2026
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SANTA CLARA, CA, USA
MARCH 2 - 5, 2026

LFSR: Beyond the Sequential

Kshiteej Kosambia – Design Engineer II
Prakash Darji – Sr Principal Design Engineer
Cadence Design Systems

cadence®

Abstract

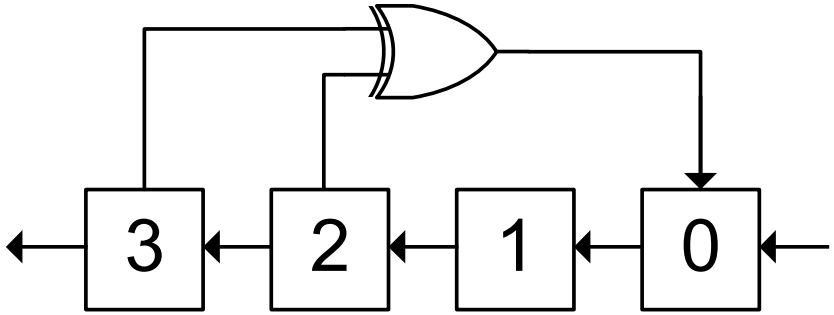
Linear Feedback Shift Registers (LFSRs) are widely used in digital systems for pseudo-random sequence generation, key expansion, error detection, and data scrambling. They are popular because they offer simple hardware implementation and good statistical properties. A key limitation, however, is their strictly sequential behavior: each new state depends on the previous one. As a result, reaching the Nth state requires N clock cycles, which adds delay in systems that need fast synchronization or random access to intermediate states. This constraint affects high-speed encryption engines and multi-clock-domain communication circuits. Existing approaches—such as storing precomputed states or running multiple LFSRs in parallel—help reduce latency but increase area and routing overhead. Mathematically, the challenge comes from the state-transition matrix L, since computing LN directly becomes impractical for longer LFSR sizes. This motivates the need for architectures that can compute Nth LFSR state transition efficiently while keeping hardware cost low.

Outline

- Introduction
- Problem Statement
- Conventional LFSR Approach
- Novel Approach
- Real Life Application
- Experimental Evaluation Results
- Conclusion

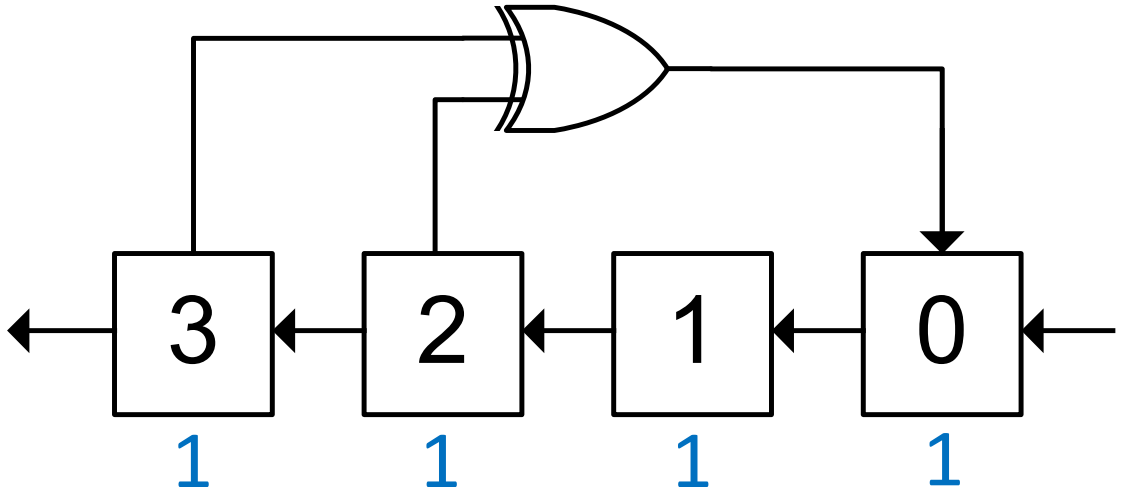
Introduction

- LFSR is a type of digital circuit made of flip-flops (Shift Register) and XOR gates.
- It generates sequences of 0s and 1s based on feedback from its previous state.
- Commonly used for generating pseudo-random numbers.
- Simple in structure, efficient in operation, and widely used in digital systems.
- Used for Scrambling Data.



