

Comparison of hardware implementation and power consumption of low-power multiple output linear feedback shift register

Shilesh Malliyoor¹ and Chao You²

^{1,2}Department of Electrical and Computer Engineering, North Dakota State University, Fargo, US- 58102
 Email: ¹shilesh.malliyoor@ndsu.edu, ²chao.you@ndsu.edu

Abstract:

A linear feedback shift register has a variety of applications. Various low-power architectures have been proposed. This paper compares two low-power multiple output architectures in terms of the hardware implementation and power consumption. A way to overcome the race around condition in these architectures, which will improve the architecture, is proposed. The polynomials are implemented by using 0.13 μm BiCMOS technology provided by IBM. This paper will also show that the improved Katti's architecture, described in this paper is more power efficient than the improved Lowy's architecture. In specific applications, such as E0 stream cipher, the efficiency can be as high as 83% for certain polynomials.

Section 1 Introduction

With advancements in large scale integration, millions of transistors can be placed on a single chip for implementation of complex circuitry. As a result of placing so many transistors in such a small space, major problems of heat dissipation and power consumption have come into the picture. Research has been conducted to solve these problems. Solutions have been proposed to decrease the supply voltage, switching frequency, and capacitance of transistors [1]. A linear feedback shift register (LFSR) is used in a variety of applications such as Built-in self test (BIST) [2], cryptography, error correction code, and in the field of communication for generating pseudo-noise sequences. Due to the versatile nature of LFSR, various low-power architectures have been proposed. This paper compares power consumption and hardware implementation of two low-power LFSR architectures, and provides a solution to the race around condition in these architectures.

A LFSR is a shift-register where the output bit is an XOR function of some input bits. The initial value of the LFSR is called the seed. The outputs that influence the inputs are called taps. A LFSR is represented as a polynomial mod 2. The coefficients of the polynomial are either 1s or 0s. For example, if the taps are at the 2nd, 3rd, 4th and 5th bits, the polynomial is $1+x^2+x^3+x^4+x^5$ as shown in Figure 1. The structure shown is called a serial architecture. If a polynomial is primitive, the corresponding LFSR can produce 2^n-1 distinct patterns. The implementation shown in Figure 1 is a Type I structure where the XORs are external from shift registers.

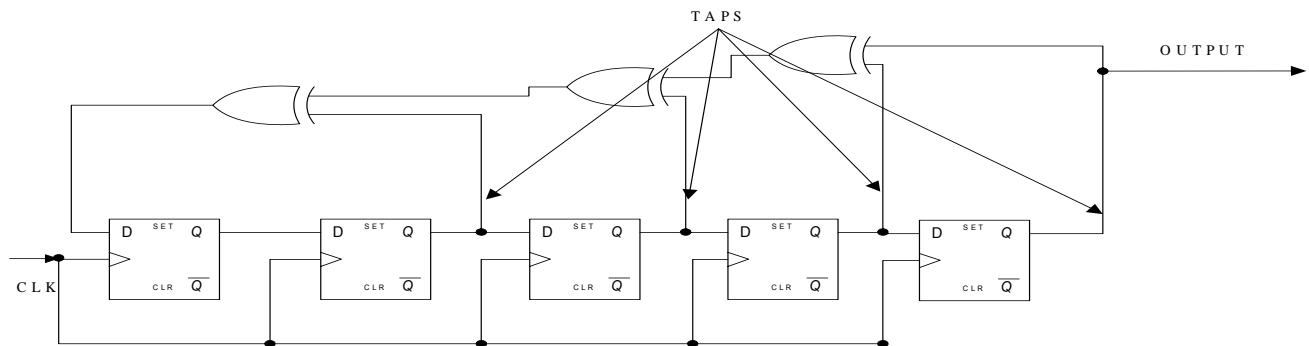


Figure 1 Type I LFSR structure for $1+x^2+x^3+x^4+x^5$.

In the architecture shown above, at each clock cycle, the output bit of each flip-flop is shifted sequentially to the input of the next flip-flop. The output bits of the flip-flops are also used for XOR functions in the feedback loop. All the flip-flops change their states and are active at each clock cycle resulting in high power consumption. Lowy proposed a parallel architecture of LFSR [3]. In Lowy's architecture, because only one flip-flop is active in every clock cycle, the amount of power dissipated by flip-flops is reduced. However, the switch minimization, resulting in only one flip-flop changing state, makes the circuitry complicated. Another disadvantage of Lowy's architecture is its inability to produce full-length distinct random patterns [4]. As only one flip-flop changes its state at every clock cycle, the primitive polynomial is changed into a non-primitive polynomial. The changing of the polynomial leads to a low percentage of distinct patterns generation, which is as low as 45% [4].

