Ein FPGA basierter *Elliptic-Curve* Kryptoprozessor mit variabler Schlüssellänge für hohen Datendurchsatz

Markus Ernst und Sorin A. Huss, Integrierte Schaltungen und Systeme, Fachbereich Informatik, Technische Universität Darmstadt, D-64283 Darmstadt

Kurzfassung

Eine Folge des zunehmenden Datenverkehrs auf öffentlichen, ungesicherten Netzen ist in der Sensibilisierung der Benutzer hinsichtlich der Abhör- und Manipulationssicherheit persönlicher Daten zu sehen. Insbesondere Anwendungen im Bereich des E-Commerce werden sich nur dann am Markt durchsetzen können, wenn vertrauliche Informationen *sicher* über öffentliche Medien ausgetauscht werden können. Die Basistechnologie hierzu ist die *Public-Key* Kryptographie. Aufgrund besonderer Vorteile bezüglich der zu verwendenden Schlüssellängen gewinnen Verfahren auf der Basis Elliptischer Kurven zunehmend an praktischer Bedeutung. Die zum Erzielen eines hohen Sicherheitsstandards durchzuführenden Berechnungen sind bei diesen Verfahren sehr aufwendig und entsprechend rechenzeitintensiv.

In diesem Beitrag wird ein neuer Ansatz zur Realisierung hochperformanter Kryptoprozessoren vorgestellt. Aufgrund einer Generator basierten Vorgehensweise ist der Prozessor parametrisierbar, um unterschiedliche Systemanforderungen bei E-Commerce Anwendungen optimal erfüllen zu können. Unter Verwendung von Modellbeschreibungen in VHDL, Syntheseverfahren und rekonfigurierbarer FPGA Architekturen wird gezeigt, daß hochperformante Kryptoprozessoren für unterschiedliche Sicherheitsanforderungen einfach und schnell implementiert werden können, die darüber hinaus auch in bereits bestehende Kryptosysteme als Koprozessor integrierbar sind.

1. Einleitung

Viele E-Commerce Anwendungen sind u.a. dadurch gekennzeichnet, daß vertrauliche Informationen über öffentliche Kommunikationsnetze (z.B. das Internet) ausgetauscht werden. Diese Informationen müssen vor dem unberechtigten Zugriff Dritter geschützt werden. Die Basistechnologie, um diesen Schutz zu gewährleisten, ist die *Public-Key* Kryptographie. Neben dem weit verbreiteten RSA-Verfahren gewinnen *Public-Key* Verfahren auf der Basis Elliptischer Kurven (EC) zunehmend an Bedeutung. Die Sicherheit dieser Verfahren hängt im wesentlichen von der verwendeten Schlüssellänge ab (z.B. 1024 RSA-Bits entsprechen ca. 160 EC-Bits). Im Vergleich zu RSA bedeutet dies, daß beim Einsatz von EC-Verfahren das gleiche Sicherheitsniveau bereits mit wesentlich kürzeren Schlüsseln erreicht werden kann. Die geringere Schlüssellänge ermöglicht wiederum effizientere Implementierungen und somit einen höheren Durchsatz der zu verschlüsselnden Daten.

Die Operation $K \bullet P$, d.h. die Multiplikation eines Punktes $P(x,y)$ auf einer Elliptischen Kurve mit einer natürlichen Zahl K, stellt die Basisoperation im Bereich der EC-Kryptographie dar. Sie ist sehr aufwendig und entsprechend rechenzeitintensiv. Die zur Berechnung von $K \bullet P$ benötigte Zeit bestimmt im wesentlichen das Laufzeitverhalten von EC-Implementierungen. Der am Institut für Integrierte Schaltungen und Systeme (ISS) der TU Darmstadt entwickelte parametrisierbare *Elliptic-Curve* Kryptoprozessor führt genau diese Operation in Hardware durch. Die Performanz von EC-Kryptosystemen (z.B. *Online-Banking* Server) kann durch den Einsatz dieses Prozessors um mehrere Größenordnungen gesteigert werden.

