Efficient Power-Analysis-Resistant Dual-Field Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture

Jen-Wei Lee, *Student Member, IEEE*, Szu-Chi Chung, *Student Member, IEEE*, Hsie-Chia Chang, *Member, IEEE*, and Chen-Yi Lee, *Member, IEEE*

***Abstract*— Elliptic curve cryptography (ECC) for portable applications is in high demand to ensure secure information exchange over wireless channels. Because of the high computational complexity of ECC functions, dedicated hardware architecture is essential to provide sufficient ECC performance. Besides, crypto-ICs are vulnerable to side-channel information leakage because the private key can be revealed via power-analysis attacks. In this paper, a new heterogeneous dual-processing-element (dual-PE) architecture and a priority-oriented scheduling of right-to-left double-and-add-always EC scalar multiplication (ECSM) with randomized processing technique are proposed to achieve a power-analysis-resistant dual-field ECC (DF-ECC) processor. For this dual-PE design, a memory hierarchy with local memory synchronization scheme is also exploited to improve data bandwidth. Fabricated in a 90-nm CMOS technology, a 0.4-mm^2 160-b DF-ECC chip can achieve 0.34/0.29 ms 11.7/9.3 μJ for one GF(p)/GF(2^m) ECSM. Compared to other related works, our approach is advantageous not only in hardware efficiency but also in protection against power-analysis attacks.**

***Index Terms*— Elliptic curve cryptography (ECC), dual fields, heterogeneous processing-element architecture, parallel computations, power-analysis attacks.**

I. Introduction

PUBLIC-KEY cryptosystem is necessary for secure information exchange in wireless communication applications. In 1978, the RSA modular exponentiation algorithm [1] was presented as the first achievable scheme, but it is currently threatened by the quick factoring attack in cryptanalysis. Elliptic curve cryptography (ECC), specified in IEEE P1363 [2] and FIPS P186-3 [3], can provide the same level of security with shorter key size than the RSA method. Thus, with the use of short and user-friendly key size, the ECC-based encryption engine becomes more attractive in related applications.

To date, several works of the ECC hardware implementation have been published [4]–[13], [30], [31] aiming at speed improvement, but very few designs are suitable for portable devices affected by resource constraints such as system performance, silicon area, and energy supply. To save hardware complexity, single finite field architecture either for prime field GF(p) [6], [7], [30]–[32] or extension binary field GF(2^m) [4], [11], [12], and fixed modulus approach on specific ECs [8], [9], [30] can be used. However, the applications of IEEE P1363 including digital signature are approved for supporting dual-field (DF) functions on arbitrary ECs. Exploiting carry-save adder trees in multipliers is a common technique to integrate DF data path [5], [10], [13], but the limit of integration for distinct arithmetic units still results in large hardware cost, where a 160-b design reported in [5] occupies over 100 000 gates.

In addition to the hardware performance, even though the ECC schemes are secure at cryptanalysis, the private data stored in an unprotected hardware device will be extracted by physical attacks [33]. For ECC hardware implementation, by using the conventional double-and-add (DA) binary method with a primary base point P on ECs, the execution time and intermediate values of elliptic curve scalar multiplication (ECSM) computing the multiple points $KP = P + \cdots + P$ depend on the private key K. Therefore, as presented in [14], the key information can be revealed through the simple power-analysis (SPA) attack by directly interpreting single power measurement and the differential power-analysis (DPA) attack by statistical methods as well.

The double-and-add-always (DAA) algorithm with uniform operations [9] and randomized scalar approach [15] is usually used to avoid SPA attack and DPA attack, respectively, but the high computational overhead leading to significant performance loss is inevitable due to the extra EC point calculation with the enlarged key size. Adopting parallel computations with a homogeneous accelerator [8], [10], [13], [16] is a common technique to enhance throughput. However, in practice, this approach by directly duplicating the arithmetic units has less hardware utilization for various operations. Also, the doubling attack described in [17] is a more powerful one, which can work on SPA- and DPA-resistant designs using left-to-right (LR) ECSM algorithm with less memory storage than the right-to-left (RL) approach.

In this paper, we target at providing a hardware-efficient ECC design solution to support DF functions on arbitrary ECs

Manuscript received March 26, 2012; revised December 4, 2012; accepted December 30, 2012. Date of publication February 8, 2013; date of current version December 20, 2013. This work was supported by the National Science Council (NSC) and Ministry of Economic Affairs (MOEA) of Taiwan, under Grants NSC100-2220-E-009-016, NSC101-2220-E-009-060, and MOEA101-EC-17-A-01-S1-180.

The authors are with the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu 30010, Taiwan (email: jenweilee@gmail.com; phonchi@si2lab.org; hcchang@mail.nctu.edu.tw; cylee@si2lab.org).

Digital Object Identifier 10.1109/TVLSI.2013.2237930

TABLE I
FORMULAS OF EC POINT CALCULATION

Field	ECPA: $P_3 \leftarrow P_1 + P_2$	ECPD: $P_3 \leftarrow 2P_1$
GF(p)	$\lambda = \frac{P_{1y}-P_{2y}}{P_{1x}-P_{2x}}$ $P_{3x} = \lambda^2 - P_{1x} - P_{2x}$ $P_{3y} = \lambda(P_{2x} - P_{3x}) - P_{2y}$	$\lambda = \frac{3P_{1x}^2 + a_p}{2P_{1y}}$ $P_{3x} = \lambda^2 - 2P_{1x}$ $P_{3y} = \lambda(P_{1x} - P_{3x}) - P_{1y}$
GF(2^m)	$\lambda = \frac{P_{1y}+P_{2y}}{P_{1x}+P_{2x}}$ $P_{3x} = \lambda^2 + \lambda + P_{1x} + P_{2x} + a_b$ $P_{3y} = \lambda(P_{2x} + P_{3x}) + P_{3x} + P_{2y}$	$\lambda = P_{1x} + \frac{P_{1y}}{P_{1x}}$ $P_{3x} = \lambda^2 + \lambda + a_b$ $P_{3y} = \lambda(P_{1x} + P_{3x}) + P_{3x} + P_{1y}$

EC point subtraction can be achieved by performing the ECPA with modification of coordinate values such as $(x, y) \rightarrow (x, -y)$ over GF(p) and $(x, y) \rightarrow (x, x + y)$ over GF(2^m).

with power-analysis resistance. For effective implementation of the ECSM, we introduce a single-chip heterogeneous dual-processing-element (dual-PE) architecture deploying various types of PEs with full pipelining and arithmetic unit integration techniques. In addition, based on these specific accelerators for parallel computations, a priority check-in scheme of RL-DAA ECSM with randomized base point technique is exploited to reduce the execution time from a large amount of idling operation and counteract the SPA, DPA, and doubling attacks. Through performance analysis, the proposed design method shows the benefits in hardware utilization against the computational overhead from uniformed processing. Furthermore, a two-level memory hierarchy with local memory synchronization scheme is proposed to reduce the active hardware resource. Compared to previous work using shift-register memory architecture [18], a power saving of 14.2% can be achieved.

The remainder of this paper proceeds as follows. Sections II and III illustrate the basic field arithmetic in ECC functions and the device security, respectively. Section IV presents the proposed operation scheduling for ECSM calculation by parallel computations. Our heterogeneous dual-PE DF-ECC architecture with memory hierarchy is introduced in Section V. The power measurement and experimental results as well as comparisons with previous works are given in Section VI. Finally, Section VII concludes this paper.

II. DF ARITHMETIC FOR ECC FUNCTIONS

As described in IEEE P1363 [2], the standardized EC over GF(p) is $y^2 = x^3 + a_p x + b_p$, where $x, y \in$ GF(p) and $4a_p^3 + 27b_p^2 \neq 0 \pmod p$, and the other one over GF(2^m) is $y^2 + xy = x^3 + a_b x^2 + b_b$ with $x, y \in$ GF(2^m) and $b_b \neq 0$. For the ECC schemes, the most time-critical operation is the ECSM, which consists of serial EC point addition and doubling (ECPA and ECPD). The DF arithmetic of ECPA and ECPD in affine coordinates is summarized in Table I.

In [19], the well-known Montgomery multiplication (MM) algorithm was shown to be an efficient approach to achieve the finite field multiplication in a specific Montgomery domain without high-precision division. For a given m-bit field length, the Montgomery domain is to represent an integer a by $A \equiv a \cdot r \pmod p$, where r is the Montgomery constant and is equal to 2^m over GF(p) and x^m over GF(2^m). In order to perform the division in Montgomery domain, Kaliski [20] first proposed an iterative algorithm that takes average $1.23m$ iterations with two MMs at the last stage. However, the iteration time is still large and the final MMs result in long hardware latency. In [21], through modifying the identities and reducing the iteration time by a high radix method, we proposed a fast Montgomery division (MD), shown in Algorithm 1, which can be performed in average $0.66m$ iterations without any MM operation. Note that the ECSM can be achieved in several coordinate systems, where the computational complexity analysis can be referred to [9] and [22] independently. With our proposed radix-4 MD and the radix-4 MM given in Algorithm 2, the EC point calculation is carried out faster in affine coordinates than that in projective coordinates, where the iteration time ratio MD/MM $\cong$ 1.32.

