

****

实验二、组合逻辑设计——SM4轮函数实现

2023年 11月 9日

**学号：20231023**

**姓名：龙熙**

**网络空间安全学院**

**目录**

[实验二、组合逻辑设计——SM4轮函数实现（实验指导书部分） 3](#_Toc150277193)

[1 实验指导 3](#_Toc150277194)

[1.1 背景知识——单钥分组密码 3](#_Toc150277195)

[1.1.1 密码体制的定义 3](#_Toc150277196)

[1.1.2 分组密码 4](#_Toc150277197)

[1.2 国家密码算法SM4 4](#_Toc150277198)

[1.3 基本实验要求 7](#_Toc150277199)

[1.4扩展要求 9](#_Toc150277200)

[1.5 提示 9](#_Toc150277201)

[实验二、组合逻辑设计——SM4轮函数实现（实验报告部分） 11](#_Toc150277202)

[2 实验报告 11](#_Toc150277203)

[2.1 实验背景与需求分析 11](#_Toc150277204)

[2.2 系统设计 11](#_Toc150277205)

[2.2.1 总体设计思路 11](#_Toc150277206)

[2.2.2 接口设计 11](#_Toc150277207)

[2.2.3 XXX模块 11](#_Toc150277208)

[2.2.4 YYY模块 11](#_Toc150277209)

[2.3 功能仿真测试 12](#_Toc150277210)

[2.3.1 测试程序设计 12](#_Toc150277211)

[2.3.2 功能仿真过程 12](#_Toc150277212)

[2.3.2 实验关键结果及其解释 12](#_Toc150277213)

[2.4 设计实现 12](#_Toc150277214)

[2.4.1 综合过程 12](#_Toc150277215)

[2.4.2 实验关键结果及其解释 12](#_Toc150277216)

[2.5 小结 12](#_Toc150277217)

[参考文献 12](#_Toc150277218)

[附录A 系统参数值（十六进制表示） 13](#_Toc150277219)

[附录B 固定参数数据表（十六进制表示） 13](#_Toc150277220)

[附录C SM4算法的S盒数据（十六进制表示） 13](#_Toc150277221)

# 实验二、组合逻辑设计——SM4轮函数实现（实验指导书部分）

## 1 实验指导

通过实现SM4加密算法的轮函数部分，利用ALINX XILINX Artix7 A7 XC7A35T HDMI实验硬件实验平台（以下简称“实验板”）通过Verilog HDL语言实现可综合代码，加深对组合逻辑设计理念的认识。

### 1.1 背景知识——单钥分组密码

#### 1.1.1 密码体制的定义

密码体制的语法定义如下：

* 明文消息空间：所有可能的明文集合。
* 密文消息空间：所有可能的密文集合。
* 加密密钥空间：所有可能的加密密钥集；解密密钥空间：可能的解密密钥集。
* 有效的密钥生成算法ζ：。
* 有效的加密算法：。
* 有效的解密算法：。

对于和，将加密变换表示为

读作“是在密钥下的加密”；

将解密变换表示为

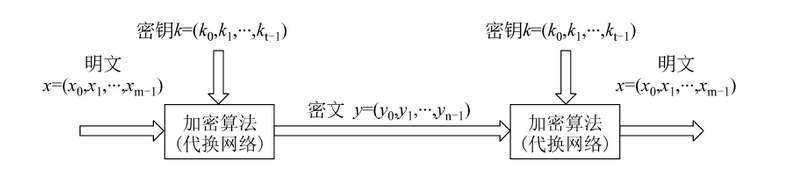
读作“是在密钥下的解密”。对于所有的和所有的，一定存在，使得。

密钥体制主要分为单钥密码体制（对称加密）与双钥密码体制（非对称加密）。在单钥密码体制中，加密和解密使用同样的密钥，即。这意味着加密方必须与解密方分享密钥。常见的单钥密码有：DES,AES。在公钥密码体制中，加密和解密使用不同的密钥，对于每个，存在，这两个密钥不同，但互相匹配；加密密钥（公钥）不必保密，解密密钥（私钥）必须保密。公钥用于加密数据，私钥用于解密数据。常见的双钥密码体制有：RSA。

