

基于 HLS 技术的 Rijndael 算法 IP 核实现与优化

孙桂玲, 纪永鑫, 张潺潺, 李维祥

(南开大学 信息技术与科学学院, 天津 300071)

摘 要: 为了降低传统设计模式在应对大规模 SoC 设计时带来高复杂度, 使用高层次综合 HLS 技术进行了 Rijndael 算法 IP 核的设计、综合与仿真. 针对 Rijndael 算法中的多种运算模块, 研究并设计了面向硬件的编码方式及优化方案. 通过对比, 使用高层次综合技术设计的 IP 核在各方面都接近或超越了使用传统方式设计的 IP 核, 而设计复杂度大大降低, 证明了使用 HLS 方法进行设计的优越性.

关键词: 高层次综合; Catapult Synthesis; Rijndael; SystemC

中图分类号: TN47

文献标识码: A

文章编号: 1000-7180(2010)04-0205-04

Implementation and Optimization of Rijndael Arithmetic IP Core Based on HLS Technology

SUN Gui-ling, JI Yong-xin, ZHANG Chan-chan, LI Wei-xiang

(College of Information Technical Science, Nankai University, Tianjin 300071, China)

Abstract: In order to reduce the high complexity of the traditional SoC design method, this paper presents the design, synthesis and simulation flow of the Rijndael arithmetic IP Core using high level synthesis (HLS) technology. A hardware-oriented coding style and its optimization scheme specifically for the Rijndael arithmetic is also studied and designed. Through a comparison with a traditionally designed IP Core, the IP Core designed by HLS technology outperforms in nearly all respects, with a significantly reduced design complexity, which greatly proves the advantage of HLS.

Key words: high level synthesis (HLS); Catapult Synthesis; Rijndael; SystemC

1 引言

伴随着 SoC 技术的广泛运用以及深亚微米技术的发展, 使得在一个芯片上集成百万甚至上亿个晶体管成为现实, 但是也给 IC 设计带来了极高的复杂性. 传统的基于硬件描述语言的设计方法难以满足要求. 因此, 行为级描述的设计思想和与之对应的高层次综合技术(HLS)必将成为下一代 EDA 设计方法的主流.

高层次综合首先将算法描述进行编译, 得到数据流图 CDFG. 随后对 CDFG 进行调度和分配, 其主要目的是在满足约束条件的情况下使规定的目标函数最小, 此后得到数据通路的信息, 最后根据系统行

为设计状态机. 最终结果可以由数据通路和状态机生成可综合的 RTL 级代码. 另外高层次综合允许设计者进行设计空间搜索. 设计者可通过设定约束, 如延迟、面积、功耗等, 在高层次综合工具的帮助下, 验证和比较多种设计方案等.

2 针对硬件实现的 C 代码优化

2.1 并行化运算

与基于通用处理器的实现相比, 硬件实现的一大优势在于可以有多个运算单元同时工作, 因此, 尽量将运算操作并行化, 可以减少运算周期, 如使用括号减少多次运算间的优先级, 使用公共子串表达式等.

收稿日期: 2009-04-02; 修回日期: 2009-07-08

基金项目: 国家“八六三”计划项目(2006AA01Z217); 天津市科技支撑计划重点项目(08ZCKFGX00500)

2.2 算符优化

即使许多现代的 FPGA 已经集成了如乘加器等专用 DSP 单元,乘法运算仍旧是资源和功耗消耗较大的运算,而除法器 and 浮点运算单元等需要更大的逻辑单元,同时也增加了综合与布线的工作量,往往成为系统的关键路径,降低系统的最大工作频率。

不过,当某个操作数为常数时,乘除法可以简单转换成移位和加法运算,可以用很低的代价实现,因此,将部分乘除法运算特殊化处理为多个常数表达式可以起到减小面积和延迟的作用^[1-2]。

2.3 减少操作相关性

过长的操作依赖会导致硬件串行化及过多的组合逻辑级联,增加了运算周期和时钟延迟,具体的优化包括简化逻辑,循环展开,使用多变量循环等。

2.4 针对吞吐率优化的代码风格

流水线是增加数据吞吐(Throughput)的重要手段,而不同的代码风格可能导致流水线调度时产生冲突,进而调度失败.考虑如下的循环^[3]:

```
void throughput (args)
    outer: loop
        middle: loop
            inner: loop
                main code
```

