Design of Reconfigurable Logic Block Based Sequential Circuits Using Look Up Table Logics

Sk. Haleem Muneesa[1], Jakkala Yoga Deepika[1], Obulam Yogendra Lakshmi Prasanna[1], Gummadi Sumasree[1], Shaik Thaslim[1]

[1]Department of Electronics and Communication Engineering, Geethanjali Institute of Science and Technology, Nellore.

Abstract: Reconfigurable sequential circuits find applications in various digital systems, including communication networks, data processing units, embedded systems, and FPGA-based designs. Their ability to adapt and reconfigure their functionality on-the-fly allows them to accommodate dynamic requirements and optimize the use of hardware resources. Traditional implementations of sequential circuits involve static configurations, where the logic and functionality are fixed during synthesis. While these methods are straightforward to design and implement, they lack adaptability and cannot be modified without redesigning the entire circuit. The proposed method involves the utilization of a dedicated Reconfigurable Logic Block (RLB) within the sequential circuits, allowing for dynamic configuration changes without altering the overall circuit structure. The RLB can be programmed to provide different logic functions using look up tables, multiplexers, enabling the sequential circuit such as counters and shift registers to change its behaviour.

Keywords: Sequential Circuits, Counters, Shift registers, Look up tables, Reconfigurable Logic Block, D Flip Flops.

1.Introduction

Counters in digital circuits are fundamental components used to generate sequences of binary numbers. They are widely employed in various applications such as frequency division, digital clocks, event counting, and addressing memory locations. Counters can be synchronous or asynchronous and come in different types such as binary counters, BCD (Binary-Coded Decimal) counters, and asynchronous counters. In VLSI (Very Large-Scale Integration), counters are digital circuits used to count events or sequences of events. They are widely used in various applications such as frequency synthesis, digital signal processing, and timing generation. Counters can be classified based on their type (e.g., binary, BCD, asynchronous, synchronous), counting direction (up or down), and triggering mechanism (e.g., synchronous, or asynchronous). They are typically composed of flip-flops and combinational logic gates arranged in specific configurations to achieve the desired counting behaviour.

Shift registers are fundamental components in digital circuits used for storing and shifting data serially. They are commonly employed in applications like data storage, serial-to-parallel conversion, parallel-to-serial conversion, and data manipulation.

- Serial-in Serial-out (SISO): Data is entered serially at one end and shifted out serially at the other end.
- Serial-in Parallel-out (SIPO): Data is entered serially and output in parallel.
- Parallel-in Serial-out (PISO): Data is entered in parallel and shifted out serially.
- Parallel-in Parallel-out (PIPO): Data is entered and output in parallel.

Shift registers in VLSI (Very Large-Scale Integration) refer to the implementation of shift registers using integrated circuit technology, typically using CMOS (Complementary Metal-Oxide-Semiconductor) fabrication processes. In VLSI design, shift registers are often optimized for area, speed, and power efficiency to meet the requirements of modern digital systems. Overall, shift registers play a crucial role in VLSI design, enabling the implementation of various digital functionalities in modern integrated circuits.

2. Literature Survey

Sanadhya, et.al [1] developed Power optimization method, which was an essential issue in designing digital circuits that are battery-powered and portable. The market demands and technical constraints have already necessitated the importance of efforts in advancing energy-efficient circuits. Govindaraj, et.al [2] developed Low-power designs, that are getting increased significance in numerous applications like high-performance computing and wireless communication due to the rise in power dissipation. Power dissipation of VLSI circuits in test mode was much higher than in the normal operation mode due to the high frequency of applied test patterns. Pomeranz, et.al [3] implemented Iterative synthesis consists of local design modifications to improve design parameters or correct design errors. Incremental test generation was suggested for evaluating the effects of the modifications on the testability of the design. Angadi, et.al [4] developed the term “built in self-test” (BIST) refers to a method of circuit design that facilitates self-testing. The circuit was put through its paces by generating test patterns in this system. Circuit Under Test (CUT) refers to the intended target of the testing.

