SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

On the Implementation of Unified Arithmetic on Binary Huff Curves

Presented by: Benedikt Gierlichs

Santosh Ghosh, Amit Kumar,
Amitabh Das, and Ingrid Verbauwhede

Contents

Introduction

- Elliptic Curve Against Side-channel Attacks
  - Double-and-add always, Montgomery Ladder
  - Atomic Operations
  - Unified Formula
    - Edwards Curve
    - **Huff Curve**

    } – Complete addition formula

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

Unified Binary Huff Curve (UBHC)

$$E_{/F_{2^m}} : aX(Y^2 + fYZ + Z^2) = bY(X^2 + fXZ + Z^2)$$
$$where\ a, b, f \in \mathbb{F}^*_{2^m}\ and\ a \neq b.$$

Let, $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$ then $P+Q$:

$$\begin{cases} X_3 = (Z_1Z_2 + Y_1Y_2)\left((X_1Z_2 + X_2Z_1)(Z_1^2Z_2^2 + X_1X_2Y_1Y_2) + \alpha X_1X_2Z_1Z_2(Z_1Z_2 + Y_1Y_2)\right) \\ Y_3 = (Z_1Z_2 + X_1X_2)\left((Y_1Z_2 + Y_2Z_1)(Z_1^2Z_2^2 + X_1X_2Y_1Y_2) + \beta Y_1Y_2Z_1Z_2(Z_1Z_2 + X_1X_2)\right) \\ Z_3 = (Z_1Z_2 + X_1X_2)(Z_1Z_2 + Y_1Y_2)(Z_1^2Z_2^2 + X_1X_2Y_1Y_2), \end{cases}$$

where $\alpha = \frac{a+b}{b}$ and $\beta = \frac{a+b}{a}$.

Same formula is used to compute both P+Q and 2P.
UNIFIED!

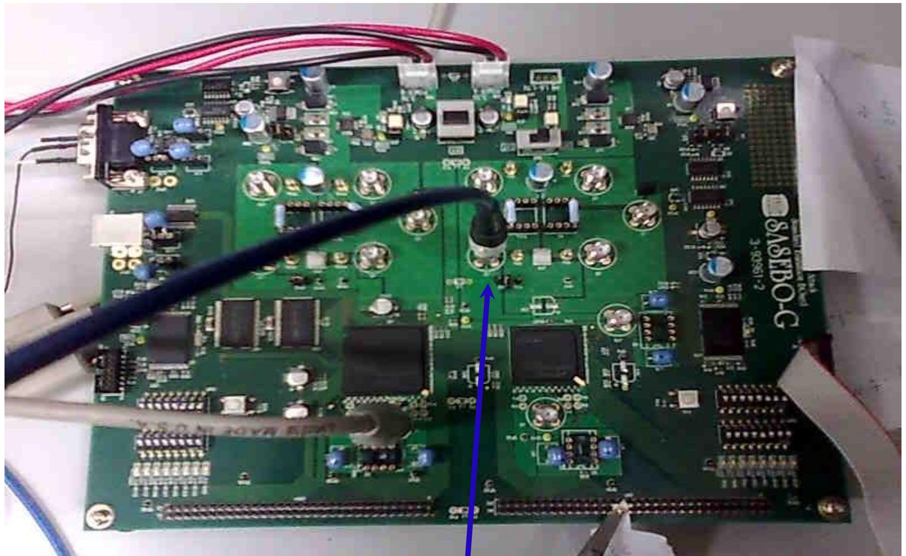
This is computed as:

$$\begin{aligned} &m_1 = X_1X_2, \quad m_2 = Y_1Y_2, \quad m_3 = Z_1Z_2, \\ &m_4 = (X_1 + Z_1)(X_2 + Z_2) + m_1 + m_3, \quad m_5 = (Y_1 + Z_1)(Y_2 + Z_2) + m_2 + m_3, \\ &m_6 = m_1m_3, \quad m_7 = m_2m_3, \quad m_8 = m_1m_2 + m_3^2, \quad m_9 = m_6(m_2 + m_3)^2, \\ &m_{10} = m_7(m_1 + m_3)^2, \quad m_{11} = m_8(m_2 + m_3), \quad m_{12} = m_8(m_1 + m_3), \\ &X_3 = m_4m_{11} + \alpha m_9, \quad Y_3 = m_5m_{12} + \beta m_{10}, \quad Z_3 = m_{11}(m_1 + m_3). \end{aligned}$$

- J. Devigne and M. Joye, “Binary huff curves,” CT-RSA 2011, LNCS 6558, pp. 340–355, Springer-Verlag, 2011.

