基于增强型延时感知 CSE 算法的 AES S 盒电路优化设计

戴 强,戴紫彬,李 伟
(解放军信息工程大学,河南郑州 450001)

摘 要: 针对高级加密标准(AES)S-盒优化,提出了一种增强型延时感知公共项消除(CSE)算法. 该算法能够在不同延时约束条件下优化多常数乘法运算电路,并给出从最小延时到最小面积全范围的面积-延时设计折中. 采用该算法优化了基于冗余有限域算术的 S 盒实现电路,确定了延时最优、面积最优的两种 S 盒构造. 实例优化结果表明所提出算法的优化效率高、优化结果整体延时小. 所设计的 S 盒电路基于 65nm CMOS 工艺库综合,结果表明,对比于已有文献中 S 盒复合域实现电路,所提出面积最优 S 盒电路的面积-延时积最小,比目前最小面积与最短延时的 S 盒组合逻辑分别减少了 17.58% 和 19.74%.

关键词: 高级加密标准 (AES); S 盒; 复合域; 延时感知公共项消除
中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112 (2019)01-0129-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.01.017

Construction of Optimum Circuit for AES S-Box Based on an Enhanced Delay-Aware Common Subexpression Elimination Algorithm

DAI Qiang, DAI Zi-bin, LI Wei
(*PLA Information Engineering University, Zhengzhou, Henan* 450001, *China*)

Abstract: Aiming at the optimization of advanced encryption standard (AES) S-box, an enhanced delay-aware common subexpression elimination algorithm is proposed. Under different delay constraints, the proposed algorithm can not only optimize multiple constant multiplication circuit, but also provide all of the design trade-offs, from the shortest feasible delay to the smallest area. Two constructions of S-box based on redundant finite field arithmetic which have optimal delay or the optimal area are derived using the algorithm. The results obtained through optimizing examples show the algorithm achieves high optimization efficiency and better overall delay optimization effect. In 65nm CMOS technology, the proposed S-box circuit which has the optimal area has the minimum area-delay product among the S-boxes based on composite field architecture. Compared with the smallest S-box and the shortest delay S-box, it saves about 17.58% and 19.74% of the area-delay product respectively.

Key words: advanced encryption standard (AES); S-box; composite fields; delay-aware common subexpression elimination

1 引言

S 盒运算电路是 AES 硬件实现[1]中资源消耗最多的部分,是 AES 硬件电路优化的关键[2]. 针对小面积、高性能 S 盒研究,众多文献基于正规基[3,4]、多项式基[5]、混合基[6]、冗余基[7,8]的复合域运算提出了多种设计方案. 相比于正规基、多项式基或混合基,基于冗余基的复合域 S 盒实现[7,8]的关键路径延时最短. 然而,文献[7]仅给出了基于冗余有限域算术的 S 盒实现个例,并未讨论多项式系数与映射矩阵对该 S 盒实现面积、延时的影响,且未对 S 盒电路实现进行优化. 为了保持该 S 盒电路关键路径延时短的优势,并在此基础上获得面积最优的 S 盒复合域实现电路,一方面需要确定合适的多项式系数与映射矩阵,另一方面需要对 S 盒电路进行

收稿日期:2017-12-25;修回日期:2018-05-16;责任编辑:李勇锋
基金项目:国家自然科学基金(No.61404175)

优化[9].

S 盒复合域实现电路中,可优化的主要部分是其中的多常数乘法(Multiple Constant Multiplication,MCM)运算电路,包括仿射矩阵、映射矩阵电路等. 对 MCM 电路的优化,常用公共项消除(Common Subexpression Elimination,CSE)算法[2]. 已有的 CSE 算法中,Exhaust-CSE 算法[3]、Greedy-CSE 算法[5]、MTP-CSE 算法[2,9]缺乏对电路关键路径延时的控制,NRCSE(Non-Recursive CSE)算法[10]、SPCSE(Shortest Path CSE)算法[11]的优化效率不高,DACSE(Delay Aware CSE)算法[12]的优化结果中各输出信号延时无法达到最优. 针对这些问题,本文提出了一种增强型延时感知 CSE 算法(Enhanced Delay Aware CSE,EDACSE),可在给定延时约束条件下实现 CSE 优化过程中对电路延时的控制,并能够获得满足延时约束条件下面积最优、各输出信号延时最优的优化结果. 实例优化结果表明 EDACSE 算法具有优化效率高、优化结果整体延时小的特点. 针对基于冗余有限域算术的 S 盒实现,利用 EDACSE 算法确定了使 S 盒组合逻辑电路延时最小与面积最小的两种 S 盒构造. 相比于已有的复合域 S 盒实现电路,设计的两种 S 盒电路在面积-延时积上具有明显优势.

2 基于冗余有限域算术的 S 盒实现

S 盒的复合域运算包括 GF(2^8)域乘法逆运算和仿射运算,如式(1)所示.

$$\boldsymbol{F}^{\mathrm{T}}=\boldsymbol{M}(\boldsymbol{\delta}^{-1}(\boldsymbol{\delta}\boldsymbol{X}^{\mathrm{T}})^{-1})+\boldsymbol{V}^{\mathrm{T}} \tag{1}$$

