Laufzeitoptimierte VHDL Bibliothek zur Verifikation und Simulation kryptographischer Prozessoren

Mathias Schmalisch · Hagen Ploog · Dirk Timmermann

Universität Rostock

Fachbereich Elektrotechnik und Informationstechnik, Universität Rostock

Dipl.-Ing. Mathias Schmalisch

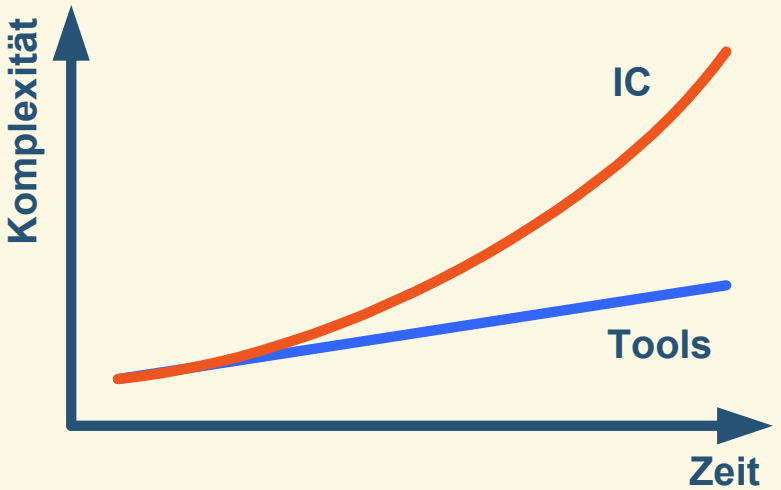
mathias.schmalisch@uni-rostock.de

Übersicht

- Motivation
- Arithmetik
- Implementierung
- Optimierung der Bibliothek
- Validierung
- Zusammenfassung

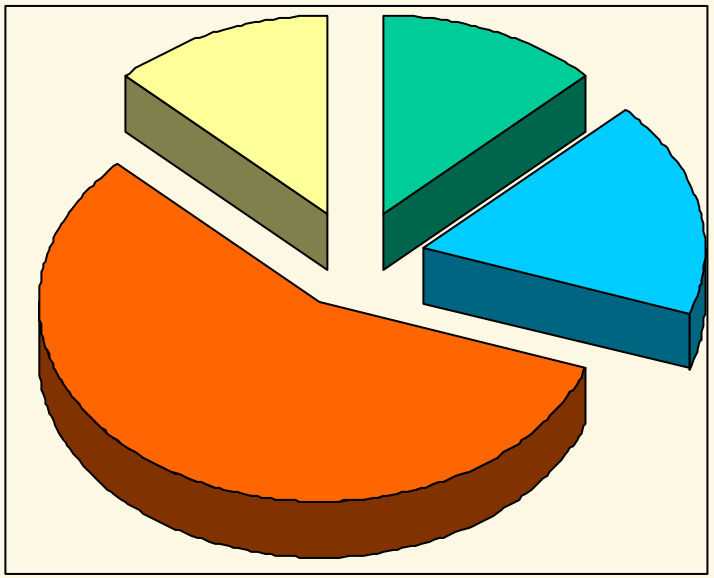
Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Chipentwurf



- Produktzyklen werden immer kürzer
- Exponentieller Anstieg der Komplexität in modernen IC‘s
- Steigerung der Produktivität von Entwicklungstools wächst nur linear