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

FPGA Implementation of UBHC



- Side-channel Attack Standard Evaluation Board (SASEBO-G)
- Implemented on the xc2vp30-fg676-5 device
- The datapath consists of an n-bit Hybrid Karatsuba multiplier, some binary field adders and squaring circuits.
- There are 17 F_{2^n} multiplications in unified addition formula.

Measure power consumption during UBHC point multiplication.

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

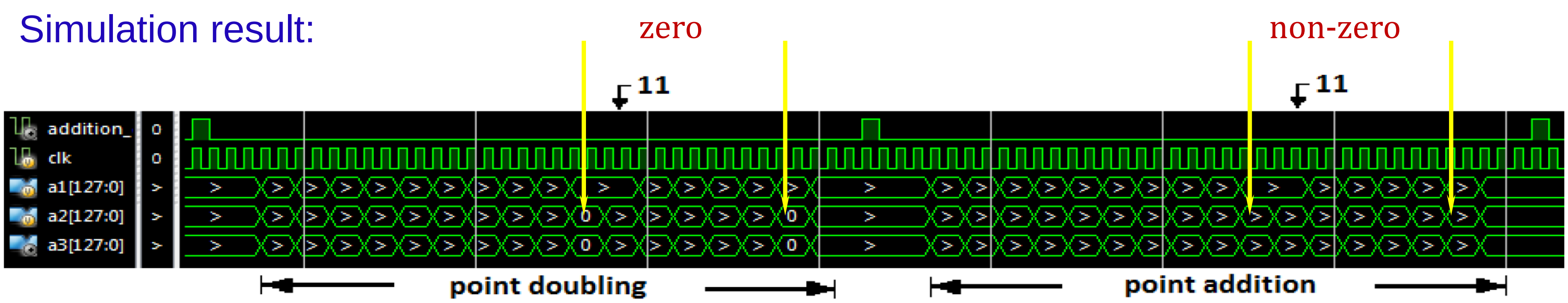
Pin-point the side-channel source

$$\begin{cases} X_3 = (Z_1Z_2 + Y_1Y_2)\,(X_1Z_2 + X_2Z_1)(Z_1^2Z_2^2 + X_1X_2Y_1Y_2) + \\ \qquad\qquad \alpha X_1X_2Z_1Z_2(Z_1Z_2 + Y_1Y_2)) \\ Y_3 = (Z_1Z_2 + X_1X_2)\,((Y_1Z_2 + Y_2Z_1)(Z_1^2Z_2^2 + X_1X_2Y_1Y_2) + \\ \qquad\qquad \beta Y_1Y_2Z_1Z_2(Z_1Z_2 + X_1X_2)) \\ Z_3 = (Z_1Z_2 + X_1X_2)(Z_1Z_2 + Y_1Y_2)(Z_1^2Z_2^2 + X_1X_2Y_1Y_2), \end{cases}$$

Become zero when $(X_1, Y_1, Z_1) = (X_2, Y_2, Z_2)$

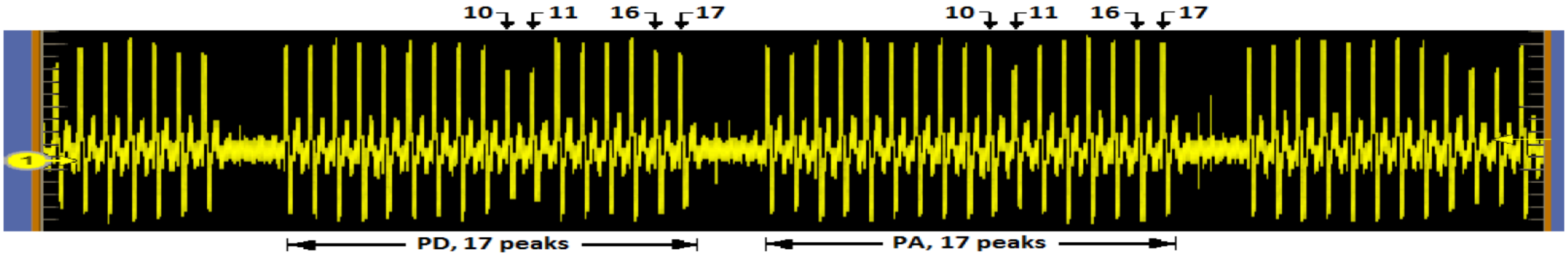
- Point doubling executes 10-th and 16-th multiplications with a zero operand.
- The same multiplications for point additions are with non-zero operands.

