VLSI DESIGN OF DYNAMIC LOGIC BASED SUCCESSIVE FFM

K. Jansikaramala1, K. Susmitha2, P. Sandhya2, O.B. Sivakumari2, N. Rajeshkumar2
Department of Electronics and Communication Engineering
Annamacharya Institute of Technology and Sciences, Tirupati - 517520
jansikaramala@gmail.com1, susmithakapireddy495@gmail.com2, pinnapedasandhya484@gmail.com2, obsivakumari488@gmail.com2, nunna4urajesh@gmail.com2

Abstract- This paper, a high-speed power-efficient VLSI design of a finite field multiplier in GF(2m) is presented. The proposed design has a serial-in parallel-out Structural design and performs the multiplication operation using a reordered normal basis. The main construction block of the multiplier in dynamic logic to reduce the critical path delay. Reduction in dynamic power consumption is achieved by limiting the contention current between the transistor and pull-down network at the beginning of the estimation phase by employ a control circuit. The semicustom layout of the multiplier was realized in 65-nm CMOS technology. The proposed design methodology can also be used in the design of similar FFM possessing regular Structural designs.

Index Terms—Dynamic logic, elliptic curve cryptography (ECC), finite field arithmetic, reordered normal basis (RNB), serial-in parallel-out (SIPO) finite field multiplier.

I. INTRODUCTION

Efficient computations of finite field arithmetic are largely important in cryptographic applications where field operations are significantly used, namely, elliptic curve cryptography (ECC). The binary extension field GF(2m) is a closed set of 2m elements, meaning that arithmetic operations over the field elements are conducted without leaving the set. Each element of a finite field can be expressed by a bit sequence of length m. A field can be thought of as a vector space spanned by a vector set of m linearly independent elements, called a basis. Choosing the basis by which field elements are represented plays an important role in the efficient Design of finite field operations. A number of bases over finite fields have been proposed in the literature, among which polynomial basis (PB) and normal basis (NB) are primarily used in practice. Although the use of PB is most common in software designs, NB offers a virtually cost-free squaring operation performed by a single cyclic shift over the field element’s coordinates, thus making it the better choice for hardware design. Among the set of finite field arithmetic operations, the efficient design of field multiplication is of utmost importance, as field operations of greater complexity (e.g., exponentiation and division) can be performed by the consecutive use of field multiplication.

It is proven that an NB exists for every field in GF(2m). In general, the multiplication operation in NB can be modelled as a matrix-vector multiplication, where a matrix multiplication is required to be performed for each of the product coordinates. The hardware complexity of the multiplication operation is directly affected by the number of nonzero elements inside the multiplication matrix. This number is referred to as the complexity of NB and is denoted by CN. For a given m, CN varies between the two extreme values of 2m+1 and m2 and is minimal in the case of two subclasses of NB, known as type I and II optimal NBs (ONBs). RNB can effectively simplify the multiplication operation by defining it as a closed-form formula rather than a matrix operation. A fully parallel Structural design would be a natural choice for applications in which speed is of great priority. Additionally, by cryptographic standards, the use of high-order fields (m > 160) is recommended to ensure a high level of security. However, considering the fact that a parallel Structural design has an area complexity of O(m2), a large m will result in a big, power greedy design not suitable for resource constrained applications. Contrastingly, a fully serial (sequential) multiplier has an area complexity of O(m), resulting in a significantly smaller structure. Despite their smaller size, sequential multipliers require m clock cycles to complete a full multiplication operation as compared to only one cycle in the case of a fully parallel Structural design. Thus, it is desirable to reduce the multiplication delay of a sequential multiplier to compensate for this shortcoming. In this paper, we present an optimized VLSI design of a serial-in parallel-out (SIPO) RNB multiplier in GF(2m). The regularity of this Structural design has been previously exploited to construct a high-speed custom-layout multiplier by implementing the main building block of the Structural design in dynamic logic. However, this performance improvement in terms of critical path delay is obtained at the cost of a significant increase in power consumption. To improve the performance of the multiplier by employing a custom-designed dynamic logic circuit that effectively reduces the power dissipation of the dynamic circuit. It is shown that the new design significantly increases the maximum operating frequency compared to its equivalent static CMOS realization, as well as successfully reduces the power consumption to a comparable.
II. EXISTING SYSTEM

Finite field computation is of a great importance because of its wide range of applications in error control coding, coding theory and especially cryptography. Because of the ever-growing applications of public-key cryptography in resource-constrained environments, its power-efficient design has recently become a necessity. Public-key protocols based on elliptic curve cryptography (ECC) rely on a hierarchy of operations such as scalar multiplication which, in turn, depends on elliptic curve group operations e.g. point addition and point doubling. At the base of this hierarchy are fundamental finite field arithmetic operations: finite field addition and finite field multiplication. Finite field multiplication plays a key role in field computation since more complicated operations such as exponentiation and inversion can be carried out with consecutive use of multiplication. An important factor that directly affects the efficiency of a multiplication operation is choice of the basis by which field elements are represented.

