

A NEW PARALLEL ARCHITECTURE FOR LOW POWER LINEAR FEEDBACK SHIFT REGISTERS

Abdullah Mamun and Rajendra Katti
 Department of Electrical and Computer Engineering
 North Dakota State University

ABSTRACT

Low power dissipation is very critical in today's electronic designs. Components which are widely used in design, such as sequence generators like linear feedback shift registers (LFSR), should consume as little power as possible. Two recent works on parallel architecture of LFSR, one by M. Lowy and another by M. E. Hamid and C. I. H. Chen, have reduced dynamic power consumption significantly compared to the conventional architecture and showed the way to generate multiple outputs. In this paper we propose design improvements on these parallel architectures. The proposed method reduces dynamic power dissipation significantly, simplifies design process for single and multiple output generation, and eliminates the need of some hardware.

1. INTRODUCTION

The ever-increasing density and complexity of today's digital designs demand low power consumption. This becomes more critical for components, which are widely used. The sequence generator circuit, Linear Feedback Shift Register (LFSR), is widely used in data compression circuitry, encryption circuitry, Built-in Self-Test (BIST), communication circuitry, error correction circuitry etc.

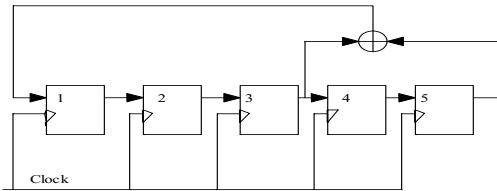


Figure 1. Conventional serial architecture of LFSR.

The conventional serial architecture of an LFSR with polynomial, $F(X) = 1 + X^3 + X^5$ is shown in Figure 1. Here length of the LFSR, which is denoted by N, is 5 and the number of taps or number of terms XORed, which is denoted by M, is 2. Power consumption in the serial architecture is high as all the flip-flops are clocked in every clock cycle and only one bit of information is generated per clock cycle. The output can be taken from input or output of any flip-flop. When the output of i successive cycles are generated in one cycle then the LFSR is an i-output (or multiple output) LFSR.

The parallel architecture of an LFSR designed by M. Lowy [1] reduces dynamic power consumption and can generate more than one bit output per clock cycle. Here only one flip-flop is updated every clock cycle and instead of the bits moving from left to

right in each clock cycle, the taps of the XOR tree move from right to left [1].

Lowy's design for the polynomial $F(X) = 1 + X^4 + X^5 + X^6 + X^7$, is shown in Figure 2. In clock T_0 , content of flip-flops 4, 5, 6, and 7 are XORed and stored in flip-flop 7 in the next cycle. In cycle T_1 content of flip-flop 3, 4, 5, and 6 are XORed and stored in flip-flop 6 and so on. In Figure 2 the flipflop clocks are signals T_i 's (not shown in figure) are obtained from a control circuit. Each T_i is asserted only in cycle i. The switches are controlled by signals that are obtained by OR-ing the T_i 's. This ensures that the correct flip-flop outputs are XORed and stored in the correct flip-flop in each cycle.

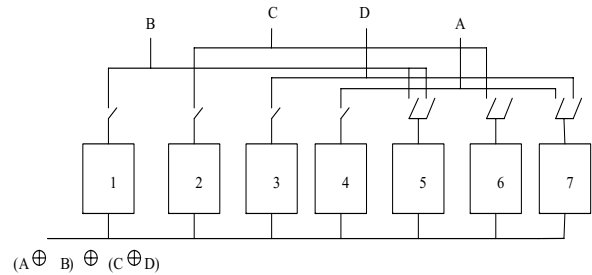


Figure 2: Lowy's design for $F(X) = 1 + X^4 + X^5 + X^6 + X^7$

In Lowy's design for an N-length LFSR, an N-phase generator generates N signals to clock the flip-flops. A control unit controls the operation of approximately M+N switches. The N-phase generator is realized by a Johnson counter of length N/2 and the control unit is realized by M+N OR gates.

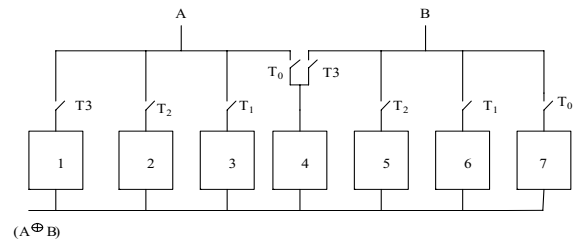


Figure 3: M. E. Hamid's design for $F(X) = 1 + X^4 + X^7$

A subsequent paper by M. E. Hamid and C. I. H. Chen[2] improved Lowy's design by increasing the number of distinct patterns generated and by lowering power consumed in some cases. This method uses a new polynomial of the form $F(X) = 1 + X^{N/2} + X^N$. M. E. Hamid and C. I. H. Chen's design for the polynomial $F(X) = 1 + X^4 + X^7$ is shown in Figure 3.

