A NOVEL ENERGY-EFFICIENT COUNTER ARCHITECTURE FOR MULTISTAGE LFSR

[1] K. RAVI BABU, [2] NAGULA. SUMATHI.

[1] ASSOCIATE PROFESSOR, DEPT OF ECE, IN SREE VAHINI INSTITUTE OF SCIENCE AND TECHNOLOGY, TIRUVURU, ANDHRA PRADESH 521235, AP.

[2] PG SCHOLAR, DEPT OF ECE, IN SREE VAHINI INSTITUTE OF SCIENCE AND TECHNOLOGY, TIRUVURU, ANDHRA PRADESH 521235, AP.

Abstract:

When dealing with applications that call for vast arrays of counters, linear-feedback shift register (LFSR) clocks have shown to be an excellent choice. When compared to current designs, these counters may increase both area and performance. To convert the LFSR count order to binary order, however, requires a lot of extra logic equipment. Incorporating both binary and LFSR counters into a multistage system is the subject of this research. If you want to boost the area improvement while keeping the latency and power the same as the old counter, combine the two of them. In this study, we use Xilinx 14.7 to create a multistage LFSR counter.

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

I.INTRODUCTION

A plethora of enormous counters packed into compact spaces is now required due to developments in applications like single photon detection. In order to calculate depth, time-of-flight (TOF) cameras need counters to keep track of clock cycles, while photon counting cameras measure the amount of photons in an interval. To increase the total amount of pixels in the recording devices, it is vital to reduce the size of a counting in these applications. Each of the pixels in the camera carries a distinct counter. Iteration, Forward Lookup Table (LUT), and time-memory offset are the three approaches to decode a binary LFSR sequence that are evaluated. The iteration technique repeatedly checks each value of the counter against the full sequence of LFSR counters. The average number of comparisons needed in an n-bit LFSR is 2n-1. The direct LUT approach, on the other hand, makes use of annxn LUT, which decodes the LFSR state directly. By combining the two approaches, we can show the time / storage tradeoff as follows: assign a value to the counter and then repeat the LFSR pattern until the two values match. The decoded value is then obtained by subtracting the stored value from the number of iterations. For ring generator event counters, an additional discrete logarithm-based technique was developed and modified.
For large matrix applications, it is necessary to binary decode each cell before processing continues; however, for systems on a chip, this decoding must be performed on the chip itself. Due to the high volume of changes that must be executed, the embeddability and speed of the decoding logic are imperative requirements. But in an area that is the scope of LFSR or over time, all of those strategies expand tremendously. It is not possible to use LFSR counters for just one photon detection because to the massive built-in LUTs required by numerous sample array designs.

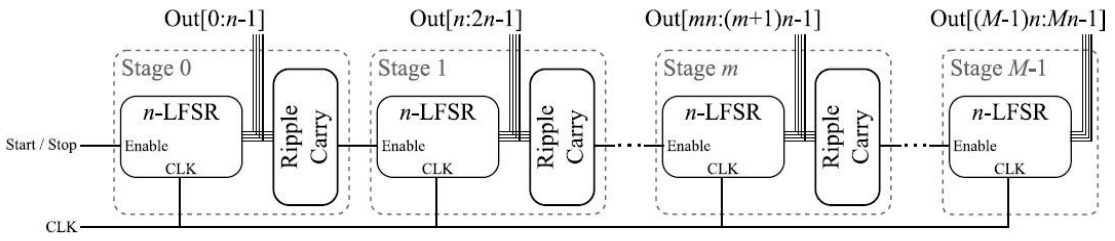


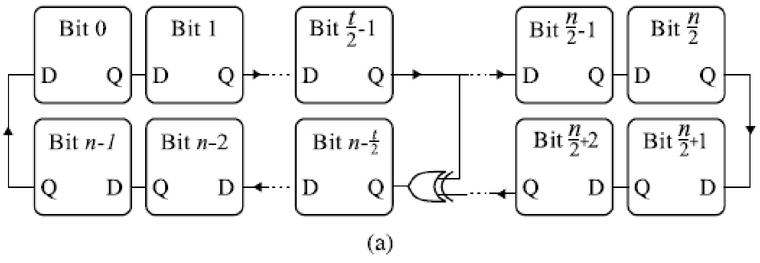
Fig. 1. Block diagram of the multistage LFSR counter.

