A Novel Architecture of Area Efficient Counter For multistage LFSR

[1]Peringinga Dilip Kumar,[2]S. Ali asgar ,[3]Dr.P.Krishna Murthy

[1]M.Tech PG Scholar,[2]Associate professor, M.Tech,[3]HOD, M.Tech, Ph.D

[1,2,3] Department OF E.C.E.

[1,2]Chadalawada Ramanamma Engineering College, Tirupati, Andhra Pradesh, India.

Abstract:

Linear-feedback shift register (LFSR) counters have been shown to be well suited to applications requiring large arrays of counters. These counters can improve the area and performance compared with existing designs. However, significant logic is required to decode the count order of LFSR into binary order which takes additional logic circuitry. This paper presents a Multistage LFSR counter design which uses LFSR counters as well as binary counters. Combination of these two counters gives a better improvement in terms of area while maintaining the delay and power same like existing counter. Multistage LFSR counter is implemented in this paper on Xilinx 14.7 version.

Index Terms—3-D imaging, binary counters, decoding logic, Binary counter, event counters, linear-feedback shift register (LFSR), single-photon detection.

I.INTRODUCTION

With recent advancements in applications such as single photon detection, it has become necessary to implement a large number of massive counters in small areas. These include time-of-flight (TOF) cameras with a depth measurement range, where counters are required to count clock cycles, and photon counting cameras, which count the number of photons in an interval. Reducing the footprint of a counter in these applications is essential to increase the number of pixels in the cameras, as each pixel in the camera contains a separate counter. Three different methods of decoding a binary LFSR sequence are compared: the iteration method, the Forward Lookup Table (LUT) method, and time-memory offset. The iteration method iterates over the entire sequence of LFSR counters and compares each against the value of the counter. For an n-bit LFSR, this requires about 2n-1 comparisons on average. Instead, the direct LUT method uses annxn LUT that directly decodes the state of the LFSR. The time / memory tradeoff, presented in a combination of both methods, is to store 2 (N / 2) LFSR counter values in the table and repeat the LFSR sequence until the counter value matches the table value. Then the number of iterations is subtracted from the stored value to get the decoded value. Another algorithm based on discrete logarithms was introduced and adapted for use with ring generator event counters.

Large matrix applications require each cell in the matrix to be binary decoded for further processing, while systems on a chip require this on chip decoding. This requirement dictates that the decoding logic must be embeddable and fast, as many transformations must be performed. However, all of the above methods grow exponentially over time or in a region the size of LFSR. For single photon detection applications, there are several example array designs that cannot be implemented with LFSR counters without large built-in LUTs.

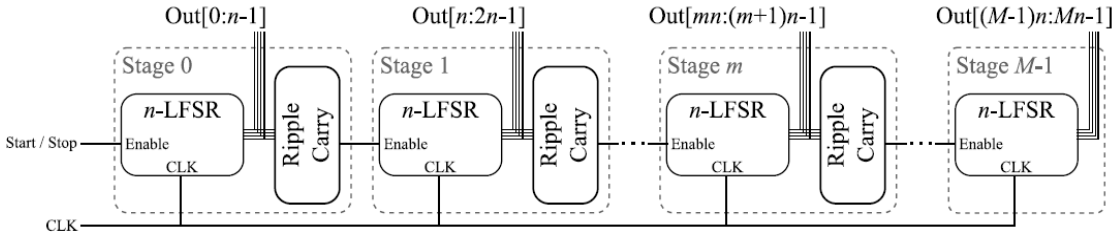


Fig. 1. Block diagram of the multistage LFSR counter.

This article proposes a new counter design based on multiple LFSR stages that can be decoded using logarithmic logic that increases with the counter size rather than exponentially. Although direct combining of LFSR counters would result in significant performance degradation similar to binary ripple counters, this article presents a method for distributing the ripple signal over time and compensates for this in a generalized decoding logic and n-bit LFSR will be referred to as n-LFSR.

II.EXISTING METHOD

The general scheme of the counter design is shown. There are *M* identical *n*-LFSR blocks that are controlled by an enable signal. When the $(m-1)$th*n*-LFSR undergoes a specific state change, the enable signal is asserted so that the *m*th*n*-LFSR advances one state. This allows the entire $M \times n$ bit state space to be traversed. In large arrayed designs, the counter can also act as a high-speed serial readout chain. This is achieved with minimal additional logic that bypasses the LFSR feedback and ripple-carry blocks. The multistage counter scheme reduces the counter into *M*in dependent modules, allowing each *n*-LFSR to be decoded separately by an $n \times n$bit LUT rather than an $(M \times n) \times (M \times n)$bit LUT. For small *n*, the LUT can easily be implemented on chip.

