AES 中 SubBytes 算法在 FPGA 的实现

姜 强,谢 军

(电子科技大学 通信与信息工程学院,四川 成都 611731)

摘 要:介绍了 AES 中,SubBytes 算法在 FPGA 的具体实现。构造 SubBytes 的 S-Box 转换表可以直接查找 ROM 表来实现。通过分析 SubBytes 算法得到一种可行性硬件逻辑电路,从而实现 SubBytes 变换的功能。

关键词:高级加密标准;SubBytes;现场可编程逻辑门阵列

中图分类号:TP309 **文献标识码**:A

SubBytes algorithm of AES in the FPGA implementation

JIANG Qiang, XIE Jun

(School of Communication and Information Engineering, University of Electronic Science and Technology, Chengdu 611731, China)

Abstract: This article describes the SubBytes algorithm of AES in FPGA implementation. Build SubBytes of S-Box transform table is directly through ROM table to achieve. By analyzing the SubBytes algorithm, the article find a kind of hardware logic circuits which can achieve SubBytes transform function.

Key words: AES; SubBytes; FPGA

高级加密标准 AES(Advanced Encryption Standard)是一种对称密码算法。数据分组长度为 128 bit,密钥长度可以为 128 bit、192 bit 和 256 bit 3 种形式,不同的密钥长度其迭代的次数分别为 10 轮、12 轮、14 轮。一般在实际应用中,经常选择密钥长度为 128 bit 或 256 bit。

AES 密码算法中字节替换(SubBytes)的关键是如何构造 S-Box。参考文献[1]中阐述了 AES 标准;RIJMEN V 提出了构造 S-Box 方法[2]及 SubBytes 在伽罗华有限域(GF 域)的变化[3];OSWALD E 介绍了 ASIC[4]的应用;EAEMEN J 详细介绍了 AES 算法[5]。本文具体介绍 SubBytes 在 FPGA 构造中 S-Box 的实现。

1 数学基础

1.1 字节

AES 密码算法中的每个基本数据字节 $a=\{a_7,a_6,a_5,a_4,a_3,a_2,a_1,a_0\}$都包含 8 bit。任何一个字节都可以在伽罗瓦有限域 $GF(2^8)$中以多项式表示:$a(x)=\sum_{i=0}^{7}a_i x^j=a_7x^7+a_6x^6+a_5x^5+a_4x^4+a_3x^3+a_2x^2+a_1x^1+a_0$ 多项式中的系数 a_i 就是对应字节中的位。

1.2 加法

多项式 $a(x),b(x)\in GF(2^8)$相加时记作:$a(x)\oplus b(x)$,将多项式中对应幂的系数进行“异或”,即

$$a(x)\oplus b(x)=\sum_{i=0}^{7}a_i x^i\oplus\sum_{i=0}^{7}b_i x^i=\sum_{i=0}^{7}(a_i\oplus b_i)x^i。$$

1.3 乘法

多项式在 $GF(2^8)$上的乘法对于将 2 个多项式相乘的结果再对 1 个模为 8 的不可约分多项式取模。在 AES 中,该不可约分多项式为:$m_8(x)=x^8+x^4+x^3+x+1$。例如:

$$(x^6+x^4+x^2+x+1)\otimes(x^7+x+1)\bmod(m_8(x))=(x^{13}+x^{11}+x^9+x^8+x^6+x^5+x^4+x^3+1)\bmod(x^8+x^4+x^3+x+1)=x^7+x^6+1$$

2 SubBytes 算法的结构

SubBytes 是 AES 加密算法中唯一的非线性变换,每个字节都是通过 1 个 S-Box 变换得到的,其具体变换过程如图 1 所示。

S-Box 通过函数:$S-Box[a]=f(g(a))$得到,其中 $g(a)$:

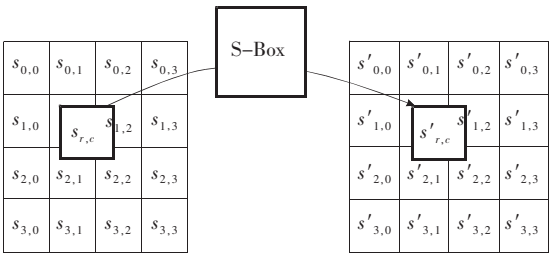

图 1 S-Box 转换

$a \rightarrow a^{-1} \in GF(2^8)$；$f(a)$是一种仿射变换。通常域 $GF(2^8)$是域 $GF(2)$的扩展，因此将多项式 a 转换成在 $GF(2)$域中的线性多项式 $a=a_h x+a_l$，其中 $a \in GF(2^8)$，a_h、$a_l \in GF(2^4)$。在域 $GF(2)$和 $GF(2^4)$域中乘法的不可约分多项式分别为 $m_2(x)=x^2+\{1\}x+\{e\}$和 $m_4(x)=x^4+x+1$。所以若 $a(x) \otimes b(x)=(a_h x+a_l) \otimes (b_h x+b_l)=\{0\}x+\{1\}$，根据有限域乘法规则可知：