Deciphering this new counter design, which is based on many LFSR stages, will be easier using logarithmic logic, which grows with the size of the counter instead of exponentially. This article introduces a technique for spreading a ripple signal over time, which compensates for the performance degradation that would occur when LFSR counters are directly combined—a phenomenon that is comparable to that of binary ripple counters. As a result, n-LFSR will be used to describe n-bit LFSR.

II.EXISTING METHOD

This is a picture of the counter's overall layout. An enable signal controls M n-LFSR blocks which are similar to one another. The allow signal is asserted to advance the mthn-LFSR by one state whenever a (m − 1)thn-LFSR goes through a certain state change. A full traversal of the M × n bit space of states is made possible by this. The counter may double as a fast serial reading chain in big arrayed devices. To get around the LFSR feedback with ripple-carry blocks, very little more logic is needed. An n×nbit LUT, instead of a (MÏn)×(MÏn)bit LUT, can decode each n-LFSR independently thanks to the multistage counter approach, which divides the countdown into Min dependant modules. It is easy to implement the LUT on chip for small n.

.



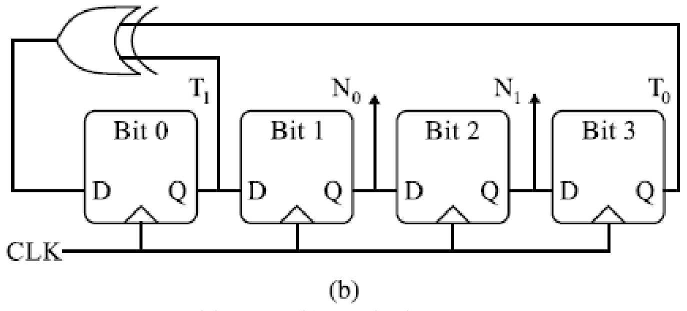


Fig. 2. (a) The classic n-bit ring generator's structural layout. (b) Typical many-to-one 4-LFSR structure.

Missing states in the LFSR sequence will result in vast blocks of clock states being missing from the counter's state space since every phase of the counter is activated once per period following the preceding stage. This highlights the significance of designing the n-LFSR for a maximum length. To include the vacant state in the count sequence, more logic is needed since the greatest sequence number for an n-LFSR is limited to 2n − 1. By using NOR and XOR functions, it is possible to stop the feedback loop upon detection of the 0x000...1 state. In order for the counter to include all states in the 2M×n phase space, this logic for sequence extension increases the total number of cycles of each of its component LFSRs to 2n. In addition, this makes the multistage counter suitable for uses where conventional LFSRs would not work, such as autonomous counters, which need coverage of all states.

There are a number of different types of LFSR feedback, such as ring generators, many-to-one, and one-to-many (often called Galois and Fibonacci LFSRs, respectively). The most common and ideal implementation of LFSRs is the ring generator, which uses the shift register to create a ring and the taps to create sub loop within the ring. Nevertheless, the critical path is dominated by the sequence-extension, which necessitates more logic in the LFSR. As an alternative, many-to-one type LFSRs are used, which enable the integration of feedback logic and sequence-extension logic into a singular logic block for the purpose of logic minimisation. Due to the multistage counter's scalability options, compact, single-tap LFSRs are given preference.

.

