SAND2015-9124C

Energy-Efficient Implementations of $GF(p)$ and $GF(2^m)$ ECC

Andrew D. Targhetta[1,2], Donald E. Owen Jr.[2],
Francis L. Israel[2], Paul V. Gratz[1]

[1]Texas A&M University
[2]Sandia National Laboratories

October 21st, 2015

ories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of San
well International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under cont

Encryption in Ultra-low Energy Domain

- Security is of critical importance, but the energy per operation is paramount to the device's utility!
- Applications include...
  - Low-Power Sensor Networks
  - Implantable Medical Devices (IMD)
  - Identification tags
  - ...and more

The Problem

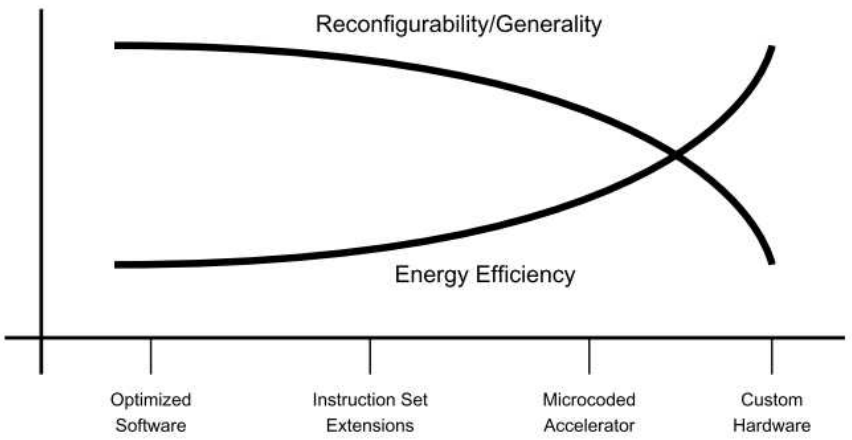
Elliptic Curve Cryptography (ECC)

- Energy-efficient public-key cryptography [Potlapally et al., 2006]
- Necessary for secure communication
- Energy cost is *still* prohibitive for ultra-low energy devices!

The Solution

Hardware acceleration can improve the energy efficiency of elliptic curve cryptography!

- Off-load computation to energy efficient accelerator
- Trade some reconfigurability for increase in efficiency



Contribution

- Development of an improved $GF(2^m)$ coprocessor
- Energy and performance evaluation across a range of ECC key-sizes, including $GF(p)$ 521-bit and $GF(2^m)$ 571-bit
- Evaluation of the energy benefit of an instruction cache for ECC

Table of contents

Background

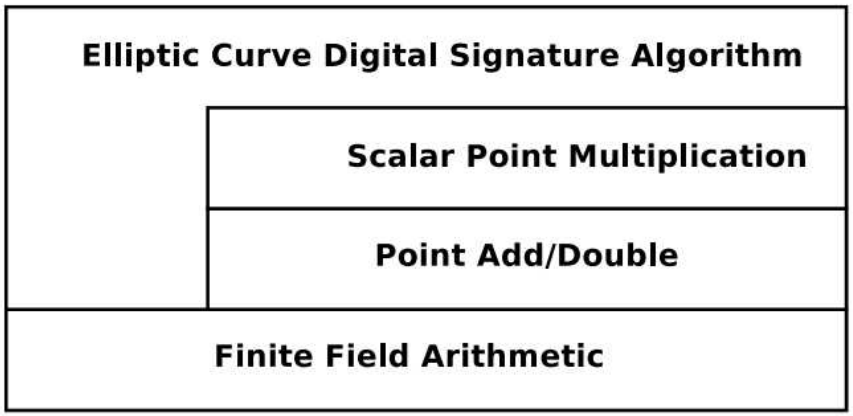
Design

Evaluation

Conclusion

Finite-field Arithmetic

- ECC utilizes both $GF(p)$ and $GF(2^m)$
- Multi-precision computations such that key-size $\gg$ machine width
- Add, subtract, *multiply*, and inversion with reduction



$GF(p)$ and $GF(2^m)$

$GF(p)$, a.k.a. *prime-field* arithmetic

- Uses integer math with *modulo* as the reduction operator
- Example: $(3 + 5)$ *modulo* $7 = 1$

$GF(2^m)$, a.k.a. *binary-field* arithmetic

- Uses polynomial arithmetic s.t. coefficients are *modulo* 2
- $(x^6 + x^4 + x^3 + 1) + (x^5 + x^4 + x^2 + 1)$
  $= x^6 + x^5 + x^3 + x^2$

Binary-fields, $GF(2^m)$

- Attractive for HW because add is simply XOR (carry-less) and requires no reduction
- Squaring algorithm has $O(n)$ complexity as opposed to $O(n^2)$
- Reduced computational complexity has the potential to save energy

Overview of Approach

Explore design space:

- Start with an efficient baseline

Overview of Approach

Explore design space:

- Start with an efficient baseline
- Optimize software for baseline

Overview of Approach

Explore design space:

- Start with an efficient baseline
- Optimize software for baseline
- Add instruction set extensions

Overview of Approach

Explore design space:

- Start with an efficient baseline
- Optimize software for baseline
- Add instruction set extensions
- Add instruction cache

Overview of Approach

Explore design space:

- Start with an efficient baseline
- Optimize software for baseline
- Add instruction set extensions
- Add instruction cache
- Add full coprocessor

Overview of Approach

Explore design space:

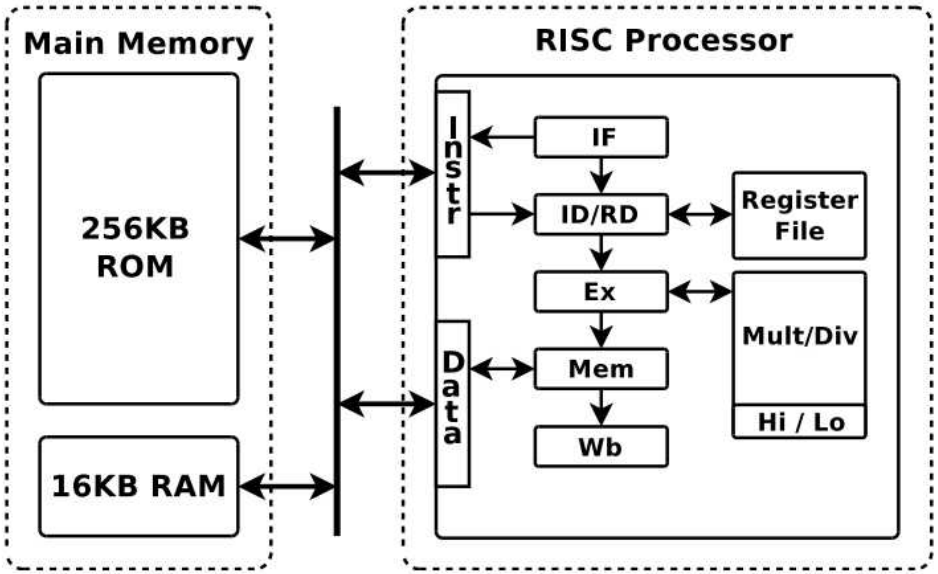
- Start with an efficient baseline
- Optimize software for baseline
- Add instruction set extensions
- Add instruction cache
- Add full coprocessor

Energy efficient baseline

Typical embedded System On a Chip (SoC)...

