View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Xiamen University Institutional Repository

中国集成电路
China Integrated Circuit
设计

基于轮内流水线技术的高性能 AES 硬件实现设计

郑行，王静，王云峰
（厦门大学 电子工程系，福建 厦门，361005）

摘要：为了提升 AES 的性能，本文采用轮内流水线技术进行 AES 硬件设计。在对 AES 轮单元复杂的字节代换 / 逆字节代换、列变换 / 逆列变换进行了算法分析的基础上，进行了 AES 轮单元的轮内 7 级流水线设计。特别是采用常数矩阵乘积形式和复用列变换进行了逆列变换设计，降低了硬件资源的占用。采用 Xilinx ISE10.1 工具进行了各个型号 FPGA 的硬件实现，实验数据表明文中提出的硬件实现方案提升了 AES 的数据吞吐率与吞吐率 / 面积比。

关键词：AES；复合域算法；轮内流水线

The hardware implementation of a high performance AES based on inner pipeline

ZHENG Xing, WANG Jing, WANG Yun-feng
(Department of Electronic Engineering, Xiamen University, Xiamen 361005, China)

Abtract: An inner pipelined hardware design of AES is presented in this paper for the performance improvement of AES. Based on the algorithmic analysis of the SubBytes/invSubBytes and MixColumns/invMixColumns, a 7-stage pipelined structure, which applies the invMixColumns design to the multiplexing MixColumns and adopts the form of arithmetic product of constant matrix, is proposed to reduce the cost of the hardware resources. The implementation of the proposed is carried out in several FPGAs using the Xilinx ISE10.1and the results have shown an improvement in the data throughout ratio and the ratio of data throughout and area.

Keywords: AES; Composite Field Algorithm; Inner pipeline

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

1 概论

高级加密标准 （Advanced Encryption Standard，AES）是 2000 年美国国家标准技术研究所（NIST）提出的新型加密标准，用来取代上一代的数据加密标准 （Data Encryption Stand，DES）。NIST 已经在 FIPS-197[1]上发表了详细的 AES 算法。与 AES 软件实现相比，硬件实现具有处理速度快、更安全的优势。因此，随着 AES 密码算法的广泛应用，AES 的高性能硬件解决方案已经成为当前密码学的研究热点。

AES 算法最主要的运算为轮转换运算。因此轮内流水线是当前最主要 AES 快速硬件实现技术，其不但可以在每轮之间插入轮外流水线寄存器，而且在每轮中也可以插入轮内流水线寄存器来缩短每轮关键路径的长度，从而大幅度提高 AES 的数据吞吐率。

Zhang[2]利用等价的复合域 $GF(((2^2)^2)^2)$算法来实现一个快而紧凑的 ASE 字节代换操作所要求的复合域乘法逆，但其设计的轮内各级流水线路径长度划分不够最优。Hodjat[3]利用完全环展开的高度轮内流水线架构，提出两种字节代换方案：一个是对于字节代换操作利用 BlockRAMs 来实现查找表，另一个是利用复合域 $GF((2^4)^2)$ 算法。这种利用复合域算法是由 Rijmen[4]建议并且由 Wolkerstorfer[5]证明。同样 Hodjat[3]在 7 级流水线设计中各级关路径划分也不是最优；并且环展开结构导致 AES 的吞吐率 / 面积比率比较小。Zambreno[6]同样采用完全环展开流水线设计，其中最深流水线设计是完全展开 3 级流水线，关键路径相对比较长。Good[7]根据级联的 FPGA 查找表（LUT）数来划分关键路径，进行完全环展开流水线设计，虽然增加了吞吐率，但更多的流水级划分增加了更多的寄存器而增加了面积。

由于关键路径决定硬件的最高工作频率，所以轮内流水线技术需要合理放置流水线寄存器，平衡各级流水线路径长度。本文分析 Zhang[2]中各级流水线关键路径长度，对其进行了优化设计。为了平衡各级流水线路径长度，减小实现方案的关键路径，本文重新进行了流水线划分，缩短最长关键路径长度，从而提高流水线工作频率和吞吐率。此外，在复用 AES 列混合和逆列混合操作中采用乘积形式设计，使其适于流水线设计且节省资源，增大了吞吐率 / 面积比。采用 Xilinx ISE10.1 工具进行了各个型号 FPGA 的硬件实现，实验数据所提出的硬件实现方案提升了 AES 的数据吞吐率与吞吐率 / 面积比。

