Fast Modular Exponentiation Hardware

Theory and Methods

Christopher Nguyen

University of Maryland, Baltimore County
Baltimore, MD 21250, USA

April 20, 2011

Problem and Motivation

Fast modular exponentiation is necessary.

- Most forms of public-key cryptography rely on modular exponentiation.
- Key lengths of public-key encryption systems are growing.
  - Yesterday: 1024-bit
  - Today: 2048-bit
  - Tomorrow: 4096-bit
- Modular exponentiation, the limiting operation, is slow using conventional number systems! Moreso, it is slower using residue number systems!!

Background

Residue Number System

Given a set of moduli $m_1, \ldots, m_n$, it is possible to represent a number x as remainders, or *residues*, of the moduli.

Example: $x = 10$ with $m_i = \{2, 3, 5\}$

$$x \bmod m_1 = x \bmod 2 \equiv 0$$
$$x \bmod m_2 = x \bmod 3 \equiv 1$$
$$x \bmod m_3 = x \bmod 5 \equiv 0$$

x represented as a residue number over the moduli is $(0, 1, 0)$.

Background

Chinese Remainder Theorem

Let $M = \prod_{i=0}^{n} m_i$.

If the set of moduli for an RNS are pairwise relatively prime, then numbers in the range $[0, M]$ have a unique representation in the RNS formed by the moduli.

- The proof is demonstrates existence, uniquness, and *construction*
- This theorem is at the heart of all residue number systems

More Problems

Advantages to RNS:

- Addition, subtraction, and multiplication are inherently parallel.
- There is no problem with overflow.
- Taking advantage of consecutive parallel operations receive a huge performance boost (e.g. DSP).

Disadvantages to RNS:

- Other operations become harder: reconstruction, magnitude comparison, and division.
- Computation often requires two RNS systems to be useful.
- For general-purpose computing, conventional number system algorithms still beat RNS algorithms.

Current Methods

Base Extension

- Converts one RNS base to another
- Shenoy and Kumaresan's Fast Base Extension

Montgomery Multiplication

- Let $a, b, c \in \mathbb{Z}_M$. Let $\hat{x} \equiv xR \bmod M$ for some special R. $\hat{c} \bmod M \equiv \hat{a}\hat{b}R^{-1} \bmod M$.
- Cost of "Montgomerification" is amortized over several modular multiplications.
- Main feature of many authors: Bajard and Kornerup, Bernal, Blum and Paar, and Fournaris.

New Theory and Methods

Phatak combines the following features:

- Joint integer and fractional representation
- Low precision approximation
- Precomputed look-up tables
- Small redundant modulus

These are used to increase efficiency of the hard operations:

- Reconstruction (base extension)
- Scaling (division by a constant)
- Magnitude comparison

Note: Magnitude comparison is not necessary for the modular exponentiation algorithm.

Road Ahead

Problem

Phatak has already proven his method and efficiency; it is fast from a theoretical stance. *But theoretically fast does not mean it is fast enough to become the standard implementation!*

Road Ahead

Solution

Implement the algorithms in hardware and demonstrate performance with hard data. But how do we do this?

1. Generate schematics using HDL code generation.
2. Prove correctness of the schematic using a statistically significant set of test cases.
3. Measure and report performance using standard metrics in the literature: execution time and chip area.

FPGAs are our target platform. They feature cost-efficiency and field-programmability. They are the current standard for hardware development.

Questions?