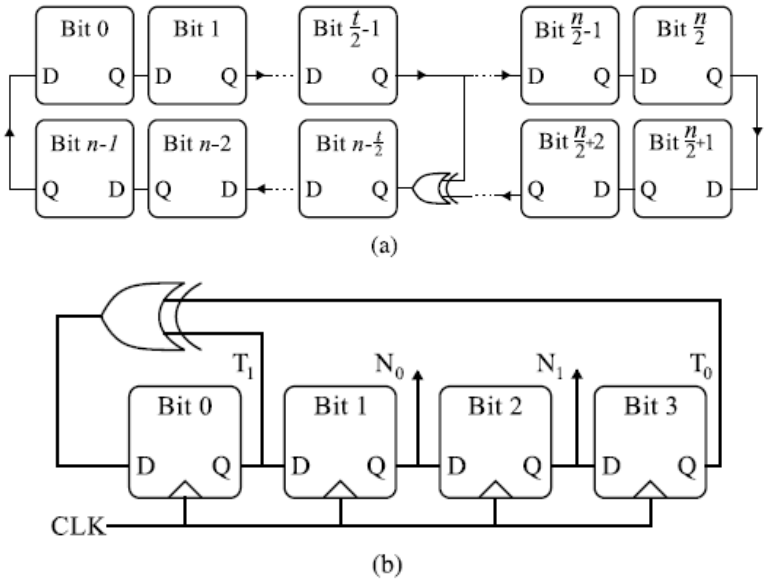


Fig. 2. (a) Structure of a conventional *n*-bit ring generator. (b) StructureOf conventional many-to-one 4-LFSRs.

Each stage of the counter is triggered once per period of the previous stage, so missing states from the LFSR sequence will cause large blocks of counter states to be missing from the counter state space. Thus, it is important that the *n*-LFSR is designed for a maximal length. The maximal sequence length of an *n*-LFSR is only $2n - 1$, so additional logic is requiredto incorporate the missing state into the count sequence. Thiscan be achieved using a NOR and XOR function to disable the feedback logic when the 0x000 . . .1 state is detected. This sequence-extension logic extendsthe sequence length of the individual component LFSRs to $2n$ so that the counter covers every state in the $2M \times n$ state space. This also allows the multistage counter to be used in applications that require every state to be covered, such as self-starting counters, where traditional LFSRs would not be applicable.

Several LFSR feedback styles exist, including many-to-one, one-to-many (alternatively known as Fibonacci and Galois LFSRs, respectively), and ring generators. Ring generators are typically regarded as the optimal way to implement an LFSR, where the shift register forms a ring and taps form sub loops within the ring. However, the sequence-extension requires additional logic in the LFSR, dominating the critical path. Instead, many-to-one style LFSRs are used, allowing the feedback logic and the sequence-extension logic to be combined into a single logic block for logic minimization. The multistage counter allows flexibility in choice of the size of the *n*-LFSR, so that small single-tap LFSRs are preferentially chosen.

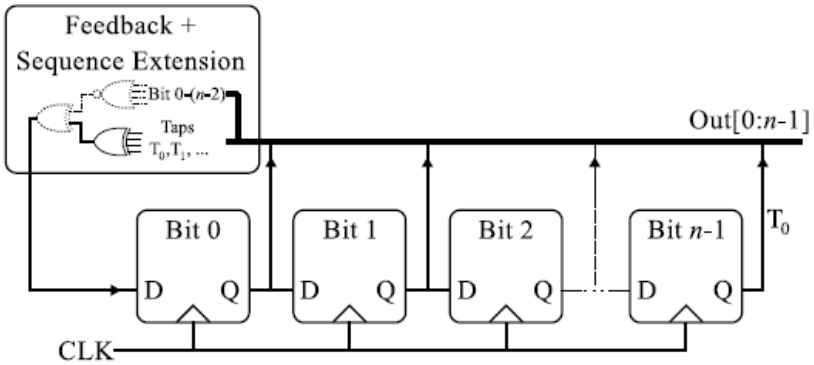


Fig.3 Structure of a proposed multistage *n*-LFSR block with sequence-extension logic (dotted components). The entire feedback block is implemented as a single logic block.