2 AES 高级加密标准（AES）

AES 算法结构如图 1 所示，包括字节代换和逆字节代换（SubBytes 和 invSubBytes）、行移位和逆行移位（ShiftRows 和 invShiftRows）、列混合和逆列混合 （MixColumns 和 invMixColumns） 和轮密钥加（AddroundKey）四种基本操作，称为 AES 轮单元。行移位 / 逆行移位是状态矩阵中各行的循环左 / 右移。轮密钥加是将前一变换后的状态矩阵与一个轮密钥按位异或，得到一个新的状态矩阵。所以高性能 AES 算法硬件实现的重点在于字节代换 / 逆字节代换，列混合 / 逆列混合。

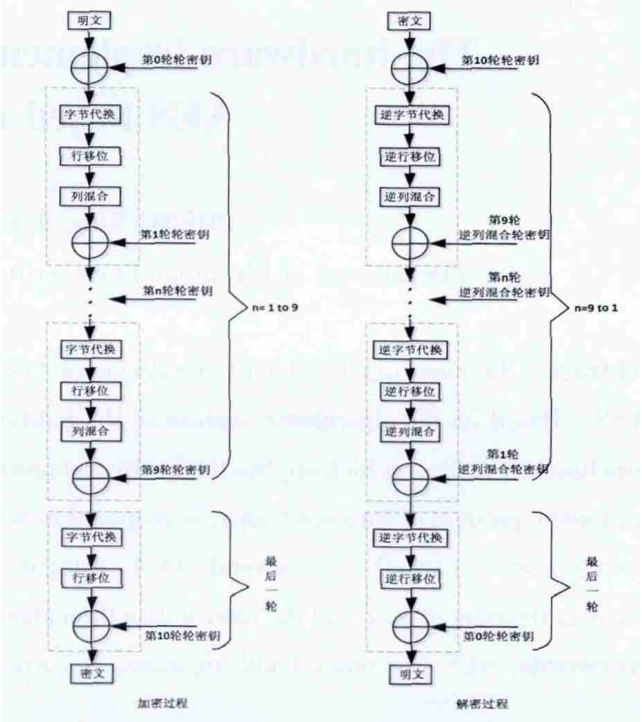

图 1 AES 加密和解密流程图

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

字节代换/逆字节代换是利用S盒/逆S盒将中间态中的每一个字节非线性变换为另一个字节，有效地实现每个字节数据中各位的混淆。替代过程为：如果状态中的一个字节为XY，则S盒/逆S盒中第X行第Y列的字节就是字节代换/逆字节代换的输出。S盒按如下方式构造：1、行X列Y的字节值初始化为十六进制数据的{XY}。2、把S盒中的每个字节映射为在有限域$GF(2^8)$中的逆。AES使用不可约多项式$m(x)=x^8+x^4+x^3+x+1$来构造$GF(2^8)$。3、S盒中的每个字节为$(a_7\,a_6\,a_5\,a_4\,a_3\,a_2\,a_1\,a_0)$，然后进行如下仿射变化$AT$：

$$b_i=a_i\oplus a_{(i+4)\bmod 8}\oplus a_{(i+5)\bmod 8}\oplus a_{(i+6)\bmod 8}\oplus a_{(i+7)\bmod 8}\oplus c_i \quad (1)$$

其中c_i是值为0×63字节c的第i位，即$(c_7\,c_6\,c_5\,c_4\,c_3\,c_2\,c_1\,c_0)=(01100011)$，从而代换成$(b_7\,b_6\,b_5\,b_4\,b_3\,b_2\,b_1\,b_0)$。其替代过程如图2所示。逆S盒则先进行逆仿射变化$AT^{-1}$：

$$b_i=a_{(i+2)\bmod 8}\oplus a_{(i+5)\bmod 8}\oplus a_{(i+7)\bmod 8}\oplus d_i \quad (2)$$

再计算$GF(2^8)$中的逆，其中d_i是值为0×05字节d的第i位，即$(d_7\,d_6\,d_5\,d_4\,d_3\,d_2\,d_1\,d_0)=(00000101)$。

