学号：131199

东 南 大 学

学术型研究生学位论文开题报告
及 论 文 工 作 实 施 计 划

院（系、所） 电子科学与工程学院

学科（专 业） 微电子学与固体物理学

研究生姓名 李小泉

学科门类与学位级别 工学硕士

导师姓名 孙伟锋

入学年月 2013 年 9 月

开题报告日期 2015 年 11 月 23 日

东南大学研究生院

填 表 须 知

1、 论文开题报告由研究生本人向审议小组报告并听取意见后，由研究生本人填写此表。

2、 论文开题报告填写完成后，必须经导师审批，通过后方能提交。

3、博士生应在第四学期内、硕士生应在第三学期内完成此开题报告。开题报告经研究生秘书在网上审核确认（硕士生至少半年、博士生至少一年）后方可申请答辩。

4、研究生开题前应填写查新报告。查新报告对理、工、医、管等学科博士生作为必要环节。博士生查新工作可委托图书馆负责，也可在完成网络文献检索类研究生课程的学习或参加学校组织的网络文献检索培训后，自行组织查新检索，自行组织查新需要详细文献查新述评作为附件。自行查新报告须经导师审查后由开题报告审核专家组审核签字（或盖章）。硕士生和文科博士生开题查新参考上述办法，不作硬性要求。

5、本表一式两份，一份研究生自留放入本人“研究生档案材料袋”；一份由院（系、所）保存并归入院（系、所）研究生教学档案。

6、学科门类与学位级别指的是工学（或理学等）博士、硕士。

7、本表下载区：http://seugs.seu.edu.cn/s/97/t/1707/aa/b8/info43704.htm 。本表电子文档打印时用 A4 纸张，格式不变，内容较多可以加页。

一、学位论文开题报告

<table>
<tr><td>论 文
题 目</td><td colspan="9">面向分组加密算法的可重构阵列处理单元的优化与设计</td></tr>
<tr><td>研 究
方 向</td><td colspan="9">可重构计算</td></tr>
<tr><td rowspan="2">题 目
来 源</td><td>国家</td><td>部委</td><td>省</td><td>市</td><td>厂、矿</td><td>自
选</td><td>有无合
同</td><td>经费数</td><td>备注</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>√</td><td>无</td><td></td><td></td></tr>
<tr><td rowspan="2">题 目
类 型</td><td>基础
研究</td><td>应用
研究</td><td>综合
研究</td><td>其它</td><td colspan="5"></td></tr>
<tr><td></td><td></td><td>√</td><td></td><td colspan="5"></td></tr>
<tr><td>论文开题前上网检索情况（硕士生可不作要求；理、工、医、管学科博士生应填写并附查新报告）</td><td colspan="9"></td></tr>
</table>

开题报告内容

（包括：立题依据及价值、研究内容及方法、可行性分析、预期的成果、预计的困难及解决办法）

一、立题依据及价值

1. 课题背景

随着计算机技术和通信技术的发展，信息安全问题逐渐成为人们关注的社会问题，密码技术是保证信息的机密性、安全性和可用性等安全要求的基本手段。密码算法是现代安全应用的基础，也是信息系统安全性的根本所在，实现高效灵活的密码算法是高性能信息系统的重要指标和根本保障，因此也成为信息安全领域的重要课题。

密码算法应用常常需要处理较大的信息量，或具有较大的计算强度，通用处理器常常不能满足其速度要求，安全性也不如专用硬件，因此国内外对密码处理专用硬件的研究和开发十分活跃。密码算法的主要硬件加速方式如表 1 所示，可分为三类：特定密码算法 ASIC、密码算法处理器和可重构密码处理结构。

表 1 密码算法实现方案对比

实现方案		优点	缺点
特定密码算法（ASIC）		消耗资源少，吞吐率高	不同的算法需要重新设计
密码算法处理器		配置依托于主处理器，易于编程，结构简单	算法支持有限
可重构密码处理结构	指令驱动型	可实现自动化配置，架构简单	指令逻辑复杂，占用大部分的周期
	数据驱动型	适合处理器阵列架构，更高的吞吐率	配置自动化困难

