VLSI Design Assignment Presentation

KU Summer Semester 2011
Low-Resource Block Ciphers

Thomas Plos
IAIK – Graz University of Technology
Thomas.Plos@iaik.tugraz.at
www.iaik.tugraz.at
Security-Related RFID Applications
RFID Technology

RFID … *Radio Frequency IDentification*

Contactless communication technique

Reader communicates with tag

Tags are highly constrained…

- … passive operation? → power is limited
- … high volume? → low price (small chip size)
- … battery supplied? → low energy

→ Security is challenging!
How is Security Achieved?

By using cryptographic algorithms

- Symmetric (block cipher/stream cipher)
- Asymmetric

Security services depending on application

- Anti cloning
- Anti eavesdropping
- Anonymity
- Secure key exchange/update
- Protected memory access
- Data integrity
Proprietary Algorithms

Proprietary algorithms were widely used...

... small circuits
... low power
... short execution time

... but ...

... implementations kept secret
... not standardized
... not publically evaluated

→ Easy target of attack if once published
Recent Incidents with Proprietary Algorithms

Most famous examples:
- DST-40
- KeeLoq
- CRYPTO-01
- Legic Prime
- Hitag2

→ Shift towards more secure and publically-available algorithms!
Which Algorithms are Suitable?

Near future: symmetric algorithms
Further future: asymmetric algorithms

Block ciphers
- Basic principle:
- Large “pool” of publically-available symmetric block ciphers (e.g. AES, XTEA, …)
- Not all block ciphers extensively evaluated for suitability in RFID applications
Assignment

Evaluate suitability of various symmetric block ciphers for integration on passive low-cost tags

Constraints
  - Power consumption
  - Chip size

Optimization goals
  - Minimize chip area
  - At given max. power consumption and performance

Compare with published work
# List of Block Ciphers

<table>
<thead>
<tr>
<th>Camellia-128</th>
<th>mCRYPTON-128</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEFIA-128</td>
<td>MISTY1</td>
</tr>
<tr>
<td>FOX-64</td>
<td>NOEKEON</td>
</tr>
<tr>
<td>HIGHT</td>
<td>SEA 96,8</td>
</tr>
<tr>
<td>IDEA</td>
<td>Serpent-128</td>
</tr>
</tbody>
</table>
Goals

Symmetric block ciphers

- Exploring
  - Three different algorithms (explore only given variant: e.g. Camellia-128 \(\rightarrow\) only variant with 128-bit key)
  - Randomly chosen per group!!!
  - Encryption & decryption
  - Input block and key are loaded into module at beginning of operation

- Estimating
  - Cost of approaches
  - \(A, t, P\) (area, clock cycles, power)

- Implementing
  - Most promising algorithm
  - Full functional standalone standard-cell test chip
    - Bus interface, data path, control logic

Selected algorithm

- Hardware-oriented high-level model (Java, C++, C)
- Architecture description

Optimization goal

- Circuit with smallest chip area
- Constraints:
  - \(20\text{ms}@f_{\text{CLK}}=847.5\text{kHz}\) for (for encrypting/decrypting one block of input)
  - \(P_{\text{MAX}}=1\text{mW}/\text{MHz}\)
- Also of interest
  - Low-power optimization
  - Simple architecture
    - “Keep the things easy”
  - Useful IO interface
    - IP module
    - AMBA APB
Requirements

Cost estimation
- Target technology: austriamicrosystems c35b4 0.35 µm CMOS
- For all three algorithms

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

Hardware architecture
- Optimization goal: area (with time and power constraints)

Hardware implementation
- Synchronous design
  • Synchronous clock
  • Asynchronous power-on reset
- Synthesizable HDL
  • VHDL or Verilog
- Simulation using TCL
  • TCL scripts
    - No HDL test benches
  • Testdata files

Test chip
- Standard-cell layout for c35b4 process
- Power simulation using Synopsys NanoSim

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

KU Journal
- “IngenieurInnen-Tagebuch”
Schedule & Deadlines

Tue March 15\textsuperscript{th} 2011 \quad Assignment presentation

Tue March 22\textsuperscript{nd} 2011 \quad TUGonline registration + group notification

Tue April 5\textsuperscript{th} 2011 \quad IAIK designflow presentation

Tue May 10\textsuperscript{th} 2011 \quad Intermediate deadline (email)

Tue June 21\textsuperscript{st} 2011 \quad Submission deadline (email) + (including colloquium)
Deliverables Intermediate

High-level model 30% of grading
- Model reflecting architecture (for algorithm with best estimate)
- Reference implementation is NOT a high-level model

