AES Encryption and Decryption Algorithm for High-Speed Design FPGA-Based

ZHOU Yong-hong[1], SHAO Jin-xiang[2], XIAO Shun-wen[1], Tang Zheng-ming[1]
1. Physics and Electronic Information College, China West Normal University，Nan Chong 637002, China
2. Da Qing Oilfield Co., Ltd. 7th Oil Production Plant，Da Qing 16357, China
Email:scnczyh@163.com

Abstract: For the widely used 32-bit data platform, implementation of AES-128 encryption / decryption algorithm designing on FPGA. Encryption/decryption unit uses part of the external pipeline technology, using logical lock technology optimized S-BOX and key storage in the system optimization. With less system resources, in the 100MHz clock frequency, obtained 3200Mbit/s data throughput. Applies to clock frequency is not higher than 100MHz the mid-and low-end equipment.

Keywords: FPGA, AES, part of the external pipeline, logical lock, high-speed design

基于 FPGA 的 AES 加解密算法高速设计

周永宏[1] ，邵金祥[2]，肖顺文[1]，唐正明[1]
1. 西华师范大学物理与电子信息学院 四川 南充 637002
2. 大庆油田有限责任公司第七采油厂 黑龙江 大庆 163517
Email:scnczyh@163.com

【摘要】针对应用广泛的 32 位数据平台，用 FPGA 实现了密钥长度为 128 位的 AES 加解密算法。在加/解密单元采用了部分外部流水线技术，在系统优化中采用了逻辑锁技术，对 S 盒和密钥存储进行了优化。以较少的系统资源，在 100MHz 时钟频率下，获得了 3200Mbit/s 的数据吞吐量。适用于时钟频率不高于 100MHz 的中低端设备。

【关键词】FPGA，AES，部分外部流水线，逻辑锁，高速设计

1 引言

2000年10月2日，美国国家标准和技术研究所宣布采用Rijndael算法作为高级加密标准，并于2002年5月26日正式生效，取代了DES 成为新一代的主流对称加密算法， AES将在今后很长的一段时间内在信息安全中扮演重要角色。因此对AES实现的研究成为国内外的热点，已在信息安全领域得到广泛应用。AES算法的FPGA实现方法已有不少讨论，详见文献[1][2][3]，主要是通过提高系统时钟频率及以较大的电路面积和功耗来换取较高的数据吞吐率，并且实现AES的硬件平台数据宽度基本上都是128位。本文针对现在应用广泛的32位数据平台，通过对AES的核心——加解密单元的FPGA实现方法进行改进，采用部分外部流水线技术来实现，并在系统优化过程中采用逻辑锁技术，从而在电路面积与数据吞吐率这两个关键指标中找到平衡点。本系统实现的是AES_128，适用于时钟频率不高于100MHz的中低端设备。

基金项目：西华师范大学科研启动基金（05B019）四川省教育厅科研基金重点项目（07ZA127）
Fund :Scientific Research Foundation of China West Normal University (05B019) ; Research Foundation of Sichuan Education Department projects (07ZA127)

2 AES算法的整体结构设计

从文献[4]中对AES算法的描述可以知道，算法的结构很清晰，因此，选择设计方案进而构造出整体系统结构框架成为了设计的关键。AES算法硬件实现的基本逻辑框架如下图1所示。它包括了如下单元：

