Journal of Electronics and Power Engineering (JEPE)
Volume 02, Issue 01, Jan-Apr 2025, pp.54-66

Research Article

Implementation and Comparative Analysis of Various Low Power FSM Synthesis Techniques

Mehak Sharma[1], Vandana Khanna[2,*], Anu Tonk[3] and Shaifali Ruhil[4]

[1-4] Assistant Professor, Department of Multidisciplinary Engineering, The NorthCap University, Gurugram, India.

[*]Corresponding Author: Vandana Khanna. Email: vandanakhanna@ncuindia.edu

Received: 15/02/2025; Revised: 25/03/2025; Accepted: 20/04/2025; Published: 30/04/2025.

Abstract: This paper discusses the implementation and comparative analysis of the techniques to lower the power consumption in Finite State Machines (FSMs). FSM partitioning and clock gating have been discussed and incorporated in two modules: sequence detector and traffic light controller to investigate their performance in terms of power consumption and thus identify the most effective approach. The FSM Partitioning technique works in conjunction with the heuristic algorithm approach to decompose a single FSM into two separate FSMs. Thus, power consumption can be reduced since only one sub-machine is clocked at any given time while the other remains inactive, resulting in decreased power dissipation for the overall circuit. Additionally, the Clock Gating technique is introduced, which involves disabling the clock signal to the sequential blocks of the FSM during periods of inactivity or idleness. These techniques have been implemented and simulated in the aforementioned circuits using the XILINX_ISE_14.7. It has been found that FSM Partitioning results in a power saving of 72% and 87.5% when applied to a 5-bit sequence detector and a traffic light controller, respectively, compared to the conventional design. Similarly, applying clock gating techniques results in power savings of 33% and 75% for a 5-bit sequence detector and a traffic light controller, respectively, compared to the conventional design. This work presents two techniques that yield a good amount of power saving in both case studies of sequence detector and traffic light controller. Also, the better effectiveness of FSM partitioning technique has been established.

Keywords: FSM, Clock gating, FSM partitioning, Power consumption, Sequence Detector

1.Introduction

A Finite State Machine (FSM) is a device that can retain the current state of a specific entity at any given time. FSM experiences state transitions determined by input signals, resulting in specific outputs based on predefined rules [1]. At any given moment, only one state can be active within the machine. The FSM is designed to transition between different states to perform various actions. Coin-operated turnstile, traffic lights, a safe are few examples of FSM which are operated on states.

FSMs can be categorized into two primary types:

Every potential input in every state of a deterministic finite state machine (DFSM) [2] has a unique transition. The input and present state unquestionably determine the upcoming state. Although this kind of FSM is easy to comprehend and use, complex systems could need additional states and transitions.

Nondeterministic Finite State Machines (NFSMs) [3] allow for more than one transition from a given state for a certain input. Since the next state is selected nondeterministically, the machine may take more than one alternative route. Although they offer greater flexibility, NFSMs can be more difficult to analyze and put into practice.

Finite State Machines can also be classified as Moore and Mealy. In the first, the outputs are synchronized with the circuit clock, and are dependent on current state. Conversely, in the second one, the output values are determined by both the current state and the current inputs of the machine.

FSMs find application in various domains, including the design of digital circuits [4], modelling protocols and communication systems, controlling robots and automation processes, as well as parsing and analysing languages. They provide a structured and systematic approach to modelling and understanding systems with discrete behaviour, enabling the design and analysis of complex systems in a manageable and scalable manner.

Authors in [5] have touched the important topics of computer communication systems, and the formal specification technology of communication protocols. The FSM has been widely utilized in the description, realization, and testing of communication protocols because it is an intuitive, understandable, and simple to use description approach. It can successfully increase the accuracy and integrity of particular systems. This paper also provided the design and implementation of FSM based on BlueStack.

Ari and Mondada [6] have discussed that the robots have memory-equipped embedded processors that can be utilized to store the state of an algorithm at the current time. Finite state machines outline the circumstances in which the robot's state can change and the subsequent actions that must be executed. Battenberg cars have been used to show finite state machines in the beginning, followed by an algorithm that directs the robot to look for and then approach an item.