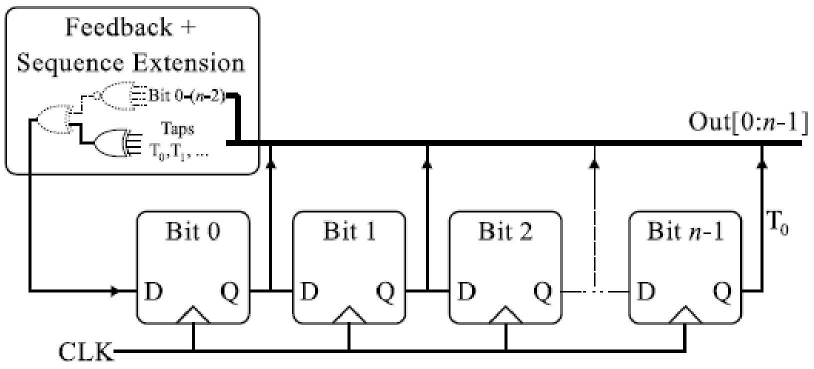


Fig.3 The schematic of an n-LFSR block having sequence-extension circuitry (dotted components) that is supposed to be multistage. Everything about the feedback block is built into a single logic block.

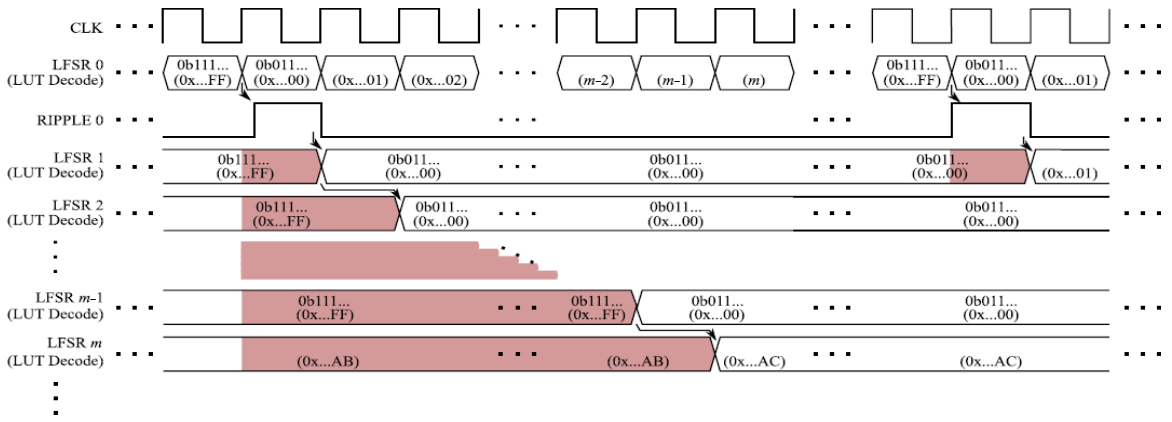


Fig. 4. Figure out how the multiple-stage LFSR counter works using this timing diagram. The ripple-carry logic is shown using arrows. More work by the decoder logic is necessary for the highlighted states.

There is no topological difference between the appropriate ring generator and a single-tap many-to-one LFSR. To add the missing state to the count sequence, more logic is needed since the maximum sequence duration for an n-LFSR is limited to 2n − 1. To do this, we may use the NOR and XOR functions to deactivate the feedback mechanism when we detect the 0x000...1 state. In order for the counter to include all states within the 2M×n state space, this logic for sequence extension lengthens the sequence of the individual LFSRs to 2n. In addition, this makes the multistage counter suitable for uses where conventional LFSRs would not work, such as autonomous counters, which need coverage of all states.

III.PROPOSED METHOD

The authors suggest a multistage counter architecture that makes use of binary and LFSR counters. Compared to the current design, which has the same characteristics, the amalgamation of both of these counters is an upgrade. The four-stage concept is altered in this design proposal; the first stage now includes an LFSR and the remaining bits are a binary counter. Performance will be higher than the current design thanks to this arrangement. This study proposes a multistage counter system that uses one LFSR plus three binary clocks. After the first step, the binary counters take the role of the LFSR counter. Ripple carry logic is inserted between two counters to minimise performance deterioration. The values of the respected counters' outputs and the completion of bit sequences are used by ripple-carry logic. You can see the suggested counter in picture 6.