Algorithm 1 Radix-4 Montgomery Division [21]

Input $A \equiv ar \pmod p$, $B \equiv br \pmod p$, p and m
Output $R = \text{MD}(A, B) \equiv AB^{-1}r \pmod p \equiv ab^{-1}r \pmod p$
1. Let $U = p, V = B, R = 0, S = A, i = 0$
2. **While** $(V > 0)$ **do**
3. $\quad c \equiv U \pmod 4, d \equiv V \pmod 4, t = 2$
4. $\quad$ **If** $i = m - 1$ **then**
$\quad\quad R \equiv 2R \pmod p, S \equiv 2S \pmod p, t = 1$
5. $\quad$ **else if** $c = 0$ **then** $U = \frac{U}{4}, S \equiv 4S \pmod p$
6. $\quad$ **else if** $d = 0$ **then** $V = \frac{V}{4}, R \equiv 4R \pmod p$
7. $\quad$ **else if** $c = d$ **then**
8. $\quad\quad$ **If** $U > V$ **then** $U = \frac{U-V}{4}$,
$\quad\quad R \equiv R - S \pmod p, S \equiv 4S \pmod p$
9. $\quad\quad$ **else** $V = \frac{V-U}{4}$,
$\quad\quad S \equiv S - R \pmod p, R \equiv 4R \pmod p$
10. $\quad$ **else if** $c = 2$ **then**
11. $\quad\quad$ **If** $\frac{U}{2} > V$ **then** $U = \frac{\frac{U}{2}-V}{2}$,
$\quad\quad R \equiv R - 2S \pmod p, S \equiv 4S \pmod p$
12. $\quad\quad$ **else** $V = \frac{V-\frac{U}{2}}{2}, U = \frac{U}{2}$,
$\quad\quad S \equiv 2S - R \pmod p, R \equiv 2R \pmod p$
13. $\quad$ **else if** $d = 2$ **then**
14. $\quad\quad$ **If** $U > \frac{V}{2}$ **then** $U = \frac{U-\frac{V}{2}}{2}, V = \frac{V}{2}$,
$\quad\quad R \equiv 2R - S \pmod p, S \equiv 2S \pmod p$
15. $\quad\quad$ **else** $V = \frac{\frac{V}{2}-U}{2}$,
$\quad\quad S \equiv S - 2R \pmod p, R \equiv 4R \pmod p$
16. $\quad$ **else**
17. $\quad\quad$ **If** $U > V$ **then** $U = \frac{U-V}{2}$,
$\quad\quad R \equiv R - S \pmod p, S \equiv 2S \pmod p, t = 1$
18. $\quad\quad$ **else** $V = \frac{V-U}{2}$,
$\quad\quad S \equiv S - R \pmod p, R \equiv 2R \pmod p, t = 1$
19. $\quad$ **If** $i < m$ **then** $i = i + t$
20. $\quad$ **else** $R \equiv \frac{R}{2^t} \pmod p, S \equiv \frac{S}{2^t} \pmod p$
21. **Return** R

III. POWER-ANALYSIS ATTACKS AND RESISTANCE

Algorithm 3 shows the LR-DA ECSM algorithm. With this approach, since the EC point calculation depends on the hamming weight of the key in Step 4, the SPA attack

Algorithm 2 Radix-4 MM

Input $A \equiv ar \pmod p$, $B \equiv br \pmod p$, p and m
Output $R = \mathrm{MM}(A, B) \equiv ABr^{-1} \pmod p \equiv abr \pmod p$
1. Let $V = (A_{m-1}, A_{m-2}, \ldots, A_0)_2$, $R = 0$, $S = B$
2. **For** i from 0 to $\lceil \frac{m}{2} \rceil - 1$ **do**
3. **If** $m \pmod 2 = 1$ and $i = \lceil \frac{m}{2} \rceil - 1$ **then**
$R \equiv \frac{R+V_0 \cdot S}{2} \pmod p$, $V = \frac{V}{2}$
4. **else**
$R \equiv \frac{R+V_0 \cdot S+V_1 \cdot 2S}{4} \pmod p$, $V = \frac{V}{4}$
5. **Return** R

Algorithm 3 LR-DA ECSM

Input K and P
Output KP
1. Let $Q_0 \leftarrow 0$
2. **For** i from $m-1$ to 0 **do**
3. $Q_0 \leftarrow 2Q_0$
4. **If** $K_i = 1$ **then** $Q_0 \leftarrow Q_0 + P$
5. **Return** Q_0

is a threat to reveal the key value through recording power traces over time. As shown in Algorithm 4, the LR-DAA ECSM performing the uniformed EC point calculation in each iteration can resist the SPA attack [9], but it requires on average 50% ECPA operation overhead. Besides, the DPA attack can still be conducted because of the key-dependent point coordinates in Step 5. To protect this, a randomized base point technique [15] can be applied for eliminating the correlation between point coordinates and key value. At initialization, the primary input point P is masked by adding a randomly selected point M for which $N = KM$. Then the ECSM is achieved by computing $K(P + M) = KP'$ and subtracting N before returning such that $KP' - N = KP$. For each consequent ECSM calculation, the random points M and N are refreshed by performing $M \leftarrow (-1)^{\alpha}2M$ and $N \leftarrow (-1)^{\alpha}2N$ with a random bit α. This randomized base point technique also defeats the fault attacks by injecting a low-order point [34].

As described in [17], the doubling attack using a predecided pair of primary input points P and $2P$ is able to classify the bit value of private key from matching the power segment waveforms of ECPD operations. To formally illustrate the doubling attack on the design using LR-DAA ECSM with randomized base point technique, where $K(2P') = K(2P + 2M)$ is executed after computing KP' with probability 1/2, the jth ECPD operations for input points P' and $2P'$ are given as follows:

$$2(2(\cdots(2(2(2P' + K_{m-2}P') + K_{m-3}P') + K_{m-4}P') + \cdots) + K_{m-(j-1)}P')$$

and

$$2(2(\cdots(2(2(2(2P') + K_{m-2}(2P')) + K_{m-3}(2P')) + K_{m-4}(2P')) + \cdots) + K_{m-(j-1)}(2P'))$$

Algorithm 4 LR-DAA ECSM

Input K and P
Output KP
1. Let $Q_0 \leftarrow 0$, $Q_1 \leftarrow P$
2. **For** i from $m-1$ to 0 **do**
3. $Q_0 \leftarrow 2Q_0$
4. $Q_1 \leftarrow Q_0 + P$
5. $Q_0 \leftarrow Q_{K_i}$
6. **Return** Q_0

Algorithm 5 RL-DAA ECSM

Input K and P
Output KP
1. Let $Q_0 \leftarrow 0$, $Q_1 \leftarrow 0$, $Q_2 \leftarrow P$
2. **For** i from 0 to $m-1$ **do**
3. $Q_1 \leftarrow Q_0 + Q_2$
4. $Q_2 \leftarrow 2Q_2$
5. $Q_0 \leftarrow Q_{K_i}$
6. **Return** Q_0

Input Point	Calculation	K =	1	0	0	1	0	1	1
P'	ECPD	Q_0 =	0	2P'	4P'	8P'	18P'	36P'	74P'
	ECPA	Q_1 =	P'	3P'	5P'	9P'	19P'	37P'	75P'
2P'	ECPD	Q_0 =	0	4P'	8P'	16P'	36P'	72P'	148P'
	ECPA	Q_1 =	2P'	6P'	10P'	18P'	38P'	74P'	150P'

Fig. 1. Example of the doubling attack for the LR-DAA ECSM.

respectively. According to these formulations, if the bit $K_{m-(j-1)}$ is zero, then the $(j-1)$th ECPD for the case of input point $2P'$ is the same as the jth ECPD with input point P'. On the other hand, if the value of $K_{m-(j-1)}$ is non-zero, the ECPD operations are different because of the ECPA calculation. An example of the doubling attack for Algorithm 4 is shown in Fig. 1. As a result, the zero bits and nonzero bits of the key value can be distinguished from collisions and noncollisions by comparing the correlation of ECPD power traces.

The RL-DAA ECSM shown in Algorithm 5 [17] is a countermeasure of doubling attack. Unlike the LR approach, the collision operations definitely exist for all possible key values because the ECPD in Step 4 is independent of the ECPA in Step 3.

IV. Proposed Priority-Oriented Scheduling for RL-DAA ECSM With Parallelism Exploration

Although Algorithm 5 prevents the private key from being revealed by detecting the difference between ECPD operations with specific primary input points, the read-after-write scheduling hazard inherently exists in EC point calculation. The ECPA $Q_{1_i} \leftarrow Q_{0_{i-1}} + Q_{2_{i-1}}$ for the ith iteration in Step 3 can only be processed after finishing the ECPD $Q_{2_{i-1}} \leftarrow 2Q_{2_{i-2}}$ for the previous iteration in Step 4. This operand dependency results in a long latency for idling through parallel computations. For exploring parallelism in ECSM calculation, Algorithm 6 shows the reformulation of Algorithm 5. By using a temporary point Q_T to store the values of point $Q_{2_{i-1}}$ before starting the ith ECPD, the iterative EC point calculation $Q_{2_i} \leftarrow 2Q_{2_{i-1}}$ in Step 4 and $Q_{1_i} \leftarrow Q_{0_{i-1}} + Q_{T_i} =$

Algorithm 6 Modified RL-DAA ECSM

Input K and P
Output KP
1. Let $Q_T \leftarrow 0, Q_0 \leftarrow 0, Q_1 \leftarrow 0, Q_2 \leftarrow P$
2. **For** i from 0 to $m-1$ **do**
3. $\quad Q_T \leftarrow Q_2$
4. $\quad Q_2 \leftarrow 2Q_2$
5. $\quad Q_1 \leftarrow Q_0 + Q_T$
6. $\quad Q_0 \leftarrow Q_{K_i}$
7. **Return** Q_0

Algorithm 7 Proposed Priority-Oriented Scheduling

1. Prioritize tasks:
   MD is *high priority*
   MM is *medium priority*
   ADD and SUB are *low priority*
2. Create ECPD and ECPA to be a thread individually
3. Initialize task and thread counter:
   $u = 1, L = 1$
4. **While** $(L \leq m)$ **do**
5. Get uth task in Lth thread
6. **If** (task priority $<$ *high*) **then**
   Assign task on PE
7. **else**
8. **If** (PE ID is GFAU) **then**
   Assign task on PE
9. **else** $/*$ Interleaved Processing $*/$
   Push task into FIFO, exchange PE ID,
   and then wait until GFAU is available
10. **If** (uth task is the last task) **then**
11. **If** (Lth thread is independent of all $(L+1)$th threads) **then** $u = 1, L = L + 1$
12. **else**
    Wait until all parallel Lth threads are done,
    $u = 1, L = L + 1$
13. **else**
    $u = u + 1$
14. ECSM is done

$Q_{0_{i-1}} + Q_{2_{i-1}}$ in Step 5 can be computed into two parallel threads, where the field operations of EC point calculation are regarded as the tasks.

A design method for accelerating Algorithm 6 by parallel computations is to exploit two duplicated PEs of homogeneous architecture, where each PE specifically performs the ECPD in Step 4 or ECPA in Step 5. With this approach, the overall execution time in each iteration of processing GF(p) and GF(2^m) ECSM is dominated by the ECPD operations. The homogeneous architecture using two identical PEs can outperform the single PE design by nearly two times in speed, but the hardware complexity doubles as well.