Introduction (Cont.)

State	Bit 3	Bit 2	Bit 1	Bit 0
S0	1	1	1	1
S1	1	1	1	0
S2	1	1	0	0
S3	1	0	0	0
S4	0	0	0	1
S5	0	0	1	0
S6	0	1	0	0
S7	1	0	0	1
S8	0	0	1	1
S9	0	1	1	0
S10	1	1	0	1
S11	1	0	1	0
S12	0	1	0	1
S13	1	0	1	1
S14	0	1	1	1
S0	1	1	1	1



Problem Statement

- If application requires N^{th} state, conventional LFSRs requires multiple clock cycles to reach the N^{th} state.
- This delay can be a bottleneck in applications requiring fast access to specific LFSR state.
- N number of cycles required to generate N^{th} state of LFSR.
- This delay can limit the responsiveness of a system.

Conventional LFSR Approach

- The polynomial equation for the LFSR is $P(X) = X^5 + X^2 + 1$.
- Requirement is to obtain the 4th LFSR state in a single clock cycle.
- The conventional LFSR equation must be iteratively applied four times .i.e.,
  - The lfsr_1 state is derived from lfsr_0.
  - The lfsr_2 state is derived from lfsr_1.
  - The lfsr_3 state is derived from lfsr_2.
  - The lfsr_4 state is derived from lfsr_3.

Conventional LFSR Approach (Cont.)

- $P(X) = X^5 + X^2 + 1$

```
lfsr_1[4] = lfsr_0[0];
lfsr_1[3] = lfsr_0[4];
lfsr_1[2] = lfsr_0[3] ^ lfsr_0[0];
lfsr_1[1] = lfsr_0[2];
lfsr_1[0] = lfsr_0[1];
```

```
lfsr_3[4] = lfsr_2[0];
lfsr_3[3] = lfsr_2[4];
lfsr_3[2] = lfsr_2[3] ^ lfsr_2[0];
lfsr_3[1] = lfsr_2[2];
lfsr_3[0] = lfsr_2[1];
```

```
lfsr_2[4] = lfsr_1[0];
lfsr_2[3] = lfsr_1[4];
lfsr_2[2] = lfsr_1[3] ^ lfsr_1[0];
lfsr_2[1] = lfsr_1[2];
lfsr_2[0] = lfsr_1[1];
```

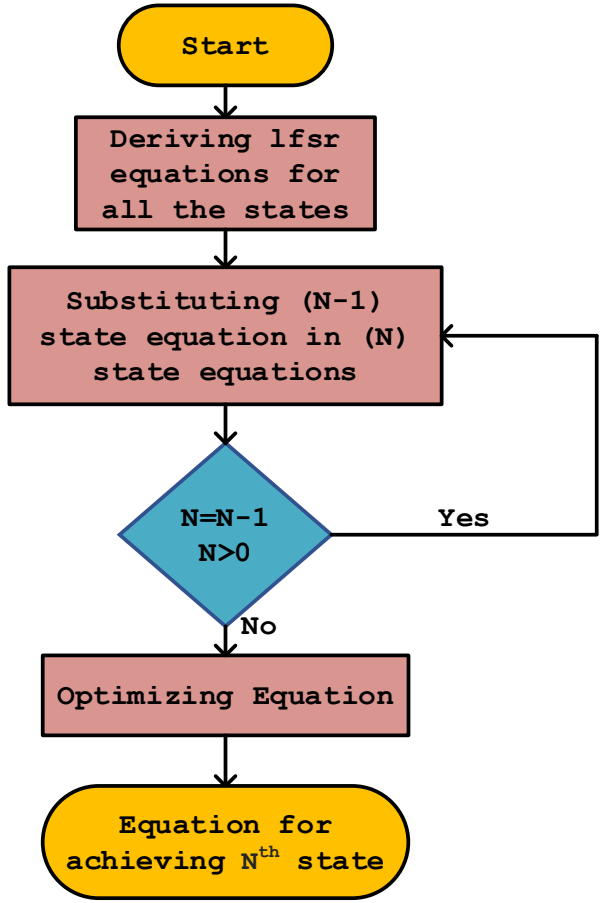
```
lfsr_4[4] = lfsr_3[0];
lfsr_4[3] = lfsr_3[4];
lfsr_4[2] = lfsr_3[3] ^ lfsr_3[0];
lfsr_4[1] = lfsr_3[2];
lfsr_4[0] = lfsr_3[1];
```