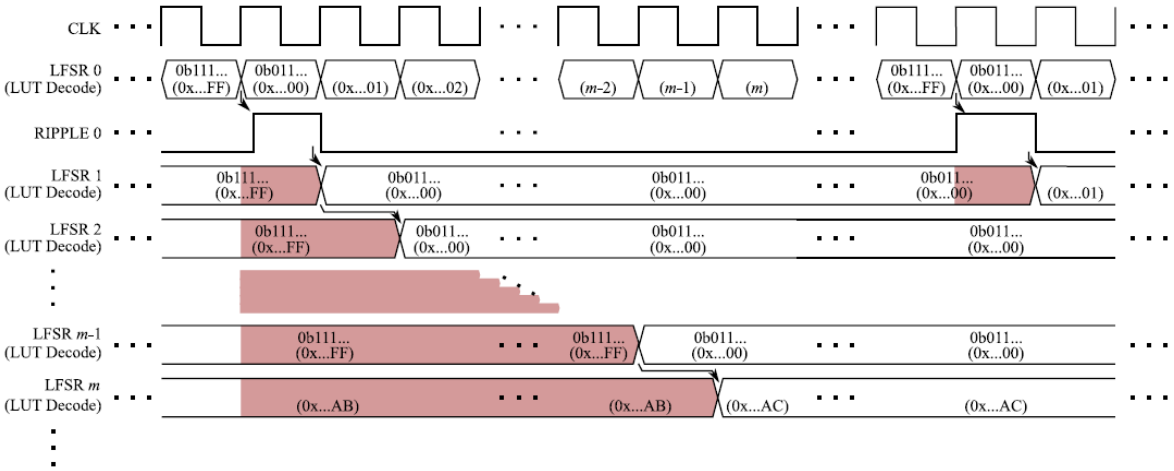


Fig. 4. Timing diagram of the operation of the multistage LFSR counter. Arrows show the operation of the ripple-carry logic. Highlighted states require further processing by the decoding logic.

A single-tap many-to-one LFSR is topologically indistinguishable from the corresponding ring generator. The maximal sequence length of an *n*-LFSR is only $2n - 1$, so additional logic is required to incorporate the missing state into the count sequence. This can be achieved using a NOR and XOR function to disable the feedback logic when the 0x000 . . .1 state is detected. This sequence-extension logic extends the sequence length of the individual component LFSRs to $2n$ so that the counter covers every state in the $2M \times n$ state space. This also allows the multistage counter to be used in applications that require every state to be covered, such as self-starting counters, where traditional LFSRs would not be applicable.

III.PROPOSED METHOD

Multistage counter design is proposed which uses LFSR counters as well as binary counters. Combination of these two counters gives a better improvement than the existing design with same features.

In this proposed design the four-stage design is modified i.e., first consists of LFSR and remaining bits consists of binary counter. Because of this arrangement the performance will be better than existing design. Multistage counter design is proposed in this paper which constitute of one LFSR and three binary counters. In the first stage LFSR counter is placed and in subsequent stages LFSR is replaced by the binary counters. In between two counters ripple carry logic is placed to reduce the performance degradation. Ripple-carry logic uses the output values of respected counter and also the completion of sequence of bits. The proposed counter is shown in below figure 6.

