# Scaling ECC Hardware to a Minimum

Workshop

CRASH 2005

CRyptographic Advances in Secure Hardware

September 6th-7th 2005, Leuven (Belgium).

Johannes.Wolkerstorfer@iaik.tugraz.at

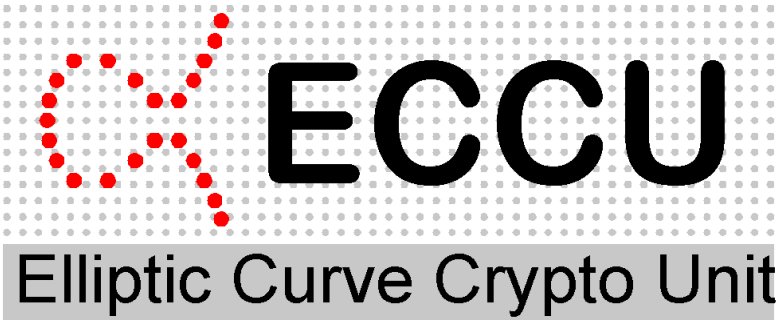
*Institute for Applied Information Processing and Communications (IAIK) — VLSI Group*

*Faculty of Computer Science*
*Graz University of Technology*

VLSI

# Outline

- Motivation
- Short introduction to ECC
- Implementation options of ECC (hardware)
- Optimization goals
- Design methodology
- ECC processor suitable for RFID

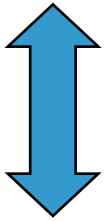



- Conclusions and future directions

# Motivation

- Small ECC hardware: What for?
  - Application
    - Authentication of small devices
      - RFID tags: Privacy, anti forgery
    - Security of sensor network nodes
      - Secure wireless communication
  - Goals
    - Low power
    - Low die size
    - Implementation security
  - Alternatives
    - Symmetric crypto: Key distribution issue
    - Other algorithms: RSA, XTR, NTRU

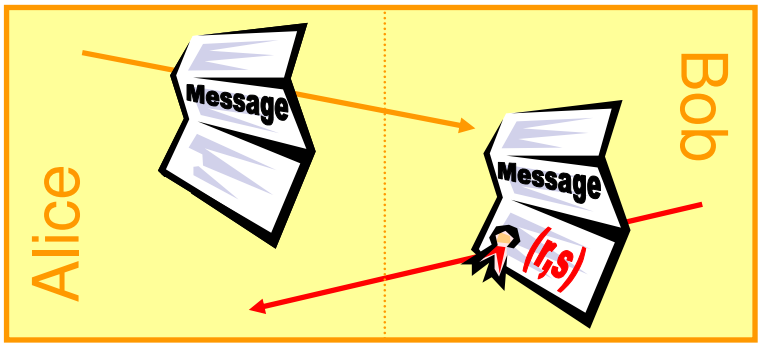
© Picture: Tagstore



© Picture: TI

# Elliptic-Curve Cryptography

- Protocol
  - Challenge-response authentication
- Algorithm
  - ECDSA: elliptic-curve digital signature algorithm
- Computation
  - Scalar multiplication
    - Repeated Doubling and addition of curve points
      - Finite-field operations (160-bit … 256-bit)



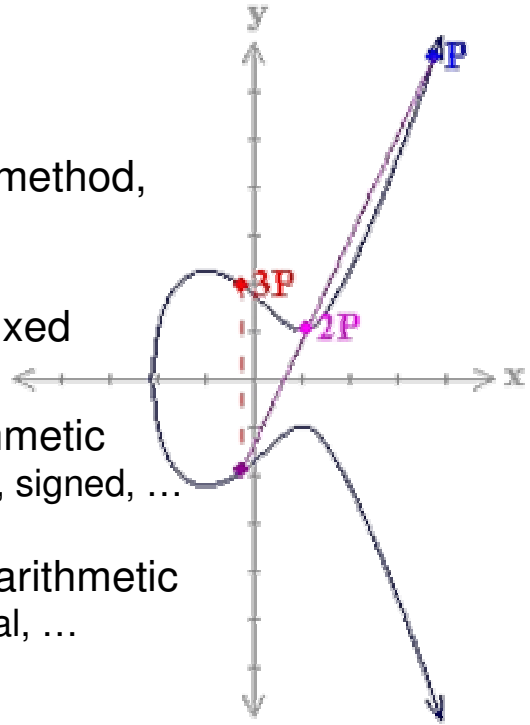

***ECDSA***

**e = SHA-1(Message)**
**k = random(1, n-1)**
**R = k*($P_x$,$P_y$) = ($R_x$,$R_y$)**
**r = $R_x$ mod n**
**s = $k^{-1}$·(e + d·r)**