#### 1.1.2 分组密码

在许多密码系统中，单钥分组密码是系统安全的一个重要组成部分。分组密码易于构造拟随机数生成器、流密码、消息认证码和杂凑函数等，还可进而成为消息认证技术、数据完整性机构、实体认证协议以及单钥数字签名体制的核心组成部分。

分组密码是将明文消息编码表示后的数字序列，划分成长为的组，各个分组分别在密钥控制下变换成等长的输出数字序列，其加密函数，是维矢量空间，为密钥空间，如图 1所示。它与流密码的不同之处在于输出的每一位数字不是只与相应时刻输入的明文数字有关，而是与一组长为的明文数字有关。在相同的密钥下，分组密码对长为的输入明文组所实施的变换时等同的，所以只需研究对任一组明文数字的变换规则。这种密码实质上是字长为的数字序列的代换密码。



注：图片来自《网络安全——技术与实践（第3版）》

图 1 分组密码框图

现代分组密码算法形式特点如下：

* 明文数据按定长进行分组，对每组使用算法进行加密；
* 维持相对稳定的算法密钥；
* 运算简单，便于在软硬件平台实现；

### 1.2 国家密码算法SM4

美国在1997年确定DES分组密码算法作为数据加密的通用标准。随着现代信息技术的进步，DES已无法满足人们对数据安全的要求，NIST公司在全球范围内征集AES的新标准。随着AES标准的确定，分组密码的研究也进入AES时代，诸如SEED、Camellia等优秀的密码算法随之出现。中国也在2006年公开发布了商用分组密码——SM4。

我国国家密码管理总局在2012年3月将SM4分组密码算法与SM2、SM3、祖冲之ZUC序列密码算法作为国家密码标准发布。SM4分组密码算法作为我国完全独立自主设计的商用密码算法，其核心技术由国家部门所掌握，使得我国在分组密码加密技术中不再受制于外国，进一步确保我们国家在无线通信领域的信息安全。

SM4算法采用非平衡Feistel设计结构，数据按128比特定长分组，按32比特为一字作为迭代运算的最小单位，结合32轮函数的迭代运算，实现分组数据的加密。该算法的迭代函数结构与AES算法相似，都是基于S盒的非线性变换运算和比特移位的线性变换运算，32组轮密钥通过异或运算实现中间数据的加密处理。

SM4的算法核心分为两大部分，其一是数据处理的加解密算法，另一部分则是提供用于32组轮密钥的密钥拓展算法，两者结构相似，密钥拓展算法同样处理128比特的初始密钥输入，通过与加解密类似的迭代结构，用32轮函数迭代生成32组轮密钥，用于加解密模块的函数处理。从结构上来看，SM4相较于 AES的算法结构更为简单，加密、解密、密钥拓展结构基本相同，在实际的FPGA硬件设计中，有利于硬件模块的迭代复用，有效降低硬件资源的占用，适合于资源有限的硬件设备。SM4算法结构如图 2所示。



图 2 SM4算法加解密（左）与密钥拓展（右）流程

（1）算法定义

本文中涉及SM4算法的符号定义如下所述：令符号表示32比特的向量集合，32比特为一字，是SM4中迭代运算的最小单位；令符号表示8比特的向量集合，为一字节；记非线性变换S盒运算为*Sbox*(.)；按位异或运算表示为符号；符号<<<表示32比特的循环移位运算。