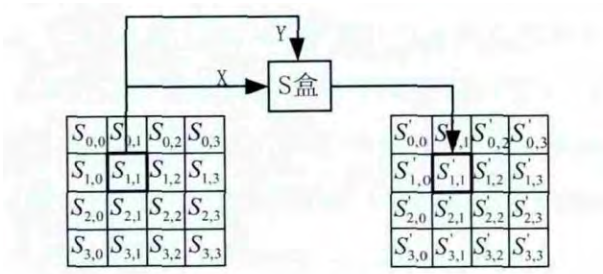

图2 字节代换过程

列混合/逆列混合是中间态逐列进行变换，与固定矩阵相乘。列混合是对一个状态数组逐列进行变换，它把每个列都看成$GF(2^8)$中的一个四项多项式是$S(x)$，再乘以固定多项式$a(x)$并模x^4+1，其中：$a(x)=\{03\}x^3+\{01\}x^2+\{01\}x+\{02\}$。相似的，逆列混合就每列乘以固定多项式$b(x)$并模$x^4+1$，其中：$b(x)=\{0b\}x^3+\{0d\}x^2+\{09\}x+\{0e\}$。其相应矩阵的表示方式如下：

$$\begin{bmatrix}S'_{0,C}\\S'_{1,C}\\S'_{2,C}\\S'_{3,C}\end{bmatrix}=\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&02&02&03\\03&01&01&02\end{bmatrix}\begin{bmatrix}S_{0,C}\\S_{1,C}\\S_{2,C}\\S_{3,C}\end{bmatrix}$$

$$\begin{bmatrix}S_{0,C}\\S_{1,C}\\S_{2,C}\\S_{3,C}\end{bmatrix}=\begin{bmatrix}0e&0b&0d&09\\09&0e&0b&0d\\0d&09&0e&0b\\0b&0d&09&0e\end{bmatrix}\begin{bmatrix}S'_{0,C}\\S'_{1,C}\\S'_{2,C}\\S'_{3,C}\end{bmatrix}$$

$$0\leqslant c\leqslant 3 \quad (3)$$

3 高性能AES硬件实现设计

所谓流水线技术，就是在组合逻辑电路中插入流水线寄存器，来缩短组合逻辑路径长度，达到提高工作频率、实现高吞吐率的目的。采用轮内流水线技术进行高性能AES硬件实现设计的关键就是如何利用最简单的组合电路实现AES的轮单元及如何进行流水线划分使得关键路径最小。

3.1 AES轮内分级流水线实现理论基础

AES轮单元中，行移位不需要逻辑门，轮密钥加只需简单的异或门，只有字节代换和列混合操作相对比较复杂。因此字节代换/逆字节代换、列混合/逆列混合的组合电路实现是高性能AES硬件实现的难点。

AES字节代换由乘法逆($GF(2^8)$)和仿射变换(AT)组成，逆字节代换由逆仿射变换(AT^{-1})和乘法逆($GF(2^8)$)组成，其中关键是如何用复合域$GF(2^n)^m$实现有限域$GF(2^8)$的乘法逆。

有限域$GF(2^8)$以不可约多项式$m(x)=x^8+x^4+x^3+x+1$构造出来，它的加法运算等价于两个字节的异或(XOR)运算，它的乘法运算等价于模不可约多项式$m(x)$的运算：

$$x^8 \bmod m(x)=[m(x)-x^8]=x^4+x^3+x+1 \quad (4)$$

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

所谓复合域算法[8]，就是将有限域 $GF(2^8)$ 的运算转换成有限域 $GF((2^n)^m)$ 下的操作，其中 $n\times m=8$，$GF(2^8)$ 与 $GF((2^n)^m)$ 是同构的。复合域运算前后，要进行域变换。域变换中同构映射可用一个 8×8 二进制矩阵 δ 和 δ^{-1} 实现域变换，其中：

$$\delta=\begin{bmatrix}1&0&1&0&0&0&0&0\\1&1&0&1&1&1&1&0\\1&0&1&0&1&1&0&0\\1&0&1&0&1&1&1&0\\1&1&0&0&0&1&1&0\\1&0&0&1&1&1&1&0\\0&1&0&1&0&0&1&0\\0&1&0&0&0&0&1&1\end{bmatrix}$$

$$\delta^{-1}=\begin{bmatrix}1&1&1&0&0&0&1&0\\0&1&0&0&0&1&0&0\\0&1&1&0&0&0&1&0\\0&1&1&1&0&1&1&0\\0&0&1&1&1&1&1&0\\1&0&0&1&1&1&1&0\\0&0&1&1&0&0&0&0\\0&1&1&1&0&1&0&1\end{bmatrix}\quad(5)$$

通过域变换，将有限域 $GF(2^8)$ 转换到复合域 $GF((2^4)^2)$。这样有限域 $GF(2^8)$ 中任意元素就等价到复合域 $GF((2^4)^2)$ 中的元素。