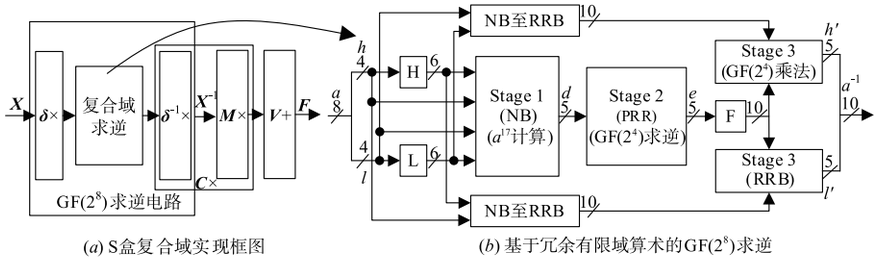
式中:上标 T 为向量的转置符,$\boldsymbol{X}$ 为输入向量,$\boldsymbol{M}$ 为仿射矩阵,$\boldsymbol{V}$ 为仿射运算过程的常向量,$\boldsymbol{F}$ 为 S 盒变换后输出向量,$\boldsymbol{\delta}$ 和 $\boldsymbol{\delta}^{-1}$ 为基于复合域乘法逆运算的映射矩阵和逆映射矩阵. 通常,将仿射运算矩阵和逆映射矩阵合并为一个矩阵 $\boldsymbol{C}$ 以简化电路结构,则 $\boldsymbol{C}=\boldsymbol{M}\boldsymbol{\delta}^{-1}$.

基于冗余有限域算术的复合域 S 盒[7]实现框图如图 1 所示. 图 1(a)中复合域求逆运算使用 Itoh-Tsujii 算法高效实现,其过程主要包括输入 a 的 16 次幂与 17 次幂计算、GF(2^4)求逆运算、GF(2^4)乘法运算三大部分[7],分别对应图 1(b)中 Stage1、Stage2、Stage3. 图 1(a)首先通过 $\boldsymbol{\delta}\times$ 完成域 GF(2^8)至复合域 GF((2^4)^2)的转换得到输入 a,其中复合域 GF((2^4)^2)上元素以正规基(Normal Base,NB)(α^{16},α)表示(α 为不可约多项式 $f(x)=x^2+\mu x+\nu$ 的根). 图 1(b)中 Stage1 利用域 GF(2^4)的正规基(β^3,β^4,β^2,β^1)表示(β 为四阶全一多项式 $g(x)=x^4+x^3+x^2+x+1$ 的根)计算输入 a 的 17 次幂,再将计算结果转换成多项式环表示(Polynomial Ring Representation,PRR)结果. Stage2 完成基于 PRR 的 GF(2^4)上元素求逆运算. Stage3 完成基于冗余表示基(Redundantly Represented Basis,RRB)的 GF(2^4)上元素乘法运算. H、L 与 F 模块用于实现 Stage1 与 Stage3 中的共享子表达式,从而减小硬件开销. 图 1(b)的输出 a^{-1} 为 RRB 表示,对其进行 RRB 至 NB 的转换后,再经过逆映射矩阵与仿射变换可得到最终 S 盒输出.

图 1 所示 S 盒实现中,(μ,ν)组合不同,a^{17} 计算的逻辑表达式与 $\boldsymbol{\delta}$、$\boldsymbol{C}$ 的值随之改变,而 Stage2 与 Stage3 的表达式并不会改变(其表达式可参见文献[7],不再详述),则整个 S 盒电路优化的关键在于随(μ,ν)组合取值不同而改变的 a^{17} 计算与 $\boldsymbol{\delta}\times$、$\boldsymbol{C}\times$ 运算电路. 本文将 a^{17} 计算与 $\boldsymbol{\delta}\times$、$\boldsymbol{C}\times$ 运算部分电路称为该 S 盒的可调节电路,并设其面积为 A_{adjust},则

$$A_{\mathrm{adjust}}=A_{\delta}+A_{C}+A_{a^{17}} \tag{2}$$

其中 A_{δ}、A_{C}、$A_{a^{17}}$ 分别表示 $\boldsymbol{\delta}\times$、$\boldsymbol{C}\times$、$a^{17}$ 计算各部分运算电路面积. 为获得面积最小的 S 盒构造,需选择合适的(μ,ν)组合与 $\boldsymbol{\delta}$、$\boldsymbol{C}$ 以使 A_{adjust} 最小. 为此,首先要通过 CSE 算法消除 MCM 电路中的冗余资源后,对比不同 $\boldsymbol{\delta}\times$、$\boldsymbol{C}\times$ 运算电路实现所需最少门数,从而获得实现门数最少的 $\boldsymbol{\delta}$、$\boldsymbol{C}$.



(a) S盒复合域实现框图 (b) 基于冗余有限域算术的GF(2^8)求逆

图1 基于冗余有限域算术的复合域S盒实现

3 EDACSE 算法

考虑计算速度与优化效率的折中,本文设计的 EDACSE 算法优先消除出现频率最高的 CS,并在消除后进行逻辑网络的延时评估. 为得到整体延时最好的优化结果,EDACSE 算法将逻辑网络各个输出信号的关键路径延时之和作为一个优劣评判标准. EDACSE 算法如下所示.

算法 1　EDACSE 算法

输入:逻辑网络表达式,给定延时约束 T_C;

输出:优化后的逻辑网络表达式;消除的公共项集合;优化后实现所需门数,优化后关键路径延时

S1:变量初始化. 初始化三个变量:$N_{\min}$(最小门数),$T_{\min}$(最短关键路径延时)与 C_{tra}(消除 CS 的集合). $N_{\min}$ 以用于矩阵直接实现的门数 N_{dir} 初始化. $T_{\min}$ 以一个大于 CS 消除后可能的关键路径延时上限的值初始化. 变量 C_{tra} 以空集初始化.