128比特明文分组输入，记为；

128比特密文分组输出，记为；

128比特算法初始密钥，记为；

SM4密钥拓展运算定义参数：

系统参数；

固定参数；

（2）轮函数运算

SM4算法中包含了32轮的函数迭代运算，每一轮迭代函数中都包含了线性变换*L*和非线性变换。令符号*F*(.)表示轮函数，其运算公式为：

|  |  |  |
| --- | --- | --- |
|  |  |  |

*F*(.)主要由合成置换*T*构成，其通过线性变换*L*和非线性变换来实现分组密码的扩散原则，将*L*与的合成运算记为，该运算的具体过程如下所示：

1）非线性变换，该运算由4组并行的S盒置换构成，其运算流程为：

|  |  |  |
| --- | --- | --- |
|  |  |  |

式中符号代表该变换的输入，符号代表非线性变换的输出。

2）线性变换*L*，该变换由四组并行的循环移位运算构成，线性变换*L*的输入来自于非线性变换的输出，该变换的运算流程为：

|  |  |  |
| --- | --- | --- |
|  |  |  |

式中符号为线性变换*L*的输入，符号为变换输出。

（3）加/解密算法

SM4加/解密算法由32轮的轮函数*F*(.)迭代构成，每一轮的函数迭代需要通过密钥拓展得到的对应轮密钥参与运算，加密算法的运算流程如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

经过32轮的迭代运算，最后四轮输出的变换数据经由逆序变换*R*后输出得到最终的密文数据：

|  |  |  |
| --- | --- | --- |
|  |  |  |

SM4算法的解密运算与加密结构相似，区别仅在于轮密钥的使用顺序上与加密运算相反，因此加密运算最后的逆序变换*R*是为了确保后续解密运算时数据顺序的一致性。

（4）密钥扩展(Key Expansion)算法

SM4算法的32轮函数迭代需要32组轮密钥参与运算，这些轮密钥由密钥扩展算法对初始密钥的拓展运算生成，密钥扩展的算法结构与SM4的加密运算相似，区别在于密钥扩展运算中的线性变换*L*的移位运算有所不同，该算法的线性变换记为：

|  |  |  |
| --- | --- | --- |
|  |  |  |

非线性变换与加密算法保持一致，因此将与的合成运算记为，密钥拓展算法的函数迭代流程为：

|  |  |  |
| --- | --- | --- |
|  |  |  |

式中数据*K*初始化经由初始密钥*MK*和系统参数*FK*异或运算后得到：

|  |  |  |
| --- | --- | --- |
|  |  |  |

通过上述对SM4算法的结构分析，SM4算法的基本运算主要由S盒的非线性变换、循环移位的线性变换以及异或的逻辑运算构成，在硬件设计上较适合于查找表、位运算的方式进行实现，非常适合于FPGA的硬件平台。

### 1.3 基本实验要求

基本实验要求为：

1. 在硬件编程前，先利用软件编程实现SM4的加解密的轮函数部分（流程图如图 3虚线框内所示）与密钥扩展的轮函数（流程图如图 4虚线框内所示）部分，以确保后续实现的两种组合逻辑电路的正确性；（语言不限）
2. 用硬件编程实现SM4加解密算法中的轮函数部分与密钥扩展算法的轮函数部分；
3. 分别对实现的两种轮函数的组合逻辑电路进行功能仿真，对给定的输入电路应有正确的输出；
4. 对两种轮函数电路进行综合(Synthesis)，不要求实现(Implementation)，截图证明自己编写的Verilog代码可综合。