- 5-stage, RISC pipelined processor
- No MMU or cache
- Multi-cycle multiplication unit
- Minimal memory configuration

"Pete"



Baseline Software

- Operand scanning multi-precision multiplication
- NIST fast reduction techniques
- Sliding window scalar point multiplication
- Three dimensional coordinate systems [Brown et al., 2001]

Instruction Set Extensions (ISE)

- Improve efficiency of product-scanning multiplication [Großschädl and Savaş, 2004]
- Decrease computation time significantly
- While only marginally increasing power

$GF(2^m)$ ISE

- Require minimal modifications to the processor core

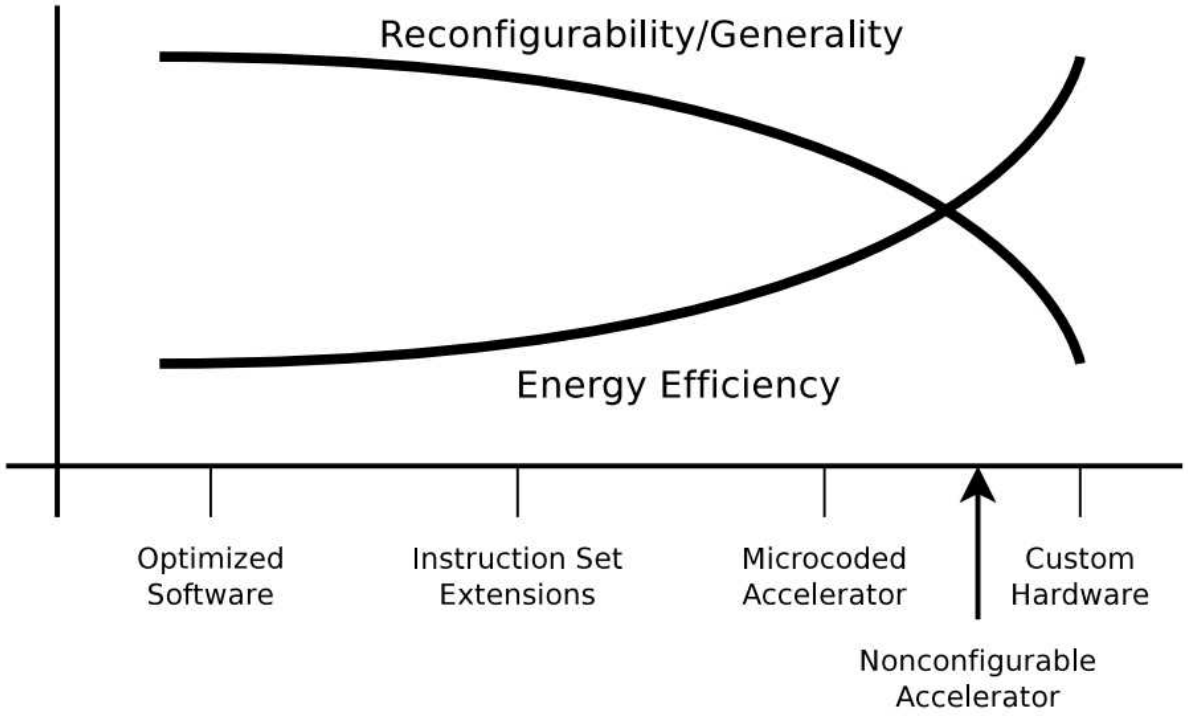
Format	Operation
MULGF2 rs, rt	Carry-less Multiply
MADDUGF2 rs, rt	Carry-less Multiply-Accumulate

Instruction Fetch Energy

Energy breakdown showed fetching instructions from ROM is costly

- RISC processor fetches every clock cycle
- Energy of memory access is related to size of memory
- Program ROM is the largest memory in our system
- Solution: Add an instruction cache to our system!

Moving further towards the right...

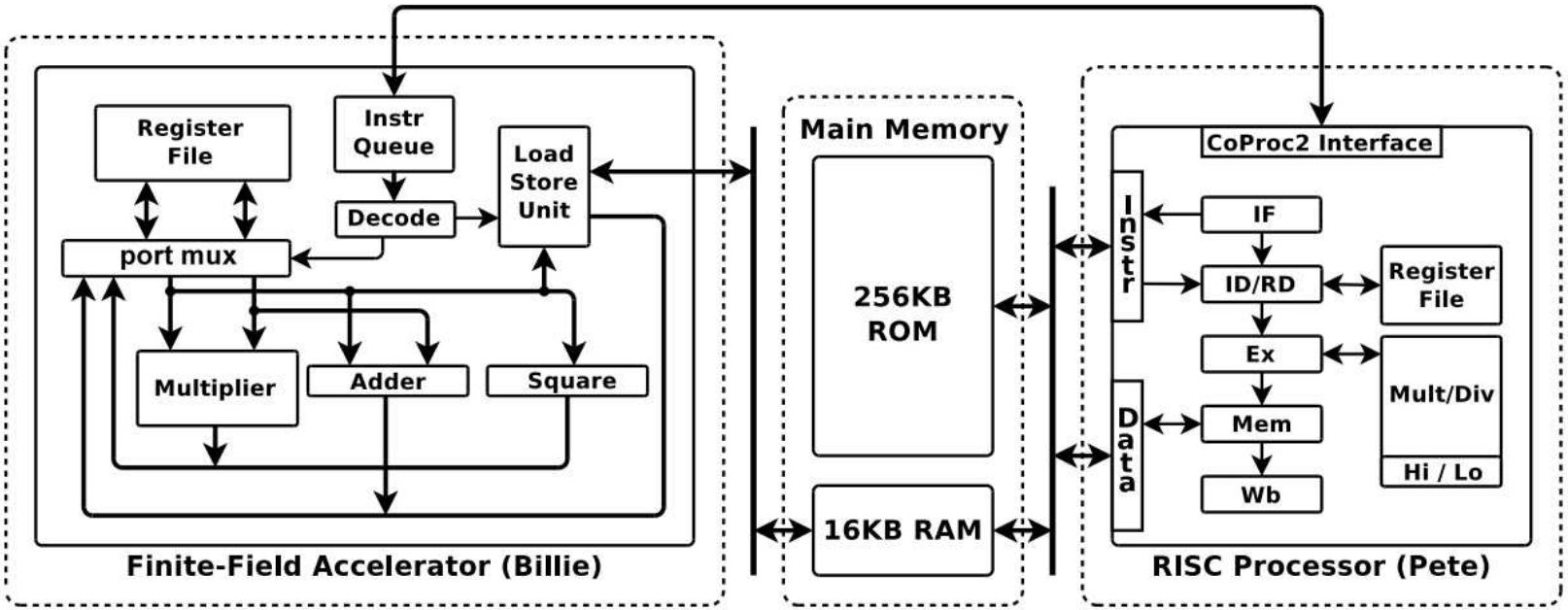


Binary-field Accelerator

Further reduce energy with a binary accelerator:

- Improvement over previously proposed designs [Guo and Schaumont, 2009]
- Non-configurable architecture tuned to field
- Performs carry-less addition, multiplication, and squaring
- Similar approach as the original IBM 360 floating point unit [Anderson et al., 1967]

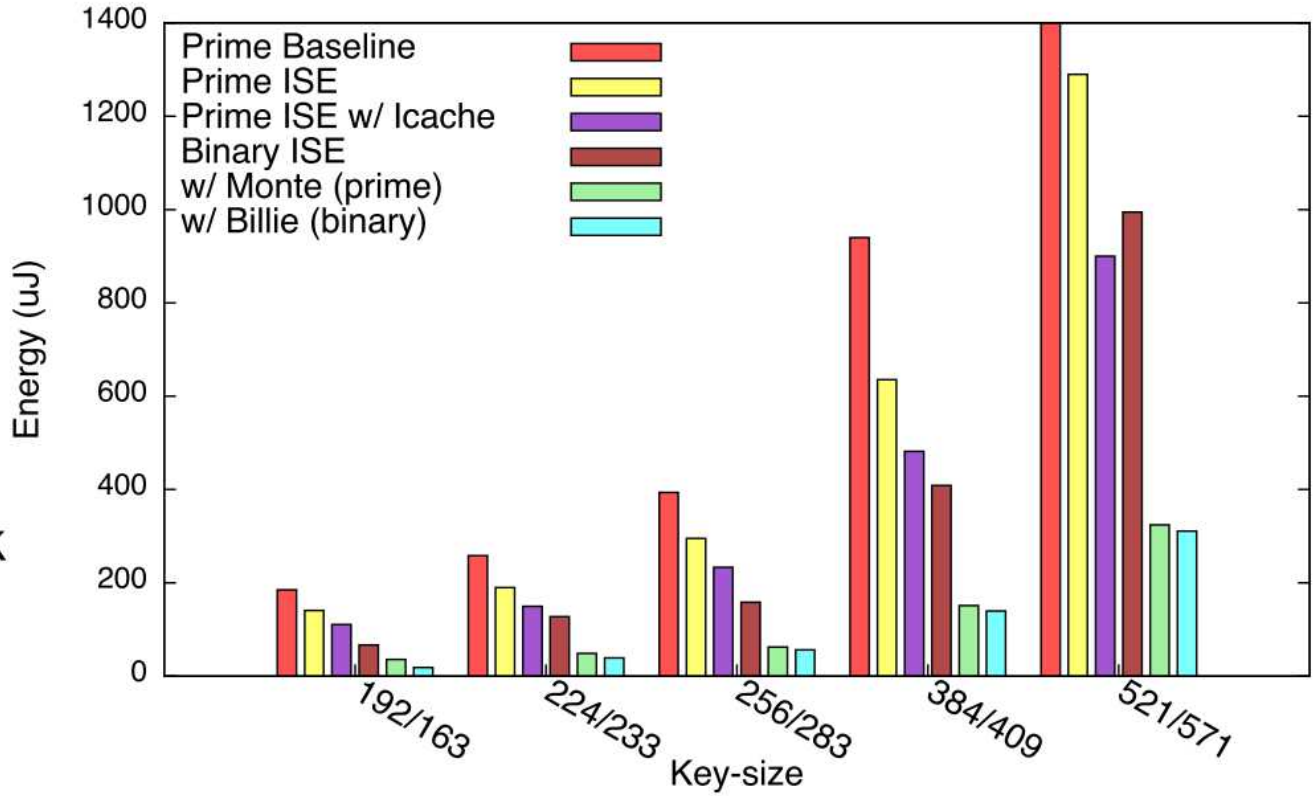
Pete and Billie Overview



- 16 entry register file
- DMA to shared memory
- Multiple functional units, including a digit serial multiplier [Kumar et al., 2006]