Im Folgenden wird der Algorithmus zur Berechnung von $K \bullet P$ vorgestellt. In Abschnitt 3 und 4 wird auf die

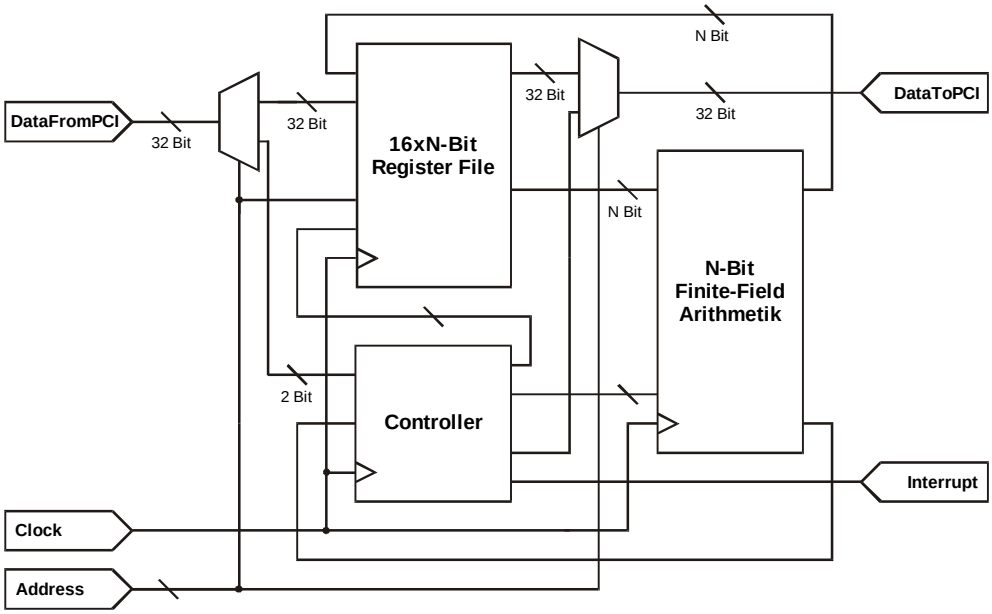

Abbildung 3: Prozessorarchitektur

Instanzen des Kryptoprozessors wurde am ISS ein dediziertes Generatorprogramm entwickelt. Das damit generierte VHDL-Modell des Prozessors bildet die Grundlage für die anschließende Hardware-Synthese und die weiteren Entwurfsschritte zur Erzeugung der FPGA-Konfigurationsdatei.

Es wurde ein Generator basierter Ansatz zur Erzeugung der VHDL-Beschreibung gewählt, da die ausschließliche Modellierung der Funktionalität mit synthesefähigen VHDL-Sprachkonstrukten [7] nicht möglich ist. Dies liegt hauptsächlich an der speziellen Struktur des verwendeten Massay-Omura Multiplizierers, der im Wesentlichen aus einem sehr großen XOR-Baum besteht [1]. Zwei *N*-Bit Eingangsvektoren werden auf diese Weise auf ein einzelnes Bit abgebildet. Da sich die Struktur dieses XOR-Baumes mit dem zugrundeliegenden Endlichen Körper bzw. der Schlüssellänge *N* ändert, ist es nicht möglich ein entsprechendes generisches, synthesfähiges VHDL-Modell allein auf RT-Level anzugeben.

Mit diesem VHDL-Generator sind wir in der Lage VHDL-Beschreibungen des Kryptoprozessors für unterschiedliche Schlüssellängen zu erzeugen. Hierdurch ist es leicht möglich zukünftig steigende Sicherheitsanforderungen, d.h. größere Schlüssellängen zu unterstützen. Speziell für FPGA-basierte Zielplattformen ergeben sich weitere Vorteile durch den Generator gestützten Ansatz: Da die Anzahl der instanziierten Massay-Omura Multiplizierer parametrisierbar und somit variabel ist, lassen sich die jeweils verfügbaren FPGA-Ressourcen optimal ausnutzen, was wiederum die Voraussetzung zum Erreichen der maximalen Performanz ist. Da die generierten VHDL-Modelle nicht auf eine spezielle FPGA-Familie zugeschnitten sind, ist es somit möglich den technologischen Fortschritt bei FPGAs (größere und schnellere Bausteine) direkt in bessere Performanz umzusetzen.

Auch in Bezug auf Qualitätssicherung und Validierung ergeben sich Vorteile durch die Generator basierte Vorgehensweise. Implementierungen für kleine Schlüssellängen (z.B. 18-Bit) können für erschöpfende, funktionale Tests verwendet werden. Dies ist notwendig um sicherzustellen, daß der VHDL-Genrator selbst richtig funktioniert. Anhand von praxisrelevanten Kryptoprozessoren mit Schlüssellängen $N = 160$ sind wirklich erschöpfende Tests nicht durchführbar.

5. Prozessorarchitektur

Auf dem am ISS entwickelten und hier vorgestellten *Elliptic-Curve* Kryptoprozessor ist der komplette *Double-and-Add* Algorithmus zur Berechnung von $K \bullet P$ in Hardware implementiert. Die Architektur des Prozessors ist in Abbildung 3 dargestellt und besteht im wesentlichen aus drei funktionalen Blöcken.