随着移动互联网的飞速发展，对系统安全性的要求也越来越迫切，保障系统安全所需投入的处理资源将越来越多，安全应用的范围也会越来越广，密码算法与可重构技术的结合，可以满足性能和安全方面的需求，具体地，可重 1，构密码系统架构在这一应用领域的优势体现在以下几点[1]：

（1）可重构系统的计算能力能够满足密码算法的性能需求，结构能够根据特定的密钥定制硬件，使算法执行更加高效；

（2）可重构系统可以根据实际需求改变硬件实现的算法，使同一硬件能够高效地支持种类繁多的算法，具有很大的灵活性；

（3）可重构系统具有扩展性，能够适应不断被提出新的更安全的算法，同时支持随时修改密钥，满足某些特殊情况下的白片需求。

2. 国内外研究现状

目前对密码算法在可重构阵列上的实现的研究已经有很多基础，很多论文和研究机构对这一个课题都进行了比较深入的探索。比较著名的几个架构和各自的特点如下：

COBRA[2][3][4]是一款面向对称密钥算法提出的指令级分布式可重构处理器，通过多达 40 余种对称密钥算法的映射实现进行验证，AES 算法的实现面积效率为 0.216Gbps/Mgates。为了提供充分的灵活性和并行计算能力，COBRA 在每个计算单元中实现了所有需要支持的算子，将所有需要的功能放在一条串行的路径上，通过配置选择功能开关，其优点是可以有更多的功能级联，在一个 PE 里面可以做更多的工作，缺点是单个 PE 的延迟很大，而且功能串行的需求在不同的算法中有不同的表现很难兼顾所有算法。在迭代架构中有优势但是不适合阵列的多级流水架构。

Celator[5]是由艾克斯—马赛大学研发的面向分组密码算法和哈希函数的可重构架构，其计算阵列基于脉动结构设计，采用二维互联结构完成计算单元间的数据传输，每个计算单元支持逻辑操作和算术操作，通过有限状态机控制计算阵列的数据访问和计算操作，有效控制了整体架构的硬件资源开销，DES 算法的实现面积效率为 0.25Gbps/mm2。该架构将所有需要的功能并行地放在不同的路径上，通过配置选择某一条路径完成某一个功能，优点是在保证 PE 功能完整的同时可以使 PE 的主频变得很高，缺点是 PE 中某一时刻只有一个功能单元在工作，电路利用率低。

Cyptoraptor[8][9]是由德州大学奥斯汀分校研发的一款高性能、低功耗、高灵活的密码处理器，面向分组密码、流密码、哈希函数三类百余种对称密钥算法进行架构探索，是目前研究支持算法最多的一款，DES 和 AES 算法的实现面积效率分别为 6.75Gbps/mm2 和 20.25Gbps/mm2。该架构通过对不同的模块进行延迟分析，结合算法的功能特征，功能模块先串行组合再并行组合。优点是平衡不同的功能单元的延迟，提供了简单功能单元的串行并且和功能并行结构具有相近的高主频，不足的地方在于功能串行组合是算法相关的，不能兼顾所有算法。

国内研究机构也在面向密码算法的可重构处理器方面积累了一定的研究，主要集中于国防科技大学和解放军信息工程大学。国防科技大学通过分析架构可编程性和数据流计算特性与控制逻辑属性的关系，提出了一种基于可编程数据流计算的体系结构框架 ProDFA [9]，解放军理工大学提出了 RCPA[10]架构，这两种架构 PE 内部的功能模块的串并连接可以通过配置进行动态组合，优点是 PE 内部各个功能单元的利用率提高，可以在一个 PE 里实现更多的功能，缺点是更加复杂的 PE 内部互连、配置。增加了面积和延迟。与串行的问题一样，如果不插寄存器那么 PE 的延迟就是所有功能单元的和，如果插寄存器解决延迟就要引入多周期。

表 2 不同架构中 PE 方案对比

PE 种类	功能灵活性	延迟	硬件开销	硬件利用率
功能串行	高	高	高	很低
连接可配置	很高	很高	很高	高
功能并行	无	低	高	很低
串并混合	低	低	较高	低
改进的功能单元串并混合	高	低	低	高