S2:识别所有逻辑表达式中的 CSs. 通过 CSE 技术消除 CSs,并且消除的 CSs 作为变量 C_{opt} 串行保存. 在选定一个 CS 进行约减后,计算该 CS 延时,并根据 CS 延时计算逻辑网络中各逻辑表达式的最短 CPD. 最短关键路径延时以 T_{CPD} 表示. 若 $T_{\text{CPD}} > T_C$,则撤销本次对该逻辑表达式的 CS 消除. 若整个网络中消除 CS 后逻辑表达式 CPD 不超过约束的个数 $n<2$,则撤销对整个逻辑网络的 CS 消除. 继续选择下一个 CS 进行消除.

S3:所有 CS 消除结束后,计算优化后逻辑表达式所需门数,记作 N_{opt};计算优化后逻辑网络的关键路径延时,记作 T_{opt};计算优化后逻辑网络各表达式的路径延时之和 Sum_T_{opt}.

S4:若 $N_{\min} > N_{\text{opt}}$,优化电路的参数在该步骤保存,即在该周期中获得的 N_{opt},T_{opt} 与 C_{opt},Sum_T_{opt},分别赋值给 $N_{\min}$、$T_{\min}$、C_{tra}、$Sum\_T_{\min}$,然后算法将转至步骤 7. 否则,算法将转至步骤 6.

S5:若 $N_{\min} = N_{\text{opt}}$:

若 $T_{\min} > T_{\text{opt}}$,则保存优化电路的参数,即在该周期中获得的 N_{opt}、T_{opt}、C_{opt} 分别赋值给 $N_{\min}$、$T_{\min}$、C_{tra},算法将转至步骤 7;

若 $T_{\min} = T_{\text{opt}}$ 且 $Sum\_T_{\min} > Sum\_T_{\text{opt}}$,则优化电路的参数在该步骤保存,即在该周期中获得的 N_{opt}、T_{opt}、C_{opt}、Sum_T_{opt} 分别赋值给 $N_{\min}$、$T_{\min}$、C_{tra}、Sum_T_{opt},否则算法跳转至步骤 7.

S6:若搜索结束,算法跳转至步骤 7 以输出由算法得到的最佳结果. 否则,算法返回步骤 2 以检查另一个可能的消除解决方案.

S7:最佳的结果 $N_{\min}$、$T_{\min}$、C_{tra} 作为算法输出,算法结束.

由于实际处理时矩阵更易被计算机程序处理,因此算法输入的原始逻辑表达式将转换为矩阵形式进行 CS 消除. 作为算法输入的延时约束 T_C,存在下限值 $T_{C\min}$. 对于 MCM 电路,假设初始输入信号延时均为 0Tx,则可根据最快二叉树(Fastest-Binary-Tree,FBT)结构计算延时约束下限,如式(3)所示.

$$T_{C\min} = T_{\text{FBT}} = \max_{i=0,\cdots,n-1}(T_i) = \max_{i=0,\cdots,n-1}(\lceil \log_2 N_i \rceil) \quad (3)$$

这里 i 表示逻辑网络中表达式的序号,N_i 表示各表达式所包含的信号数. 给定的延时约束需满足 $T_C \geqslant T_{C\min}$.

EDACSE 算法的步骤 S2 中,设消除的 CS 为 $x_p + x_q$,则设计中的 $x_p + x_q$ 由一个新变量 x_{new} 代替. x_{new} 的延时为 $t_{\text{new}} = \max(t_p, t_q) + 1$. 由于 CS 消除后逻辑表达式中各信号延时不一,采用延时驱动二叉树(Delay-Driven-Binary-Tree,DDBT)结构构造电路,以获得逻辑表达式的最短关键路径延时 T_{CPD}. T_{CPD} 计算公式如式(4)所示.

$$T_{\text{CPD}} = t_{\min} = \left\lceil \log_2 \sum_{i=1}^{n} 2^{d_i} \right\rceil \quad (4)$$

其中⌈⌉表示向上取整运算. 逻辑表达式中各变量 x_i 的延时为 d_i,则 T_{CPD} 为实现逻辑表达式 $y = x_1 + x_2 + x_3 + \cdots + x_n$ 的最小延时. 若整个网络中消除 CS 后逻辑表达式 T_{CPD} 不超过 T_C 的个数 $n<2$,则此时消除该 CS 将无法节省实现门数,因此撤销该 CS 消除,而 DACSE 算法仅判断整个逻辑网络的关键路径延迟是否超过 T_C,并不能判断消除部分 CS 能否节省实现门数,因此 DACSE 的优化效率将略低于 EDACSE. 另外,若 CSE 优化过程中存在多个实现门数相同的优化结果,EDACSE 算法将依据优化结果中各输出信号延时进一步选择延时与整体延时最小的结果,而 DACSE 算法则无此功能,因此 EDACSE 的整体延时优化效果将优于 DACSE,优化结果分析可见 5.1 小节.

