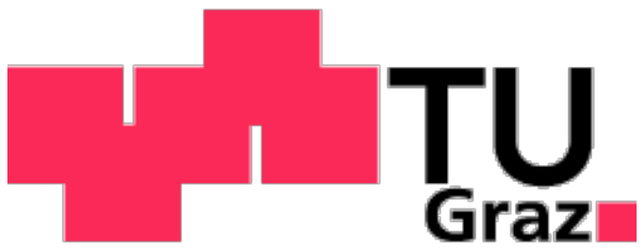
Solving the Discrete Logarithm of a 113-bit Koblitz Curve with an FPGA Cluster

Erich Wenger and Paul Wolfger
Graz University of Technology
WECC 2014, Chennai, India



We solved…

- the discrete logarithm of a 113-bit Koblitz Curve.
- Challenge generated using SHA-256
- Extrapolated 24 days on 18 Virtex-6 FPGAs

ECDLP Records

- In 2000:
  Binary Koblitz curve - ECC2K-108
  using 9,500 PCs in 126 days

- In 2004:
  Binary elliptic curve - ECC2-109
  using 2,600 PCs for 510 days

- In 2012:
  Elliptic curve over 112-bit prime field
  using 200 Playstation 3 for 6 months

TU Graz Records

- IT-Security Lecture
  - 2012: 75 bit in days on quad-core
  - 2013: 80 bit in 17 days on Core i5-2400
- Master project:
  - Virtex 6 FPGA
  - 83 bit in (avg) 4.1 days
- Room for improvement…

BlueKrypt | Cryptographic Key Length Recommendation

In most cryptographic functions, the key length is an important security parameter. Both academic and private organizations provide recommendations and mathematical formulas to approximate the minimum key size requirement for security. Despite the availability of these publications, choosing an appropriate key size to protect your system from attacks remains a headache as you need to read and understand all these papers.

This web site implements mathematical formulas and summarizes reports from well-known organizations allowing you to quickly evaluate the minimum security requirements for your system. You can also easily compare all these techniques and find the appropriate key length for your desired level of protection. The lengths provided here are designed to resist mathematic attacks; they do not take algorithmic attacks, hardware flaws, etc. into account.

Choose a Method

- Lenstra and Verheul Equations (2000)
- Lenstra Updated Equations (2004)
- **ECRYPT II Recommendations (2012)**
- NIST Recommendations (2012)
- ANSSI Recommendations (2010)
- Fact Sheet NSA Suite B Cryptography (2013)
- Network Working Group RFC3766 (2004)
- BSI Recommendations (2014)

Compare all Methods

ECRYPT II

ECRYPT is a network of excellence in cryptology. This report [3] is driven by the "Security Level" you want to reach. To each of these levels corresponds a symmetric key size from which equivalent asymmetric key sizes are built in a similar way as the one used in NESSIE.

Level	Protection	Symmetric	Asymmetric	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash
1	Attacks in "real-time" by individuals *Only acceptable for authentication tag size*	32	-	-	-	-	-

The higher the security level…

…the lower the speed.

With knowledge on the best attacks…

…**realistic** security bounds are possible.

…potentially **smaller** parameters can be used.

…potentially **faster** algorithms can be used.