综上所述，对不同的架构中 PE 设计方案进行了归纳总结，如表 2，在归类总结了已有架构中对 PE 的设计方案和每种 PE 各自的优劣后提出了新的设计要求。引入基本的功能串并组合方案，在这个基础上对原有的方案进

行改进，通过提高 PE 的功能灵活性来提升硬件利用率，减少硬件开销，使架构的性能面积比得到提升。

二、研究内容及方法

1. 研究基础

现有架构的 PE 设计方案主要有以下几种方案，如表 3 所示：

表 3 现有的几种 PE 结构及其优缺点分析

方案	方案特征	文献	优点	缺点
功能单元串行设计	将所有需要的功能放在一条串行的路径上，通过配置选择功能开关。	[2][3][4]	可以有更多的功能级联，在一个 PE 里面可以做更多的工作。	单个 PE 的延迟很大，而且功能串行的需求在不同的算法中有不同的表现很难兼顾所有算法。在迭代架构中有优势但是不适合阵列的多级流水架构。
功能单元的内部连接可按需配置设计	PE 内部的功能模块的串并连接可以通过配置进行动态组合。	[10]	PE 内部各个功能单元的利用率提高，可以在一个 PE 里实现更多的功能。	更加复杂的 PE 内部互连、配置。增加了面积和延迟。与串行的问题一样，如果不插寄存器那么 PE 的延迟就是所有功能单元的和，如果插寄存器解决延迟就要引入多周期。
功能单元并行设计	将所有需要的功能并行地放在不同的路径上，通过配置选择某一条路径完成某一个功能。	[6][7]	在保证 PE 功能完整的同时可以使 PE 的主频变得很高。	PE 中某一时刻只有一个功能单元在工作，电路利用率低。
功能单元串行和并行混合设计	通过对不同的模块进行延迟分析，结合算法的功能特征，功能模块先串行组合再并行组合。	[4][5][6][9]	平衡不同的功能单元的延迟，提供了简单功能单元的串行并且和功能并行结构具有相近的高主频	功能串行是算法相关的，不能兼顾所有算法

本文在对现有的架构进行分析时发现了如下几个设计缺陷：

- 经验的、简单的功能组合

现有架构的功能组合的标准是在延迟不够的并行路径上加上一个延迟小的功能来平衡不同路径的延迟，比较多的情况是加抑或逻辑[8]、字节置换[3][4]和移位[3][8]。这些延迟平衡的组合缺少足够的算法分析，很多时候这些组合并不能被很多的算法所使用。

- 阵列采用同构 PE，功能单一，利用率低

算法的一轮中的操作是固定的，而且一般算法的一轮会被映射到架构中的多行阵列中，同构阵列中为了通用性必须为所有的 PE 设计算法所需的所有功能。同构的 PE 为实现算法，在同一个 PE 里面堆砌算法的所有功能，照成很大的资源浪费。只需要有正确位置的某一行 PE 提供算法所需的功能，关键问题在于在哪个位置提供那个功能。

- 功能组合的依据来自一轮顺序操作，比较局限

现有的架构虽然对目标算法的轮函数进行了分析，提炼轮函数中的各种操作组合，但是对于大多数分组加密算法来说，真正有价值的操作组合在每轮的首尾的位置，而中间的位置作为关键路径