1. 加解密单元，用于对输入数据块进行加密/解密，是设计的核心。

2. 密钥扩展单元，基于初始密钥计算、存储一系列中间子密钥。

3. 输入接口，用于接收输入数据块以及初始密钥，并进行串并转换。

4. 输出接口，用于暂存加密/解密单元产生的结

果，并进行并串转换以送给外部逻辑。

5. 控制单元，用于产生控制信号以协调所有其他单元的运作。

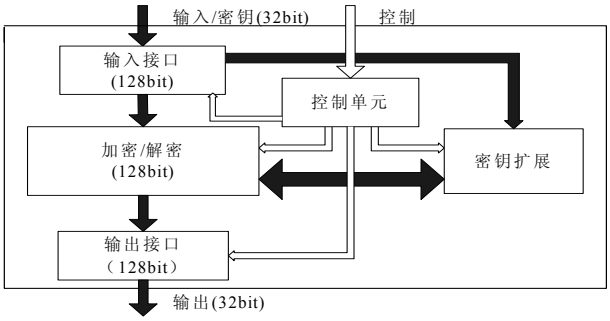

Figure 1. The logical structure diagram of AES algorithm hardware implementation

图 1. AES 算法硬件实现逻辑结构示意图

从外部经输入接口模块读入密钥和明文，并在每完成一个分组输入后产生一个启动脉冲，触发控制模块进行控制信号的输出，来辅助加解密模块和密钥扩展模块进行工作，当分组加密完成后，生成输出启动脉冲，通过输出模块依次将数据输出。

3 加解密部件硬件实现方式

AES 加解密过程中的轮运算特点使得其 FPGA 实现有多种方式可以选择[2]。（1）基本结构，如图 2 所示。在这种结构中，只有一个寄存器和多路开关，以及与算法中的一轮迭代对应的组合逻辑电路。输入的数据块通过多路开关送入电路，并被存放在寄存器中，在接下来的一个时钟内完成一轮加密计算，计算的结果又通过多路开关反馈回电路，并存放在寄存器中。这样加密一组数据所需的时钟数等于加密的轮数。（2）循环展开结构，如图 3 所示。循环展开结构和基本结构的唯一区别就是组合逻辑电路部分实现的是 K 轮加密而不是 1 轮。在循环展开结构中，加密一组数据所需时钟数为 1。在这种结构中，速度的提高是以空间的增加为代价的。加密/解密部件所用的总空间差不多与 K 成正比，因此只有每轮循环占用空间小，同时每轮时延与多路开关延时、寄存器的时钟输出延时和信号建立时间之和相比也较小的算法才适合。（3）内部循环流水线结构，如图 4 所示。它由基本结构发展而来，具体做法是在每一个轮函数中插入寄存器，将一轮运算分成多个步骤，每个时钟完成一个步骤，这种方式在多篇文献中被讨论并使用，其优点是可以通过并行执行这些步骤，从而提高算法运行的时钟频率。（4）外部流水线结构，如图 5 所示。外部流水线结构由循环展开结构发展而来。具体方法是在组合电路中与每一轮加密对应的部件之间都插入额外的寄存器，因此，它的流水线级数为加密的轮数 K。

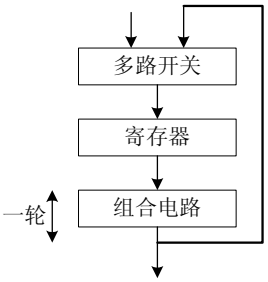

Figure 2 Basic structure.

图 2 基本结构

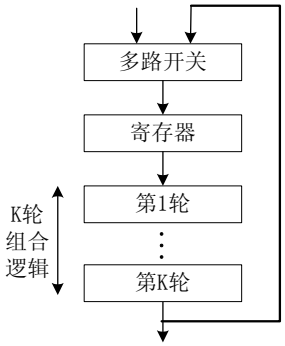

Figure 3. Loop deployable structure.