Solution Overview

- Proposed approach aims to directly access the desired N^{th} state output of a LFSR at the very next clock cycle.
- By analyzing the mathematical relationships between the LFSR's current state and its subsequent states.
- Derive an equation that directly calculates the N^{th} state output.
- This derived equation allows the system to bypass the internal state transitions and directly compute the N^{th} state value combinatorially.
- Providing the desired output on the first clock edge.

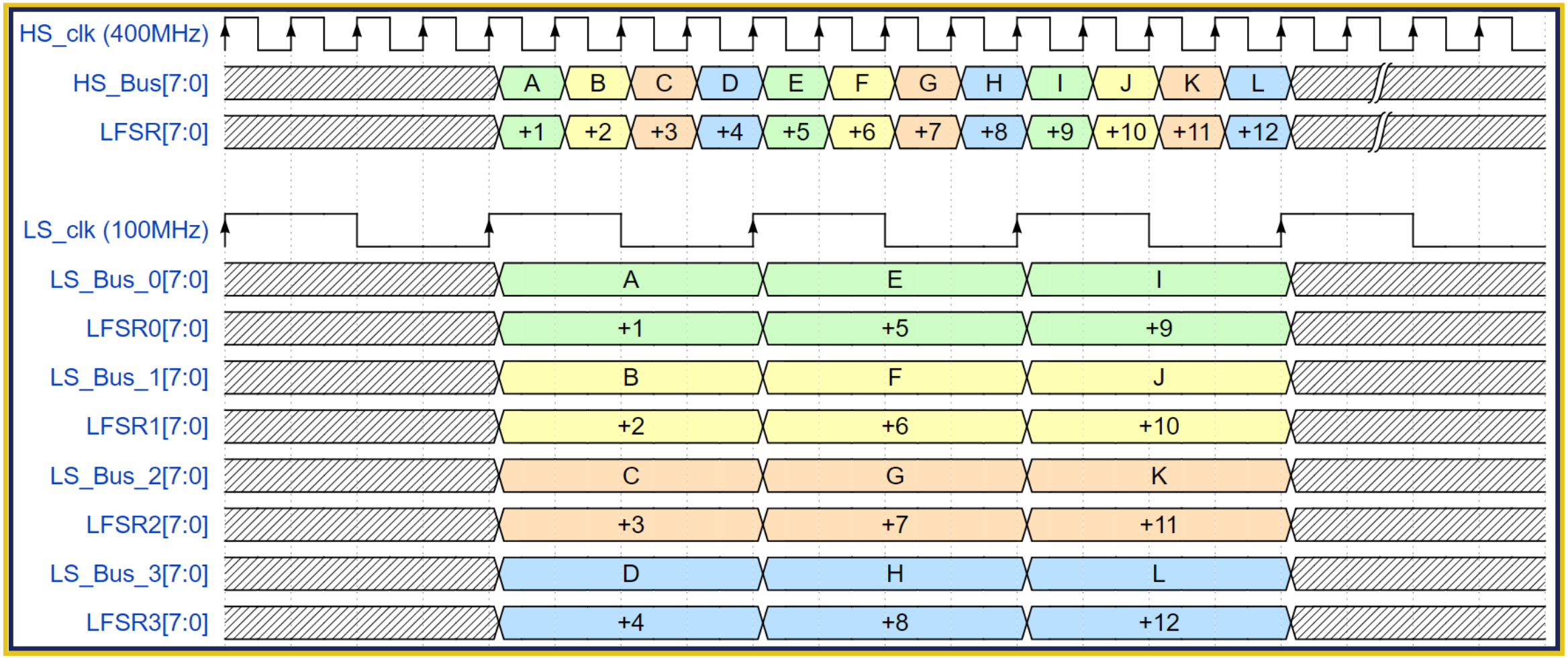
Solution Overview (Cont.)

