Side Channel Analysis Attacks on FPGA Implementations of Cryptographic Algorithms

Sıddıka Berna Örs

Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC,
Kasteelpark Arenberg 10, B–3001 Leuven-Heverlee, Belgium

siddika.bernaors@esat.kuleuven.ac.be

Outline

- Motivation
- Power-Analysis Attacks
- Virtex 800 Field Programmable Logic Array (FPGA)
- The Measurement Setup
- Power Consumption Characteristics of FPGA
- Attacking an Implementation of an Elliptic-Curve Point Multiplication

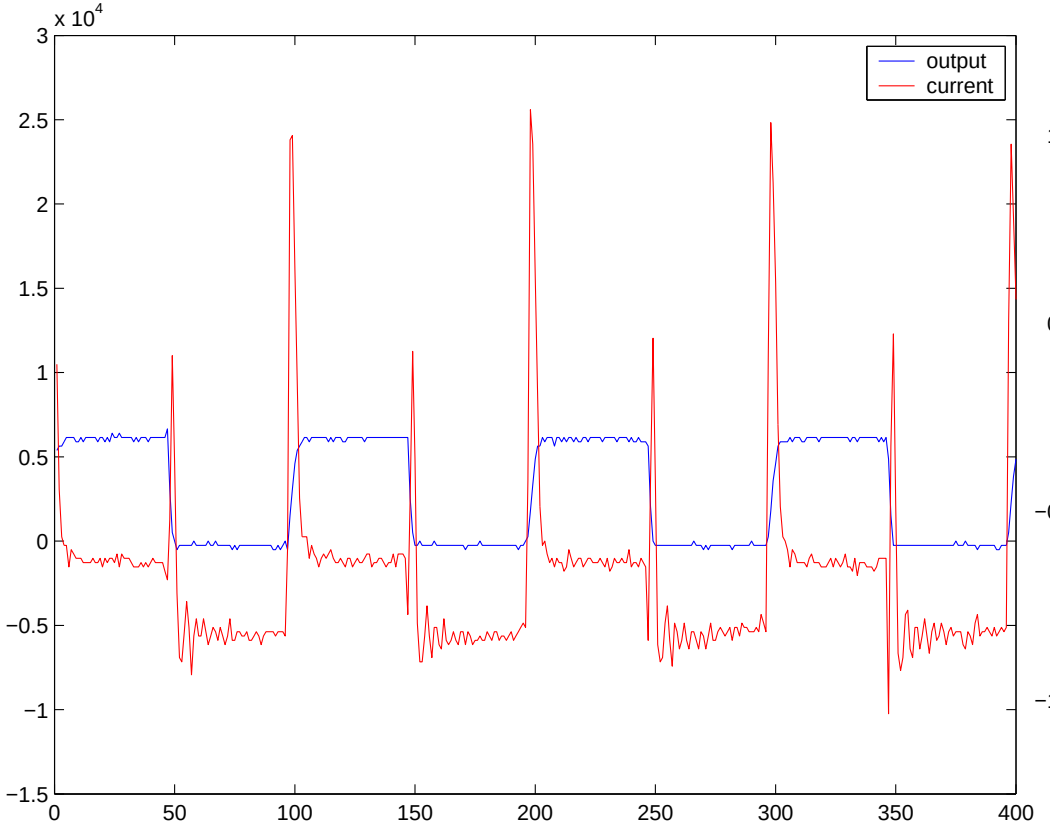
Motivation

- For implementations of cryptographic algorithms, not only the speed and the size of the circuit, but also their security against implementation attacks such as side-channel attacks are important.

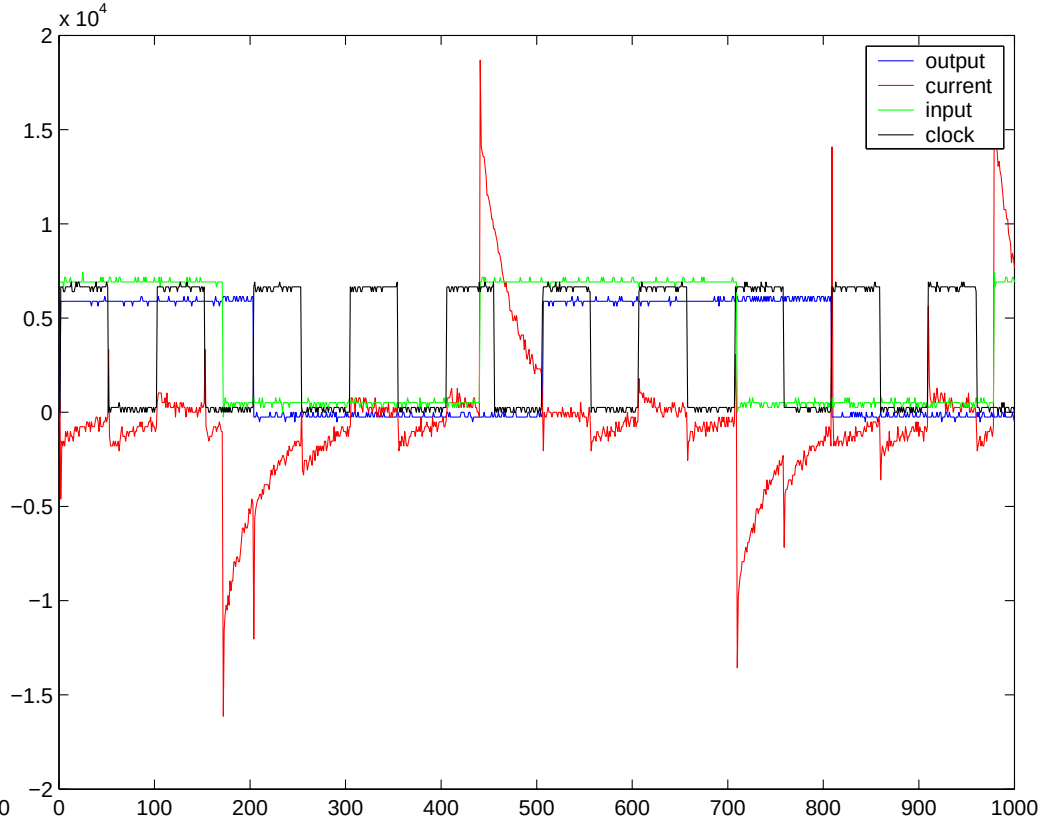
- Field Programmable Gate Arrays (FPGAs) are becoming increasingly popular, especially for rapid prototyping.

- The flexibility of FPGAs is an important advantage in lab environments. It is therefore natural to use FPGAs to assess the vulnerability of hardware implementations to power-analysis attacks.

Power -Analysis Attacks: Why do they work?



