Efficient High-Throughput Architectures for High-Speed Parallel Scramblers

JianWei Chen[1], Hongchin Lin[1], Yun-Ching Tang[1,2]
[1]*Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan*
[2]*Department of Electronics Engineering, Hsiuping Institute of Technology, Taichung, Taiwan*
hclin@nchu.edu.tw

Abstract — **A generalized algorithm with efficient architectures for high-speed parallel scramblers with reduced registers is proposed. The algorithm can be applied to any scrambler polynomials with three terms to achieve small numbers of registers and fan-outs. The critical paths only have one register and one XOR gate, which are merged into a dynamic differential circuit for implementation. The results show that more than 50% chip area can be reduced in comparison with literatures, and the power dissipation is only 3.7mW at 1.6GHz with 16 parallel outputs, which is equivalent to 25.6Gbps, using TSMC 0.18 μm CMOS technology.**

I. INTRODUCTION

In communications, there are various types of noises. Single coding technology may not achieve ideal error-correcting capability. In order to improve the ability, the Reed-Solomon code and the LDPC code can be used at the same time. Besides, scramblers and interleavers are usually used to prevent random and burst errors. The scrambler plays the role to stabilize and enhance the quality of transmission. However, the serial operation with single bit output may not be feasible for the future high speed transmission, especially for optical communications. Therefore, a high-throughput parallel scramble is required.

The first parallel scrambler was applied to optical communications [1]. It generates parallel outputs by calculating the polynomials with a lot of XOR operations with numerous fan-outs of each register. Even though the number of registers is smaller, the circuit is more complex, more delay and power consumption. Another parallel scrambler reported in recent years [2] utilized the look-ahead method with more registers. The critical path was limited to one XOR and one register. However, for some generating polynomials, the algorithm requires many registers, which will increase chip area and power consumption.

Various scrambler polynomials have been adopted. Figure 1 shows the synchronous frame scrambler in the IEEE 802.11a/g/n [3][4][5], which adopt the modulo-2 polynomial $x^7 + x^4 + 1$. The generated pattern is a sequence with maximum length of 127. In the 802.3ae [6], for physical coding sublayer (PCS), the polynomial is $x^{58} + x^{39} + 1$. For WAN interface sublayer (WIS), it is $x^7 + x^6 + 1$. In the IEEE 1394b [7], the polynomial is $x^{11} + x^9 + 1$. The purpose is to disperse the input signals equally. For example, in the worst situation, the signals contain consecutive zeros or ones. After they are scrambled, the ratio of the numbers of zero to one is about 50.5%. After the scrambled data are received, the same polynomial is used to descramble the data, and the original signals are recovered.

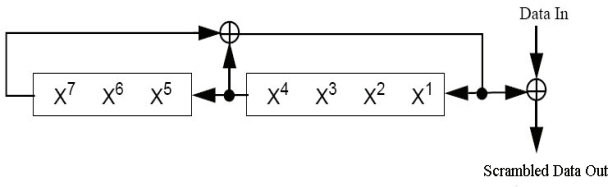

Figure 1. Synchronous frame scrambler in IEEE 802.11a/g/n

II. ARCHITECTURE OF PARALLEL SCRAMBLER

To develop the general parallel scrambler algorithm for the aforementioned various scramblers, the generating polynomial *P(x)* has three terms as given in Eq. (1), where *A* is the number of shift registers in the conventional serial scrambler and *B* is another integer with *A* > *B*.

$$P(x) = x^A + x^B + 1 \tag{1}$$

For convenience in the following derivation, *D* is defined as the difference between *A* and *B*. If q_k is the (*k*+1)'th generated bit in the serial scrambler, *Q(j)* given below means the *M* parallel output sequence in the *j*'th cycle of parallel scrambler.

$$Q(j) = \{q_{M\times j}, q_{M\times j+1}, \ldots, q_{M\times j+M-2}, q_{M\times j+M-1}\} \tag{2}$$

For example of $P(x) = x^7 + x^4 + 1$, the initial value $\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$ are stored in the registers $\{x_7, x_6, x_5, x_4, x_3, x_2, x_1\}$, respectively. Since the generating function depends on generating polynomial *P(x)*, the output bits q_i for $i \geq 7$ can be determined using the following methods.