此外，有限域 $GF(2^8)$ 可由更小的有限域 $GF(2)$、$GF(2^2)$ 和 $GF((2^2)^2)$ 使用不可约多项式迭代地建立，相应域变换及不可约多项式[9]：

$$\begin{cases}GF(2)\Rightarrow GF(2^2):P_0(x)=x^2+x+1\\GF(2^2)\Rightarrow GF((2^2)^2):P_1(x)=x^2+x+\phi\\GF((2^2)^2)\Rightarrow GF(((2^2)^2)^2):P_2(x)=x^2+x+\lambda\end{cases}\quad(6)$$

其中 $\phi=\{10\}_2$，$\lambda=\{1100\}_2$。

有限域 $GF(2^8)$ 中任意元素可以表示成 $bx+c$，文献[9]采用 $x^2+x+\lambda$ 为不可约多项式，则元素 $bx+c$ 的乘法逆可以转换为有限域 $GF(2^4)$ 中求解，由下式计算得出：

$$(bx+c)^{-1}=b(b^2\lambda+c(b+c))^{-1}x+(c+b)(b^2\lambda+c(b+c))^{-1}\quad(7)$$

由于字节代换、逆字节代换均含有求解乘法逆过程，因此硬件实现字节代换 / 逆字节代换时，通常采用复用乘法逆结构[8]来实现，如图 3 所示。当选择域变换、乘法逆、逆域变换与仿射变换通路时，实现的是字节代换，用于加密过程；当选择逆仿射变换与域变换、乘法逆、逆域变换通道时，实现的是逆字节变换，用于解密过程。因此可以采用复合域运算代替查找 S 盒 / 逆 S 盒实现字节代换 / 逆字节代换，硬件设计时可以仅通过组合逻辑来实现。不但降低了字节代换 / 逆字节代换硬件实现的面积复杂度，还便于进行 AES 轮单元的流水线设计，提升 AES 的性能。

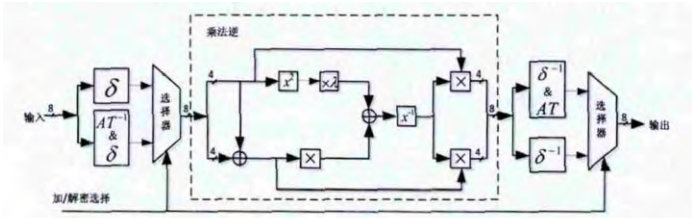

图 3 字节代换 / 逆字节代换

由列混合和逆列混合变换矩阵可知，在加密过程中列混合需要在有限域 $GF(2^8)$ 上进行 $\{02\}_{16}u$ 和 $\{03\}_{16}u$ 常数乘运算；在解密过程中逆列混合需要在有限域 $GF(2^8)$ 上进行 $\{09\}_{16}u$，$\{0b\}_{16}u$，$\{0d\}_{16}u$ 和 $\{0e\}_{16}u$ 常数乘运算。而常数 $\{02\}_{16}$、$\{03\}_{16}$、$\{09\}_{16}$、$\{0b\}_{16}$、$\{0e\}_{16}$ 通过二进制矩阵 δ 实现域变换后变成常数 $\{5f\}_{16}$、$\{5e\}_{16}$、$\{75\}_{16}$、$\{2a\}_{16}$、$\{57\}_{16}$，非“0”位增多，从而乘法运算变得复杂[2]。因此列混合和逆列混合不用复合域算法来实现，而是直接在有限域 $GF(2^8)$ 下实现更为合理。

设有限域 $GF(2^8)$ 上任意元素 u 的二进制形式为 $\{u_7,u_6,u_5,u_4,u_3,u_2,u_1,u_0\}$，则其多项式表达形式如式 8：

$$u=u_7x^7+u_6x^6+u_5x^5+u_4x^4+u_3x^3+u_2x^2+u_1x+u_0\quad(8)$$

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

由于$\{02\}_{16}$的多项式表示形式为 x，所以

$$\{02\}_{16}u = x\cdot(u_7x^7+u_6x^6+u_5x^5+u_4x^4+u_3x^3+u_2x^2+u_1x+u)$$
$$= u_6x^7+u_5x^6+u_4x^5+(u_3+u_7)x^4+(u_2+u_7)x^3+u_1x^2+(u_0+u_7)x+u_7$$

进而$\{02\}_{16}u$的二进制结果为$\{u_6,u_5,u_4,u_3\oplus u_7,u_2\oplus u_7,u_1,u_0\oplus u_7,u_7\}$，可以得出$\{02\}_{16}u$的逻辑门数为 3XOR，而最大延迟为 1XOR。