$$\{0\}x+\{1\}=a(x) \otimes b(x)=(a_h x+a_l) \otimes (b_h x+b_l)=$$

$$(a_h b_h x^2+a_h b_l x+a_l b_h x+a_l b_l) \bmod (m_2(x))=$$

$$(a_h b_h x^2+a_h b_l x+a_l b_h x+a_l b_l) \bmod (x^2+\{1\}x+\{e\})=$$

$$(a_h b_h+a_h b_l+a_h b_l)x+(a_l b_l+a_h b_h \otimes \{e\})$$

根据等式两边系数关系可得：

$$\begin{cases} b_h=a_h \otimes ((a_h^2 \otimes \{e\}) \oplus (a_h \otimes a_l) \oplus a_l^2)^{-1} \\ b_l=(a_h \oplus a_l)((a_h^2 \otimes \{e\}) \oplus (a_h \otimes a_l) \oplus a_l^2)^{-1} \end{cases} \Rightarrow$$

$$\begin{cases} b_h=a_h \otimes d \\ b_l=(a_h \oplus a_l) \otimes d \end{cases} \qquad d=((a_h^2 \otimes \{e\}) \oplus (a_h \otimes a_l) \oplus a_l^2)^{-1}$$

因此通过推导得出 $a^{-1}=(a_h x \oplus a_l)^{-1}=(b_h x+b_l)$，由此可得到 a^{-1} 的硬件结构图，如图 2 所示。

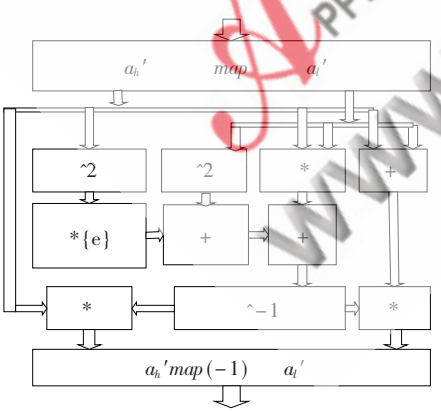

图 2 硬件结构图

map 算法[3]如下：

$$a_A=a_1 \oplus a_7 \quad a_B=a_5 \oplus a_7 \quad a_C=a_4 \oplus a_6$$

$$a_{l0}=a_C \oplus a_0 \oplus a_5 \qquad a_{h0}=a_C \oplus a_5$$

$$a_{l1}=a_1 \oplus a_2 \qquad a_{h1}=a_A \oplus a_C$$

$$a_{l2}=a_A \qquad a_{h2}=a_B \oplus a_2 \oplus a_3$$

$$a_{l3}=a_2 \oplus a_4 \qquad a_{h3}=a_B$$

map^{-1} 算法[3]如下：

$$b_A=a'_{l1} \oplus a_{h3} \qquad b_B=a'_{h0} \oplus a'_{h1}$$

$$a_0^{-1}=a'_{l0} \oplus a'_{h0} \qquad a_1^{-1}=b_B \oplus a'_{h3}$$

$$a_2^{-1}=b_A \oplus b_B \qquad a_3^{-1}=b_B \oplus a'_{l1} \oplus a'_{l2}$$

$$a_4^{-1}=b_A \oplus b_B \oplus a'_{l3} \qquad a_5^{-1}=b_B \oplus a'_{l2}$$

$$a_4^{-1}=b_A \oplus a'_{l2} \oplus a'_{l3} \oplus a'_{h0} \qquad a_7^{-1}=b_B \oplus a'_{l2} \oplus a'_{h3}$$