Power measurement of inverter



Power measurement of D-flipflop

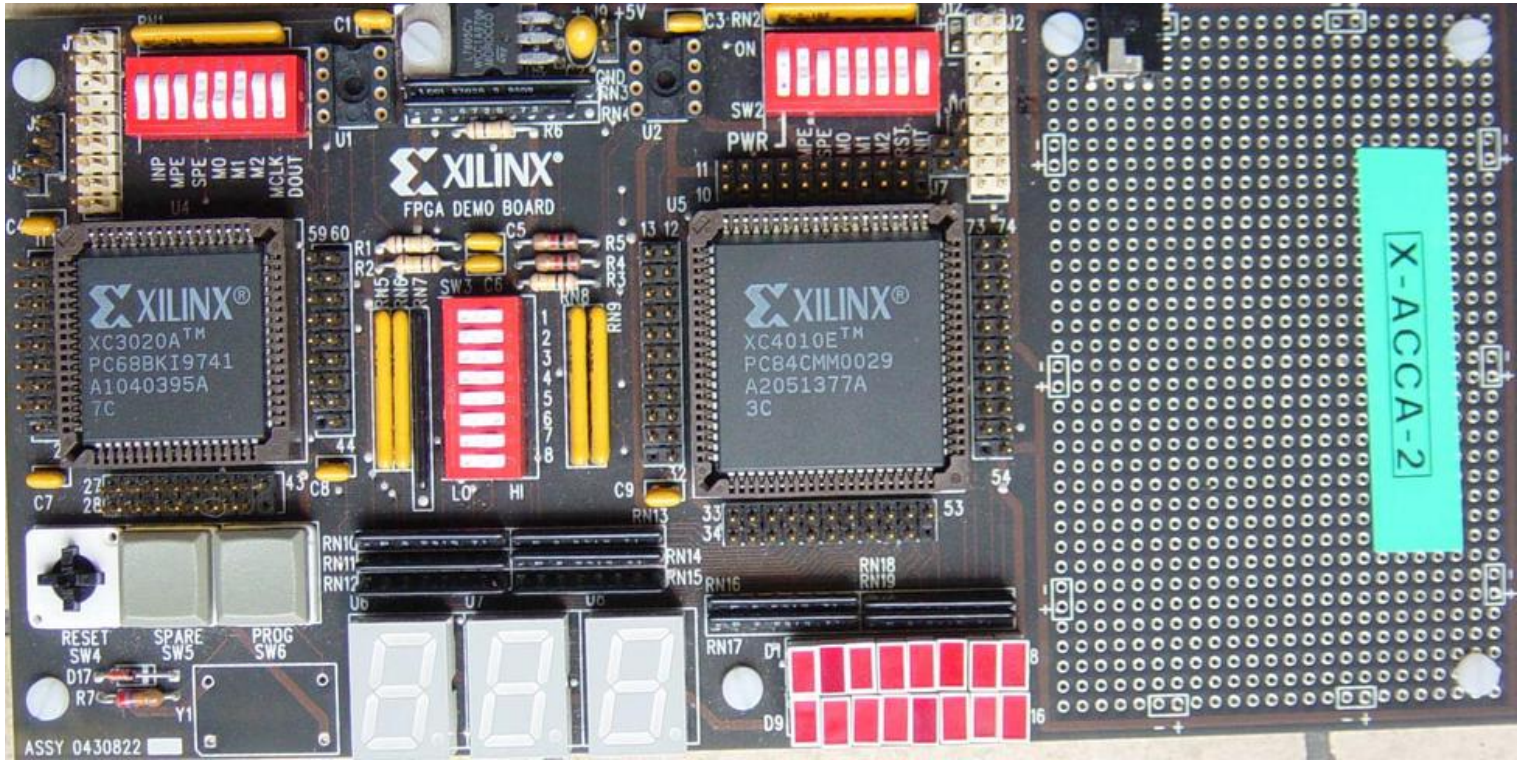
Types of Power-Analysis Attacks

Simple Power-Analysis Attacks:
It is assumed that every instruction has its unique power-consumption trace. An attacker simply monitors the device's power consumption while it performs a cryptographic operation.

Differential Power-Analysis Attacks:
The attacker writes a simple computer program that executes the algorithm where a part of the key is used. The program calculates the result for different inputs for the same key values. These values allow to predict the power consumption, which is for example related to the Hamming-weight of the internal data.

The attacker feeds the same input values which he used in the model to the real device and measures its power consumption. Then the attacker correlates the predictions of the model with the real power consumption values.

Why did we need to design our FPGA Board?



Xilinx FPGA Demo Board

FPGA Decision

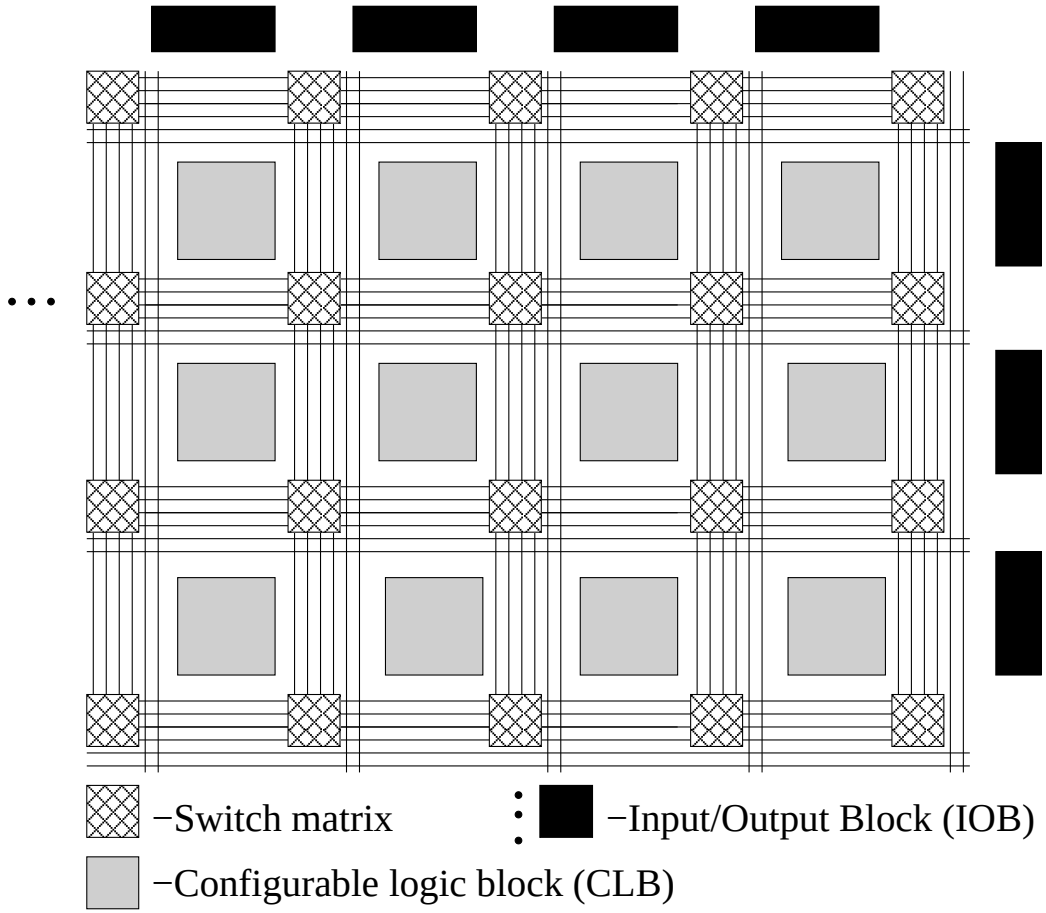
We use a Xilinx XCV800 FPGA from the Virtex series in a HQ240C package. Reasons for this particular choice include:

1. The resources are sufficient to implement a 160-bit elliptic curve point multiplication.

2. This is the most powerful FPGA that can be used for hand-mounting on the board.

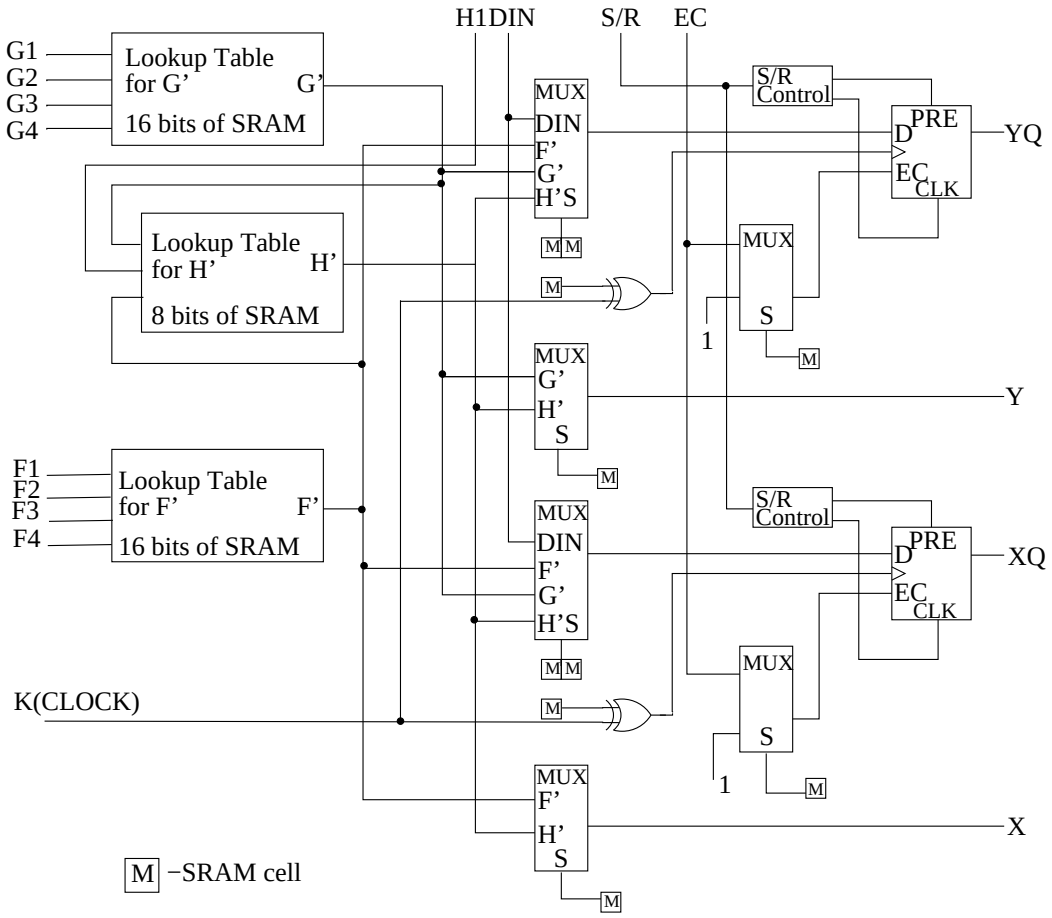
3. The architecture is made of combinational and memory elements. Because of this property it is a good representative of application specific integrated circuits (ASICs).