Both these designs [1, 2] have the following disadvantages. (1) The design processes are difficult as both the processes require switch control circuitry to control switches. (2) For Lowy's design switch minimization and control circuitry is very complex and requires large amounts of hardware [1]. (3) Multiple output generation is one of the key benefits of the parallel architecture. Multiple output generation can help to reduce supply voltage, which in turn reduces the power consumption further [1]. Lowy's method of double output generation requires (1) approximately three times increase in the number of switches, (2) changing one-clock flip-flop to two-clock flip-flops (3) doubling the number of XOR gate requirement. Hamid's design reduces some switch requirement but does require two clock flip-flops and double the number of XOR gates [2]. More than two output generation is practically impossible due to the complexities of the design.

To overcome the above difficulties, a new design method is proposed in this paper. In both Lowy's design and Hamid's design, taps move. In the proposed design additional XOR gates are used and they are permanently connected to the respective flip-flops. The proposed method is very simple in design, reduces power consumption significantly, eliminates the control unit, and is capable of generating up to $N/2$ outputs at a time with minimum hardware requirement and complexity. The power saving can be as high as 65.48% compared to Lowy's method and 39.58% compared to Hamid's method. The paper is presented in the following order. In Section 2, the proposed method for single output generation is described using the polynomial proposed by Hamid. Then the proposed method is described for an arbitrary primitive polynomial. After this multiple output generation is described briefly. The dynamic power consumption for all the polynomials is calculated and compared with previous methods in Sections 3 and 4.

2. THE PROPOSED METHOD

In the new method XOR gates are permanently connected to the respective flip-flops, thereby reducing the number of switches needed. The number of XOR gates required is $N/2$ for even order Hamid's polynomial and N for other polynomials.

2.1. New method using even order Hamid's polynomial

Clock cycle	1	2	3	4	5	6
XOR results of	3,6	2,5	1,4	6,3	5,2	4,1
Stored at	6	5	4	3	2	1

Table 1: Expected update of flip-flops, $F(X) = 1 + X^3 + X^6$

Let us use the polynomial $F(X) = 1 + X^3 + X^6$. Table 1 shows the storing method of calculated results in order to obtain the parallel implementation. In cycle 1, XOR result of flip-flops 3 and 6 (which is calculated in the previous cycle) is stored in flip-flop 6. Similarly in clock cycle 2, 3, 4, 5, and 6 XOR results of flip-flops (2,5); (1,4); (6,3); (5,2); and (4,1) should be stored in flip-flop 5, 4, 3, 2, and 1 respectively. It should be noted that XOR results of flip-flop (3,6) are same as XOR results of (6,3). Thus number of XOR gates needed is $6/2 = 3$. Table 1 enables the

construction of Figure 4. In general, for even order polynomial of the form $F(X) = 1 + X^{N/2} + X^N$, $N/2$ XOR gates are needed.

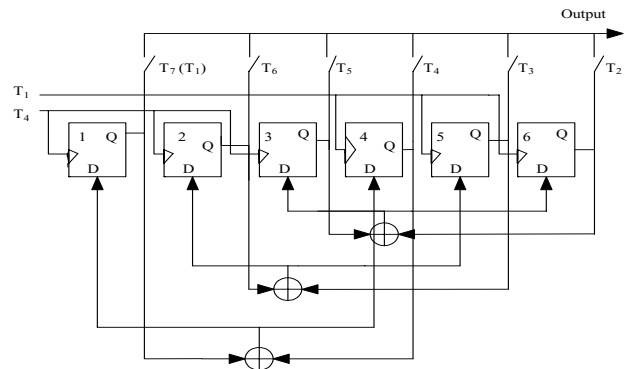


Figure 4: Single output realization of $F(X) = 1 + X^3 + X^6$.

2.2. New method using odd order Hamid's polynomial

For odd order polynomials, LFSRs have been tested with both ceiling and floor value of $N/2$ and the list of polynomials are given in [2]. Let us consider the polynomial for $N=7$, $F(X) = 1 + X^4 + X^7$. Table 3 shows the storing procedure in order to obtain the parallel implementation.

Clock cycle	1	2	3	4	5	6	7
XOR results of	4,7	3,6	2,5	1,4	3,7	2,6	1,5
Stored at	7	6	5	4	3	2	1

Table 2: Expected update of flip-flops, $F(X) = 1 + X^4 + X^7$.