不利于组合更多的操作，只通过顺序分析算法的单轮无法获取这些组合关系。

2. 研究内容

在分析了现有架构中的设计缺陷后，本文针对这些缺陷提出了对应点改进方案，如图 1 所示：

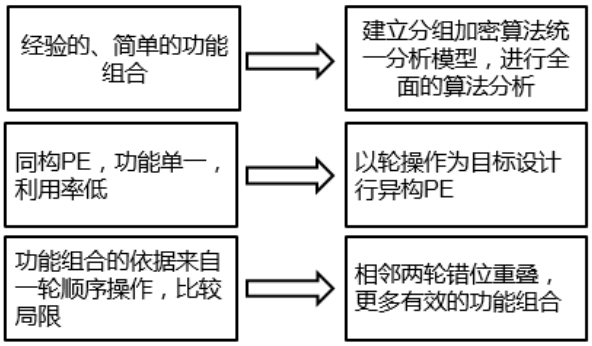

图 1 针对已有 PE 结构存在的问题提出了对应的改进方案

3. 设计方案

整体方案分为算法建模、功能单元电路参数获取、算法聚类分析、PE 架构探索和设计验证，整体流程如下图 2 所示：

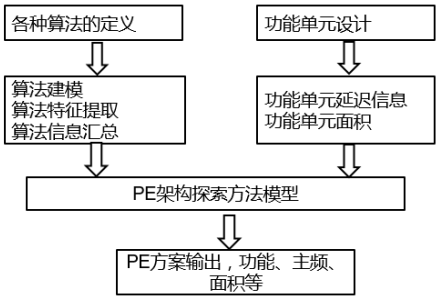

图 2 整体流程

- 算法建模

算法的流程图是一个有向的无环图，就是一个 AOV 网络，而为了获取关键路径，把 AOV 网络转换成 AOE 网络。点表示阶段，边表示具体的算子，边上的权表示算子的延迟，如图 3 所示：

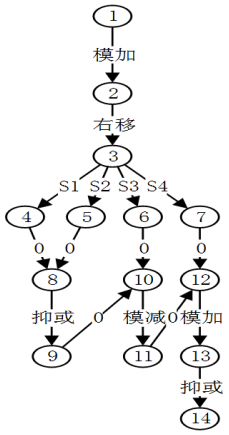

图 3 算法图建模

- 算子电路参数获取

PE 功能子电路是组成 PE 的基本元素，这些基本元素决定了 PE 的功能、面积、延时。PE 结

构探索的前提是先获取这些基本功能单元的电路参数，为 PE 方案的延迟和面积评估提供数据支持。

- 算法聚类分析

关键路劲的长度决定了在电路上完成这个算法所需要的资源，这个信息在后面的架构探索中非常关键，它决定了架构的中异构 PE 的种类，以及整个阵列的大小，因此在设计之初先对这些算法按照关键路径的长度进行聚类分析；试验中选择了 k-means 聚类算法，这是一个很常见的聚类算法，它算法采用误差平方准则函数作为聚类收敛依据，也就是说它的聚点标志了一个关键路径。