4　基于 EDACSE 算法的可调节电路优化设计

(μ,ν) 组合不同,a^{17} 计算结果中各逻辑表达式实现的硬件复杂性不同. 采用事后统计的方法,在求出 120 种 (μ,ν) 组合下的 a^{17} 计算表达式后,统计计算结果 d 中 $d_0 \sim d_4$ 的逻辑表达式中参与异或运算的项数(简称为异或项数)最大值 $N_{X\max}$ 的取值,共存在 21、17、14、11 这四种取值. 由于 a^{17} 计算(即 Stage1)与乘法运算结果中存在公共项,本文将 a^{17} 计算(Stage1)与乘法运算部分(Stage3)设置为一个分组,并提取公共项进行化简. 化简过程中,进一步采用 OR 门替换优化策略($a \vee b = a + b + ab$)优化 a^{17} 计算结果逻辑表达式后发现:当 $N_{X\max}$ 为 11 或 14 时,a^{17} 计算结果实现所需门数为 15XOR + 10AND + 5OR 或 15XOR + 5AND + 10OR,关键路径延时为 $3T_X + T_A$ 或 $3T_X + T_O$;而当 $N_{X\max}$ 的取值为 17、21 时,化简后表达式实现所需门数远大于 15XOR + 10AND + 5OR 或 15XOR + 5AND + 10OR,且关键路径延时大于 $3T_X + T_A$ 或 $3T_X + T_O$. 根据式(2),欲使可调节电路的面积 A_{adjust} 最小,需要选择能够使 $A_{a^{17}}$、A_δ 与 A_C 之和最小的 (μ,ν) 组合与 $\boldsymbol{\delta}$、$\boldsymbol{C}$,因此在不同 $\boldsymbol{\delta}$、$\boldsymbol{C}$ 矩阵对实现所需门数相同或相差较小的情况下,应选择对应 (μ,ν) 组合下 a^{17} 计算结果表达式中 $N_{X\max}$ 为 11 或 14 的 $\boldsymbol{\delta}$、$\boldsymbol{C}$ 矩阵对.

给定 $\text{GF}((2^4)^2)$ 上不可约多项式 $f(x) = x^2 + \mu x + \nu$ 中 μ,ν 的值,映射矩阵使用文献[13]中算法构造. 对于确定的某一 (μ,ν) 组合,可生成 8 个映射矩阵. 存在 120 种 (μ,ν) 组合可使 $f(x) = x^2 + \mu x + \nu$ 为不可约多项式,因此共有 960 个可用映射矩阵及其对应的联合矩阵. 本文将 $\boldsymbol{\delta}$ 与其对应的 $\boldsymbol{C}$ 称为一个矩阵对. 为得到使 S 盒电路面积或延时最小的 $\boldsymbol{\delta}$、$\boldsymbol{C}$,穷举搜索了 960 个矩阵

对. 首先计算 960 个矩阵对直接实现后的延时,存在关键路径延时之和最小为 5Tx 的 16 个矩阵对,其中 $\boldsymbol{\delta}$ 实现关键路径为 2Tx,$\boldsymbol{C}$ 实现关键路径为 3Tx. 设置 $\boldsymbol{\delta}$ 实现延时约束为 2Tx,$\boldsymbol{C}$ 实现延时约束为 3Tx,表 2 列举了采用 EDACSE 算法优化这 16 个矩阵实现后所需面积、延时,其中总面积、总延时分别为矩阵对计算实现所需面积之和、延时之和,X 表示异或门面积,Tx 表示异或门延时.

表 1 矩阵对实现对应面积/时间复杂性

编号	总面积 ($\boldsymbol{\delta}$ 计算/$\boldsymbol{C}$ 计算)	延时和 ($\boldsymbol{\delta}$ 计算/$\boldsymbol{C}$ 计算)	$N_{X\max}$
1 – 8	36X (14X/22X)	5Tx (2Tx/3Tx)	14
9 – 16	36X (12X/24X)	5Tx (2Tx/3Tx)	17

表 1 中 16 个矩阵对实现的总面积均为 36X,延时和均为 5Tx,因此无法根据总面积、总延时确定最优的矩阵对. 为此进一步计算了各矩阵对对应(μ,ν)组合下 a^{17} 计算表达式中异或项数最大值 $N_{X\max}$,存在 8 个矩阵对对应 $N_{X\max}=14$,其余 8 个矩阵对对应 $N_{X\max}=17$,根据前文所述,应选择对应 $N_{X\max}=14$ 的矩阵对.

本文选择矩阵对 $\boldsymbol{\delta}_7$、$\boldsymbol{C}_7$,此时电路的整体延时最小,且在整体延时最小的多种 S 盒构造中,其电路实现面积最小. 本文将矩阵对为 $\boldsymbol{\delta}_7$、$\boldsymbol{C}_7$时的 S 盒结构称为构造一,即延时最优的 S 盒构造,其中 $\boldsymbol{\delta}_7=[0x9A,0x10,0x64,0x45,0xC2,0x1F,0x4C,0x99]$,$\boldsymbol{C}_7=[0x17,0x15,0x78,0xC7,0xBD,0xA5,0x8E,0xFA,0x6A,0xBB]$. 基于 EDACSE 算法对 $\boldsymbol{\delta}_7\times$ 与 $\boldsymbol{C}_7\times$ 运算逻辑表达式的优化结果分别如式(5)、(6)所示.

$$Y=\boldsymbol{\delta}_7\boldsymbol{X}=\begin{cases}y_0(@2)=x_4+x_8\\ y_1(@2)=x_9+x_2\\ y_2(@2)=x_2+x_5+x_{10}\\ y_3(@2)=x_1+x_7+x_8\\ y_4(@2)=x_6+x_7+x_8\\ y_5(@0)=x_5\\ y_6(@2)=x_3+x_5+x_{10}\\ y_7(@2)=x_0+x_9\end{cases}\quad\begin{cases}x_8(@1)=x_0+x_2\\ x_9(@1)=x_3+x_7\\ x_{10}(@1)=x_1+x_4\end{cases}\tag{5}$$

