1 Introduction

Working of RSA algorithm is heavily based upon computations on primary numbers. Two large prime numbers are generated and speed of generation and validation of these is crucial. Speeding up processes would let us break the RSA.

In this paper we will present software and hardware assisted implementations of GCD.

2 Method

2.1 Hardware-assisted implementation of the GCD algorithm

Hardware implementation done in VHDL. A program for computations on GCD was created in VHDL and so was testbench.

2.2 Implementation details of GCD

2.2.1 Processes

There are two processes in the program:

- State updater Reacts to clock and reset signal. On clock event, all variables with suffix `next` are copied to variables of type `current`.
- Logic

The implementation is based on state machines. State machines are specific case of finite automata. The program consists of four states.

- `idle`
- `start`
- `shift`
- `sub`

The process starts in `idle` state and continues to `start` state. The `busy` flag is lit and the process jumps between `shift` and `sub` until contents of `a_reg` and `b_reg` are equivalent. The `busy` flag is no longer lit and result is written out.

It is possible that the states `idle` and `start` are redundant, but this project aims to present a sample implementation of GCD.

3 Results