A number of bases over finite fields have been proposed in literature and are used in practice, including polynomial basis, normal basis, dual basis and redundant basis. Among them, polynomial basis is widely used in software design due to the fact that it requires fewer machine instructions in general. On the other hand, normal basis offers a very low cost squaring operation performed by a single circular shift operation over the coordinates of field elements, thus making it suitable for hardware design. This advantage has been widely exploited to accelerate the inversion operation by performing a series of field squaring and field multiplication operations based on Fermat’s Little Theorem (FLT). In normal basis, multiplication operation is generally modelled as a matrix-vector multiplication where a matrix multiplication is required to be carried out to generate each element of the product coordinates. It is evident that the computational complexity of multiplication operations depends on the number of nonzero elements inside the multiplication matrix. This quantity is referred to as the complexity of normal basis and is denoted by CN. It has been shown that CN is a function of field size m and selected irreducible polynomial and can vary between a lower bound of 2m + 1 and a higher bound of m2. For two subclasses of normal basis known as type I and II Optimal Normal Basis (ONB) the complexity of normal basis is minimal, i.e. 2m+ 1. Reordered Normal Basis (RNB) is a permutation of type-II ONB.
III. PROPOSED SYSTEM

A. Design of the Multiplier’s Main Building Block in Dynamic Logic

As can be seen in Fig 3, the xor-and-xor module contains the critical path of the SIPO Structural design. This path is made of the two XOR gates in addition to the AND gate inside the xor-and-xor-module. In an attempt to reduce the multiplication delay, the critical path of the multiplier in dynamic logic. Although using dynamic logic for implementing the main building block of the multiplier can effectively increase the maximum operating frequency, this technique has a deteriorating effect on the power dissipation. The increase in power consumption stems from a higher internal switching activity which is an inherent characteristic of dynamic logic circuits. Consequently, the resulting design would consume much more dynamic power compared to its static CMOS counterpart. In order to alleviate the negative effects incurred by using dynamic logic based designs over static CMOS, several techniques have been proposed in the past few years, e.g. high-speed dynamic, XOR-based dynamic, conditional-keeper dynamic, single-phase dynamic and current comparison-based dynamic. Primarily, the focus of the existing techniques is on design strategies which compensate for leakage current in deep sub-micron technologies, narrower noise margins, contention delay at estimation phase and the transistor stacking effect.

Therefore, these techniques would be better suited for high fan-in circuits in which the Pull-Down Network (PDN) contains a large number of parallel paths to the ground, such as high fan-in multiplexers, comparators and more general OR-like cells. Furthermore, the relatively large number of transistors required to implement these techniques compared to the total number of transistors used in the design of the small xor-and-xor-module imposes significant power and area overheads. Such techniques are not applicable to the multiplier in discussion. In this work, the power dissipation problem is tackled by reducing the contention current drawn at the very beginning of the estimation phase.

Depending on the value of the input signals, contention may occur between the Pull-Up Network (PUN) and the Pull-Down Network (PDN) of a dynamic circuit during the estimation phase. This contention, though short in time, forms a conducting path from VDD, across PUN and PDN, to ground causing high amplitude current spikes. The basic idea is to limit the contention current by utilizing a new conditional keeper to compensate for the power overhead caused by the higher switching activity of the circuit. Fig. 3 shows a schematic of the circuit designed to implement XOR-AND-XOR function in dynamic logic. This circuit is responsible to realize an XOR Operation between two different coordinates of B, followed by an AND applied to the result and one of the A coordinates. Finally, this is combined with another XOR that, when paired with a flip-flop, forms an accumulation unit. In terms of the variables used, this circuit realizes logic function \((b_1 \_ b_2) : a\). The static pull-up network is merely composed of a single pMOS transistor charging the dynamic node Q to VDD during the pre charge phase. The pull-down network, on the other hand, consists of 12 transistors (N4-N15) which discharge the dynamic node at the presence of appropriate combinations of the input values. The PDN is connected to a footer transistor, N16, which reduces the leakage current due to the stacking effect and opens a path to the ground during the estimation phase. Transistors P2 and N2 generate a control signal to an nMOS keeper depending on the voltage of the dynamic node and the logic state of the clock signal. Transistors P1 and N1 form the output inverting stage, providing the required current to drive the output flip-flop. In the presented schematic, the input signals are referred to as B1, B2, A and C. The proposed dynamic circuit operates in two phases as follows: During the pre recharge phase, pull-up transistor P0 steadily charges the dynamic node.
If the dynamic node is initially in a low state, node C is quickly charged to VDD by P2, which turns on the keeper transistor to speed up the pre-charging process. The voltage of the dynamic node rises until it reaches a certain level, at which time the output switches to a low state, causing P2 to discharge node C and then turn off the keeper transistor.