所有对 args 的访问都将被转换成模块 I/O,除最内部循环之外的对 args 的访问都将获得较低的吞吐等级,进而阻塞流水,因此应将所有 I/O 操作放在 inner 循环内.此外,args 最好应设定为指针类型,如果设为普通变量,则对其读写需要访问最外圈循环外的寄存器,严重影响吞吐效率。

最重要的,应把主要代码放入内部循环中,外圈循环中的代码会极大增加综合复杂度,导致更大的资源占用和更长的延迟。

当多重循环不可避免时,使用循环展开(Unrolling)或者多循环融合(Merging)的方式可以尽量地减少吞吐率的损失。

3 Rijndael 算法的实现与优化

Rijndael 算法每一轮变换均由 4 个不同的变换组成^[4],包括 S 盒变换(SubBytes),行变换(ShiftRow),列变换(MixColumns)和密钥轮加(AddRoundKey),分别对应非线性层、线性混合层和密钥加层.其中,线性混合层确保多轮之上的高度扩散;非线性层将 SBox 并行化;密钥加层将单轮子简单地异或到中间状态上,实现一次性掩盖.此外,在

第一轮之前,应用了一个初始密钥加层,为了使攻击者无法从明文端剥去其它计算部件.而最后一轮的线性混合层与前面各轮的线性混合层不同.另外最后一轮不需要列变换^[4]。

3.1 有限域乘法实现与优化

有限域乘法主要用在混列转换(MixColumns)算法中,其定义如下:

$GF(2^m)$ 可视为所有小于 m 次的多项式的集合。

$$GF(2^m) = \{a_{m-1}x^{m-1} + \dots + a_1x + a_0 \mid a_i \in \{0,1\}\} \quad (1)$$

设 $GF(2^m)$ 上的两个元素为:

$$A(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \dots + a_1x + a_0$$

$$B(x) = b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \dots + b_1x + b_0 \quad (2)$$

定义它们的乘积为

$$\begin{aligned} P(x) &= A(x)B(x) \bmod F(x) \\ &= p_{m-1}x^{m-1} + p_{m-2}x^{m-2} + \dots + p_1x + p_0 \end{aligned} \quad (3)$$

因此,有限域上乘法的运算实质上是根据乘数,将被乘数不断移位模 2 加,对得到的乘积,再进行模运算,就可以得到最终乘法结果.当使用这种算法时,对 $GF(2^m)$ 需要 m 周期才能计算出结果.在 AES 加密的 MixColumns 运算中,乘数只有 2 和 3,对乘 2 计算,用上述公式可得到

$$\begin{aligned} b \cdot x &= b_7x^8 + b_6x^7 + \dots + b_0x \pmod{m(x)} \\ &= b_6x^7 + b_5x^6 + b_4x^5 + (b_3 \oplus b_7) + \\ &\quad b_1x^2 + (b_0 \oplus b_7)x + b_7 \end{aligned} \quad (4)$$

经简化后代码为

```
if (x < 0x80) // b7=0
```

```
    return (x << 1);
```

```
else
```

```
    return ((x<<1)^(0x1b));
```

乘 3 算法可类推为 $gfmultby02(x) \sim x$

经过优化后的乘法运算可以单周期实现。

3.2 字节转换(SubBytes)

SubBytes 是一个以字节为单位的非线性取代运算.通过找出每个字节在 $GF(2^8)$ 中的乘法反元素,并经过一个仿射(Affine)转换运算得到结果。

在硬件设计中,SubBytes 实际是查表操作^[5],由于每一轮有 128bit,按每 8bit 为地址寻址需 16 次,如果只有一个存放 SBox 的 ROM,仅 SubBytes

就需 16 周期,如果将其全部展开为 16 个 SBox,则只要单周期.对于 Catapult,可以在 Design Goal 里选择 Latency 或者 Area 来决定是否展开.

