VLSI Design

KU Sommersemester 2008
SHA-1 Low-Power Module
Motivation

Least-Cost signature generating device

- Wireless RFID label
  - Can prove genuineness of products
    - By generating digital signature
  - Elliptic-curve digital signature

Reader

© Finkenzeller

© Texas Instruments
Digital Signature using ECC

ECDSA

\[ e = \text{SHA-1}(\text{Message}) \]
\[ k = \text{random}(1, n-1) \]
\[ R = k \cdot (P_x, P_y) = (R_x, R_y) \]
\[ r = R_x \mod n \]
\[ s = k^{-1} \cdot (e + d \cdot r) \]

– Central operation: Scalar ECC multiplication \( k \cdot P \)
  • Calculated by finite-field arithmetic
    – Modular arithmetic in prime fields or binary fields

– Also important: SHA-1
RFID Requirements

(Very) small area
- To keep production cost low

Low power consumption (HF 13.56 MHz)
- For long reading range (~ 1m)
- < 15μA * 2,5 V in total
- -> small clock frequencies
  - 106 kHz – 847,5 kHz

Throughput
- Not so important
- ECC can compute most beforehand
ECC Hardware Architecture

Tailored to RFID requirements
- Small datapath width
- Takes more clock cycles
- Has lower power consumption
  - Less energy efficient (!)
- Memories crucial

RFID Tag

- Analog Front End
- Digital part
  - Digital front end
  - RFID controller
  - Crypto unit

RAM

control
datapath

16
From ECC to Digital Signature

Hash is missing!

Idea

- Reuse ECC-RAM
- Just a new datapath

SHA-1 datapath
SHA-1

Secure Hash Algorithm

- Computes digital fingerprint
- 160-bit hash value
- Arbitrary long input (< $2^{64}$)

- Input blocks
  - 512 bits

- Message expansion

© Wikipedia.org / Matt Crypto
Possible Improvements

Lower datapath width
- 16 bits instead of 32

Datapath
- Local storage
- Or using ECC memory?

Constants (5 * 32-Bit)
- In ECC memory?
- In datapath?

Clock gating
- Do not clock unused flipflops
- No overhead for multiplexors
Goals

SHA-1 hardware
- Explore
  - Different architectures
    - Optimization of proposed solution
- Estimate
  - Cost of approaches (A*t*P)
- Implement
  - Most promising variant
  - Standard-cell test chip

Constraints
- Padding done externally
- Maximum power $P_{\text{max}}$: 8 $\mu$A
  - @ 2.5 Vdd
- Clock frequency = 106 kHz
- Max clock frequency > 66 MHz

Selected architecture
- Hardware-oriented high-level model
- Executable specification

Optimization
- Lowest $A*t*P$
  - $P < P_{\text{max}}$
  - $t$ … #clock cycles

Also of interest
- Simple architecture
  - “Keep the things easy”
- Low storage requirements
- Control unit
  - Keep overhead at minimum
- IO interface: 8-bit
  - Amba APB?
Requirements

Hardware architecture
- Optimization goal: throughput

High-level model
1. True high level model
2. Model reflecting proposed architecture

Cost estimation
- Target technology: 0.35 µm CMOS
- Austriamicrosystems c35b4
- Hardware implementation
- Synchronous design
  - Synchronous clock
  - Asynchronous power-on reset
- Synthesizable HDL
  - Verilog or VHDL
- Simulation using TCL
  - TCL scripts
  - Testdata files

Test chip
- Standard-cell layout for c35b4
- Near-Spice power simulation

Design document
- Introduction including motivation
- Short description of algorithms
- Hardware estimates
  - Area, time, power consumption
- Hardware implementation
  - General considerations
  - Architecture and modules
- Hardware results
  - Area, time, power consumption
  - Picture of layout
- Comparison of estimates and results
- Conclusions
- Literature

KU Journal
- “IngenieurInnen-tagebuch”
Relevant Literature

   
   2. National Institute of Standards NIST, 
      FIPS PUB 180-3, Secure Hash Standard. 
      
   3. Austriamicrosystems, 0.35 µm CMOS Libraries (C35), 
      [-> asic.austriamicrosystems.com] oder 
      StudentNet: [thalys.student.iaik.tugraz.at]
Schedule

Mon 25. Feb 2008 12:00, HS i13, IT-Sec Summer Term Kick-Off
Wed 05. Mar 2008 10:15, SR IAIK, KU01 Assignment Presentation
Wed 16. Apr 2008 10:15, SR IAIK, KU02 Designflow Presentation
Wed 30. Apr 2008 10:15, SR IAIK, KU03 Intermediate Presentation
< Fri 15. June 2008, office J. Wolkerstorfer, Submission Deadline
    (including colloquium)

Deadlines

- Registration       <= 29. Feb 2008    (TUGonline)
- Intermediate       <= 30. April 2008  (email submission)
- Final              <= 15. June 2008    (email + “Abgabegespräch”)
Deliverables Intermediate

High-level model
  – Model reflecting architecture (for variant with best estimate)