- $P(X) = X^5 + X^2 + 1$

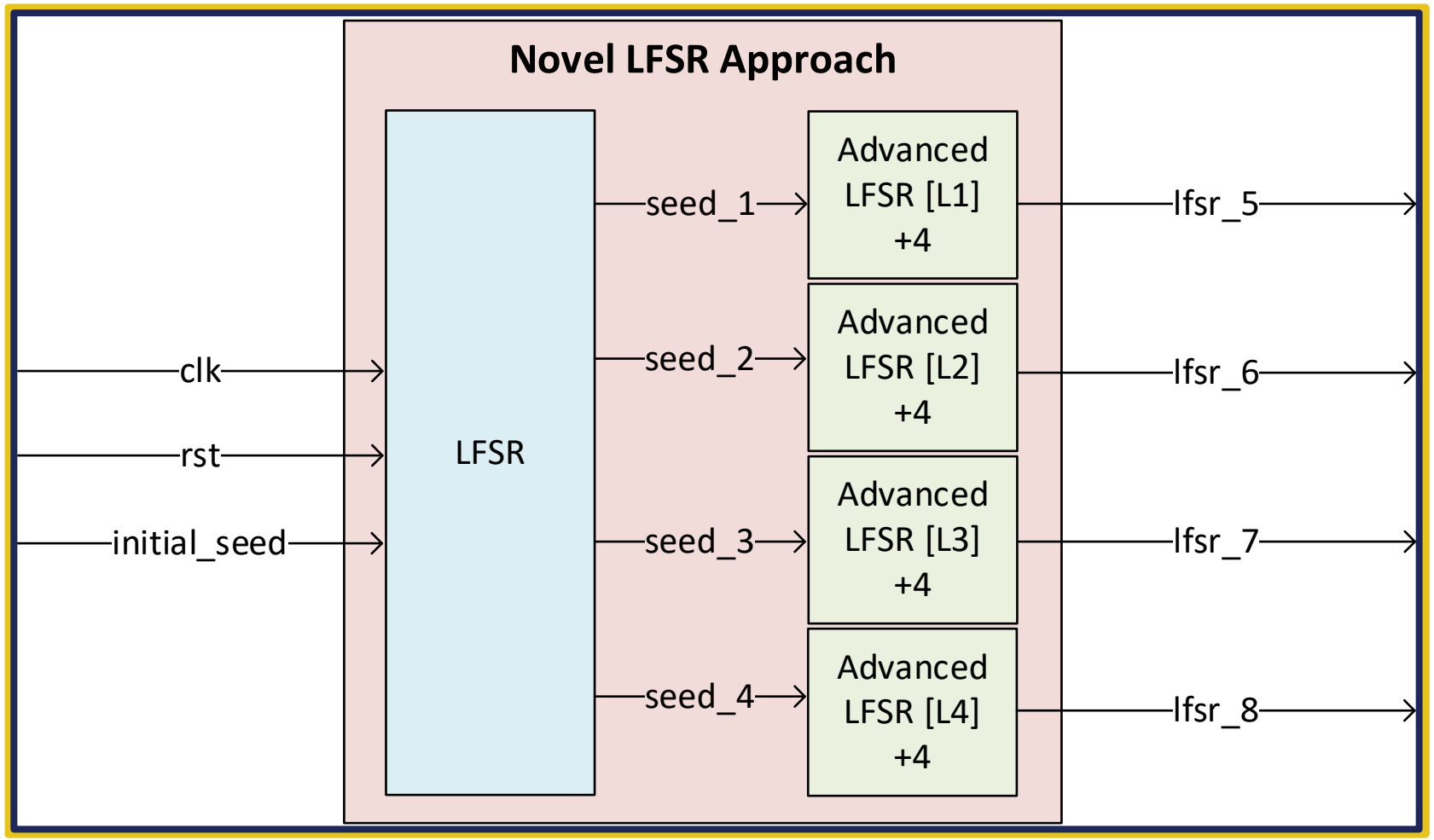


- **Enter the width**
  - 5
- **Polynomial Equation**
  - 10101
- **Value of Nth state Output**
  - 4
- **Output**
  - lfsr_4[4]=lfsr_0[3] ^ lfsr_0[0]
  - lfsr_4[3]=lfsr_0[2]
  - lfsr_4[2]=lfsr_0[1] ^ lfsr_0[3] ^ lfsr_0[0]
  - lfsr_4[1]=lfsr_0[0] ^ lfsr_0[2]
  - lfsr_4[0]=lfsr_0[4] ^ lfsr_0[1]