Virtex 800 FPGA



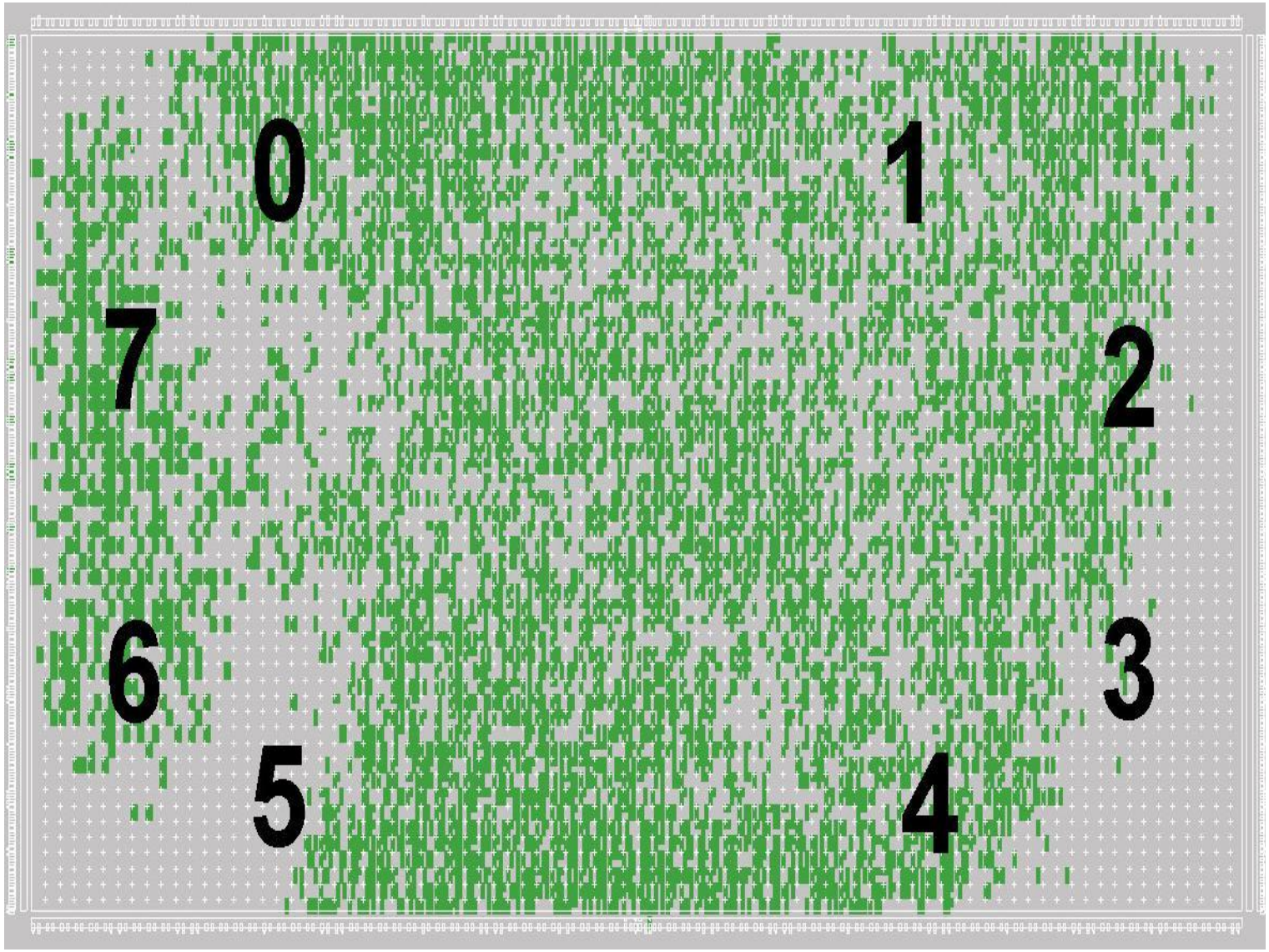
The FPGA architecture

Configurable Logic Block



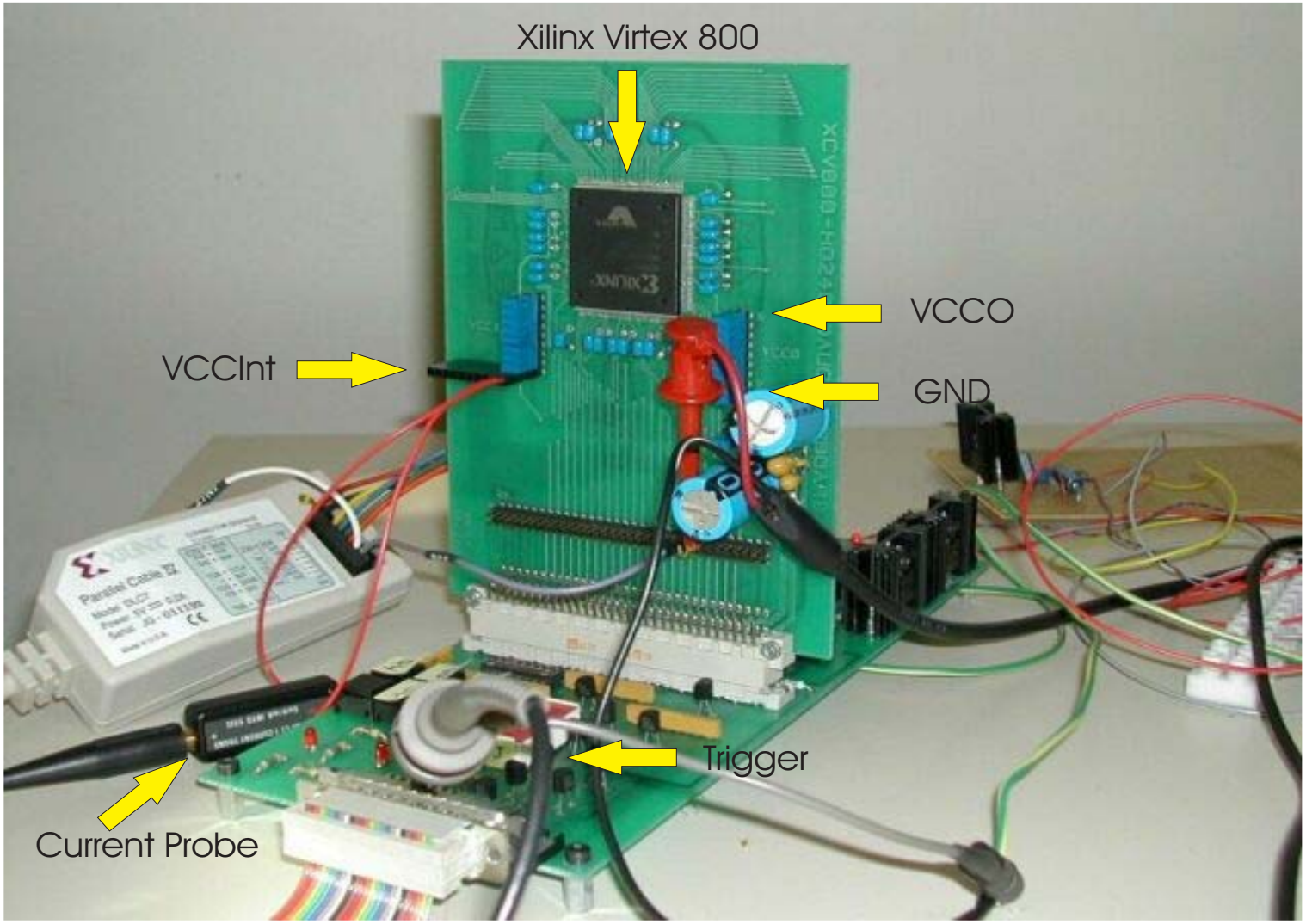
Simplified diagram of CLB

I/O Banking



FPGA Floor Plan

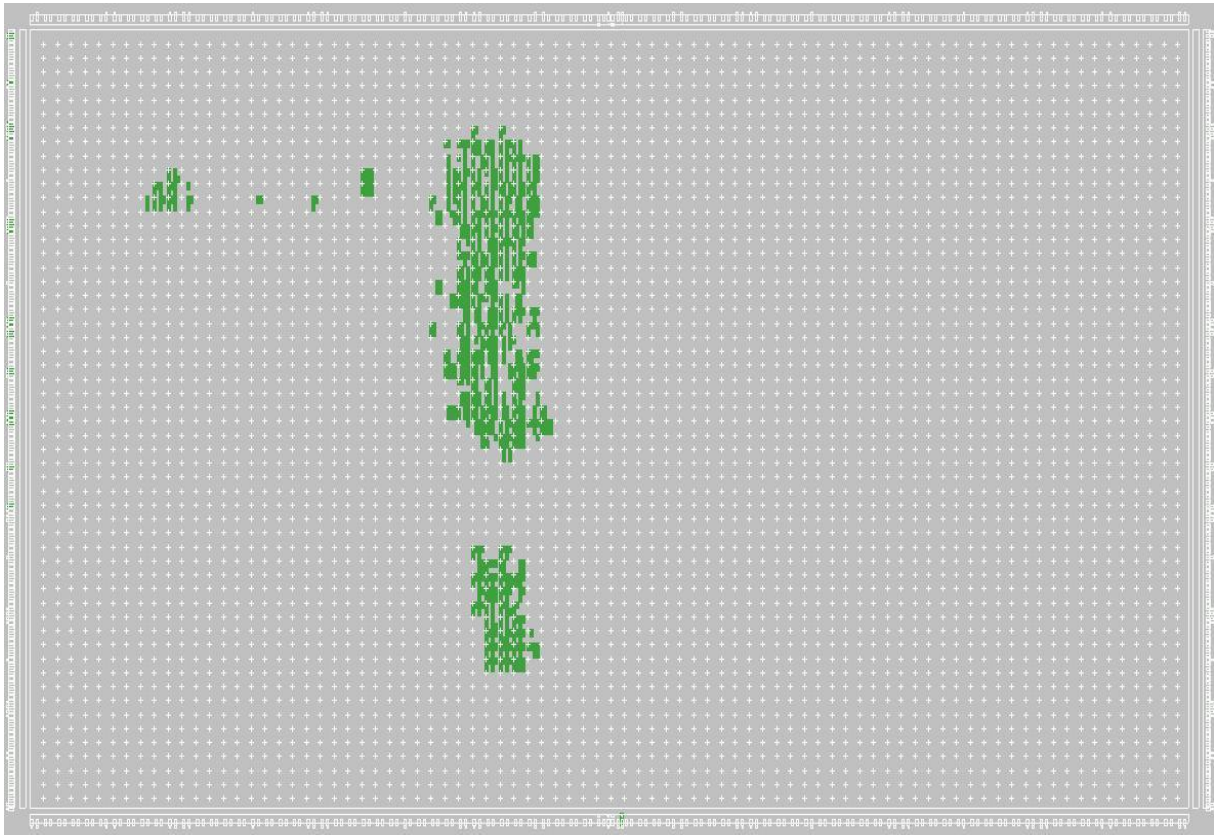
The Measurement Setup



The measurement setup. On the daughter board the current probe is connected to VCCINT. Alternatively it can be connected to the VCCO of the individual banks, or the GND.

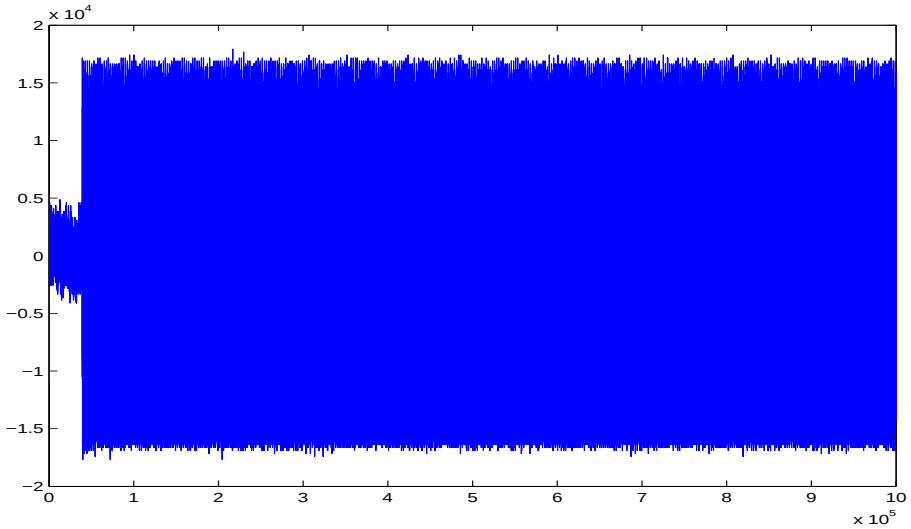
Power Consumption Characteristics 1/3

The circuit does not use all of the FPGAs resources, then the noise which is produced by the unused parts might be larger than the signal produced by the circuit.