3.3 移行转换(ShiftRows)

在 ShiftRows 中,每一行以不同的偏移量做环状位移,第 0 行不动,第 n 行分别位移 n 个字节. ShiftRows 操作可以直接交换赋值顺序,即改变连线,不需要消耗任何资源.

3.4 混列转换(MixColumns)

在这个转换中,把状态列当作一个存在 $GF(2^8)$ 中的多项式.在模 $x^4 + 1$ 下与一个固定的多项式 $c(x)$ 作乘法.表示如下:

$$c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 03 \cdot x + 02 \quad (5)$$

$c(x)$ 与 $x^4 + 1$ 互质,令

$$b(x) = c(x) \cdot a(x) \bmod (x^4 + 1) \quad (6)$$

以矩阵乘法表示见(7)式:

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (7)$$

其中的乘法均为在 $GF(2^8)$ 上的伽罗华域乘法,其优化如 3.1 节所述.

3.5 轮密钥加(AddRoundKey)

这个运算主要是把每一个回合密钥(Round Key)简单异或到每一个状态列中.在硬件上直接使用逻辑单元实现.

3.6 密钥扩展(KeyExpansion)

扩充后的密钥是一个 4-byte 的线性数组,表示为 $W[4][Nb(Nr + 1)]$,前 Nk 个字组包含了加密密钥,第 i 轮的轮密钥 $ExpandedKey[i]$ 由 W 中的第 $Nb * i$ 列到第 $Nb * (i + 1) - 1$ 列给出.对于 128 位实现, $Nb = 4$.

对于每轮的 4 个密钥,有如下关系: $key[i] = key[i - 4] \cdot key[i - 1]$,而对第 $4n$ 个密钥, $key[4n] = key[4n - 1] \cdot SubWord(RotWord(key[4n - 1]) \cdot RCon)$,很容易导出递推关系(以最后一个为例): $key[i + 3] = key[i - 1] \cdot key[i - 2] \cdot key[i - 3] \cdot key[i - 4] \cdot SubWord(RotWord(key[i - 4] \cdot RCon))$.通过这个公式,就可以在一个周期内算出四个密钥,而且可以与加密运算并行执行^[6].

其代码如下:

```
KeyScheduleOut → set_slc(0, k1 ^ SubWord(RotWord
(temp ^ RCon));
```

```
KeyScheduleOut → set_slc(32, k2 ^ k1 ^ SubWord(Rot-
```

```
Word(temp ^ RCon));
```

```
KeyScheduleOut → set_slc(64, k3 ^ k2 ^ k1 ^ SubWord
(RotWord(temp ^ RCon));
```

```
KeyScheduleOut → set_slc(96, k4 ^ k3 ^ k2 ^ k1 ^ Sub-
Word(RotWord(temp ^ RCon));
```

将该函数需完成的操作按周期划分,可以分为读取 bit 和 KeySchedule, RotWord, SubWord, 异或, 其中 RotWord 只是移位,通过改变连线即可,不需时间.因此,耗费周期的操作主要是读取 S 盒和异或,所需时间与 SBox 的实现方式有关,图 1 和图 2 是几种实现方式的时序比较.其中图 1 使用的是 rom_regout,图 2 是 mgc_rom.