Chandra, et.al [5] developed modern VLSI technology, power consumption was a key factor in all design decisions. Today's electronics sector has made low power a central feature. Power dissipation has changed dramatically because of the demand for low power, taking on equal importance to performance and space. Priyadarshini, et.al [6]

developed the design of a Novel D Flip Flop (DFPFP). This D Flip Flop circuit was examined using supply voltage level techniques. The main purpose of these methods was to lower the power consumption caused by leakage currents. Minakshi, et.al [7] developed digital circuits, energy reduction was the most important parameter in the design of handy and battery-operated devices. Flip- flop was an important component in any digital system. Shah, et.al [8] developed a sense amplifier-based flip flop (SAFF) was presented appropriate for high speed, high data activity and low power operations. Yinghua, et.al [9] developed the Boolean satisfiability (SAT) attack was an oracle-guided attack that can break most combinational logic locking schemes by efficiently pruning out all the wrong keys from the search space. Ramakrishnan, et.al [10] developed Nowadays, electronic systems are a significant part of human life. The sophistication and complexity of these systems increase with developing technology. Harikrishna, et.al [11] developed square root was a vital mathematical operation which has a lot of applications. Square rooters are used in computer graphics, global positioning system (GPS), digital signal processing (DSP) computations, mathematical calculations, and data processed.

Nousheen, et.al [12] developed This literature presents a novel hardware-oriented image compression algorithm and its very large-scale integration (VLSI) implementation for wireless sensor networks. The proposed novel image compression algorithm consists of a fuzzy decision, block partition, digital halftoning, and block truncation coding (BTC) techniques. Geonhwi, et.al [13] implemented a high-speed low-power CMOS synchronous up/down counter with a novel compact toggle flip-flop is proposed to achieve energy- and area-efficient speed enhancement. Moraitis, et.al [14] developed Field-Programmable Gate Array (FPGA) technology in recent years have resulted in an expansion of its usage in a very wide spectrum of applications. Irith, et.al [15] developed test data compression to enable compressed tests to be stored on a tester and decompressed on-chip. Despite the potential for test compaction beyond conventional scan-based tests, test data compression had not been applied before to transparent-scan sequences.

3. Proposed Methodology

Figure 1 shows the architecture of proposed counter. The detailed operation as follows

Step 1: When TD equals 0, it's crucial that the NOR gate output is also zero, activating the red-coloured path exclusively. This ensures that only specific operations occur. Conversely, if TD equals 1, the output of the LUT6_2 is mandated to be zero. In simpler terms, depending on the value of TD, different pathways are triggered: with TD at 0, the red path is exclusively activated by the NOR gate's zero output, while with TD set to 1, the LUT6_2 output becomes zero, steering operations accordingly.

Step 2: In this step, we navigate through LD, which represents the Load input. When LD equals 0, the system stores the Previous data, specifically Q_3. However, when LD equals 1, it switches to collecting external data. This means that depending on the value of LD, the system either retains the current data from Q_3 or retrieves fresh information from an external source to proceed with further processing

Step 3: When $\overline{U}$/D equals 0, the system executes up counting, incrementing values in a sequential manner. Conversely, when $\overline{U}$/D equals 1, it switches to down counting mode, decrementing values in a reversed sequence. Thus, depending on the state of $\overline{U}$/D, the system's counting behaviour adjusts accordingly, either progressing upward or downward through the numerical sequence.

Step 4: When $\overline{U}$/D equals 0, the XOR gate output mirrors that of the MUX, leading to an up-counter configuration. $\overline{U}$/D This complementation triggers a reverse counting sequence. The red-coloured line serves as the critical path, ensuring proper signal propagation and synchronization. In essence, the system dynamically adjusts its counting direction based on the state of $\overline{U}$/D, with the red path playing a pivotal role in maintaining operational integrity.