- Simulation result:



SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

Power Analysis of UBHC

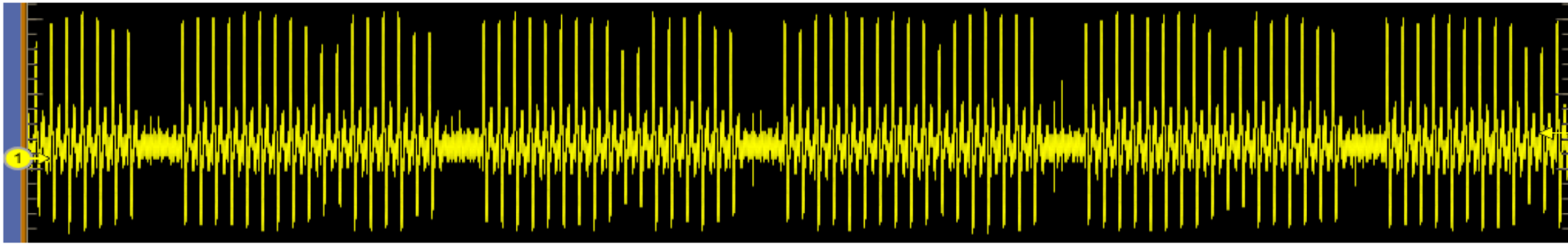


Observations:

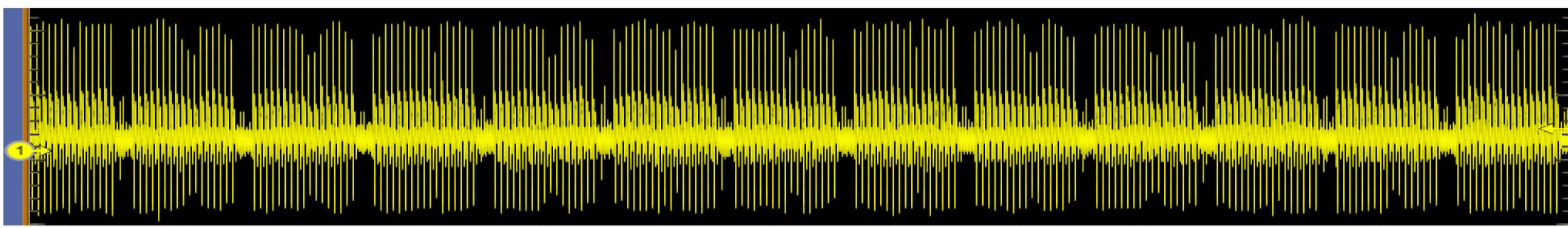
- 17 peaks for executing both PD and PA
  - 17 multiplication cycles
- 11-th peak is lower than other peaks for both PD and PA
  - Operand “a1” remain unchanged from its previous value
- 10-th peak is lower for some point operations
  - Due to “a2”, which is zero for PD
- 16-th and 17-th peaks are also lower with 10-th peak for PD
  - “a2” is zero at 16-th multiplication cycle
  - Transition of datapath from a non-zero (at 15-th) to zero (at 16-th)
  - From a zero (at 16-th) to non-zero (at 17-th)
    - Causes power consumption lower than a non-zero to non-zero transition

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

SPA results of UBHC point multiplication



- PD followed by PD indicates respective secret scalar bit value zero
- PD followed by PA indicates the same as one



Proposed Countermeasure

Modified UBHC point addition arithmetic:

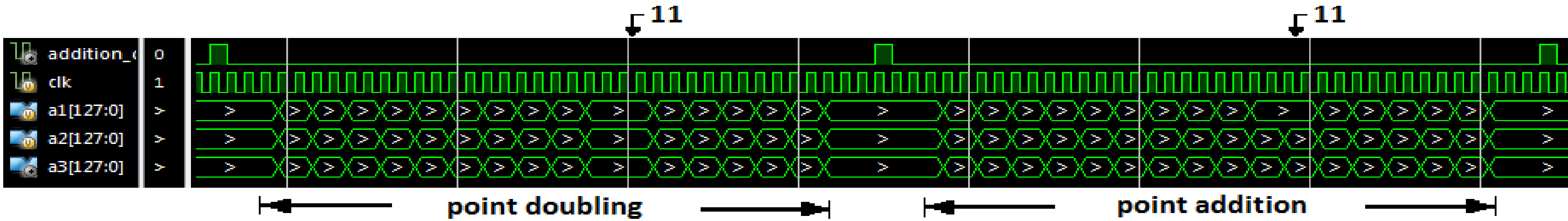
$$m_1 = X_1X_2, \quad m_2 = Y_1Y_2, \quad m_3 = Z_1Z_2,$$
$$m_4 = (X_1 + Z_1)(X_2 + Z_2), \quad m_5 = (Y_1 + Z_1)(Y_2 + Z_2),$$
$$m_6 = m_1m_3, \quad m_7 = m_2m_3, \quad m_8 = m_1m_2 + m_3^2,$$
$$m_9 = m_6(m_2 + m_3)^2, \quad m_{10} = m_7(m_1 + m_3)^2, \quad m_{11} = m_8(m_2 + m_3),$$
$$Z_3 = m_{11}(m_1 + m_3),$$
$$X_3 = \alpha m_9 + m_4m_{11} + Z_3,$$
$$Y_3 = \beta m_{10} + m_5m_8(m_1 + m_3) + Z_3.$$