Elliptic Curve Discrete Logarithm Problem

$$Q = kP$$

Parallelized Pollard's Rho Algorithm

We are looking for…

$$X_1 = X_2$$

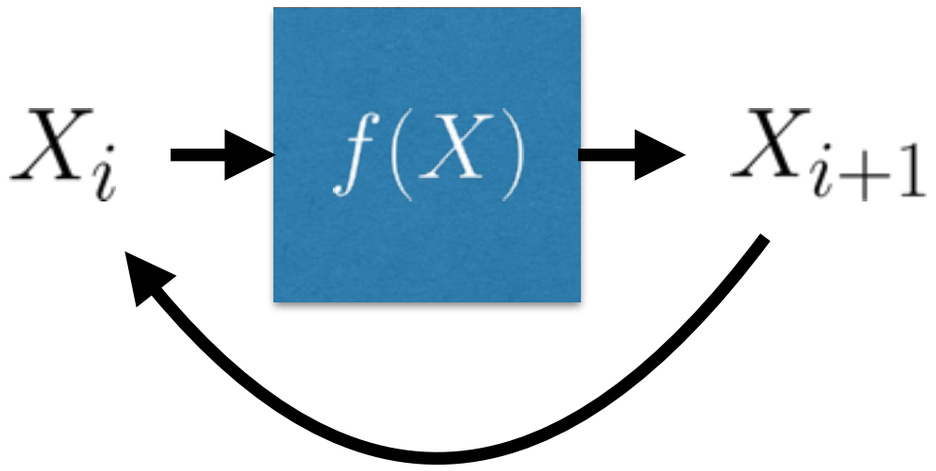
$$a_1 P + b_1 Q = a_2 P + b_2 Q$$

$$(a_1 - a_2) P = (b_2 - b_1) Q$$

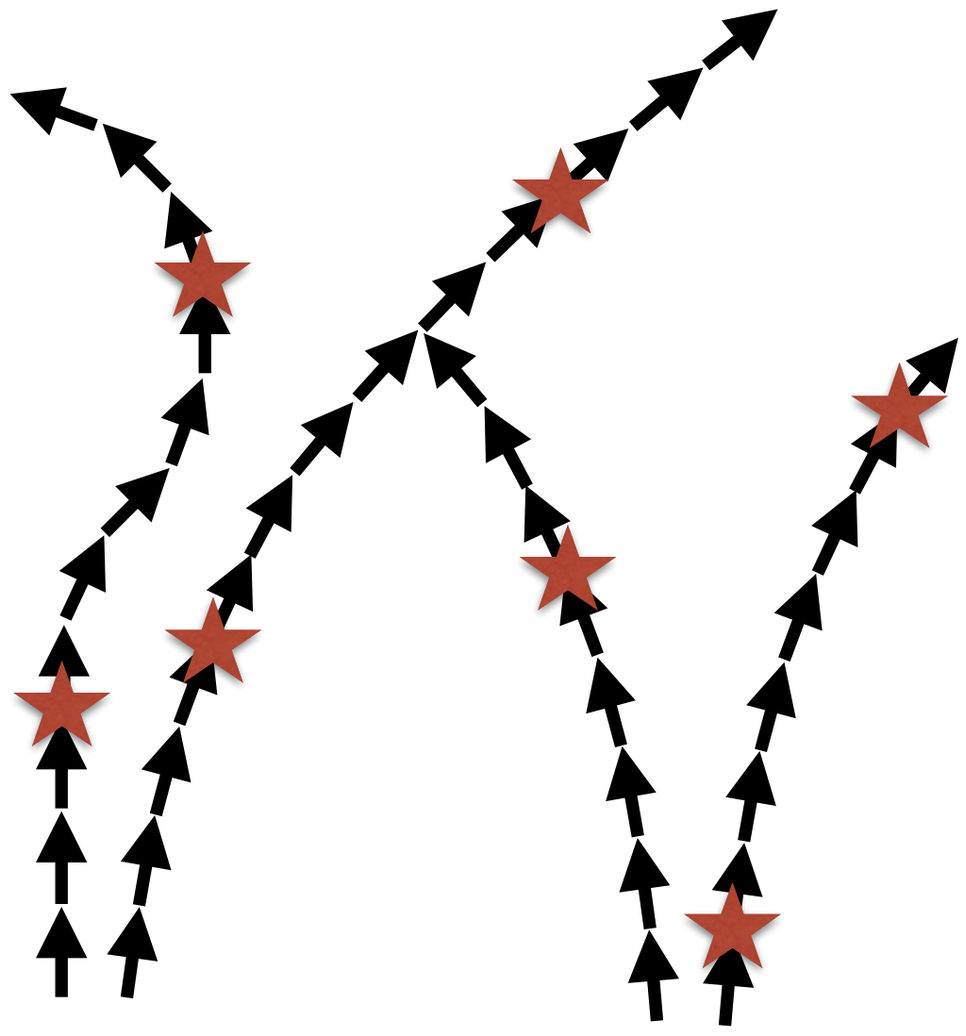
$$(a_1 - a_2) P = (b_2 - b_1) kP$$

$$k = \frac{a_1 - a_2}{b_2 - b_1} \bmod n$$

Pollard's Rho Algorithm



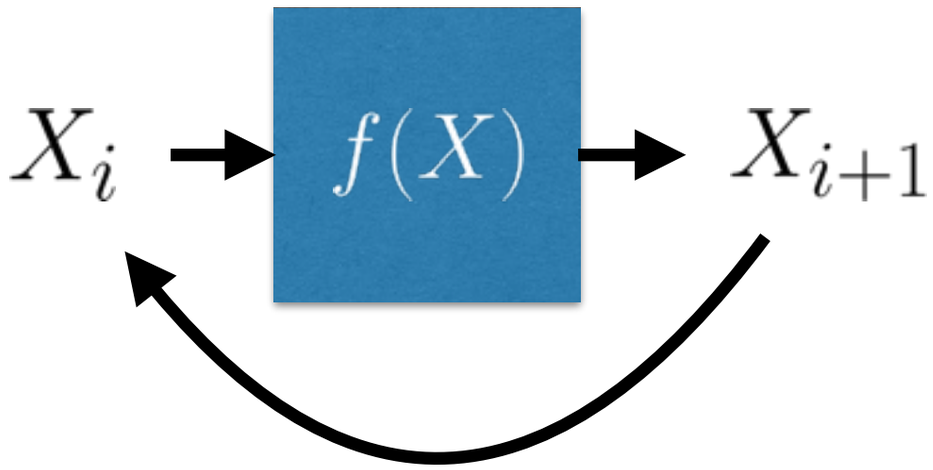
Iteration function



Parallelized Pollard's Rho

$$\frac{\sqrt{\pi n/2}}{m}$$

Iteration Function



$$X = aP + bQ$$

$$f(X) = X' = a'P + b'Q$$

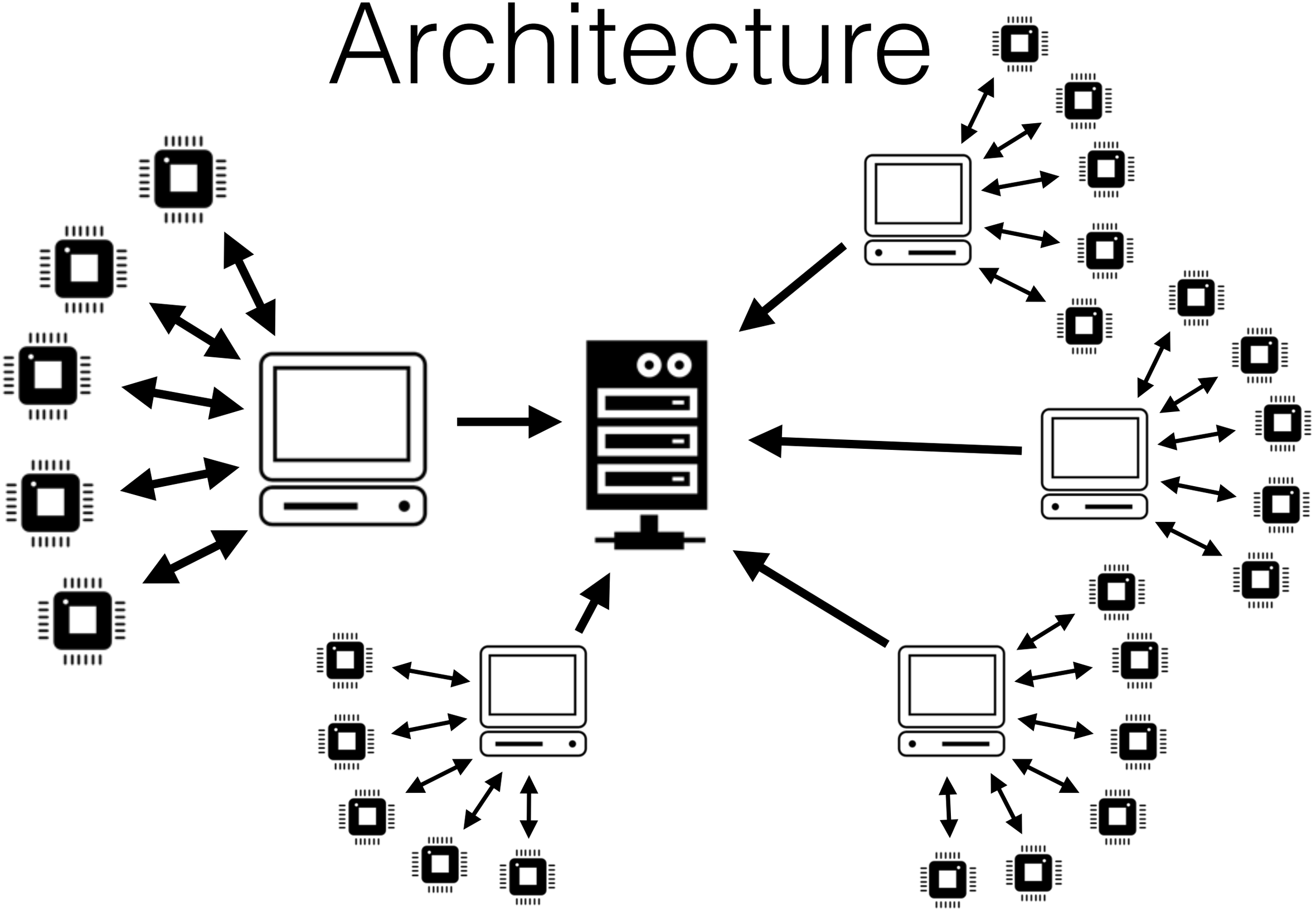
$$f(X) = ?$$

Iteration Function

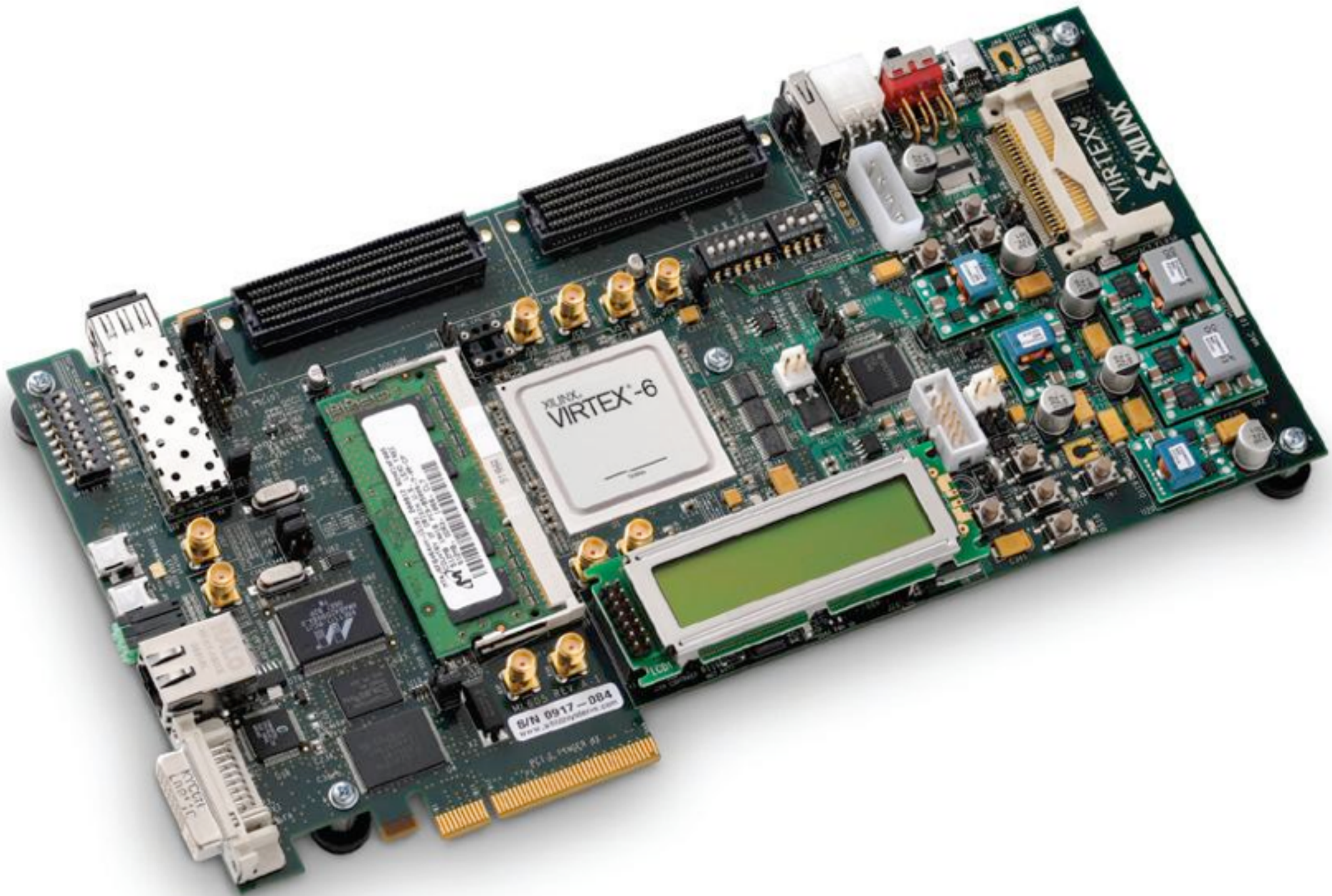
41-bit Koblitz Curve

Reference	Iteration function
Teske [29]	$f(X_i) = X_i + R[j]$
Wiener and Zuccherato [31]	$f(X_i) = \min_{0 \le l < m} \left\{ \sigma^l (X_i + R[j]) \right\}$
Gallant *et al.* [14]	$f(X_i) = X_i + \sigma^l (X_i)$
Bailey *et al.* [4]	$f(X_i) = X_i + \sigma^{(l \bmod 16)/2+3} (X_i)$

