第35卷 第24期 计 算 机 工 程 2009年12月
Vol.35 No.24 Computer Engineering December 2009
·安全技术· 文章编号：1000—3428(2009)24—0162—03 文献标识码：A 中图分类号：TP309

基于FPGA的16位数据路径的AES IP核

张新贺，张月华，刘鸿雁
(辽宁科技大学电子与信息工程学院，鞍山 114051)

摘 要：提出一种基于FPGA的16位数据路径的高级加密标准AES IP核设计方案。该方案采用有限状态机实现，支持密钥扩展、加密和解密。密钥扩展采用非并行密钥扩展，减少了硬件资源的占用。该方案在Cyclone II FPGA 芯片EP2C35F484上实现，占用20 070个逻辑单元(少于60%的资源)，系统最高时钟达到100 MHz。与传统的128位数据路径设计相比，更方便与处理器进行接口。
关键词：高级加密标准；IP核；加密

16-bit Datapath AES IP Core Based on FPGA

ZHANG Xin-he, ZHANG Yue-hua, LIU Hong-yan
(School of Electronic and Information Engineering, Liaoning University of Science and Technology, Anshan 114051)

【Abstract】This paper presents an architecture for 16-bit datapath Advanced Encryption Standard(AES) IP core based on FPGA. It uses finite state machine, and supports encryption, decryption and key expansion. The round-key is calculated before the beginning of encryption/decryption. It consumes less hardware resources. It is implemented on Cyclone II FPGA EP2C35F484, which consumes 20 070 logic elements, less than 60% of the resources. The IP core can operate at a maximum clock frequency of 100 MHz. Compared with 128-bit datapath AES, it can interface with CPU easily.
【Key words】Advanced Encryption Standard(AES); IP core; encryption

1 概述

2000年10月，NIST选择rijndael代替DES(Data Encryption Standard)，成为高级加密标准(Advanced Encryption Standard, AES)。AES算法是当前密码算法设计最高水平的反映，具有抗攻击能力强、密钥建立时间短、灵敏性高和内存需求低等特点。AES适合硬件实现。国外的一些公司如Helion Technology Limited, Dillon Engineering General Company Information推出了AES IP核[1-2]，GMU大学和一些研究机构也推出了AES IP核[3]。但这些IP核的数据路径都为128位，很多应用领域在寻找更小数据路径的IP核。

本文基于FPGA技术设计数据路径宽度为16位、密钥分组长度为128位的AES IP核，完成数据加密、解密和密钥扩展。根据AES算法的特点，在对硬件实现方式进行分析后，提出一种基于FPGA的16位数据路径AES IP核的设计方案。

2 AES简介

AES算法包括加密、解密和密钥扩展。加密算法通过对128位明文数据进行字节替换、行移位、列混合、密钥加等运算，实现加密。解密算法通过对128位密文数据进行逆字节替换、逆行移位、逆列混合、密钥加等运算，实现解密。每一轮变换都有一个不同的轮密钥。轮密钥由密钥扩展算法产生。AES标准指定3种不同的密钥长度：128位，192位和256位，分别需要10轮、12轮和14轮的变换。本文只研究密钥长度为128位的情况。AES直接解密算法不仅步骤本身与加密算法不同，而且步骤出现的次序也不同。为了便于硬件实现，采用了等效解密算法，可以使加密和解密算法在宏观上具有相似的结构，从而对加密与解密采用相同的轮外结构，而每一轮的内部也有许多部分可以实现硬件资源的共用。AES算法流程如图1所示。