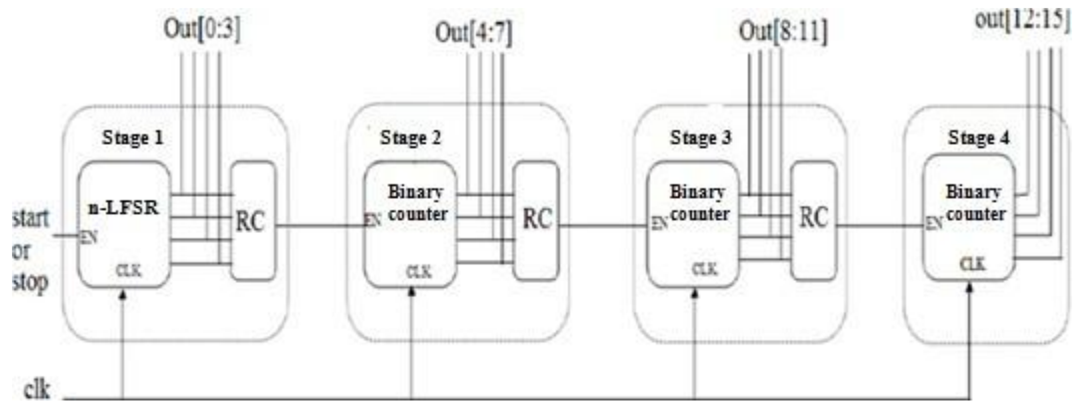


Fig. 6: planned multi-stage LFSR counter architecture

The output of the n-bit LFSR is pseudo-random at this first stage. You have to transform the LFSR pseudo-random count to binary order for single photon applications. Testing apps may make use of the first stage's output, whereas counter applications make use of the subsequent stages. This work involves decoding logic. This suggested counter makes use of the same decoding mechanism as the current multi-stage counter. When converting LFSR count values to binary order, the decoding logic shown in this study uses fewer comparisons than previous decoding algorithms.

n-LFSR Implementation:

D flip-flops are the most fundamental building blocks of an n-LFSR. D stands for data, which may be any information from the input side that is being sent out to the output side. One common kind of flip-flop is the D type. A "information" or "delay" flip-flop is another name for it. You can tell whether the clock is rising or falling by looking at the value that the D flip-flop stores for the D-input. The Q output is that captured value. Occasionally, Q remains unchanged. A memory cell, zero-order hold, or delay line are all ways of looking at the D flip-flop.

To far, the most widely used devices for creating and manipulating pseudorandom sequences have been linear feedback shift registers. Common applications for LFSRs include pseudorandom number generators and event counters. A key component of LFSR applications is feedback, which is used to generate binary sequences. All zero states are unlawful when using an XOR gate as a feedback element, whereas all states are illegal when using an XNOR gate, meaning that state does not exist in the binary sequence. The maximum number of states that may be in an LFSR sequence is $2(n-1)$. To include all the phases, it is necessary to incorporate the "0000" state in LFSR, which is currently unavailable in the designs. For you to get the whole sequence, you will need one XOR gate with two inputs and one NOR gate with three.

Tap bits also determine the LFSR sequence length. A LFSR counter uses those bits—called a tap bit—as its influence input. Bits are considered non-tap if they do not change the input of the counter. The feedback components of these tap bit are dependent on which sequence the connections were made between the shift register and the flip-flops. Thus, the order of the connections between the feedback components determines the sequence length of an LFSR counter. To reduce the amount of transistors used in the feedback circuitry, it is preferable to choose the tap layout that provides the longest possible sequence with the fewest taps. n-LFSR design schematic.

BINARY COUNTER:

Binary counter hardware is an array of flip-flops. Flip-flops input their outputs into each other. Flip-flop connections synchronise or asynchronize binary counters. Instead of n-bit LFSR in phases 2, 3, and 4, we employ a 4-bit up-counter. Thus, decoding logic is needed to transform the order to binary at these points. It will reduce the circuit's hardware cost.