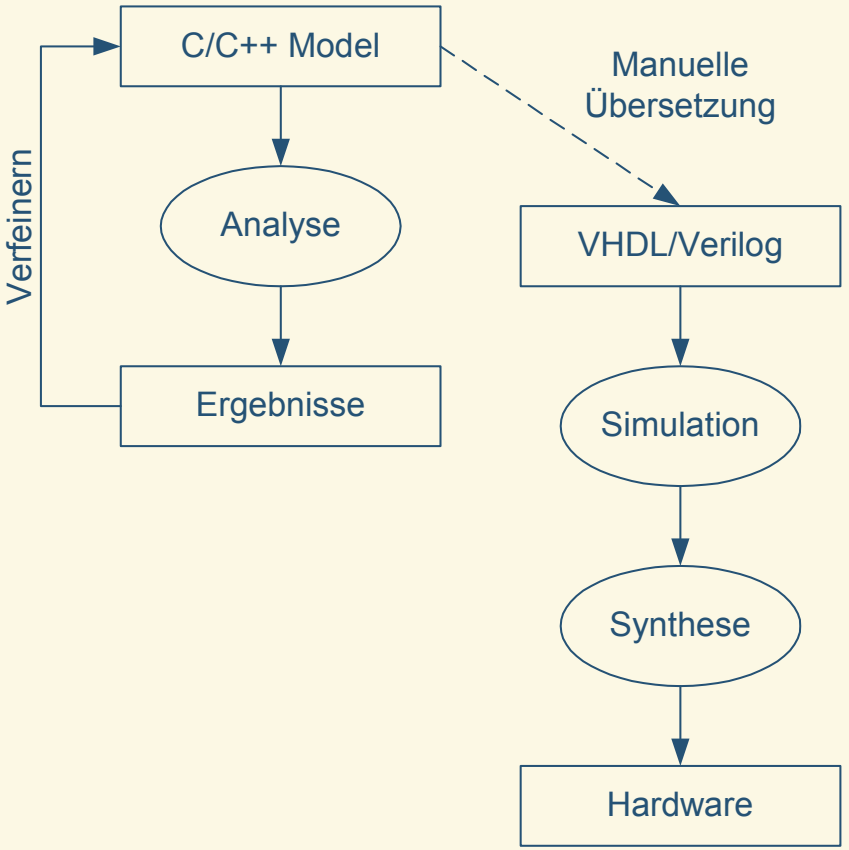
Aufwand Simulation & Verifikation



- Der Aufwand für die Simulation und Verifikation liegt bei aktuellen IC‘s zwischen 50 - 70%
- Mit steigender Komplexität der IC‘s wächst auch der Simulationsaufwand
- Längere Spezifiktionsphase verkürzt die Kodierungsphase, aber nicht die Simulations- & Verifikationsphase
- Dringender Bedarf für die Optimierung der Simulation und Verifikation

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Referenzmodell



- Referenzimplementierung üblicherweise in C/C++, daher Testvektorenaustausch z.B. über File-I/O
- Manuelle Übersetzung ist zusätzliche Fehlerquelle
- Eine homogene Lösung ist vorteilhafter (Referenzmodel in VHDL)
- Automatische Generierung von Testmustern und -vektoren

Motivation

- IEEE Standartbibliotheken fehlen wichtige Funktionen für Bitvektoren:
  - Division
  - Modulo
  - ...

- Zusätzliche VHDL-Bibliothek für Simulation kryptographischer Funktionen wird benötigt

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

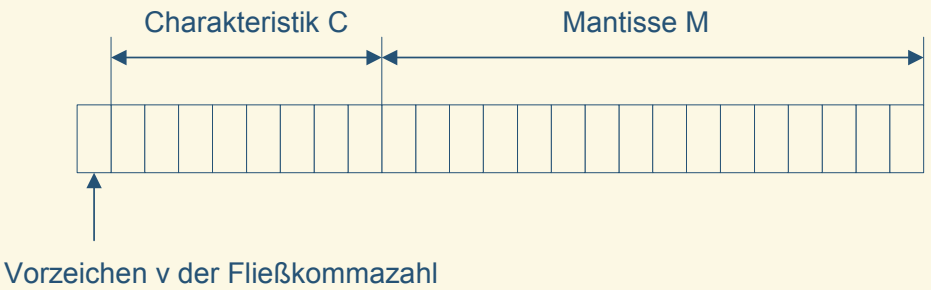
Übersicht

- Motivation
- **Arithmetik**
- Implementierung
- Optimierung der Bibliothek
- Validierung
- Zusammenfassung

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Langzahlarithmetik

- Fließkommazahlen für Kryptographie ungeeignet, da bei großen Zahlen gerundet wird



- Ausweg: **Langzahlarithmetik**
  - Bis auf die letzte Stelle genau

Multiple Precision Arithmetic (MPA)

- Operationen mit großen Zahlen werden auf kleinere Funktionen abgebildet
  - z.B. Addition von 1024 Bit Zahlen mit 16 Bit Addierer
- Umwandlung durch folgenden Formel:

$$X = \sum_{i=0}^{x-1} x_i 2^i = \sum_{i=0}^{s-1} d_i \left(2^w\right)^i = \sum_{i=0}^{s-1} d_i W^i$$

- Vorteil:
  - Überprüfung von Zwischenergebnissen leicht möglich

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Darstellung vorzeichenbehafteter Zahlen