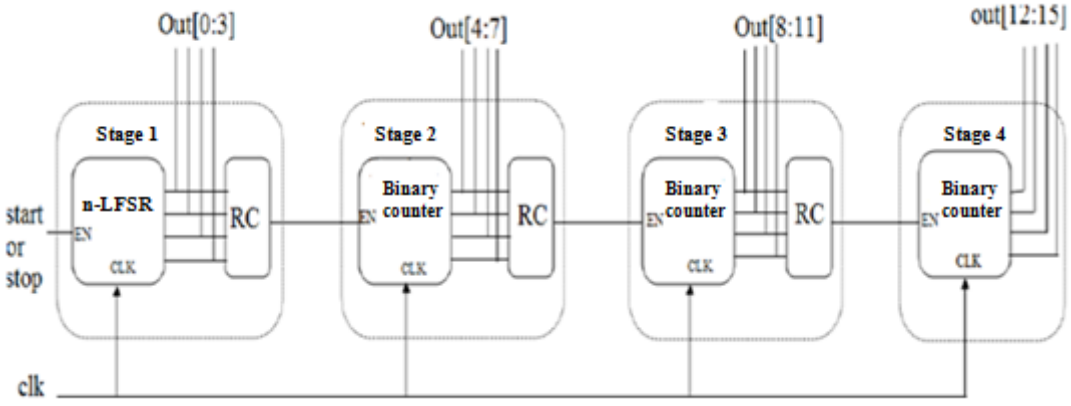


Fig. 6: proposed multistage LFSR counter design

In this first stage output of n bit LFSR is pseudo random. In single photon applications it is necessary to convert LFSR pseudo random count into binary order. The first stage output can be used of testing applications remaining stages are used in counter applications. Decoding logic done this work. Decoding logic which is used in existing multi stage counter is used in this proposed counter also. Decoding logic used in this paper takes less comparisons to convert LFSR count values into binary order compared with former decoding techniques.

n-LFSR Implementation:

The basic flip-flop used in the designing a n-LFSR is a D flip-flop. D means data whatever the data present at the side of input that is coming out to output side. The D flip flop is widely used. It is also known as a "data" or "delay" flip-flop. The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge of the clock). That captured value becomes the Q output. At other times, the output Q does not change. The D flip-flop can be viewed as a memory cell, a zero-order hold, or a delay line.

Linear feedback shift registers have been the most popular devices to date for generating and handling pseudorandom sequences. LFSR's usually used as a event counters and pseudorandom number generators. In applications of LFSR, feedback plays major role in generating binary sequences. If XOR gate is used as a feedback element then all zero state is illegal whereas XNOR is used as a feedback then all state is illegal means that state is not present in binary sequence. The maximum sequence length of LFSR is $2^{(n-1)}$states. in order to all the number of stages it means including "0000" state in LFSR which is not available in the existing designs. one two input XOR gate and one 3 input NOR gate are required in order to get the full sequence.

The sequence length of a LFSR is also depend on tap bits. Those bits are influence input of a LFSR counter, those bits are called as a tap bit. Bits which are not affecting the input of counter consider as a non-tap bit. These tap bits are depending upon the feedback elements in

which order they connected in between flip-flops or shift register. Therefore, sequence length of a LFSR counter also varies in which order feedback elements are connected. The tap configuration that gives a maximal sequence length with the least number of taps is preferred to minimize the number of transistors in the feedback logic. the diagram of n-LFSR design.

BINARY COUNTER:

A binary counter is a hardware circuit that is made out of a series of flip-flops. The output of one flip-flop is sent to the input of the next flip-flop in the series. A binary counter can be either asynchronous or synchronous, depending on how the flip-flops are connected together. In this we use 4 bit up-counter in the place of n bit LFSR in the stage2, stage 3 and stage 4. so, there is no need of decoding logic in these stages to convert the order into binary. Hence the hardware overhead will be reduced in the overall circuit.

RIPPLE-CARRY LOGIC IMPLEMENTATION:

The ripple-carry logic detects when the n-LFSR undergoes the 1111 to 1110 transition and signals next stage to increment one state on the next clock edge. While a straightforward concatenation of stages of counters would cause a significant performance reduction. This paper introduces a ripple carry technique to distribute the ripple signal in time and compensates for this in a generalized decoding logic scheme.

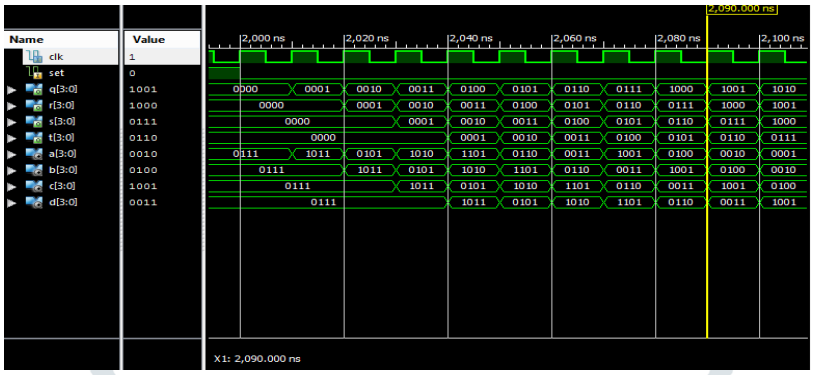
DECODING LOGIC:

In this paper, multi stage counter is designed with one LFSR in first stage and three binary counter in remaining three stages. In between two counters, ripple carry logic is placed to reduce the degradation of data. the design is shown in the above figure 5. Ripple carry logic also plays a vital role in functioning of counter. The first stage LFSR counter has the random output. These random output are again given to the decoding logic which is used to convert the random numbers into binary order. In decoding logic, LUT convert the most of bits into binary order. When the number of bits of multi stage counter, then decoding done by LUT doesn't gives correct output. So some additional logic is used to convert the bits in binary order.