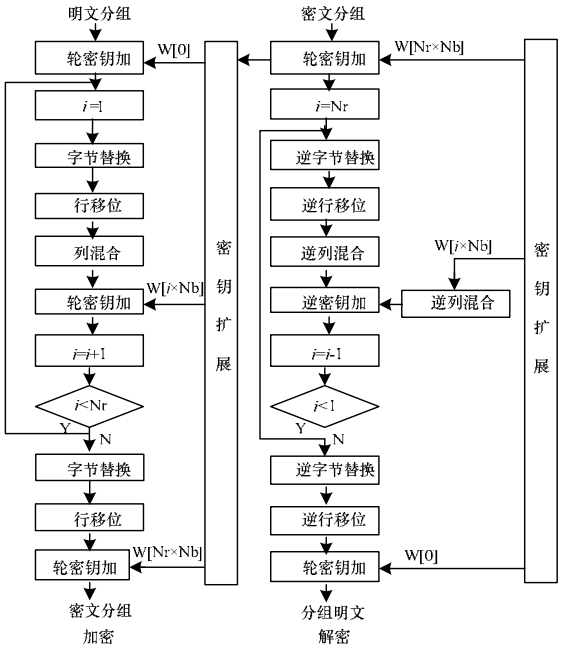

图1 AES算法流程

密钥扩展算法由输入的初始密钥生成每一轮变换所需的轮密钥，应在轮变换的密钥加运算开始前准备好该轮变换所需的轮密钥。密钥的扩展可采用并行扩展和非并行扩展2种方法[4]。并行密钥扩展指加密与密钥扩展同时进行，每次只生成一轮轮密钥，即轮密钥的生成与轮变换是并行的，在进

作者简介：张新贺(1980－)，男，讲师、硕士，主研方向：EDA技术，信息安全；张月华，讲师、硕士；刘鸿雁，教授
收稿日期：2009-05-25 **E-mail：**cdaszxh@sina.com

行轮变换的同时，生成下一轮的轮密钥。非并行密钥扩展是一次生成所有的轮密钥，在进行轮密钥加运算时，直接在存储轮密钥的寄存器中选取需要的轮密钥即可。一方面可使加解密在时序上完全一致；另一方面，在不是频繁更换密钥的情况下，只在加解密之前进行扩展，使用起来快速方便。这需要一些额外的寄存器存储轮密钥，但节省了解密过程的密钥扩展电路。综合考虑速度和面积等因素，本设计采用非并行扩展方式。在进行加密、解密前，先将密钥扩展完成，将扩展后的轮密钥存储在 176 Byte 的寄存器中，在每一轮运算时，直接在该寄存器中选择轮密钥。AES 的更多介绍可从公开发表的 FIPS-197[5]获得。

AES IP 核是指采用 VHDL 等硬件语言对 AES 算法进行描述，通过仿真、综合和验证，最终转化为可供 FPGA 使用的 IP 核。AES 算法 IP 核包括加密算法的硬件设计、解密算法的硬件设计和密钥扩展的硬件设计。该 IP 核采用有限状态机实现，包括主状态机和从状态机。主状态机由 7 个模块组成，在每个时钟周期对条件进行判断，在不同的状态之间跳转，完成不同的操作。在模块中引入从状态机。引入有限状态机后，模块的硬件工作模式和状态图描述一一对应，有利于 RTL 代码的编写。

3 AES IP 核设计

3.1 主状态机设计

该 AES IP 核包含以下模块：系统空闲模块(idle)，密钥加载模块(load_key)，密钥扩展模块(key_expander)，系统等待模块(wait)，明文/密文加载模块(load_data)，数据加密/解密模块(enc/dec)，密文/明文输出模块(data_out)。AES IP 核主状态机如图 2 所示。