Zhang[2]采用复用列混合实现逆列混合，即相当于两个参数矩阵相加形式实现逆列混合。参数矩阵相加形式如式 9 所示：

$$\begin{bmatrix}0e&0b&0d&09\\09&0e&0b&0d\\0d&09&0e&0b\\0b&0d&09&0e\end{bmatrix}=\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&02&02&03\\03&01&01&02\end{bmatrix}\oplus\begin{bmatrix}03&02&03&02\\02&03&02&03\\03&02&03&02\\02&03&02&03\end{bmatrix}\cdot\{04\}\quad(9)$$

其中$\{03\}_{16}u=\{02\}_{16}u\oplus u$，$\{04\}_{16}u$可以通过连续两次$\{02\}_{16}u$实现。则式 9 的硬件实现需要总门数为 193XOR，最大延迟为 7XOR。

3.2 改进的 AES 轮内流水线设计

基于式 1、式 2、式 5、式 7、式 9，Zhang[2] 进行了 7 级轮内流水级设计，其划分如图 4 所示。其关键路径出现在第 2 级，为 5XOR+1AND，而最小延迟出现在第 3 级，为 3XOR+2AND，划分不够均衡。第二阶段的 x^2 与 $\times\lambda$，虽然划分了在同一阶段，但分别实现，以至于没有进行整体优化，浪费了硬件资源。此外其逆列混合采用常数矩阵加法的形式实现，也占用了较多的硬件资源。

本文采用常数矩阵乘积形式实现逆列混合运算，其变换矩阵分解如式 10 所示。

$$\begin{bmatrix}0e&0b&0d&09\\09&0e&0b&0d\\0d&09&0e&0b\\0b&0d&09&0e\end{bmatrix}=\begin{bmatrix}02&03&01&01\\01&02&03&01\\01&02&02&03\\03&01&01&02\end{bmatrix}\cdot\begin{bmatrix}05&00&04&00\\00&05&00&04\\04&00&05&00\\00&04&00&05\end{bmatrix}\quad(10)$$

标记上面参数矩阵依次为 C_0,C_1,C_2，则逆列混合可化为如下形式：

$$S=C_0\cdot S'=(C_1\cdot C_2)\cdot S'=C_1\cdot(C_2\cdot S')\quad(11)$$

其中$\{05\}_{16}u=\{04\}_{16}u\oplus u$，$C_2\cdot S$，可以化为：

$$S''_{0,C}=(\{04\}\times S'_{0,C})\oplus(\{04\}\times S'_{2,C})\oplus S'_{0,C}$$
$$S''_{1,C}=(\{04\}\times S'_{1,C})\oplus(\{04\}\times S'_{3,C})\oplus S'_{1,C}$$
$$S''_{2,C}=(\{04\}\times S'_{0,C})\oplus(\{04\}\times S'_{2,C})\oplus S'_{2,C}$$
$$S''_{3,C}=(\{04\}\times S'_{1,C})\oplus(\{04\}\times S'_{3,C})\oplus S'_{3,C}$$
$$0\leqslant c\leqslant 3\quad(12)$$

