A high speed coprocessor for elliptic curve scalar multiplication over $\mathbb{F}_p$

Nicolas Guillermin

DGA/IRMAR

CHES - august 18^{th} 2010

Plan

An alternative for multiprecision arithmetic

Let $M = \prod_{i=1}^{n} m_i$ a set of coprime

RNS form of $X < M$ is $\{x_1, ..., x_n\}$ such that

$$x_i = |X|_{m_i} \qquad (1)$$

An alternative for multiprecision arithmetic

Let $M = \prod_{i=1}^{n} m_i$ a set of coprime

RNS form of $X < M$ is $\{x_1, ..., x_n\}$ such that

$$x_i = |X|_{m_i} \tag{1}$$

$$X = \left| \sum_{i=0}^{n-1} (|x_i.M_i^{-1}|_{m_i}) \times M_i \right|_M \tag{2}$$

$$M_i = \frac{M}{m_i} \tag{3}$$

RNS arithmetic

Easy and fast:

- addition $RNS(a+b) = \{a_i + b_i\}$
- subtraction $RNS(a-b) = \{a_i - b_i\}$
- product $RNS(a \times b) = \{a_i \times b_i\}$
- division $RNS(\frac{a}{b}) = \{\frac{a_i}{b_i}\}$ if $gcd(b, M) = 1$

No carry propagation, no quadratic algorithm but over $\frac{\mathbb{Z}}{M\mathbb{Z}}$

Montgomery RNS Algorithm [1]

Algorithm 1 $Red_{Montg}(X, P, M, \tilde{M})$

Require: $A < 4P^2$ in M and $\tilde{A}$ in $\tilde{M}$, P prime, M and $\tilde{M}$ up to respectively $4P$ and $2P$

Ensure: $S \equiv A \times M^{-1}[P]$, $A < 2P$ in M and $\tilde{M}$

$Q \leftarrow X \times P^{-1}$ in M

$\tilde{Q} \leftarrow B_{ext}(Q, M, \tilde{M})$

$\tilde{R} \leftarrow \tilde{A} + \tilde{Q} \times \tilde{P}$ in $\tilde{M}$

$\tilde{A}' \leftarrow \tilde{R} \times M^{-1}$ in $\tilde{M}$

$A' \leftarrow B_{ext}(\tilde{A}', \tilde{M}, M)$

return A' in M and $\tilde{A}'$ in $\tilde{M}$

Algorithm $B_{ext}(X, M, \tilde{M})$

Among all algorithm given by public literature, the best choice is the Kawamura et al. improvement of Posch et al. algorithm [2]:

- n^2 complexity
- easy to parallelize
- massive use of multipliers
- fits perfectly in an FPGA

Algorithm $B_{ext}(X, M, \tilde{M})$

Among all algorithm given by public literature, the best choice is the Kawamura et al. improvement of Posch et al. algorithm [2]:

- n^2 complexity
- easy to parallelize
- massive use of multipliers
- fits perfectly in an FPGA

Kawamura's architecture "Cox-Rower" gives a good basis to realize an elliptic curve coprocessor.

RNS and ECC related work

An overview of the relations between RNS and elliptic curves is given in Duquesne et al. [3]

RNS and ECC related work

An overview of the relations between RNS and elliptic curves is given in Duquesne et al. [3]

For Weierstrass curves, the best choice is:

- Montgomery ladder
- projective coordinates
- Kummer surface

Advantage for security:

- balanced algorithm with no dummy operation
- easy adaptation of the classical countermeasures against SCA.

Addition and doubling formulæ

Step	Computation	reduction
prec. $P+Q$	$A = Z_P X_Q + X_P Z_Q$, $B = 2X_P X_Q$ $C = 2Z_P Z_Q$, $D = a_4 A + a6C$	2 2
Z_{P+Q}	$A^2 - BC$	1
X_{P+Q}	$BA + CD - xZ_{P+Q}$	1
total		6

Step	Computation	reduction
prec. $2P$	$A' = 2X_P Z_P$, $B' = X_P^2$, $C' = Z_P^2$ $D' = -4bA'$, $E' = aA$	3 2
X_{2P}	$A'D' + (B' - E')^2$	1
Z_{2P}	$2A'(B' + E') - D'C'$	1
total		7

Arithetic over $\mathbb{F}_p$: RNS vs Multiprecision

For multiprecision

- Lower complexity of multiplication and reduction ($2n^2 + n$ vs $2n^2 + 4n$)
- No modular reduction by m_i
- relatively small p for ECC than for RSA