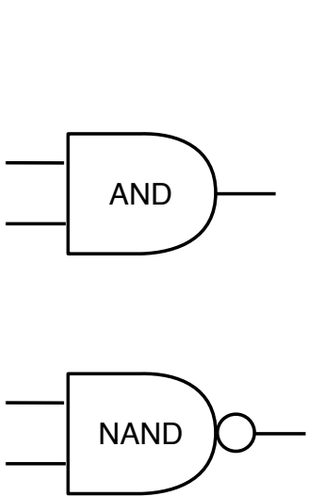
Architecture



FPGA Development Board

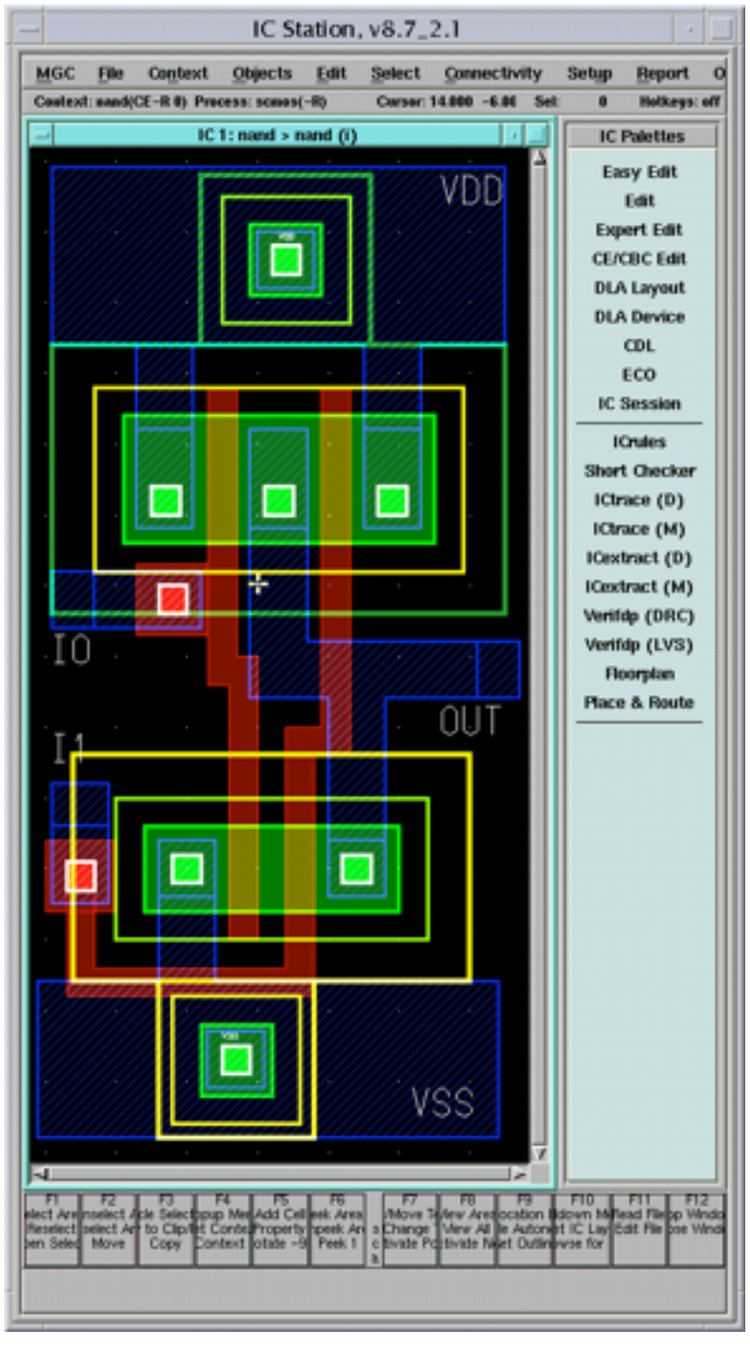


ASIC Design