其中

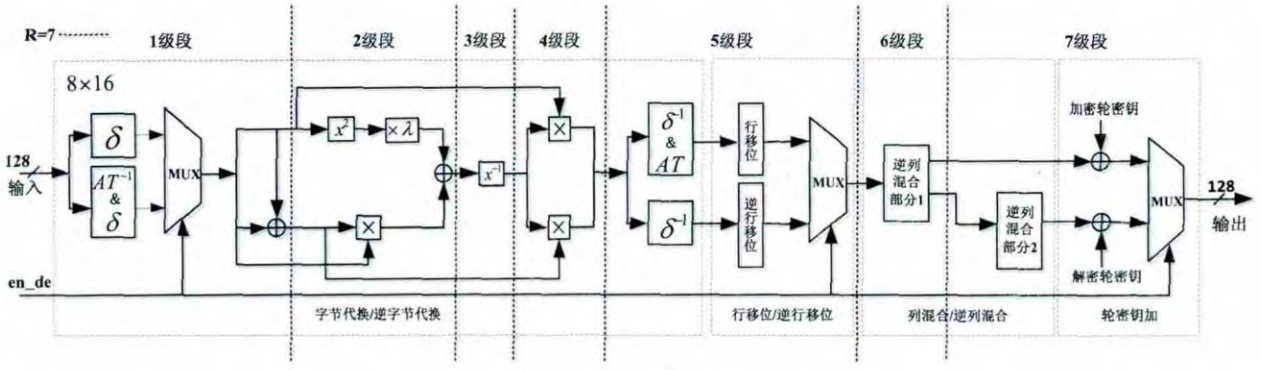

图 4 Zhang [2]的轮内流水线结构

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$\{04\}_{16}u = \{02\}_{16}u \cdot \{02\}$$
$$= \{u_5, u_4, u_3 \oplus u_7, u_2 \oplus u_7, u_1, u_0 \oplus u_7, u_7, u_0\}$$
$$\oplus \{0,0,0,u_6,u_6,0,u_6,u_6\}$$
$$= \{u_5, u_4, u_3 \oplus u_7, u_2 \oplus u_7 \oplus u_6, u_1$$
$$\oplus u_6, u_0 \oplus u_7, u_7 \oplus u_6, u_6\}$$

乘积形式的逆列混合设计如图 5。C_2 占用的逻辑门数 / 最大路径延迟逻辑门数为 58XOR/4XOR，列混合单元 C_1 占用的逻辑门数 / 最大路径延迟逻辑门数为 124XOR/3XOR。由于 C_1 为列混合变换矩阵，因此此部分可以复用进行列混合 / 逆列混合变换的硬件实现，其结构如图 6 所示。

本文对图 4 所示轮内流水线进行了改进，如图 7 所示。首先重新进行了轮内流水线的划分，使得各级路径延迟更加均衡，关键路径降为 4XOR+1MUX。并且合并模块 x^2 与 $\times\lambda$，使得此部分占用硬件资源由 7 XOR 降为 4 XOR。采用乘积形式实现逆列混合，所占硬件资源由 193 个 XOR 降为了 182 个 XOR。改进后的各级流水路径长度与原结构对比如表 1 所示。

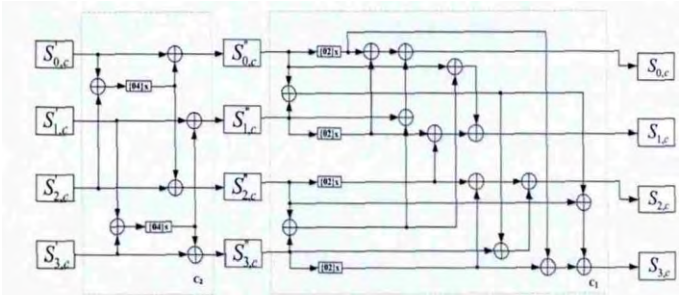

图 5 乘积形式的逆列混合

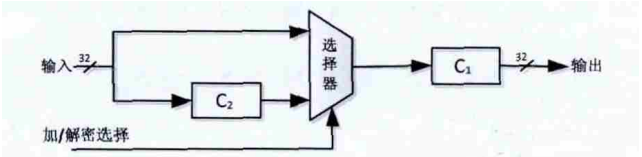

图 6 复用列混合实现列混合 / 逆列混合复用

3.3 性能分析

本文采用 Xilinx ISE10.1 工具进行了各个型号 FPGA 的硬件实现，所得数据如表 2 所示。本文的设计在 Virtex-E XCV2000E-8 器件上实现工作频率为 173.8 MHz，比 Zhang[2]相同 Virtex-E 器件上实现要快 3.2%。本文设计在 Sparatan-3 XC3S4000-5 器件上实现比更多级流水线实现高吞吐率的 Good[7]的吞吐率 / 面积有效性高 16.3%。此外，本文设计在 Virtex-2 器件上实现工作频率比 Fu[10]，Zambreno[6]和 Good[7]还高，在 XC2V3000-6 上实现工作频率达到 263.7 Mhz，吞吐率达到 33.7 Gbps，吞吐率 / 面积有效性比相应最高的 Fu[10]高 56.5%。在 Virtex-2 Pro XC2VP40-7 器件上实现工作频率是 Hodjat[3]相同类型器件上的 2 倍，实现吞吐率为 43.7 Gbps，吞吐率 / 面积有效性更是达到 3.009 Mbps/slice。在 Virtex-4XC4VLX200-11 器件上实现工作频率比 Fan[11]相同器件实现快 60.8%，达到吞吐率 51.4 Gbps，吞吐率 / 面积有效性更是远远高于 Fan[11]，达到本文实现最高的 3.539 Mbps / slice。最后，在 Virtex - 5 XC5VLX110T-1 器件上实现工作频率是 Vanitha[12]高 2 倍多，实现吞吐率 52.9 Gbps，吞吐率 / 面积（Mbps/LUT）有效性比 Vanitha [12] 高 29.8%。在 Virtex-5 XC5VLX110T-3 器件上实现工作频率