Arithetic over $\mathbb{F}_p$: RNS vs Multiprecision

For multiprecision

- Lower complexity of multiplication and reduction ($2n^2 + n$ vs $2n^2 + 4n$)
- No modular reduction by m_i
- relatively small p for ECC than for RSA

For RNS

- Easy to parallelize algorithm
- No carry propagation
- Lazy reduction of $AB + CD$ pattern
- Good trade off between speed and security

Plan

1. Residue Number System and ECC
2. Hardware architecture
3. Results and conclusion

Target FPGA

Altera Stratix family

- node : 130 nm
- Logic Element (LE) : $4 \Rightarrow 1$
- 9×9, 18×18 and 36×36 multiplier blocks

Target FPGA

Altera Stratix family

- node : 130 nm
- Logic Element (LE) : $4 \Rightarrow 1$
- 9×9, 18×18 and 36×36 multiplier blocks

Altera Stratix II family

- node : 90 nm
- Altera Logic Module (ALM) : $2 \times 4 \Rightarrow 1$ to $6 \Rightarrow 1$
- 9×9, 18×18 and 36×36 DSP blocks

Base choice

Use of pseudo-Mersenne numbers

$$m_i = 2^r - \epsilon_i \tag{4}$$

Base choice

Use of pseudo-Mersenne numbers

$$m_i = 2^r - \epsilon_i \tag{4}$$

Use of Kawamura's architecture with n Rower.

$$n \times r > log_2(p) \tag{5}$$

Base choice

Use of pseudo-Mersenne numbers

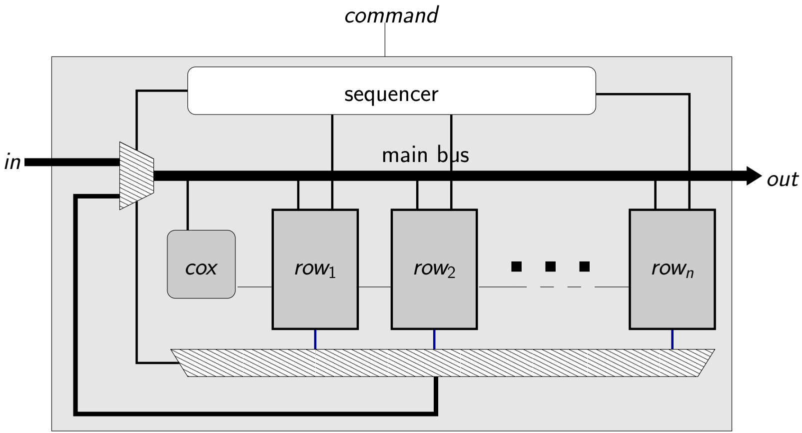
$$m_i = 2^r - \epsilon_i \tag{4}$$

Use of Kawamura's architecture with n Rower.