***Finite-Field* Arithmetik:**
Hier sind die oben beschriebenen *FF*-Operationen implementiert. Für *FF-Add* und *FF-Square* steht jeweils eine Komponente zur Verfügung. Zur Berechnung von *FF-Mult* können, wie bereits erwähnt, mehrere Massay-Omura Multiplizierer instanziiert werden. Hierdurch lassen sich die jeweils verfügbaren FPGA-Ressourcen optimal ausnutzen und somit die maximale Performanz erzielen.

Register File:
Der Prozessor verfügt über 16 *N*-Bit Register zur Speicherung von *FF*-Körperelementen bzw. von positiven ganzen Zahlen der entsprechenden Bitbreite.

Controller:
Im Controller ist der *Double-and-Add* Algorithmus zur Berechnung von *K•P* implementiert. Er ist als Endlicher Automat realisiert und stellt das Steuerwerk des Kryptoprozessors dar.

Die in Abbildung 3 angegebenen externen Signale dienen zur Kommunikation mit dem Kryptoprozessor und werden von der PCI-Schnittstelle bedient.

6. Implementierung

Die zur Entwicklung und Implementierung der Kryptoprozessoren eingesetzte *microEnable*-Karte ist mit einem XC4085XLA FPGA bestückt. Dieser Baustein verfügt über 3.136 Logikblöcke (CLBs) - das entspricht einer Komplexität von ca. 180.000 äquivalenten Gattern.

Der Entwurfsablauf gliedert sich im wesentlichen in folgende drei Schritte:

1. VHDL-Generator:
Ein VHDL-Modell des Kryptoprozessors mit fester Schlüssellänge und festem Parallelisierungsgrad wird mit dem bereits erwähnten Generator automatisch erzeugt.

2. Hardware Synthese:
Aus dem generierten VHDL-Modell wird mittels Synthese eine zur Granularität des FPGA passende, Netzliste erzeugt. Für die nachfolgend aufgeführten Implementierungs-Beispiele wurde der FPGA Compiler II V3.5 der Fa. Synopsys, Inc. eingesetzt.

3. FPGA Implementierung:
Basierend auf der synthetisierten Netzliste wird mit herstellerspezifischen Entwurfswerkzeugen *für Mapping, Placement* und *Routing* die FPGA-Konfigurationsdatei erzeugt. Für die nachfolgenden Beispiele wurde Foundation Series Software V2.1i von Xilinx, Inc. verwendet.

Für ein XC4085XLA-FPGA resultiert dieser Entwurfsablauf z.B. in einem Kryptoprozessor der Schlüssellänge *N=270* mit Parallelisierungsgrad 3. Dieser Entwurf hat eine CLB[5]-Auslastung von 82% und eine maximale Taktfrequenz von 34 MHz. Die resultierende Berechnungszeit für eine *K•P* Operation beträgt 6.84 ms (siehe Tabelle 1).

[5] Configurable Logic Block - Bezeichnung für die logischen Grundelemente von Xilinx FPGAs.

[6] Werte geschätzt

Zielplattform	Schlüssel-länge	*K•P* Performanz
C/C++ **SW** [4] (Intel PPro, 200 MHz)	191	21.10 ms
FPGA **HW** [8] (XCV400E, 76.7 MHz)	167	0.21 ms
ISS FPGA **HW** (XC4085XLA, 34 MHz)	270	6.84 ms
ISS FPGA **HW** (XC4085XLA, 36 MHz)	191	2.31 ms
ISS FPGA **HW** (XC4085XLA, 36 MHz)	173	1.76 ms
ISS FPGA **HW** [6] (XCV1000, 40 MHz)	191	0.10 ms

Tabelle 1: Performanzvergleich

Zwei weitere Beispiele für *N=191* und *N=173* verdeutlichen, wie aufgrund des vorgeschlagenen Generator basierten Ansatzes *Sicherheit* und *Performanz* gegeneinander abgewogen werden können. Eine 191-Bit Version des Kryptoprozessors mit Parallelisierungsgrad 5 hat eine CLB-Auslastung von 69% und eine maximale Taktfrequenz von 36 MHz. Das entspricht einer *K•P* Performanz von 2.31ms. Ein 173-Bit Kryptoprozessor mit Parallelisierungsgrad 6 benötigt hingegen 66% der CLB-Resourcen und funktioniert ebenfalls bis 36 MHz, was einer *K•P* Performanz von 1.76ms entspricht. Es fällt auf, daß die CLB-Auslastung mit steigendem Parallelisierungsgrad abnimmt. Das liegt an den relativ geringen Verdrahtungsresourcen der verwendeten Xilinx XC4000 FPGA-Familie. Bei neueren FPGA-Bausteinen ist das Problem nicht derart stark ausgeprägt. Die Komplexität eines FPGA des Typs XC4085XLA ist durchaus ausreichend, um einen Kryptoprozessor der Schlüssellänge 191 Bit mit Parallelisierungsgrad 9 zu implementieren, was einer CLB-Auslastung von ca. 95% entspricht. Ein solcher Entwurf läßt sich allerdings nur sehr schlecht verdrahten, so daß die resultierende maximale Taktfrequenz nur bei ca. 12 MHz liegt. Die resultierende *K•P* Performanz ist dementsprechend schlechter als bei der Variante mit Parallelisierungsgrad 5. Dieser Evaluierungsprozess, der zum Erreichen der maximalen Performanz erforderlich ist, wird durch den vorgestellten VHDL-Generator in idealer Weise unterstützt.

