# 工作完成情况

毕业设计阶段报告

题目:RSA在FPGA上实现与应用

第一阶段:RSA理论研究及软件实现

一、RSA算法原理

1.1 RSA算法简介

RSA算法是一种非对称加密算法,它基于大数的分解问题的困难性.RSA算法的安全性依赖于大数分解的复杂度.

1.2 RSA算法流程

1) 选取两个大质数p和q

2) 计算n=pq

3) 计算欧拉函数φ(n)=(p-1)(q-1)

4) 选择一个与φ(n)互质的e,即gcd(e,φ(n))=1

5) 计算出d,使得ed≡1(mod φ(n))

6) 公钥为(e,n),私钥为(d,n)

7) 加密:C=M^e mod n

8) 解密:M=C^d mod n

二、RSA软件实现

2.1 开发环境

操作系统:Ubuntu 20.04 LTS

编程语言:C语言

2.2 关键函数实现

1) 大数运算函数

#include "bignum.h"

2) 素性检测函数(Miller-Rabin算法)

int millerRabinPrimeTest(BigInteger \*n, int k) { ... }

3) 生成大素数函数

BigInteger \*generatePrime(int bits) { ... }

4) 扩展欧几里得算法求模逆

BigInteger \*modInverse(BigInteger \*a, BigInteger \*n) { ... }

5) RSA密钥生成

void generateRSAKey(int bits, BigInteger \*\*n, BigInteger \*\*e, BigInteger \*\*d) { ... }

6) RSA加密解密

BigInteger \*rsaEncrypt(BigInteger \*m, BigInteger \*e, BigInteger \*n) { ... }

BigInteger \*rsaDecrypt(BigInteger \*c, BigInteger \*d, BigInteger \*n) { ... }

2.3 测试用例

生成4096位RSA密钥对,对一段明文进行加密解密,验证正确性.

# 下阶段工作内容

硬件描述语言Verilog及FPGA开发环境学习

一、学习Verilog硬件描述语言

1.1 Verilog基础语法

模块定义、端口定义

线网声明、连续赋值语句

过程赋值语句(always块)

条件语句(if/case)

循环语句(for/while/repeat)

任务/函数定义

1.2 设计层次与模块化

顶层设计与子模块实例化

模块参数化

时序控制(时钟、复位等)

1.3 常用硬件构造块

计数器

状态机

FIFO/LIFO

算术逻辑单元(ALU)

二、FPGA开发工具Vivado学习

2.1 Vivado开发环境

项目创建与管理

设计流程(合成、实现、生成比特流等)

调试与仿真

2.2 IP核的使用

Xilinx提供的IP核(如DSP、存储等)

自定义IP核打包

2.3 约束文件编写

时钟约束、IO约束

时序约束

三、基础硬件模块实现

3.1 移位寄存器(SHIFT\_REG)  
3.2 伪随机数生成器(LFSR)  
3.3 有限状态机(FSM)  
3.4 简单ALU(加减乘除、位操作)

四、下一步工作计划

基于Verilog实现基本的大数运算模块

结合RSA算法流程,设计RSA加解密核心模块

进行模块级仿真验证

综合并在FPGA开发板上实现,测试性能

通过下一阶段的学习,我将掌握Verilog语言编程和FPGA开发的基本技能,为下一步RSA算法的硬件实现做好准备。