Step 5: TD dictates its functionality: when TD equals 0, it operates as a straightforward counting process, progressing through numerical sequences. However, if TD equals 1, it shifts focus to loading previous data, indicating a different operational mode, likely involving data manipulation or storage. Within this setup, two significant components stand out: the Look Up Table (LUT), which likely handles data processing and manipulation tasks, and the carry chain, which likely manages arithmetic operations or carries between sequential elements. These blocks play vital roles in executing the system's functions, contributing to its overall functionality and performance.

Step 6: The carry chain serves as a conduit, transmitting data from one stage to the next, ensuring seamless progression within the system. While all the Look Up Tables (LUTs) are interconnected through this chain, they operate independently, without dependence on other blocks. The LUT output consistently serves as one of the inputs for the carry chain, which also incorporates an output from a previous stage, facilitating continuous data propagation and processing throughout the system.

Step 7: The AND gate output from LUT6_2 serves as the selection line. When the selection line equals 0, the carry chain output mirrors LUT6_2's second output. Conversely, when the selection line equals 1, the carry chain output

matches the previous carry chain output. This mechanism, known as the loading operation, facilitates the transfer of data within the system, adapting the carry chain's output based on the current selection line state for subsequent processing.

Step 8: The system executes an XOR operation between the selection line and the previous output from the carry chain. This operation generates the final counting sequence, incorporating both the selected inputs and the carry chain's output to determine the next count.

Step 9: The last block in this diagram is D-Flip Flop. Stores the data output into D-Flip Flop, it will generate final output. The data flip flops serve as storage for the output of the LUT6 (Look Up Table 6). They retain this output, and collectively, they generate the final output of the system, consolidating the processed information for further use or transmission.

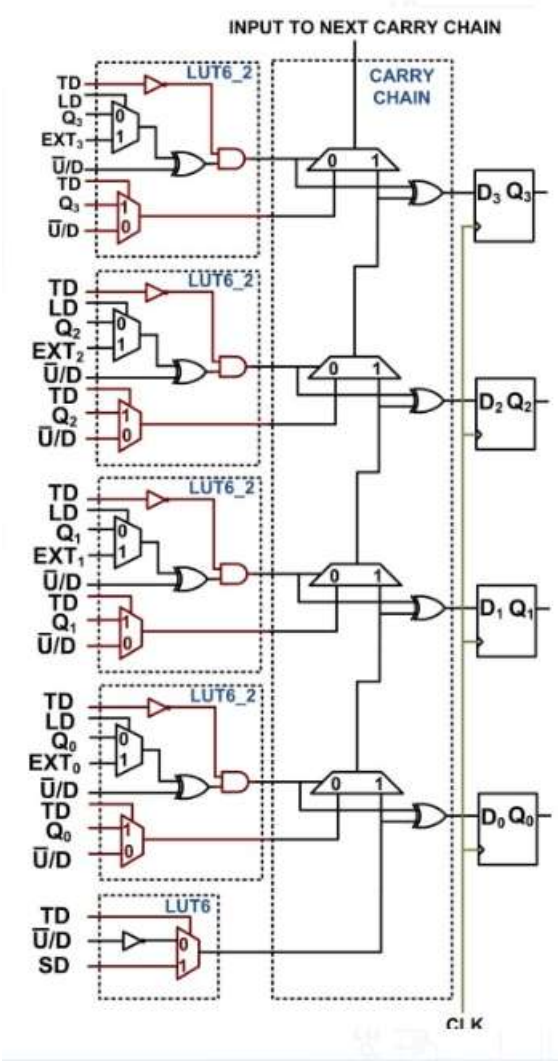


Figure 1: Proposed Counter Block Diagram.

Figure 2 shows the architecture of proposed shift register. The detailed operation as follows

Step 1: TD stands for Time Delay, which applies uniformly across all Look Up Tables (LUTs). When TD is set to 0, the Multiplexer (MUX) opts for the previous data. Conversely, when TD equals 1, the MUX switches to the next data set. This "next data" specifically refers to Q_{n-2}. In simpler terms, depending on the TD value, the MUX either fetches the data from the previous slot or skips ahead to the next slot, where the data is two slots back from the current one in the sequence.