Hamid proposed a new form of polynomial [4], the format being of the form $1+x^{n/2}+x^n$. The proposed polynomial reduces the number of switches required as well as a 3% increase in number of distinct patterns generated. Control signals, which control the switching network, are implemented with a Johnson counter and some control logics. The area required to generate control signals is large, and a new scheme for clock generation was proposed by Huang [5]. Multiple polynomial LFSR (MP-LFSR) has gained importance in BIST. MP-LFSR provides flexibility for selection of primitive polynomial [6], [7], [8]. Low-power multiple output parallel architecture structures have been proposed by Lowy [3], and by Katti [9]. A multiple output parallel architecture for a polynomial of the form $1 + x^{k_1} + x^{k_2} + x^{k_3} + \dots + x^N$ can be used to obtain k_1 outputs in a single clock cycle, which will further reduce the power consumption. This paper compares the hardware implementation and the power consumption of these architectures.

The paper is organized as follows: Section 2 explains in detail the architectures proposed in [3] and [9]; Section 3 compares and presents the drawbacks of these two architectures. A solution to overcome one of these drawbacks is also proposed. Section 4 presents a comparison of various polynomials especially with respect to an application such as the E0 cipher, which is a stream cipher. Stream ciphers have an internal state and operate serially by generating a stream of pseudo-random key bits, the keystream [10]. E0 is a standard for Bluetooth encryption. Section 5 concludes this paper.

Section 2 Background

The best known low power multiple output architecture has been proposed by Lowy [3], and Katti [9]. The following section describes these architectures in detail.

2.1 Lowy's Architecture:

The architecture proposed by Lowy is as shown in Figure 2, which considers $1+x^2+x^5$ polynomial for obtaining multiple output in a single clock cycle. The control signal T_i ($i = 1, 2 \dots N$), which is used to connect the shift register to the output tap, is a sequentially occurring waveform. The control signal is high for only $i \bmod N$ clock cycle, where N is the length of the LFSR as shown in Figure 2. The operations (5, 2) and (4, 1) are performed at T_1 , where (x, y) denotes XOR operation of x and y . The value of the XOR operation is the output of the LFSR, obtained at A and B respectively. The outputs A and B are feedback and stored in flip-flop 5 and flip-flop 4 respectively at T_2 cycle. At T_2 , operations (3, 5) and (2, 4) are performed. These values are stored in flip-flop 3 and flip-flop 4 respectively. The overall operations can be summarized as shown in Table I. With the multiple output architecture, a polynomial, of the form $1 + x^{k_1} + x^{k_2} + x^{k_3} + \dots + x^N$ can generate k_1 outputs in a single clock cycle. In the case of $1+x^2+x^5$, 2 output can be obtained in a single clock cycle. In order to produce all the outputs, 5 clock cycles are required.