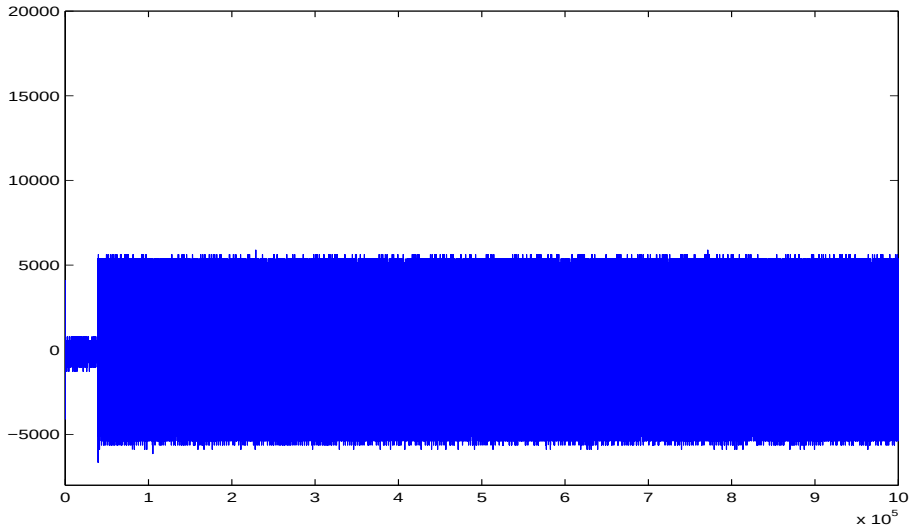


Floor plan of 8-bit EC Point Addition Circuit

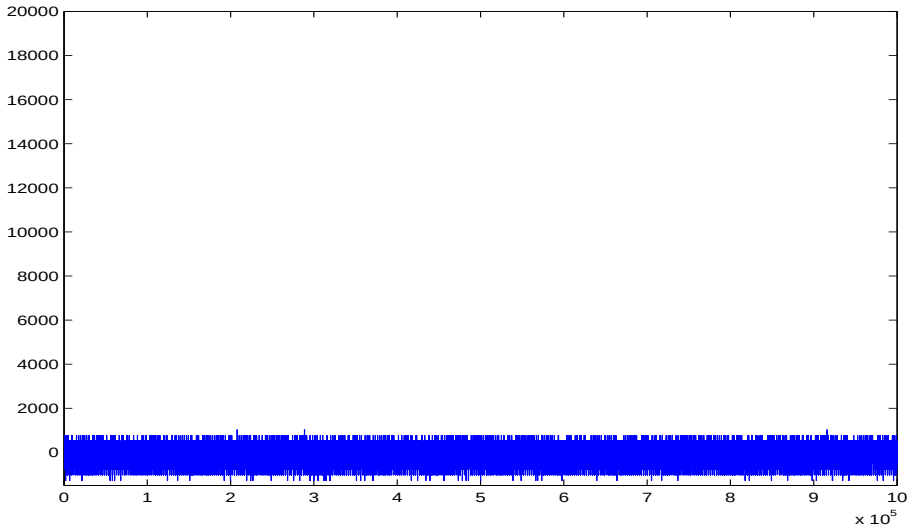
Power Consumption Characteristics 2/3



Measurement from the total VCCINT

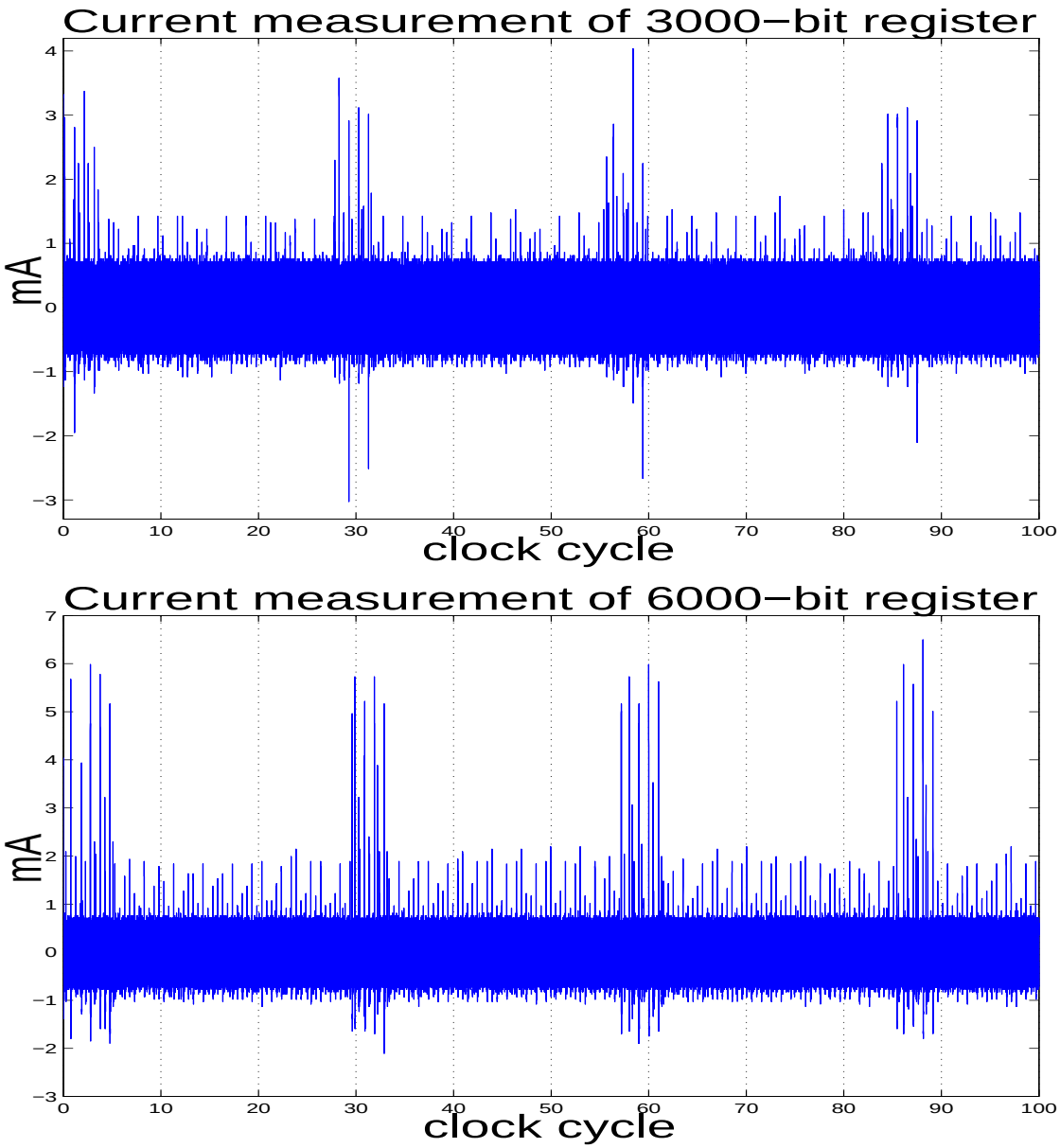


Measurement from VCCINT of the most full bank



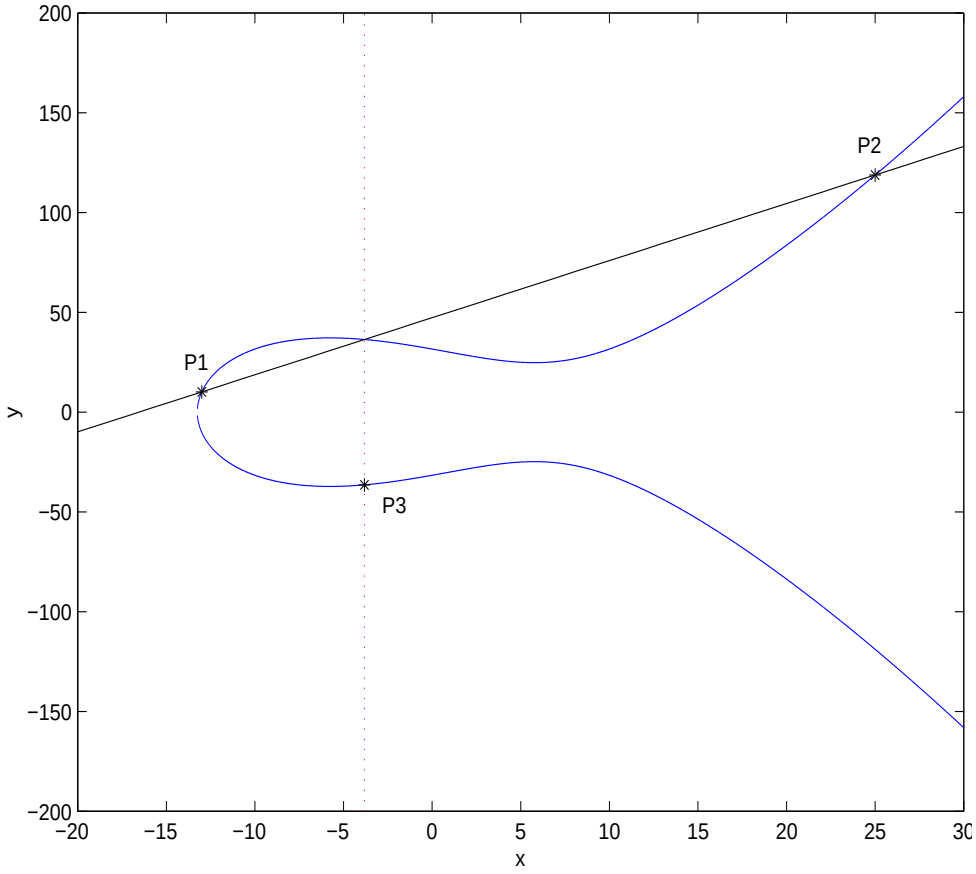
Measurement from VCCINT of the empty bank