图 3 循环展开结构

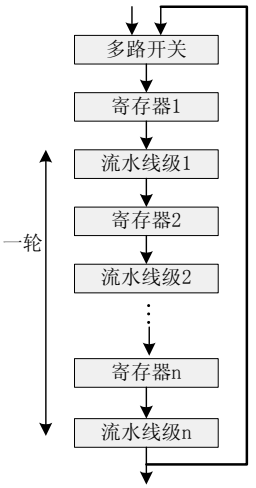

Figure 4 Internal pipeline structure

图 4 内部流水线结构

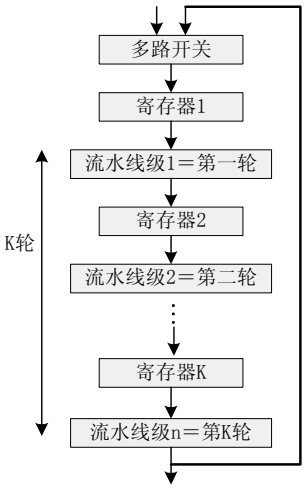

Figure 5 External pipeline structure

图 5 外部流水线结构

3.1 改进的加/解密部件实现方式

针对 AES 算法，基本结构虽然节省电路面积，但达不到高速设计的目的，至少要 10 个时钟后才能完成一组数据的加密，当要求速度不是很高的情况下这是最佳选择方案；循环展开结构是要将 10 轮迭代运算集成到一个组合逻辑中，这在时序上很难满足要求，延时大大超出了时钟周期；AES 的轮变换中，只有字节替代和列变换稍复杂些，但也没有乘除等复杂运算，完全可以在一个时钟周期内完成，所以并不适合内部循环流水线的结构；外部流水线结构唯一不足是消耗电路面积太大，而且当数据宽度是 128 位时才能使流

978-1-935068-31-0 © 2010 SciRes.

水线发挥其最大的优势。综合以上的结构和分析，在本设计中采用了一种作者称之为部分外部流水线的结构，如图 6 所示，它是基本结构同外部流水线结构的结合。也就是说，在一级流水线中进行多轮运算，每轮运算需要一个时钟周期。以基本结构进行 3 轮运算为例，在基本结构之间通过寄存器连接起来形成流水线结构，基本结构中输出是带有寄存器的输出，就不需要额外的连接基本结构的寄存器了。外部的流水线使得可以在小于 10 个时钟周期内完成一组数据的处理，提高了速度，内部的基本结构使得资源共享，节省了资源消耗，从而在面积和速度这两个指标中得到了平衡。

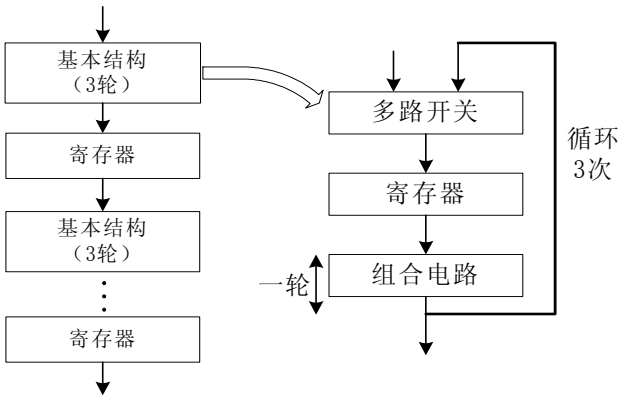

Figure 6 Part of the external pipeline structure
图6 部分外部流水线结构

3.2 流水线设计

根据上面的分析，在加解密单元采用部分外部流水线结构实现，这里就涉及到流水线分级的问题。对于流水线而言，因为存在滞后等待期，所以分级并非越多越好；同样由于分级越少，可并行执行的步骤也越少，总体速度会降低，所以也并非越少越好。本文针对 32 位数据平台，文献[5]将流水线分成两级，笔者认为不太合适，因 AES 是 128 位的，串并转换需要 4 个时钟，故将 10 轮运算分为四级流水线。由于加解密的 10 轮运算中前 9 轮是一样的，最后 1 轮少了列混合，按照前面的设计，前三级流水线的基本结构均进行 3 轮运算，最后一级流水线进行 1 轮运算，这样一来在得到第一个加解密结果之后，每 4 个时钟就能产生一个加解密结果。

4 系统优化

在 FPGA 开发中会有这样的现象：原来在硬件测试上十分成功的 FPGA 设计，在源代码并没有大改变的情况下，仅仅是增加了一部分电路描述，或改变了端口信号的引脚锁定位置，结果在综合适配后，原来设计的硬件性能将大为下降，甚至无法正常工作。比较修改前后的 Floorplan(平面布置图)可以发现芯片内部资源的使用情况发生了巨大变化。这表明，即使对原设计做极小的改变，都会使适配器对原设计的布线(Routing)和布局(Placing)策略做大幅改变和调整。FPGA 设计中的综合和适配是由软件自动完成的，当设计规模比较大时，人为很难直接介入布局/布线的优化，Quartus II 的逻辑锁技术很好的解决了上面的问题。