Step 2: LD stands for Load Input. When LD is set to 0, the Multiplexer (MUX) opts for the next data available. Conversely, when LD equals 1, the MUX selects external data. This external data, denoted as Ext_{n-1}, essentially means data sourced from an external origin, and specifically, it refers to the data one step back from the current

external source. So, in simpler terms, depending on the LD value, the MUX either picks the next available data or switches to external data, which is one step back from the current external source.

Step 3: $\bar{L}$/R represents the Left Right Shift control. When $\bar{L}$/R is set to 0, the NOT gate output is activated, making it high, while the AND gate output shifts to Q_{n-2}, which signifies the Next State. Subsequently, the Multiplexer (MUX) executes a Left Shift Operation. Essentially, this means that when $\bar{L}$/R is 0, the system undergoes a leftward shift, utilizing the data from two steps back in the sequence. In simpler terms, based on the $\bar{L}$/R value, the operation shifts data leftward, utilizing the appropriate state information to drive the process.

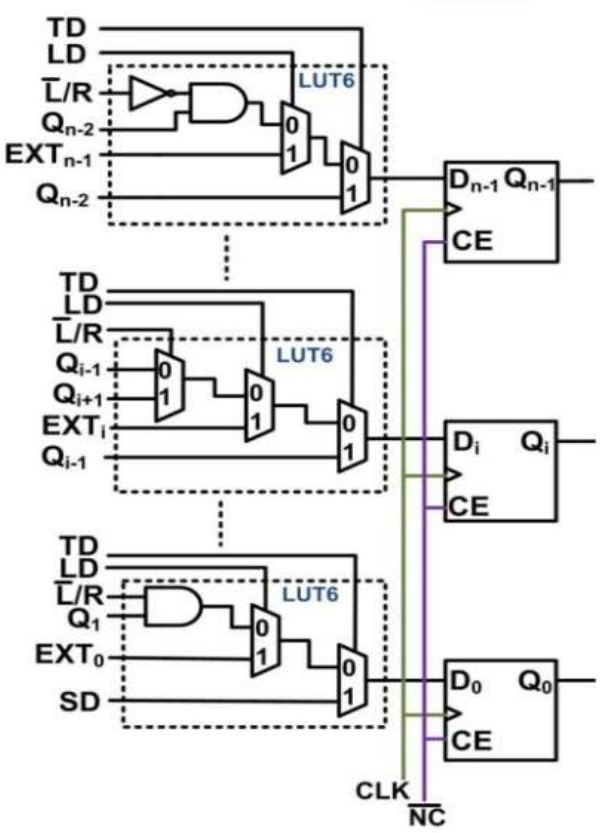


Figure 2: Proposed Shift Register Block diagram.

Step 4: Qn_2 denotes a specific data input, and it's generated through a feedback loop using previous outputs. Essentially, it represents information from two steps back in the sequence of outputs. Picture it as a sort of memory function within the system, where the current output influences the next state, but also reaches back to access data from two steps prior.

Step 5: Ext_{n-1} stands for external input data, which is loaded depending on the load input signal. This means that the external data used for processing is determined by whether the load input is active or not. These five steps outline the operation of the look-up table within the shift registers: first, it involves selecting the appropriate data based on time delay or load input conditions. Then, it executes operations such as left or right shifts, utilizing feedback to incorporate past outputs. Overall, this process enables the efficient retrieval and processing of data within the shift register system.

Step 6: $\overline{NC}$ represents the negation of the No Change condition. When $\overline{NC}$ is equal to 0, it signifies that the clear enable signal becomes low. Consequently, this means that there's no alteration in the data stored in the flip flops; they retain their current values. Conversely, if $\overline{NC}$ is equal to 1, the clear enable signal becomes high. This action results in the data flip flops being emptied or reset, effectively clearing out any stored information. So, depending on the state of $\overline{NC}$, the system either maintains its current data or resets it to an empty state.

Step 7: Q_{i+1} represents the input data for a right shift operation. It becomes enabled when $\bar{L}$/R is set to 1, indicating a right shift condition. This means that the system will utilize Q_{i+1} as input data when instructed to perform a right shift operation.