The computation time of distinct field operations is different such as $T_{\mathrm{MD}} > T_{\mathrm{MM}} >> T_{\mathrm{ADD}}, T_{\mathrm{SUB}}$, where T_{MD}, T_{MM}, T_{ADD}, and T_{SUB} represent the computation time of MD, multiplication, modular addition, and subtraction, respectively.

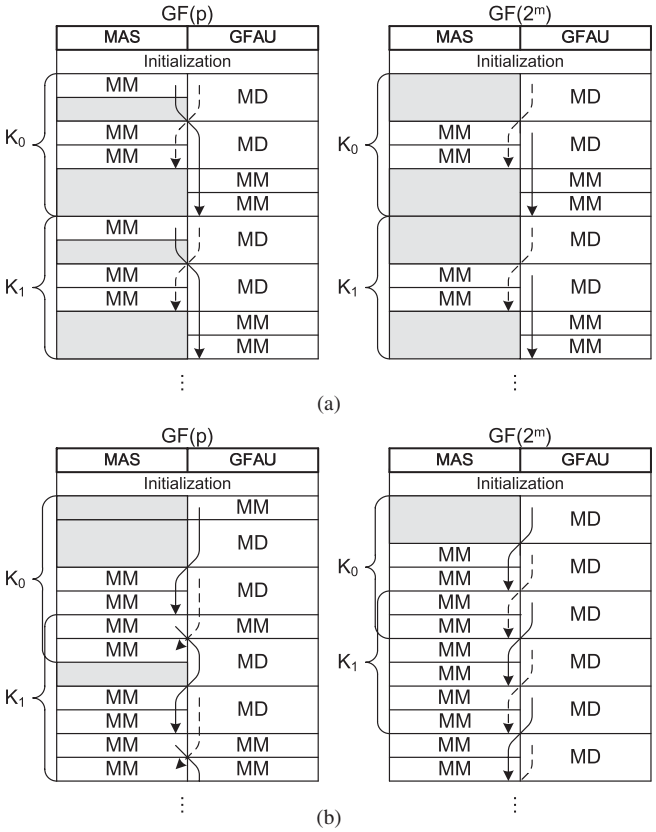


Fig. 2. Priority-oriented scheduling for (a) conventional RL-DAA ECSM and (b) modified RL-DAA ECSM, where the solid line is the ECPD operation flow and the dashed line is the ECPA operation flow.

The PE can be simplified since it is not necessary to process MD all the time. In this paper, we introduce a heterogeneous architecture including a powerful Galois field arithmetic unit (GFAU) and a synergistic multiplier–adder/subtractor (MAS) to speed up the ECSM with lower hardware complexity than that of two-GFAU design using two duplicated GFAU accelerators. The GFAU supports the overall field operations, and its detailed circuit unit design is described in Section V-A.

To further ensure that the PEs are utilized as much as possible, the priority-oriented scheduling which queues higher priority task before lower priority task is exploited. Algorithm 7 is our proposed operation scheduling for the modified RL-DAA ECSM in Algorithm 6, and it has two stages. The first stage in Step 1 is to configure the tasks with higher priority based on larger computation time. At the second stage in a loop of Step 4, the current task is processed as the capable PEs are available. Otherwise, when the current task is pushed into the instruction first-in-first-out (FIFO), it will be issued as the GFAU is available in Step 9. The task and thread counter are refreshed in Steps 10–13 after checking the thread dependence. By this interleaved processing approach, the PEs can cooperate with each other to carry out the ECSM for utilization improvement.

Fig. 2(a) and (b) illustrates the major operation of EC point calculation by Algorithms 5 and 6 with priority-oriented scheduling, respectively. In these figures, the horizontal direction is the hardware behavior, and the vertical direction is the timing. Also, the block in gray signifies the idle execution. As adopting Algorithm 5, even though the last two multiplications

of $(i-1)$th ECPD can be performed by the MAS, the tasks of ith ECPA still have to wait to be issued until generating the coordinates of $2Q_2$ in previous iteration. On the other hand, with Algorithm 6, the GFAU can immediately start calculation as the value of $2Q_{2_{i-1}}$ is stored before ith ECPA. For the average execution time at one bit of key value, the interleaved processing shown in Fig. 2(a) needs $1T_{\mathrm{MD}}+4T_{\mathrm{MM}}$ over GF(p) and $1T_{\mathrm{MD}}+4T_{\mathrm{MM}}$ over GF(2^m). In respect of the case shown in Fig. 2(b), it takes $1T_{\mathrm{MD}}+3T_{\mathrm{MM}}$ over GF(p) and $4T_{\mathrm{MM}}$ over GF(2^m). Therefore, the modified RL-DAA binary method of ECSM calculation with our proposed architecture and operation scheduling has fewer idle operations and more advantages in the hardware utilization than the conventional RL-DAA approach, where the detailed operation flow of Fig. 2(b) is described in Section V-C.

V. VLSI Architecture of Proposed Heterogeneous Dual-PE DF-ECC Processor

Fig. 3 shows the system diagram of our proposed DF-ECC processor with a standard AMBA AHB bus interface. Because of the RL binary method implementation of ECSM, the key value is saved in an n-bit register with little Endian form and scanned one single bit for every round of EC point calculation, where n is the maximum field length. The field operations including MD, MM, ADD, SUB over GF(p), and GF(2^m), required for the ECC schemes such as signature, authentication, and key exchange, are calculated by the GFAU. Also, a cooperative PE which consists of an MAS is designed to accelerate the ECSM by parallel computations. This heterogeneous dual-PE architecture has benefits in hardware efficiency because the MAS substantially reduces the execution cycles without duplicating the GFAU, which has larger hardware complexity than that of MAS. The memory is to store the point coordinates, EC parameters, and intermediate values as the PEs are performing the ECC functions, and the data transitions are manipulated by the DF-ECC control. Because the maximum operand size reaches hundreds of bits in a long length, a block of shared memory, implemented by a single-port SRAM, with hierarchical memory architecture is exploited to improve the hardware cost and even power consumption. Additionally, since the modulus, field length, and operating field signal are invariant after initialization, the circuit logic for saving these values is shared between PEs.

For power-analysis resistance, the SPA and doubling attacks can be counteracted by applying the operation scheduling of RL-DAA ECSM method in Algorithm 6. However, the DPA attack can still be conducted by noticing that the processed points depend on the key value. To achieve the randomized base point scheme against DPA attack mentioned in Section III, the EC points M, N are computed offline and stored in the device memory. Also, they are refreshed after responding KP to system or before next ECSM calculation such that $M \leftarrow (-1)^{\alpha}2M$, $N \leftarrow (-1)^{\alpha}2N$. For the flexibility, the single-bit α is randomly determined by an all-digital random number generator utilizing the cycle-to-cycle time jitter in free-running oscillators [23]. The ECSM is masked online by processing $K(P+M)$, and then it is subtracted by $N=KM$ before returning primary outputs.

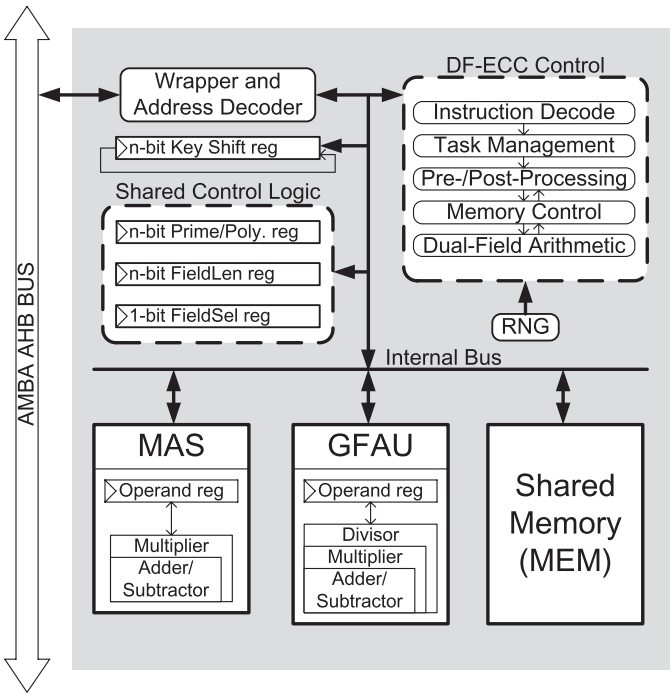


Fig. 3. Overall system block diagram of the heterogeneous dual-PE DF-ECC processor.

The primary inputs of the DF-ECC processor are the user public/private key K, coordinates of base point P, EC parameter a_p or a_b, and protocol instructions. To real-time perform these contents, the instruction decoder, task management, and pre-/post-processing of data domain conversion are combined in our DF-ECC processor. As described in Section II, the Montgomery algorithm requires a Montgomery constant r for data domain conversion because the primary inputs and outputs are in the integer domain, where the pattern-dependent Montgomery constant is usually calculated from host CPU [11]. For system runtime analysis, an Andes 32-b RISC CPU [24] operated at 80 MHz with embedded Linux OS is used to perform the computation of Montgomery constant by language C given in [25]. The experimental results show that it takes 0.64 ms for GF(p) and 0.25 ms for GF(2^m), which cannot be neglected as processing the ECC functions. In this paper, a free precomputation scheme is exploited to avoid this system retardation, and the data domain conversion is instantly carried out during ECSM calculation. The preprocessing stage is to convert the base point coordinates and EC parameter into the Montgomery domain, where it is performed by dividing the constant one with the MD such that $\mathrm{MD}(a,1) \equiv a \cdot 1^{-1} \cdot r \pmod{p} \equiv ar \pmod{p}$ for an integer input a. On the other hand, before returning the calculation results, the coordinates of multiple point KP are converted back into the integer domain at the postprocessing stage, where the MM is used as $\mathrm{MM}(ar,1) \equiv ar \cdot 1 \cdot r^{-1} \pmod{p} \equiv a \pmod{p}$. Another benefit of this approach is that the GFAU can immediately perform the overall function without extra on-chip storage for the Montgomery constant value, while the speed overhead of several modular operations can be ignored for a long execution time of ECSM.

A. Design Method of Processing Elements

Using individual arithmetic units is a widely adopted approach to enhance the hardware speed, though it usually results in low hardware utilization. The major objective of our design approach is to optimize the cost effectiveness by exploiting the following techniques.