$$2 \cdot P_1 = 2 \cdot (x_1, y_1, z_1) = (x_3, y_3, z_3) = P_3$$
$$x_3 = (3x_1^2 + az_1^4)^2 - 8x_1y_1^2$$
$$y_3 = (3x_1^2 + az_1^4)(4x_1y_1^2 - x_3) - 8y_1^4$$
$$z_3 = 2y_1z_1$$

# Elliptic-Curve Cryptography
## Implementation Options

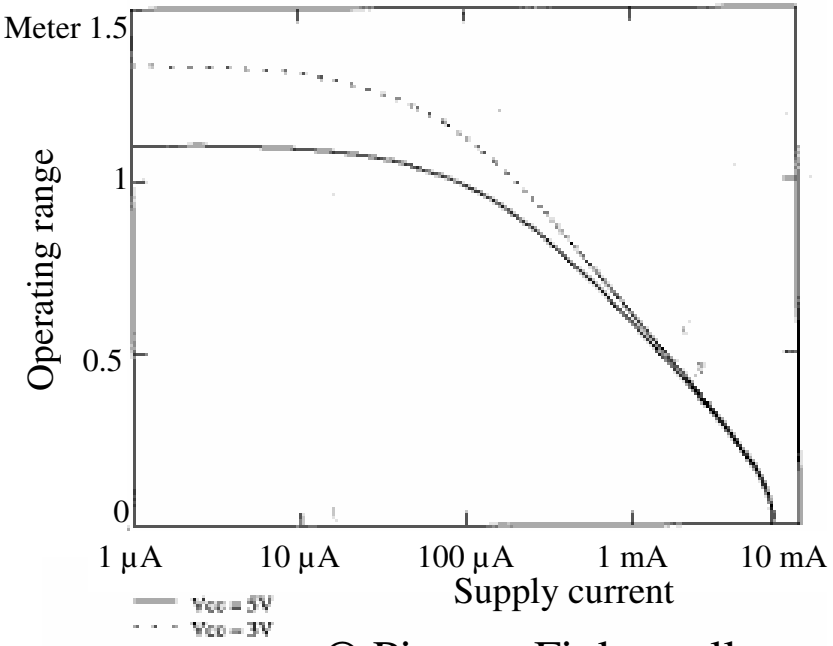
- Many options to implement ECC
  - **Elliptic curves**
    - Elliptic curves, hyper-elliptic curves
  - **Scalar multiplication**
    - Many algorithms: double-and-add, Montgomery method, NAF, window, comb, …
  - **Point operations**: addition and doubling
    - Many point representations: affine, projective, mixed
  - **Finite-field arithmetic**
    - Prime fields GF(p): modular integer arithmetic
      - Different representations: Montgomery, redundant, signed, …
      - Simplifications: generalized Mersenne primes
    - Binary fields GF($2^m$): modular polynomial arithmetic
      - Different bases: polynomial, normal, optimal normal, …
      - Simplifications: trinomials, pentanomials
    - Other fields: optimal extension fields OEF




**Designing ECC hardware is not straight-forward!**

# Requirements of Small Devices (Passively Powered Tags)

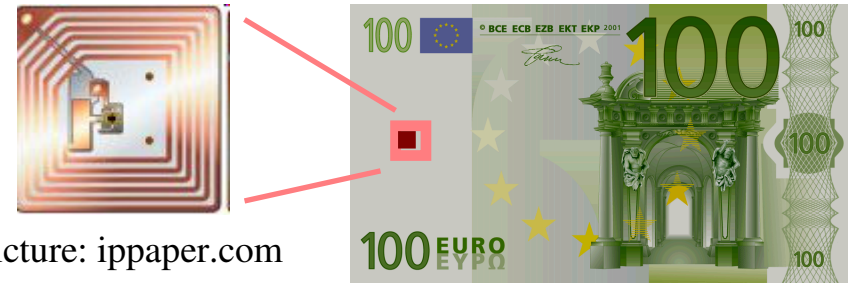
- Area
  - Determines cost
  - Maximum size depends on added value
  - Microcontroller too large
- Power
  - More stringent than area
  - Power more important than energy
  - Low clock frequency
- Power ctd.
  - Excessive peak power shortens operating range