Design document v1.0 10% of grading
- Not more than 10 pages
- Introduction including motivation, used literature
- Short description of three algorithms (difficulty of specification, suitable for HW)
- Hardware estimates (for all three algorithms)
  - Area, time, power consumption
- Hardware implementation
  - Proposed architecture (for algorithm with best estimate)
  - Architectural options (speed vs. area vs. power)

Submission deadline (May 10th 2011 23:59)
- Mail to Thomas.Plos@iaik.tugraz.at, Subject [VLSI11_gg]
Deliverables Final

Project files  50% of grading
  – Cleaned and zipped project directory

Design document v2.0  10% of grading
  – Fully blown design document (~ 15 pages)
  – In addition to document v1.0:
    • Detailed architecture description
    • Hardware results (area, time (incl. critical path), power consumption)
      1. Synthesis results
      2. Results from power simulation
      3. Picture of layout with Pads (from Virtouso)
    • Comparison of estimates and actual results
    • Conclusions

Submission (June 21st 2011, last date for colloquium)
  – Mail to Thomas.Plos@iaik.tugraz.at, Subject [VLSI11_gg]
  – Propose a date for the colloquium (Abgabegespräch)
Grading Summary

Design document v1.0 10%
High-level model 30%
Design document v2.0 10%
Project files (HDL, scripts) 50%
Colloquium (Abgabegespräch) “all know everything”
Typically one mark per group (announce problems early)
No discussions about algorithms after May 10th 2011 (distribution of algorithms may be unfair)
Example 1: 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];
}
```

Matrix notation:

\[
\begin{bmatrix}
S'_0,0 & S'_0,1 & S'_0,2 & S'_0,3 \\
S'_1,0 & S'_1,1 & S'_1,2 & S'_1,3 \\
S'_2,0 & S'_2,1 & S'_2,2 & S'_2,3 \\
S'_3,0 & S'_3,1 & S'_3,2 & S'_3,3 \\
\end{bmatrix} =
\begin{bmatrix}
2 & 0 & 3 & 0 \\
0 & 2 & 0 & 3 \\
0 & 1 & 0 & 2 \\
0 & 3 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
S_{0,c} & S_{1,c} & S_{2,c} & S_{3,c} \\
\end{bmatrix}
\]
Example 2: High-Level Model: Refined Model

```java
void ecGf2AffDouble() {     // Point double
    alu.loada(mem[mem_x]); // Point double
    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");
}
```

The reference implementation is no appropriate high-level model!
Hardware Estimation

Datapath (memory) will determine chip size
  - Control usually ~20%

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)
  - NAND2    (e.g. 55 µm² @ 0.35 µm)  ➔ 1 gate equivalent (GE)
  - Determine datapath size

Estimated chip size in GE =
  = (#flipflop*350 + #fulladder*275 + #mux*150 + etc.) / 55
  + 20% overhead

Estimated maximum clock frequency
  - Depends on critical path (longest combinational path, sum of cell delays)

Estimated power consumption
  - Per-gate values are given in µW/MHz
Hardware Implementation

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

IAIK designflow presentation on April 5th 2011
Layout
Groups and Accounts

Groups

- Groups of 3 students
  - Register on printed form NOW
  - OR email until March 22nd 2011
- Group members get listed
  - On VLSI web
- Anyone without a group?

Computer accounts

- 1 account per group
  - Username vlsiku_07
    - for group 7
  - Password see list (change under Windows at first logon)
- Without email account
- Without regular backup (!)
- Recommendation: use TUGOnline SVN repository

Computer rooms

- IAIK F1.01
  - Opposite to Seminarraum
- Linux workstations
  - (dual boot Win 7)
    - hp<xx>-lin.student.iaik.tugraz.at

Remote access

- Via SSH
  - servitus.student.iaik.tugraz.at
- Alternative
  - thalys.student.iaik.tugraz.at

Recommendation: use TUGOnline SVN repository
Contact

Web
- VLSI teaching
  http://www.iaik.tugraz.at/content/teaching/master_courses/vlsi_design/practicals/
- IAIK student net
  http://www.student.iaik.tugraz.at

Personal
- Thomas.Plos@iaik.tugraz.at
- Tel. 0316 873-5507
- IAIK, 2nd floor, room F2.11
- Contact hours: always
  - Roughly 8:30 – 18:00
  - Just drop in!

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

Organize/announce group
Read literature/websites/distributed algorithms
Run software implementations of algorithms
Make first estimations
High(est)-level model (Java/C++/C)
Play with IAIK designflow

Any questions?