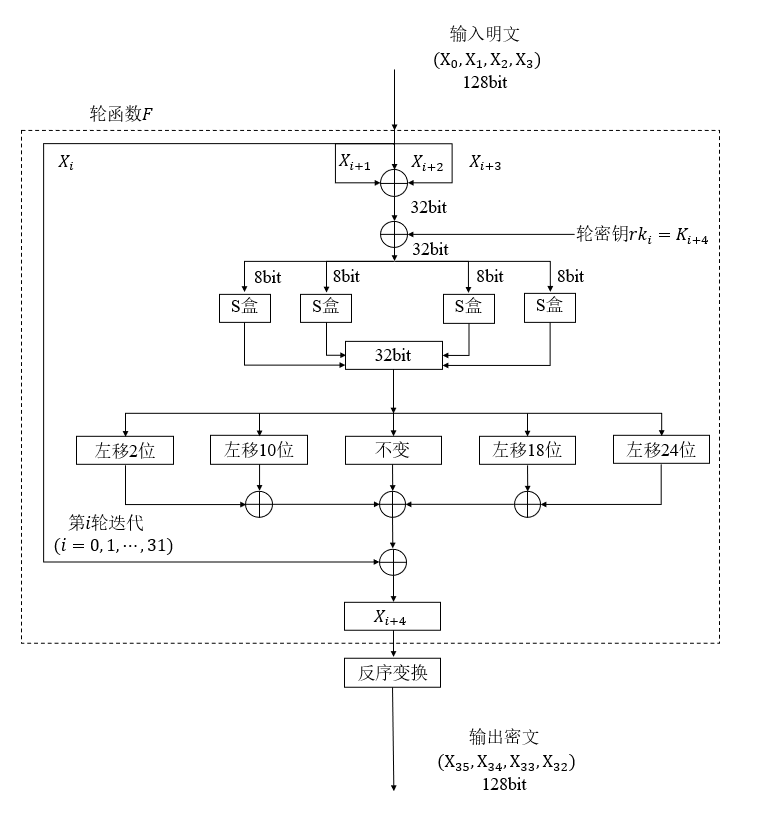


图 3 加解密算法中的轮函数流程图

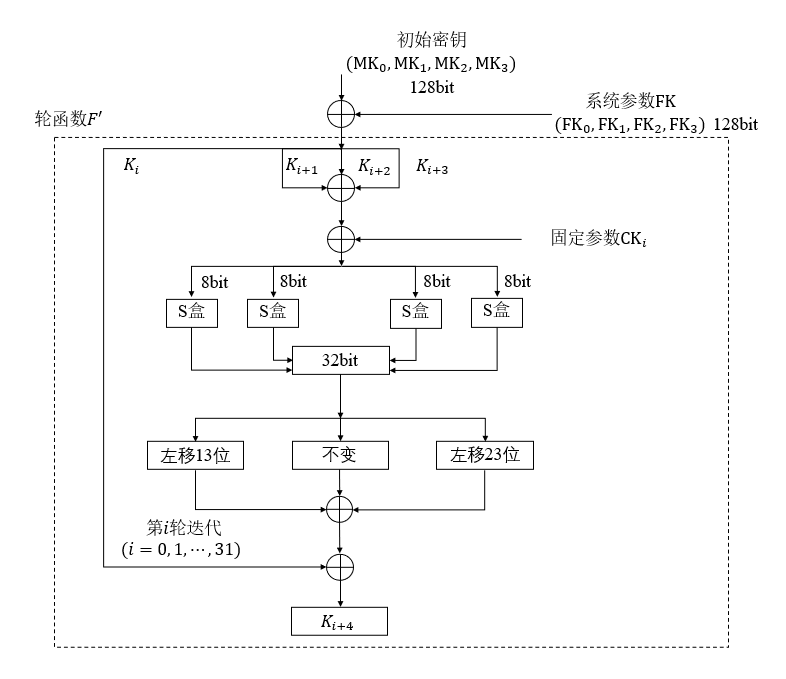


图 4 密钥拓展算法中的轮函数

本实验将采取实验报告和现场汇报演示相结合的形式。

实验报告提交链接：<https://bhpan.buaa.edu.cn/link/AAF8F768AD38F34CF686F47462539C71B7>

**实验报告提交截止期限：2023-11-30 23:59**

实验报告格式与命名：文档内格式请参考本文档，报告类型为PDF，命名规则为：姓名\_学号\_第二次实验；

现场汇报演示时间：请注意任课老师的通知。

### 1.4扩展要求

（根据完成的程度可酌情提高评价的等级）

通用异步接收器/发送器（Universal Asynchronous Receiver/Transmitter，简称UART），是最常用的设备间通信协议之一。在实验开发板上包含了USB-UAR 芯片, USB 接口采用 MINI USB 接口，可以用一根 USB 线将它连接到上 PC 的 USB 口进行串口数据通信。