© Picture: Finkenzeller

# Requirements of Small Devices (Passively Powered Tags)

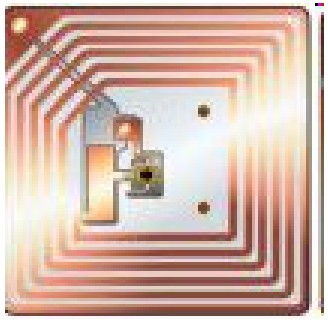
- Performance
  - Low clock frequency = low throughput
- Bandwidth
  - Kilobits / second
  - Many short messages more costly
    - than a few long messages
  - Half duplex; Reader talks first
- Security
  - Robustness against side-channels attacks




© Picture: ippaper.com

# Optimization Goals
## ECC for Passive Tags

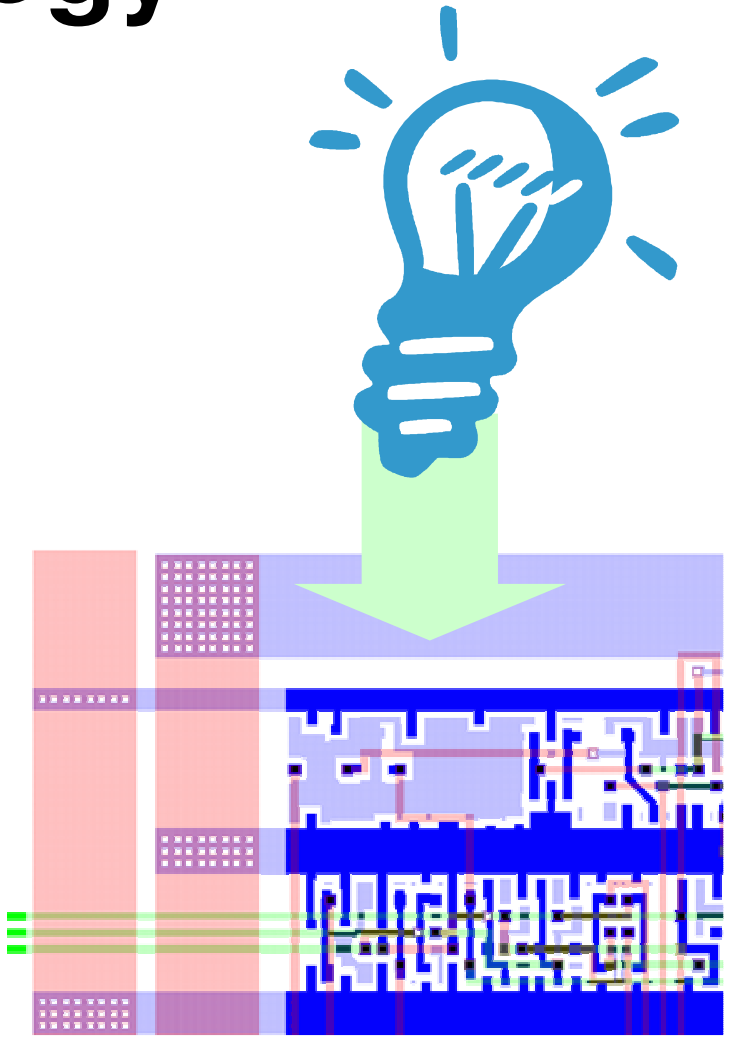
- ECC for 13.56 MHz RFID tags
  - Area
    - Less than 1 mm²
  - Power
    - Passively powered
    - I < 10 µA @1.5 V
    - To guarantee 1 m operating range
  - Performance
    - 300 µs
  - Security:
    - ECC > 160-bit
      - GF($2^{191}$)
      - GF($p_{192}$)
  - Manageable control



© Picture: ippaper.com

# Design Methodology

- Top-down design methodology
  - Design space exploration
    - Evaluation of design options
    - Optimization for target application
  - Focus on
    - High-level models
    - Early estimates
- Parameterizable VHDL
- Target technology
  - Standard-cell circuit
  - Mixed-signal technology
    - 0.35 µm — 180 nm CMOS
    - NV-RAM available