- Zweierkomplementdarstellung

$$A_{ZK} = 0a_{n-2}a_{n-3}...a_1a_0 \qquad \text{für eine n-stellige positive Zahl}$$

$$-A_{ZK} = \left(\left(r-1\right)\overline{a}_{n-2}\overline{a}_{n-3}...\overline{a}_1\overline{a}_0\right)+1 \qquad \text{für eine n-stellige negative Zahl}$$

- Vorteile:
  - Positive und negative Zahlen können gleich behandelt werden
  - Vorzeichen kann am höchstwertigsten Bit erkannt werden

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

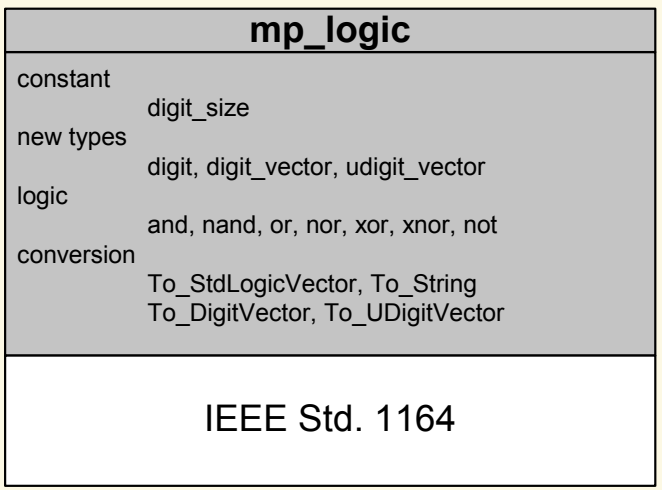
Übersicht

- Motivation
- Arithmetik
- **Implementierung**
- Optimierung der Bibliothek
- Validierung
- Zusammenfassung

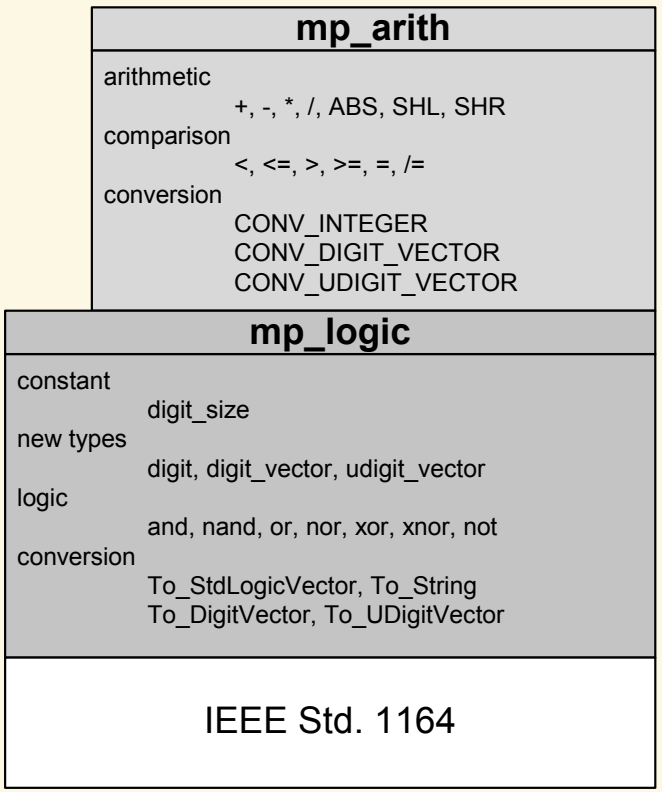
Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Aufbau der Bibliothek – mp_logic

- Aufgesetzte auf die IEEE 1164 Standardbibliothek
- Enthält neue Typen und Konstanten
- Logische Operationen
  - AND, OR, XOR, NOT, ...
- Konvertierungsfunktionen