As the size of the chip is getting smaller with the increase in latest technologies day by day, the power consumption of devices has become a major issue. During stable logics in a circuit the current which is flowing between power lines causes the static dissipation while, the switching of circuit’s system clock causes the dynamic power dissipation. Various low power techniques have been proposed for FSMs in literature. Barkalov et.al [7] has provided a detailed review on the various approaches used to reduce power consumption in FSMs. The authors have discussed many techniques developed or proposed in 40 years to reduce static or dynamic power dissipation or both of these. Most important techniques discussed are clock gating, FSM decomposition and replacement of Look-Up Tables (LUTs) to Embedded Memory Blocks (EMBs).

Lector based clock gating technique is used by Bhattacharjee and Majumder in ref. [8]. The design style is called Leakage Control Transistor (LECTOR). Here, they have applied this technique to reduce the static and dynamic power consumption. A serial adder which is one of the essential components within processor micro-architecture, it is a victim of immense power flow. The LECTOR technique minimizes the static power by stopping current in the middle of the energy line, the gated clock eliminates dynamic power and ends unwanted switching of the system.

Ashutosh et. al. [9] have discussed the use of clock gating technique to stop the clock signal to the inactive portion of the circuit and hence achieve the reduction in the power consumption. Various clock triggering mechanisms in the most common sequential circuits of ALU and FIFO have also been discovered.

Saleem and Khan [10] proposed an approach to decrease the power consumption of Field Programmable Gate Array (FPGA)-based digital circuits at the FSM design level. Given the widespread use of FPGAs in various circuits in VLSI, power consumption has become a growing

concern at multiple circuit levels, including hardware, board, design, routing, and placement levels. The authors employ the clock gating technique in their study, utilizing control signals at the FSM level. This technique helps mitigate issues such as clock swapping and state changes, ultimately leading to a reduction in dynamic power consumption within the systems.

In an effort to reduce overall power dissipation, Nag et. al. [11] have proposed the integration of clock gating and power gating techniques in FSMs. Their approach involves cutting off the power supply to the FSM when it is not in use. By identifying indicators of the idle phase, such as self-loops within the FSM or a consistent FSM output between consecutive clock pulses, the power gating technique effectively shuts down the power supply. Additionally, during these periods of inactivity, clock gating is employed to disable the clock signal to the FSM's sequential blocks. This combined approach aims to optimize power consumption in FSMs.

Mihhailov et.al. [12] proposed a Java based web tool to encode the FSM with the objective of implementing a low power FPGA. This software used the concept of stochastic estimations which can be used for low power synthesis problems for various applications. Li and Choi [13] used FSM Partitioning as an effective method to reduce the power consumption and losses. This paper proposes big performance algorithm state transition probabilities and small complexities controlling logics for implementing the partitioned FSM, as mostly 1 sub-FSM should be clocked, hence power gets saved.

Merging and splitting of internal states in an FSM design which is the initial state of FSM synthesis is proposed in [14]. The splitting of internal states reduces the activity and hence power consumption. FSM partitioning suggested by researchers is effective if it is integrated with power gating to turn off the inactive part. Using the integrated approach needs an optimized process of partitioning and encoding of FSM, for which Genetic Algorithm (GA) approach has been developed in [15].

A significant portion of the literature found on FSM power reduction methods are on the basis of theoretical studies only. Not much work has been found on comparing different techniques with the help of experimentation on Xilinx software, which is the main motivation for this research work.

