Assignment

Stream cipher implementation

Plaintext XOR Keystream = Ciphertext
Ciphertext XOR Keystream = Plaintext
Motivation

Radio Frequency Identification (RFID)

Applications of RFID
- Supply-chain management
- Access control
- ePass
- ...

Security requirements of RFID
- Threats
  - Forgery and privacy
- Very limited resources on tag for crypto
  - Algorithms with low resources required
  - Chip area and power consumption are crucial
Architecture of Security-enhanced RFID Tag

Crypto unit
- Encryption and decryption of data

---

Ecrypt - eStream

Ecrypt Stream Cipher Project
- [http://www.ecrypt.eu.org/stream/](http://www.ecrypt.eu.org/stream/)

Submitted algorithms
- 34 algorithms (9 for SW, 13 for SW and HW, 12 HW)

Evaluation is ongoing

“Good” Candidates
- Trivium, Grain v2, Phelix, MICKEY-128, Hermes8-128, Achterbahn v2, Sfinks, Decim v2, VEST-8, Mickey, Edon80, Mosquito

Your task is to evaluate three of these algorithms
- Expertise on hardware implementation required

Further interesting links
- [http://cr.yp.to/streamciphers.html](http://cr.yp.to/streamciphers.html)
Goals

Stream cipher
- Exploring
  - Different (three) algorithms
- Estimating
  - Cost of approaches
- Implementing
  - Most promising algorithm
  - Standard-cell test chip

Selected algorithm
- Hardware-oriented high-level model
- Specification

Optimization
- Goal: Circuit with best "area*throughput"
- Also of interest
  - Low-power optimization
    - Power simulations
    - Most critical
  - Low-area optimization
    - Economic impact

Requirements

Hardware architecture
- Optimization goal: area*throughput
  - Key change every 1024 bytes

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

Cost estimation
- Target technology: Austriamicrosystems c35 0.35 µm CMOS

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

Test chip
- Standard-cell layout for c35 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
- Hardware results
  - Area, time, power consumption
  - Picture of layout
- Comparison of estimates and results
- Conclusions
- Literature

KU Journal (“IngenieurInnen-tagebuch”)
Schedule

Wed Mar 29. 2006  Assignment presentation (IAIK SR)
Wed Apr 05. 2006  Designflow-Demo (IAIK SR)
Wed May 17. 2006  Zwischenbesprechung (IAIK SR)
Wed Jun 07. 2006  letzter Termin für Abgabegespräch

Deadlines
– Registration    < 26. March 2006 (TUGonline)
– Intermediate   < 17. May 2006   (email)

Deliverables Intermediate

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

Design document v1.0
– Not more than 10 pages
– Introduction including motivation
– Short description of three algorithms
– Hardware estimates (for three algorithms)
  • Area, time, power consumption
– Hardware implementation
  • Proposed architecture (for algorithm with best estimate)
  • Architectural options (speed vs. area)
– Literature

Submission
– Mail to Feldhofer (cc Wolkerstorfer)
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 consumption
      1. Synthesis results
      2. Results from layout editor / power simulator
    - Picture of layout
  - Comparison of estimates and results
  - Conclusions

Submission
- Mail to Feldhofer (cc Wolkerstorfer)
- Propose a date for Abgabespräch

```
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

```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 c35 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
Hardware Implementation

Implementation of proposed architecture
- RTL description
  - Using Verilog / VHDL
  - Bit-accurate, cycle-accurate
  - Verified by simulation
- Synthesis
- Place-and-Route
- Post-layout simulation
  - Power simulation
- Backend verification

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 visi06_07
    - for group 7
  - No password
  - 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

Remote access
- SSH + X11
  - Servitus.student.iaik.tugraz.at
  - Dual1.student.iaik.tugraz.at – Dual7.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

Martin Feldhofer
- Martin.Feldhofer@iaik.tugraz.at
- Tel. 0316 873-5551
- IAIK, 1. Stock, Raum F2.18
- Sprechstunden: immer
  • Ca. 8:00 – 17:00
  • Einfach vorbeischauen!

Johannes Wolkerstorfer
- Johannes.Wolkerstorfer@iaik.tugraz.at
- Tel 0316 873 5515
- IAIK, 1. Stock, Raum F2.15
- Sprechstunden: immer
  • Ca. 9:00 – 19:00
  • Einfach vorbeischauen!

What next?

Read literature
Run software implementations
Make first estimates
Play with IAIK designflow