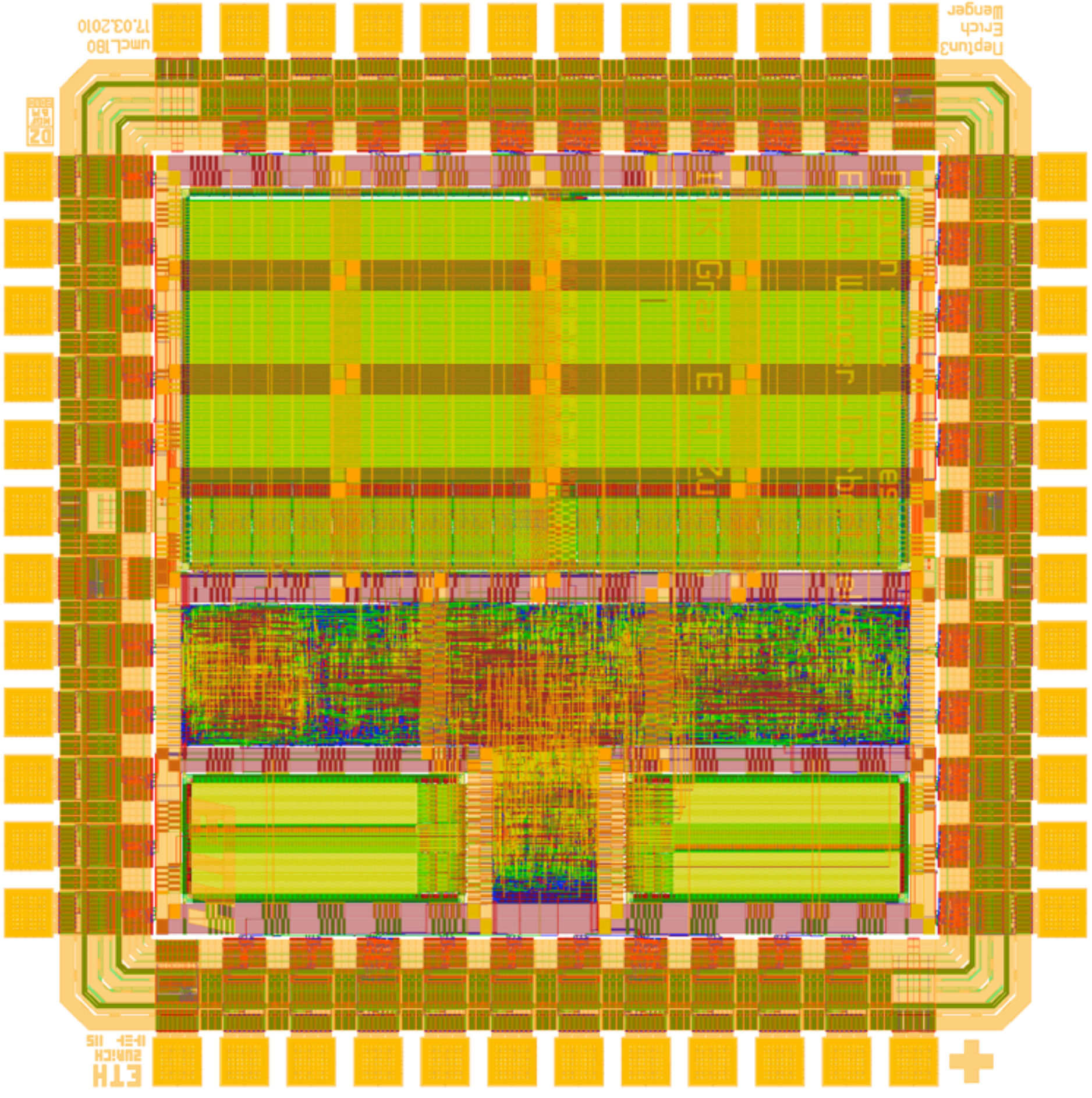
1.25

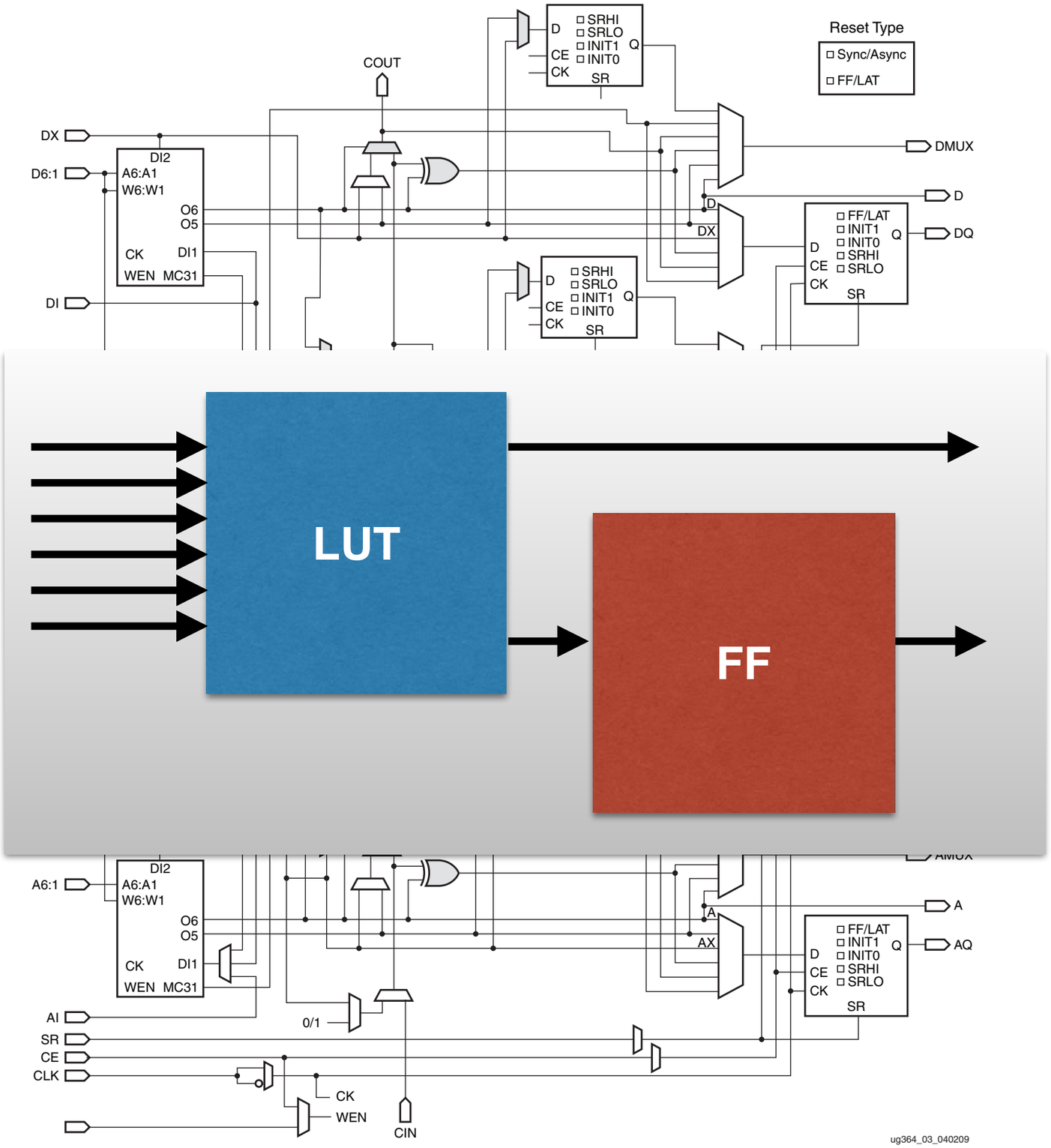
1 GE



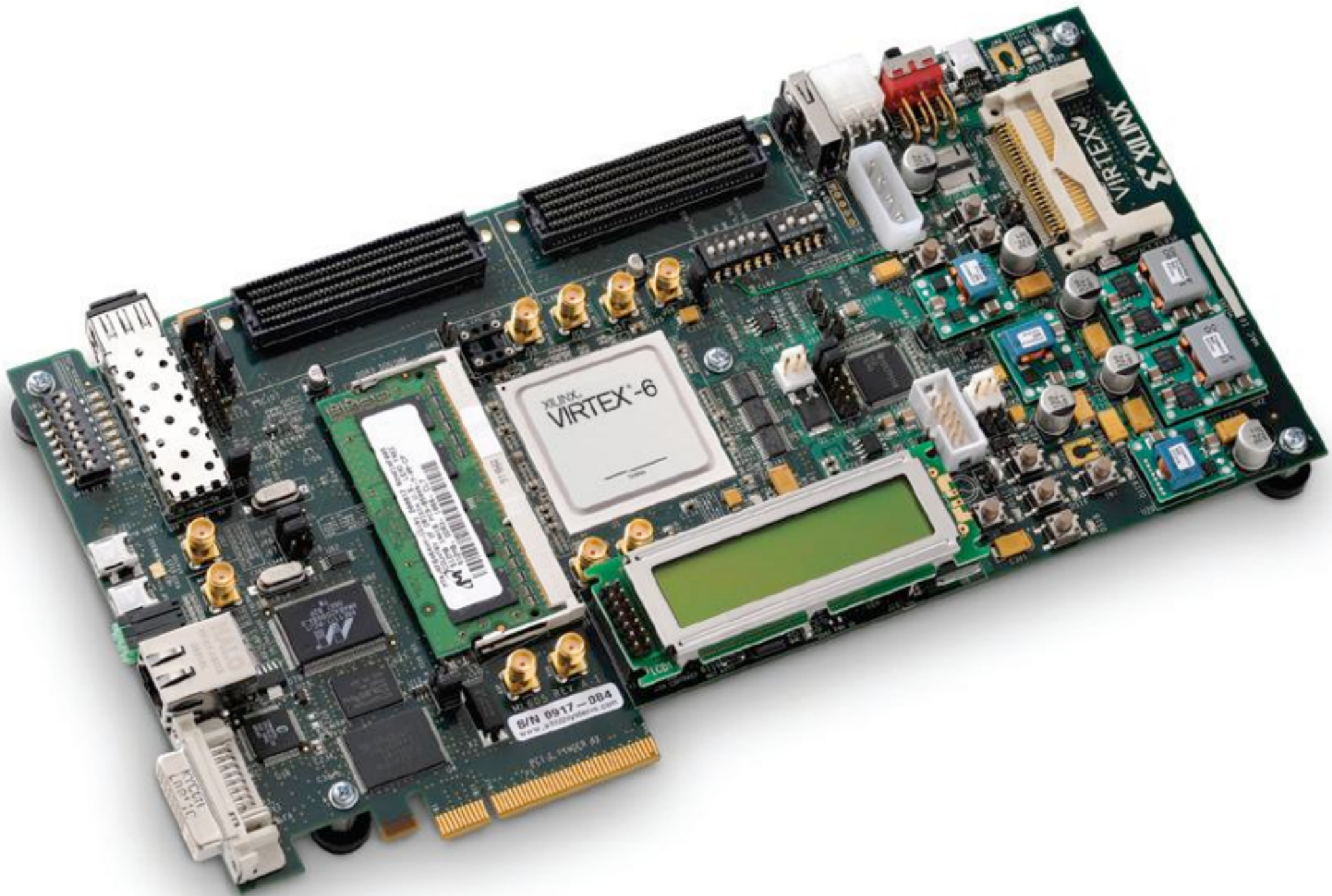
One NAND Gate:

- 4 Transistors
- 3.136 μm^2 @ UMC 90nm
- Register/Flip-flop:
  ~5 GE

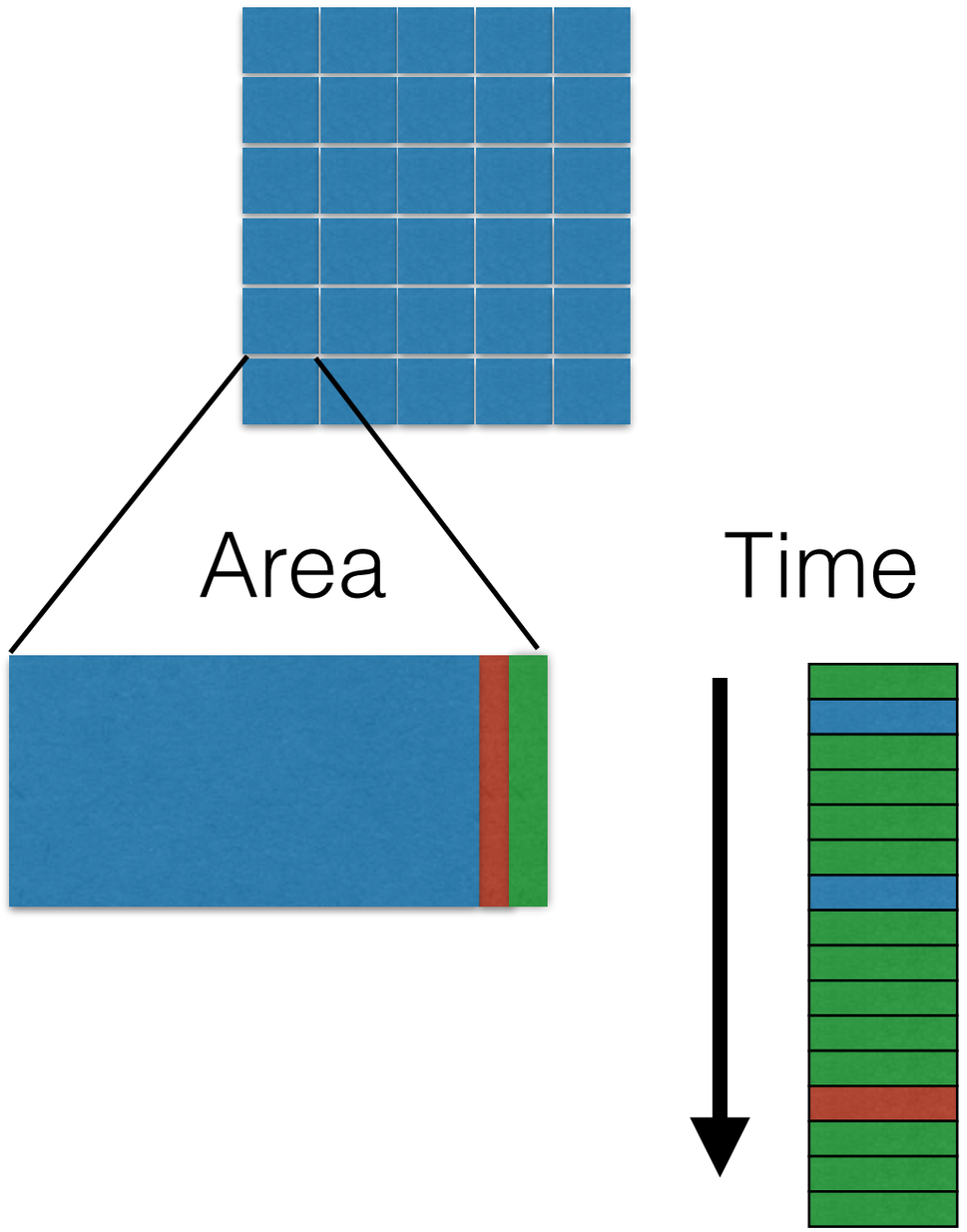

Neptun3
Erich
Wenger
umc L180
17.03.2010
ETH
ZURICH
IHEP IIS


COUT
DX
D6:1
DI
DI2
A6:A1
W6:W1
O6
O5
CK
DI1
WEN
MC31
D
SRHI
SRLO
INIT1
INIT0
CE
CK
SR
Q
Reset Type
Sync/Async
FF/LAT
DMUX
D
DQ
DX
FF/LAT
INIT1
INIT0
SRHI
SRLO
LUT
FF
A6:1
AI
SR
CE
CLK
0/1
CK
WEN
CIN
A
AQ
AX
ug364_03_040209