$$Y'=\boldsymbol{C}_7\boldsymbol{X}'=\begin{cases}y_0'(@3)=x_{10}'+x_{15}'\\ y_1'(@3)=x_0'+x_9'+x_{16}'+x_{17}'\\ y_2'(@3)=x_3'+x_5'+x_{15}'\\ y_3'(@3)=x_1'+x_3'+x_{14}'\\ y_4'(@3)=x_{13}'+x_{14}'\\ y_5'(@3)=x_1'+x_4'+x_{14}'\\ y_6'(@2)=x_{11}'+x_{16}'\\ y_7'(@3)=x_{10}'+x_{12}'+x_{17}'\end{cases}\quad\begin{cases}x_{10}'(@1)=x_0'+x_5'\\ x_{11}'(@1)=x_2'+x_7'\\ x_{12}'(@1)=x_4'+x_6'\\ x_{13}'(@1)=x_8'+x_9'\\ x_{14}'(@2)=x_{10}'+x_{11}'\\ x_{15}'(@2)=x_{12}'+x_{13}'\\ x_{16}'(@1)=x_1'+x_6'\\ x_{17}'(@1)=x_2'+x_3'\end{cases}\tag{6}$$

此时对应图 1 中 a^{17} 计算结果的逻辑表达式如式(7)所示.

$$\begin{aligned}d_0&=H_{1,2}\vee L_{1,2}+H_{3,4}\vee L_{3,4}+h_2\vee l_2+h_3l_3\\ d_1&=H_{1,2}\vee L_{1,2}+H_{1,3}L_{1,3}+h_3\vee l_3+h_4\vee l_4\\ d_2&=H_{1,3}\vee L_{1,3}+H_{1,4}L_{1,4}+H_{2,3}\vee L_{2,3}+h_4\vee l_4\\ d_3&=H_{1,4}\vee L_{1,4}+H_{2,3}\vee L_{2,3}+H_{2,4}L_{2,4}+h_1\vee l_1\\ d_4&=H_{2,4}\vee L_{2,4}+H_{3,4}L_{3,4}+h_1\vee l_1+h_2l_2\end{aligned}\tag{7}$$

S 盒构造一并非实现面积最小的结构. 为得到面积最小的 S 盒构造,设置映射矩阵实现延时约束与联合矩阵实现延时约束为 10Tx. 采用 EDACSE 算法对 960 个矩阵对进行优化后,获得了各矩阵对实现所需最少门数. 为使可调节电路实现面积最小,综合考虑矩阵对实现面积与 a^{17} 计算电路面积,最终本文选取的矩阵对为 $\boldsymbol{\delta}_S=[0xE8,0x26,0x02,0x73,0x48,0xE2,0x69,0xBB]$,$\boldsymbol{C}_S=[0x16,0xE8,0x61,0x3C,0xA3,0x10,0x45,0xE3,0xA7,0x11]$. 基于 EDACSE 算法对 $\boldsymbol{\delta}_S\times$ 与 $\boldsymbol{C}_S\times$ 逻辑表达式的优化结果分别如式(8)、(9)所示.

$$Y=\delta_S X=\begin{cases}y_0(@2)=x_1+x_8\\ y_1(@3)=x_5+x_{11}\\ y_2(@0)=x_6\\ y_3(@3)=x_0+x_{12}\\ y_4(@1)=x_8\\ y_5(@3)=x_9+x_{11}\\ y_6(@3)=x_2+x_4+x_{12}\\ y_7(@2)=x_0+x_2+x_7\end{cases}\quad\begin{cases}x_8(@1)=x_0+x_4\\ x_9(@1)=x_1+x_7\\ x_{10}(@1)=x_2+x_6\\ x_{11}(@2)=x_8+x_{10}\\ x_{12}(@2)=x_3+x_9\end{cases}\tag{8}$$

$$Y' = C_S X' = \begin{cases} y_0'(@3) = x_0' + x_{11}' + x_{12}' \\ y_1'(@3) = x_9' + x_{11}' \\ y_2'(@2) = x_1' + x_3' + x_{13}' \\ y_3'(@1) = x_{14}' \\ y_4'(@2) = x_0' + x_4' + x_{13}' \\ y_5'(@3) = x_7' + x_{11}' + x_{14}' \\ y_6'(@2) = x_2' + x_8' + x_{12}' \\ y_7'(@3) = x_8' + x_{11}' \end{cases} \begin{cases} x_{10}'(@1) = x_1' + x_2' \\ x_{11}'(@2) = x_5' + x_{10}' \\ x_{12}'(@1) = x_3' + x_7' \\ x_{13}'(@1) = x_6' + x_9' \\ x_{14}'(@1) = x_6' + x_8' \end{cases} \tag{9}$$

对应 a^{17} 计算公式如式(10)所示,关键路径延时为 3Tx + T_A或 3Tx + T_O,采用 EDACSE 算法消除冗余项后实现所需门数为 15XOR + 10AND + 5OR.

$$\begin{aligned} d_0 &= H_{1,3} \vee L_{1,3} + H_{3,4}L_{3,4} + h_1 l_1 + h_3 l_3 \\ d_1 &= H_{1,4} \vee L_{1,4} + H_{1,2}L_{1,2} + H_{2,3}L_{2,3} + h_2 l_2 \\ d_2 &= H_{2,4} \vee L_{2,4} + H_{1,3}L_{1,3} + h_0 l_0 + h_3 l_3 \\ d_3 &= h_1 \vee l_1 + H_{1,4}L_{1,4} + H_{2,3}L_{2,3} + H_{3,4}L_{3,4} \\ d_4 &= h_2 \vee l_2 + H_{1,2}L_{1,2} + H_{2,4}L_{2,4} + h_0 l_0 \end{aligned} \tag{10}$$

本文将矩阵对为 $\boldsymbol{\delta}_S$、$\boldsymbol{C}_S$时的 S 盒构造称为 S 盒构造二,此时 S 盒的实现面积最小. 自此,基于 EDACSE 算法确定了使 S 盒电路延时最小的 S 盒构造一与面积最小的 S 盒构造二.