- PE 探索方案

PE 探索方案分为初始化、关键路径切分、切分方案评估、切分最优选择等过程，整体流程如图 4 所示：

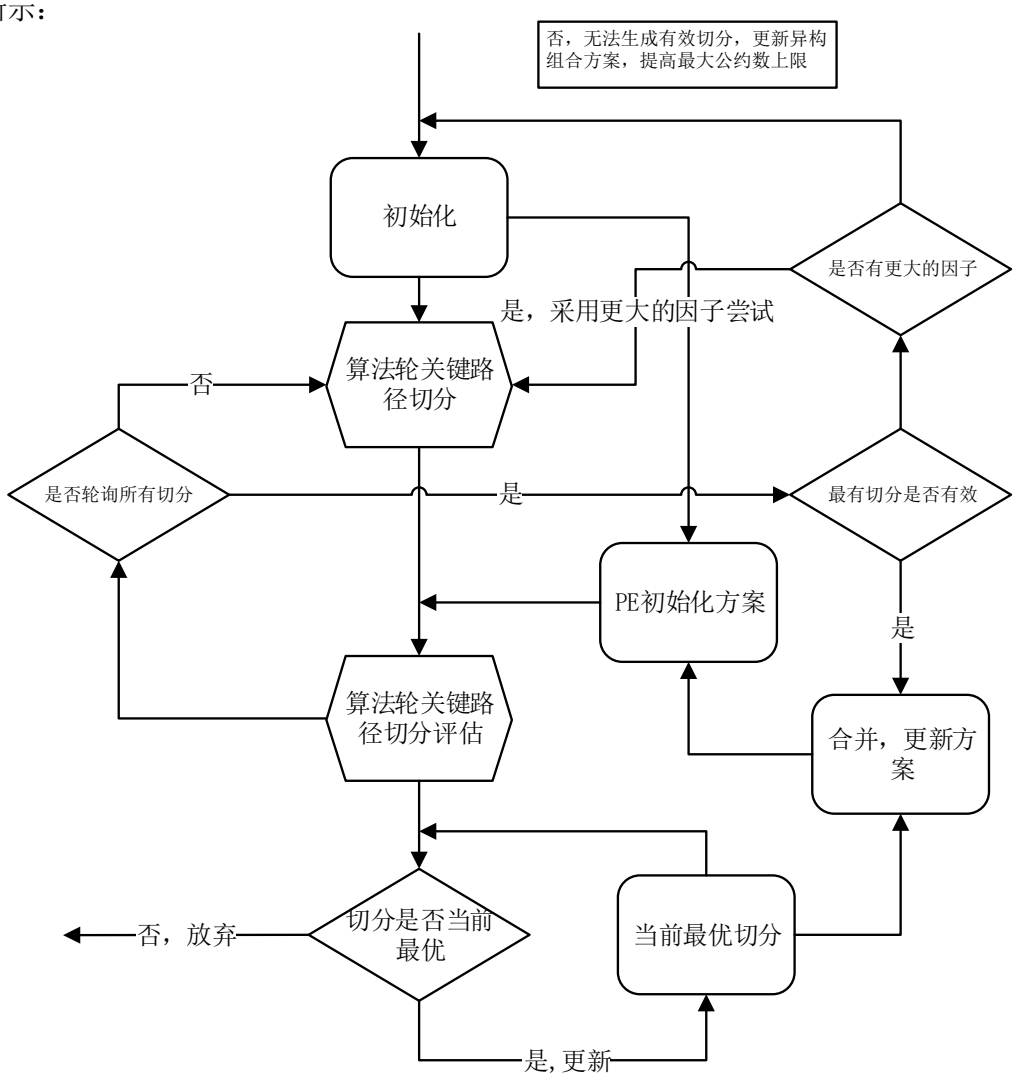

图 4 PE 探索算法流程

1) 初始化

涉及到几个全局参数的初始化，PE 延迟：由算法中最大延迟功能单元和输入性能约束共同决定，如果输入约束满足最大功能单元延迟则选择输入延迟，否则选择最大功能单元延迟。PE 异构组合方案：备选方案（1，2，3，6）和（1，2，4，8）。阵列大小：由算法集合

中最长关键路径算法、PE 异构组合和 PE 延迟共同决定。每种算法的 PE 异构数：由关键路径延迟、和 PE 异构组合方案共同决定

2) 关键路径切分

关键路径切分的分组数在算法支持的 PE 组合方案中选择。比如分组中最小的为 2，最大 PE 数为 6，那么先将关键路径进行两段式切分，将关键路径分为连续的两部分，复杂度 O(N)。然后将关键路径进行三段式切分，将关机按路径分成三部分，复杂度为 O(N^2)。最后将关键路径进行六段式切分，将关机按路径分成六部分，复杂度为 O(N^5)。一个划分是否有效的判断依据是：在当前划分下不会增加初始划分方案的延迟，如果所有的划分都无法满足这个条件，那么必须申请更多的资源。

3) 切分方案评估

准则：评估一种有效切分在初始架构下的面积消耗。对于新的算法，如果能够完全重用初始架构中的单元，那么面积消耗为 0，如果没有任何重用，那么面积消耗将是算法中所有算子单元面积和。方案：架构是由一个由功能单元构成的单向图（或树）的集合，在进行面积评估时会在这个图集合中进行后继遍历查找，如果这个图集合中能够提供所需功能，那么不用增加单元，如果不能提供，那么必须加入新的功能单元，并且需要评估加入新的功能后对整个 PE 的延迟影响，如果满足延迟要求则将功能单元合并到初始架构中去。

4) 切分最优：

轮询算法的所有有效切分方案，取面积评估最小的一种方案作为算法的最终切分方案，并更新初始架构。切分+评估的过程就是寻找面积重用最高的切分方案的过程，最优解就是 PE 在支持这种新的算法时所需要增加最少功能的情况。面积重用的依据是不同的算法中的相似的功能组合，这些相似的组合为 PE 的功能选择及组合提供了依据。架构的更新与算法切分评估是同步的，最佳的切分方案最终决定新的算法对架构的影响，同时这个切分方案会被保存，为后面的算法映射提供依据。