Q_{i-1} represents the input data for a left shift operation. It becomes enabled when $\bar{L}$/R is set to 0, signifying a left shift condition. In this case, the system will utilize Q_{i-1} as input data when instructed to execute a left shift operation.

Step 8: The data flip flops serve as storage for the output of the LUT6 (Look Up Table 6). They retain this output, and collectively, they generate the final output of the system, consolidating the processed information for further use or transmission.

4. Results and Discussion

Figure 3 shows the simulation outcome for the existing counter. Figure 4 provides a summary of the design characteristics of the existing counter. Figure 5 offers a summary of the power consumption of the existing counter. Figure 6 provides a summary of the time-related metrics for the existing counter.

Figure 3: Counter Simulation Outcome

Resource	Utilization	Available	Utilization %
LUT	32	133800	0.02
FF	32	267600	0.01
IO	69	500	13.80
BUFG	1	32	3.13

Figure 4: Counter Design Summary

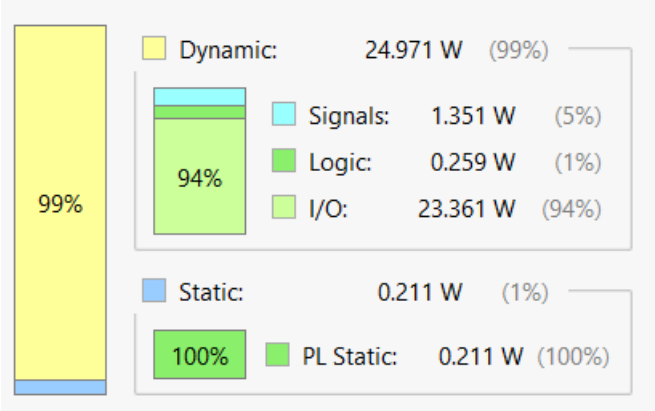


Figure 5: Counter Power Summary

Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 11	∞	3	1	2	count_reg_reg[10]/C	count_reg_reg[10]/D	0.555	0.346	0.209	-∞	
Path 12	∞	3	1	2	count_reg_reg[11]/C	count_reg_reg[11]/D	0.590	0.345	0.245	-∞	
Path 13	∞	3	1	2	count_reg_reg[15]/C	count_reg_reg[15]/D	0.590	0.345	0.245	-∞	
Path 14	∞	3	1	2	count_reg_reg[19]/C	count_reg_reg[19]/D	0.590	0.345	0.245	-∞	
Path 15	∞	3	1	2	count_reg_reg[23]/C	count_reg_reg[23]/D	0.590	0.345	0.245	-∞	
Path 16	∞	3	1	2	count_reg_reg[27]/C	count_reg_reg[27]/D	0.590	0.345	0.245	-∞	
Path 17	∞	3	1	2	count_reg_reg[31]/C	count_reg_reg[31]/D	0.590	0.345	0.245	-∞	
Path 18	∞	3	1	2	count_reg_reg[3]/C	count_reg_reg[3]/D	0.590	0.345	0.245	-∞	
Path 19	∞	3	1	2	count_reg_reg[0]/C	count_reg_reg[0]/D	0.599	0.349	0.250	-∞	
Path 20	∞	3	1	2	count_reg_reg[12]/C	count_reg_reg[12]/D	0.599	0.349	0.250	-∞	

Figure 6: Counter Time Summary.

Table 1 compares the existing and proposed counters across various metrics, including resource utilization (LUTs, FFs, IO ports), power consumption (dynamic, static, total), setup delay (total, logic, net), and hold delay (total, logic, net).

Table 1: Comparison table of existing and proposed counters for N=32.