The paper is structured as follows: The methodology is described in Section 2, followed by a discussion of various FSM low power techniques in Section 3. Two types of FSM circuits used in this study are discussed in Section 4. The implementation is covered in Section 5, the results are discussed in Section 6, and the conclusion is presented in Section 7

2.Methodology

In this work, the Sequence detector and Traffic signal controller circuits have been designed using two low power techniques- Clock Gating and FSM Partitioning. The flowchart in Fig.1 highlights the stepwise implementation of the FSM. Firstly, FSM Sequence detector and Traffic signal controller modules have been analysed which was followed by incorporation of aforementioned low power techniques. Xilinx software has been used to carry out this study. Dynamic power dissipation was found and compared for all the four scenarios:

1) Sequence detector with FSM Partitioning
2) Sequence detector with Clock Gating
3) Traffic Light Controller with FSM Partitioning
4) Traffic Light Controller with Clock Gating

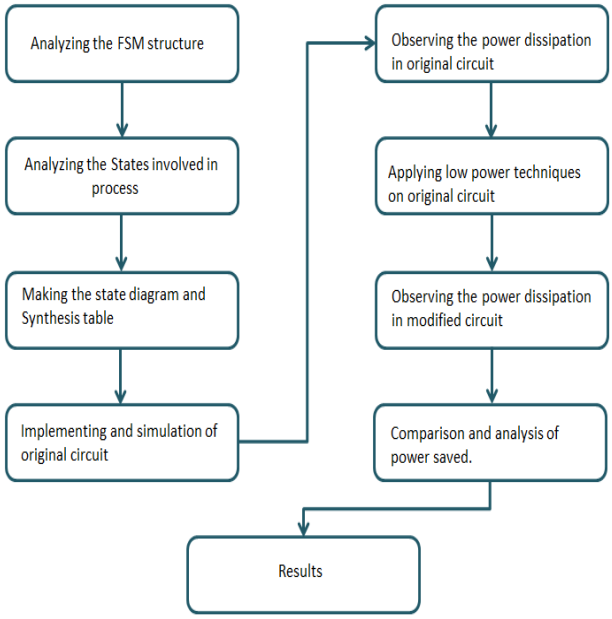

Figure 1: Circuit Design Flow

3.Low Power Techniques

FSM Partitioning: FSM partitioning is a process used in digital system design to divide a complex FSM into smaller, and manageable components. It involves decomposing the original FSM into several smaller FSMs called partitions, each responsible for a specific subset of states and transitions. The primary objective of FSM partitioning is to streamline the design process and enhance the implementation efficiency and keep the power consumption levels under control [16]. By dividing the finite state machine into smaller, manageable parts, designers can focus on individual components, reducing complexity and allowing for easier verification, testing, and debugging. Here is a step-by-step overview of the FSM partitioning process:

a) *State Identification*: Identify the states of the original FSM and analyse their functionalities. Determine the essential states and any redundant or unreachable states.

b) *State Clustering*: Group the states based on their functionalities or behaviour. States with similar characteristics or operations are clustered together.

c) *Transition Analysis*: Analyse the transitions between the clustered states. Identify the input conditions or events that trigger the transitions.

d) *Partition Creation*: Create partitions by assigning states and transitions to each partition. Ensure that the partitions are non-overlapping; meaning each state and transition belongs to only one partition.

e) *Interface Definition*: Define the interfaces between the partitions. This includes determining how information will be exchanged between partitions and any synchronization mechanism required.

f) *Partition Implementation*: Implement each partition separately, using suitable hardware or software description languages.

g) Integration: Integrate the individual partitions to reconstruct the original FSM. Ensure that the partitions work together correctly and meet the system requirements.

FSM partitioning offers several benefits, including a) Modularity: Partitioning enables the design to be divided into manageable modules, simplifying development and maintenance; b) Scalability: Partitioning allows for easy expansion or modification of individual components without impacting the entire system; c) Reusability: Partitions can be reused in other designs, reducing development time and effort; d) Verification: Smaller partitions are easier to verify and debug, improving the overall quality of the design.