1) Full Pipelining Scheme: As the iterative operations shown in Algorithm 1 are performed within one cycle, the critical path is to calculate the results of operands R or S, which consists of the UV comparison with modular operations. Since the results of operands R, S are irrelevant to the results of operands U or V, a full pipelining stage is inserted between the UV and RS data path to reduce the critical path delay. Fig. 4(a) and (b) illustrate the hardware behavior of the pipelining scheme. After initialization, the UV data path is determined at the first cycle. Then the next cycle is to set the values of operands R, S and simultaneously determine the second case of UV comparison. The following cycles can be deduced from this approach until $V = 0$.

2) Programmable Data Path of Modular Reduction With Ladder Selection: To calculate the operands within finite filed set over GF(p) in Algorithms 1 and 2, a low-level parallel architecture with 2s-complement number system is exploited. The arithmetic functions substrated by different multiple modulus are carried out simultaneously, while the correct value is sequentially determined with a ladder selection [21] by checking the signed bit. These multiple modular operations in the iterative calculation can be effectively implemented by using a programmable data path of bit-level architecture, which consists of the carry-save adders with a carry-lookahead adder at the last stage [18].

3) Modular Halving, Quartering by Bitwise Shifting: In Algorithm 1, the halving and quartering of the UV data path can be easily achieved by shifting right one and two bit positions because the least significant one and two bits of intermediate values are definitely zero. However, the least significant bit values of operands R and S are undetermined in the iterative calculation. Here, we use the modulus p to on-the-fly fix the least significant one and two bits of R, S to be zero. To simplify the illustration, the intermediate value of R, S is denoted as X, where the subscript means the bit position in binary representation. For calculating the modular halving operation $X/2 \pmod p$, it is achieved by performing $(X + X_0 \cdot p) >> 1$ since the prime p must be an odd value. For the modular quartering operation $X/4 \pmod p$, it is conducted by performing the following calculation: if $(X_1, X_0) = (0, 0)$, X is shifted right two bit positions; if $(X_1, X_0) = (1, 0)$, $(X-2p) >> 2$ is performed; if $(X_1, X_0) = (1, 1)$ or $(0, 1)$, and there are two subcases. As the least significant two bits of $X - p$ are $(1, 0)$, $(X - 3p) >> 2$ is performed because $-p_1$ is the complement value of $-3p_1$. On the other hand, it is achieved by $(X - p) >> 2$. As a result, the overall modular halving and quartering operations in Algorithms 1 and 2 can be implemented by bitwise shifting with simple logic gates without time-cost modular division.

4) Arithmetic Unit Integration: To map the multiple modular operations in Algorithm 1 into a hardware unit without using distinct circuit components and without a quite complex multiplexer of operand selection, symmetric operations such as $(R - S)/4 \pmod p$ and $(S - R)/4 \pmod p$ can be executed by using the same computational unit with a swap logic circuit. In Algorithm 1, the overall RS data path is classified into two groups: the first group includes Steps 6, 9, 12, 15, and 18, and the second one consists of the remainder. The two operands R and S are switched to each other as the processing group is different from the group in previous cycle. Furthermore, since the EC point calculation is a serial field operation, both of the temporary registers and modular operations can be shared for the operands V, R, S in Algorithms 1 and 2.

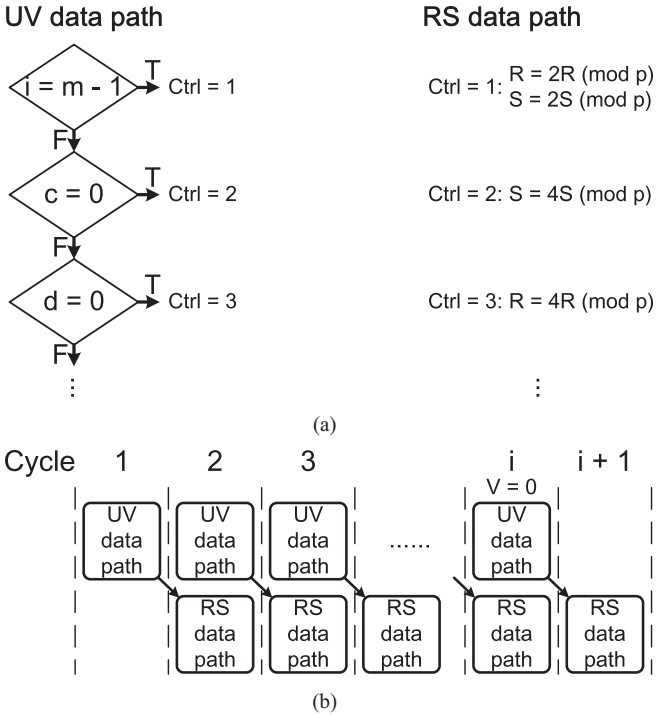


Fig. 4. (a) Separate the data path of UV comparison and RS calculation. (b) Full pipelining scheme of hardware implementation for the previous proposed MD in Algorithm 1.

Fig. 5 shows the detailed architecture of the GFAU. Without pipelining, the delay path is equal to (1) + (2) + (3) + (5) over GF(p) and (1) + (2) + (4) + (5) over GF(2^m). The delay path (1) can be eliminated because of the fully pipelined stage of data path separation, so that the RS select signal is delayed one cycle from the UV select signal. Besides, the swap logic circuit can be implemented by an exclusive-or logic operator to change the input operands of RS data path as the previous and current swap signals have inverse values. After arithmetic processing, the ladder selection is to select out the value belonging to the finite field set. Note that the MAS is implemented by similar design approach with less hardware complexity than GFAU, and the circuit components of MAS are depicted in gray in Fig. 5.

In comparison with the previous works on GF(p_{256}) arithmetic logic unit [16], [26], we also implement our heterogeneous PEs using the identical field length by the same FPGA family. Table II gives the performance results. Because of pipelined and highly integrated architecture, our design has

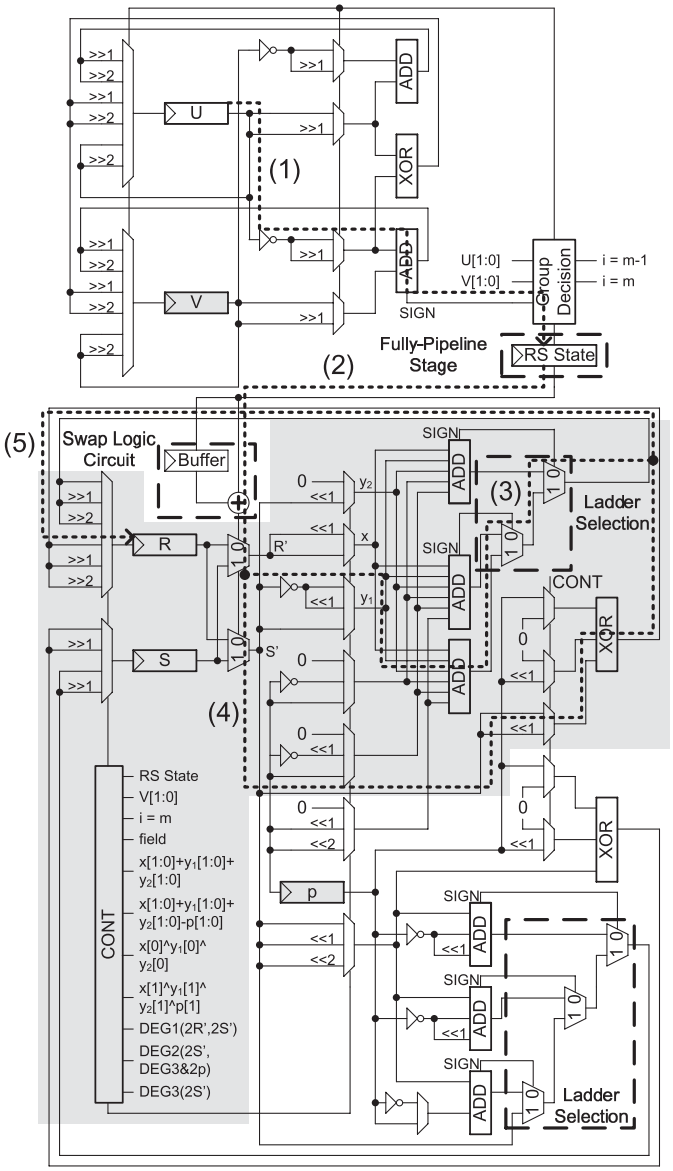


Fig. 5. Overall DF operations integrated into a fully pipelined reconfigurable GFAU.

TABLE II
IMPLEMENTATION RESULTS OF GF(p_{256}) GFAU AND MAS ON XILINX VIRTEX-II FPGA DEVICE WITH COMPARISON

	Area (Slices)	f(MHz)	Multiplication		Division	
			Time (μs/Op.)	AT	Time (μs/Op.)	AT
[26]	5477	14	18.28	1	43.89	1
[16]	5379	34	7.53	0.40	13.55	0.30
Our GFAU	9213	37	3.46	0.29	4.98	0.18
Our MAS	4843	37	3.46	0.13	-	-

AT product = area × time.

benefits in the area–time (AT) product and outperforms others by at least two times in the hardware speed.

B. Memory Hierarchy With Local Memory Synchronization

The memory bandwidth is also a critical factor of system performance for the interleaved processing within various PEs. Therefore, we design a hierarchical memory architecture shown in Fig. 6 with a local memory synchronization scheme

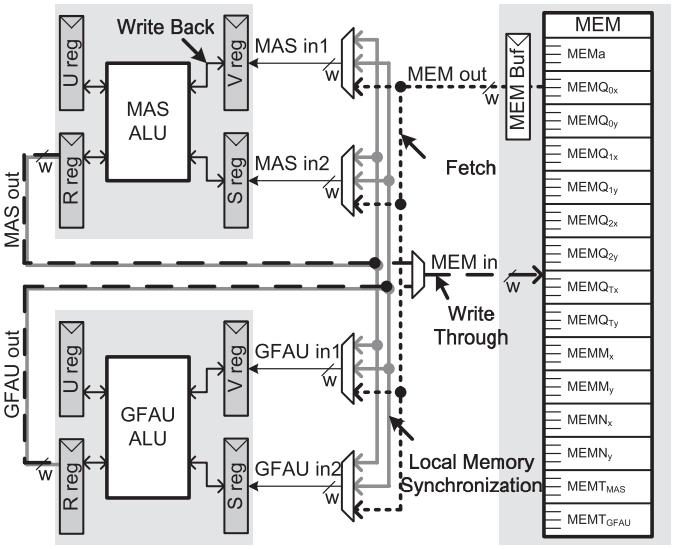


Fig. 6. Two-level memory hierarchy for heterogeneous dual-PE architecture.