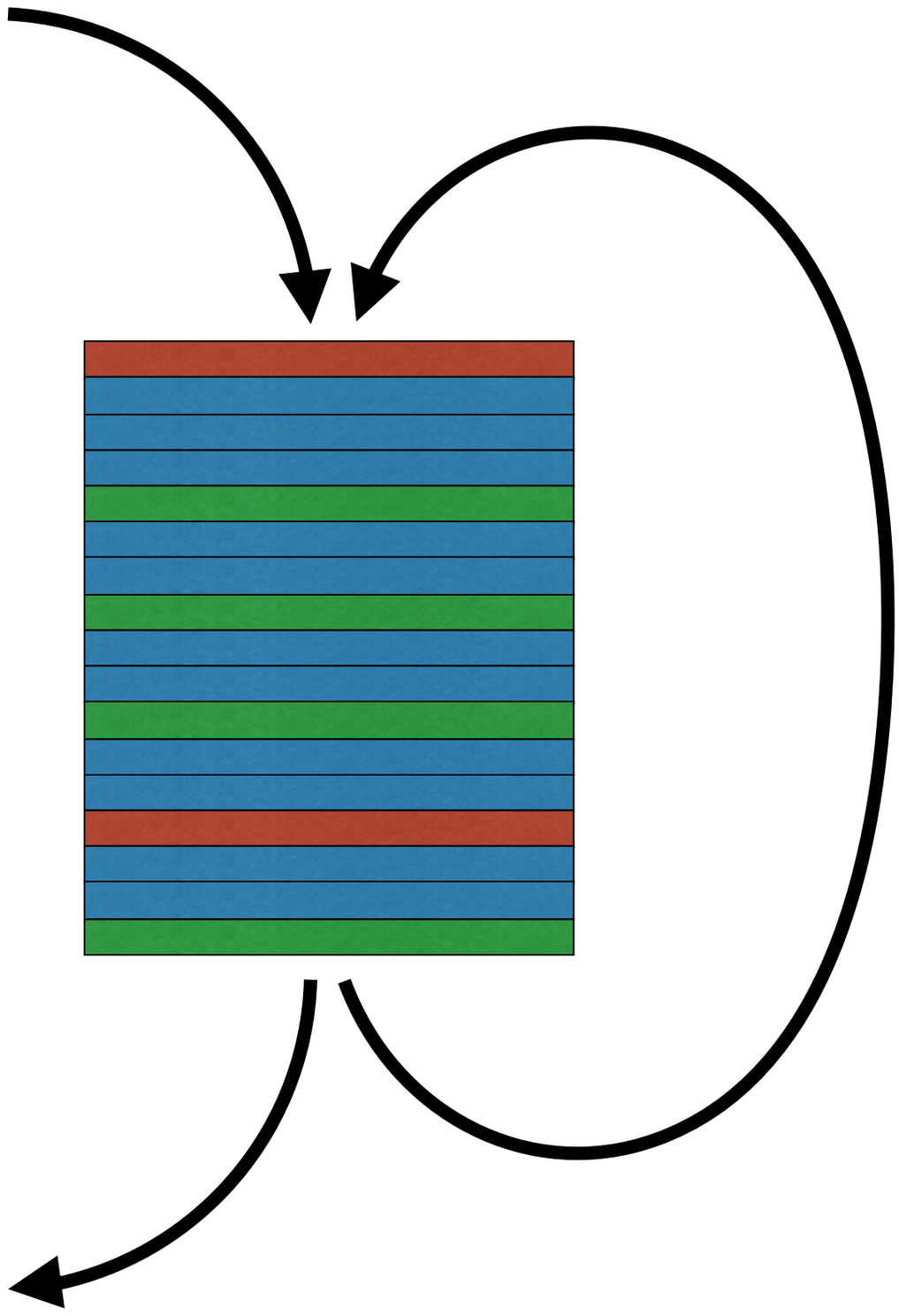
FPGA Development Board

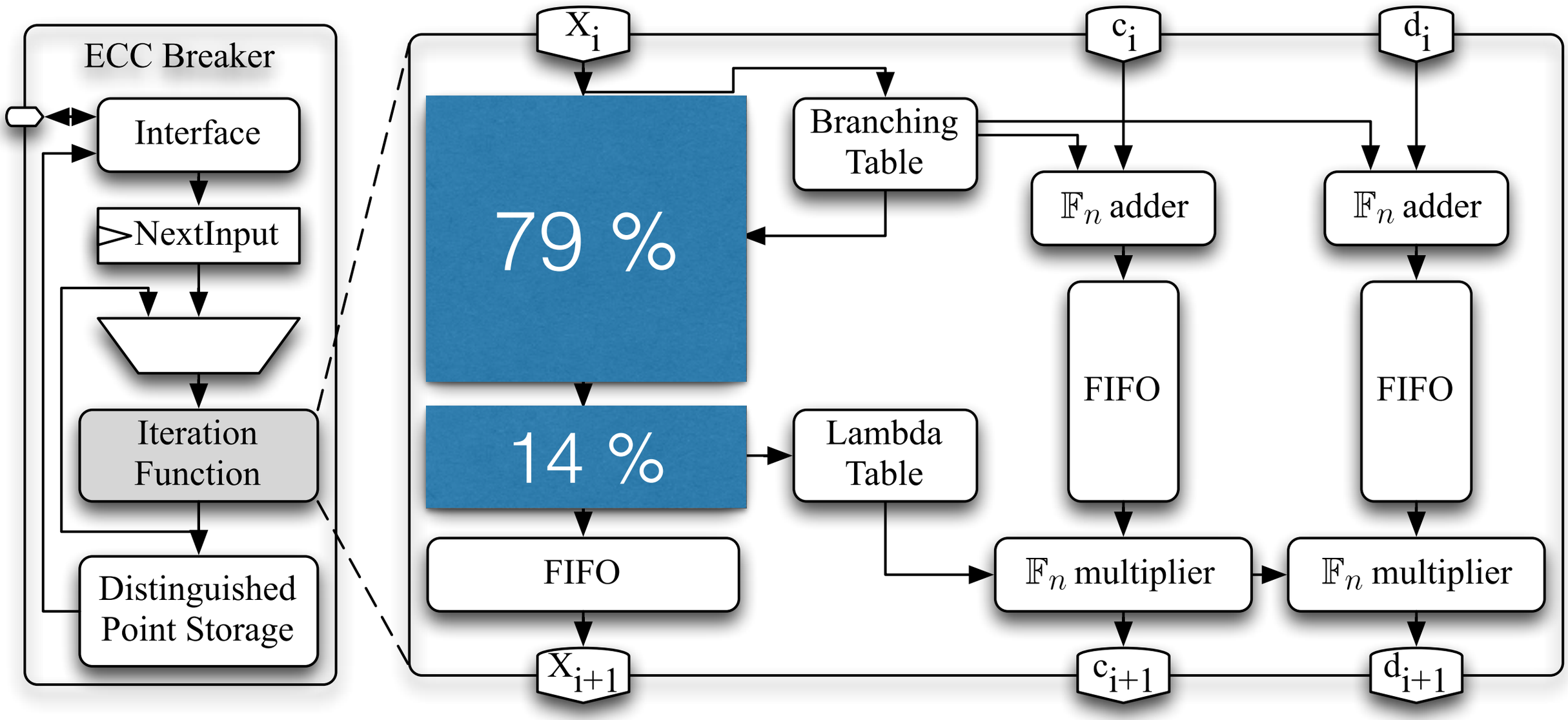


Multiple Small Cores

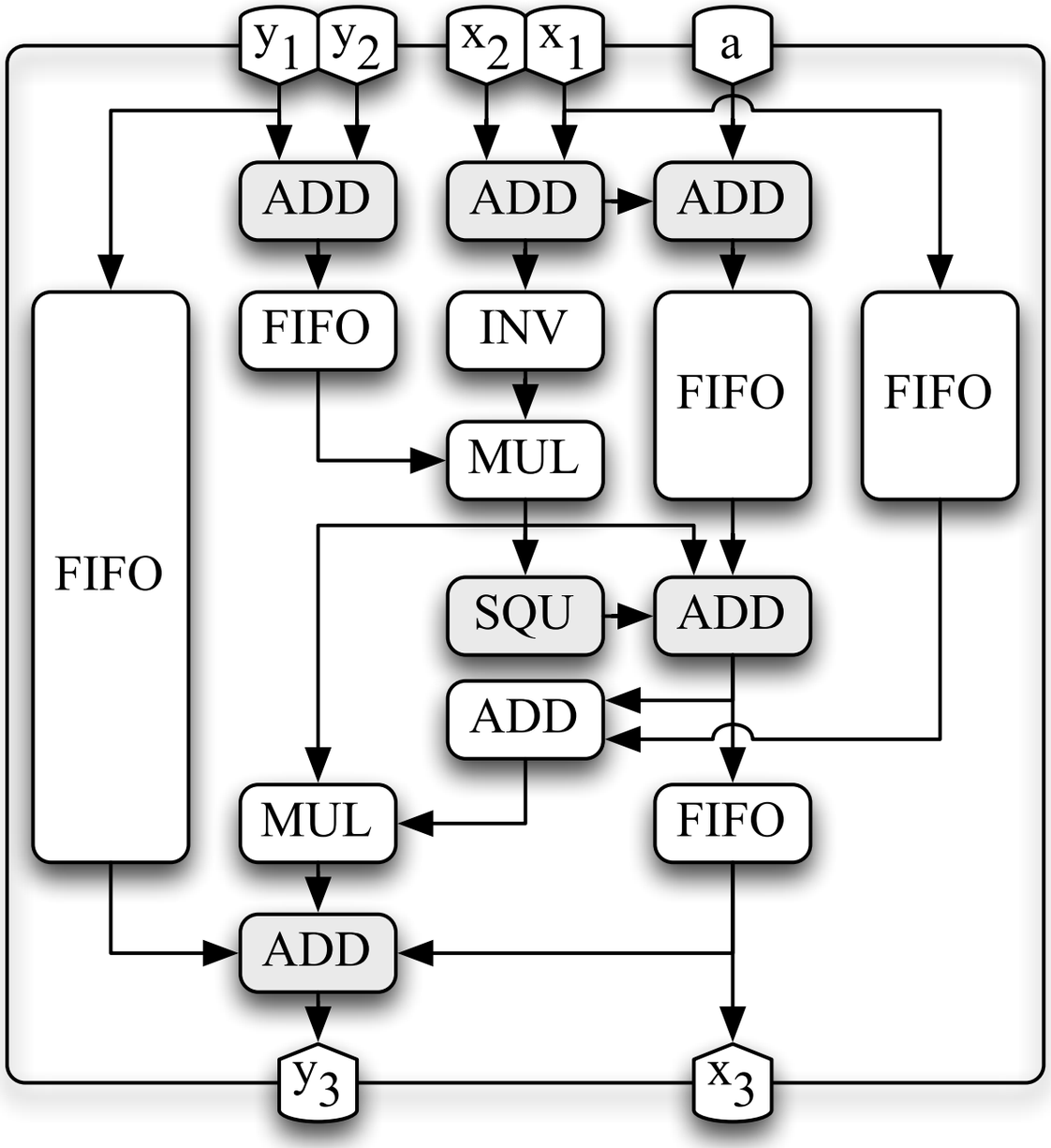


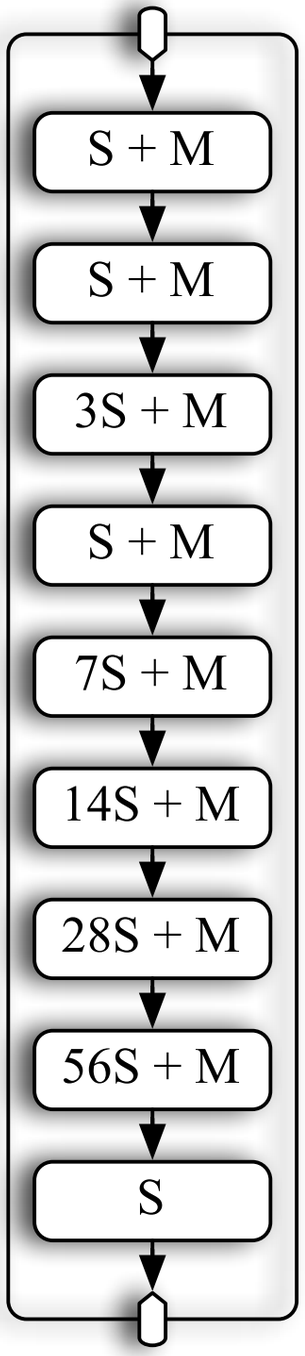
Core Idea




ECC Breaker
Interface
NextInput
Iteration Function
Distinguished Point Storage
X_i
c_i
d_i
79 %
14 %
Branching Table
Lambda Table
FIFO
$\mathbb{F}_n$ adder
$\mathbb{F}_n$ adder
FIFO
FIFO
$\mathbb{F}_n$ multiplier
$\mathbb{F}_n$ multiplier
X_{i+1}
c_{i+1}
d_{i+1}