- No zero valued operand for multiplication
  - Eliminate sources of zeros
  - Distribute $(X_1Z_2 + X_2Z_1)$ (…) and $(Y_1Z_2 + Y_2Z_1)$(…) computations
    - These additions are performed at the last stage of X_3 and Y_3
  - At X_3: $m_4m_{11} + Z_3 = 0$, and at Y_3: $m_5m_8(m_1+m_3) + Z_3 = 0$ for PD
  - Perform X_3 as: $(\alpha m_9 + m_4m_{11}) + Z_3$, and Y_3 as: $(\beta m_{10} + m_5m_8(m_1+m_3)) + Z_3$
- Costs: $15M + 2D \approx 17M$
  - Same as with the original one

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

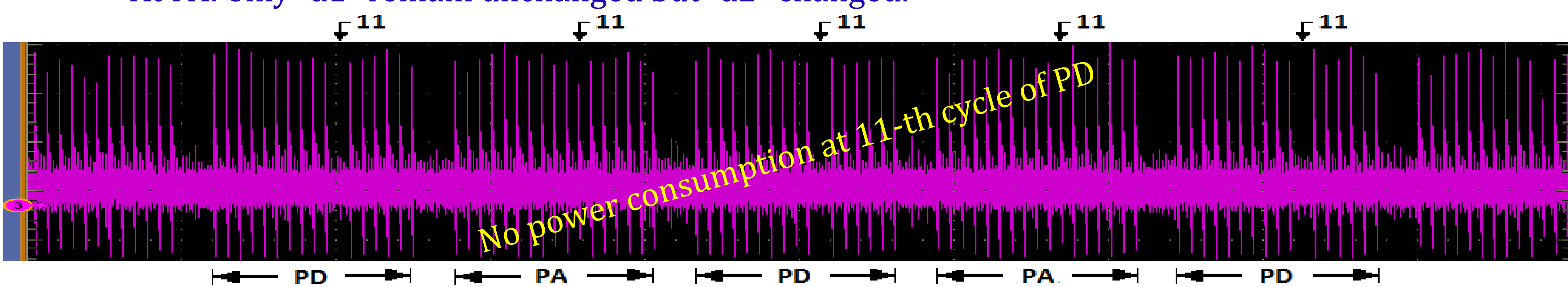
Countermeasure cont…

Simulation result of an implementation of the countermeasure:



Observations:

- At PD: both "a1" and "a2" remain unchanged at 11-th multiplication.
  - No new multiplication is performed in this cycle
- At PA: only "a1" remain unchanged but "a2" changed.



Countermeasure cont…

Causes:

1. It schedules $m_{11}(m_1+m_3)$ and $m_4 m_{11}$ at 10-th and 11-th cycles.
2. It chooses m_{11} as operand a for both multiplications.
3. It chooses $m_1 + m_3$ and m_4 as operand b.

Analysis:

- In case of PD: $m_4 = m_1+m_3$ as $(X_1+Z_1)(X_2+Z_2) = X_1X_2+Z_1Z_2$
  - Perform the same multiplication twice
- But in case of PA they are different.