Metric		**Existing Counter**	**Proposed Counter**
LUT		1286	32
FF		32	32
IO		69	69
BUFG		1	1
Dynamic Power (µW)		50.168	24.971
Static Power (µW)		0.586	0.211
Total Power (µW)		50.754	25.183
Setup delay	**Total**	113.892	12.185
	Logic	58.494	3.439
	Net	55.397	8.754
Hold delay	**Total**	1.107	0.599
	Logic	0.664	0.349
	Net	0.596	0.250

Figure 7 shows the simulation outcome for the proposed shift register. Figure 8 provides a summary of the design characteristics of the proposed shift register. Figure 9 offers a summary of the power consumption of the proposed shift register. Figure 10 provides a summary of the time-related metrics for the proposed shift register. Table 2 compares the existing and proposed shift registers across various metrics, including resource utilization (LUTs, FFs, IO ports), power consumption (dynamic, static, total), setup delay (total, logic, net), and hold delay (total, logic, net).

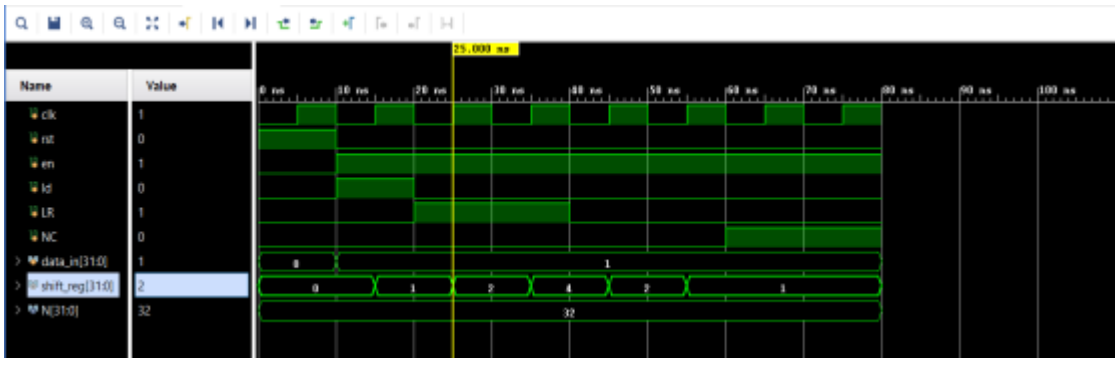


Figure 7: Proposed Shift Register Simulation Outcome

Resource	Utilization	Available	Utilization %
LUT	33	133800	0.02
FF	32	267600	0.01
IO	70	500	14.00
BUFG	1	32	3.13

Figure 8: Proposed Shift Register Design Summary

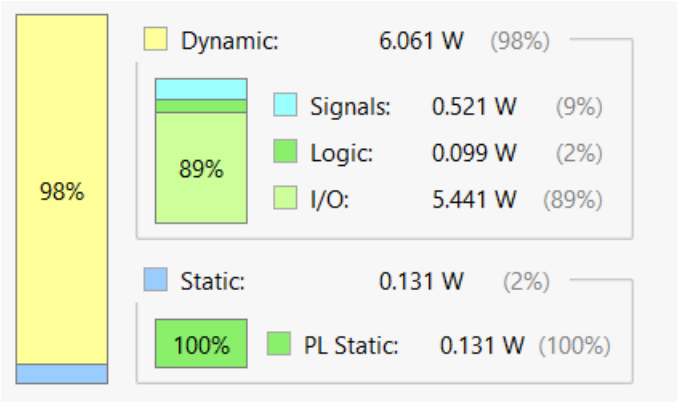
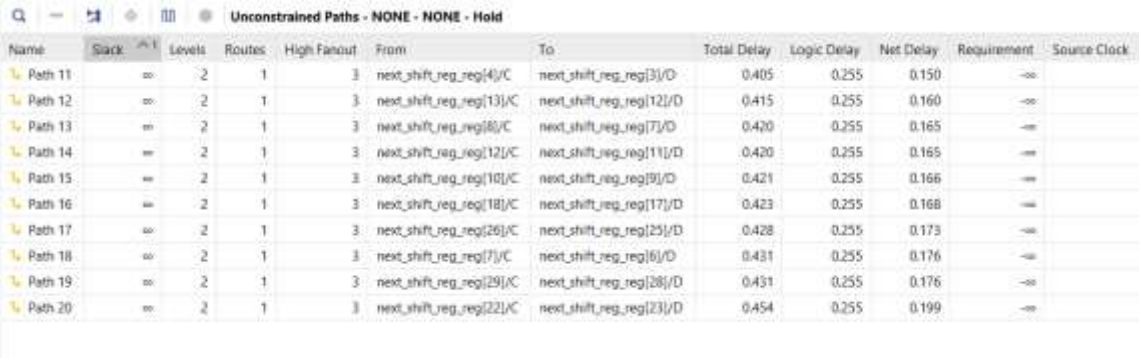