Figure 5 is constructed from Table 2. It is evident that the number of XOR gates is 7, which is equal to N .

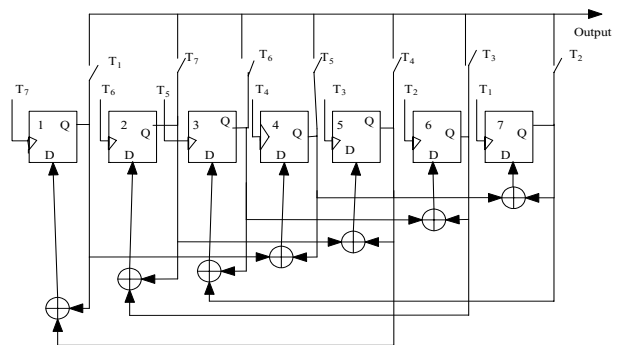


Figure 5: Single output realization of $F(X) = 1 + X^4 + X^7$.

2.3. New method using arbitrary primitive polynomial

Let us take an example of an arbitrary primitive polynomial $F(X) = 1 + X^4 + X^5 + X^6 + X^7$. The storing process is detailed in Table 3 which enables construction of Figure 6. To keep the drawing simple some connections are shown as dotted in Figure 6.

Clock cycle	1	2	3	4
XOR results of	4,5,6,7	3,4,5,6	2,3,4,5	1,2,3,4
Stored at	5	4	3	2
Clock cycle	5	6	7	
XOR results of	7,1,2,3	7,6,2,1	7,6,5,1	
Stored at	1	5	4	

Table 3: Expected update of flip-flops, $F(X) = 1+X^4+X^5+X^6+X^7$.

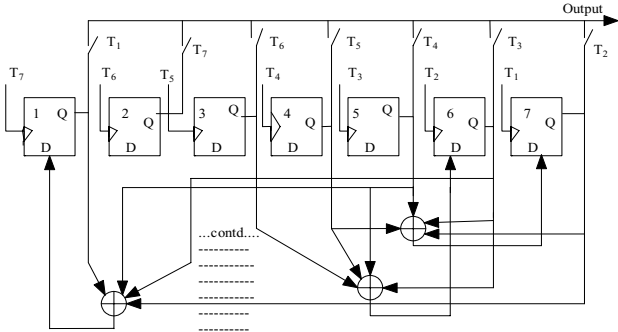


Figure 6: Single output realization of $F(X) = 1+X^4+X^5+X^6+X^7$

2.4. Multiple output generation

The proposed method can be used to generate multiple outputs. Let us consider the polynomial discussed in Section 2.1; $F(X) = 1+X^3+X^6$. From Table 1, it is evident that operation of clock cycle 1, 2, and 3 can be performed simultaneously as the operands do not depend on the result of these cycles. Similarly operations of clock cycle 4, 5, and 6 can be done simultaneously. So Table 1 can be rearranged as follows:

Clock cycle	1	2
XOR results of	3,6 2,5 1,4	6,3 5,2 4,1
Stored at	6 5 4	3 2 1

Table 4: Flip-flop updates using multiple operations per clock cycle for $F(X) = 1+X^3+X^6$

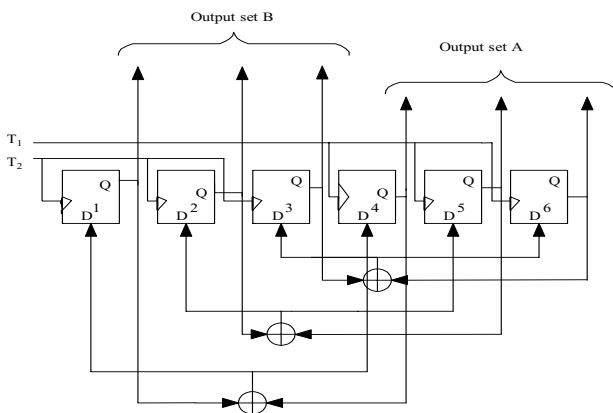


Figure 7: Multiple output generation of $F(X) = 1+X^3+X^6$.

Table 4 shows only two clock cycles are needed to do the operation instead of 6 clock cycles. In the first cycle XOR results

of flipflops (3,6); (2,5); and (1,4) (that are all calculated in the previous cycle) are stored in flip-flop number 3,2, and 1 respectively. In the next cycle XOR results of flipflops (6, 3); (5, 2) and (4, 1) are sorted in flip-flop number 3, 2, and 1 respectively.