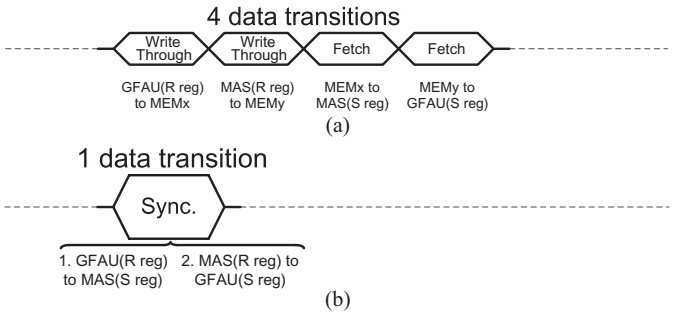


Fig. 7. Example of data access sequences MOV GFAU (R reg) to MAS (S reg) and MOV MAS (R reg) to GFAU (S reg) (a) without and (b) with local memory synchronization scheme. The data transitions through MEM for interleaved processing in (a) can be eliminated in (b).

to reduce the memory access time. Note that a w-bit register buffer is used to avoid the intrinsic latency of reading data from SRAM, where w is the data width of shared memory. For an arbitrary field length m, one data transition between the PEs and MEM needs $T_{\text{MEM}} = \lceil \frac{m}{w} \rceil + 1$ cycles. The on-demand registers, implemented by using the D-type flip-flops, are the local memory for PEs to perform arithmetic without fetching instantly used data from the shared memory every time. To ensure the data consistency, the memory management strategy is as follows.

1) *Write Back:* As the data are predicted to be used in the same PE only for next calculation such as the intermediate values for iterative calculation of MD, MM and ADD, SUB, they are saved into the on-demand registers.
2) *Write Through:* The data are written into both the on-demand registers and shared memory when they are predicted to be used for further calculation, such as the values of EC slope λ and point coordinates (x, y).
3) *Local Memory Synchronization:* As the task for interleaved processing in Algorithm 7 is issued, the data in on-demand registers are exchanged between PEs.

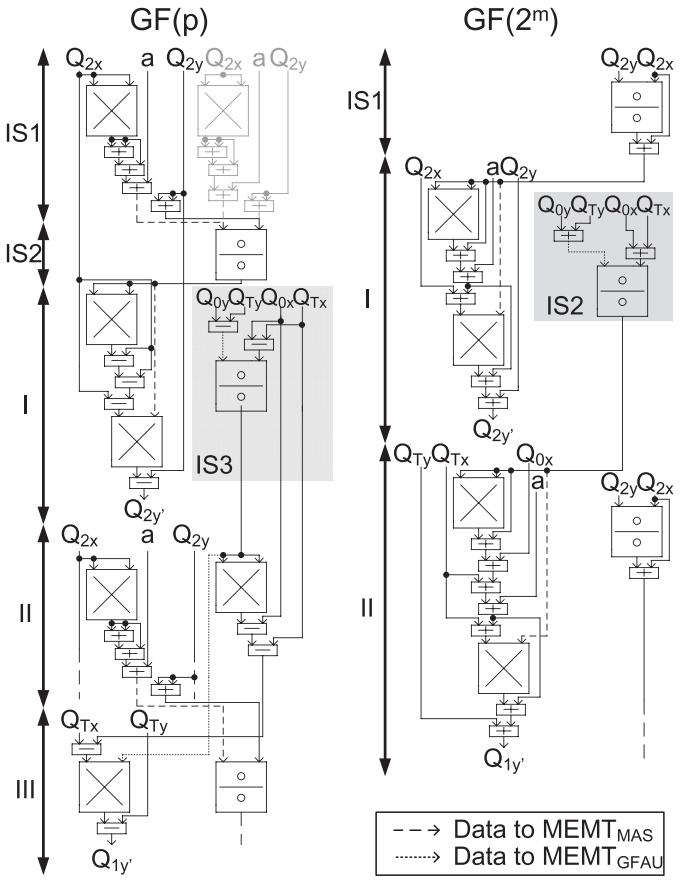


Fig. 8. Detailed data flow for the proposed scheduling of ECSM calculation over DFs.

TABLE III
TIME ANALYSIS OF PROPOSED PRIORITY-ORIENTED SCHEDULING

(a) GF(p)

Operating Stage	Computation Time
Preprocess	$T_{p,\text{PRE}} = 3T_{\text{MD}} + 6T_{\text{MEM}}$
Mask	$T_{p,\text{MK}} = T_{\text{MD}} + 2T_{\text{MM}} + 6T_{\text{SUB}} + 13T_{\text{MEM}}$
IS1	$T_{p,\text{IS1}} = T_{\text{MM}} + 4T_{\text{ADD}} + 6T_{\text{MEM}}$
IS2	$T_{p,\text{IS2}} = T_{\text{MD}} + T_{\text{MEM}}$
IS3	$T_{p,\text{IS3}} = 2T_{\text{MM}} + 4T_{\text{SUB}} + 9T_{\text{MEM}}$
I	$T_{p,\text{S1}} = T_{\text{MEM}} + 2T_{\text{MM}} + 4T_{\text{SUB}} + 8T_{\text{MEM}}$
II	$T_{p,\text{S2}} = T_{\text{MM}} + 4T_{\text{ADD}} + 7T_{\text{MEM}}$
III	$T_{p,\text{S3}} = T_{\text{MEM}} + T_{\text{MD}}$
Unmask	$T_{p,\text{UK}} = T_{\text{MD}} + 2T_{\text{MM}} + 7T_{\text{SUB}} + 15T_{\text{MEM}}$
Post-process	$T_{p,\text{POST}} = 2T_{\text{MM}} + 4T_{\text{MEM}}$

(b) GF(2^m)

Operating Stage	Computation Time
Preprocess	$T_{b,\text{PRE}} = 3T_{\text{MD}} + 6T_{\text{MEM}}$
Mask	$T_{b,\text{MK}} = T_{\text{MD}} + 2T_{\text{MM}} + 9T_{\text{ADD}} + 16T_{\text{MEM}}$
IS1	$T_{b,\text{IS1}} = T_{\text{MD}} + T_{\text{ADD}} + 2T_{\text{MEM}}$
IS2	$T_{b,\text{IS2}} = 2T_{\text{MM}} + 5T_{\text{ADD}} + 9T_{\text{MEM}}$
I	$T_{b,\text{S1}} = T_{\text{MEM}} + 2T_{\text{MM}} + 5T_{\text{ADD}} + 8T_{\text{MEM}}$
II	$T_{b,\text{S2}} = 2T_{\text{MM}} + 7T_{\text{ADD}} + 10T_{\text{MEM}}$
Unmask	$T_{b,\text{UK}} = T_{\text{MD}} + 2T_{\text{MM}} + 10T_{\text{ADD}} + 18T_{\text{MEM}}$
Postprocess	$T_{b,\text{POST}} = 2T_{\text{MM}} + 4T_{\text{MEM}}$

Fig. 7(a) and (b) gives an example to show that the data bandwidth is improved by applying the local memory synchronization scheme.Compared to a shift-register-based memory architecture [18] leading to a large amount of active circuit, the proposed hierarchical memory architecture with local memory synchronization scheme gains an average of 14.2% power reduction.

TABLE IV
IMPLEMENTATION ANALYSIS FOR DIFFERENT DF-ECC DESIGNS

Design Method	Area (mm^2/KGates)	Operating Field	Time (ms/ECSM) @f (MHz)	AT
Single-GFAU DF-ECC with Algorithm 5	0.29/70	GF(p_{160})	0.44@256	1
		GF(2^{160})	0.38@260	1
Two-GFAU DF-ECC with Algorithm 6	0.54/129	GF(p_{160})	0.25@256	1.05
		GF(2^{160})	0.19@260	0.92
Heterogeneous DF-ECC with Algorithm 5	0.39/95	GF(p_{160})	0.39@256	1.20
		GF(2^{160})	0.30@260	1.07
Heterogeneous DF-ECC with Algorithm 6	0.40/96	GF(p_{160})	0.25@256	0.77
		GF(2^{160})	0.22@260	0.78

AT product = gate count × time.

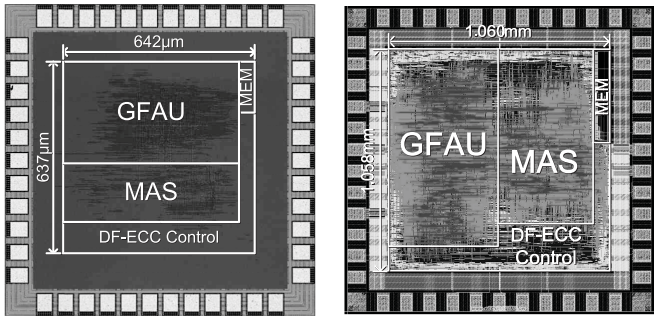


Fig. 9. (a) Die photo of our 160-b DF-ECC processor. (b) Layout view of our 521-b DF-ECC processor.

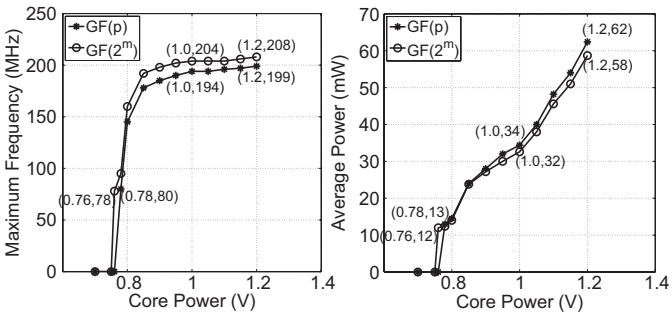
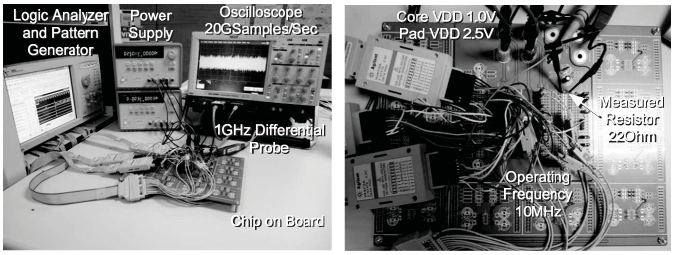


Fig. 10. Operating frequency and power consumption over supply voltage.