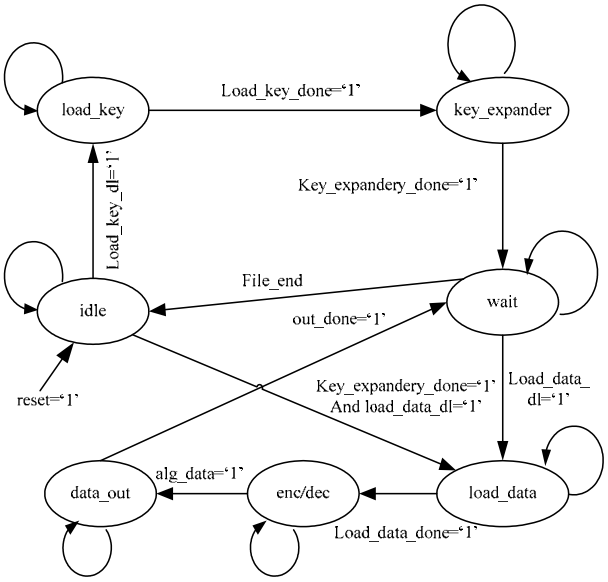

图 2 AES IP 核主状态机结构

系统复位后进入系统空闲模块，如果密钥加载信号有效，即 load_key_d1=1，则进入密钥加载模块。密钥加载模块分 8 次加载 128 位密钥，每次 16 位。如果密钥加载完成，则输出加载密钥完成信号 load_key_done=1，并跳转到密钥扩展模块；否则在密钥加载模块等待加载密钥。密钥扩展模块在 11 个时钟周期内进行轮密钥扩展，并将轮密钥存储在轮密钥寄存器中。密钥扩展完成后，跳转到系统等待模块，在该状态如果检测到加载明文/密文信号 load_data_d1=1，则跳转到明文/密文加载模块，否则，在该状态等待。明文/密文加载模块分 8 次加载 128 位明文/密文，完成后进入数据加密/解密模块，在 41 个时钟周期内进行加密/解密运算。加密/解密完成后，分 8 次将 128 位加密/解密后的数据输出，然后跳转到系统等待模块。该模块如检测到数据传输完毕信号 file_end=1，则进入系统空闲模块；否则，等待加载明文/密文信号的到来，继续加载数据。

3.2 密钥加载模块设计

密钥加载模块接收外部处理器送来的 128 位密钥，外部处理器为 16 位数据总线，该模块每次只能接收 16 位数据，128 位密钥需要 8 次才能接收完。密钥加载模块采用有限状态机实现。状态转换如图 3 所示。

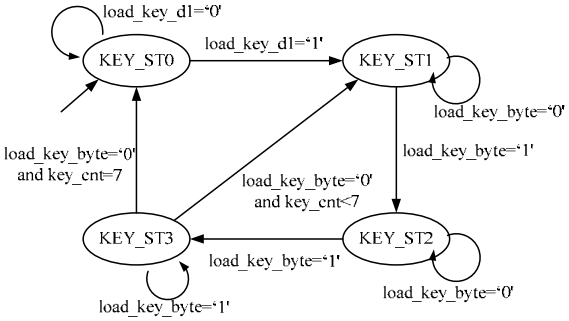

图 3 密钥加载模块状态转换

KEY_ST0, KEY_ST1, KEY_ST2, KEY_ST3 为该模块的 4 个状态。load_key_byte 为外部输入的加载 16 位密钥信号，高电平有效；load_key_d1 为外部输入信号 load_key 延时 1 个时钟周期的信号，为加载 128 位密钥信号，上升沿有效；key_cnt 为内部计数器，对输入的 16 位数据进行计数。系统复位后进入 KEY_ST0 状态，计数器赋初值 8。在此状态如果 load_key_d1=1，表示加载密钥信号有效，计数器 key_cnt 赋值 0，并进入 KEY_ST1 状态；否则，在 KEY_ST0 状态等待。在 KEY_ST1 状态，如果 load_key_byte=1(在此之前，外部处理器已经将要输出的数据放到数据总线上)，表示加载 1 个 16 位密钥的信号到来，则从数据总线上读取 16 位密钥并存储，然后跳转到 KEY_ST2 状态；否则，在 KEY_ST1 状态等待。在 KEY_ST2 状态，如果 load_key_byte=1，则将输出信号 read_key 置 1，表示密钥加载模块已经接收到 16 位密钥，并跳转到 KEY_ST3 状态；否则，在该状态等待。如果外部处理器检测到 read_key=1，则将 load_key_byte=0 复位。在 KEY_ST3 状态，如果 load_key_byte=0(表示外部处理器已经检测到 read_key=1)，将输出信号 read_key 清零；且如果计数值小于 7，则计数值加 1 并跳转到 KEY_ST1 状态，继续加载下一个 16 位密钥；如果 load_key_byte=0 且计数值等于 7，表示 128 位密钥已经接收完毕，则跳到 KEY_ST0 状态，等待下一次启动该模块 ;如果 load_key_byte=1(表示外部处理器未检测到 read_key=1)，则在该状态等待。