# ECC Hardware

## Scaling ECC to a Minimum

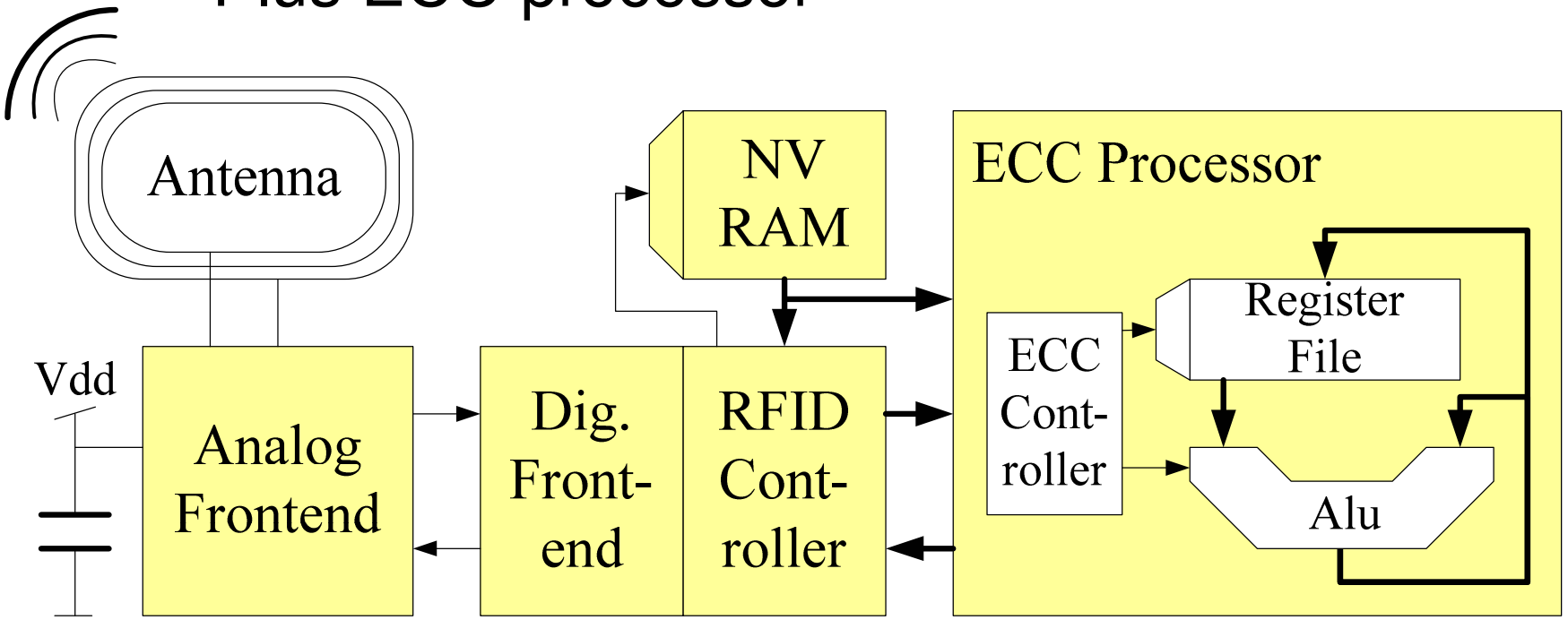
- Which functionality?
  - Scalability?
  - Different curves?
  - Different fields
    - Dual field?
  - DPA resistance
    - ECIES-dec only!
- Tricks applicable?
  - Pre-computation?
  - Early computation!
- What ECC hardware?
  - Instruction set extension
    - MAC unit
  - Finite-field coprocessor
  - EC processor
- What in hardware?
  - Multiplier!
    - Full precision!
    - Bit serial!
  - Adder / Subtractor!
  - Squarer?
  - Inversion unit?
  - Modular reduction!
    - Fixed modulus or
    - Montgomery mult.
  - Memory
    - Register file
    - RAM
  - Programmable control?