Real World Application



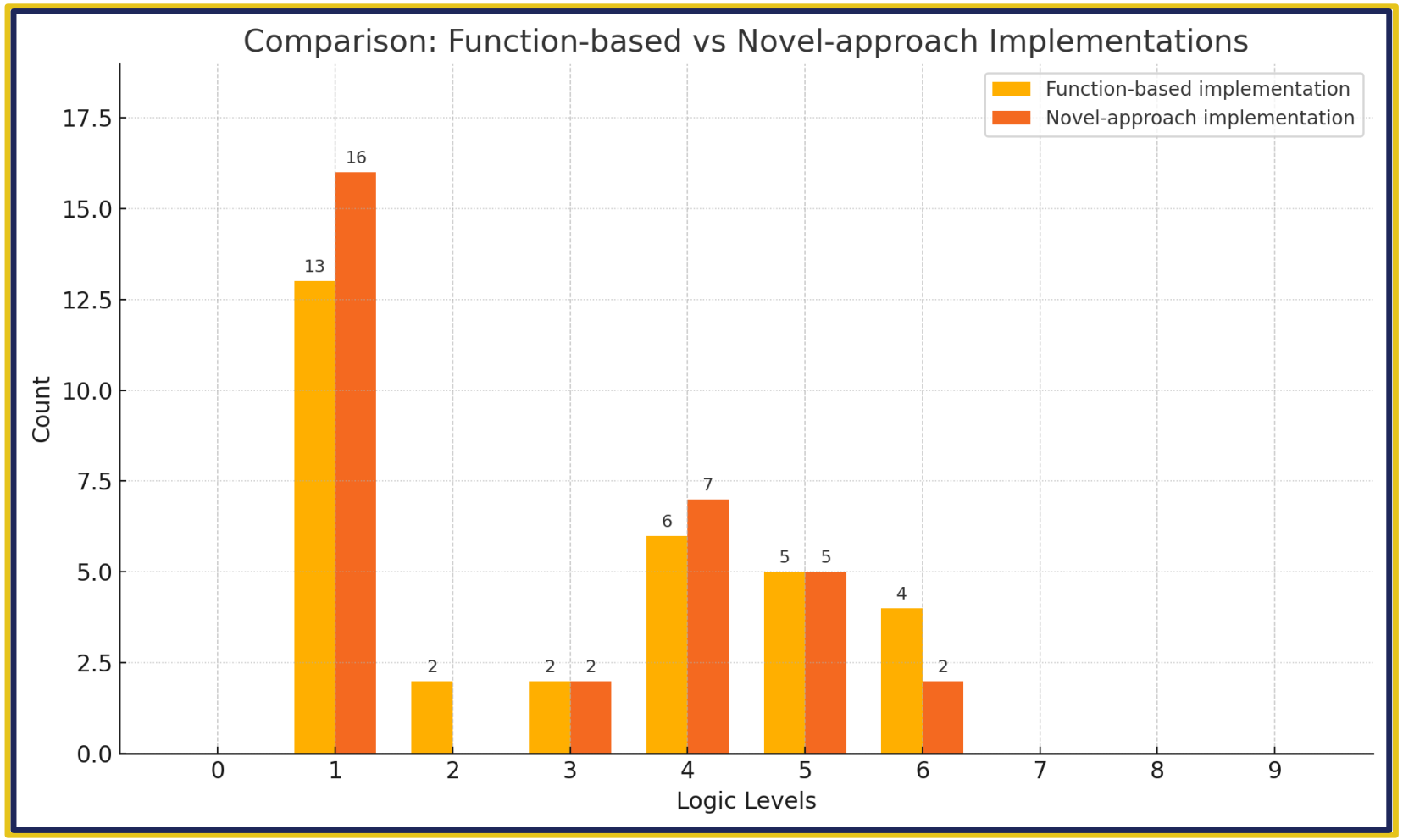
Architectural Overview



Experimental Evaluation

Technology Library	tcbn07_bwph30018p64pd_base_lvtssgnp_0p675v_m40c_cworst_CCworst_T 130
Cell Type	LVT
Operating conditions	ssgnp_0p675v_m40c_cworst_CCworst_T
Interconnect mode	Placement
Area mode	Physical library