Design document v1.0
  – ~ 5 pages; Not more than 10 pages
  – Introduction including motivation
  – Short description of algorithm
  – Hardware estimates
    • Area, time, power consumption
    • For at least two different architectures
  – Hardware implementation
    • Proposed architecture (for algorithm with best estimate)
    • Architectural options (speed vs. area vs. power)
  – Literature

Submission per mail: Johannes.Wolkerstorfer@iaik.tugraz.at
Deliverables Final

Project files: Cleaned and zipped project directory

Design document v2.0
- Fully blown design document (~ 15 pages)
- In addition to document v1.0:
  - Hardware results: Area, time (incl. critical path analysis), power
    1. Synthesis results
    2. Results from layout editor
    3. Results from power simulation
    » Picture of layout
  - Comparison of estimates and results
  - Conclusions

Submission
- Mail to Johannes.Wolkerstorfer@iaik.tugraz.at
  - Propose a date for colloquium (Abgabegespräch)
High-level model

```c
void MixColumn(word8 a[4][4]) {
    word8 b[4][4];
    int i, j;

    for(j = 0; j < 4; j++)
        for(i = 0; i < 4; i++)
            b[i][j] = mul(2,a[i][j])
                ^ mul(3,a[(i + 1) % 4][j])
                ^ a[(i + 2) % 4][j]
                ^ a[(i + 3) % 4][j];

    for(i = 0; i < 4; i++)
        for(j = 0; j < 4; j++)
            a[i][j] = b[i][j];
}
```
High-level Model: Refined Model

Example

- Elliptic-Curve Cryptography
  (Java)

```java
void ecGf2AffDouble() {   // Point double
    alu.loada(mem[mem_x]);
    alu.inv2();   // x1^-1
    alu.mul2(mem[mem_y]);   // y1/x1
    alu.add2(mem[mem_x]);   // y1/x1 + x1
    mem[mem_t1] = alu.get();   // L (lambda)
    alu.sq2(mem[mem_t1]);   // L^2
    alu.add2(mem[mem_t1]);   // L^2 + L
    alu.add2(mem[mem_a]);   // L^2 + L + a
    mem[mem_t2] = alu.get();   // x3
    alu.add2(mem[mem_x]);   // x3 + x1
    alu.mul2(mem[mem_t1]);   // (x3 + x1)*L
    alu.add2(mem[mem_t2]);   // (x3 + x1)*L + x3
    alu.add2(mem[mem_y]);   // (x3 + x1)*L + x3 + y1
    mem[mem_y] = alu.get();   // y3
    mem[mem_x] = mem[mem_t2];   // x3
    memPrint("DOUBLE");
}
```
Hardware Estimates

Datapath will determine chip size
  – Control usually < 10%

Largest standard cells (here c35b4 process):
  • Flip-flop (e.g. 350 µm² @ 0.35 µm)
  • Full-adder (e.g. 275 µm² @ 0.35 µm)
  • Multiplexer (e.g. 150 µm² @ 0.35 µm)
  – Determine datapath size

Estimated chip size = 
  = (#flipflop * 350 + #fulladder * 275 + #mux * 150) + 50% overhead

Estimated maximum clock frequency
  – Depends on critical path
    • Longest combinational path
    • Sum of cell delays
Implementation of proposed architecture

- RTL description
  - Using Verilog / VHDL
  - Bit-accurate, cycle-accurate
  - Verified by simulation
- Synthesis
- Place-and-Route
- Backend verification
- Layout Extraction
- Post-layout simulation
  - Power simulation

Hardware Implementation

High L Model
HDL Model
Synthesis
Placement
Routing

Logo © Cadence
Groups and Accounts

Groups
- Groups of 3 students
  - Register on printed form
- Group members get listed
  - On VLSI web
    - including Email address
- Anyone without a group?

Computer accounts
- 1 account per group
  - Username vlsi08_11
    - for group 11
  - No password / blank
- Without email
- Without regular backup (!)

Computer rooms
- IAIK F1.01
  - Across Seminarraum
- Linux workstations
  - Dual1.student.iaik.tugraz.at – Dual7.student.iaik.tugraz.at
- Key available on request
  - 24 hours, 7 days a week

Remote access
- FreeNX
  - Sopteron.student.iaik.tugraz.at
- File transfer
  - Thalys.student.iaik.tugraz.at
Contact

Web
- VLSI Teaching
  http://www.iaik.tugraz.at/teaching/05_vlsi-design/
- IAIK StudentNet
  http://studentinfo.iaik.tugraz.at/

Newsgroup
- lv.vlsi-design
  news://news.tugraz.at/tu-graz.lv.vlsi-design

Johannes Wolkerstorfer
- Johannes.Wolkerstorfer@iaik.tugraz.at
- Tel 0316 873-5515
- IAIK, 1. Stock, Raum F2.10
- Contact hjours: always
  - roughly 9:00 – 18:00
  - Just drop in!
What next?

Read literature

High(est)-level model
  - Java or C/C++ program

Make first estimates

Play with IAIK designflow