扩展实验要求：改造附件样例（第二次实验/串口相关/UART例子/UART简单使用例子\_基于仿真/Uart\_demo.zip）中的构建的UART（或找到更好的UART程序）；利用UART程序，将自己编写的Verilog代码实现并下载到“实验板”上，形成UART交互人机接口，进行轮函数的输入输出，如图5。



图 5

### 1.5 提示

**基本实验要求提示：**

组合逻辑在程序实现中，请权衡代码的封装和复杂性的代价，灵活地以连续赋值或过程块实现，必要地时候采用模块封装。

可使用以下结果测试自己的代码（均为16进制表示，数据均为小端格式，即高位数据存放在高地址，所有数据最左侧为最高位）：

对加解密算法中的轮函数：

|  |  |  |
| --- | --- | --- |
| 输入明文 128bit | 轮密钥 32bit | 输出密文 32bit |
| 0123456789abcdeffedcba9876543210 | 01abcdef | 2ed91e3c |
| 30313233343536373839616263646566 | ababcdcd | 3597c003 |
| fe8710abe72bc8d9108abde810374deb | 12344321 | 78550171 |

对密钥拓展算法中的轮函数：

|  |  |  |
| --- | --- | --- |
| 输入密钥128bit | 固定参数 32bit | 输出密钥 32bit |
| 0123456789abcdeffedcba9876543210 | 00070e15 | 11d2c6bf |
| e6887b77dbabb572ffa07fed7548b192 | 1c232a31 | 143a2f94 |
| 09876543212456abd7ef9310387abde6 | 383f464d | 76d1b955 |

**SM4算法实现提示：**

SM4算法官方文档：第二次实验/SM4/附件一：GMT 0002-2012 SM4分组密码算法.pdf

博客资料：https://blog.csdn.net/weixin\_34189116/article/details/93217696

**拓展实验提示：**

在第二次实验附件中，有关拓展实验内容部分的附件为（第二次实验/串口相关/UART例子/UART简单使用例子\_基于仿真/）以及生成仿真信号代码的脚本（/第二次实验/串口相关/UART信号生成——用于仿真/）

UART的样例源代码具有固定的9600波特率、8位数据位、偶校验、1位停止位，缺点是一个只能每次读写一个字节的；更好的UART（注：更稳定可靠、波特率可配置、有Telnet等上层协议）需要用户自行开发，或者在网上搜索更好的代码加以改造。

SM4算法中的系统参数、固定参数、S盒部分请参考附录。

“实验板”的I/O接口和人机接口资源请参考北航网盘“第一次实验/AX7035开发板用户手册REV1.1.pdf”。

# 实验二、组合逻辑设计——SM4轮函数实现（实验报告部分）

## 2 实验报告

### 2.1 实验背景与需求分析

Verilog实现SM4加解密算法：加/解密轮函数，密钥扩展轮函数。

### 2.2 系统设计

#### 2.2.1 总体设计思路

python实现SM4代码，用于生成正确结果，用于后续verilog实现的仿真调试。

先实现S盒替代模块。加/解密模块、密钥扩展模块实例化调用S盒替换模块，按照标准文档写出32轮for循环即可。

#### 2.2.2 接口设计

S盒替代 s\_box.v

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| clk | I | 时间信号 |
| in | I | 输入 |
| out | O | 输出 |

SM4算法 SM4.v

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 功能描述 |
| clk | I | 时间信号 |
| plain | I | 明文 |
| op | I | 1：加密 0：解密 |
| MK | I | 秘钥 |
| cipher | O | 密文 |