5　优化结果及分析

5.1　EDACSE 优化效果分析

为有效对比 EDACSE 算法与已有延时感知 CSE 算法的优化效率,选择基于冗余有限域算术 S 盒实现中的 960 个联合矩阵(联合矩阵为 10 输入 8 输出的 XOR 逻辑网络),分别采用 EDACSE 算法、DACSE 算法[11]、SPCSE 算法[12]及 NRCSE 算法[10]进行优化,得出了不同延时约束条件下的优化效率平均值,如图 2 所示. 图 2 中 SPCSE 和 NRCSE 算法的优化效率固定,EDACSE、DACSE 的优化效率随延时约束增加而有所提高. 采用 EDACSE 和 DACSE 算法优化时,绝大多数的 XOR 逻辑网络在 Tc = 4Tx 时能够达到最小面积,而在 $Tc \geqslant$ 6Tx 之后,EDACSE 与 DACSE 优化效率基本不变. 图 2 中 EDACSE 算法的优化效率明显大于 SPCSE、NRCSE,与 DACSE 优化效率几乎持平,但在特定延时约束条件(3Tx、4Tx)下 EDACSE 优化效率略高于 DACSE.

为进一步说明 EDACSE 算法相比于 DACSE 算法的优势,本文以优化结果中各信号延时的平均值作为参考标准,对比 EDACSE 与 DACSE 算法优化结果的整体延时. 为此,计算了不同延时约束条件下 960 个联合矩阵优化结果各输出信号延时平均值,进而得到 960 个平均值的平均值,结果如图 3 所示.

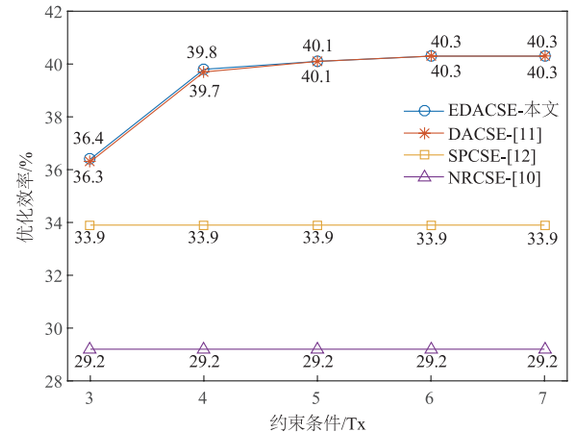


图2　各CSE算法对联合矩阵优化效率比较

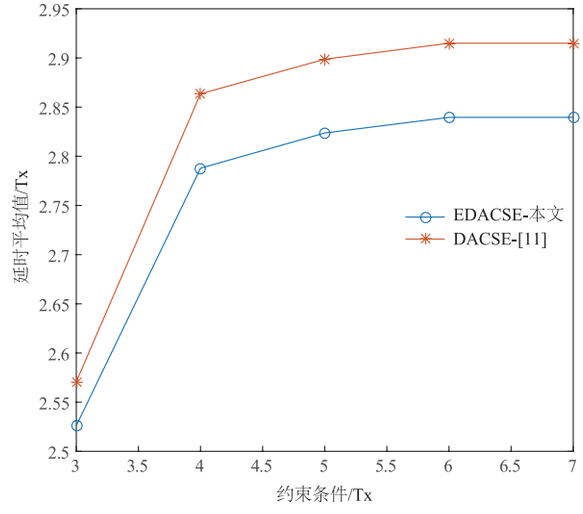


图3　整体延时效果对比

图 3 中,随着延时约束不断增加,EDACSE 与 DACSE 算法优化结果的各信号延时的平均值也不断增加,但在相同约束条件下 EDACSE 算法优化结果的各信号延时平均值明显小于 DACSE 算法,这表明 EDACSE 算法优化结果的整体延时比 DACSE 算法的优化结果小. 综上所述,对比于现有延时感知 CSE 算法,EDACSE 算法具有优化效率高、优化结果整体延时小的特点.

5.2　S 盒优化结果分析

5.2.1　理论计算

本文使用 EDACSE 算法对基于冗余有限域算术的 S 盒进行优化设计,得到了对应延时最小与面积最小的两种 S 盒电路结构. 优化后电路各部分的资源消耗与延时如表 2 所示,其中(g_X,g_A,g_O,g_N)分别表示异或门(XOR)、与门(AND)、或门(OR)、非门(NOT)的数量,(T_X,T_A,T_O)分别表示异或门延时、与门延时、或门延时数量.

表 2 S 盒电路各部分资源消耗与延时

方案	GF(2^8)域乘法逆		映射矩阵		联合矩阵		S 盒	
	资源消耗(g_X,g_A,g_O,g_N)	延时(T_X,T_A,T_O)	资源	延时	资源	延时	资源消耗(g_X,g_A,g_O,g_N)	延时(T_X,T_A,T_O)
文献[7]	(51,38,16,4)	(6,3,1)	18XOR	2Tx	38XOR	3Tx	(107,38,16,8)	(11,3,1)
构造一	(51,38,16,4)	(6,3,1)	14XOR	2Tx	22XOR	3Tx	(87,38,16,8)	(11,3,1)
构造二	(51,43,11,4)	(6,3,1)	13XOR	3Tx	17XOR	3Tx	(81,43,11,8)	(12,3,1)