在系统开发中，应用逻辑锁技术(LogicLock)可以优化设计，合理分配硬件资源，并有利于提高系统的工作速度。逻辑锁定就是在适配中对逻辑布局进行特定的约束。设计者可以在编译前为目标芯片设定一个或数个适当大小的区域，并指令适配器将指定的设计电路模块放置在该区域中。使用逻辑锁技术进行优化后，布局相对合理和集中，有利于布线，有利于系统的稳定和性能的提高，而在未使用逻辑锁定得到的适配结果中，虽然可以进行一定的约束，但总体上布局很凌乱，可以通过查看 Floorplan 得到结果。

逻辑锁区域的特征主要有两个标志：“大小”和“位置”。“大小”是指此区域的高和宽，单位是逻辑单元；而“位置”是指在目标器件中锁定区域在 Floorplan 图上所处的位置。区域的位置状态有两种：“锁定”状态和“浮动”状态。区域是浮动的，则由 Quartus II 在优化过程中确定具体位置所在，如果区域的大小设定为自动，就将在优化中自动调整大小，以便设计模块正好放进此区域。表 1 给出了详细说明。

本系统中 S 盒是查找表结构的，可以用 FPGA 内部的存储资源实现，所以将其位置锁定在存储区域，大小设定为自动，密钥存储也锁定在存储区域，大小自动，其他部分均设定为位置浮动、大小自动，以增强适配中的灵活性。另外在适配的约束中，选择速度优先模式，最终得到的优化结果如下。都是针对中低端设备，本设计相比文献[6]中介绍的 20MHz 时钟频率下得到 128Mbit/s 数据吞吐量有比较优势。

Table 1 Lock Regional partition type
表 1 锁定区域分区类型

位置	大小	说明
浮动	自动	这是逻辑锁定约束条件中最灵活的一种，在适配中 Quartus II 自动根据优化情况确定最终的锁定区域的大小和位置。
浮动	固定	这种方式中是假设定义的锁定区域的大小是合理的，否则不利于逻辑资源的有效利用。
锁定	固定	这种约束条件具有最小的灵活性，由设计者确定锁定区域的一切具体情况。

Table 2 AES optimization results
表 2 AES 优化结果

数据宽度	资源利用			处理速度		
	逻辑单元(LE)	存储单元(bit)	功耗(mW)	时钟频率(MHz)	时延(ns)	吞吐量(Mbit/s)
32	8123	274592	1034.66	100	40	3200

5 系统测试

5.1 仿真测试

在 Quartus II 软件中，利用波形输入法进行仿真。AES 加密系统的仿真相对其他设计的仿真，最大的优势就是在输入密钥相同的情况下，将明文加密仿真后得到的结果再次作为输入，并进行解密仿真，如果这时得到的结果同加密时的输入明文是一致的话，就能说明该系统的工作是正常的。

现将不同密钥下，将测试所得的明文、密文对给出，见表 3，表中数字均是 16 进制表示，密钥分别有三组，即全“0”，全“1”和 AES 标准测试密钥。测试数据是全“0”，加密后的密文以及标准测试数据。在 32 位数据宽度下，在时钟频率分别是 20MHz、25MHz、50MHz、80MHz 和 100MHz 的情况下，测试结果均是正确的。

Table 3 Test plaintext ciphertext
表 3 测试明文密文对

密钥：00000000000000000000000000000000	
明文	密文
00000000000000000000000000000000	66E94BD4EF8A2C3B884CFA59CA342B2E
66E94BD4EF8A2C3B884CFA59CA342B2E	00000000000000000000000000000000
密钥：FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	
明文	密文
00000000000000000000000000000000	A1F6258C877D5FCD8964484538BFC92C
A1F6258C877D5FCD8964484538BFC92C	00000000000000000000000000000000
密钥：2B7E151628AED2A6ABF7158809CF4F3C	
明文	密文
3243F6A8885A308D313198A2E0370734	3925841D02DC09FBDC118597196A0B32
3925841D02DC09FBDC118597196A0B32	3243F6A8885A308D313198A2E0370734