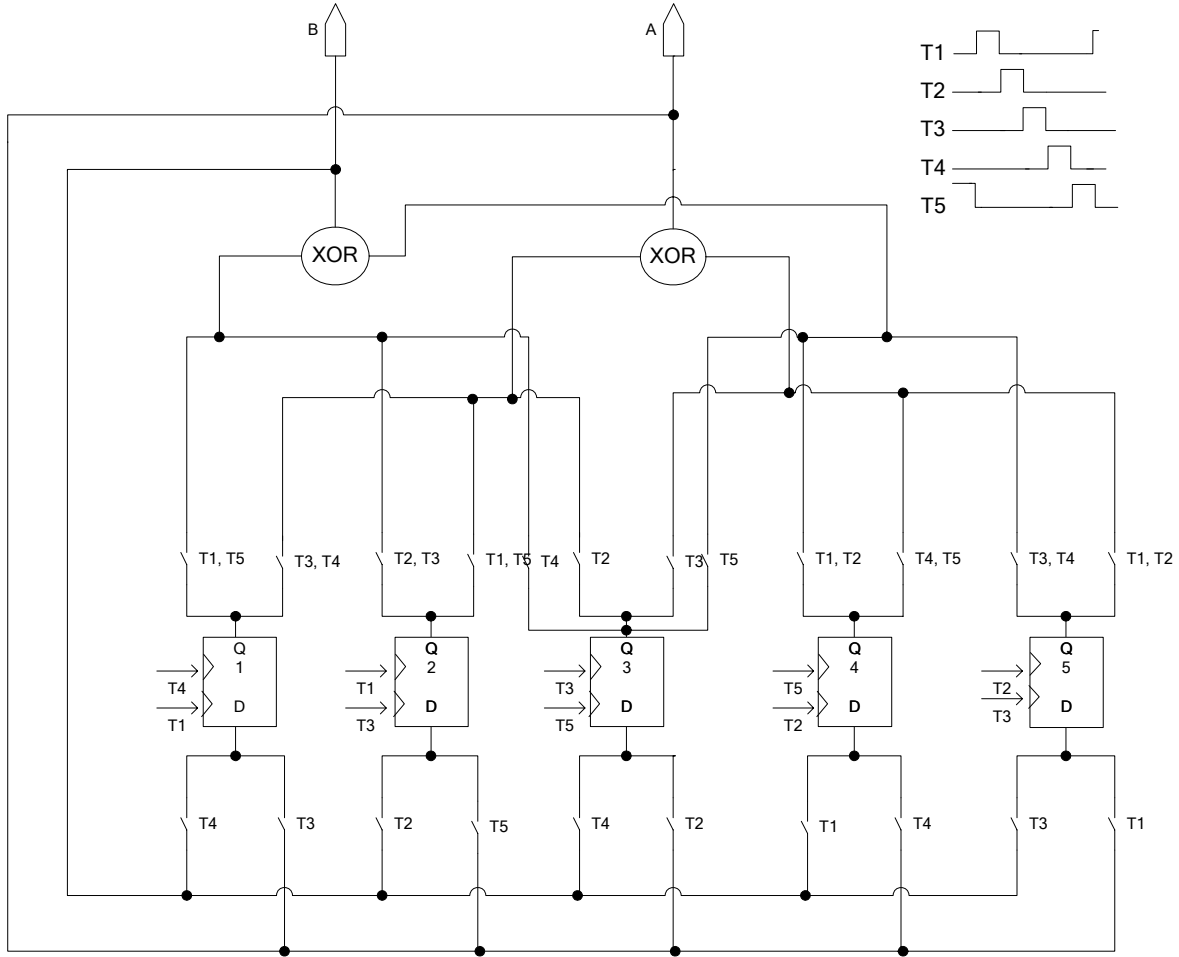


Figure 2 Structure of Lowy's Architecture for Multiple Outputs LFSR with $1+x^2+x^5$

Table I
Operation of Lowy's Architecture for Multiple Output LFSR with $1+x^2+x^5$

	T1	T2	T3	T4	T5
Output A	(5, 2)	(5, 3)	(3, 1)	(4, 1)	(4, 2)
Output B	(4, 1)	(4, 2)	(5, 2)	(5, 3)	(3, 1)
Flip-flop Updated	A → 2 B → 1	A → 5 B → 4	A → 3 B → 2	A → 1 B → 5	A → 4 B → 3

2.2 Katti's Architecture:

The same polynomial, $1+x^2+x^5$, implemented in the architecture as proposed by Katti is as shown in Figure 3. At T1, operations (5, 2) and (4, 1) are performed. Values of these operations are stored in flip-flop 5 and flip-flop 4 respectively at T2. At T2, operations (5, 3) and (4, 2) are performed. The result of the XOR operation is stored in flip-flop 3 and flip-flop 2 respectively at T3. At T3, only one operation (3, 1) is performed, and the result is stored in flip-flop 1 at T1. The operation can be summarized as shown in Table III. In this architecture, the number of control signals required are $\lceil N/k_1 \rceil$, where N is the length of LFSR and k_1 is the number of simultaneous outputs. In this example, 3 control signals are needed, and all the outputs are produced in 3 clock cycle.

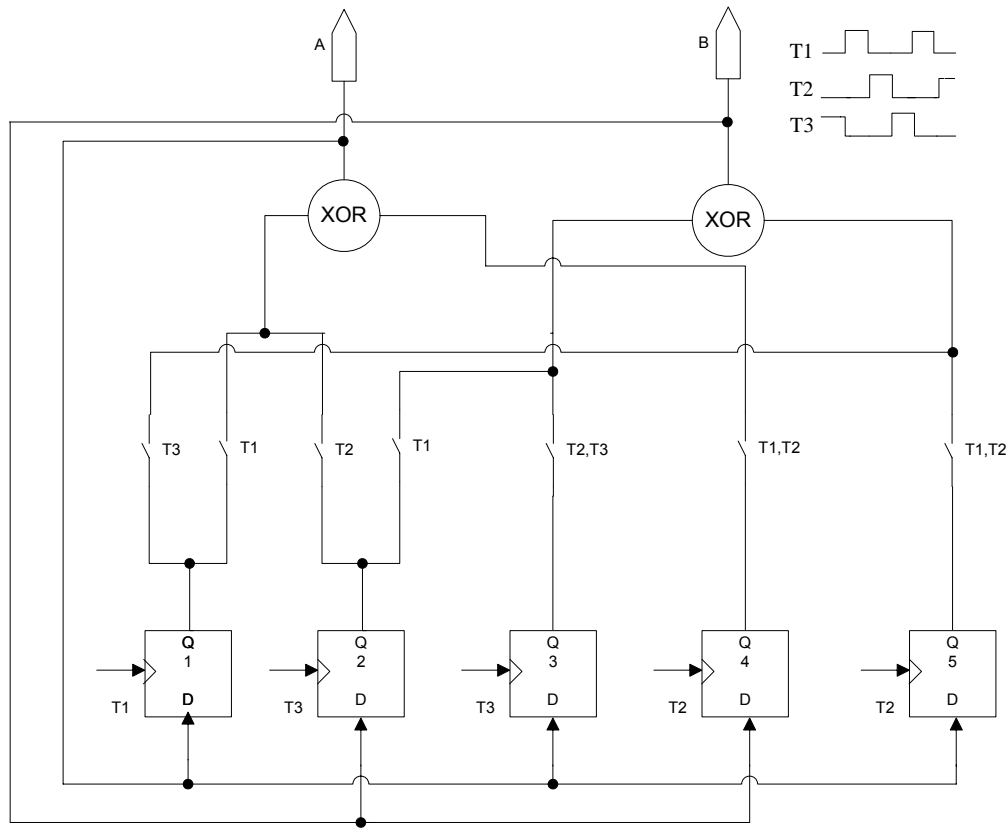


Figure 3 Structure of Katti's Architecture for Multiple Outputs LFSR with $1+x^2+x^5$