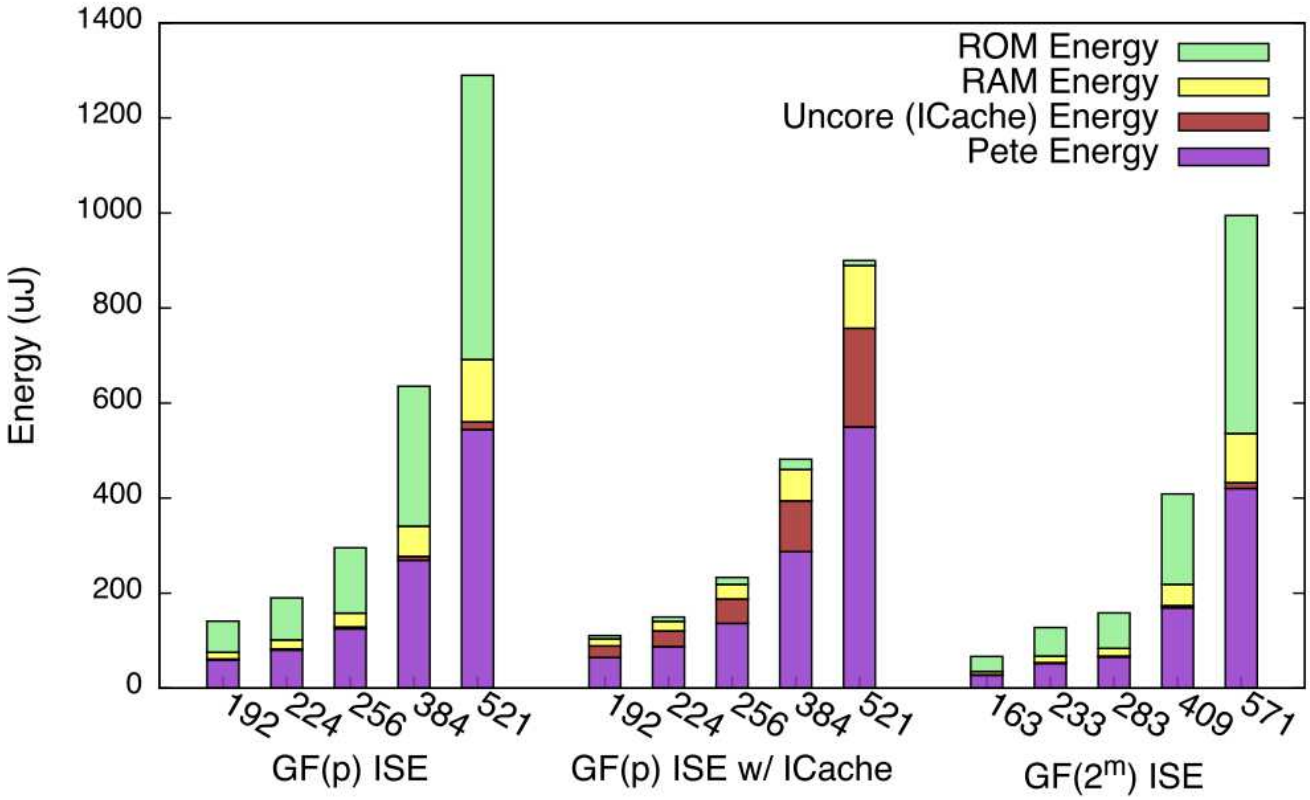
Energy per Operation vs. Key Size

- 6 diff. HW/SW configurations
- 5 equiv. security groups
- Monte is the GF(p) accelerator from our prior work



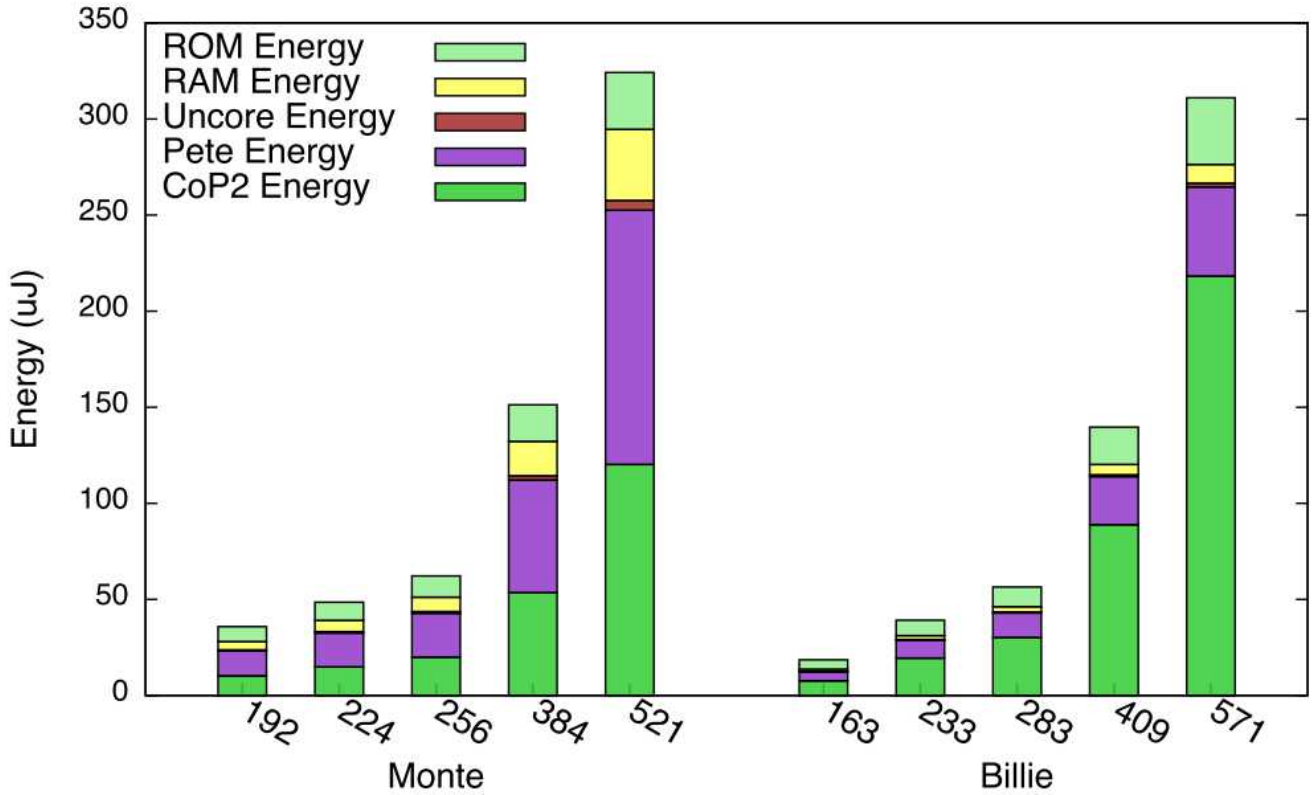
Energy Breakdown for ISE Configurations

- ICache trades ROM energy for less ICache energy
- Binary-field computation is less complex with fewer memory accesses