RIPPLE-CARRY LOGIC IMPLEMENTATION:

The following step is signalled to increase a state on the following clock edge by the ripple-carry logic, which detects whenever the n-LFSR transitions from 1111 to 1110. However, speed would suffer greatly if stages of counters were simply concatenated. In order to spread the ripple signal out across time, this study presents a ripple carry approach. A generalised decoding logic system then accounts for this.

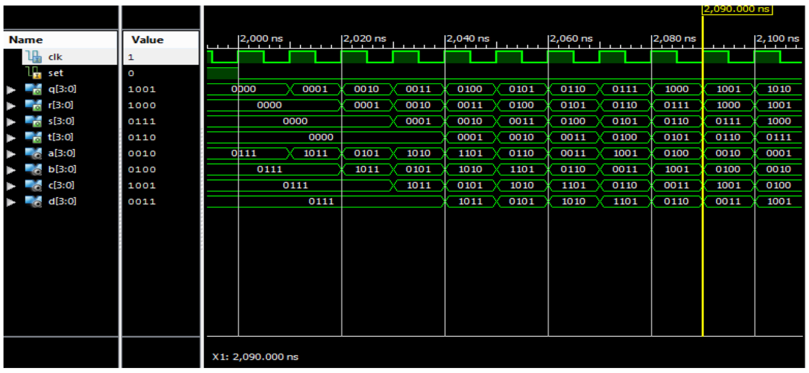
DECODING LOGIC:

A three-binary counter design using a single LFSR in the first stage makes up the multi-stage counter shown in this article. To lessen data deterioration, rippling carry logic is included between two counters. Above, in figure 5, you can see the design. An further critical component of the counter's operation is ripple carry logic. The LFSR counter's random output is in the first stage. After being generated at random, these numbers are sent back to the decoding machinery to be converted into binary order. Logic units transform the majority of bits into basic order for decoding purposes. The LUT-done decoding doesn't provide accurate results when the multi-stage counter bit count is too high. Consequently, the bits are converted to binary order using some extra logic.

V.RESULTS

Using Xilinx 14.7 version, we functionally validated both the current multi stage clock and the suggested solution multi stage counter. Without sacrificing any of the other aspects, the suggested design offers a larger floor plan than the current one.

simulation results:



Proposed Results:

Simulation Results:

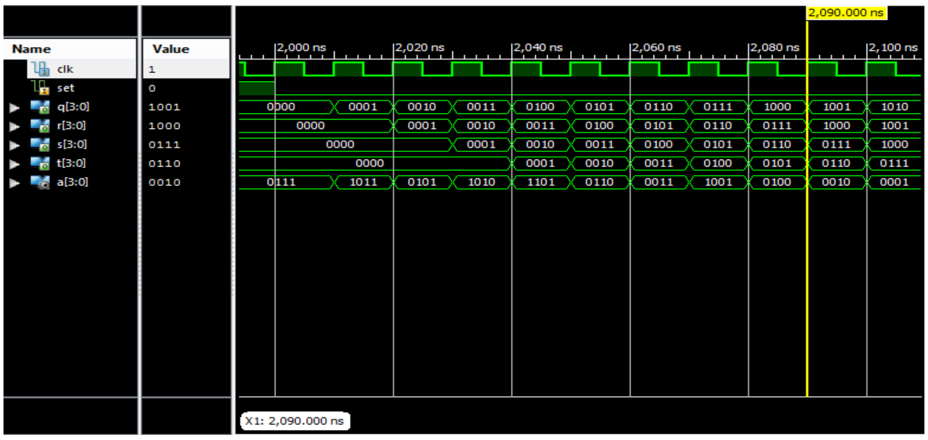


Table 1: Comparison between Existing method and Proposed method of different parameters.

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

CONCLUSION

This article details the decoding logic needed to transform the counting sequence into binary order, as well as a realistic way to implement a multistage counter and a generalised design for it. Incorporating both binary and LFSR counters into a multistage system is the subject of this research. Using Xilinx 14.7 version, we functionally validated both the current multi stage counter & the suggested multi stage counter. With no change to the delay or power consumption, the suggested design uses less space than the current one, as seen above.

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.