Table II
Operation of Katti's Architecture for Multiple Output LFSR with $1+x^2+x^5$

	T1	T2	T3
Output A	(5, 2)	(5, 3)	(3, 1)
Output B	(4, 1)	(4, 2)	
Flip-flop Updated	A → 1	A → 5 B → 4	A → 3 B → 2

Section 3 Comparison and drawbacks

In Lowy's architecture, all the operations are repeated at least twice. For the polynomial $1+x^2+x^5$, the operation (5, 2) is repeated at T1 and T3. At T2, the value of the XOR operation is stored in flip-flop 5, whereas at T4, the value is stored in flip-flop 4. As a result, the switching network which controls this operation has to be duplicated. Such duplication of switching circuitry is not present in Katti's architecture. The number of switches to control the tap for the above example in Lowy's architecture is 22, whereas the number of switches in Katti's architecture is 7. The number of control signals required in Lowy's architecture is 5, whereas in Katti's architecture only 3 are required. In Lowy's architecture each flip-flop is updated in at least two different clock cycles, so extra circuitry has to be added to achieve this operation. In Katti's architecture, each flip-flop is only updated once.

Both architectures show a race around condition during hardware implementation. In Lowy's architecture, at T1, operations (5, 2) and (4, 1) are performed. The operation (5, 2), the result of this XOR operation, is to be stored at T2 in flip-flop 5. T2 is used as a trigger for the flip-flop. When T2 goes high, T1 goes low. T1 controls the input switch to flip-flop 5. When the flip-flop is triggered, the input switch is open. There is a race around condition between the new value and the trigger, resulting in flip-flop 5 not being updated with the new value. In Katti's architecture, T2 is used to trigger flip-flop 5. T2 is also used to control the switch for operation (5, 3). The value of operation (5, 2) obtained at T1 is updated into flip-flop 5 at T2. At the same time, the new value of flip-flop 5 is to be XORed with the value of 3. A race around condition exists between the new value and the control signal T2 for the XOR operation.

One solution to avoid the race around condition is to perform the XOR operation after the flip-flops have been updated. By using different control signals for updating flip-flops and for XOR operations, the numbers of control signals required are doubled in both the architectures. $2 \times N$ control signals are needed in Lowy's architecture, and in Katti's architecture $2 \times \lceil N/k_1 \rceil$ control signals are required. The control signals used for triggering the flip-flops and the signals used to control the switches for the XOR operation must overlap as shown in Figure 4. Odd signals of T_i ($i = 1, 2 \dots 2N$) are used to control the switch for the XOR operation, and even signals are used for triggering the flip-flops. Though the number of control signals is doubled, number of clock cycles required to generate all the outputs are not doubled. For $1+x^2+x^5$ polynomial, in improved Katti's architecture, all the outputs are generated in 3.5 clock cycles. In improved Lowy's architecture, for the same polynomial, the number of clock cycle required to produce all the outputs is 5.5. The improved architectures for $1+x^2+x^5$, using Lowy's architecture and Katti's architecture, are shown in Figure 4 and Figure 5, respectively. Table III and Table IV summarize their operations, respectively.