The authors would like to thank the Chip Implementation Center (CIC) of the National Applied Research Laboratories (NARL) of Taiwan for chip fabrication and the National Science Council of Taiwan for financially supporting this research under Contract no. NSC 97-2221-E-005-092.

A. Looking ahead

The next bit can be determined using looking ahead skill [1], which figures out the previous value in registers. For example, the first bit generated by *P(x)* is the function of $x_7 + x_4$, the next is $x_6 + x_3$, and then $x_5 + x_2$.

B. Substitution

When the current function includes x_1, the looking ahead skill needs to be modified. Thus, we can replace x_1 by $x_7 + x_4$. For example, if the current function is $x_4 + x_1$, the next becomes $x_3 + x_7 + x_4$ using looking ahead and substitution.

C. Merging

For XOR operation, any two identical terms can be merged into none. For instance, if the current function is $x_5 + x_2 + x_1$, the next function will become $x_4 + x_1 + x_7 + x_4$. Here, x_4 appears twice, so the function could be merged into $x_7 + x_1$.

According to the parameters *M, A, B,* and $D = A - B$ for any parallel scramblers, the following design parameters can be determined.

(i) q_k: For given k, h is the minimum integer to satisfy the following condition.

$$k < A \cdot 2^h \tag{3}$$

Then,

$$q_k = q_{k-A\cdot 2^{(h-1)}} + q_{k-B\cdot 2^{(h-1)}} \tag{4}$$

where k must be greater than $A-1$.

(ii) Minimum number (R) of registers for M parallel output bits: The parameter g is the minimum integer to make the following relation valid.

$$M \le B \cdot 2^g \tag{5}$$

If the sequence Q from q_0 to q_k, in which $k = R+M-1$, is divided into two parts, the first half $\{q_0, q_1, \ldots, q_{R-1}\}$ requires R registers to store the data bits, and the second half $\{q_R, q_{R+1}, \ldots, q_{R+M-1}\}$ represents M parallel output bits. However, $q_{k-B\cdot 2^{(g-1)}}$ is the condition that limits the number of registers. For the term q_{R-1} in the registers, by setting $h=g$ in Eq. (3), substituting the upper limit of $k = A \cdot 2^g - 1$ into $q_{k-B\cdot 2^{(g-1)}}$ yields $q_{(A+D)\cdot 2^{(g-1)}-1}$, which is equal to q_{R-1}. Therefore, the minimum register number (R) is

$$R = (A + D) \cdot 2^{(g-1)} \tag{6}$$

For the term q_{R+M-1} in the parallel output bits, h can be set as $g+1$, using Eq. (4), so we have

$$q_{R+M-1} = q_{R+M-1-A\cdot 2^g} + q_{R+M-1-B\cdot 2^g} \tag{7}$$

Because the register of the second term on the right side of the above equation must exist, $R + M - 1 - B \cdot 2^g \le R - 1$, which proves Eq. (5).

(iii) Number of fan-outs of each register: After M is given and R is determined by Eq. (5), the sequence q_R to q_{R+M-1} is divided into two groups, q_R to q_C and q_{C+1} to q_{R+M-1}, in which the subscript C of q_C is the upper limit $A \cdot 2^g - 1$. Making use of Eqs. (4) and (6) for q_R and q_C gives

$$q_R = q_{(A+D)\cdot 2^{(g-1)}} = q_{(A+D)\cdot 2^{(g-1)} - A\cdot 2^{(g-1)}} + q_{(A+D)\cdot 2^{(g-1)} - B\cdot 2^{(g-1)}}$$

and $q_C = q_{A\cdot 2^g - 1} = q_{A\cdot 2^g - 1 - A\cdot 2^{(g-1)}} + q_{A\cdot 2^g - 1 - B\cdot 2^{(g-1)}}$. For any subscript k in q_k, it will be limited within the ranges of $(A+D)\cdot 2^{(g-1)} - A\cdot 2^{(g-1)} \le k \le A\cdot 2^g - 1 - A\cdot 2^{(g-1)}$ and $(A+D)\cdot 2^{(g-1)} - B\cdot 2^{(g-1)} \le k \le A\cdot 2^g - 1 - B\cdot 2^{(g-1)}$.