- 验证模型

验证的方法如图 5 所示：

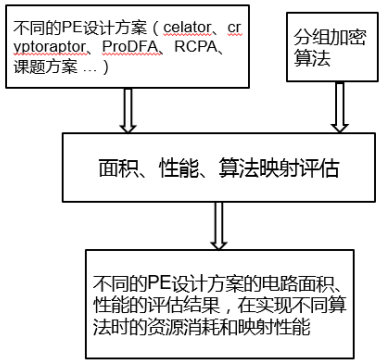

图 5 验证流程

前提：1.剔除架构中存储、配置等其它干扰因素，只考虑 PE。2.在进行对比时会与其它架构中的算法支持保持一致，PE 所包含的功能单元种类相同。增加映射算法的种类，对于论文中映射了的算法采用论文中映射的结果，对于支持的但没有映射的算法进行补充映

射，目前已经选择的算法：AES[15]、DES[16]、IDEA[17-18]、BLOWFISH[19]、CAMELLIA[20-21]、CAST128[22]、GOST[23]、RC5[24]、SEED[25]、TWOFISH[26]、SM4[27]、RC6[28]、SERPENT[29]、TEA、XTEA[30]、SKIPJECT[11-14]。

表 4 不同验证的前提场景

架构	算法支持	功能单元种类
Celator [13]	功能单元支持下的所有算法（8/16），论文中映射了AES和DES	与、或、查找表、非、抑或、直通、模加、移位
ProDFA [12]	功能单元支持下的所有算法（15/16），论文中映射了DES、AES、IDEA、Twofish、RC6	与、或、非、异或、模加减、查找表、移位、循环移位、模乘、置换/扩展
Cryptorapt or [3][4]	功能单元支持下的所有算法（14/16），论文中映射了AES Blowfish, Camellia, CAST-128, DES, GOST, Kasumi, RC5, SEED, and Twofish	与、或、非、异或、非（与、或、非、异或）、直通、模加减、查找表、移位、循环移位、置换/扩展、crossbar互联
COBRA [1][2]	功能单元支持下的所有算法（14/16），论文中映射了RC6、Rijndael、Serpent	与、或、非、异或、模加减、查找表、移位、循环移位、有限域乘法、模乘
RCPA [14]	功能单元支持下的所有算法（16/16），论文中映射了AES、DES	与、或、非、异或、模加减、查找表、移位、循环移位、有限域乘法、模乘、置换/扩展

评估方案：原始架构的映射数据来自论文中的算法映射结果，新架构的数据来自对算法在新架构中的映射评估结果。

指标：对新旧架构 PE 的主频、面积、轮函数周期数、轮函数面积、映射吞吐率、性能面积比进行了对比，对比论文 PE 结构的性能面积比提高 30%以上，具体如表 5 所示。

表 5 指标

架构	PE类型	主频 (MHz)	面积 (gates)	算法映射评估					
				算法	算法轮周期数 (cycles)	算法轮面积(gates)	算法映射面积 (gates)	算法映射性能(Gbps)	性能/面积 (0.01Mbps/gate)
Celator [13]	功能并行	917.4	7185	AES	27	193995	1939950	117.427	6.053
				DES	16	114960	1839360	58.7136	3.192
				算法3					
方案1				AES					> 7.869
				DES					> 4.15
				算法3					
ProDFA [12]	功能并行	769.2	82765	AES	8	662120	6621200	98.4576	1.487
				DES	3	248295	3972720	49.2288	1.239
				算法3					
方案2				AES					> 1.933
				DES					> 1.611
				算法3					
Cryptoraptor [3][4]	功能串并组合	609.8	109975	AES	2	219950	2199500	78.0544	3.549
				DES	3	329925	5278800	39.0272	0.739
				算法3					
方案3				AES					> 4.613
				DES					> 0.961
				算法3					
COBRA [1][2]	功能串行	204.5	71198	AES	9	640782	6407820	26.176	0.409
				DES					
				算法3					
方案4				AES					> 0.531
				DES					
				算法3					
RCPA [14]	功能连接可不配置	296.7	74834	AES	5	374170	3741700	37.9776	1.015
				DES	6	449004	7184064	18.9888	0.264
				算法3					
方案5				AES					> 1.319
				DES					> 0.344
				算法3					