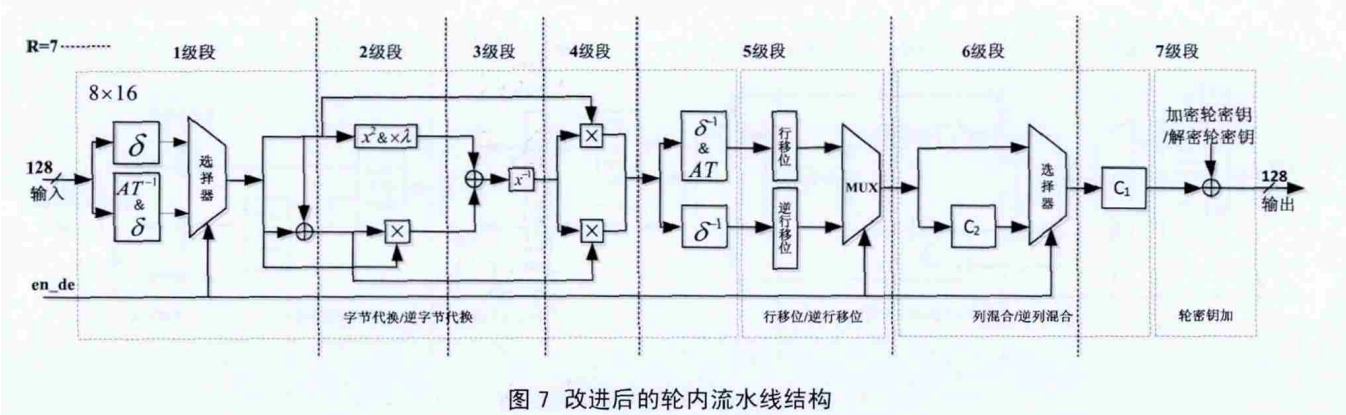

图 7 改进后的轮内流水线结构

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

523.4 MHz，实现吞吐约 67 Gbps，吞吐率 / 面积（Mbps/LUT）有效性达到 3.468 Mbps/LUT。

4 小结

本文基于复合域算法和流水线技术，进行高性能的 AES 硬件设计。通过对 AES 轮单元各级流水线路径长度的分析，平衡路径上的逻辑门数来缩短最长关键路径的长度，从而达到最优流水线设计，实现工作频率和吞吐率的提高。采用常数矩阵乘积的形式、复用列混合变换实现逆列混合变换，从而节省了整个 AES 的硬件资源，增大了吞吐率 / 面积比。最终的实现结果分析表明，本文的设计提升了 AES 的性能。

表 1 两种轮内流水线结构对比

加密解密流水段	1级段	2级段	3级段	4级段	5级段	6级段	7级段
改进后的各级流水线路径长度	4XOR+1MUX	4XOR+1AND	4XOR+2AND	4XOR+1AND	3XOR+1MUX	4XOR+1MUX	4XOR
Zhang[3]的各级流水线路径长度	4XOR+1MUX	5XOR+1AND	3XOR+2AND	4XOR+1AND	4XOR+1MUX	4XOR	4XOR

表 2 在 FPGA 上实现 AES 算法的结果与比较

方案	FPGA 型号	工作频率 (MHz)	吞吐率（Mbps）	资源（slices）	吞吐率/面积比（Mbps/slice）	模式
Hodjat[3]	Virtex-2 Pro XC2VP20	169.1	21640	9446	2.290	加密
Zambreno[6]	Virtex-2 XC2V4000	184.1	23570	16938	1.391	加密
Zhang[2]	Virtex-E XCV1000E-8	168.4	21556	11022	1.956	加密/解密
Fu[10]	Virtex-2 XC2V1000	212.5	27100	17887	1.515	加密
Good[7]	Virtex-2 XC2V8000-5	222.8	28526	31674	0.901	加密/解密
Good[7]	Spartan-3 XC3S4000-5	240.9	30835	20720	1.488	加密/解密
Fan[11]	Virtex-4 XC4VLX200	250	32000	86806	0.369	加密
Vanitha[12]	Virtex5 XC5VLX110T-1	198.3	22890	14522	2.110	加密
本文	Virtex-E XCV2000E-8	173.8	22246	16456	1.352	加密/解密
本文	Spartan-3 XC3S4000-5	195.5	25024	14453	1.731	加密/解密
本文	Virtex-2 XC2V8000-5	233.5	29888	14541	2.055	加密/加密
本文	Virtex-2 XC2V3000-6	263.7	33754	14238	2.371	加密/解密
本文	Virtex-2 Pro XC2VP40-7	341.5	43712	14528	3.009	加密/解密
本文	Virtex-4 XC4VLX200-11	402.1	51469	14545	3.539	加密/解密
本文	Virtex-5 XC5VLX110T-1	413.6	52941	19320 (LUTs)	2.740 (Mbps/LUT)	加密/解密
本文	Virtex-5 XC5VLX110T-3	523.4	66995	19320 (LUTs)	3.468 (Mbps/LUT)	加密/解密