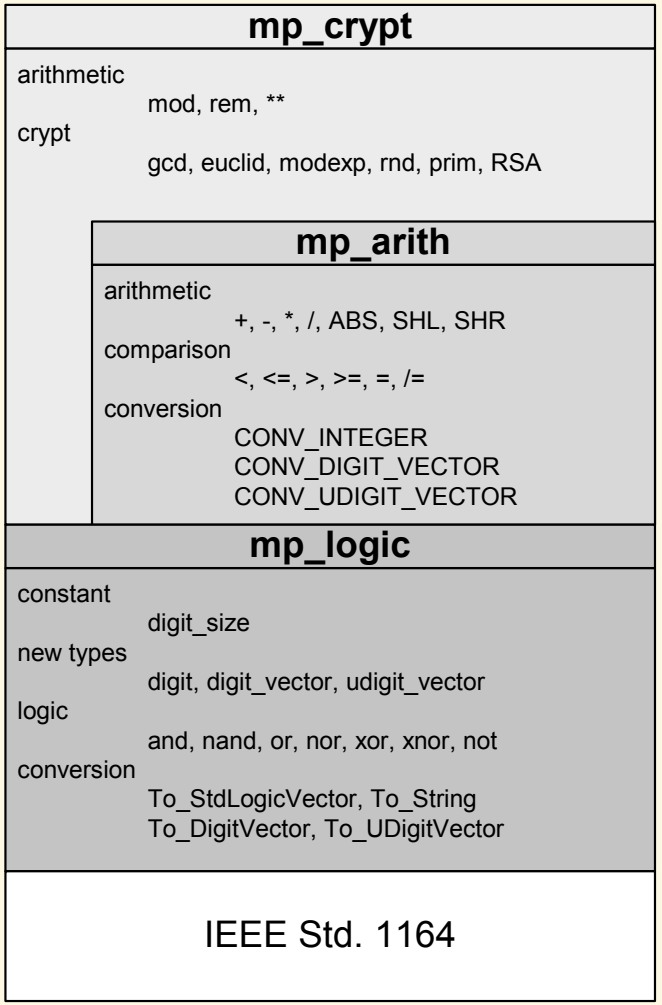


Aufbau der Bibliothek – mp_arith



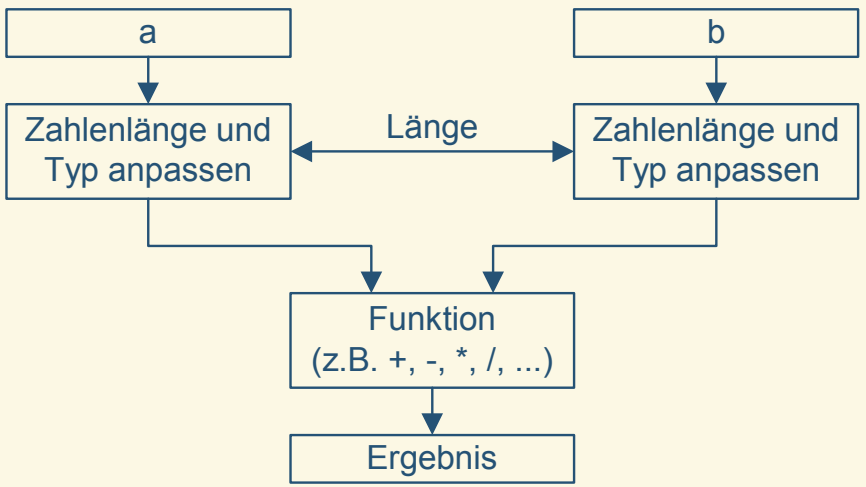
- Enthält arithmetische Funktionen
  - Addition
  - Subtraktion
  - Multiplikation
  - Division
  - Absolutwert
  - Links-/Rechtsscheiben
- Vergleichsfunktionen
- Funktionen zur Längenanpassung

Aufbau der Bibliothek – mp_crypt



- Beinhaltet erweiterte arithmetische Funktionen:
  - Modulo
  - Remainder
  - Exponentiation
- kryptographische Funktionen
  - größter gemeinsamer Teiler (GCD)
  - Euklidischer Algorithmus
  - Primzahltest
  - Modulare Exponentiation

Eine Funktion – Viele Typen



- Durch Konvertierungsfunktionen kann eine Funktion mit verschiedene Typen genutzt werden
- Erleichtert Wartung und Optimierung der Funktionen
- Erweiterung auf neue Typen durch zusätzliche Konvertierungsfunktionen möglich

Übersicht

- Motivation
- Arithmetik
- Implementierung
- **Optimierung der Bibliothek**
- Validierung
- Zusammenfassung