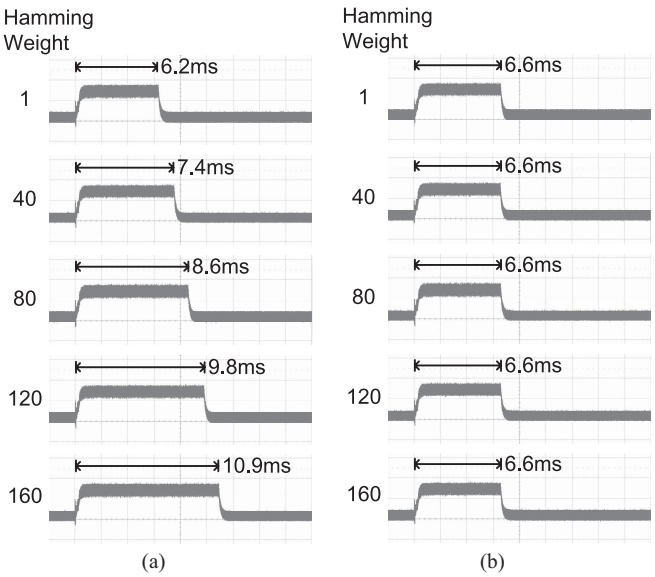
C. Performance Analysis

Fig. 8 shows the explicit scheduling of our proposed parallel computation scheme. To effectively align the data transitions during processing ECSM, the atomic block is split into several stages over GF(p) and GF(2^m). In Algorithm 6, the coordinates of Q_0 are zero until finishing the first iteration including the initial step. Thus the ECPA operation $Q_1 = Q_0 + Q_T$ can be simply achieved by moving the value of Q_T to that of Q_1. Stages IS1, IS2, IS3 over GF(p) and Stages IS1, IS2 over GF(2^m) are the initial stages to process the operations as $Q_0 = 0$. Stages I, II, III over GF(p) and Stages I, II over GF(2^m) are the operating stages between interleaved processing for the iterative ECSM calculation as $Q_0 \neq 0$. In Fig. 8, the computation in Stages IS1, IS2, IS3 over GF(p) and Stages IS1, IS2 over GF(2^m) are similar to that in Stages II, III, I



(a) (b)

Fig. 11. (a) Environment of power measurement. (b) Current flowing through the chip recorded by measuring the voltage drop via a resistor in series with the core power and supply power.



(a) (b)

Fig. 12. SPA attack on the chip using (a) LR-DA and (b) LR-DAA binary method of ECSM, where the power traces are recorded by 50.0 mV/div voltage resolution and 2.0 ms/div time base.

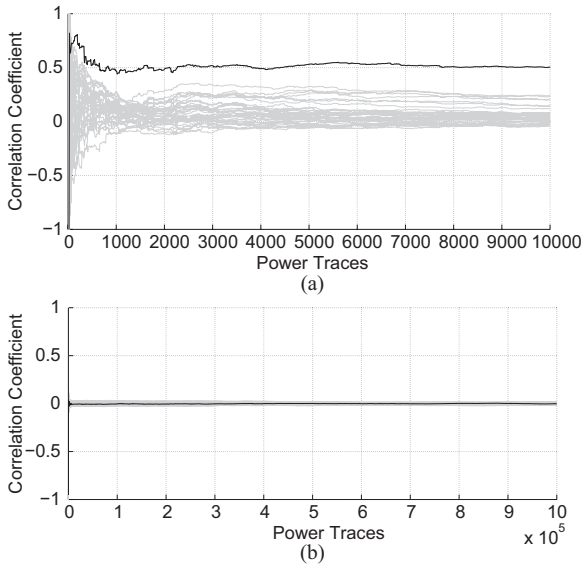
over GF(p) and Stages II, I over GF(2^m) except for disabling the ECPA operations, respectively.

On the basis of the cycle analysis results of MD, MM, ADD, and SUB operations, as well as data transitions, the execution time for the proposed heterogeneous architecture using priority-oriented scheduling can be computed. Table III gives the operation time of distinct operating stages; the execution time of one ECSM over DFs for a valid key length L_K is summarized as follows:

$$\begin{cases} \text{GF}(p) : T_{p,\text{PRE}} + T_{p,\text{MK}} + 2(T_{p,\text{IS1}} + T_{p,\text{IS2}}) \\ \quad + T_{p,\text{IS3}} + (L_K - 1)T_{p,\text{S1}} + (L_K - 2)(T_{p,\text{S2}} + T_{p,\text{S3}}) \\ \quad + T_{p,\text{UK}} + T_{p,\text{POST}} \\ \text{GF}(2^m) : T_{b,\text{PRE}} + T_{b,\text{MK}} + 2T_{b,\text{IS1}} + T_{b,\text{IS2}} \\ \quad + (L_K - 1)T_{b,\text{S1}} + (L_K - 2)T_{b,\text{S2}} \\ \quad + T_{b,\text{UK}} + T_{b,\text{POST}}. \end{cases}$$

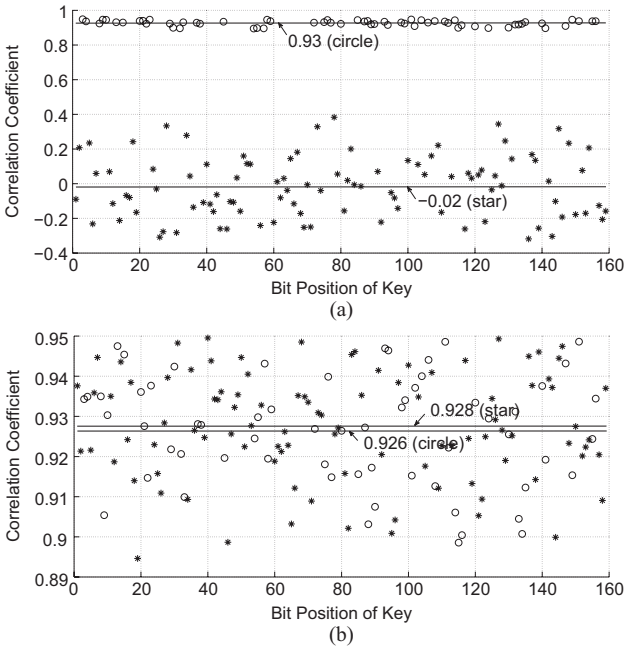
Note that $T_{\text{MM}} = 0.5$ m, $T_{\text{MD}} = 0.66$ m, $T_{\text{ADD}} = T_{\text{SUB}} = 1$, $T_{\text{MEM}} = \lceil \frac{m}{w} \rceil + 1$ with w-bit data width of shared memory. For one 160-bit ECSM, the overhead of the masking and unmasking primary point is 0.80%, and the overhead of the preprocessing and postprocessing is 0.72%.

To compare the different design methods under the consideration of power-analysis resistance, the post-layout simulations



(a) (b)

Fig. 13. Correlation coefficients of the target traces and power model over power traces obtained from the chip (a) without and (b) with randomized base point processing scheme.



(a) (b)

Fig. 14. Correlation coefficients of the power trace segment for ECPD operations with base points P and $2P$. (a) Using LR-DAA ECSM method, the mean of correlation coefficients for zero and nonzero bits is over 0.9 and near 0 due to the key-dependent collisions and noncollisions, respectively. (b) On the contrary, with RL-DAA ECSM method, the mean of correlation coefficients for zero and nonzero bits shown, is nearly equal because the collision operations are generated for all possible key values.

of ECC hardware implementation are given in Table IV. Single-GFAU [21] and two-GFAU designs are the tradeoff between hardware complexity and speed due to the difference between serial and parallel computations. By using a cooperative MAS which has lower hardware complexity than that of GFAU, the heterogeneous architecture moderates the cost from duplicating GFAU; however, parallelism ability is still required to be improved further. Algorithm 6, by reducing the data

TABLE V
COMPARISON AMONG PREVIOUS APPROACHES FOR GF(p)

	Technology	Area (mm^2/KGates)	Field	Field Length	Time (ms/ECSM) @f(MHz)	KCycles	AT	Energy (μJ/ECSM)	ECSM Method	Power-Analysis Resistance
Our Design-DF160 (Measurement@1.0V)	90-nm	0.41/98	Dual	160	0.34@194	66.2	1	11.7	RL-DAA	SPA, DPA, and doubling attacks
TCAS-II'09 [10] (Measurement@1.2V)	130-nm	1.44/169	Dual	160	0.61@121	74.0	3.09 (2.14†)	42.6 (14.2♮)	LR-DAS	-
TVLSI'11 [13] (Measurement@1.2V)	130-nm	1.35/179	Dual	160	0.39@141	54.4	2.09 (1.45†)	31.0 (10.3♮)	LR-DAS	-
Our Design-DF192 (Post-layout@1.0V)	90-nm	0.46/122	Dual	192	0.36@263	94.2	1	24.4	RL-DAA	SPA, DPA, and doubling attacks
RFIDSec'05 [35]* (Post-layout)	90-nm	0.09/23.8	Dual	192	1,300@0.545	677	704.5	39	LR-DAA	SPA and DPA attacks
Our Design-DF521 (Post-layout@1.0V)	90-nm	1.12/313	Dual	160	0.30@220	66.2	1	12	RL-DAA	SPA, DPA, and doubling attacks
				192	0.43@220	94.2	-	26		
				224	0.59@217	127.2	-	39		
				256	0.76@217	165.1	1	54		
				384	1.69@217	366.1	-	143		
				521	3.15@212	668.6	1	292		
ESSCIRC'10 [18] (Measurement@1.0V)	90-nm	0.55/170	Dual	160	1.62@154	249.5	2.93	107	LR-DAA	SPA and DPA attacks
				256	4.40@147	646.8	3.14	297		
				521	19.2@132	2,534	3.31	1,123		
Our Design-P192 (Post-layout@1.0V)	90-nm	0.41/108	GF(p)	192	0.36@263	94.2	1	23.9	RL-DAA	SPA, DPA, and doubling attacks
ISCAS'07 [32] (Post-layout)	130-nm	0.15/23.6	GF(p)	192	2.5@200	502	1.52 (1.05†)	-	LR-DAA	SPA and DPA attacks
Our Design-P256 (Post-layout)	Virtex-II Pro	8,272 CLB Slices	GF(p)	256	4.41@37	165.1	1	-	RL-DAA	SPA, DPA, and doubling attacks
TCAS-I'06 [6] (Post-layout)	Virtex-II Pro	15,755 CLB Slices	GF(p)	256	3.86@39	151.4	1.67	-	LR-DA	-

AT product = gate count (or CLB slices) × time.
Energy = average power × time.
† Normalization factor is 0.69 (90-nm/130-nm).
♮ Normalization factor is 0.33 [(90-nm/130-nm)^2 ×(1.0V/1.2V)^2].
* Support hash function.
LR-DAS: LR-DA/subtract.

hazard in Algorithm 5, has fewer idle operations as it exploits the proposed scheduling in Algorithm 7. As a result, the design using the heterogeneous architecture and the newly introduced priority-oriented scheduling with independent parallel threads for EC point calculation has advantages in hardware efficiency.