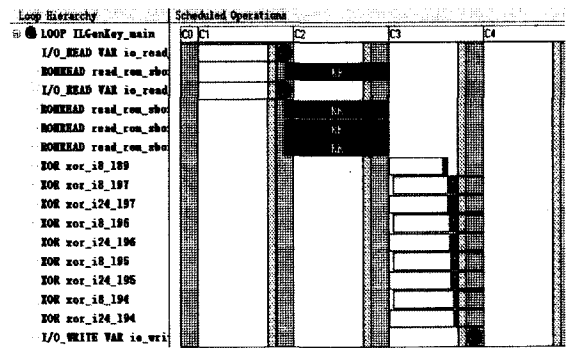


图1 rom_regout 方式实现 SBox 时序图

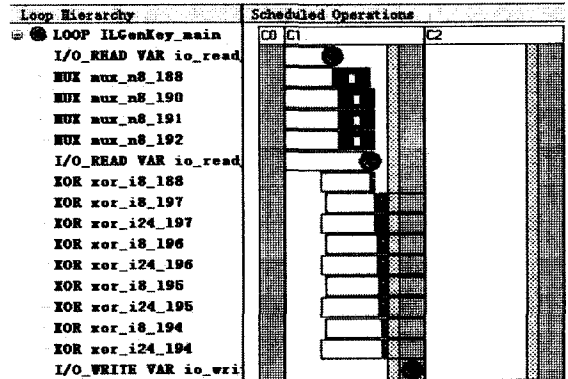


图2 mgc_rom 方式实现 SBox 时序图

从图中可看出,其差异主要体现在 S 盒读取延时上^[7],默认的 rom_regout 是时钟沿锁存输出,造成其他操作必须等待下一个时钟才能操作,而 mgc_rom 则不靠时钟,减少了延迟,从而能在单周期完成.

由上面的讨论可知,经过适当优化,每轮运算基本可以在单周期内完成,全部 10 轮运算需要 10 个周期.在最快时钟频率下运行时,可以近似达到 2GB 每秒的吞吐量.

4 设计空间搜索

对于一个给定的系统描述,其实现的方式是多

种多样的,将这些所有可能实现的方式集合在一起,即为该系统的设计空间.高层次综合的目标在于把一个以抽象的算法作为描述的系统,转换成一个具体的电路实现,并且这个实现应该是所有实现中最优的一个.这一过程可以看作为一个对设计空间的搜索过程^[8]:给定一个系统描述,根据给定的该系统相应的所有限制和性能目标,在设计空间内一定存在一个与之对应的最优解,设计的过程即是于解空间内搜索此最优解的过程.在 Catapult 中,这一过程主要是由设计者施加不同的约束来决定最终的实

现方式.

以下是不同的代码优化方式及不同的结构实现下资源与延时的比较,其中图 3 是具体数据.可以看出,未经优化的代码占用的面积最大为 22317,而最小面积只需 3112,两者相差 7.17 倍,最大延迟时钟周期为 415,而最小为 10,相差 41.5 倍.可以看出,代码和结构设计对最终结果影响极大,而结构约束甚至起到了和代码优化等同的重要性,通过不同的结构,面积和延时可随之改变,而较差的结构则会同时增大面积和延时.

Solution	Status	Total Area Score	Latency Cycles	Latency Time	Throughput Cycles	Throughput Time
Solution_1	Passed extract	22317.12	52	520.00	32	320.00
Solution_5	Passed extract	7229.79	31	310.00	31	310.00
Solution_10	Passed extract	7043.48	10	100.00	10	100.00
Solution_6	Passed extract	5290.51	415	4150.00	373	3730.00
Solution_4	Failed extract	3111.91	104	1040.00	103	1030.00
Solution_2	Failed allocate					
Solution_9	Failed allocate					
Solution_7	Passed analyze					
Solution_3	Failed allocate					
Solution_8	Failed allocate					

图 3 不同方案面积与延时数据

5 结束语

为了评测该 IP 核性能,在 Opencores.org 上选择了一款 AES IP 核作为参考,使用 Precision RTL Plus 2007a 对两者分别进行综合.器件选择 Straix II EP2S15F484C3,时钟约束 100MHz.以下是分别的综合结果.

(1) 传统 AES 核

Number of LUTs: 2262

Number of Registers: 663

Number of accumulated instances: 3199

ClockDomain0: 4.721 (211.820 MHz)

运算需 13 周期,其吞吐率为

$128\text{bit} \times 211.820\text{MHz}/13\text{Cycle} = 2085 \text{ MBits/s}$

(2) 使用 HLS 设计的 AES 核

Number of LUTs: 1562

Number of Registers: 407

Number of accumulated instances: 1969

ClockDomain0: 5.765 (173.461 MHz)

运算需 10 周期,其吞吐率为

$128\text{bit} \times 173.461\text{MHz}/10\text{Cycle} = 2224 \text{ MBits/s}$

可以看出,除了最高时钟频率,无论在资源占用还是数据吞吐率上,使用 Catapult Synthesis 综合出的 IP 核都接近或超越了手动编写的开源 IP 核,可以预计,HLS 技术必将取代 RTL 级的设计方法,成

为复杂 SOC 设计的主流.