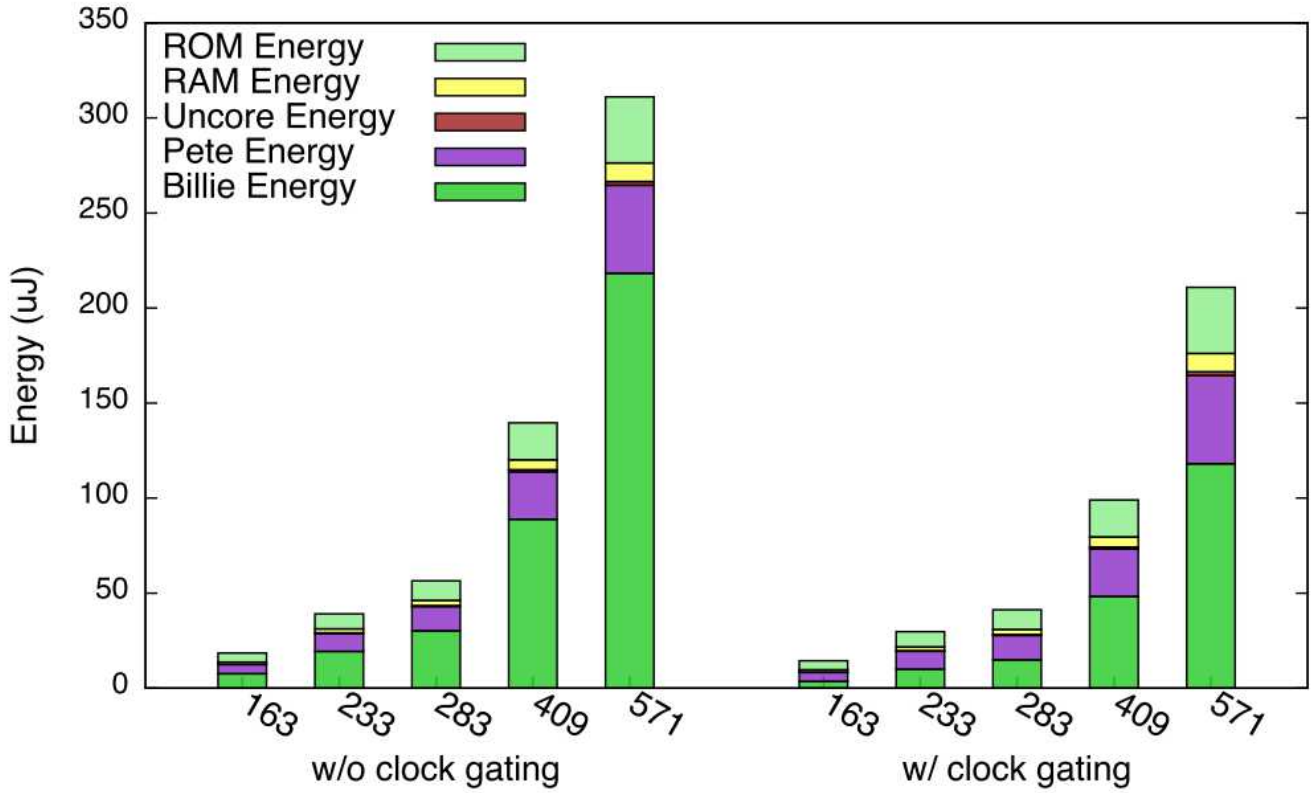
Energy Breakdown for Monte vs. Billie

- Billie's size scales with field size
- RAM energy is reduced with Billie
- Amdahl's Law strikes again (inversion)



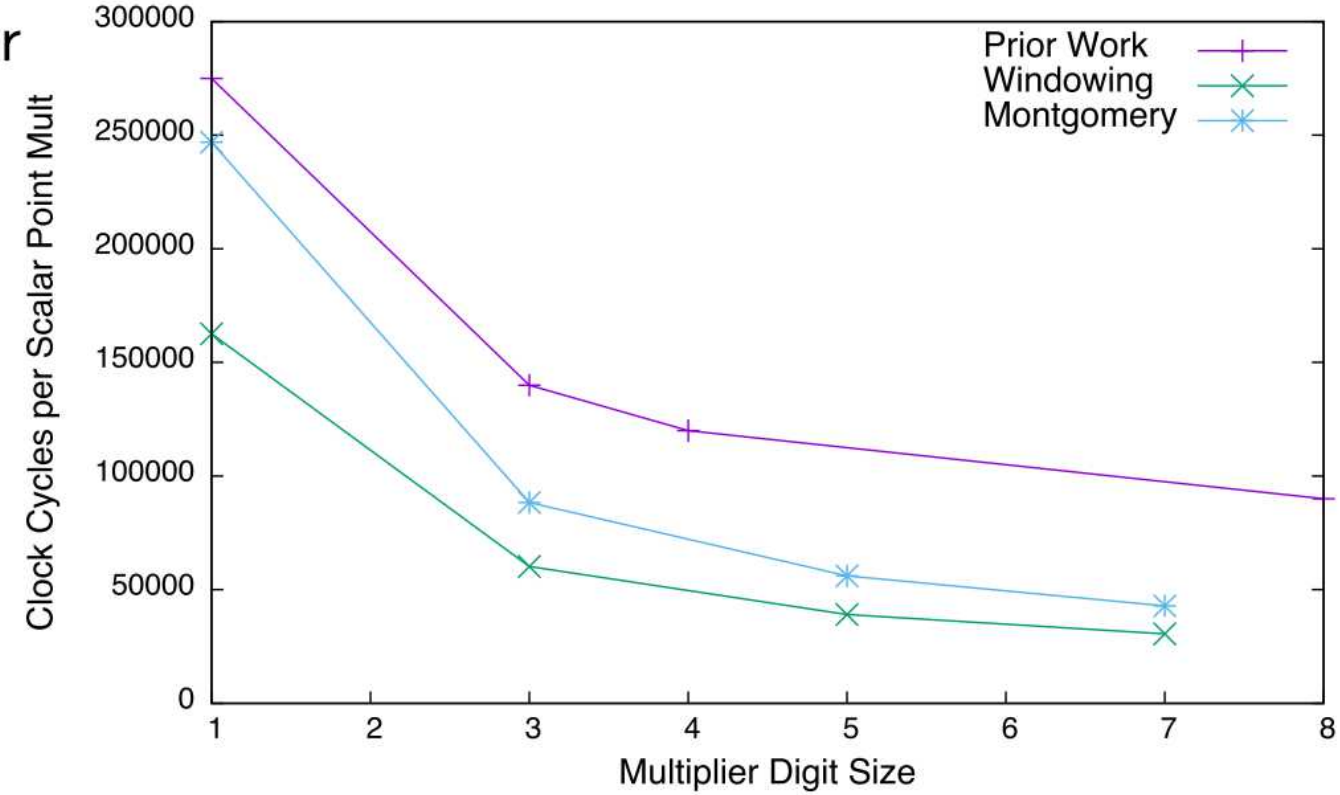
Energy Improvements with Clock Gating

- Assume no dynamic power when Billie is idle
- Provides 22% to 32.2% reduction in energy consumption



Latency Comparison

- For a 163-bit scalar multiply
- Improvement due to efficient coprocessor interface (Montgomery)
- Improvement due to windowing algorithm



Prior work: [Guo and Schaumont, 2009]

Background

Design

Evaluation

Conclusion

Conclusion

- Public-key cryptography is necessary but very costly in terms of energy
- ISA extensions with Icache — up to 2.08x improvement over baseline

Conclusion

- Public-key cryptography is necessary but very costly in terms of energy
- ISA extensions with Icache — up to 2.08x improvement over baseline
- Prime-field coprocessor — up to 6.34x improvement over baseline

Conclusion

- Public-key cryptography is necessary but very costly in terms of energy
- ISA extensions with Icache — up to 2.08x improvement over baseline
- Prime-field coprocessor — up to 6.34x improvement over baseline
- Binary-field ISA extensions — up to 2.11x improvement over prime-field ISA ext.