由此通过计算可以得到 $g(a)$变换，如图 3 所示。

x\y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	00	01	8d	f6	cb	52	7b	d1	e8	4f	29	c0	b0	e1	e5	c7
1	74	b4	aa	4b	99	2b	60	5f	58	3f	fd	cc	ff	40	ee	b2
2	3a	6e	5a	f1	55	4d	a8	c9	c1	0a	98	15	30	44	a2	c2
3	2c	45	92	6c	f3	39	66	42	f2	35	20	6f	77	bb	59	19
4	1d	fe	37	67	2d	31	f5	69	a7	64	ab	13	54	25	e9	09
5	ed	5c	05	ca	4c	24	87	bf	18	3e	22	f0	51	ec	61	17
6	16	5e	af	d3	49	a6	36	43	f4	47	91	df	33	93	21	3b
7	79	b7	97	85	10	b5	ba	3c	b6	70	d0	06	a1	fa	81	82
8	83	7e	7f	80	96	73	be	56	9b	9e	95	d9	f7	02	b9	a4
9	de	6a	32	6d	d8	8a	84	72	2a	14	9f	88	f9	dc	89	9a
a	fb	7c	2e	c3	8f	b8	65	48	26	c8	12	4a	ce	e7	d2	62
b	0c	e0	1f	ef	11	75	78	71	a5	8e	76	3d	bd	bc	86	57
c	0b	28	2f	a3	da	d4	e4	0f	a9	27	53	04	1b	fc	ac	e6
d	7a	07	ae	63	c5	db	e2	ea	94	8b	c4	d5	9d	f8	90	6b
e	b1	0d	d6	eb	c6	0e	ef	ad	08	4e	d7	e3	5d	50	1e	b3
f	5b	23	38	34	68	46	03	8c	dd	9c	7d	a0	cd	1a	41	1c

图 3 $g(a)$变换

$f(a)$仿射变换的函数如下：

$$\begin{bmatrix} a'_0 \\ a'_1 \\ a'_2 \\ a'_3 \\ a'_4 \\ a'_5 \\ a'_6 \\ a'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

根据 S-Box 变换过程的函数得到 $S-Box[a]=f(g(a))$。通过计算，最后可得 S-Box 字节替换，如图 4 所示。

3 综合及测试结果

本设计用 Verilog HDL 语言进行行为级描述，选用 Altera 公司的 EP1S10B672C6 芯片，通过 Quartus II 软件综合的逻辑单元总消耗为 85（芯片总资源为 10 570 个 LE）。

在 Modelsim SE 软件中进行仿真，其仿真结果如图 5 所示。

x\y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	06	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

图 4 S-Box 字节替换

AES 加密算法中的 SubBytes，在加密过程、密钥扩展中得到了具体的应用。本文通过分析 SubBytes 算法，介绍了一种适合硬件结构的设计方法。结合此种设计方法，在整个 AES 加密过程中可以分级流水、加快 AES 加密速度、提高硬件的工作效率。

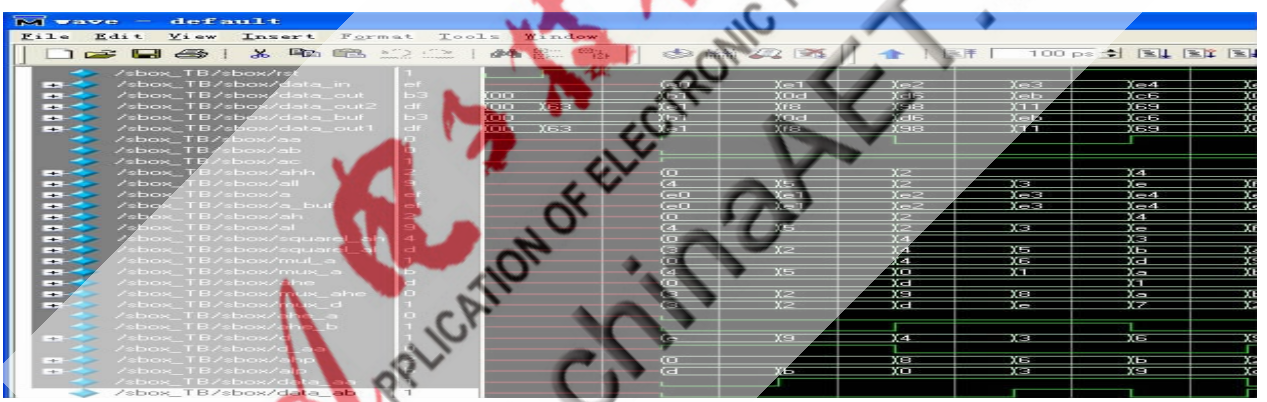

图 5 仿真结果

（收稿日期：2009-09-02）

参考文献

[1] NIST. Advanced Encryption Standard (AES)[S]. FIPS PUBS 197, National Institue of Standards and Tecnology, 2001.

[2] RIJMEN V. Effcient implementation of the Rijndael SBox [EB/OL]. http://www.esat.kuleuven.ac.be/~rijmen/rijndael,1999.

[3] WOLKERSTORFER J, OSWALD E, An ASIC implemtntation of AES S-Boxes[C]. Springer Verlag Berlin Heidelberg, 2002:67-68.

[4] OSWALD E. A side-channel analysis resistant description of the AES S-Box [C]. Springer Verlag:12th International Workshop on Fast Software Encryption, 2005:413-423.

[5] EAEMEN J.高级加密标准(AES)算法—Rijndael 的设计[M].北京：清华大学出版社，2003.

作者简介：

姜强，男，1985 年生，硕士研究生，主要研究方向：存储加密、光纤通道(FC)和 FPGA 技术。