然后，给出 32 位宽度 AES 加密系统的波形仿真结果,如图 7 所示。时钟频率是 50MHZ。

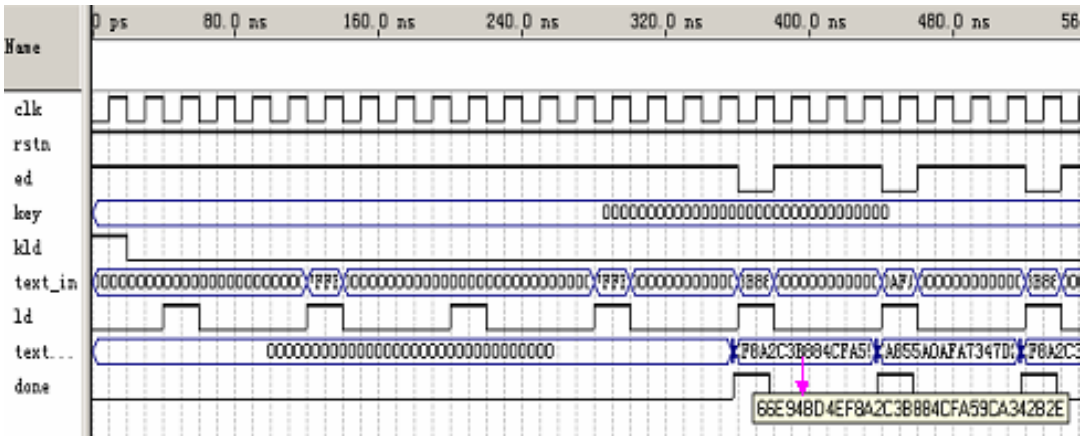

Figure 7 Simulation chart of encryption timing
图 7 加密时序仿真图

978-1-935068-31-0 © 2010 SciRes.

5.2片上测试

片上测试是借助开发板来完成的，开发板上的晶振、按钮、开关以及七段数码管方便用户控制和观察结果。作者使用的是Stratix EP1S25 DSP开发板，板上FPGA芯片是Stratix系列EP1S25F780C5，板上晶振是80MHZ。在输入数据宽度为32位，时钟频率为80MHz，加密或解密测试的数据同仿真得到的数据是相同的，该系统能够正常工作。

6 结论

本设计选择了FPGA来实现AES算法。在加解密的核心单元中采用了部分外部流水线进行设计，并在系统优化时采用逻辑锁技术对S盒和密钥存储进行了优化，以较少的系统资源获得了高达3200Mbit/s的数据吞吐量。仿真及片上测试均表明系统能正确进行加解密运算。

致谢

本论文得到了西华师范大学科研启动基金（05B019）及四川省教育厅科研基金重点项目（07ZA127）资助，在此一并致谢！

References (参考文献)

[1] Tim G, Mohammed B. AES on FPGA from the fastest to the smallest [C] Proceedings of CHES 2005.Springer, 2005: 427-441.

[2] Gaj K, Pawel Chodowie.Comparison of the hardware performance of the AES candidates using reconfigurable hardware Gael R, Francois-Xavier S, Jean-Jacques Q, et al. Compact and efficient encryption/decryption module for FPGA implementation of the AES ijndael very well suited for small embedded applications [C]Proceedings of the International Conference on Information Technology: Coding and Computing.2004,2: 583-587.

[3] National Institute of Standards and Technology.Advanced encryption standard [EB/OL]. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[4] LI Xue-mei;LU Er-hong;OU Hai-wen. Fast hardware design and implementation of AES algorithm. Application of Electronic Technique.NO.5,2006,48~50
李雪梅, 路而红, 欧海文. AES算法的快速硬件设计与实现[J]. 电子技术应用 2006年第5期, 48~50

[5] ZHANG De-xue; GUO Li; FU Zhong-qian. Design and implementation of AES algorithm based on FPGA. Journal of University of Science and Technology of China. Vol.37 No.12 1461~1465
张德学,郭立,傅忠谦. 一种基于FPGA的AES加解密算法设计与实现[J]. 中国科学技术大学学报. 第37卷第12期. 1461~1465.