Countermeasure cont…

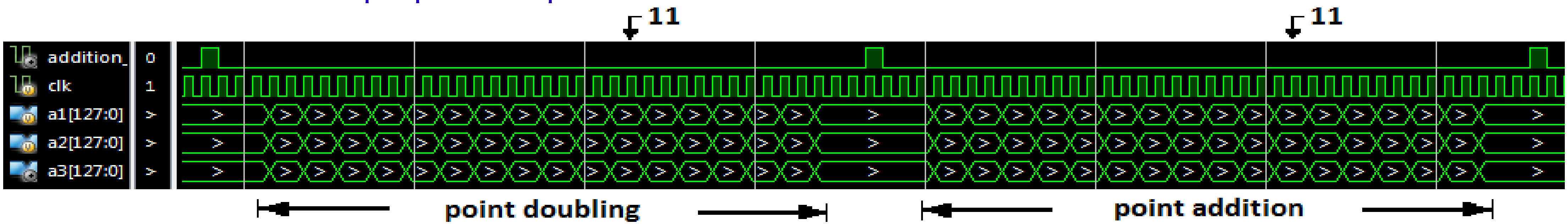
Suggested execution technique of the proposed arithmetic:

PA/PD Cycles	Operations	RTL description
1	$m_1 = x_1 \times x_2$	$temp[0] \leftarrow x_1 \times x_2$
2	$m_2 = y_1 \times y_2$	$temp[1] \leftarrow y_1 \times y_2$
3	$m_3 = z_1 \times z_2$	$temp[2] \leftarrow z_1 \times z_2$
4	$m_1 \times m_2$	$temp[3] \leftarrow temp[0] \times temp[1]$
5	$m_4 = (x_1 + z_1)(x_2 + z_2)$	$temp[4] \leftarrow (x_1 \oplus z_1) \times (x_2 \oplus z_2)$
6	$m_6 = m_1 \times m_3$	$temp[5] \leftarrow temp[0] \times temp[2]$
7	$m_{11} = m_8 \times (m_2 + m_3)$	$temp[6] \leftarrow (temp[3] \oplus temp[2]^2) \times (temp[1] \oplus temp[2])$
8	$m_9 = m_6 \times (m_2 + m_3)^2$	$temp[5] \leftarrow temp[5] \times (temp[1] \oplus temp[2])^2$
9	$m_{11} \times m_4$	$temp[4] \leftarrow temp[6] \times temp[4]$
10	$\alpha \times m_9$	$temp[5] \leftarrow \alpha \times temp[5]$
11	$Z_3 = m_{11} \times (m_1 + m_3)$	$temp[6] \leftarrow temp[6] \times (temp[0] \oplus temp[2])$
12	$m_7 = m_2 \times m_3$	$temp[5] \leftarrow temp[1] \times temp[2]$
13	$m_5 = (y_1 + z_1)(y_2 + z_2)$	$temp[4] \leftarrow (y_1 \oplus z_1) \times (y_2 \oplus z_2)$
14	$m_{10} = m_7 \times (m_1 + m_3)^2$	$temp[5] \leftarrow temp[5] \times (temp[0] \oplus temp[2])^2$
15	$m_5 \times m_8$	$temp[4] \leftarrow temp[4] \times (temp[3] \oplus temp[2]^2)$
16	$\beta \times m_{10}$	$temp[5] \leftarrow \beta \times temp[5]$
17	$(m_5 m_8) \times (m_1 + m_3)$	$temp[4] \leftarrow temp[4] \times (m_1 \oplus m_3)$
	Final outputs are:	$X_3 \leftarrow temp[4] \oplus temp[5] \oplus temp[6]$ at clock cycle 12, $Z_3 \leftarrow temp[6]$ at clock cycle 15, $Y_3 \leftarrow temp[4] \oplus temp[5] \oplus temp[6]$ at clock cycle 19.

- PA/PD independent data scheduling
- Operand value changes for every multiplications

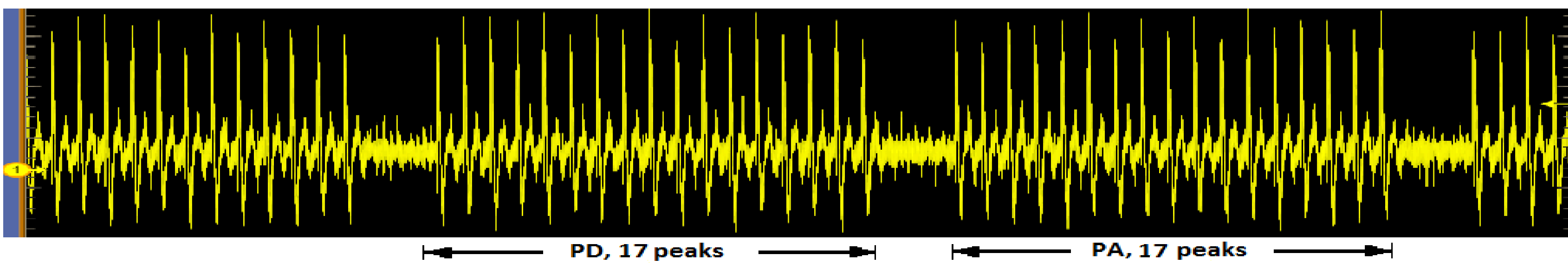
Validation of proposed technique

Simulation result of the proposed implementation and countermeasure:



Observations:

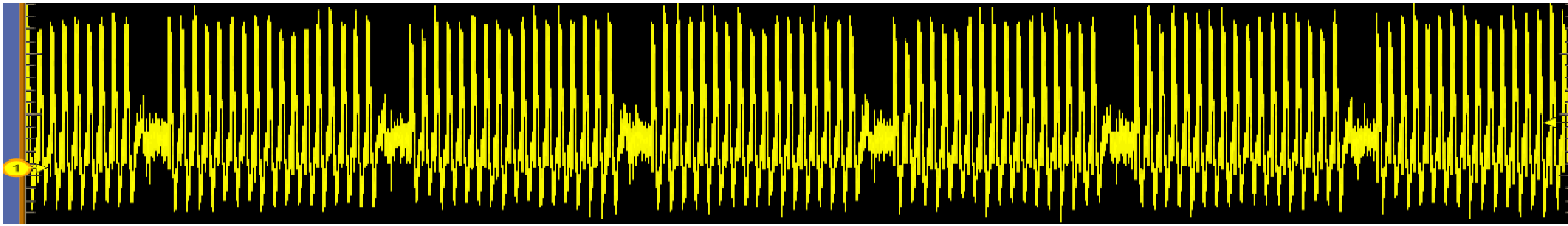
- At PD and PA: values both “a1” and “a2” change at every cycles



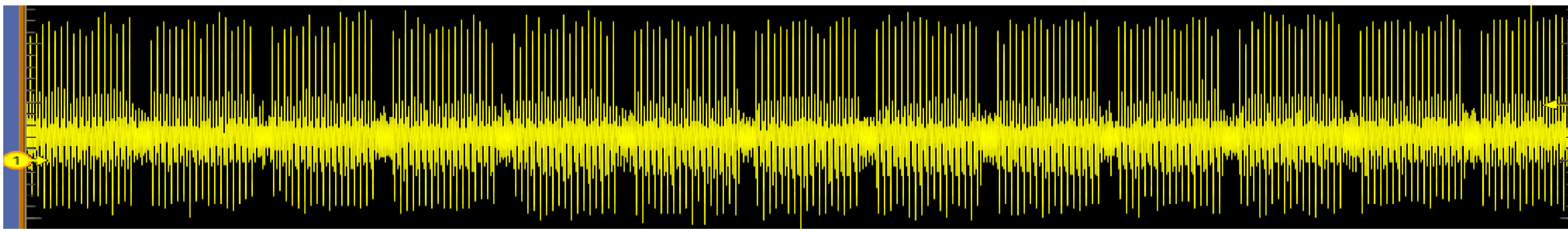
- No observable difference between PD and PA power consumption graphs

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

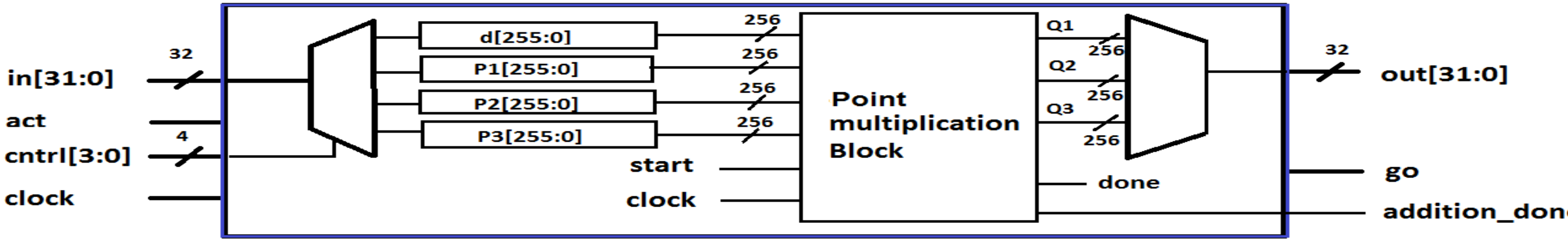
SPA results of UBHC point multiplication



- PD and PA cannot be identified by observing these power profiles
- Secret scalar bit cannot be guessed