Figure 9: Proposed Shift Register Power Summary


Unconstrained Paths - NONE - NONE - Hold

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 11	∞	2	1	3	next_shift_reg_reg[4]/C	next_shift_reg_reg[3]/D	0.405	0.255	0.150	∞	
Path 12	∞	2	1	3	next_shift_reg_reg[13]/C	next_shift_reg_reg[12]/D	0.415	0.255	0.160	∞	
Path 13	∞	2	1	3	next_shift_reg_reg[8]/C	next_shift_reg_reg[7]/D	0.420	0.255	0.165	∞	
Path 14	∞	2	1	3	next_shift_reg_reg[12]/C	next_shift_reg_reg[11]/D	0.420	0.255	0.165	∞	
Path 15	∞	2	1	3	next_shift_reg_reg[10]/C	next_shift_reg_reg[9]/D	0.421	0.255	0.166	∞	
Path 16	∞	2	1	3	next_shift_reg_reg[18]/C	next_shift_reg_reg[17]/D	0.423	0.255	0.168	∞	
Path 17	∞	2	1	3	next_shift_reg_reg[26]/C	next_shift_reg_reg[25]/D	0.428	0.255	0.173	∞	
Path 18	∞	2	1	3	next_shift_reg_reg[7]/C	next_shift_reg_reg[6]/D	0.431	0.255	0.176	∞	
Path 19	∞	2	1	3	next_shift_reg_reg[29]/C	next_shift_reg_reg[28]/D	0.431	0.255	0.176	∞	
Path 20	∞	2	1	3	next_shift_reg_reg[22]/C	next_shift_reg_reg[23]/D	0.454	0.255	0.199	∞	

Figure 10: Proposed Shift Register Time Summary

Table 2: Comparison table of existing and proposed shift registers for N=32.

Metric		**Existing Shift Register**	**Proposed Shift Register**
LUT		540	33
FF		28	32
IO		68	70
BUFG		1	1
Dynamic Power (µW)		17.467	6.061
Static Power (µW)		0.168	0.31
Total Power (µW)		17.635	6.192
Setup delay	104.52	6.599	12.185
	51.841	1.218	3.439
	52.610	5.381	8.754
Hold delay	1.254	0.454	0.599
	0.531	0.255	0.349
	0.772	0.199	0.250

5. Conclusion

In the design of reconfigurable logic block-based sequential circuits using lookup table logic, successfully implemented counters and shift registers with a significantly reduced number of lookup tables, resulting in decreased area, power consumption, and delay. This proposed counter, for instance, utilizes only 32 lookup tables, a stark

contrast to the existing counter's 1286 lookup tables. This reduction in the number of lookup tables not only minimizes the area occupied by the counter but also contributes to a substantial decrease in power consumption. While the existing counter requires 50% of the total power, this proposed counter operates efficiently with just 25% of the total power. Additionally, the delay in this proposed counter is notably reduced compared to the existing counter. Similarly, in the realm of shift registers, this proposed design boasts a mere 33 lookup tables, a significant improvement over the existing shift register's 540 lookup tables. This reduction in the number of lookup tables translates to a more compact design with reduced area requirements. Furthermore, the power efficiency of this proposed shift register is remarkable, demanding only 6% of the total power, compared to the existing shift register's 17%. The delay in this proposed shift register is also considerably lower compared to its counterpart. In conclusion, this innovative approach to reconfigurable logic block-based sequential circuits has resulted in a substantial decrease in area, power consumption, and delay for both counters and shift registers. These advancements represent a significant leap forward in the efficiency and performance of these essential circuit components.