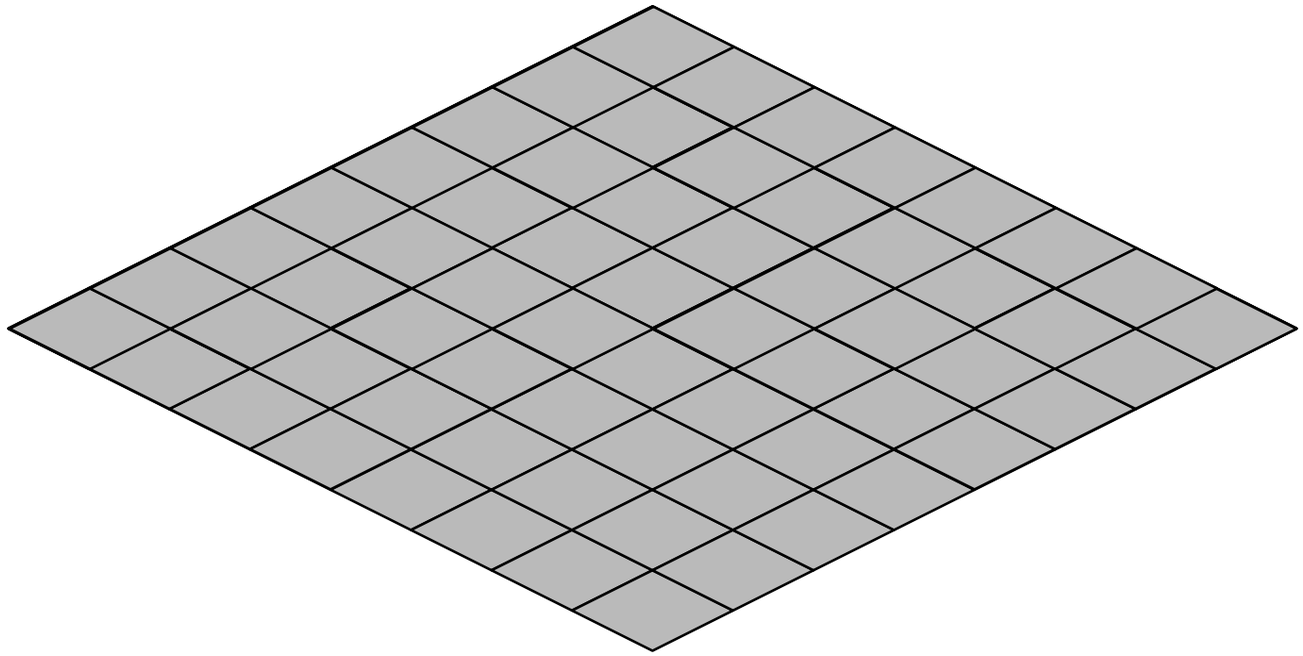
Point Addition and FF Inversion



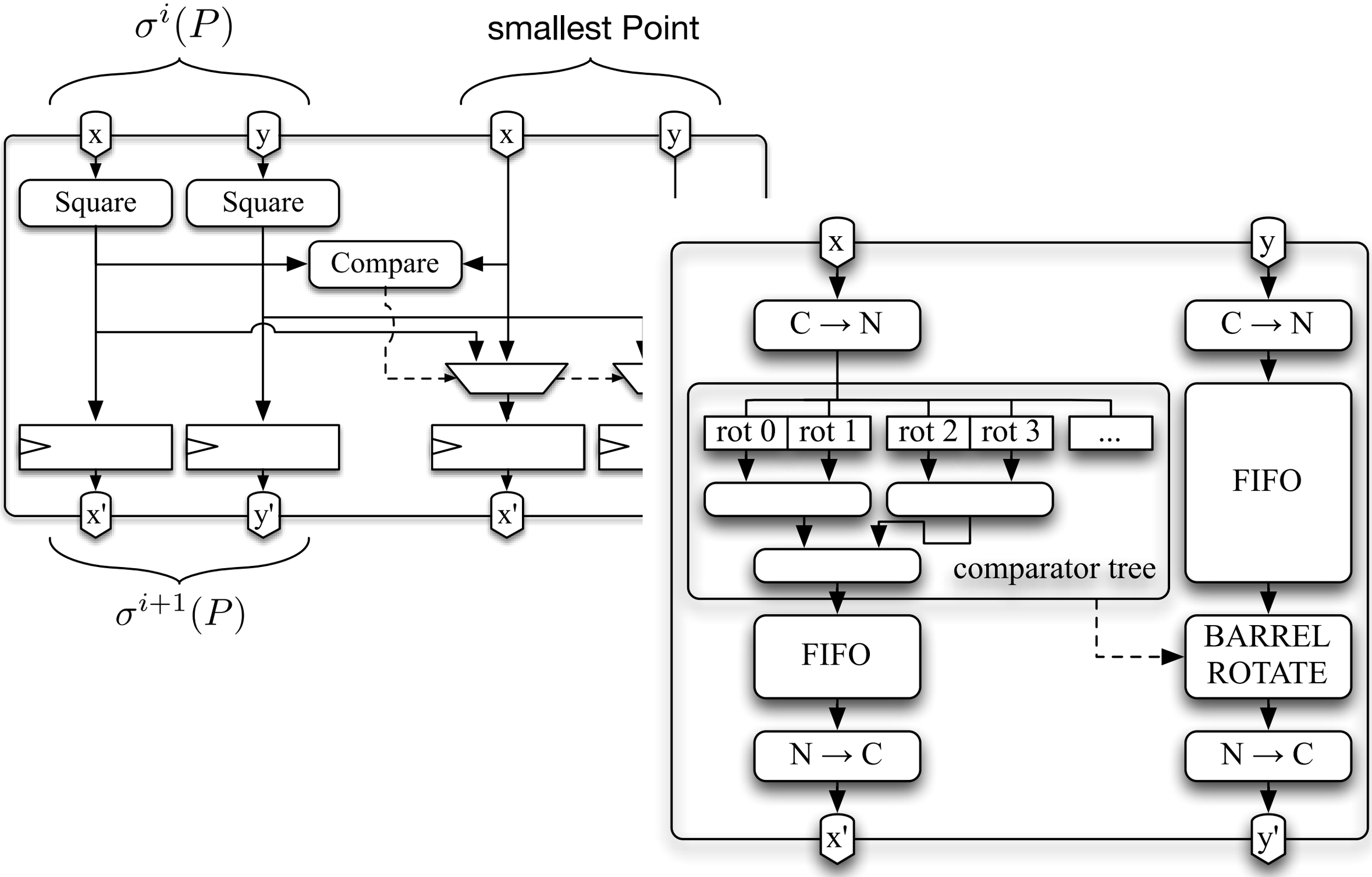


Binary Field Multiplier

Method	Size
Parallel	5,497 LUTs
Mastrovito	7,104 LUTs
Bernstein’s Batch Binary Edwards	4,409 LUTs
Recursive Karatsuba	3,757 LUTs



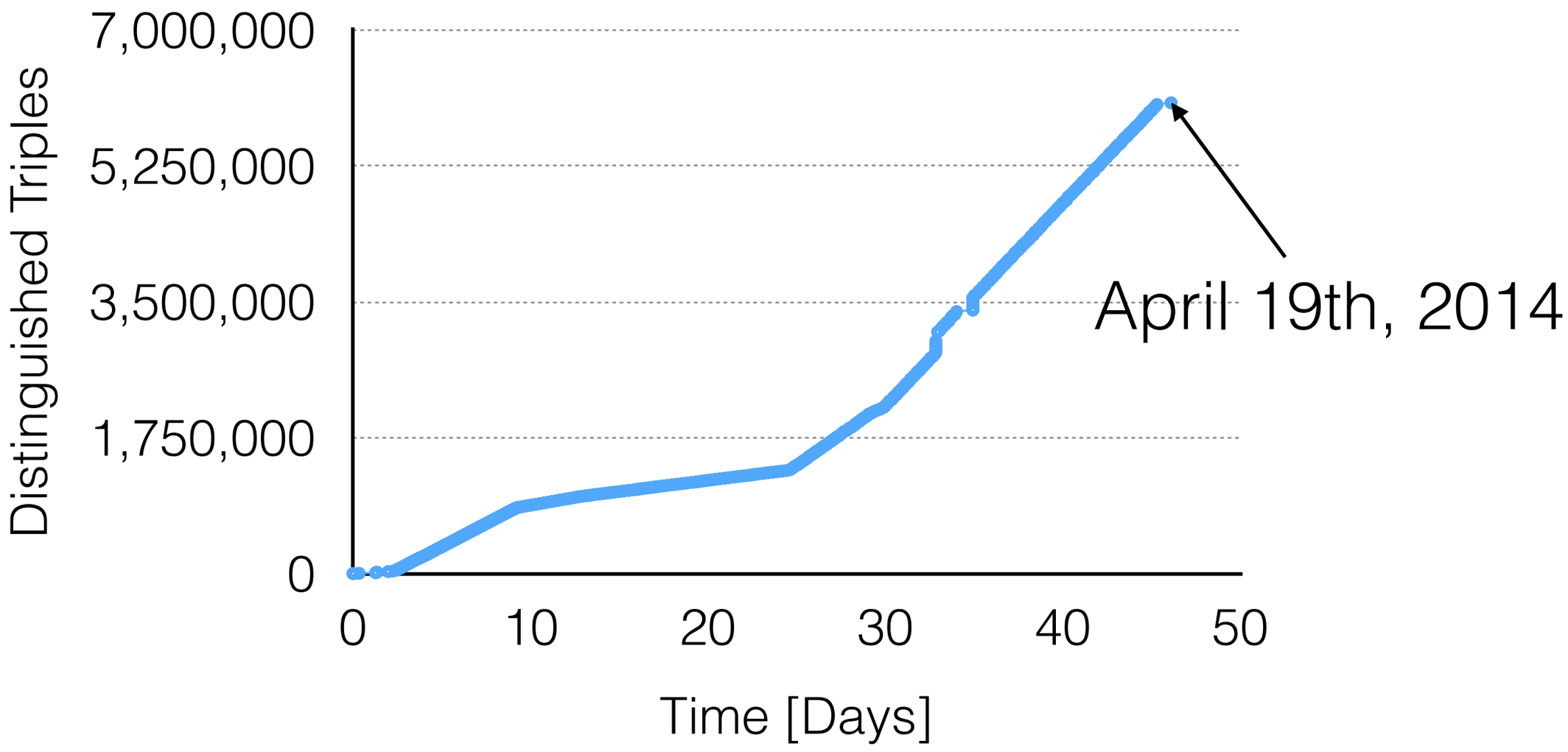
Point Automorphism



Details

- 210 pipeline stages
- Per default: canonical basis
- Normal basis used for point automorphism module
- Karatsuba Multiplier for $\mathbb{F}_{2^m}$
- Itoh-Tsujii $\mathbb{F}_{2^m}$ Inversion
- $\mathbb{F}_n$ Montgomery Multiplier based on DSP slices

Computation Time



Extrapolated: 24 days

Challenge Generation

```
import hashlib
PX = str_to_poly(hashlib.sha256(str(0)).hexdigest())
PY=PolynomialRing(K, 'PY').gen()
P_ROOTS = (PY^2+PX*PY+PX^3+a*PX^2+b).roots()
P=E([PX,P_ROOTS[0][0]]); P=P*h

QX = str_to_poly(hashlib.sha256(str(1)).hexdigest())
Q_ROOTS = (PY^2+QX*PY+QX^3+a*QX^2+b).roots()
Q=E([QX,Q_ROOTS[0][0]]); Q=Q*h
```

Different FPGAs

Series	Development Kit	LUTs used	maximum Frequency	Price
Virtex-6	ML605	38%	261 MHz	2,495 USD
Spartan-6	LX150T	-	147 MHz	995 USD
Artix-7	AC701	62%	264 MHz	999 USD
Virtex-7	VC707	28%	313 MHz	3,495 USD
Kintex-7	KC705	42%	313 MHz	1,695 USD