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

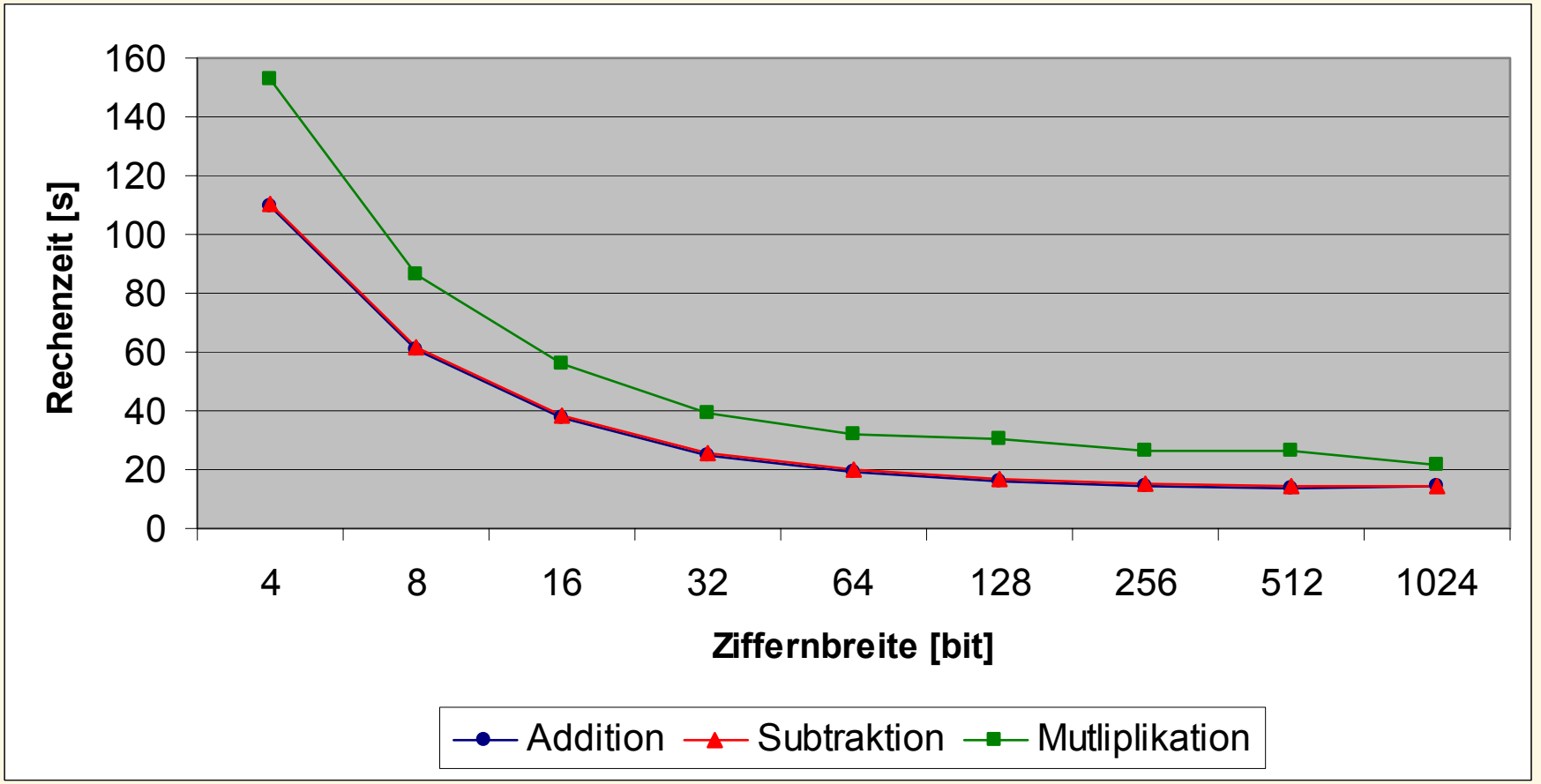
Performance Messung

- Es gibt oftmals mehrere Möglichkeiten um eine Funktion zu implementieren
- Auswahl der schnellsten Implementierung durch Performance Messung möglich
- Geschwindigkeitsmessung ist nur über das C-Interface des Simulators zu realisieren