Clock Gating: Clock gating is a power-saving method. Its objective is to minimize power used by selectively disabling the clock signal to specific sections of a circuit that are not actively involved in productive tasks. This allows for more efficient use of power resources. The clock signal serves as the primary timing reference in a digital system, controlling the sequential execution of operations within the circuit [17-19]. To implement clock gating, a gating circuit is inserted between the clock source and the target registers or functional units. This gating circuit consists of logic gates that determine whether the clock signal should be allowed to pass through or be blocked. When the clock is blocked, the targeted elements cease operation, effectively conserving power by preventing unnecessary consumption. The decision to gate the clock is typically based on certain conditions or control signals within the circuit. For instance, if a particular section of the circuit does not need to operate under specific conditions or when there is no meaningful input data, the clock signal to that section can be gated off to save power. Once the conditions change and the section needs to resume operation, the clock gating circuit enables the clock signal, allowing the circuit to function again. Clock gating is implemented on various steps of granules, going to fine-grained gating of individual flip-flops or logic gates to coarse-grained gating of larger blocks or subsystems. Fine-grained gating offers precise control over power consumption but may entail additional complexity and area overhead. On the other hand, coarse-grained gating is simpler to implement but may not achieve the same level of power savings. In summary, it's an effective way employed in designs to minimize power consumption. By selectively controlling the clock signal, unnecessary switching and activity in inactive circuit components are minimized, resulting in significant power savings.

4.FSM Circuits

Sequence detector and Traffic signal controller circuits are used in this work.

Sequence Detector: A sequence detector refers to a digital circuit utilized to identify a specific pattern or particular sequence of bits from a continuous stream of digital data. Sequence detectors have a wide-range of applications across various fields, including telecommunications, data processing, and pattern recognition [20]. It is typically used in communication systems, to detect the start or end of a transmission, or to detect special control codes that indicate the start of a packet or the end of a packet.

Primary goal of a sequence detector is to detect the occurrence of a particular sequence of symbols, which can be either a fixed pattern or a variable pattern with certain constraints. The symbols can be binary digits (0s and 1s) or characters from a larger set, depending on the application. Sequence detectors can be implemented using various techniques, such as state machines, finite state machines or regular expressions. These techniques involve defining a set of states and transitions based on the desired sequence pattern. The input stream is processed sequentially, and the detector transitions depending upon the current input symbol and the

defined transitions. For example, let's consider a simple sequence detector for a binary input stream. Suppose we want to detect the sequence "101" in the stream. The detector would have three states: State 0, State 1, and State 2. Initially, the detector is in State 0. Upon receiving a '1', it transitions to State 1. If the next input is '0', it remains in State 1. Finally, if the next input is '1', the detector transitions to State 2, indicating that the desired sequence has been detected. If at any point the input deviates from the expected pattern, the detector can reset to State 0 or a designated initial state to start looking for the sequence again.

Traffic Signal Controller: A device used to regulate the operation of traffic signals at an intersection is a traffic signal controller. It is normally mounted in a cabinet close to the intersection and wired to the signal system. Inputs from the traffic signal system, such as the presence of vehicles, are monitored by the controller, who then makes use of this data to regulate signal timing and phasing. It is a device that manages the operation of traffic signals at intersections [21]. Its primary function is to regulate the flow of vehicular and pedestrian traffic to ensure safety and efficiency on the roads. The traffic signal controller monitors the current traffic conditions and makes decisions on when and for how long each traffic signal should display a green, yellow, or red light. It operates based on predetermined timing plans or in response to real-time traffic detection systems. The controller receives inputs from various sources, including vehicle detectors, pedestrian push buttons, and centralized traffic management systems. These inputs help the controller assess the traffic volume and patterns at an intersection and determine the appropriate signal timings. Depending on the intersection's complexity and traffic demands, different types of traffic signal controllers are used. They can range from simple fixed-time controllers, where the signal timings are preprogrammed and don't change, to more advanced controllers that use real-time data to adapt the signal timings dynamically. Traffic signal controllers play a crucial role in optimizing traffic flow and reducing congestion. By coordinating the timing of traffic signals at multiple intersections, they can create green corridors, allowing vehicles to travel smoothly along a corridor without excessive stops and delays.

5.Implementation

5.1 Applying FSM partitioning and Clock gating on Sequence Detector

FSM partitioning technique is firstly implemented to optimize the two digital systems. Decomposition of FSMs is done to reduce the power consumption. Heuristic algorithm approach is used to decompose the FSMs into two different state machines operating one after another. State diagrams of original FSM and Decomposed FSMs are shown in Fig. 2 and Fig. 3.