参考文献:

- [1] Philippe Coussy, Adam Morawiec. High-level synthesis [M]. Berlin: Springer Science Business Media, 2008: 29-52.
- [2] Mentor Graphics. Catapult synthesis user's and reference manual [M]. Wilsonville: Mentor Graphics, 2007: 23-117.
- [3] Mentor Graphics. Catapult synthesis style guide [M]. Wilsonville: Mentor Graphics, 2007: 64-93.
- [4] 景为平,徐晨,陈海进. AES 密码算法的结构优化与实现[J]. 微电子学与计算机, 2007, 24(2): 36-38.
- [5] 高磊,戴冠中. AES 算法中 SubBytes 变换的高速硬件实现[J]. 微电子学与计算机, 2006, 23(7): 47-49.
- [6] Andreas Brokalakis, Athanasios P Kakarountas, Costas E Goutis. A high-throughput area efficient FPGA implementation of AES-128 encryption[J]. Signal Processing Systems Design and Implementation, 2005 (11): 116-121.
- [7] Mentor Graphics. Catapult synthesis C++ to hardware concepts [M]. Wilsonville: Mentor Graphics, 2007: 33-36.
- [8] Gu Zhengyu, Wang Jia. Dick Robert P, et al. Incremental exploration of the combined physical and behavioral design space [C]// Proceedings of the design of Automation Conference. Anaheim, 2005. (下转第 212 页)

处理图像的精确度. 由于本算法仅限于针对灰度图像的处理, 下一步应考虑将该模型拓展到彩色图像的应用中.



图 7 peppers 打印机模型结果图

参考文献:

- [1] Ulichney R A. Digital halftoning[M]. Cambridge, MA: MIT Press, 1987:233-331.
- [2] 江泽涛. 一种基于 HVS 和 EZW 的水印算法[J]. 微电子学与计算机, 2008, 25(2):48-50.
- [3] Ryu Byong Tae, Lee Jong. Resolution enhancement techniques for halftoned images[C]//Proceedings of SPIE - The International Society for Optical Engineering. San Jose, United States, 2007:1479-1483.
- [4] 梁若莹. 彩色喷墨绘图机分色技术与数字半调技术研究[D]. 西安: 西安电子科技大学, 2002:32-40.
- [5] Lee Je-Ho, Allebach Jan P. Inkjet printer model-based halftoning[C]//IEEE International Conference on Image Processing, Rochester, United States, 2002:3-4.
- [6] 叶玉芬. 自适应反馈视觉感知差的误差扩散半色调算法[J]. 微电子学与计算机, 2006, 23(6):155-159.
- [7] 孔月萍, 于乐兵. 基于数字半调技术的打印机模型[J]. 微计算机信息, 2008(6):257-259.
- [8] Tsi-YiChao, Hsueh-Ming Hang. Inverse halftoning of scanned images[J]. IEEE Transactions on Image Processing, 1995(3):420-423.

作者简介:

李智杰 男, (1980-), 硕士, 讲师. 研究方向为图形图像处理等.
 崔文竹 女, (1983-), 硕士研究生. 研究方向为图形图像处理等.
 刘欣 男, (1978-), 硕士, 助理工程师. 研究方向为图形图像处理等.
 李小莹 女, (1972-), 助理工程师. 研究方向为图形图像处理等.

(上接第 208 页)

作者简介:

孙桂玲 女, (1964-), 教授, 博士生导师. 研究方向为信息检测与智能控制系统、无线传感器网络.
 纪永鑫 男, (1985-), 硕士研究生. 研究方向为信息检测与

智能控制系统、无线传感器网络.

张潺潺 男, (1986-), 硕士研究生. 研究方向为信息检测与智能控制系统、无线传感器网络.

李维祥 男, (1947-), 教授, 博士生导师. 研究方向为信息检测与智能控制系统、无线传感器网络.