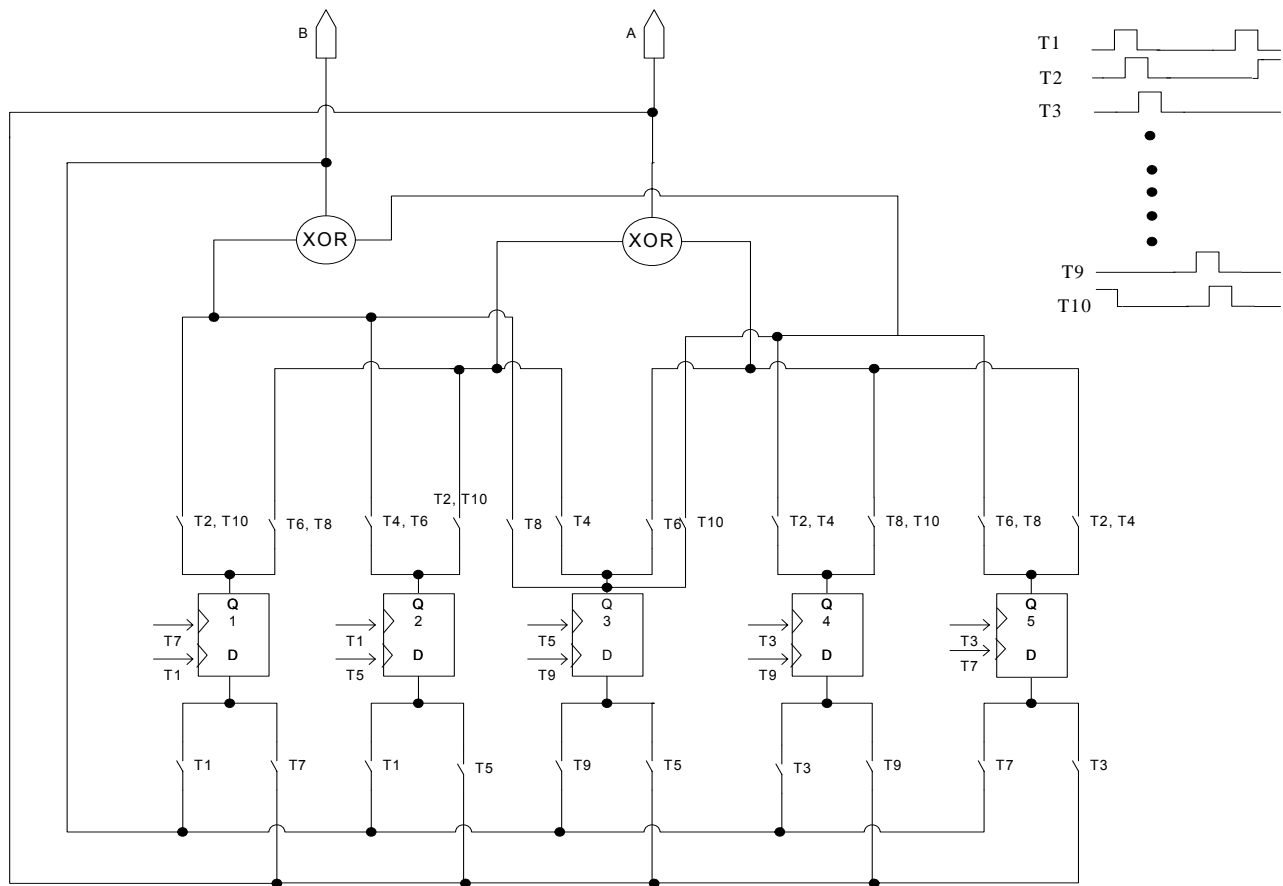


Figure 4 Improved structure of Lowy's architecture for multiple output LFSR with $1+x^2+x^5$

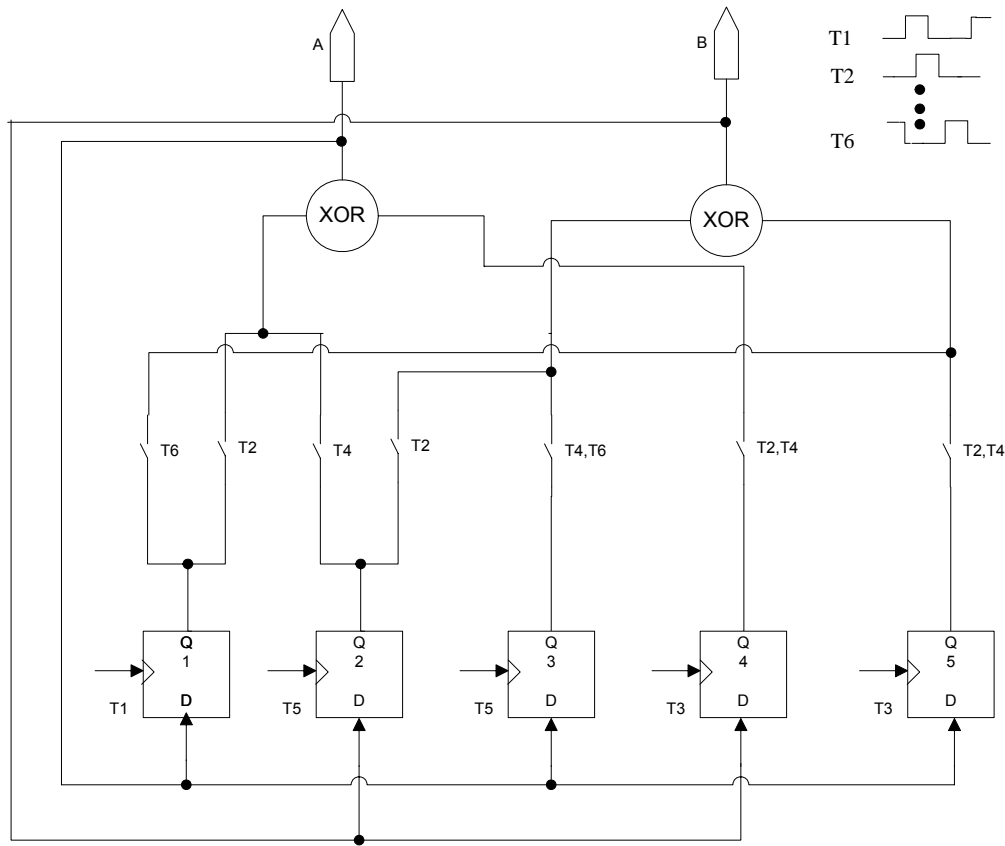


Figure 5 Improved structure of Katti's architecture for multiple output LFSR with $1+x^2+x^5$

Table III
Operation of Improved Lowy's Architecture for Multiple Output LFSR with $1+x^2+x^5$

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
Output A		(5, 2)		(5, 3)		(3, 1)		(4, 1)		(4, 2)
Output B		(4, 1)		(4, 2)		(5, 2)		(5, 3)		(3, 1)
Flip-flop Updated	A → 2 B → 1		A → 5 B → 4		A → 3 B → 2		A → 1 B → 5		A → 4 B → 3	