After some manipulations, the following two relations are obtained.

$$D \cdot 2^{(g-1)} \le k < A \cdot 2^{(g-1)} \tag{8.1}$$

$$D \cdot 2^g \le k < (A + D) \cdot 2^{(g-1)} \tag{8.2}$$

Similarly, q_{C+1} and q_{R+M-1} are expressed as

$$q_{C+1} = q_{A\cdot 2^g} = q_{A\cdot 2^g - A\cdot 2^{(g-1)}} + q_{A\cdot 2^g - B\cdot 2^{(g-1)}} \text{ and}$$

$$q_{R+M-1} = q_{(A+D)\cdot 2^{(g-1)} + M - 1}$$

$$= q_{(A+D)\cdot 2^{(g-1)} + M - 1 - A\cdot 2^{(g-1)}} + q_{(A+D)\cdot 2^{(g-1)} + M - 1 - B\cdot 2^{(g-1)}}$$

Another two relations can be derived as

$$0 \le k < M - B \cdot 2^{(g-1)} \tag{8.3}$$

$$D \cdot 2^g \le k < M + D \cdot 2^g - B \cdot 2^{(g-1)} \tag{8.4}$$

If k complies with one, two, three, or all of these four relations Eqs. (8.1) to (8.4), register q_k will have one, two, three, or four fan-outs, respectively. In other words, the possible maximum fan-outs will be four if these four relations are all satisfied.

To verify the proposed algorithm, assume the polynomial is $x^7 + x^4 + 1$ with $M=16$ parallel outputs. When $k=23$, using Eqs. (3) and (4) gives $h=2$ and $q_{23}=q_9+q_{15}$. Similarly, when $k=31$, $h=3$ and $q_{31}=q_3+q_{15}$ can be obtained. For both cases, the number of register $R = (7+3)\cdot 2^1 = 20$ with $g=2$. To estimate the fan-outs of register q_k, k has to be less than R. When $k=11$, Eq. (8.1) gives $3\cdot 2^1 \le 11 < 7\cdot 2^1$. The other three relations are not satisfied, so the fan-out of q_{11} is one. When $k=17$, since Eqs. (8.2) and (8.4) are valid, the fan-outs of q_{17} are two. Table I lists examples of different polynomials with M=16. The number of registers, R, is determined firstly. The k of output q_k is larger or equal to R, while the k in fan-out of register q_k is less than R.

Table I Examples to find R, output q_k and fan-out q_k for M=16

Polynomial	Output q_k		R	Fan-out q_k	
x^7+x^4+1	k=23	q_9+q_{15}	20	k=11	1
	k=31	q_3+q_{15}		k=17	2
x^7+x^6+1	k=23	q_9+q_{11}	16	k=11	2
	k=31	q_3+q_7		k=15	1
x^9+x^7+1	k=23	q_5+q_9	22	k=11	2
	k=43	q_7+q_{15}		k=17	2

For some polynomial like x^7+x^6+1, the proposed algorithm has the identical architecture reported in Ref. [2]. However, for the polynomial x^7+x^4+1, the proposed approach utilizes

less registers. Table II gives the gate-level comparison of the proposed architecture and Ref. [2] for 16 parallel outputs. Both methods have the same critical path of one register plus two-input XOR operation. The proposed architecture has less registers. Even though its register may have one more fan-out, the influence to delay will not be significant.

Table II Comparison of architectures for M=16

	Proposed	Ref. [2]
Polynomial	$x^7 + x^4 + 1$	
Number of registers	20	28
Critical path	One register with 2-input XOR	One register with 2-input XOR
Max. fan-outs	3	2

III. Circuit Design

To design the parallel scrambler using the proposed algorithm, only two logic blocks are required. One is a register. Another one is a register with 2-inout XOR. Due to the symmetric operations of these two logic blocks, the differential logic [8][9] is appropriate. Here, the modified differential dynamic logic gates are proposed to enhance speed and reduce power consumption.

Figures 2 and 3 show the differential logic gates for a register and a register with 2-input XOR, respectively. *A* and *B* are the inputs, while *OUT* is the output. There are three phases, *CLK1*, *CLK2* and *CLK12*, to make the gates work. The clock patterns are illustrated in Figure 4. One clock cycle is divided into 6 small time slots t_1 to t_6. *CLK1* and *CLK2* are two non-overlapped clocks in order to latch the data at the outputs and have more clock skew tolerance. Note that t_3 and t_6 are much shorter than the others, since their functions are to avoid signals directly passing from the inputs to the outputs.