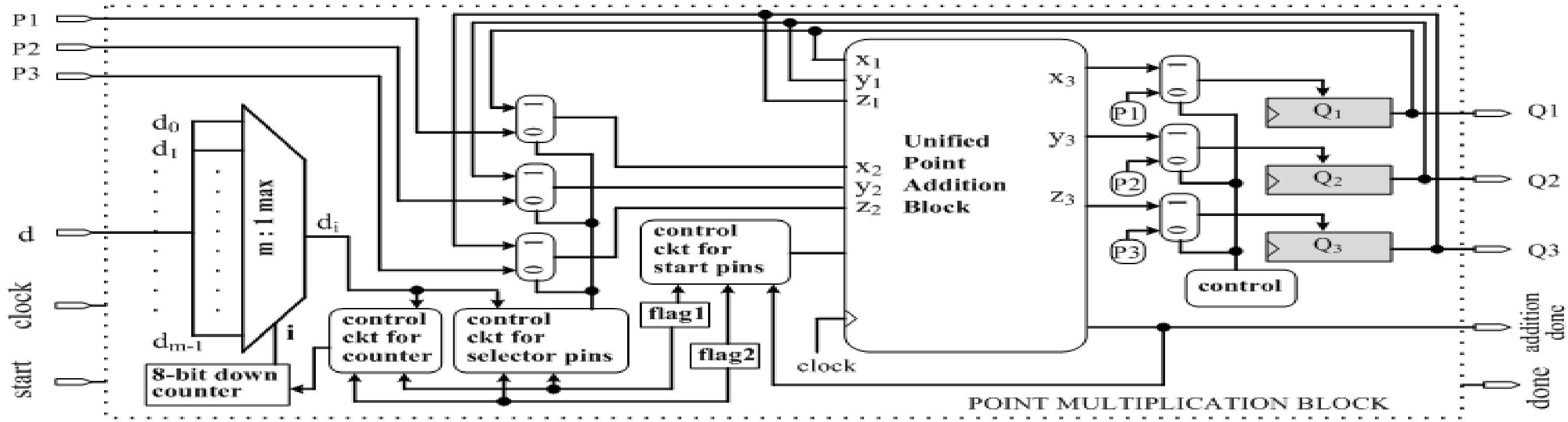
Architectural description



- Consists of four n-bit registers
  - x, y, z coordinates and integer d
- 32-bit input and output data interface
- Total 5-bit control signals
  - "act" to enable/disable the whole elliptic curve processor block
  - Four bit cntrl signal to select different modes
    - Input mode
    - Selection of input
    - Ready for output
- Two status signals
  - To keep track of every point addition
    - Can be discarded before package
  - End of a point multiplication

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

Datapath of Point Multiplication



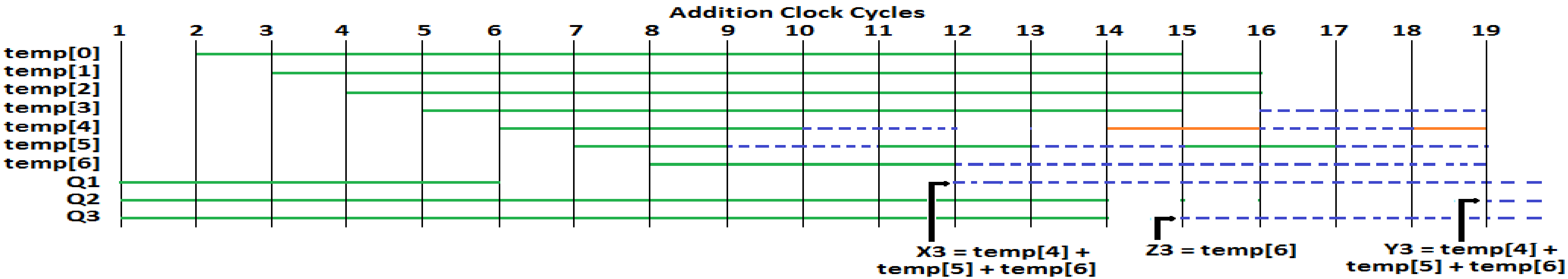
- Executes left-to-right binary algorithm
- "flag1" and "flag2" indicates PD and PA
- Point Addition Block
  - One Hybrid Karatsuba multiplier [17]
  - Only one clock for an n-bit multiplication
  - 20 clock cycles per point addition
    - 17 multiplication clock cycles
    - 3 for data ready and restore
  - Seven temporary registers

- Intermediate and final results are stored at Q_i registers
  - Optimum number of registers
  - Total 14 registers
    - 4 input: d, P_1, P_2, P_3
    - 3 output: Q_1, Q_2, Q_3
    - 7 temporary: temp[i], $0 \le i \le 6$

[17]. Rebeiro, C., Mukhopadhyay, D.: High speed compact elliptic curve cryptoprocessor for FPGA platforms. INDOCRYPT 2008.