Different Targets

Target	Iterations	Costs [USD]	Days (Estimated)
ECC2K-112	8.5 x 10	42,000	22
ECC2-113	90 x 10	42,000	118
ECC2K-130	4,055 x 10	1,000,000	127
ECC2-131	46,239 x 10	10,000,000	145
ECC2-163	3,030 x 10	1,000,000,000	189,934

Open Issues

- Power problems
  - Maximum frequency: 165 MHz vs 275 MHz
  - Multiple instances
- Negation map and fruitless cycles

Random Facts

- Necessary budget:
  - 18 FPGAs: 2,500 USD x 18 = 45,000 USD
  - Power consumption: different budget :-)
  - 1.5 man-years: 100,000 USD (different budget)
- Money actually spent: 20 EUR on chocolate

Room for improvement

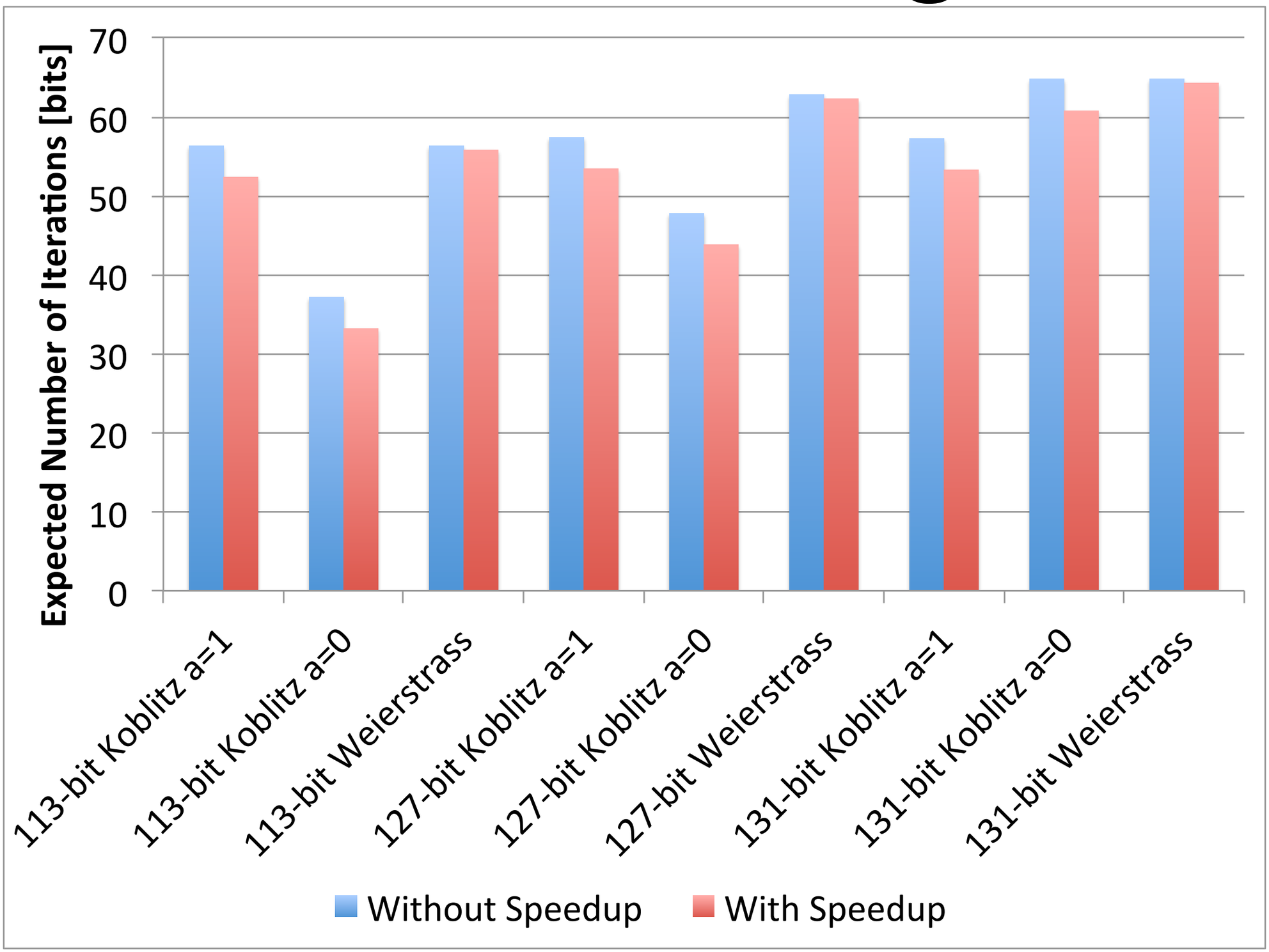
YES!!!

2x speed equals 2 extra bits to attack

128x speed equals 14 extra bits to attack

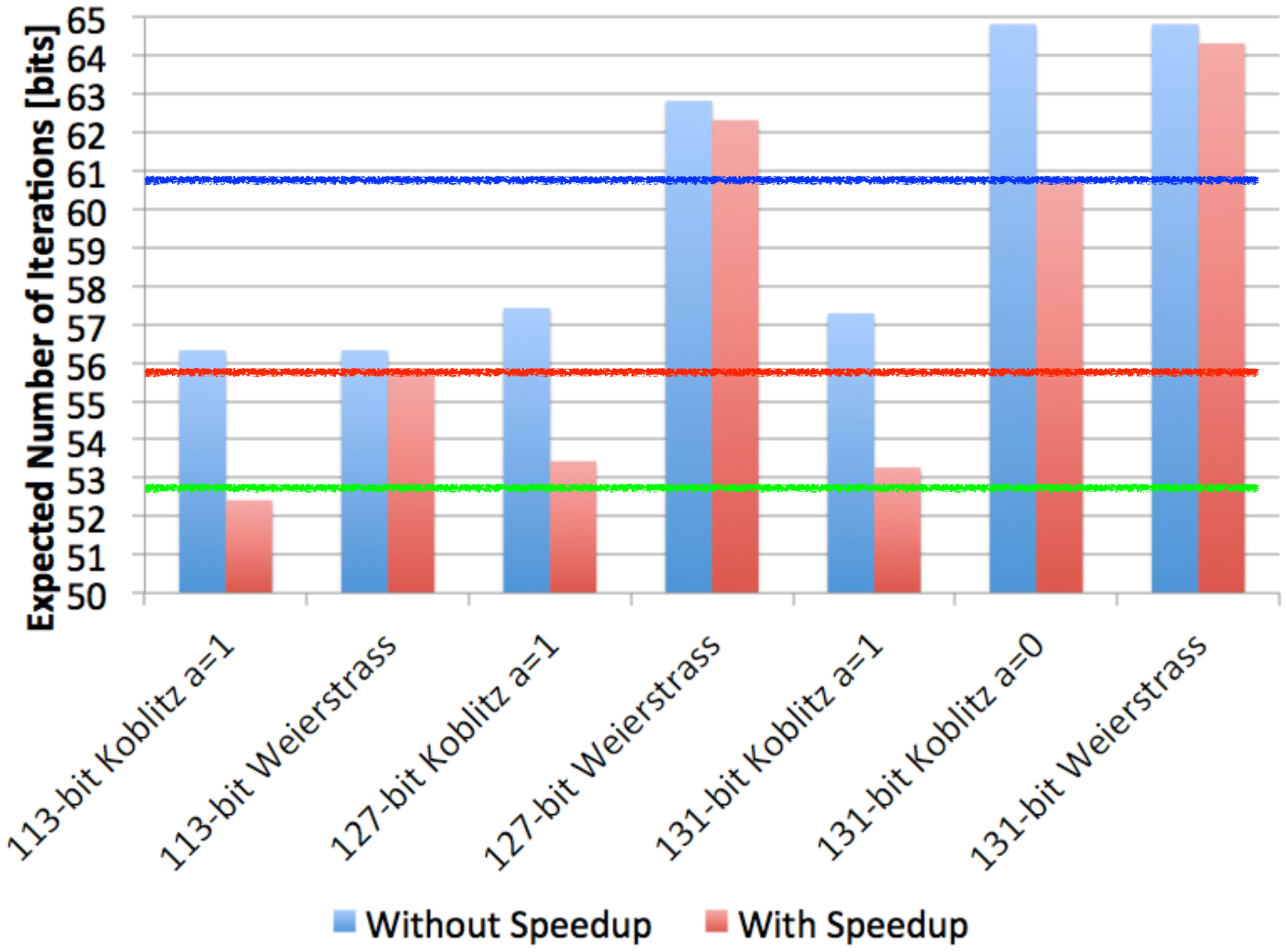
Prime numbers: 109, 113, 127, 131, …

New Challenges



Prime numbers: 109, 113, 127, 131, …

New Challenges



Solving the Discrete Logarithm of a 113-bit Koblitz Curve with an FPGA Cluster

Erich Wenger and Paul Wolfger
Graz University of Technology
WECC 2014, Chennai, India