Table IV
Operation of Improved Katti's Architecture for Multiple Output LFSR with $1+x^2+x^5$

	T1	T2	T3	T4	T5	T6
Output A		(5, 2)		(5, 3)		(3, 1)
Output B		(4, 1)		(4, 2)		
To Flip-flop	A → 1		A → 5 B → 4		A → 3 B → 2	

One major drawback of Katti's architecture is that during some clock cycles, not all the output bits are valid. For example, in the case of $1+x^2+x^5$ polynomial at T6, only one XOR operation is performed, which is connected to output tap A. During this cycle, the output bit at tap B is not valid. In order to integrate such LFSR architecture in circuits, such as stream ciphers, extra control circuits must be introduced to avoid the use of invalid bits.

On the other hand, in Lowy's architecture of multiple output, none of the output bits are invalid, which comes at the expense of duplicating the entire switching circuit. Duplication of the switching circuit makes the hardware implementation more complex. Since each flip-flop is updated during more than one cycle, additional circuitry has to be introduced.

Section 4 Implementation

Both implementations of the improved architectures were done by using 0.13 μm BiCMOS technology provided by IBM. First a VHDL code at Behavioral or Register-Transfer-Level (RTL) was written to implement the various polynomials in both the architectures. The VHDL code was synthesized using Physically Knowledgeable Synthesis (PKS). The library that was used was for the worst case scenario, the supply voltage was 1.6 V and the operating temperature was 100 $^{\circ}\text{C}$. During synthesis, the RTL description was mapped to generic hardware components, such as gates, flip-flops, etc. After generic mapping, the design was compiled with the Standard Cell library provided by IBM. The timing and power values contained in the technology file were also in the worst case scenario. This synthesized netlist was then imported into First Encounter (FE) for placing and routing the design. In the process of importing a design, Library Exchange File (LEF) for 5 metal layers was also loaded. The LEF contains the information about the layout structure for standard cells. Once the design had been imported, the standard cells were placed in the core area. After placement, routing was done with minimum congestion and minimum delay using Nanoroute. During routing, the design was connected to a power grid as well as the I/O pads. Figure 6 and Figure 7 show the actual hardware floorplan for $1+x^2+x^5$ implemented using improved Katti's and improved Lowy's multiple output architectures, respectively. Improved Katti's architecture for $1+x^2+x^5$, needs 126 gates. The core area is $60.8 \times 30 \mu\text{m}^2$ and 80% of the core area is occupied by gates. Improved Lowy's architecture for same polynomial as above requires 191 gates and occupies 82 % of the $65 \times 47 \mu\text{m}^2$ core area.

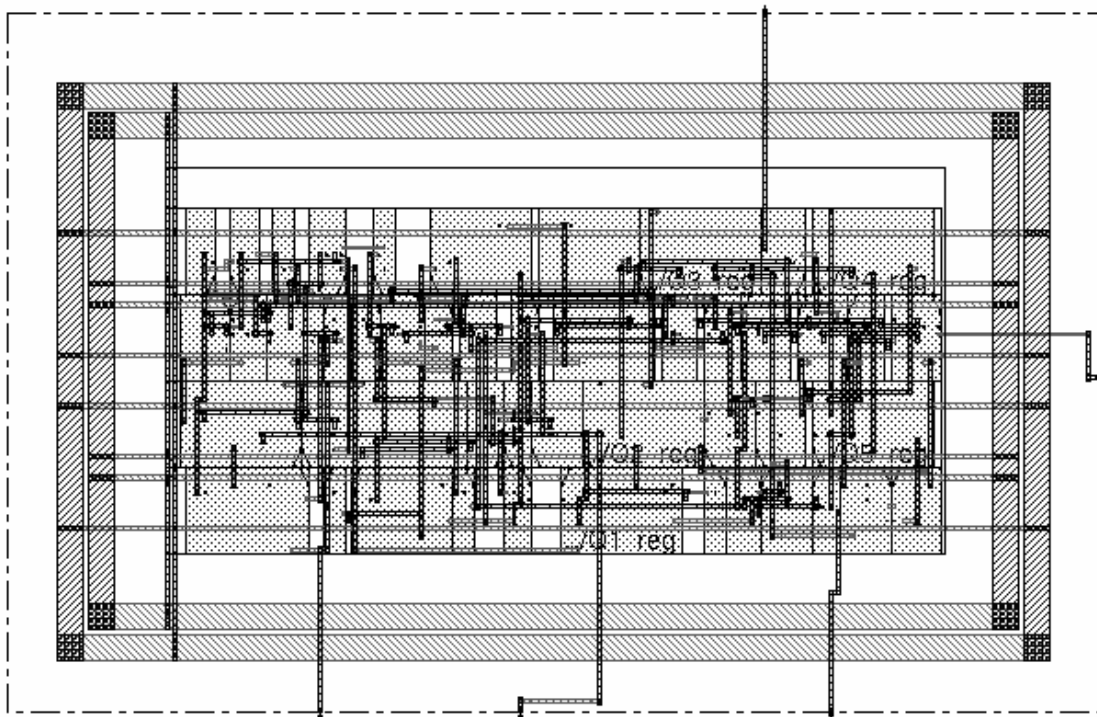


Figure 6 Floorplan of improved of Katti's architecture for multiple output LFSR with $1+x^2+x^5$