设计的 S 盒与其它文献中 S 盒的理论计算面积与延时如表 3 所示. 为了直观对比各文献 S 盒实现的面积复杂性,表 3 展示了各文献中 S 盒实现面积的理论计算值. 假设设计均采用两输入逻辑门实现,在 65nm CMOS 工艺下,两输入异或门 XOR 的面积为 3.6μm^2,两输入与门 AND、或门 OR 的面积均为 1.8μm^2,两输入与非门 NAND、或非门 NOR 的面积均为 1.44μm^2,反相器 NOT 面积为 1.08μm^2,由此理论计算出 S 盒实现所需面积. 需要指出的是,表中 S 盒资源消耗均未包括仿射运算中常向量加操作所需资源. 由于逻辑门的延时随其驱动能力不同而不同,因此表 3 中未给出延时具体数值,但根据 $T_x > T_A = T_O$ 可对比出各文献 S 盒理论延时大小.

表 3 各文献 S 盒面积与延时信息对比

方案	资源消耗 (g_A,g_X,g_O,g_N)	面积理论计算值 (μm^2)	延时
文献[5]	(36,126,0,0)	518.40	$4T_A + 25T_X$
文献[13]	(35,120,0,0)	495.00	$4T_A + 19T_X$
文献[14]	(36,96,0,0)	410.40	$4T_A + 20T_X$
文献[2]	(35,93,0,0)	397.80	$3T_A + 20Tx$
文献[3]	(36,91,0,0)	392.40	$4T_A + 23Tx$
文献[4]	(35,89,0,0)	383.40	$3T_A + 18Tx$
文献[7]	(38,107,16,4)	486.72	$3T_A + T_O + 11T_X$
构造一	(38,87,16,4)	414.72	$3T_A + T_O + 11T_X$
构造二	(43,81,11,4)	393.12	$3T_A + T_O + 12T_X$

由表 3 可知,本文 S 盒构造一与文献[7]S 盒的延时最小,构造二 S 盒的延时次之,但 S 盒构造二的面积小于构造一与文献[7]. 文献[4]的面积值最小,且延时仅大于文献[7]与构造一、二. 文献[3]的面积值略大于文献[4],但其延时远大于文献[4]. 综合表 3 中数据,本文 S 盒构造一、二的延时具有明显优势,而 S 盒构造二的面积与文献[3,4]较为接近,若考虑面积-延时积,本文 S 盒构造一、二的面积延时积应大于表 3 中其它文献.

5.2.2 实际综合

本文采用 Verilog 语言对 S 盒电路进行描述,并利用综合工具基于 65nm CMOS 工艺标准单元库进行综合,综合时禁止 flatten 优化策略并设置面积优先. 选择表 3 中理论面积值较小的文献[3]与文献[4]的 S 盒设计,在相同条件下进行综合. 表 4 展示了文献[3,4]与本文 S 盒电路在最大与最小延时点时对应的延时与面积信息.

表 4 实际综合结果

方案	最小延时点		最大延时点	
	延时(ns)	面积(μm^2)	延时(ns)	面积(μm^2)
文献[3]	2.22	442.80	3.50	333.72
文献[4]	1.68	631.44	2.80	333.00
构造一	1.45	695.52	2.18	354.60
构造二	1.49	635.04	2.24	343.08

备注:最小延时点对应面积最大;最大延时点对应面积最小

表 4 中各文献间 S 盒面积、延时的大小关系与表 3 的理论分析结果一致. 表 5 对比了本文两种 S 盒电路、文献[3]及近两年文献[7,15]中复合域 S 盒设计的性能参数,其中文献[7]为表 3 中除本文 S 盒设计外的理论延时最小者,文献[15]为基于进化算法的最新复合域 S 盒设计. 由于文献[15]中给出的综合结果是在 180nm 工艺下获得,为公平比较,采用工艺换算计算公式[16]得到其在 65nm 下的等效面积、延时(表 5 中记作文献[15]*). 文献[7]与本文 S 盒均是基于冗余有限域算术实现,但文献[7]没有讨论(μ,ν)组合及$\boldsymbol{\delta}$、$\boldsymbol{C}$对 S 盒电路面积、延时的影响,且其给出的 S 盒电路实现并未进行 CSE 优化,因此表 5 中文献[7]的综合面积明显大于本文 S 盒.

表 5 各文献 S 盒性能参数对比

方案	工艺(nm)	延时(ns)	面积(μm^2)	面积-延时积 ($\mu m^2 \times ns$)
文献[3]	65	3.50	333.72	1168.02
文献[4]	65	2.80	333.00	932.40
文献[7]	65	2.18	439.20	957.46
文献[15]	180	6.54	2960.49	-
文献[15]*	65	2.36	386.05	911.08
构造一	65	2.18	354.60	773.03
构造二	65	2.24	343.08	768.50

表5 中文献[7]与文献[4]的 S 盒分别是目前已知延时最小与面积最小的 S 盒复合域实现. 本文构造一 S 盒的延时与文献[7]相同,但面积比文献[7]减少了19.26%. 构造二的 S 盒设计,延时比文献[7]增加了2.75%,但面积比文献[7]减少了 21.89%,其面积-延时积比文献[7]减少了 19.74%. 构造一、二的 S 盒面积比文献[4]分别增加了 6.49%、3.03%,但其延时比文献[4]分别减少了 22.14%、20.00%,且面积-延时积比文献[4]分别减少了 17.09%、17.58%. 根据表 5 中综合结果,本文面积最优 S 盒构造(即构造二)电路的面积-延时积最小,而本文构造一 S 盒电路的延时最小,且面积-延时积仅大于构造二.