Therefore, at the end of the pre-charge phase, the dynamic node is fully charged and the keeper is held off to avoid negative impacts on delay and power consumption at the beginning of the next phase. At the beginning of the estimation phase, the clock signal switches to a high state, keeping the pull-up transistor turned off. At this moment, two different scenarios could occur depending on the logic values of the input signals.

In the first scenario, a conducting path is formed from the dynamic node to the ground, discharging the dynamic node through the PDN network. In this case, when the dynamic voltage falls below VDD - Vth,N2, the source and drain junctions of transistor N3 are reversed and the accumulated charge on node C is fully discharged through N3. This prevents the keeper transistor from being turned on. In the second scenario, the dynamic node is evaluated to a high state. N2 is turned on in the case that the leakage current reduces the voltage of the dynamic node. The behavior of the circuit shown in Figure is explained in more detail in our recent work.

**IV. SIMULATION RESULTS**

The simulations were performed in Cadence’s Analog Environment using Spectre simulator to measure the power consumption and the maximum operating frequency of the circuit. To ensure the correct functionality of the circuit, a pre-simulation stage was required in which the test data set was generated. To do so, the functional behavior of the multiplier was also modeled in MATLAB. Then, a large array of random 233-bit paired vectors were created and fed into the MATLAB code to generate a set of golden product coordinates. Input pairs and their corresponding outputs were stored in two separate files. During the analog simulation, a Verilog-A module read the input files and fed an input pair into the multiplier for each multiplication operation.

These outputs were later verified by comparing them against the golden set created by the MATLAB code. The simulation result showed that the circuit was correctly functional up to a clock rate of 3.84 GHz.

The power consumption of the multiplier was measured to be 13.01 mW/GHz averaged over 100 consecutive multiplication operation. As previously emphasized in Section 6.1, the main objective of this work is to compare the performance of the proposed design with that of a static CMOS design to demonstrate that the new dynamic logic circuit can further reduce the multiplication delay of the multiplier while preserving the total power consumption. To achieve a fair and accurate comparison, we also designed the layout of the static CMOS multiplier in the same 65nm CMOS process using standard cells from TSMC’s libraries.

![Figure 4: The proposed layout for a 233-bit sequential RNB multiplier designed in dynamic logic.](image)

The layout of the static design was constructed based on the same structure shown in Fig. 5.

![Figure 5: Block diagram of a full 233-bit RNB multiplier](image)

The final layout of the multiplier is presented in Fig. 6. Note that the Load module was designed in static CMOS and then was incorporated in the layout to provide the same functionality as its counterpart in dynamic logic. To ensure consistency in all of the measurements, the same set of random inputs was applied to the static multiplier during the simulations conducted. This realization has a maximum operating frequency of 2.94 GHz and requires 158.44ns to finish a single multiplication operation. Including the power rings, the size of the layout is 153m, 71_m, equal to an area of 10; 863_m2. The
required area is reduced to 9; 574m² when not considering the outer rings.

V. CONCLUSION

The VLSI design of a 233-bit SIPO finite field multiplier was presented. The field size of 233 is currently recommended by the NIST for embedded security applications using ECC. The proposed design is very regular, possessing a repeating pattern of a single building block designed in dynamic logic, which can be readily scaled to any arbitrary size multiplier by cascading the appropriate number of blocks. In an attempt to alleviate the high-power dissipation of the dynamic circuit stemming from higher internal switching activities, the original design of this building block was modified to reduce the contention current drawn at the very beginning of the estimation phase. The post place-and-route simulations showed the correct functionality of the design up to a clock range of 3.85 GHz, achieving a much higher operating speed while consuming marginally less power compared to the static CMOS counterpart. The same design methodology can be utilized to improve the operating speed of other similar regular Structural designs without compromising power consumption.

REFERENCES