The operation starts from t_1. Since *CLK1* is high to turn on the two NMOS switches with their gates tied to *CLK1*, the signals will be passed to the two PMOS cross coupled nodes. At t_2, *CLK12* goes to low to make the PMOS switch with its gate tied to *CLK12* pulling the two cross coupled nodes to V_{DD}. The time t_3 is a "buffer" time as mentioned above. During t_4, *CLK2* becomes high to activate the tri-state inverters. The data will be transferred to the outputs and latched. Next, *CLK12* is low to turn off the PMOS switch at t_5, which could be short for preparation to receive the next data. Finally, t_6 is also short, since all the clocked controlled transistors are off and ready for the next data bit.

IV. Simulation and Implementation Results

To demonstrate the complete circuit of parallel scrambler, the two symbols for the functional circuit blocks in Figures 2 and 3 are sketched in Figure 5. The labels R and X-R in the boxes correspond to Figures 2 and 3, respectively. Here, the complements of input and output signals are not included in the symbols. Figure 6 is the proposed parallel scrambler for $x^7 + x^4 + 1$ with $M=16$, where P_0 to P_{15} are the output nodes. It was designed and is being fabricated using TSMC 0.18μm CMOS technology with power consumption of 3.7mW at 16GHz in area of 95μm×70μm as shown in Figure 7.

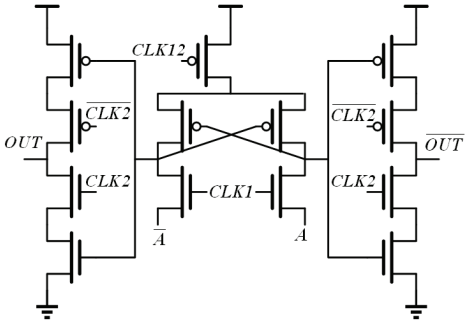

Figure 2. The differential circuit schematic of one register

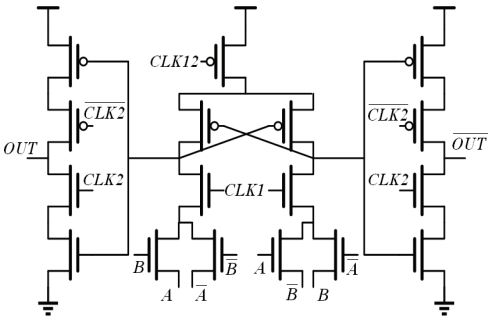

Figure 3. The differential circuit schematic of one register with 2-input XOR operation

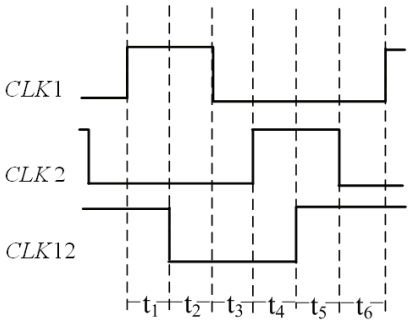

Figure 4. The clock patterns of CLK1, CLK2, and CLK12

The chip-level comparison of the proposed parallel scrambler and Ref. [2] is listed in Table III. Although the polynomial functions are different, the proposed scrambler requires 4 more registers, the area is still much smaller. In terms of throughput/area, the proposed circuit is more efficient. The reasons may be the proposed logic is area-efficient and the double-edge triggered registers with higher parasitic effects were utilized in Ref. [2].

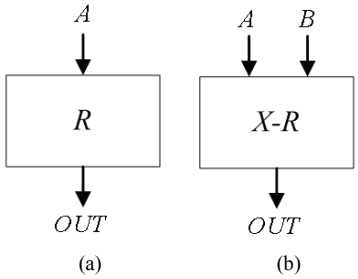

Figure 5. The symbols of the two logic gates (a) a register (b) a register with 2-input XOR