VI. POWER MEASUREMENT AND EXPERIMENTAL RESULTS

Our proposed 160-b DF-ECC processor (Design-DF160) is fabricated by UMC 90-nm CMOS 1P9M technology; a photograph of the chip is shown in Fig. 9(a) with 0.41 mm^2 core area. Verified by Agilent 93 000 system on a chip test system with the recommended ECs given in both IEEE P1363 [2] and Certicom SEC2 [27], the measurement results show that the DF-ECC chip using 1.0 V supply power performs one GF(p_{160}) ECSM in 0.34 ms@194 MHz with 11.7 μJ and one GF(2^{160}) ECSM in 0.29 ms@204 MHz with 9.3 μJ. The maximum frequency and power dissipation versus supply voltage are plotted in Fig. 10.

The power-analysis verification environment is shown in Fig. 11(a) and (b). Note that, to evaluate the resistance of various power-analysis attacks, the LR-DA ECSM in Algorithm 3, fixed base point processing scheme, and LR-DAA ECSM in Algorithm 4 are also implemented into this test chip with the external control signal.

Fig. 12(a) and (b) show the power traces for different hamming weights of the key over time obtained from the chip performing LR-DA ECSM and LR-DAA ECSM, respectively. As the chip is processing, it consumes 1.79 mW@10 MHz, which results in a voltage drop above 50 mV across the measured resistor. From these waveforms, the key value in the chip using LR-DA ECSM can be distinguished by visual inspection because the processing time is dependent on the hamming weight of the key. Contrarily, by exploiting the

TABLE VI
COMPARISON AMONG PREVIOUS APPROACHES FOR GF(2^m)

	Technology	Area (mm^2/KGates)	Field	Field Length	Time(ms/ECSM) @f(MHz)	KCycles	AT	Energy (μJ/ECSM)	ECSM Method	Power-Analysis Resistance
Our design-DF160 (Measurement@1.0V)	90-nm	0.41/98	Dual	160	0.29@204	62.5	1	9.3	RL-DAA	SPA, DPA, and doubling attacks
TCAS-II'09 [10] (Measurement@ 1.2 V)	130-nm	1.44/169	Dual	160	0.37@146	54.3	2.20 (1.52†)	30.5 (10.1♮)	LR-DAS	-
TVLSI'11 [13] (Measurement@1.2V)	130-nm	1.35/179	Dual	160	0.27@158	43.0	1.70 (1.18†)	21.6 (7.1♮)	LR-DAS	-
Our design-DF192 (Post-layout@1.0V)	90-nm	0.46/122	Dual	192	0.32@263	84.7	1	18.2	RL-DAA	SPA, DPA, and doubling attacks
RFIDSec'05 [35]* (Post-layout)	90-nm	0.09/23.8	Dual	192	800@0.545	426	487.7	24	LR-DAA	SPA and DPA attacks
Our Design-DF521 (Post-layout@1.0V)	90-nm	1.12/313	Dual	163	0.26@238	62.5	1	14	RL-DAA	SPA, DPA, and doubling attacks
				233	0.52@238	124.3	-	34		
				283	0.76@238	181.3	1	55		
				409	1.58@235	372.5	1	141		
ESSCIRC'10 [18] (Measurement@1.0V)	90-nm	0.55/170	Dual	163	1.15@188	216.2	2.40	76	LR-DAA	SPA and DPA attacks
				283	3.33@182	606.1	2.36	225		
				409	8.20@166	1,361	2.82	480		
Our design-B163 (Postlayout@1.0V)	90-nm	0.24/65	GF(2^m)	163	0.22@277	62.5	1	8.2	RL-DAA	SPA, DPA, and doubling attacks
TC'08 [9] (Synthesis@1.2V)	130-nm	- /12.5	GF(2^m)	163	244@0.001	275.8	213.2 (147.6†)	8.94 (3.0♮)	LR-DAA	SPA attack
MWSCAS'09 [11] (Post-layout@1.8V)	180-nm	2.10/69	GF(2^m)	163	1.89@181	228.1	9.12 (4.56‡)	257 (15.4§)	LR-DA	-
ICITA'05 [29] (Synthesis@3.3V)	350-nm	- /46	GF(2^m)	163	3.05@44	134	9.81 (2.52♭)	-	LR-DAS	-
RFIDSec'06 [36] (Synthesis@3.3V)	350-nm	- /16	GF(2^m)	163	27.9@13.56	376.8	31.22 (8.03♭)	-	LR-DAA	SPA and DPA attacks
Our design-B192 (Postlayout@1.0V)	90-nm	0.32/84.6	GF(2^m)	192	0.32@263	84.7	1	17.1	RL-DAA	SPA, DPA, and doubling attacks
CHES'06 [28] (Synthesis@3.3 V)	350-nm	- /29.4	GF(2^m)	192	118@12	1,416	128.1 (32.95♭)	-	-	-

AT product = gate count × time.

Energy = average power × time.

† Normalization factor is 0.69 (90-nm/130-nm).

‡ Normalization factor is 0.50 (90-nm/180-nm).

♭ Normalization factor is 0.26 (90-nm/350-nm).

♮ Normalization factor is 0.33 [(90-nm/130-nm)^2 ×(1.0V/1.2V)^2].

§ Normalization factor is 0.08 [(90-nm/180-nm)^2 ×(1.0V/1.8V)^2].

* Support hash function.

LR-DAS: LR-DA/subtract.

LR-DAA approach, SPA attack cannot be successful to reveal the key value due to regular processing in fixed time even for different hamming weights of the key.

For the LR-DAA ECSM shown in Algorithm 6, the dependence between the point coordinate value Q_0 and the bit value of the key still exists in each iteration. Thus, with a chosen base point P, the key value can be distinguished by matching the power trace segment of accessing the memory storage for point coordinate Q_0. In Fig. 13(a), the correlation coefficients for all possible hamming distances of the point coordinate Q_0 are plotted over power traces, and that of the correct key hypothesis is plotted in black. In this case, as more than 300 power traces are used, the correlation of the correct key is the highest one among that of all the other key hypotheses, and then the key value can be found easily. However, even after collecting 10^6 power measurements from the chip using the randomized base point technique, the correlation coefficients of correct and incorrect hypotheses shown in Fig. 13(b) cannot be scattered. They are near zero because the processed data are uncorrelated to power model, indicating that there is no information bias of the key value extracted by the DPA attack.

The LR-DAA ECSM and randomized base point technique can effectively resist the SPA attack and DPA attack, respectively. But, as described in Section III, the LR binary method implementation is still threatened by the doubling attack because it generates the collisions of ECPD operations at the zero bits between two power traces with the chosen base points P and $2P$. Fig. 14(a) gives the doubling attack on the chip using LR-DAA ECSM approach, where the correlation coefficients for zero and nonzero bits of the key are drawn in circle and star, respectively. The bit value of the key can be distinguished from a difference of at least 0.5 in the correlation coefficients. However, as the RL-DAA ECSM method is applied, the zero and nonzero bits of the key cannot be revealed because the ECPD operations are independent of the key value, where its doubling attack results are shown in Fig. 14(b).

Based on our proposed programmable design architecture, six additional ECC designs, including the 192-b DF (Design-DF192), 521-b DF (Design-DF521), 192-b GF(p) (Design-P192), 256-b GF(p) (Design-P256), 163-b GF(2^m) (Design-B163), and 192-b GF(2^m) (Design-B192) ECC processors, are also implemented to compare with the previous works; the layout view of Design-DF521 is shown in Fig. 9(b). The chip performance and implementation results in comparison with those of other related ECC hardware implementations over GF(p) and GF(2^m) are summarized in Tables V and VI, respectively. Note that, taking into consideration the scaling effect of fabrication technology and supply voltage, the normalization factor of the AT product and energy can be referred to [37] and [38], respectively. The normalization factor of the AT product is proportional to the ratio of minimum gate length for the transistor; the normalization factor of energy is proportional to the square ratio of minimum gate length for transistor multiplied by the square ratio of supply voltage. By interleaved processing, the ECSM operation without duplicating PEs, our heterogeneous dual-PE ECC processor with arithmetic unit integration outperforms previous works using four identical multiplier architectures [10], [13], separated arithmetic units [6], [11], [29], [32], [36], and a single integrated arithmetic unit [9], [18], [28], [35] in terms of cost effectiveness. Moreover, since an operation scheduling in a key-independent manner with randomized intermediate values is used to protect the chip from power-analysis attacks including SPA, DPA, and doubling attacks, our design supports a higher security level.

VII. Conclusion

A hardware-efficient DF-ECC processor supporting arbitrary field length was presented in this paper. A key-independent operation scheduling with masked intermediate data technique was also exploited to counteract SPA, DPA, and doubling attacks. Both the hardware speed and circuit utilization could be improved by introducing a heterogeneous architecture with fully pipelined PEs, where the data path could be programmed to fulfill user-demanded security requirement. Furthermore, we proposed a local memory synchronization scheme to decrease the data access time for power reduction.

After having fabricated in the UMC 90-nm CMOS process, the proposed 160-b DF-ECC processor with 0.41 mm^2 core area executed one complete ECSM operation including data domain conversion in 0.34 ms over GF(p_{160}) and 0.29 ms over GF(2^{160}). Performance comparison and power measurement showed that our flexible architecture is superior to related ECC designs over DFs in both the cost effectiveness and device security. These benefits demonstrate that our proposed solution is well suited for mobile device applications.