3.3 明文/密文加载模块设计

明文/密文加载模块接收外部处理器送来的数据。加密时，输入的是 128 位明文；解密时，输入的是 128 位密文。128 位明文/密文需要 8 次才能够接收完毕。该模块的设计方法与密钥加载模块类似。

3.4 密钥扩展模块设计

该模块由 2 个状态 KP_ST0 和 KP_ST1 组成。系统复位后首先进入 KP_ST0 状态，在该状态判断密钥加载完成信号 load_key_done 延时 1 个时钟周期的信号 load_key_done_d1

的状态，如果为 1，则进入 KP_ST1 状态，进行密钥扩展；否则，在该状态等待。在 KP_ST1 状态，进行轮密钥扩展，在该状态定义一个 11 进制计数器 kp_cnt，当计数值等于 *i* 时，进行第 *i* 轮密钥扩展，并将扩展后的轮密钥存储在相应的存储单元中，如果 kp_cnt<10，计数器值加 1，则在 KP_ST1 状态继续进行密钥扩展，否则，计数器值加 1 跳转至 KP_ST0 状态等待下一次启动该模块。

3.5 数据加密/解密模块设计

数据加密/解密模块对 128 位明文/密文进行加密/解密操作，总共需要 10 次循环加密/解密。按照算法流程，首先进行初始轮运算；然后是 9 次轮变换，最后是 1 次结尾轮变换。每个轮变换包含字节替换/逆字节替换、行移位/逆行移位、列混合/逆列混合、密钥加运算；结尾轮包含除列混合/逆列混合以外的 3 个步骤。状态转换如图 4 所示。

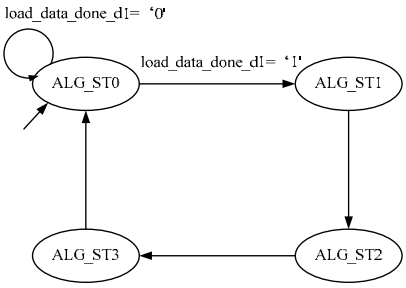

图 4 数据加密/解密模块状态转换

当系统复位后，进入 ALG_ST0 状态，在该状态检测明文/密文加载模块是否完成，即判断 load_data_done_d1 是否为 1，如果为 1，表示已经完成，则进入 ALG_ST1 状态；否则，停留在该状态。在 ALG_ST1 状态，进行初始轮变换；耗时 1 个时钟周期，完成后进入 ALG_ST2 状态。在 ALG_ST2 状态，进行中间轮变换，包括 4 个运算，在每一个时钟周期完成 1 个运算，共需要 4 个时钟周期；在该状态定义 1 个计数器 alg_cnt(计数范围 0~9)控制中间轮的循环次数，如果 alg_cnt=9，则进入 ALG_ST3 状态，否则，继续在该状态循环。在 ALG_ST3 状态，进行结尾轮变换，包括 3 个运算，需要 3 个时钟周期完成，完成后进入 ALG_ST0 状态。完成该模块共需 1+1+4×9+3=41 个时钟周期。