参考文献

[1] National Institute of Standards and Technology (NIST): Advanced Encryption Standard (AES) [S]. Federal Information Processing Standards (FIPS) Publication 197, November 2001.

[2] Zhang X, Parhi K K. High-speed VLSI architectures for the AES algorithm [J]. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2004, 12 (9): 957-967.

[3] Hodjat A, Verbauwhede I. A 21.54 Gbits/s fully pipelined AES processor on FPGA [C]. Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on. IEEE, 2004: 308-309.

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

[4] Rijmen V. Efficient Implementation of the Rijndael S-box[J]. Katholieke Universiteit Leuven, Dept. ESAT. Belgium, 2000.
[5] Wolkerstorfer J, Oswald E, Lamberger M. An ASIC implementation of the AES SBoxes[M]. Topics in Cryptology—CT-RSA 2002. Springer Berlin Heidelberg, 2002: 67-78.
[6] Zambreno J, Nguyen D, Choudhary A. Exploring area/delay tradeoffs in an AES FPGA implementation [M]. Field Programmable Logic and Application. Springer Berlin Heidelberg, 2004: 575-585.
[7] Good T, Benaissa M. Pipelined AES on FPGA with support for feedback modes (in a multi-channel environment) [J]. IET Information Security, 2007, 1 (1): 1-10.
[8] 刘政林, 曾永红, 邹雪城, 等. 复合域算法的 AES S 盒电路实现 [J]. 应用科学学报, 2008, 26 (6): 622-626.
[9] Satoh A, Morioka S, Takano K, et al. A compact Rijndael hardware architecture with S-box optimization [M]. Advances in Cryptology—ASIACRYPT 2001. Springer Berlin Heidelberg, 2001: 239-254.
[10] Fu Y, Hao L, Zhang X, et al. Design of an extremely high performance counter mode AES reconfigurable processor[C]. Embedded Software and Systems, 2005. Second International Conference on. IEEE, 2005: 7 pp.
[11] Fan C P, Hwang J K. Implementations of high throughput sequential and fully pipelined AES processors on FPGA [C]. Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on. IEEE, 2007: 353-356.
[12] Vanitha M, Sakthivel R, Subha S. Highly secured high throughput VLSI architecture for AES algorithm[C]. Devices, Circuits and Systems (ICDCS), 2012 International Conference on. IEEE, 2012: 403-407.

作者简介

郑行,厦门大学电子工程系在读硕士研究生,研究方向:集成电路设计与应用;
王静,厦门大学电子工程系在读硕士研究生,研究方向:集成电路设计与应用;
王云峰,副教授,厦门大学电子工程系硕士研究生导师。

上接第 46 页

[4] R. Schreier and G. C. Temes. Understanding delta-sigma data converters. Piscataway, NJ: IEEE Press/Wiley, 2005, 11-26.
[5] M. Rebeschini. Delta sigma data converters: theory, design, and simulation. New York, NY: IEEE Press, 1996, 32-60.
[6] K. Souri, Y. Chae and K. A. A. Makinwa. A CMOS temperature sensor with a voltage-calibrated inaccuracy of ±0.15℃ (3σ) from -55 to 125℃. IEEE J. Solid-State Circuits, 2013, 48 (1): 292-301.
[7] J. Markus, J. Silva, and G. C. Temes. Theory and applications of incremental Sigma-Delta converters. IEEE Transactions on Circuits and Systems I, 2004, 51 (4): 678-690.
[8] B. Razavi, B. A. Wooley. Design techniques for high speed, high resolution Comparators. IEEE J. Solid-State Circuits, 1992, 27 (12):1916-1926.

作者简介

魏榕山,副教授,硕士生导师,主要从事集成电路的研究与设计。
陈锦锋,硕士研究生,主要从事 Sigma-Delta 模数转换器的研究与设计。
陈传东,讲师,主要从事集成电路的研究与设计。

?1994-2016 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net