Histogram Level



Synthesis Timing Results

- Function–based approach results:

Cost Group	Critical Path Slack	TNS	No of gates on Critical Path	Violating Paths
clk	0.2	0.0	5	0
default	No paths	0.0		
Total		0.0		0

- Novel approach–based approach results:

Cost Group	Critical Path Slack	TNS	No of gates on Critical Path	Violating Paths
clk	2.4	0.0	4	0
default	No paths	0.0		
Total		0.0		0

Area and Power Evaluation

Function–based approach results:

```
Instance Count
--------------
Leaf Instance Count              91
Physical Instance count           0
Sequential Instance Count        16
Combinational Instance Count     75
Hierarchical Instance Count       0

Area
----
Cell Area                                22.810
Physical Cell Area                       0.000
Total Cell Area (Cell+Physical)          22.810
Net Area                                 9.160
Total Area (Cell+Physical+Net)           31.970

Power
-----
Leakage Power                            7.587 nW
Dynamic Power                            243451.825 nW
Total Power                              243459.412 nW
```

Novel–based approach results:

```
Instance Count
--------------
Leaf Instance Count              90
Physical Instance count           0
Sequential Instance Count        16
Combinational Instance Count     74
Hierarchical Instance Count       0

Area
----
Cell Area                                22.925
Physical Cell Area                       0.000
Total Cell Area (Cell+Physical)          22.925
Net Area                                 9.114
Total Area (Cell+Physical+Net)           32.039

Power
-----
Leakage Power                            7.676 nW
Dynamic Power                            237539.883 nW
Total Power                              237547.559 nW
```

Floorplan Utilization and Physical Data

Function–based approach results:

Floorplan Utilization	74.26%
Max Fanout	18 (load_seed)
Min Fanout	1 (n_24)
Average Fanout	2.5
Terms to net ratio	3.1982
Terms to instance ratio	3.9011
Runtime	258.129073 seconds
Elapsed Runtime	553 seconds
Genus peak memory usage	3085.75
Innovus peak memory usage	4198.804688
Hostname	sjpipg116.cadence.com
Physical Data	
---	---
Total Net Length	294.98 um
Average Net Length	2.68 um
Routing Congestion	H: 0.00% V: 0.00%

Novel–based approach results:

Floorplan Utilization	72.05%
Max Fanout	18 (load_seed)
Min Fanout	1 (n_44)
Average Fanout	2.5
Terms to net ratio	3.1818
Terms to instance ratio	3.8889
Runtime	261.127622 seconds
Elapsed Runtime	560 seconds
Genus peak memory usage	3092.25
Innovus peak memory usage	4153.867188
Hostname	sjpipg116.cadence.com
Physical Data	
---	---
Total Net Length	293.72 um
Average Net Length	2.69 um
Routing Congestion	H: 0.00% V: 0.00%

Implementation Results

- Novel approach–based design shows consistently better performance, power, and structural efficiency.
- Logic-level histogram and 5 GHz timing analysis show a shallower, balanced structure with critical-path slack improving from +0.2 ns to +2.4 ns.
- The longest path is reduced by one gate, enabling higher achievable clock frequency and easier timing closure.
- Innovus™ results show ~2.4% lower dynamic power, similar total area, and improved floorplan uniformity.
- A single-cell reduction becomes significant in SoCs with hundreds/thousands of instances, improving total area, power, and routing.
- Overall, the novel approach provides a superior PPA trade-off and scales cleanly in high-performance digital systems.

Conclusion

- Conventional LFSRs cannot instantly access arbitrary internal states due to their sequential nature.
- The proposed advanced LFSR reformulates state transitions into direct-access equations, enabling constant-time Nth-state computation.
- Demonstrated in a multi-clock-domain system, it provides immediate scrambling-state alignment with zero latency.
- The architecture reduces hardware overhead, maintains continuous throughput, and supports high-speed applications.
- Advanced LFSRs become essential for modern systems requiring fast synchronization and deterministic state access.