References

[1] Sanadhya, Minakshi, and Devendra Kumar Sharma. "Study of Adiabatic Logic-Based Combinational and Sequential Circuits for Low-Power Applications." In Low Power Architectures for IoT Applications, pp. 47-84. Singapore: Springer Nature Singapore, 2023.

[2] Govindaraj.V, S. Dhanasekar, K. Martinsagayam, Digvijay Pandey, Binay Kumar Pandey, and Vinay Kumar Nassa. "Low-power test pattern generator using modified LFSR." Aerospace Systems (2023): 1-8.

[3] Pomeranz, "Testability Evaluation for Local Design Modifications." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems (2023).

[4] Angadi, Aditi, Shreya Umarani, Totashri Sajjanar, Rakshita Karnam, Kotresh Marali, and Shrikanth Shirakol. "Architectural Design of Built in Self-Test for VLSI Circuits using LFSR." In 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), pp. 1-7. IEEE, 2023.

[5] Chandra, B. Ravi, Chinta Pranitha, Aunupati Ediga Preethi, Konkala Pavani, Mantriki Rajini, and Houdekari Mounika Bai. "Implementation of Ripple Carry Adder Using Full Swing Gate Diffusion Input." In 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 44-50. IEEE, 2023.

[6] Priyadarshini, V., M. Manaswini, M. Krishna Babu, and M. Mallika. "DESIGN OF LOW POWER AND HIGH-SPEED CMOS D-FLIPFLOP USING HYBRID LOW POWER TECHNIQUES."

[7] Sanadhya, Minakshi, and Devendra Kumar Sharma. "D flip-flop design by adiabatic technique for low power applications." Indonesian Journal of Electrical Engineering and Computer Science 29, no. 1 (2023): 141-146.

[8] Shah, Owais Ahmad, Geeta Nijhawan, and Imran Ahmed Khan. "A glitch free variability resistant high speed and low power sense amplifier-based flip flop for digital sequential circuits." Engineering Research Express 5, no. 3 (2023): 035046.

[9] Hu, Yinghua, Yuke Zhang, Kaixin Yang, Dake Chen, Peter A. Beerel, and Pierluigi Nuzzo. "On the Security of Sequential Logic Locking Against Oracle-Guided Attacks." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (2023).

[10] Ramakrishnan, Kannan, and K. Vidhya. "Design and Optimization of Sequence Generator Using NN Transvidkan Gate." (2023).

[11] Harikrishna, A., Chowdamjollu Sai Siddartha, Kokollu Venkatesh, Guduru Bose, and Burla Bhuvan Shyam Reddy. "A LOW POWER BINARY SQUARE ROOTER USING REVERSIBLE LOGIC." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 14, no. 2 (2023): 435-443.

[12] NOUSHEEN, MD SANA, and T. VIJAY KUMAR. "IMPLEMENTATION OF LOSSLESS IMAGE COMPRESSION USING FUZZY BASED MODIFIED GOLOMB RICE ENCODING." Journal of Engineering Sciences 14, no. 02 (2023).

[13] Geonhwi, Bomin Joo, and Bai-Sun Kong. "CMOS Clock-Gated Synchronous Up/Down Counter with High-Speed Local Clock Generation and Compact Toggle Flip-Flop." IEEE Transactions on Circuits and Systems I: Regular Papers (2023).

[14] Moraitis, Michail. "FPGA Bitstream Modification: Attacks and Countermeasures." IEEE Access 11 (2023): 127931-127955.

[15] Srinivasarao, G., Penchaliah, U., Devadasu, G. et al. Deep learning based condition monitoring of road traffic for enhanced transportation routing. J Transp Secur 17, 8 (2024). https://doi.org/10.1007/s12198-023-00271-3

[16] Irith. "Test Data Compression for Transparent-Scan Sequences." IEEE Transactions on Very Large-Scale Integration (VLSI) Systems 31, no. 4 (2023): 601-605.