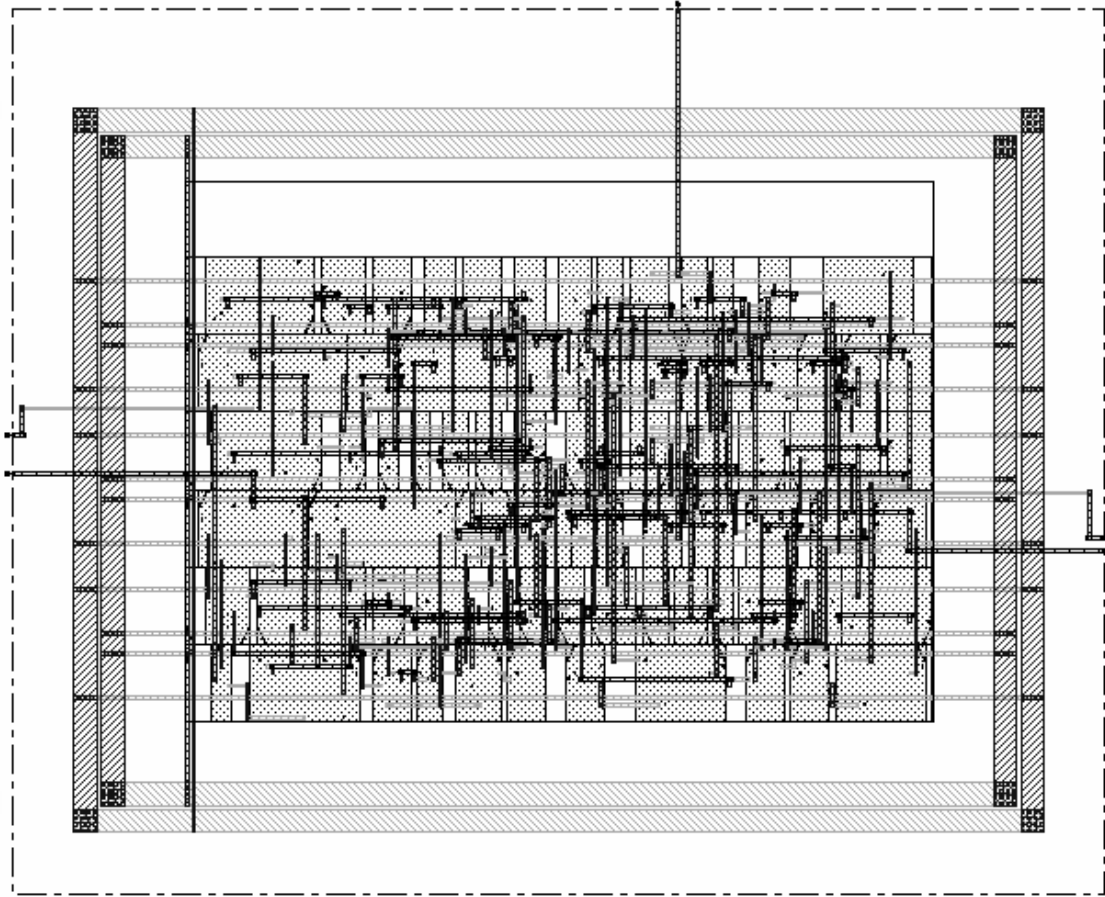


Figure 7 Floorplan for improved of Lowy's architecture for multiple output LFSR with $1+x^2+x^5$

Table V compares the hardware implementation of improved Lowy's multiple output architecture, as shown in Figure 4, and improved Kattii's architecture, as shown in Figure 5, in terms of static, dynamic, and total power consumed by these architectures for various polynomials. The first polynomial is the example considered for explaining the structure of both Kattii's and Lowy's architecture. It is clear that the architecture proposed by Kattii is more power efficient compared to the multiple output architecture of Lowy. Static power is influenced by the gate count. As the number of gates increases, the static power also increases. As the length of LFSR increases, the gate count in Lowy's architecture increases substantially, resulting in higher static power consumption as compared to Kattii's architecture. The length of the LFSR also effects the dynamic power consumption. The other factor that influences the dynamic power consumption is the maximum number of multiple output. The polynomial $1+x^3+x^{16}$ consumes more power than $1+x^2+x^5$ due to the longer length of LFSR. Whereas in case of $1+x^{14}+x^{15}$, though the length of LFSR is 15, it consumes more power than $1+x^3+x^{16}$, because the former has 14 simultaneous outputs in a single clock, and the latter has only 3 simultaneous outputs. In Lowy's architecture, the dynamic power consumption increases drastically as the length of LFSR and the number of multiple output increases. The total power consumption is the sum of static and dynamic power. The major advantages of these multiple output architectures, as compared with serial LFSR, are that not only are they power efficient but they also provide more than one output in a single cycle.