From Table 4 it is evident that only 3 XOR gates are required as XOR connections required in the two cycles are symmetrical, i.e. (3,6) and (6,3) are the same XOR connection. Table 4 enables construction of Figure 7.

Figure 7 shows $N/2$ outputs are generated in each clock cycle. The clock T2 can be generated by inverting T1. So there is no need to have a multiphase generator as required by Lowy's design. Outputs A and B generate the XOR results of (3,6); (2,5); (1,4) and (6,3); (5,2); and (4,1) respectively. Some important aspects of this structure are,

- Circuit is very simple and generates $N/2$ outputs, whereas using previous methods more than two outputs are almost impossible to generate due to complexities involved.
- Only $N/2$ XOR gates are required.
- No need to have N -phase generator. T2 can be generated by inverting the clock.
- No need to have multi-clock flip-flops, which is required by earlier methods [2].
- No switches are required, whereas for only double output generation Lowy's method requires about $3*(N+M)$ switches [2].
- No extra XOR gates are required for multiple outputs, whereas previous methods require doubling the XOR gate requirement for double output generation.
- As the clock rate is reduced by $N/2$, supply voltage can be reduced which in turn reduces power consumption [1].

The idea of multiple output generation can also be extended for odd order or arbitrary polynomial. But this might have the following costs depending on the polynomial used: I. Maximum number of outputs generated is $(N+1)/2$ where N is odd. II. The number of XOR gates required is N . III. N -phase generator is required. IV. Switches will be required to extract the output V . $(N+1)$ clock flip-flop might be required, which can be eliminated using $(N+1)$ input OR gates to generate clock input of flip-flop.

3. DYNAMIC POWER CALCULATION

For dynamic power calculation, the same notations of Hamid's paper are used to make the comparison simpler. The dynamic power can be given by

$$P = 1/t_p \times C_{total} \times V_{dd}^2 \times (\text{percentage activity})$$

Where t_p is the clock period, C_{total} is the total capacitance driven by gate outputs, V_{dd} is the supply voltage and the percentage activity is 50%. The notations used in the calculations are:

P_{diff} = power dissipation of the D flip-flop with one output capacitance; P_{XOR} = power dissipation of an XOR gate with one output capacitance; P_{OR} = power dissipation of an OR gate with one output capacitance; P_{AND} = power dissipation of an AND gate with one output capacitance; P_{INV} = power dissipation of an inverter with one output capacitance.

In single output generation, the circuits consist of a multiphase generator, an array of N flip-flops, and a set of XOR gates. The multiphase generator is realized using Lowy's method as described in [1]. If the power dissipated by each clock is

included in the power dissipation of flip-flops, the load on each AND gate of the multiphase generator is one output switch (see Figures 4, 5 or 6). So the power consumption in the multiphase generator is, $P_m = 2 P_{dff} + 2 P_{AND}$ [1].

3.1. Proposed method using Hamid's polynomial

3.1.1. Even order polynomial

From Figure 4, it is evident that, on average, only one flip-flop is activated in every clock cycle. The load on each flip-flop is one output switch and one XOR gate. Assuming that the activated flipflop changes state only 50% of the time, the power consumed in the array of flip-flops is, $P_{1even} = (\frac{1}{2} \times P_{dff} \times (1+1)) = P_{dff}$, here $\frac{1}{2}$ denotes the fact that the flipflop changes state only 50% of the time [1]. On an average, in each clock cycle one XOR gate is activated. Load on each XOR gate is one flip-flop. So, similarly, power consumed by the XOR tree, $P_{2even} = (\frac{1}{2} \times P_{XOR} \times 1) = \frac{1}{2} P_{XOR}$. Total power consumed in the even order polynomial case is, $P_{even, single output} = P_m + P_{1even} + P_{2even} = 3P_{dff} + 2 P_{AND} + \frac{1}{2} P_{XOR}$

3.1.1. Odd order polynomial

From Figure 5, it is evident that load on each flip-flop is one output switch and two XOR gates. Also in each clock cycle two XOR gates are activated. Using the above procedure, it is found that

$$P_{1odd} = (\frac{1}{2} \times P_{dff} \times (2+1)) = 3/2 P_{dff}; P_{2odd} = (\frac{1}{2} \times P_{XOR} \times 2) = P_{XOR}$$

$$P_{odd, single output} = P_m + P_{1odd} + P_{2odd} = 7/2 P_{dff} + 2 P_{AND} + P_{XOR}$$

3.2. Proposed method using Arbitrary polynomial

Usually, for an arbitrary primitive polynomial, the number of taps need not be more than 2 or 4 [3]. If the number of taps is 2 the power consumption for the proposed method is the same as the proposed method for odd order Hamid's polynomial.