三、 可行性分析

本课题以学术上的论文为基础，分析了现有论文中的几个架构中的设计不足，并且提出了对应的解决方案。在方案实施上采用采用自动化的方法，对所有的目标算法进行了建模分析，在此基础上对这些算法进行迭代分析最终生成一个异构的 PE 架构。

四、 预期的成果

- 对新旧架构 PE 的主频、面积、轮函数周期数、轮函数面积、映射吞吐率、性能面积比进行了对比，对比论文 PE 结构的性能面积比提高 30%以上。
- 发表一篇高水平论文

五、 预计的困难及解决办法

- 算法建模
  建立一个统一的模型能够来描述所有的分组加密算法，并且提供算法在这个模型的基础上来对这些算法进行关键路径等信息的分析。
- 异构 PE 的迭代探索
  如何根据输入的算法模型和算子电路参数经过迭代最终收敛成一个 PE 架构，课题中采用初始化、关键路径切分、切分方案评估、切分最优选择等过程对整个方案进行切分，针对切分的几个过程分别进行了分析实现。

六、 参考文献

[1]王莉．密码算法的可重构系统实现研究[D]：[博士学位论文]．湖南：国防科学技术大学，2004

[2] Elbirt A J et al. “Instruction-Level Distributed Processing for Symmetric-Key Cryptography.” Parallel and Distributed Processing Symposium. 2003. Apr. 22, 2003. pp. 78-87.

[3] Elbirt, Adam J., and Christof Paar. "An instruction-level distributed processor for symmetric-key cryptography." Parallel and3 Distributed Systems, IEEE Transactions on 16.5 (2005): 468-480. [4] E. Ahmed and J. Rose, “The effect of LUT and clustersize on deep-submicron FPGA performance and density,” IEEE Trans. VLSI Syst., vol. 12, no. 3, pp. 288–298, Mar. 2004.

[4] LOMONACO, M. 2004. Cryptarray a scalable and reconfigurable architecture for cryptographic applications.Masters thesis, University of Central Florida.

[5] Sun, Kang, et al. "Design of a novel asynchronous reconfigurable architecture for cryptographic applications." Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on. Vol. 2. IEEE, 2006.

[6] 陈韬, 罗兴国, 李校南, & 李伟. (2014). 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. 电子与信息学报, 36, 12.

[7] 杨晓辉. (2007). 面向分组密码处理的可重构设计技术研究 (Doctoral dissertation, 硕士论文], 解放军信息工程大学).

[8] Chiou D. Cryptoraptor: high throughput reconfigurable cryptographic processor[C]//Proceedings of the 2014 IEEE/ACM International Conference on Computer Aided Design. IEEE Press, 2014: 154-161.

[9] Chiou D. Cryptoraptor: High Throughput Reconfigurable Cryptographic Pro cessor for Sy mmetric Key Encryption and Cryptographic Hash Functions [D]. The University of Texas at Austin 2014.

[9] Yan M, Yang Z, Liu L, et al. ProDFA: Accelerating Domain Applications with a Coarse-Grained Runtime Reconfigurable Architecture[C]// 2013 International Conference on Parallel and Distributed Systems. IEEE, 2012:834-839.

[10] Dai Z B, Yang X H, Ren Q, et al. The research and design of reconfigurable cipher processing architecture targeted at block cipher[C]// ASIC, 2007. ASICON '07. 7th International Conference on. IEEE, 2007:814-817.

[11] Buchty, Rainer, Nevin Heintze, and Dino Oliva. "Cryptonite-A Programmable Crypto Processor Architecture

for High-Bandwidth Applications." Organic and Pervasive Computing--ARCS 2004: International Conference on Architecture of Computing Systems, Augsburg, Germany, March 23-26, 2004, Proceedings. Vol. 2981. Springer Science & Business Media, 2004.