Table V
Hardware Comparison in terms of Power

Polynomial	Improved Lowy's Architecture				Improved Katti's Architecture				Efficiency %
	Power in μw			Gate Count	Power in μw			Gate Count	
	Dynamic	Static	Total		Dynamic	Static	Total		
$1+x^2+x^5$	1.055	0.401	1.456	126	0.386	0.405	0.791	191	45.67
$1+x^3+x^{16}$	3.104	1.972	5.076	469	0.753	0.663	1.416	296	72.10
$1+x^{14}+x^{15}$	4.531	4.449	8.980	1070	2.233	0.674	2.907	205	67.62
$1+x^8+x^{12}+x^{20}+x^{25}$	8.901	1.566	10.467	623	1.361	1.438	2.799	523	73.25
$1+x^{12}+x^{16}+x^{24}+x^{31}$	7.797	8.073	15.870	2199	2.543	1.923	4.466	718	71.85
$1+x^4+x^{24}+x^{28}+x^{33}$	8.902	6.755	15.657	1979	0.917	2.607	3.524	863	77.49
$1+x^4+x^{28}+x^{36}+x^{39}$	11.438	10.652	22.090	2490	0.923	2.674	3.597	935	83.71

The last four polynomials in the table above are used in the E0 cipher, which is a stream cipher and also standard for Bluetooth encryption. Stream ciphers operate in a serial manner. If stream ciphers can be operated in parallel fashion, the encryption can be done faster and consume less power. In order to make encryption faster, the first step is to implement the LFSR in parallel form with multiple output. A parallel stream cipher will not only be faster but also more power efficient.

All of the polynomials shown in the table above were also implemented in actual hardware. The gate counts shown in the table above were obtained from floorplans. The dynamic power consumption depends upon the activity factor of the gates. For $1+x^{14}+x^{15}$, implemented using improved Katti's architecture, though the gate count is less compared to $1+x^3+x^{16}$, dynamic power consumption is high because the activity factor of the gates is high. Similarly, $1+x^8+x^{12}+x^{20}+x^{25}$, implemented using improved Lowy's architecture consumes more dynamic power compared to $1+x^{14}+x^{15}$ although the gate count is less for the former.

Section 5 Conclusion

A comparison of Lowy's low-power multiple output architecture and Katti's low-power multiple output architecture, in terms of the hardware implementation and power consumption, was presented in this paper. The race around condition in the architectures as proposed by Lowy and Katti can be eliminated by performing the XOR operation after the flip-flops have been updated. The overlapping between the signals that control the switches for the XOR operation and the signals used for triggering flip-flops ensures the proper operation of the LFSR. Though the number of control signals required for both architectures is doubled, the architectures are more power efficient. This paper shows that the improved Katti's architecture is more power efficient than the improved Lowy's architecture, and in case of E0 stream cipher, the efficiency is as high as 83% for certain polynomials.

References:

- [1] Chandrakasan, P., Burstein, A., and Brodersen, R., 'Low power memory and arithmetic design for signal processing', IEEE Wkshp. LowPower Electronics, Phoenix AZ, 1993
- [2] Könemann, B., 'LFSR-Coded Test Patterns for Scan Designs', Proc European Test Conference, 1991, pp. 237-242
- [3] Lowy, M., 'Parallel Implementation of Linear Feedback Shift Register for Low Power Applications', IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 1996, Vol., 43, No. 6, pp. 458-466
- [4] Hamid, M., and Chen, C., 'A Note to Low- Power Linear Feedback Shift Registers', IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 1998, Vol. 45, No. 9, pp. 1304-1307
- [5] Huang, T., and Lee, K., 'A low-Power LFSR Architecture', Proceedings of the 10th Asian Test Symposium, IEEE Computer Science, 2001, pp. 470

- [6] Venkataraman, S., Rajski, J., Hellebrand, S., Tarnick, S., 'An Efficient Bist Scheme Based On Reseeding Of Multiple Polynomial Linear Feedback Shift Registers', IEEE Computer-Aided Design, 1993, pp. 572-577
- [7] Hellebrand, S., Rajski, J., Tarnick, S., Venkataraman, S., Courtois, B., 'Built-In Test for Circuits with Scan Based on Reseeding of Multiple Polynomial Linear Feedback Shift Registers', IEEE Transaction on Computers, 1995, Vol. 44, No. 2, pp. 223-233
- [8] Hellebrand, S., Rajski, J., Tarnick, S., 'Generation of Vector Patterns Through Reseeding of Multiple Polynomial Linear Feedback Shift Registers', Proceedings of International Test Conference, 1992, pp. 120-129
- [9] Katti, R., Ruan, X., Khattri, H., 'Multiple-Output Low-Power Linear Feedback Shift Register Design', IEEE Transactions on Circuits and Systems I: Regular Papers, 2006, Vol. 53, No. 7, pp. 1487 - 1495
- [10] Galanis, M., Kitsos, P., Kostopoulos, G., Sklavos, N., Koufopavlou, O., Goutis, C., 'Comparison of The hardware Architecture and FPGA Implementation of Stream Cipher', Electronics, Circuits and Systems, 2004. ICECS 2004. Proceedings of the 2004 11th IEEE International Conference, 2004, pp. 571 - 574