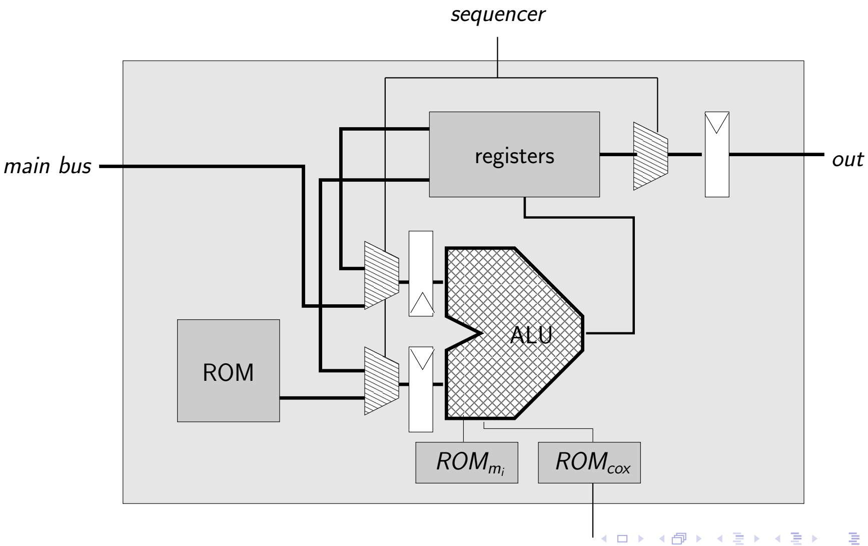
$$n \times r > log_2(p) \tag{5}$$

A whole multiplication/accumulation can be done at each cycle

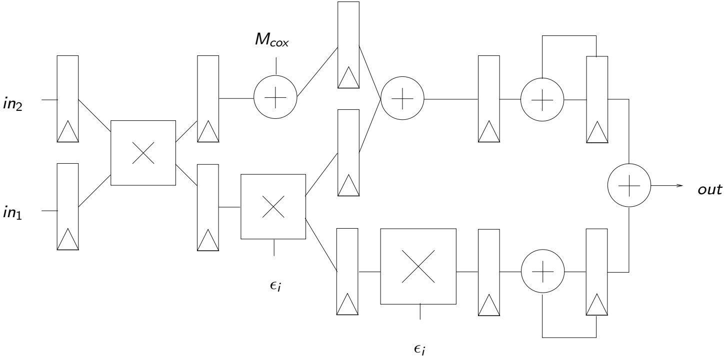
Overview



Multiplication Rower



6 stage pipeline



For 160 bits curves on Stratix, 91MHz.

How to avoid idle state?

By using inherent pallelism of the formulæ

Step 1	Step 2	Step 3	Step 4	Step 5
A B C	D Z_{P+Q}	X_{P+Q} A' B' C'	D' E'	X_{2P} Z_{2P}
3	2	4	2	2

For the 6 stage pipeline and 160 bits curve,
less than 10% of idle state

Other operations

Modular inversion

Multiprecision $\longrightarrow$ RNS

RNS $\longrightarrow$ Multiprecision

Other operations

Modular inversion

- Use a $p-2$ exponentiation
- Less than 10% of the total time and no gate required.

Multiprecision $\longrightarrow$ RNS

- Use of the classical MSW to LSW approach
- No additional gate and almost no time.

RNS $\longrightarrow$ Multiprecision

- Use of $m_0 = 2^r$ and $\tilde{M}$ as a base
- No additional gate and less than 3% of the time

Plan

Results

Family	curve	model	n	r	size	DSP	frequency	speed
Stratix	160	EP1S20F484C5	5	34	11431 LE	74	92.6	0.57 ms
	192	EP1S30F780C5	6	33	12480 LE	80	89.6	0.72 ms
	256	EP1S60F780C5	8	33	16200 LE	125	90.7	1.17 ms
	384	EP1S80F1020C5	11	36	25279 LE	176	90.0	2.25 ms
	512	EP1S80F1020C5	15	35	48305 LE	176	79.6	4.03 ms
Stratix II	160	EP2S30F484C3	5	34	5896 ALM	74	165.5	0.32 ms
	192	EP2S30F484C3	6	33	6203 ALM	92	160.5	0.44 ms
	256	EP2S30F484C3	8	33	9177 ALM	96	157.2	0.68 ms
	384	EP2S60F484C3	11	36	12958 ALM	177	150.9	1.35 ms
	512	EP2S60F484C3	15	35	17017 ALM	244	144.97	2.23 ms

comparison

paper	curve	FPGA family	FPGA model	size	freq.(MHz)	speed
This work	160 any	Stratix	EP1S20F484C5	11431 LE	92.6	0.57 ms
	256 any	Stratix	EP1S60F780C5	16200 LE	90.7	1.17 ms
	160 any	Stratix II	EP2S30F484C3	6203 ALM	165.5	0.32 ms
	256 any	Stratix II	EP2S30F484C3	9177 ALM	157.2	0.68 ms
Schinianakis	160 any	Virtex	XCV1000E-8	21000 LUT	58	1.77 ms
	256 any	Virtex	XCV1000E-8	36000 LUT	39.7	3.95 ms
Mentens	160 any	Virtex II-pro	XC2VP30	2171 sl.	72	1 ms
	256 any	Virtex II-pro	XC2VP30	3529 sl.	67	2.27 ms
Guneysu	224 NIST	Virtex 4	XC4VFX12	1580 sl.	487	0.36 ms
	256 NIST	Virtex 4	XC4VFX12	1715 sl.	490	0.49 ms

As a conclusion

RNS is a competitive alternative for high speed implementation of elliptic curves:

- fast and secure
- easy to scale
- easy to integrate
- easy to migrate to other technologies

References

Jean-Claude Bajard, Laurent-Stéphane Didier, and Peter Kornerup.
An rns montgomery modular multiplication algorithm.
IEEE Transactions on Computers, 47(7):766–776, 1998.

Shinichi Kawamura, Masanobu Koike, Fumihiko Sano, and Atsushi Shimbo.
Cox-rower architecture for fast parallel montgomery multiplication.
In *Advances in Cryptology EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 523–538. Springer Berlin / Heidelberg, 2000.

M. Ecegovac S. Duquesne, J.C. bajard.
Combining leak-resistant arithmetic for elliptic curves define over $\mathbb{F}_p$ and rns representation.