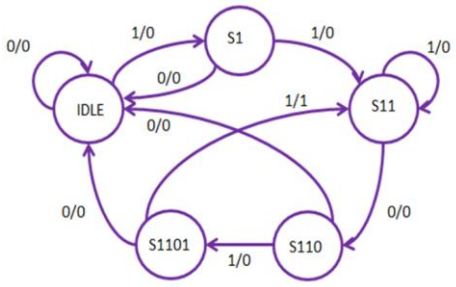

Figure 2: State Diagram of Original FSM (Mealy machine) for 11011 Sequence Detector

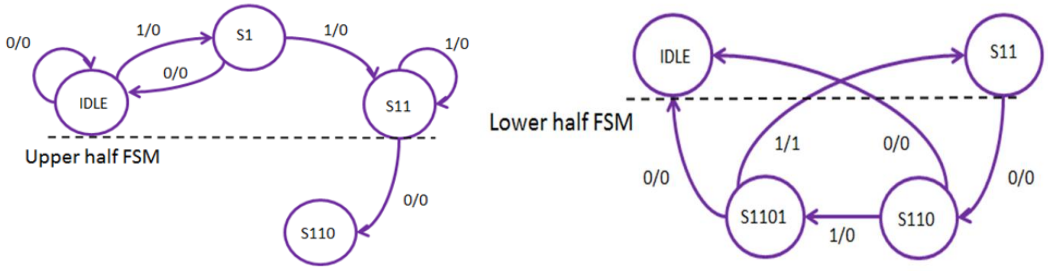

Fig. 3 State Diagrams of Decomposed FSM (a) Upper half and (b) Lower half

A sequence detector has been designed to detect the sequence "11011". Table 1 shows the truth table of the sequence detector. As seen in Fig. 2, there are 5 states i.e. IDLE, S1, S11, S110, S1101 and a series of inputs are provided to progress our sequence detector circuit until the desired output sequence is detected. Decomposed FSMs and their state diagrams are shown in Fig. 3. Fig. 3(a) shows the upper half of the sequence detector which detects 110 sequences while Fig. 3(b) shows the lower half of the sequence detector which detects the remaining sequence. Further, Heuristic Algorithm approach is used to decompose original FSM circuit.

Table 1: Truth Table for 11011 Sequence Detector

	CURRENT STATES			I/P		NEXT STATES			O/P
States	Y2	Y1	Y0	X	States	Y2'	Y1'	Y0'	Z
IDLE	0	0	0	0	IDLE	0	0	0	0
IDLE	0	0	0	1	S1	0	0	1	0
S1	0	0	1	0	IDLE	0	0	0	0
S1	0	0	1	1	S11	0	1	0	0
S11	0	1	0	0	S110	0	1	1	0
S11	0	1	0	1	S11	0	1	0	0
S110	0	1	1	0	IDLE	0	0	0	0
S110	0	1	1	1	FIN	1	0	0	0
FIN	1	0	0	0	IDLE	0	0	0	0
FIN	1	0	0	1	S11	0	1	0	1