Conclusion

- Public-key cryptography is necessary but very costly in terms of energy
- ISA extensions with Icache — up to 2.08x improvement over baseline
- Prime-field coprocessor — up to 6.34x improvement over baseline
- Binary-field ISA extensions — up to 2.11x improvement over prime-field ISA ext.
- Binary-field coprocessor — 1.94x improvement over Monte for 163/192-bit

Conclusion

- Public-key cryptography is necessary but very costly in terms of energy
- ISA extensions with Icache — up to 2.08x improvement over baseline
- Prime-field coprocessor — up to 6.34x improvement over baseline
- Binary-field ISA extensions — up to 2.11x improvement over prime-field ISA ext.
- Binary-field coprocessor — 1.94x improvement over Monte for 163/192-bit

Questions???

Backup Slides


Anderson, S., Earle, J., Goldschmidt, R., and Powers, D. (1967).
The ibm system/360 model 91: floating-point execution unit.
IBM Journal, 11(1):34–53.


Brown, M., Hankerson, D., López, J., and Menezes, A. (2001).
Software implementation of the NIST elliptic curves over prime fields.
Springer, New York, NY.


Großschädl, J. and Savaş, E. (2004).
Instruction set extensions for fast arithmetic in finite fields gf(p) and gf(2^m).
In *Cryptographic Hardware and Embedded Systems-CHES 2004*, pages 133–147. Springer.


Guo, X. and Schaumont, P. (2009).
Optimizing the hw/sw boundary of an ecc soc design using control hierarchy and distributed storage.
In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 454–459. European Design and Automation Association.


Hankerson, D., Menezes, A. J., and Vanstone, S. (2004).
Guide to elliptic curve cryptography.
Springer, New York, NY.


Kumar, S., Wollinger, T., and Paar, C. (2006).
Optimum digit serial gf(2^m) multipliers for curve-based cryptography.
Computers, IEEE Transactions on, 55(10):1306–1311.


Montgomery, P. L. (1985).
Modular multiplication without trial division.
Mathematics of computation, 44(170):519–521.


Muralimanohar, N., Balasubramonian, R., and Jouppi, N. P. (2009).

Cacti 6.0: A tool to model large caches.
HP Laboratories.

Potlapally, N. R., Ravi, S., Raghunathan, A., and Jha, N. K. (2006).
A study of the energy consumption characteristics of cryptographic algorithms and security protocols.
Mobile Computing, IEEE Transactions on, 5(2):128–143.

Targhetta, A. D. and Gratz, P. V. (2011).
An Energy Efficient Datapath for Asymmetric Cryptography.
In *The 3rd Workshop on Energy Efficient Design (WEED)*.

Wander, A. S., Gura, N., Eberle, H., Gupta, V., and Shantz, S. C. (2005).
Energy analysis of public-key cryptography for wireless sensor networks.
In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 324–328. IEEE.

Yip, G. (2006).
Expanding the synopsys primetime® solution with power analysis.
http://www.synopsys.com/Tools/Implementation/SignOff/CapsuleModule/ptpx-wp.pdf.

Future Work

- Evaluate ICache w/ binary-field ISA ext.
- Continued work on Billie
  - Model large register file in SPICE
  - Accelerate inversion
  - Fixed sized accelerator
- Investigate Koblitz Curves
- Investigate post-quantum algorithms

Why should *ECC* be used over *RSA*?

Due to sub-exponential attacks on RSA, ECC requires smaller keys for equivalent security...

Key Length (Bits)[Hankerson et al., 2004]					
RSA	1024	2048	3072	8192	15360
ECC	160	224	256	384	512

Suggested ISA extensions

ISA Extensions for $GF(p)$ [Großschädl and Savaş, 2004].

Format	Operation
MADDU rs, rt	Multiply and Accumulate Unsigned
M2ADDU rs, rt	Multiply, Double, and Accumulate Unsigned
ADDAU rs, rt	Add to Accumulator Unsigned
SHA	Shift Accumulator to the right by 32 bits

$GF(2^m)$ Math

$GF(2^7)$ multiplication assuming $f(x) = x^7 + x + 1$:

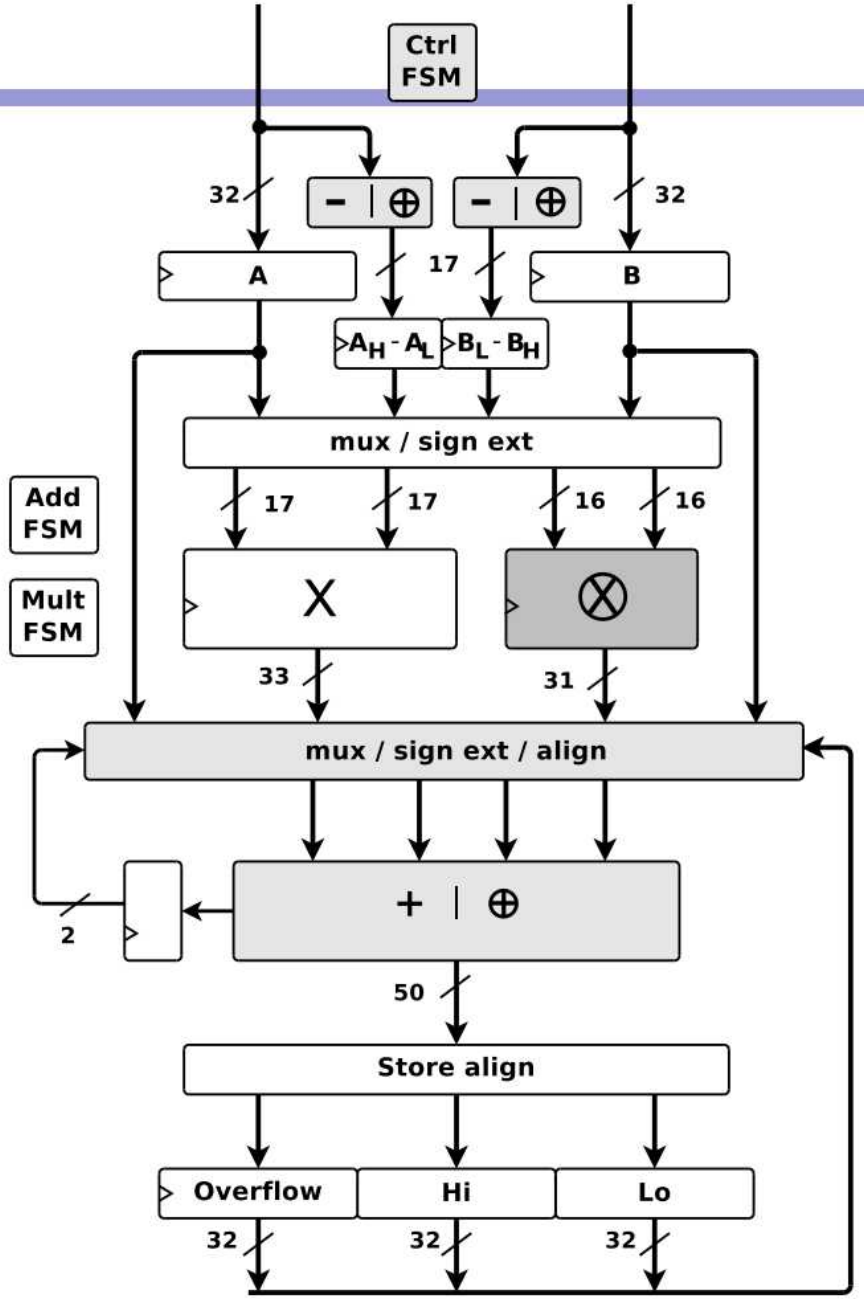
- $(x^6 + x^3 + x) \times (x^6 + x^2 + 1) = x^3 + x + 1$
- *Multiplication:*
  $a(x) \times b(x) = x^{12} + x^9 + x^8 + x^7 + x^6 + x^5 + x$
- *Reduction:*
  modulo $f(x) = x^3 + x + 1$
- Addition and subtraction are the same operation and do not require reduction