[12] Yan M, Yang Z, Liu L, et al. ProDFA: Accelerating Domain Applications with a Coarse-Grained Runtime Reconfigurable Architecture[C]// 2013 International Conference on Parallel and Distributed Systems. IEEE, 2012:834-839.

[13] Fronte D, Perez A, Payrat E. Celator: A Multi-algorithm Cryptographic Co-processor[C]// Reconfigurable Computing and FPGAs, 2008. ReConFig '08. International Conference on. IEEE, 2008:438-443.

[14] Dai Z B, Yang X H, Ren Q, et al. The research and design of reconfigurable cipher processing architecture targeted at block cipher[C]// ASIC, 2007. ASICON '07. 7th International Conference on. IEEE, 2007:814-817.

[15] M.D. Galanis, P. Kitsos, G. Kostopoulos, N. Sklavos, O. Koufopavlou, and C.E. Goutis.Comparison of the hardware architectures and FPGA implementations of stream ciphers.In Proceedings of the 11th IEEE International Conference on Electronics, Circuits andSystems, ICECS'04, pages 571–574, 2004.

[16] Berndt M Gammel, Rainer Göttfert, and Oliver Kniffler. The achterbahn stream cipher.Submission to eSTREAM, 2005.

[17] Praveen Gauravaram, Lars R Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schläffer, and Søren S Thomsen. Grøstl–a sha-3 candidate.Submission to NIST, 2008.

[18] Dianelous Georgoudis, Damian Leroux, and Billy Simon Chaves. The âĂIJfrogâĂİ encryption algorithm. NIST AES Proposal, 1998.

[19] Chih-Peng Fan and Jun-Kui Hwang. Implementations of high throughput sequentialand fully pipelined AES processors on FPGA. In International Symposium on IntelligentSignal Processing and Communication Systems. ISPACS'07, pages 353–356, 2007.

[20] Horst Feistel. Cryptography and computer privacy. Scientific american, 228:15–23,1973.

[21] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, TadayoshiKohno, Jon Callas, and Jesse Walker. The skein hash function family (2008). Submittedto SHA-3 Competition.

[22] Whitfield Diffie and George Ledin. Sms4 encryption algorithm for wireless networks.IACR Cryptology ePrint Archive, page 329, 2008.

[23] Hans Dobbertin. Ripemd with two-round compress function is not collision-free. Journal of Cryptology, 10(1):51–69, 1997.

[24] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. Ripemd-160: A strengthenedversion of ripemd. In Fast Software Encryption, pages 71–82. Springer, 1996.

[25] Donald Eastlake and Paul Jones. Us secure hash algorithm 1 (sha1), 2001.

[26] Patrik Ekdahl and Thomas Johansson. Snow-a new stream cipher. In Proceedings ofFirst Open NESSIE Workshop, KU-Leuven, 2000.

[27] Adam J. Elbirt. Reconfigurable computing for symmetric-key algorithms, 2002.

[28] A.J. Elbirt. Fast and efficient implementation of AES via instruction set extensions.In 21st International Conference on Advanced Information Networking and ApplicationsWorkshops, AINAW'07., volume 1, pages 396–403, 2007.

[29] A. Hodjat and I. Verbauwhede. Speed-area trade-off for 10 to 100 Gbits/s throughpuAES processor. In Conference Record of the Thirty-Seventh Asilomar Conference oSignals, Systems and Computers, volume 2, pages 2147–2150 Vol.2, 2003.

[30] A. Hodjat and I. Verbauwhede. A 21.54 Gbits/s fully pipelined AES processor on FPGAIn 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machine FCCM'04, pages 308–309, 2004.

研究生签名：

年　　月　　日

二、学位论文工作实施进度与安排

起讫 日期	工　作　内　容　和　要　求	备　注
2015.4~2015.7	算法选择，算法分析与建模	
2015.8	子电路设计与参数获取	
2015.9~10	算法聚类分析，PE 方案的收敛算法设计	
2015.11~12	设计验证	
2016.1~2016.3	毕业论文撰写及答辩	