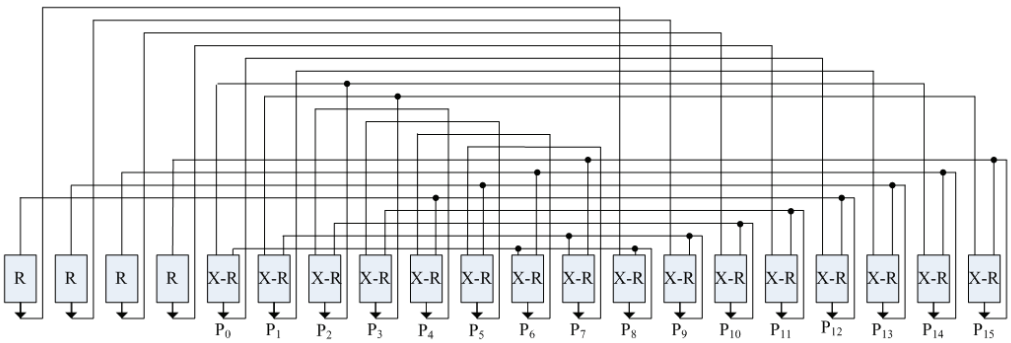

Figure 6. The parallel scrambler of $x^7 + x^4 + 1$ with $M = 16$ based on the proposed algorithm

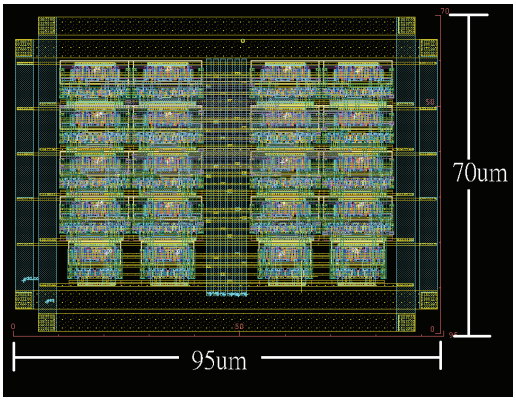

Figure 7. Layout of the proposed parallel scrambler

Table III Performance comparison of different architectures after implementation

Architecture	Proposed	Ref. [2]
Technology	CMOS 0.18 μm	CMOS 0.18 μm
Function	$x^7 + x^4 + 1$	$x^7 + x^6 + 1$
No. of parallel outputs	16	16
Number of register	20	16
Max. fan-outs	3	3
Frequency (GHz)	1.6	1.25 (double-edge)
Throughput (Gbps)	25.6	40
Area (μm^2)	95×70	150×90
Throughput/Area (Mbps/μm^2)	3.85	2.96

V. CONCLUSIONS

The generalized parallel scrambler algorithm and architecture is proposed for less registers and fan-outs, as well as the shortest critical path for high-speed operation. The specially designed dynamic registers were used to meet requirement of high speed, low power and small chip area. The post layout simulation gives power dissipation of 3.7mW at 1.6GHz with throughput of 25.6Gbps in area of 95μm×70μm using TSMC 0.18μm CMOS technology.

REFERENCES

[1] Srini W. Seetharam, Gary J. Minden, Joseph B. Evans, "A Parallel SONET Scrambler/Descrambler Architecture," in Proc. IEEE ISCAS'93, May 1993, vol. 3, pp. 2011-2014.

[2] C.-H. Lin, C.-N. Chen, Y.-J. Wang, J.-Y. Hsiao, and S.-J. Jou, "Parallel Scrambler for High-Speed Applications," IEEE transactions on circuits and system—II: express briefs, Vol.53, No.7, pp. 558-562, July 2006.

[3] IEEE 802.11a, IEEE standard for information technology Telecommunication and information exchange between systems local and metropolitan area networks Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band, 1999.

[4] IEEE 802.11g, IEEE standard for information technology Telecommunication and information exchange between systems local and metropolitan area networks Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Further Higher Data Rate Extension in the 2.4 GHz Band, 2003.

[5] IEEE P802.11n, Draft standard for information technology Telecommunication and information exchange between systems local and metropolitan area networks Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput, 2009.

[6] IEEE 802.3ae, IEEE standard for Information technology Telecommunications and information exchange between systems local and metropolitan area networks Specific requirements. Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2002.

[7] IEEE 1394b, IEEE standard for high-performance serial bus. Amendment 2, 2002.

[8] S. Y. Choe, G. A. Rigby, and G. R. Hellestrand, "Half-rail differential logic," ISSCC Digest of Technical Papers, pp. 420-421, 1997.

[9] H. Lin, Y.-F. Chen and H.-C. She, "A Low-Power 3-Phase Half Rail Pass-Gate Differential Logic," *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 148-151, May 6-9, 2001.