**Explore your needs early!**

# Architecture

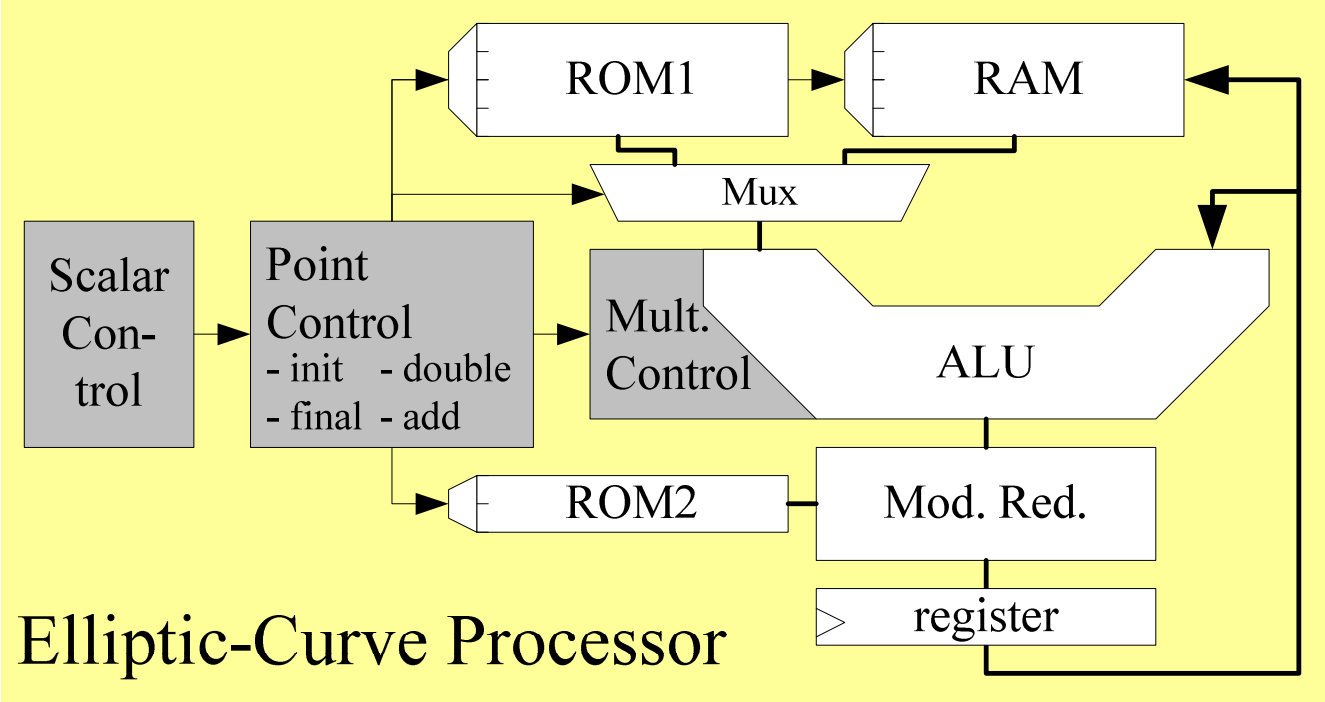
## ECC-Enabled RFID Tag

- Conventional tag architecture
  - Plus ECC processor



# Architecture

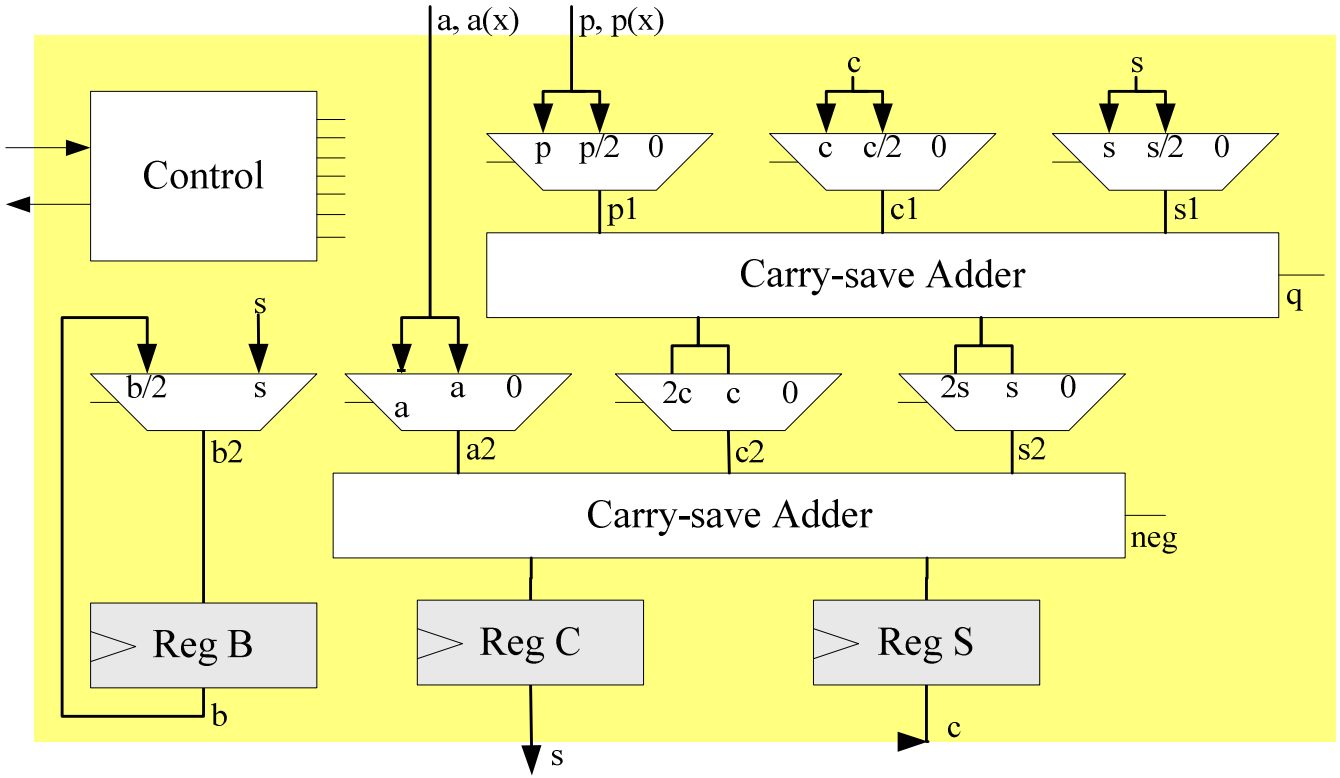
## ECC Processor: ECCU




- Full-precision architecture
- Supports different finite fields

# Architecture

## Dual-Field Arithmetic Unit

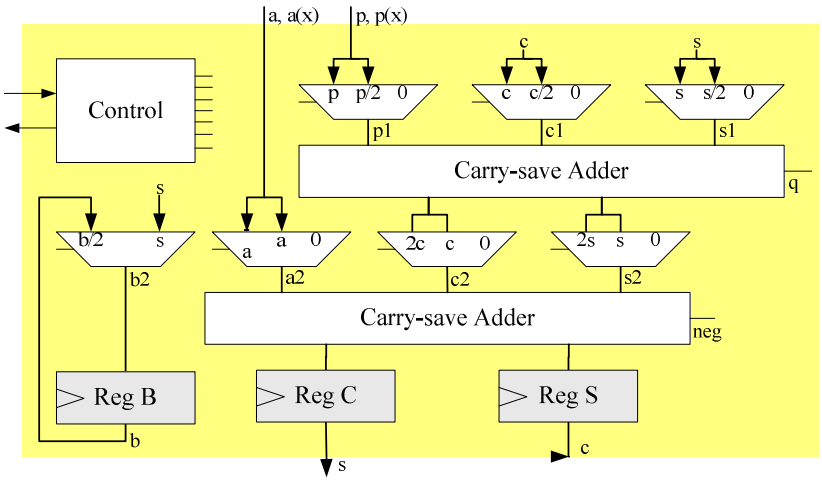



- Operates in GF($2^m$) and GF(p)
- Redundant representation of GF(p)

# Results: Arithmetic unit

- Operations supported by arithmetic unit
- Many HW resources reused
- Dual-field capability at almost no overhead
- Uses Montgomery multiplication

| ***Name*** | ***Function*** (s,c)= | ***Name*** | ***Function*** (s,c)= |
|---|---|---|---|
| Clear | (0, 0) | | |
| Hold | (s', c')→(s'', 0) | Load | (a, 0) |
| Add | (s+a, c) | Sub | (s-a, c) |
| Shftl | (2s, 2c) | Shftr | ((s+p·q)/2, c/2) |
| $Mul_0$ | ($a \cdot b_0$, 0) | $Mul_i$ | ($(s+p \cdot q)/2+a \cdot b_i$ +, c/2) |