V.RESULTS

The proposed multi stage counter and existing multi stage counter both are functionally verified by using Xilinx 14.7 version. the proposed design has better area when compared to existing design without degrading the other factors.

simulation results:



Proposed Results:

Simulation Results:

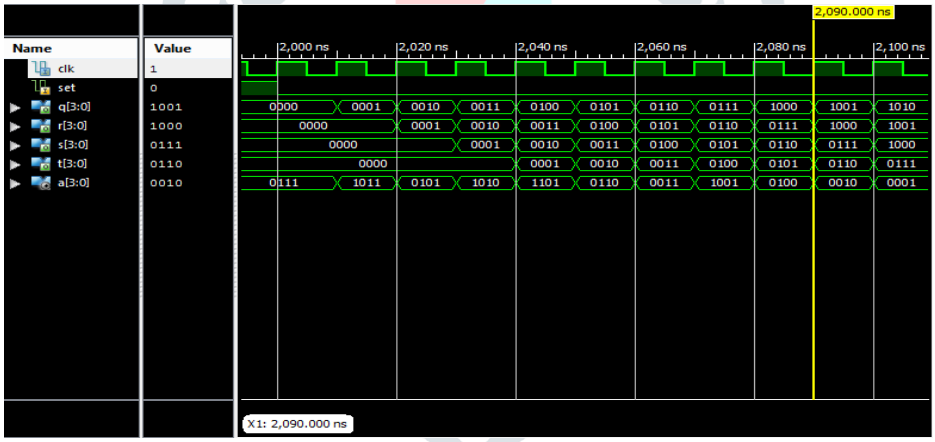


Table 1: Comparison between Existing method and Proposed method of different parameters.

	Existing method	Proposed method
Number of slices	17	11
Number of 4 input LUTS	23	20

CONCLUSION

This paper presents a generalized design and a practical implementation of multistage counter as well as the decoding logic required to convert the count sequence into binary order. This paper presents a Multistage LFSR counter design which uses LFSR counters as well as binary counters. The proposed multi stage counter and existing multi stage counter both are functionally verified by using Xilinx 14.7 version. The above results shows that the proposed design consists of less area when compare to the existing design while maintaining the delay and power same.

VII.REFERENCES

[1] D. Bronzi, Y. Zou, F. Villa, S. Tisa, A. Tosi, and F. Zappa, "Automotive three-dimensional vision through a single-photon counting SPAD camera," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 782–795, Mar. 2016.

[2] I. Vornicu, R. Carmona-Galán, and A. Rodríguez-Vázquez, "A CMOS 0.18 μm 64×64 single photon image sensor with in-pixel 11 b timeto- digital converter," in *Proc. Int. Semiconductor Conf. (CAS)*, 2014, pp. 131–134.

[3] M. Perenzoni, D. Perenzoni, and D. Stoppa, "A 64×64-pixels digital silicon photomultiplier direct TOF sensor with 100-MPhotons/s/pixel background rejection and imaging/altimeter mode with 0.14% precision up to 6 km for spacecraft navigation and landing," *IEEE J. Solid-StateCircuits*, vol. 52, no. 1, pp. 151–160, Jan. 2017.

[4] J. M. Pavia, M. Scandini, S. Lindner, M. Wolf, and E. Charbon, "A 1×400 backside-illuminated SPAD sensor with 49.7 ps resolution, 30 pJ/sample TDCs fabricated in 3D CMOS technology for near-infrared optical tomography," *IEEE J. Solid-State Circuits*, vol. 50, no. 10, pp. 2406–2418, Oct. 2015.

[5] C. Niclass, M. Soga, H. Matsubara, M. Ogawa, and M. Kagami, "A 0.18-μm CMOS SoC for a 100-m-range 10-frame/s 200×96-pixel time-of-flight depth sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 315–330, Jan. 2014.

[6] R. Ballabriga, M. Campbell, E. Heijne, X. Llopart, L. Tlustos, and W. Wong, "Medipix3: A 64 k pixel detector readout chip working in single photon counting mode with improved spectrometric performance," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc.Equip.*, vol. 633, pp. S15–S18, May 2011.