6　结论

本文提出了增强型延时感知 CSE 算法 EDACSE,可在给定延时约束条件下实现 CSE 优化过程中对电路延时的控制,并且能够获得满足约束条件下面积最优、各输出信号延时最优的优化结果. 采用 EDACSE 算法对基于冗余有限域算术的复合域 S 盒实现进行优化,优化结果表明相比于现有延时感知 CSE 算法,EDACSE 算法能够更有效的消除电路冗余资源,具有优化效率高、优化结果整体延时小的特点. 基于 EDACSE 算法分别确定了使 S 盒组合逻辑电路延时最优与面积最优的两种 S 盒构造. 在 65nm CMOS 工艺下对两种 S 盒电路进行综合,结果表明延时最优 S 盒构造的面积-延时积,比最小面积[4]和最短延时[7]S 盒组合逻辑电路分别减少了 17.09% 和 19.26%;面积最优 S 盒构造的面积-延时积,比最小面积[4]和最短延时[7]S 盒组合逻辑电路分别减少了 17.58% 和 19.74%.

参考文献

[1] Banik S, Bogdanov A, Regazzoni F. Atomic-AES: A compact implementation of the AES encryption/decryption core [A]. Proceedings of the 17th International Conference on Cryptology in India [C]. Kolkata: Springer, 2016. 173 – 190.

[2] 曾纯,吴宁,张肖强,等. 基于多因子 CSE 算法的 AES S 盒电路优化设计[J]. 电子学报, 2014, 42(6): 1238 – 1243.
ZENG Chun, WU Ning, ZHANG Xiao-qiang, et al. The optimization circuit design of AES S-box based on a multiple-term common subexpression elimination algorithm[J]. Acta Electronica Sinica, 2014, 42(6): 1238 – 1243. (in Chinese)

[3] Canright D. A Very Compact Rijndael S-box[R]. California: Naval Postgraduate School, 2005.

[4] ZHANG Xiao-qiang, WU Ning, ZHOU Fang, et al. Optimization of area and delay for implementation of the composite field advanced encryption standard S-box[J]. Journal of Circuits, Systems, and Computers, 2016, 25(5): 1 – 29.

[5] A Satoh, S Morioka, K Takano, et al. A compact Rijndael hardware architecture with S box optimization[A]. Colin Boyd. Lecture Notes in Computer Science[C]. Australia: Springer Berlin Heidelberg, 2001. 239 – 254.

[6] Y Nogami, K Nekado, T Toyota, et al. Mixed bases for efficient inversion in F((22)2)2 and conversion matrices of SubBytes of AES[A]. Proceedings of 12th International Workshop on Cryptographic Hardware and Embedded Systems[C]. Santa Barbara: Springer, 2010. 234 – 247.

[7] R Ueno, N Homma, Y Sugawara, et al. Highly efficient GF (2^8) inversion circuit based on redundant GF arithmetic and its application to AES design[A]. Proceedings of 17th International Workshop on Cryptographic Hardware and Embedded Systems [C]. Saint-Malo: Springer, 2015. 63 – 80.

[8] R Ueno, S Morioka, N Homma, et al. A high throughput/gate AES hardware architecture by compressing encryption and decryption datapaths[A]. Proceedings of 18th International Workshop on Cryptographic Hardware and Embedded Systems [C]. Santa Barbara: Springer, 2016. 538 – 558.

[9] ZHANG Xiao-qiang, Ning WU, YAN Gai-zhen, et al. Hardware implementation of compact AES S-box [J]. IAENG International Journal of Computer Science, 2015, 42(2): 125 – 131.

[10] M Martínez-Peiró, E I Boemo, L Wanhammar. Design of high-speed multiplierless filters using a nonrecursive signed common subexpression algorithm [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2002, 49(3): 196 – 203.

[11] ZHANG Xiao-qiang, WU Ning, ZHOU Fang, et al. An optimized delay-aware common subexpression elimination algorithm for hardware implementation of binary-field linear transform [J]. IEICE Electronics Express, 2014, 11(22): 1 – 8.

[12] 张肖强. 基于复合域运算的 AES 密码电路优化设计方法研究[D]. 南京:南京航空航天大学, 2016.
Xiaoqiang Zhang. Research on Optimization Design Method of AES Implementation Based on Composite Field Arithmetic[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2016. (in Chinese)

[13] X Zhang, KK Parhi. On the optimum constructions of composite field for the AES algorithm[J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2006, 53(10): 1153 – 1157.

[14] M M Wong, M L D Wong, A K Nandi, et al. Construction

of optimum composite field architecture for compact high-throughput AES S-boxes[J]. IEEE Transactions on VLSI Systems,2012,20(6):1151 – 1155.

[15] LIU Yao-ping, WU Ning, ZHANG Xiao-qiang, et al. A compact implementation of AES S-box using evolutionary algorithm[J]. Chinese Journal of Electronics, 2017, 26(4):688 – 695.

[16] L Bin. Parallel AES encryption engines for many-core processor arrays[J]. IEEE Transactions on Computers, 2013,62(3):536 – 547.

作者简介

戴 强 男,1991 年生于江西乐安. 信息工程大学博士生,主要研究方向为安全专用芯片设计、密码硬件故障检测与容忍、可重构计算.

E-mail:xierunyan123@163.com

戴紫彬 男,1966 年生于河南商丘. 信息工程大学教授,博士生导师. 研究方向为专用芯片设计、可重构芯片、可重构 SoC 设计.

李 伟 男,1983 年生于天津. 博士,副教授,主要研究方向可重构计算、密码处理器研究.