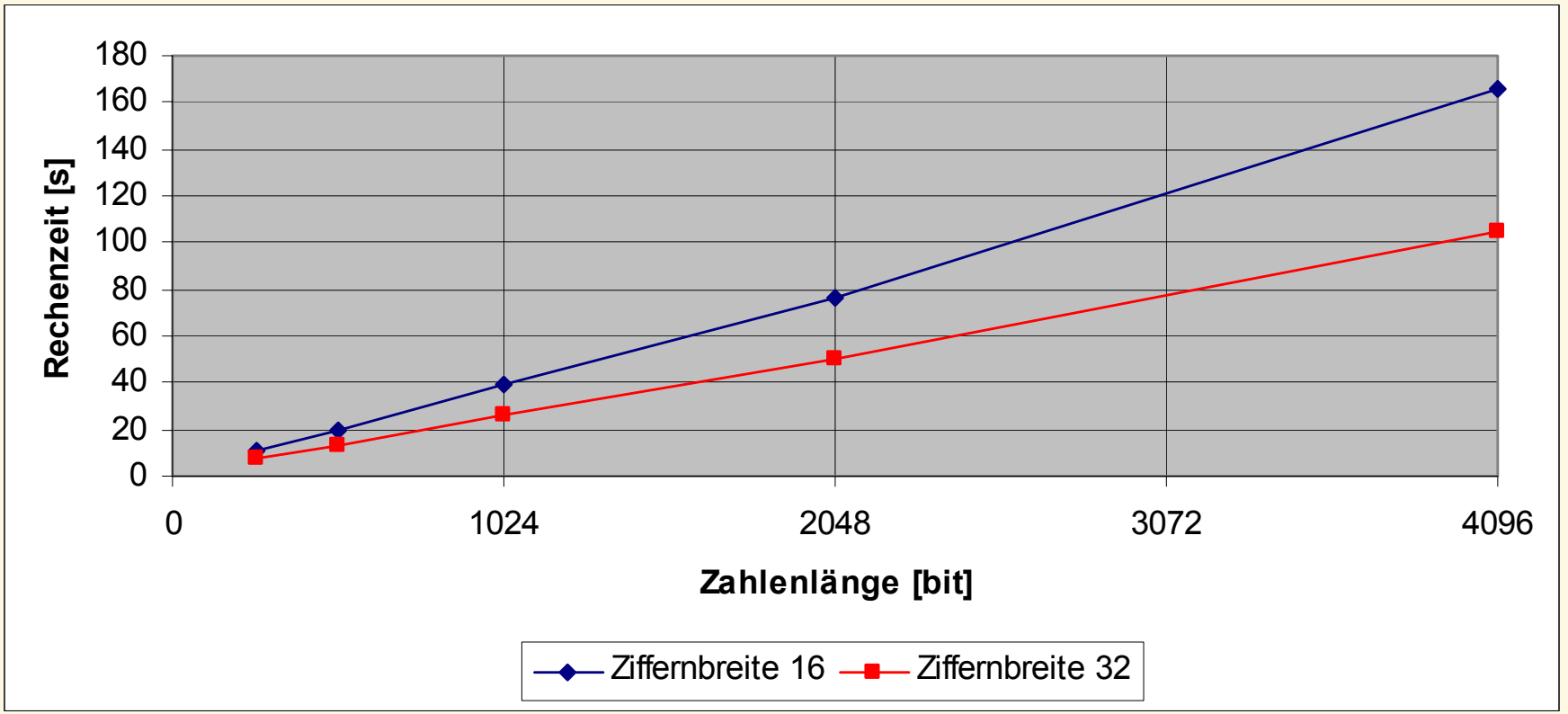
- z.B. Performanceunterschiede von 30% bei Addition

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Abhängigkeit von Ziffernbreite und Zahlenlänge



Abhängigkeit von Ziffernbreite und Zahlenlänge



Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Übersicht

- Motivation
- Arithmetik
- Implementierung
- Optimierung der Bibliothek
- **Validierung**
- Zusammenfassung

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Validierung

- Mit FreeLIP-C-Bibliothek von A. Lenstra
- Zufällige Testvektoren wurden über das C-Interface an die Bibliothek weitergereicht
- Test der einzelnen Funktionen über mehrer Tage, um Funktionalität zu sichern

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Übersicht

- Motivation
- Arithmetik
- Implementierung
- Optimierung der Bibliothek
- Validierung
- **Zusammenfassung**

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de

Zusammenfassung & Ausblick

- Homogene Konzeptions- und Entwicklungsumgebung von großem Vorteil
  - Sprache von Referenzmodell und Hardwarebeschreibung gleich
- Simulation komplexer IC‘s wird zunehmend zum Problem
- VHDL Bibliothek deckt fehlende Funktionalitäten der Standardbibliotheken ab
- Ausblick:
  - Erweiterung der Bibliothek für Elliptic Curve Cryptography

Dipl.-Ing. Mathias Schmalisch
mathias.schmalisch@uni-rostock.de