Power Consumption Characteristics 3/3

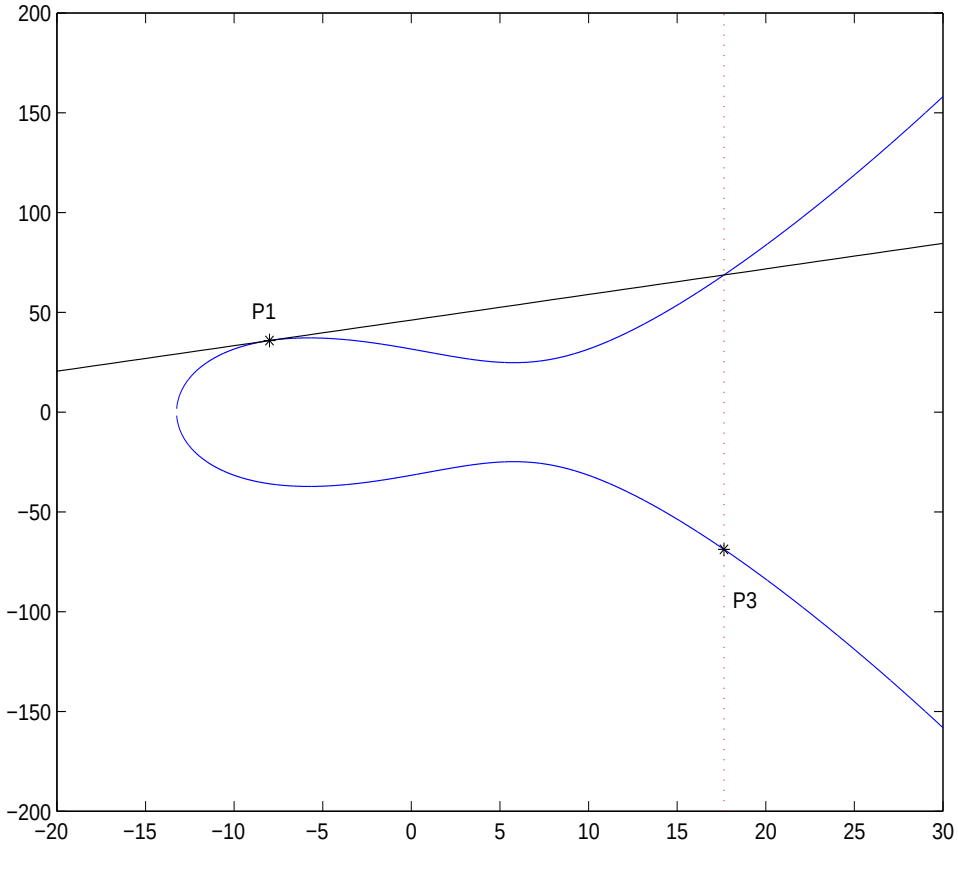


Elliptic Curve Group over R

Definition: set of the solutions of Weierstrass equation $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ over a field and the point at infinity $\mathcal{O}$



Adding two points on Elliptic Curve



Doubling a point on Elliptic Curve

Elliptic Curve Group over $GF(p)$ $p > 3$, by affine coordinates

$$E : y^2 = x^3 + ax + b$$

$$P_1 = (x_1, y_1),\ P_2 = (x_2, y_2) \text{ and } P_3 = (x_3, y_3) = P_1 + P_2$$

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & if\ P_1 \neq P_2 \\ \frac{3x_1^2 + a}{2y_1} & if\ P_1 = P_2 \end{cases}$$

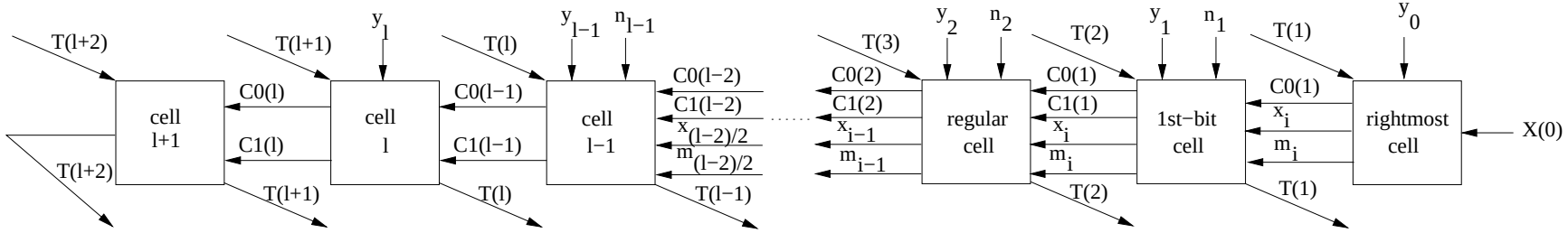
We have implemented the arithmetic for a 160-bit prime field with a Montgomery modular multiplier (MMM) without final subtraction.