Analysis and Results

- Optimization of temporary registers
  - Life time analysis



- Changes of a line style in a lifeline indicates the register is reassigned.
- A lifeline with same line style from clock i to j indicates:
  - The register is assigned with a value at i-th clock
  - The value is used finally at (j-1)-th clock cycle
  - The register is reassigned with new value at j-th clock

SeCoE, Intel Corporation, United States
IIT Kharagpur, India
COSIC, KU Leuven, Belgium

Results Cont…

- Area and timing results of scalar multiplication on FPGA

Device	128 − bit			233 − bit			256 − bit		
	Slice	Clock [MHz]	Time [μs]	Slice	Clock [MHz]	Time [μs]	Slice	Clock [MHz]	Time [μs]
Virtex-2Pro	8,345	110	37	19,043	110	67	21,423	98	82
Virtex-4	8,713	138	29	19,352	134	55	21,325	103	78
Virtex-6	3,924	182	22	7,150	172	43	11,083	146	55
Virtex-7	3,432	195	21	6,032	183	40	9,115	162	49

- Performance comparison with existing designs

Work	Platform	Field [m]	Slices Count	Clock [MHz]	Latency [μs]	Area × Latency × [10^5]
Ours	$XC4V140$	233	19,352	134	55	10.6
Unified Edwards [6]	$XC4V140$	233	21,816	50	170	37.1
Unified Huff [7]	$XC4V140$	233	20,437	81	73	14.9

Reduced area –
Improved efficiency –

[6]. Chatterjee, A., Sengupta, I.: FPGA implementation of Binary Edwards curve using ternary representation. In: GLSVLSI 2011.
[7]. Chatterjee, A., Sengupta, I.: High-speed unified elliptic curve cryptosystem on FPGAs using binary Huff curves. VDAT 2012.

Performance Analysis

- Unified binary Huff curve (UBHC) formula is faster than the unified formula on Edwards curve [2].
  - Costs of Edwards: 18M+7D (or 21M+4D)
  - Costs of Huff: 15M + 2D
- An n-bit point multiplication on proposed UBHC arithmetic
  - Costs: 25.5n M
  - Not a cheap solution against side-channel attacks
  - Costly than double-and-add always with Lopez-Dahab
    - Costs: 19n M
  - Much costly than Montgomery ladder, based on Lopez-Dahab fast point multiplication [16] trick
    - Costs: 6n M.
- Side-channel security is not the main goal of a Huff curve
- **Complete addition formula for all subgroups**
  - Even in a subgroup that does not contain the **points at infinity**
  - Secure against exceptional procedure attacks and batch computing

Huff Curve with proposed arithmetic is the current winner!!!

[2]. Bernstein, D.J., Lange, T., Rezaeian Farashahi, R.: Binary Edwards Curves. CHES 2008..
[16]. L´opez, J., Dahab, R.: Fast multiplication on elliptic curves over GF(2^m) without precomputation. CHES 1999.

Corrections!!

- Page 356, In paragraph before Architectural Description:

"There are 18 peaks for computing 18 multiplications." would be "There are 17 peaks for computing 17 multiplications."

- Page 361, just before Conclusion:

"In this respect, the Huff curve is one step ahead compared to its competitors Edwards [2] and Generalized Hessian curves [9] − on both of which the point addition is complete only on some specific subgroups."

It is a misrepresentation which would be:

- Edwards [2]
  - Complete addition formula same as Huff curve
  - Unified formula with 15M+2D Vs. 21M + 4D costs (without any assumptions)
  - Cost may be reduced with assumptions [*].
- Generalized Hessian curves [9]
  - Complete addition formula [**] with suitably chosen parameters.
    - If $\mathbf{c}$ is not a cube in $\mathbf{F}$, where $X^3 + Y^3 + \mathbf{c}Z^3 = \mathbf{d}XYZ$ with $\mathbf{c},\mathbf{d} \in \mathbf{F}$, $\mathbf{c} \neq 0$, $\mathbf{d}^3 \neq 27\mathbf{c}$
  - Unified formula, 12M in Projective coordinates.

[2]. Bernstein, D.J., Lange, T., Rezaeian Farashahi, R.: Binary Edwards Curves. CHES 2008..
[9]. Farashahi, R.R., Joye, M.: Efficient Arithmetic on Hessian Curves. PKC 2010.
[*]. http://hyperelliptic.org/EFD/g12o/index.html
[**]. http://cr.yp.to/talks/2009.07.17/slides.pdf

Thank you

Questions? santosh.ghosh@intel.com