Acknowledgment

The authors would like to thank United Microelectronics Corporation, Hsinchu, Taiwan, for chip fabrication, and National Chip Implement Center, Hsinchu, for providing measurement facilities. The authors would also like to thank Prof. C.-C. Chung for technical support in chip implementation.

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Comm. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] *Standard Specifications or Public-key Cryptography*, IEEE Standard 1363, Jan. 2000.

[3] *Digital Signature Standard*, FIPS Standard P186-3, Jun. 2009.

[4] J. Goodman and A. Chandrakasan, "An energy-efficient reconfigurable public-key cryptography processor," *IEEE J. Solid-State Circuits*, vol. 36, no. 11, pp. 1808–1820, Nov. 2001.

[5] A. Satoh and K. Takano, "A scalable dual-field elliptic curve cryptographic processor," *IEEE Trans. Comput.*, vol. 52, no. 4, pp. 449–460, Apr. 2003.

[6] C. J. McIvor, M. McLoone, and J. V. McCanny, "Hardware elliptic curve cryptographic processor over $GF(p)$," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 9, pp. 1946–1957, Sep. 2006.

[7] G. Chen, G. Bai, and H. Chen, "A high-performance elliptic curve cryptographic processor for general curves over $GF(p)$ based on a systolic arithmetic unit," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 5, pp. 412–416, May 2007.

[8] K. Sakiyama, L. Batina, B. Preneel, and I. Verbauwhede, "Multicore curve-based cryptoprocessor with reconfigurable modular arithmetic logic units over $GF(2^n)$," *IEEE Trans. Comput.*, vol. 56, no. 9, pp. 1269–1282, Sep. 2007.

[9] Y. K. Lee, K. Sakiyama, L. Batina, and I. Verbauwhede, "Elliptic-curve-based security processor for RFID," *IEEE Trans. Comput.*, vol. 57, no. 11, pp. 1514–1527, Nov. 2008.

[10] J.-Y. Lai and C.-T. Huang, "A highly efficient cipher processor for dual-field elliptic curve cryptography," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 5, pp. 394–398, May 2009.

[11] J.-H. Hong and W.-C. Wu, "The design of high performance elliptic curve cryptographic," in *Proc. IEEE Int. Midwest Symp. Circuits Syst.*, Aug. 2009, pp. 527–530.

[12] J.-H. Chen, M.-D. Shieh, and W.-C. Lin, "A high-performance unified-field reconfigurable cryptographic processor," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 8, pp. 1145–1158, Aug. 2010.

[13] J.-Y. Lai and C.-T. Huang, "Energy-adaptive dual-field processor for high-performance elliptic curve cryptographic applications," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 8, pp. 1512–1517, Aug. 2011.

[14] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Int. Cryptol. Conf. Adv. Cryptol.*, 1999, pp. 388–397.

[15] J. S. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," in *Proc. Cryptograph. Hardw. Embedded Syst.*, vol. 1717. 1999, pp. 292–302.

[16] S. Ghosh, D. Mukhopadhyay, and D. Roychowdhury, "Petrel: Power and timing attack resistant elliptic curve scalar multiplier based on programmable GF(p) arithmetic unit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 8, pp. 1798–1812, Aug. 2011.

[17] P.-A. Fouque and F. Valette, "The doubling attack—why upwards is better than downwards," in *Proc. Cryptograph. Hardw. Embedded Syst.*, vol. 2779. 2003, pp. 269–280.

[18] J.-W. Lee, Y.-L. Chen, C.-Y. Tseng, H.-C. Chang, and C.-Y. Lee, "A 521-bit dual-field elliptic curve cryptographic processor with power analysis resistance," in *Proc. Eur. Solid-State Circuits Conf.*, Sep. 2010, pp. 206–209.
[19] P. L. Montgomery, "Modular multiplication without trial division," *Math. Comput.*, vol. 44, no. 170, pp. 519–521, Apr. 1985.
[20] B. S. Kaliski, "The Montgomery inverse and its applications," *IEEE Trans. Comput.*, vol. 44, no. 8, pp. 1064–1065, Aug. 1995.
[21] Y.-L. Chen, J.-W. Lee, P.-C. Liu, H.-C. Chang, and C.-Y. Lee, "A dual-field elliptic curve cryptographic processor with a radix-4 unified division unit," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 713–716.
[22] H. Cohen, A. Miyaji, and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," in *Proc. Adv. Cryptolog.*, vol. 1514. 1998, pp. 51–65.
[23] J. Golic, "New methods for digital generation and postprocessing of random data," *IEEE Trans. Comput.*, vol. 55, no. 10, pp. 1217–1229, Oct. 2006.
[24] *Andes*. (2010) [Online]. Available: http://www.andestech.com/p2-3.htm
[25] M. Rosing, *Implementing Elliptic Curve Cryptography*. Greenwich, CT: Manning Publications Co., 1999.
[26] A. Daly, W. Marnane, T. Kerins, and E. Popovici, "An FPGA implementation of a GF(p) ALU for encryption processors," *Microprocess. Microsyst.*, vol. 28, nos. 5–6, pp. 253–260, 2004.
[27] *SEC 2: Recommended Elliptic Curve Domain Parameters*. (2000, Sep. 20) [Online]. Available: http://www.secg.org/collateral/sec2_final.pdf
[28] M. Koschuch, J. Lechner, A. Weitzer, and J. Großschädl, "Hardware/software co-design of elliptic curve cryptography on an 8051 microcontroller," in *Proc. Cryptograph. Hardw. Embedded Syst.*, vol. 4249. 2006, pp. 430–444.
[29] J. Park, J.-T. Hwang, and Y.-C. Kim, "FPGA and ASIC implementation of ECC processor for security on medical embedded system," in *Proc. IEEE Int. Conf. Inf. Technol. Appl.*, vol. 2. 2005, pp. 547–551.
[30] T. Güneysu and C. Paar, "Ultra high performance ECC over NIST primes on commercial FPGAs," in *Proc. Cryptograph. Hardw. Embedded Syst.*, vol. 5154. 2008, pp. 62–78.
[31] N. Guillermin, "A high speed coprocessor for elliptic curve scalar multiplications over F_p," in *Proc. Cryptograph. Hardw. Embedded Syst.*, vol. 6225. 2010, pp. 48–64.
[32] F. Fürbass and J. Wolkerstorfer, "ECC processor with low die size for RFID applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, May. 2007, pp. 1835–1838.
[33] J. Fan, X. Guo, E. D. Mulder, P. Schaumont, B. Preneel, and I. Verbauwhede, "State-of-the-art of secure ECC implementations: A survey on known side-channel attacks and countermeasures," in *Proc. IEEE Int. Symp. Hardw.-Oriented Security Trust*, Jun. 2010, pp. 76–87.
[34] J. Fan, B. Gierlichs, and F. Vercauteren, "To infinity and beyond: Combined attack on ECC using points of low order," in *Proc. Cryptograph. Hardw. Embedded Syst.*, vol. 6917. 2011, pp. 143–159.
[35] J. Wolkerstorfer, "Is elliptic-curve cryptography suitable to secure RFID tags?" in *Proc. Workshop RFID Light-Weight Cryptograph.*, Aug. 2005, pp. 1–13.
[36] S. S. Kumar and C. Paar, "Are standards compliant elliptic curve cryptosystems feasible on RFID?" in *Proc. Workshop RFID Security*, Jul. 2006, pp. 1–19.
[37] H.-Y. Hsu, A.-Y. Wu, and J.-C. Yeo, "Area-efficient VLSI design of Reed-Solomon decoder for 10GBase-LX4 optical communication systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 43, no. 4, pp. 1019–1027, Nov. 2006.
[38] C.-C. Wong and H.-C. Chang, "High-efficiency processing schedule for parallel turbo decoders using QPP interleaver," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 6, pp. 1412–1420, Jun. 2011.

Jen-Wei Lee (S'12) received the B.S. degree in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2007, where he is currently pursuing the Ph.D. degree in electronics engineering.

His current research interests include cryptographic arithmetic, VLSI design of crypto-ICs, and SoC platform for security applications.

Szu-Chi Chung (S'12) received the B.S. degree in EECS Undergraduate Honors Program from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2011, where he is currently pursuing the Ph.D. degree in electronics engineering.

His current research interests include VLSI implementation of security systems.

Hsie-Chia Chang (S'01–M'03) received the B.S., M.S., and Ph.D. degrees from National Chiao Tung University, Hsinchu, Taiwan, in 1995, 1997, and 2002, respectively, all in electronics engineering.

He was with OSP/DE1 in MediaTek Corporation, from 2002 to 2003, working in decoding architectures for Combo single chip. In February 2003, he joined the Faculty of the Electronics Engineering Department, National Chiao Tung University, where he has been a Professor since August 2010. His research interests include algorithms and VLSI architectures in signal processing, especially for error control codes and crypto-systems. Recently, he has also committed himself to designing high code-rate ECC schemes for flash memory and multi-Gb/s chip implementations for wireless communications.

Dr. Chang was the recipient of the Outstanding Youth Electrical Engineer Award from Chinese Institute of Electrical Engineering in 2010 and the Outstanding Youth Researcher Award from Taiwan IC Design Society in 2011. He served as the Associate Editor of the IEEE Transactions on Circuits and Systems I: Regular papers since 2012. He also served as a Technique Program Committee (TPC) member for IEEE A-SSCC 2011 and 2012.

Chen-Yi Lee (S'89–M'90) received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 1982, and the M.S. and Ph.D. degrees from Katholieke University Leuven, Leuven, Belgium, in 1986 and 1990, respectively, all in electrical engineering.

He was with IMEC/VSDM from 1986 to 1990, working in architecture synthesis for DSP. In February 1991, he joined the Faculty of the Electronics Engineering Department, National Chiao Tung University, where he is currently a Professor. He is also active in various aspects of short-range wireless communications, system-on-chip design technology, very low power designs, and multimedia signal processing. He has authored or co-authored more than 200 journal/conference papers in his areas of expertise and holds more than 25 ROC/USA patents. His current research interests include VLSI algorithms and architectures for high-throughput and energy-efficient DSP applications.

Dr. Lee served as the Director (2000–2003) of Chip Implementation Center (CIC), an organization for IC design promotion in Taiwan. He was the former IEEE CAS Taipei Chapter Chair (2000–2001), the SIP task leader (2003–2005) of National SoC Research Program, and the microelectronics program coordinator (2003–2005) of Engineering Division under National Science Council of Taiwan.