In Tabelle 1 ist die Performanz der vorgestellten Kryptoprozessoren im Vergleich zu führenden Hard- und Softwareimplementierungen dargestellt. Die angegebenen *K•P* Performanzwerte wurden auf einem

Standard PC (Intel Pentium III, 550 MHz) unter MS Windows NT 4.0 in Echtzeit gemessen.

Gegenüber der Softwareimplementierung aus [4], bei der die gleiche Schlüssellänge verwendet wird, ergibt sich ein Beschleunigungsfaktor von 9. Bei [8] handelt es sich um die derzeit schnellste, dokumentierte Hardware-Implementierung für eine allerdings feste Schlüssellänge *N=167*. Als Zielplattform wird ein Xilinx-FPGA der neuesten Generation (VirtexE) eingesetzt. Diese Bausteine ermöglichen im Vergleich zu FPGAs der XC4000 Serie wesentlich komplexere und schnellere Implementierungen. Ab März 2001 wird die neue *microEnableII* PCI-Karte allgemein verfügbar sein, welche ebenfalls mit einem Virtex FPGA bestückt sein wird. Der Entwurf eines 191-Bit Kryptoprozessors für diese neue Zielplattform wurde am ISS bereits vorab durchgeführt. Erste, sicherlich noch zu verbessernde Performanzwerte für eine 191-Bit *K•P* Operation liegen bei 0.1 ms, was deutlich unter dem in [8] publizierten Wert liegt.

7. Zusammenfassung

Der an der TU Darmstadt entwickelte *Elliptic-Curve* Kryptoprozessor führt die sehr rechenzeitintensive Basisoperation im Bereich der *Elliptic-Curve* Kryptographie *(K•P)* in Hardware durch. Der Prozessor basiert auf einer Standard PCI-Karte mit einem rekonfigurierbaren Logikbaustein (FPGA), in dem die Funktionalität des Prozessors implementiert wird. Aufgrund eines Generator basierten Ansatzes ist der Kryptoprozessor parametrisierbar, um unterschiedliche Systemanforderungen optimal erfüllen zu können. Über eine PCI-Schnittstelle kann dieser Kryptoprozessor auch in bereits bestehende Kryptosysteme einfach integriert werden, wodurch sich die Performanz derartiger Systeme ohne großen Aufwand um mehrere Größenordnungen steigern läßt.

Literatur

[1] J. Massey and J. Omura:
Computational Method and Apparatus for Finite Field Arithmetic, U.S. Patent 4-587-627, 1986.

[2] O. Hauck, A. Katoch and S. A. Huss:
VLSI System Design Using Asynchronous Wave Pipelines: A 0,35µm CMOS 1.5 GHz Elliptic Curve Public Key Cryptosystem Chip. Proc. IEEE ASYNC 2000, Eilat, April 2000.

[3] M. Rosing:
Implementing Elliptic Curve Cryptogarphy, Manning Publications Co., Greenwich, 1999. ISBN 1-884777-69-4

[4] E. De Win, S. Mister, B. Preneel and M. Wiener:
On the Performance of Signature Schemes based on Elliptic Curves. Proc. Algorithmic Number Theory Symposium III, LNCS 1423, J. P. Buhler, Ed., Springer-Verlag, pp. 252-266, 1998.

[5] Silicon Software: *microEnbale* Benutzerhandbuch, 1999.

[6] Xilinx: Programmable Logic Data Book, 1999.

[7] IEEE Standard 1076.6: Standard for VHDL Register Transfer Level Synthesis, IEEE Standards Department, New York, 1999.

[8] G. Orlando and C. Paar:
A High-Performance Reconfigurable Elliptic Curve Processor for $GF(2^m)$. Workshop on Cryptographic Hardware and Embedded Systems, Proc. CHES 2000, Worcester MA, USA, August 2000.