#### 2.2.3 模块设计

s\_box.v

|  |
| --- |
| module SM4(  input [7:0] in,  input clk,  output [7:0] out,  );  case(in)  8'h00: out = 8'hd6;  8'h01: out = 8'h90;  8'h02: out = 8'he9;  8'h03: out = 8'hfe;  8'h04: out = 8'hcc;  8'h05: out = 8'he1; …….. |

sm4.v

先for循环轮秘钥生成32轮轮函数，再利用for循环加解密。

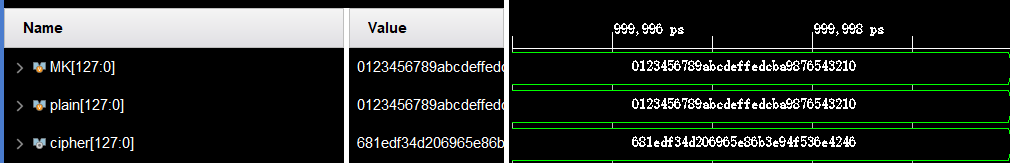
### 2.3 功能仿真测试

#### 2.3.1 测试程序设计

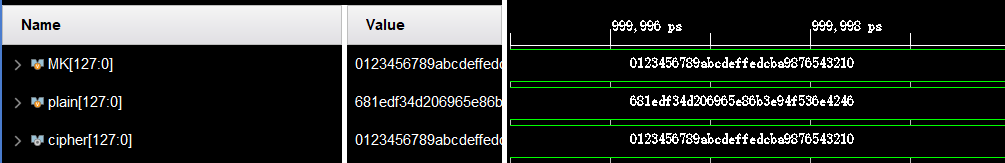
|  |
| --- |
| module SM4\_sim;  reg [127:0] MK;  reg [127:0] plain;  wire [127:0] cipher;    SM4 uut(  .MK(MK),  .cipher(cipher),  .plain(plain)  );    initial begin  MK = 0'h0123456789abcdeffedcba9876543210;  plain = 0'h0123456789abcdeffedcba9876543210;;  end  endmodule |

#### 2.3.2 功能仿真过程

加密

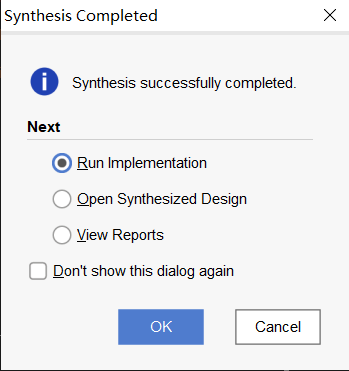


解密



### 2.4 设计实现

#### 2.4.1 综合过程



### 2.5 小结

## 参考文献

## 附录A 系统参数值（十六进制表示）

## 附录B 固定参数数据表（十六进制表示）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00070e15 | 1c232a31 | 383f464d | 545b6269 |
|  | 70777e85 | 8c939aa1 | a8afb6bd | c4cbd2d9 |
|  | e0e7eef5 | fc030a11 | 181f262d | 343b4249 |
|  | 50575e65 | 6c737a81 | 888f969d | a4abb2b9 |
|  | c0c7ced5 | dce3eaf1 | f8ff060d | 141b2229 |
|  | 30373e45 | 4c535a61 | 686f767d | 848b9299 |
|  | a0a7aeb5 | bcc3cad1 | d8dfe6ed | f4fb0209 |
|  | 10171e25 | 2c333a41 | 484f565d | 646b7279 |

## 附录C SM4算法的S盒数据（十六进制表示）

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

提示：S盒的输入可以表示为XX（注意颜色，绿色表示行数，蓝色表示列数），硬件实现时可使用case语句（可将表格粘贴至CSV文件（Excel打开）中，再用TXT格式打开可得到较好处理的数据）。

示例：，