$$P_{arbitrary, single output, M=2} = 7/2 P_{dff} + 2 P_{AND} + P_{XOR}$$

If the number of taps is 4 as is in Figure 6, load on each flip-flop is 4 XOR gates and an output switch. Also in each clock cycle 4 XOR gates are activated. Here it should be noted that these XOR gates are 4 input and they can be replaced by three 2-input XOR gates. So the power consumption is given by,

$$P_{arbitrary, single output, M=4} = 9/2 P_{dff} + 2 P_{AND} + 2 (3 P_{XOR})$$

$$= 9/2 P_{dff} + 2 P_{AND} + 6 P_{XOR}$$

3.3. Even order polynomial with multiple outputs

In the multiple output circuit only two phase clock is required which can be generated using an inverted clock. The load on this inverter is $N/2$ flip-flops per cycle. So the power consumed in the 2-phase generator is $P_m = \frac{1}{2} N/2 P_{INV} = \frac{1}{4} N P_{INV}$. Load on each flip-flop is one XOR gate. In each clock cycle, number of flip-flops activated is $N/2$. Therefore power consumed in flip-flop array is $P_1 = \frac{1}{2} P_{dff} N/2 = \frac{1}{4} P_{dff} N$. At every clock cycle, $N/2$ XOR gates are activated. Load on each XOR gate is two flip-flops. So power consumed in XOR array is $\frac{1}{2} P_{XOR} N/2 \times 2 = \frac{1}{2} P_{XOR} N$. Therefore, total power consumed in even order $N/2$ output polynomial

$$P_{even, N/2 outputs} = N/4 P_{INV} + \frac{1}{4} N P_{dff} + N/2 P_{XOR}$$

This power can be further reduced by decreasing the supply voltage or frequency, as described in [1].

4. COMPARISON

Power consumption in the serial architecture, Lowy's method, and Hamid's method are respectively as follows [2]:

$$P_{Serial} = N P_{dff}/2 + (M-1) P_{XOR}/2$$

$$P_{Lowy} = 3 P_{dff} + 2 M P_{AND} + M P_{OR} + (M-1) P_{XOR}/2$$

$$P_{HAMID} = 3 P_{dff} + 4 P_{AND} + 2 P_{OR} + P_{XOR}/2$$

To compare different methods, we used a 0.18 μ process standard cell library i.e. the capacitances are $C_{dff}=0.0027$ pf, $C_{XOR}=0.0042$ pf, $C_{OR}=0.0026$ pf, $C_{INV}=.0027$ pf, $C_{AND}= 0.0215$ pf. $V_{dd}=1.8$ V. Frequency is set to 30MHz. For the serial architecture, the power consumption increases with N and a primitive polynomial of degree 64, having 4 taps is used for comparison. Power consumptions of all other methods are constant and independent of N. Table 5 shows the power comparison of new method with other methods for single outputs.

Proposed Method	Power	Percentage improvement over			
		Serial	Hamid	Lowy M=2	Lowy M=4
P_{even}	1.409	92.41	30.83	30.83	60.48
P_{odd}	1.745	91.79	25.21	25.21	57.26
$P_{arbitrary M=2}$	1.745	91.79	25.21	25.21	---
$P_{arbitrary M=4}$	4.048	80.96	---	---	0.83

Table 5: Single output power comparison.

From Table 5 it is evident that the new method saves considerable power compared to others. The savings can be as high as 65.48% compared to Lowy's method if the new method uses the polynomial proposed by Hamid. For the serial architecture, savings can be more when N increases.

5. CONCLUSION

Two previous papers one by Lowy and other by Hamid, describe parallel implementation of LFSRs to reduce dynamic power consumption and increase throughput. This paper proposes modifications to these designs resulting in significant reduction in dynamic power consumption, elimination of control circuitry, which is huge in Lowy's method, simplification of the design process especially for multiple output generation of LFSRs with even order Hamid's polynomial. Multiple output generation can be used to obtain additional power savings by reducing the frequency or decreasing the supply voltage.

6. REFERENCES

- [1] M. Lowy, "Parallel implementation of linear feedback shift registers for low power applications," *IEEE Trans. Circuits Syst. II*, vol. 43, pp. 458–466, June 1996.
- [2] Hamid, Muhammad E.; Chen, Chien-In Henry, "Note to low-power linear feedback shift registers," *IEEE Trans. on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, Issue 9, pp 1304-1307, September 1998.
- [3] Abramovici, Miran, et al, "Digital System Testing and Testable Design" New York, IEEE Press, 1990. pp 440.