$GF(2^m)$ Computation

- Attractive for HW because add is simply XOR (carry-less)
- Denoted in the following way: $a(x) = a_{m-1}x^{m-1} + \cdots + a_2x^2 + a_1x + a_0$ where x is the indeterminate of the polynomial, and the coefficients, $a_{m-1}, \cdots, a_2, a_1, a_0 \in [0, 1]$.

Binary-field Support

- Changes in light gray and additions in dark gray
- Increases static power by 2.65%
- Decreases overall power by 2.56%

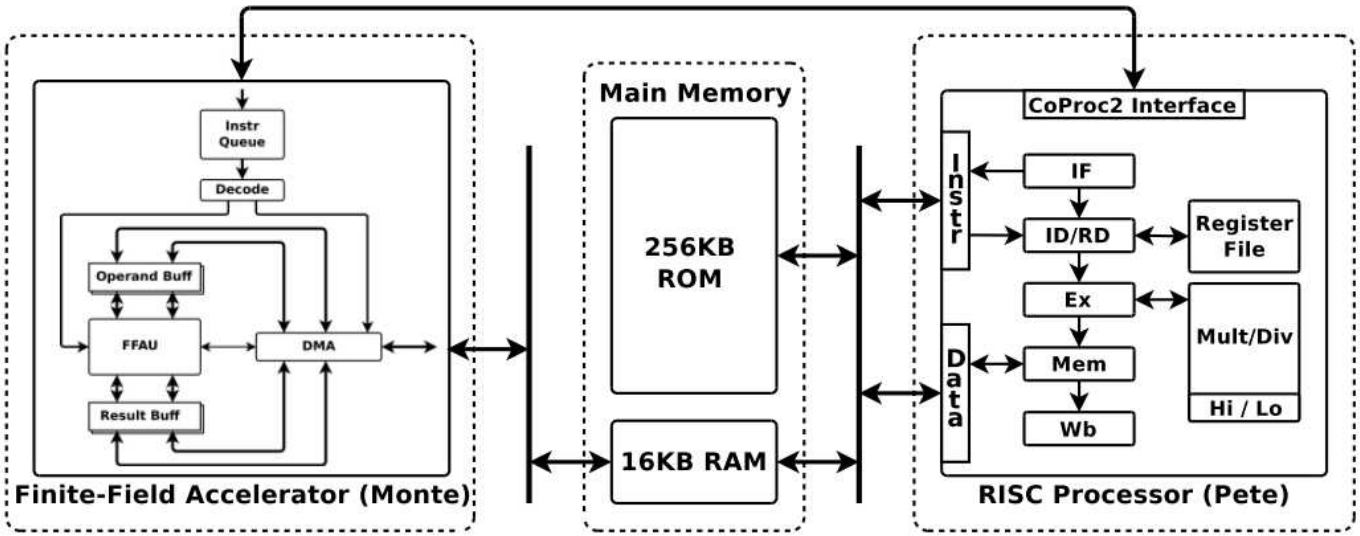


Further improve efficiency...

Add a dedicated $GF(p)$ coprocessor for modular arithmetic...

- Performs prime-field math much more efficiently [Targhetta and Gratz, 2011]
- Reduces instruction fetching with small microcode ROM
- Utilizes coprocessor interface for command and control
- Shares RAM with "Pete"

Pete and Monte overview



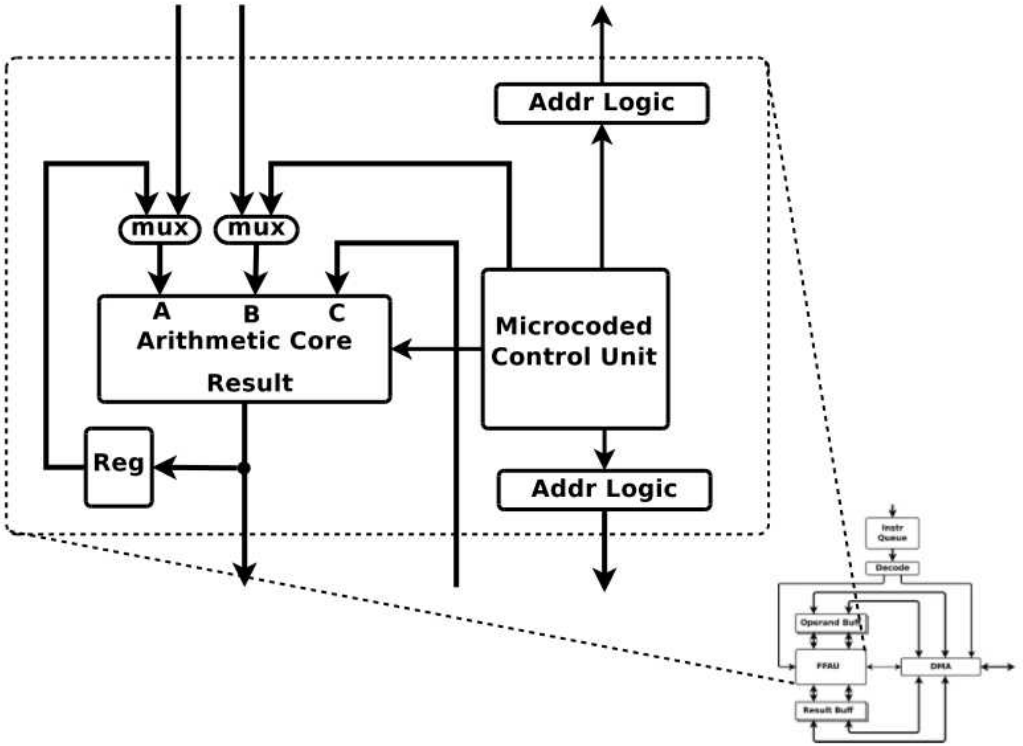
- Double buffered scratch pad memory
- Direct Memory Access (DMA) to shared memory
- Instruction queue (out-of-order processing)