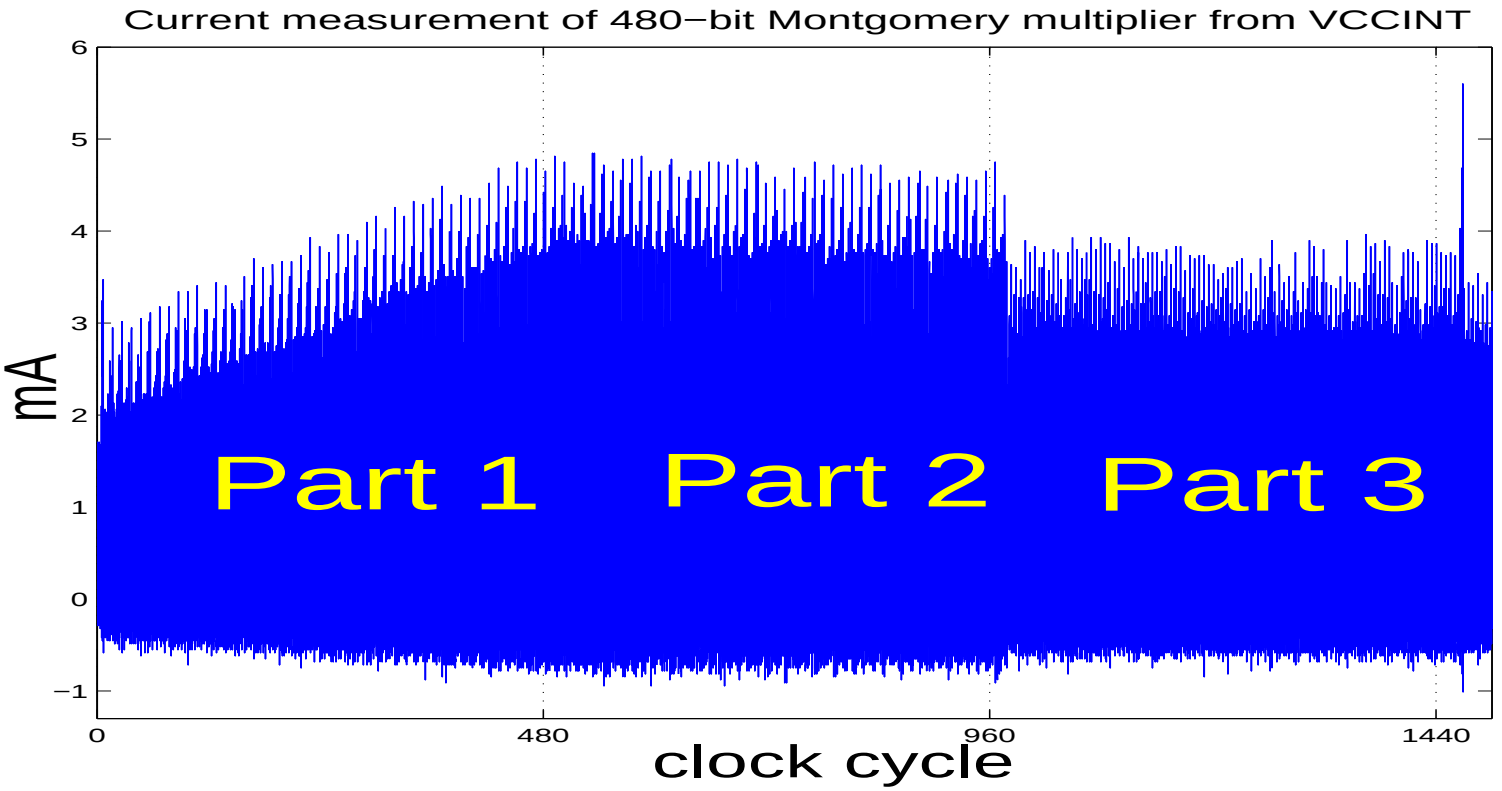
Montgomery Modular Multiplier 1/2

$$Mont(x, y) = xyR^{-1} \mod N,$$

$$m_i = t_{i-1,1} \oplus x_i \times y_0,\ T_i = 2^{-1}(T_{i-1} + x_i \times Y + m_i \times N),\ \text{where}$$

$$i = 0, \cdots, l+1 \text{ and } T_{-1} = 0.$$





Montgomery Modular Multiplier 2/2

- Part 1: The number of bits of T which are updated is increasing until clock cycle 480.

- Part 2: After 480th clock cycle all the bits of the T register have a value and all of them are updated before clock cycle 960.

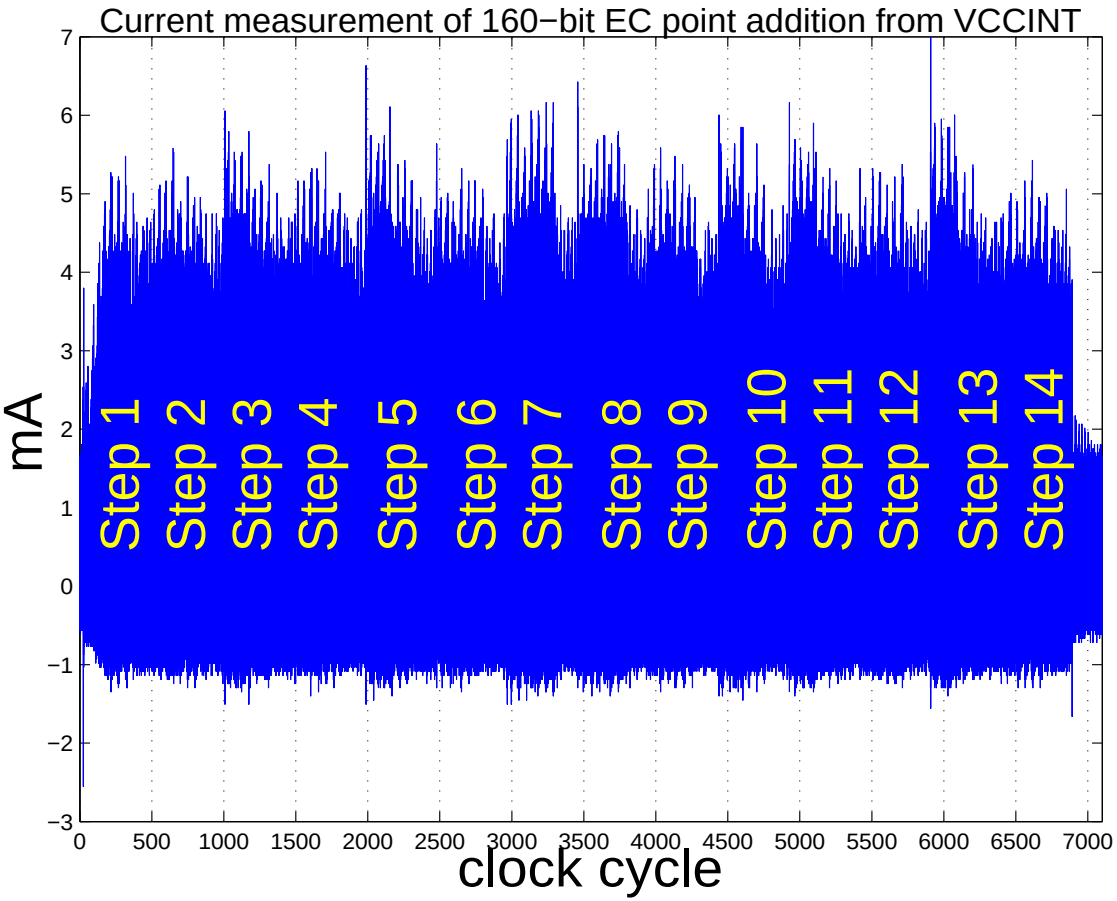
- Part 3: Because there is no new input on the LSB of the systolic array, starting from clock cycle 961 the number of bits of the T register that are updated decreases.

Elliptic Curve Point Addition

Input: $P_1 = (x, y, 1, a),\ P_2 = (X_2, Y_2, Z_2, aZ_2^4)$

Output: $P_1 + P_2 = P_3 = (X_3, Y_3, Z_3, aZ_3^4)$

1. $T_1 \leftarrow Z_2^2$
2. $T_2 \leftarrow xT_1$
3. $T_1 \leftarrow T_1 Z_2, \quad T_3 \leftarrow X_2 - T_2$
4. $T_1 \leftarrow yT_1$
5. $T_4 \leftarrow T_3^2, \quad T_5 \leftarrow Y_2 - T_1$
6. $T_2 \leftarrow T_2 T_4,$
7. $T_4 \leftarrow T_4 T_3, \quad T_6 \leftarrow 2T_2$
8. $Z_3 \leftarrow Z_2 T_3, \quad T_6 \leftarrow T_4 + T_6$
9. $T_3 \leftarrow T_5^2$
10. $T_1 \leftarrow T_1 T_4, \quad X_3 \leftarrow T_3 - T_6$
11. $T_6 \leftarrow Z_3^2, \quad T_2 \leftarrow T_2 - X_3$
12. $T_3 \leftarrow T_5 T_2,$
13. $T_6 \leftarrow T_6^2, \quad Y_3 \leftarrow T_3 - T_1$
14. $aZ_3^4 \leftarrow aT_6$