3.6 密文/明文输出模块设计

密文/明文输出模块将加密/解密后的数据密文/明文输出。每次输出 16 位，需要 8 次将 128 位密文/明文送给外部处理器。该模块的状态转换如图 5 所示。out_cnt 为内部计数器，计数范围为 0~8。系统复位后，首先进入 OUT_ST0 状态，计数值为 8，如果数据加密/解密运算完成信号 alg_done_d1=1，则计数器赋值 0，并进入 OUT_ST1 状态；否则，在该状态等待。在 OUT_ST1 状态，如果检测到外部处理器的读信号 read_byte=0，将要输出的 16 位数据送到数据总线上，并进入 OUT_ST2 状态；否则，在 OUT_ST1 状态等待 read_byte 信号变低。在 OUT_ST2 状态，该模块的输出信号 out_byte 置 1(表示该模块要输出 16 位密文/明文)，如果 read_byte=1(表示外部处理器正在读数据)，则进入 OUT_ST3 状态；否则，该状态等待 read_byte=1 信号的到来。如果外部处理器检测到 out_byte=1，则读取数据总线上的数据，并将 read_byte 清零。在 OUT_ST3 状态，如果 read_byte=0，输出信号 out_byte 清零，且如果计数值小于 7，表示 128 位密文/明文数据未输出完毕，则进入 OUT_ST1 状态，继续发送数据；如果 read_byte=0 且计数器为 7，表示 128 位数据发送完毕，则进入 OUT_ST0 状态，等待再次启动该模块；如果 read_byte=1，表示外部处理器没有读完数据线上的数据，则在该状态等待。

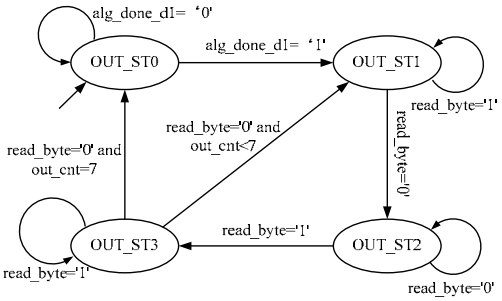

图 5 密文/明文状态转换

4 FPGA 实现结果

对密钥分组长度为 128 位的 AES 算法进行了系统仿真，见图 6。128 位密钥分 8 次加载，每次加载 16 位密钥。当密钥加载信号 load_key 变为高电平时，启动加载 16 位密钥，而后不管 load_key 的状态如何，依次加载剩余的 112 位密钥。128 位解密/解密后的结果分 8 次输出，每次输出 16 位数据。仿真数据采用高级加密标准资料提供的测试数据[5]。仿真结果与 C 语言程序结果一致，验证了该设计的正确性。

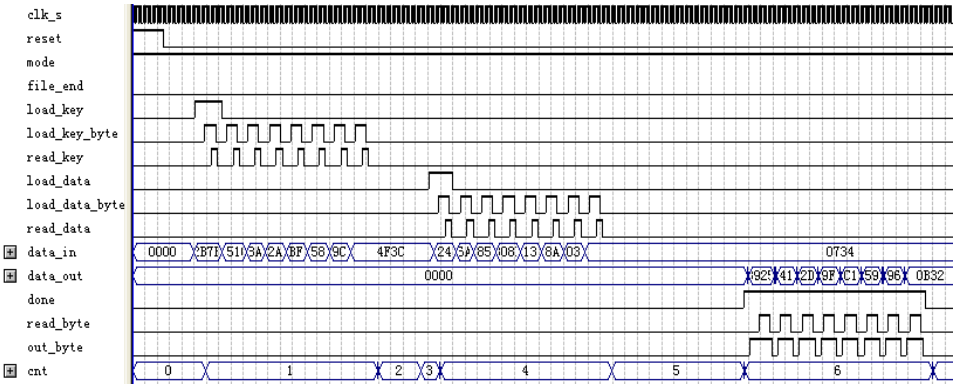

图 6 AES IP 核仿真结果

本设计很好地适配到 Cyclone II EP2C35F484 中，占用 20 070 个逻辑单元、2 519 个寄存器、48 个 引脚，资源占用率不到 60%。最高时钟频率达 100 MHz，完成 128 位 AES 算法(忽略密钥加载和密钥扩展)需要的时间等于明文/密文加载时间+加密/解密时间+密文/明文输出时间。加密/解密时间为 41×10 ns=0.41 ms，非常短。明文/密文加载时间和密文/明文输出时间取决于外部处理器的速度，即外部处理器的速度决定整个 AES IP 核的速度，算法本身的运行时间可以忽略不计。

5 结束语

本文的创新观点是提出一种 16 位数据路径宽度的 AES IP 核的设计方案，它容易与 16 位处理器进行接口。采用等

(下转第 167 页)