Monte Instructions

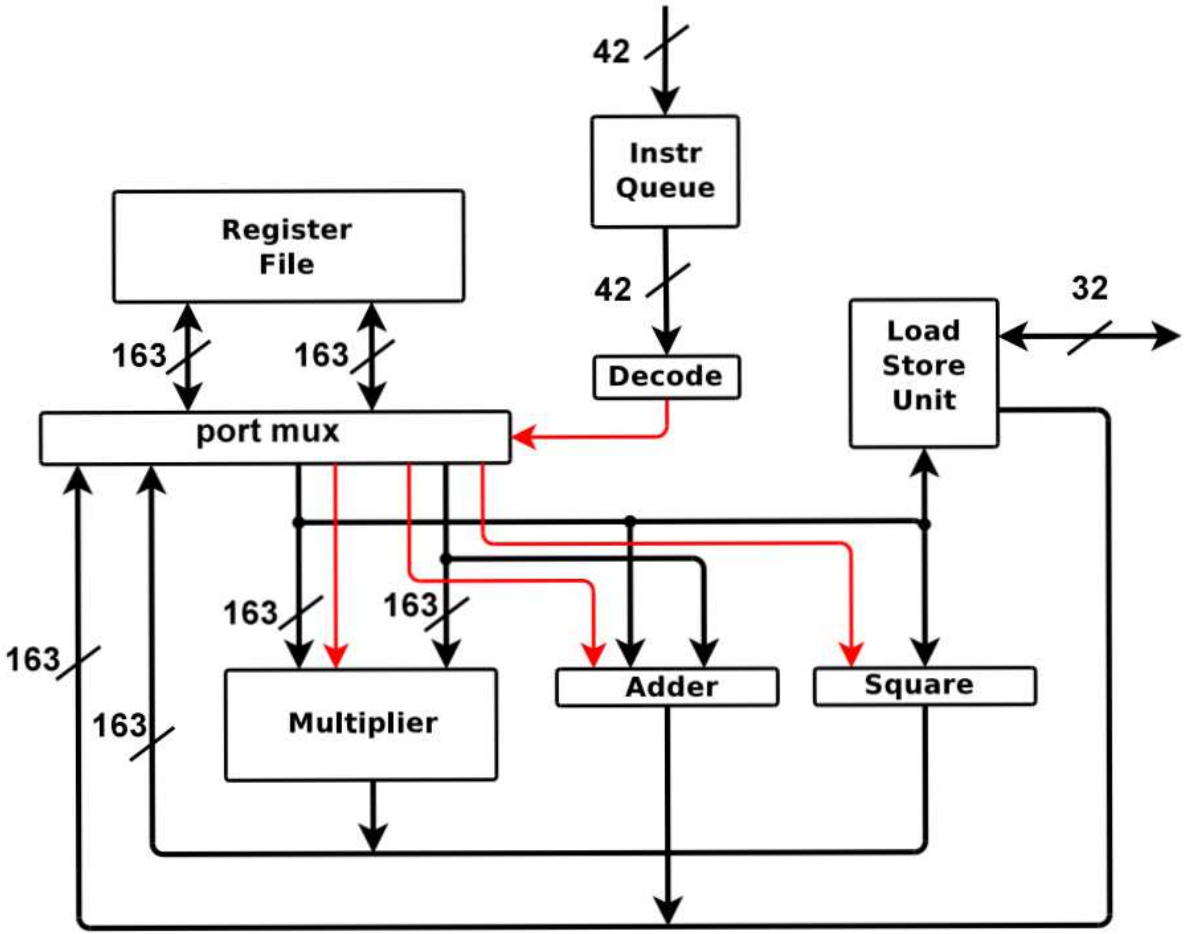
- Fetched and decoded by Pete, then forwarded to Monte (coprocessor interface)
- Allow reconfiguration of field width
- Include mod ADD, SUB, MULT
- Handle transfer of data to/from shared memory

FFAU overview

- Microcoded control unit
- Pipelined arithmetic core
- Computes modular add, subtract, and multiply
- Utilizes Montgomery multiplication [Montgomery, 1985]



Billie Microarchitecture

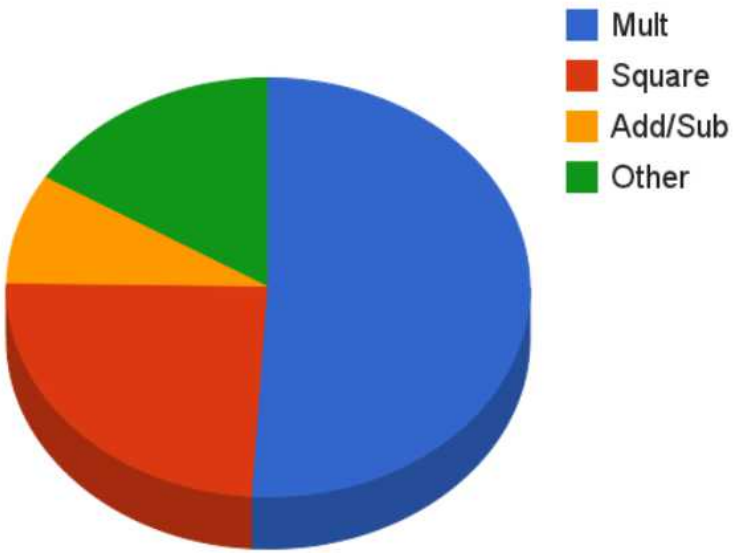


Methodology

- Estimate energy in logic:
  - HDL models of Pete, Monte and Billie synthesized to 45nm
  - Synopsys Prime-Time simulated energy consumption while performing ECC [Yip, 2006]
- Estimate energy in memory:
  - Test bench counts reads and writes to memories
  - Cacti estimates energy per read/write and static energy [Muralimanohar et al., 2009]

Motivation

- 75% multiply/square
- 9% add/sub
- ∼16% other



Portion of time spent performing modular math for P384 ECDSA.

Energy in Sensor Network Domain

For example in sensor network domain:

- Consumes approx. 72% of the energy allotted for communication handshaking [Wander et al., 2005]
- Only 5% to 10% of energy budget is available for handshakes!