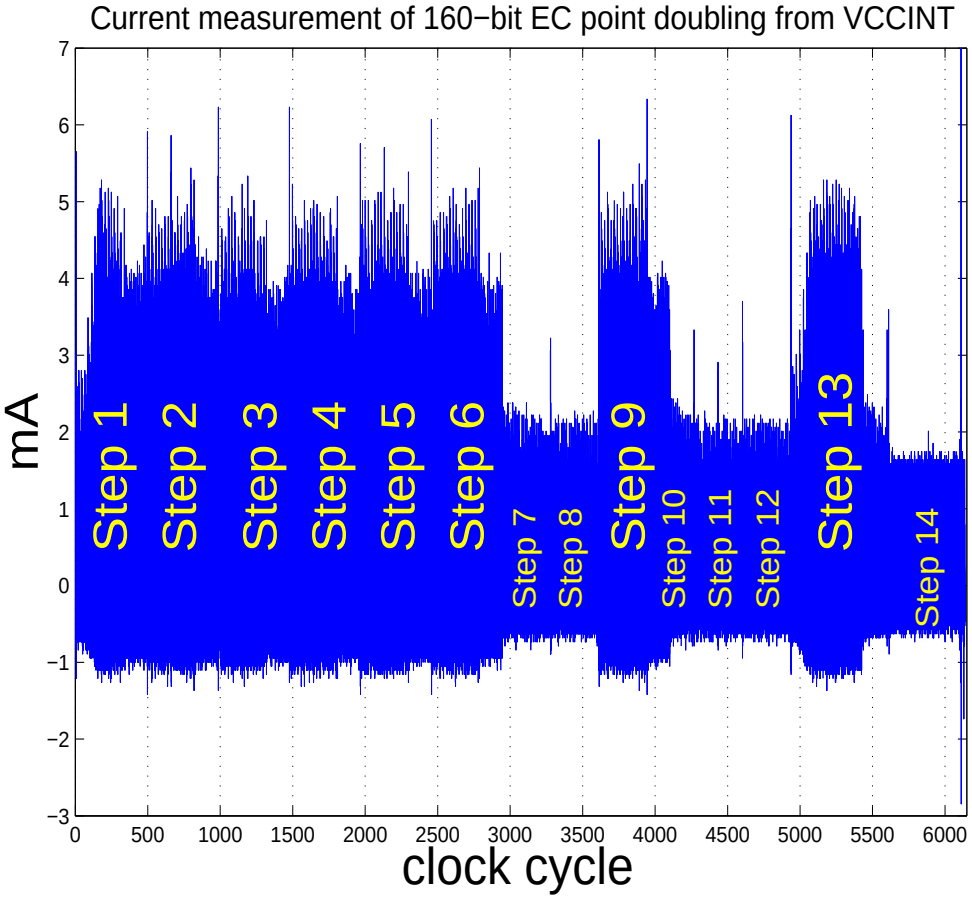


Elliptic Curve Point Doubling

Input: $P_1 = (X_1, Y_1, Z_1, aZ_1^4)$

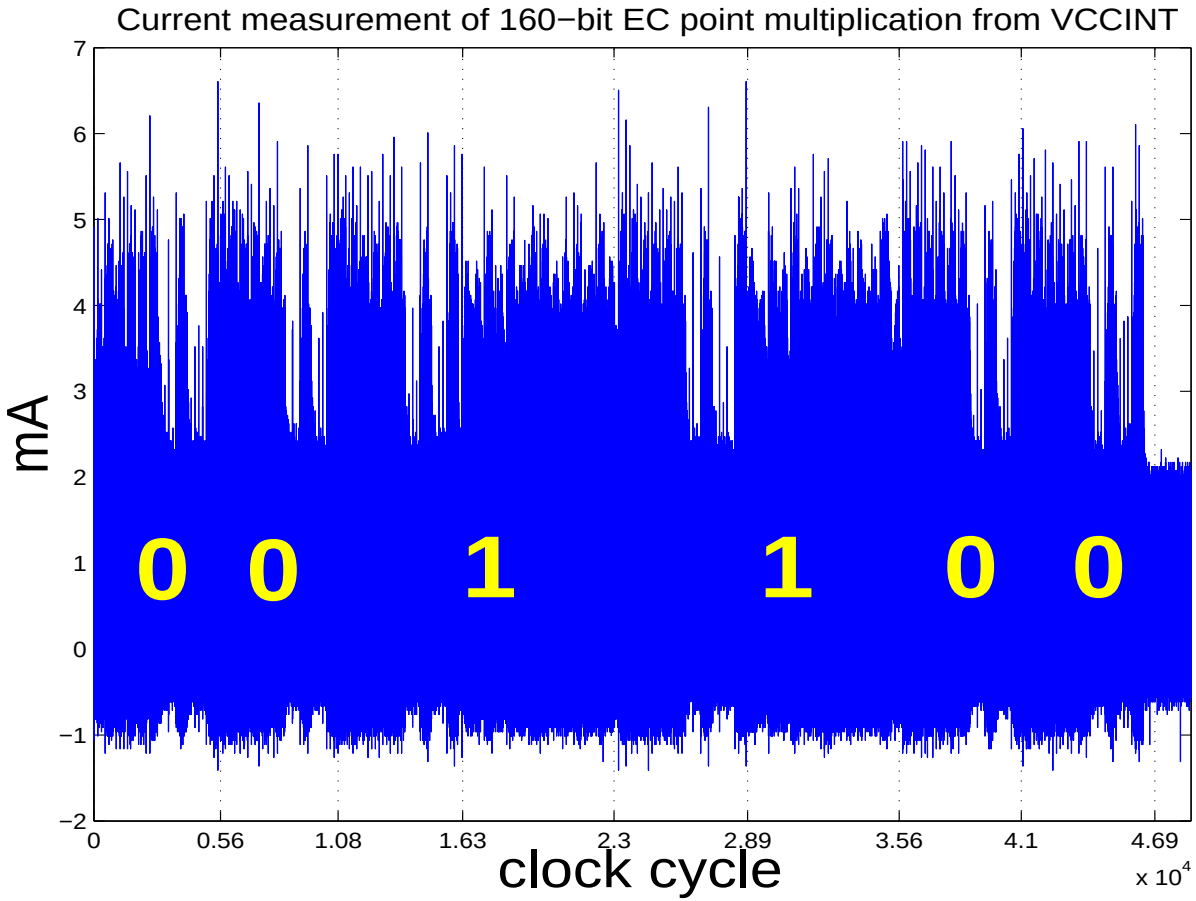
Output: $2P_1 = P_3 = (X_3, Y_3, Z_3, aZ_3^4)$

1.	$T_1 \leftarrow Y_1^2$,	$T_2 \leftarrow 2X_1$
2.	$T_3 \leftarrow T_1^2$,	$T_2 \leftarrow 2T_2$
3.	$T_1 \leftarrow T_2 T_1$,	$T_3 \leftarrow 2T_3$
4.	$T_2 \leftarrow X_1^2$,	$T_3 \leftarrow 2T_3$
5.	$T_4 \leftarrow Y_1 Z_1$,	$T_3 \leftarrow 2T_3$
6.	$T_5 \leftarrow T_3 \left(aZ_1^4\right)$,	$T_6 \leftarrow 2T_2$
7.		$T_2 \leftarrow T_6 + T_2$
8.		$T_2 \leftarrow T_2 + \left(aZ_1^4\right)$
9.	$T_6 \leftarrow T_2^2$,	$Z_3 \leftarrow 2T_4$
10.		$T_4 \leftarrow 2T_1$
11.		$X_3 \leftarrow T_6 - T_4$
12.		$T_1 \leftarrow T_1 - X_3$
13.	$T_2 \leftarrow T_2 T_1$,	$aZ_3^4 \leftarrow 2T_5$
14.		$Y_3 \leftarrow T_2 - T_3$



Elliptic Curve Point Multiplication

Input: EC point $P = (x, y)$, integer k,
$k = (k_{l-1}, k_{l-2}, \cdots, k_0)_2$, $k_{l-1} = 1$
Output: $Q = (x', y')$
$Q \leftarrow P$
for i from $l - 2$ downto 0
    $Q \leftarrow 2Q$
    if $k_i = 1$
        $Q \leftarrow Q + P$



The key used during this measurement is 1001100.

Conclusion

- We introduced a new platform for evaluating power analysis.
- We characterized the power consumption of a XILINX Virtex 800 FPGA and concluded that it is similar to the power consumption of an ordinary ASIC in CMOS technology.
- Therefore, it is possible to draw conclusions about the vulnerability of a certain circuit by performing power-analysis attacks on an FPGA-implementation.
- Consequently, our approach describes the first cheap and efficient way to conduct power-analysis attacks on a real implementation (i.e., not on a software simulation) of a circuit in a very early stage of the design flow.