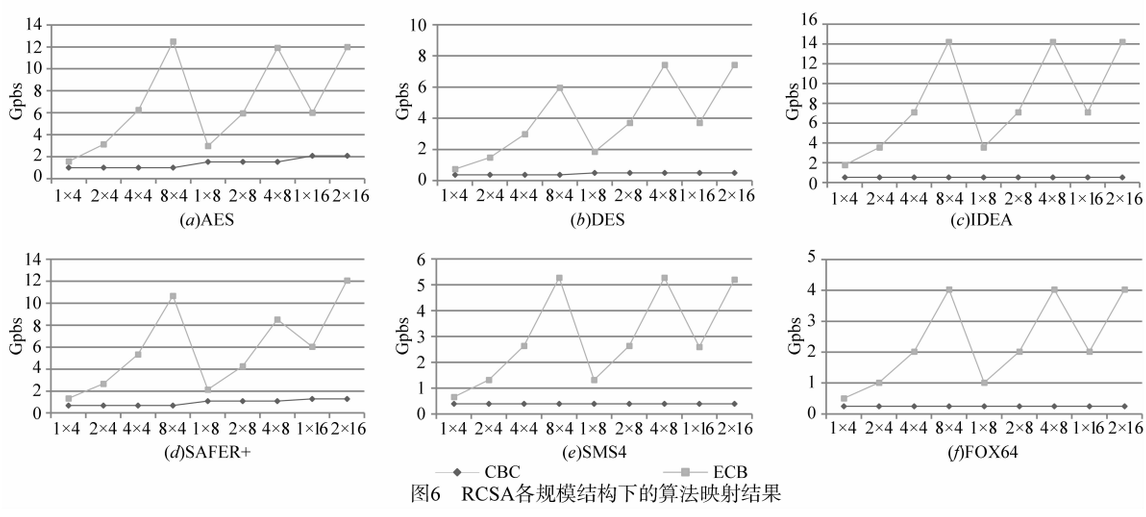


图6 RCSA各规模结构下的算法映射结果

5 结束语

为充分挖掘分组密码处理的并行性,将分组密码的并行性概括为四维度并行性FDP,并根据此提出了分组密码的四维度并行处理模型FDPM.以FDPM为依据,本文设计了一种面向分组密码的可重构流处理架构RCSA.在65nm CMOS工艺下对RCSA各规模结构进行了逻辑综合和功能仿真,并进行了SP、Feistel和L-M三种结构算法的映射和分析,详细分析了AES算法在各规模结构下的CBC/ECB加密性能.与其他密码处理器相比,该架构各规模结构均具有一定的性能优势,且面积能效比和功能单元利用率较高.由于实验条件和时间精力的限制,本文未对RCSA各规模架构的功耗进行仿真分析,下一步可根据需求建立相应的功耗模型并进行功耗仿真分析,完善该模型及架构体系.

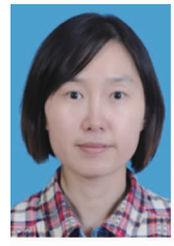
参考文献

[1] MICHAEL G,LILIAN B,GUY G,et al. Design and implementation of a muli-core crypto-processor for software defined radios[A]. Proceedings of International Symposium on Reconfigurable Computing Architectures,Tools and Ap-

plications[C]. New York:IEEE,2011. 29 – 40.
[2] 孟涛,戴紫彬. 分组密码处理器的可重构分簇式架构[J]. 电子与信息学报,2009,31(2):453 – 456.
MENG Tao,DAI Zi-bin. Reconfigurable clustered architecture of block cipher processor[J]. Journal of Electronics & Information Technology,2009,31(2):453 – 456. (in Chinese)
[3] GOKHAN S, DEREK C. Cryptoraptor: high throughput reconfigurable cryptographic processor[A]. Proceedings of IEEE/ACM International Conference on Computer-Aided Design[C]. New York:ACM,2014. 154 – 161.
[4] WANG Bo, LIU Leibo. A flexible and energy-efficient reconfigurable architecture for symmetric cipher processing[A]. Proceedings of IEEE International Symposium on Circuits and Systems[C]. New York: IEEE, 2015. 1182 – 1185.
[5] LIU B, BAAS B M. Parallel AES encryption engines for many-core processor arrays[J]. IEEE Transactions on Computers,2013,62(3):536 – 547.
[6] 杨晓辉,戴紫彬,张永福. 可重构分组密码处理结构模型研究与设计[J]. 计算机研究与发展,2009,46(6):962 – 967.
YANG Xiao-hui, DAI Zi-bin, ZHANG Yong-fu. Research and design of reconfigurable computing targeted at block cipher processing[J]. Journal of Computer Research and Development,2009,46(6):962 – 967. (in Chinese)
[7] 李校南,王雪瑞,戴紫彬,等. 可重构分簇式分组密码处理架构[J]. 计算机应用与软件,2014,31(1):315 – 318.
LI Xiao-nan, WANG Xue-rui, DAI Zi-bin, et al. Reconfigurable clustered block cipher processing architecture[J]. Computer Applications and Software,2014,31(1):315 – 318. (in Chinese)
[8] IQBAL N, SIDDIQUE M A, HENKEL J. RMOT: recursion in model order for task execution time estimation in a software pipeline[A]. Proceedings of IEEE Design, Automation & Test in Europe Conference & Exhibition Dresden[C]. New York:IEEE,2010. 953 – 956.
[9] 陈韬,罗兴国,李校南,等. 一种基于流处理框架的可重构分簇式分组密码处理结构模型[J]. 电子与信息学报,2014,36(12):3027 – 3034.
CHEN Tao, LUO Xing-guo, LI Xiao-nan, et al. An architecture of stream based reconfigurable clustered block cipher processing array[J]. Journal of Electronics & Information Technology,2014,36(12):3027 – 3034. (in Chinese)
[10] 郭岩松,刘雷波. 一种面向分组密码的粗粒度可重构阵列及 AES 算法映射[J]. 微电子学与计算机,2015,32(9):1 – 5.
GUO Yan-song, LIU Lei-bo. A block cipher oriented coarse-grained reconfigurable array and AES algorithm mapping[J]. Microelectronics & Computer,2015,32(9):1 – 5. (in Chinese)
[11] DIMITRIS T, ALEXANDROS S, DIONISIS P. CCproc: an efficient cryptography coprocessor[A]. Proceedings of 16th IFIP/IEEE International Conference on Very Large Scale Integration[C]. New York:IEEE,2008. 160 – 163.

作者简介

王寿成　男,1992 年生于甘肃金昌. 解放军信息工程大学硕士研究生. 从事可重构计算、信息安全方面的有关研究.
E-mail:jeremy_419@163.com



李功丽　女,1981 年生于河南信阳. 博士研究生,主要从事信息安全、专用芯片设计等方面的研究工作.