# Results

- Cycle Count

| Field | Mul | Inv | EC |
|---|---|---|---|
| $p_{192}$ | 197 | 11k2 | 677k |
| $p_{224}$ | 229 | 14k4 | 905k |
| $p_{256}$ | 261 | 17k7 | 1M1 |
| $2^{191}$ | 197 | 6k2 | 426k |
| $2^{233}$ | 241 | 7k5 | 635k |
| $2^{283}$ | 289 | 8k8 | 920k |

- Area (0.35 µm CMOS)

| Size [bit] | Area [mm²] | Gates [GE] |
|---|---|---|
| 192 | 1.31<br>0.45+0.66+0.2 | 23k |
| 224 | 1.51<br>0.54+0.77+0.2 | 27k |
| 256 | 1.71<br>0.62+0.89+0.2 | 31k |

ALU + RAM + Control

# Results

## On Actual CMOS Processes

- Size, performance, and power

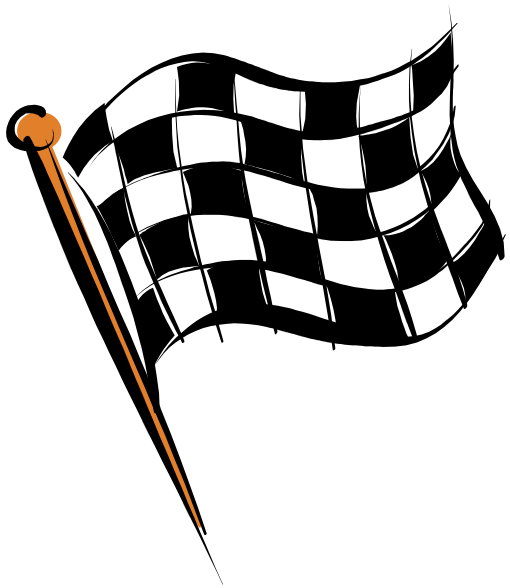
| ***CMOS*** | ***ECC Processor* (196-bit)** | | | |
|---|---|---|---|---|
| $l_{gate}$ [nm] | Area [mm²] | Power [µW/MHz] | $f_{max}$ [MHz] | EC [kP/s] |
| 350 | 1.31 | 500 | 68.5 | 101.1 |
| 180 | 0.35 | 170 | 225 | 332.1 |
| 90 | 0.09 | 55 | 600 | 885.6 |

- Most efficient ECC processor (area)
  - Reported in literature so far!

# Results

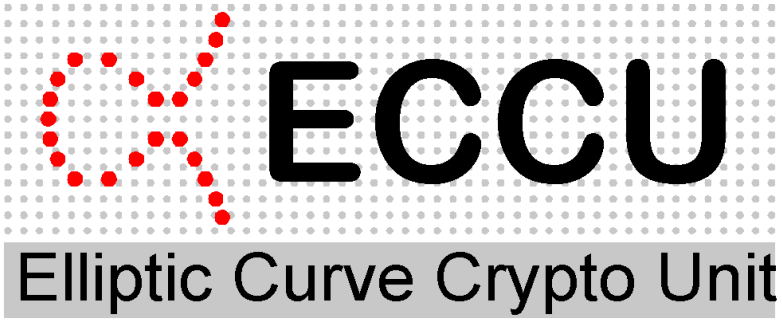
## Does ECCU fit RFID?

- Area on 0.35 µm CMOS
  - No – too large: 1.31 mm²
- Area on 180 nm CMOS
  - **YES** – 0.35 mm² is realistic
- Power
  - **YES**! Constraints can be met by
    - Lowering clock frequency (e.g. 175 kHz @180 nm)
- Performance (@ 180 nm)
  - Poor on RFIDs: > 1 second (@ 175 kHz)
  - But: 330 ops / second (@ $f_{max}$ = 225 MHz)



# Conclusions

- Thorough analysis
  - Many choices for ECC
  - Tailored hardware!
- Achievements
  - Novel arithmetic unit
    - Dual-field operation: GF(p) and GF($2^m$)
  - Area (and power consumption)
    - Suitable for RFID implementation
- Outlook
  - Hardwired control
  - More efficient register file



# Future directions
## ECC Hardware

- Challenges to solve
  - Problems of ECC
    - Too many standards
      - Restriction to prime fields useful?
    - Unclear situation with patents
  - Hashes
    - Bulky HW implementation
  - Protocols
    - Authentication protocols
      - Without hashes
      - Standards