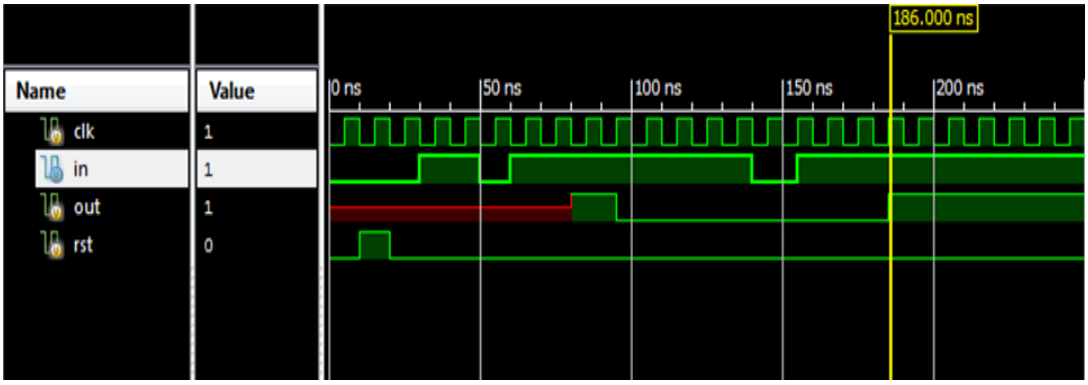

Figure 4: Output waveform for original 11011 Sequence Detector

Fig. 4 displays the 11011 sequence detector's waveform without any use of FSM low power technique. Input is denoted as in, output is denoted as out, clock signal is denoted as clk, and reset signal is denoted as rst. Figure 5(a) and Fig. 5(b) display the partitioned upper half and lower half of the sequence detector's simulated waveforms respectively. The clock signal is shown as clk, and the reset signal is shown as reset. The input signal is shown as data. The output signal is indicated as *detected* for the upper half of decomposed FSM and indicated as *found* for the lower half of decomposed FSM. Upon the discovery of the 110 sequence, the output will be high in Fig. 5(a) and as soon as the 011 sequence is identified, the output will be high in Fig. 5(b).

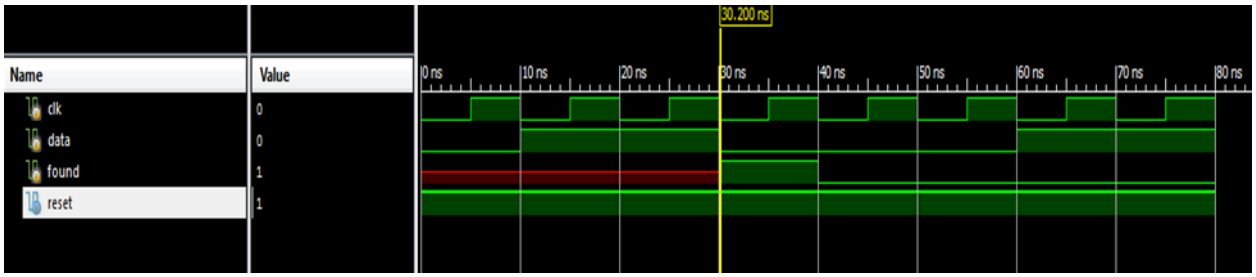

Figure 5(a): Output waveform of Upper half of decomposed FSM after simulation

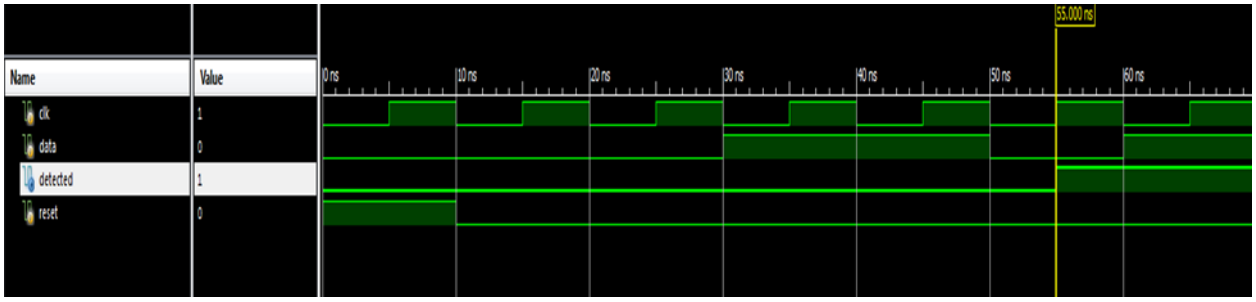

Figure 5(b): Output waveform of Lower half of decomposed FSM after simulation

Further, a Verilog code [18] is used to implement a sequence detector circuit with clock gating technique. This reduces the power consumption by updating the state machine and output signals when the clock signal is passed through the clock gating logic. The clock gating logic is implemented using a wire enable that's set high when the state machine is in state 3'b110. This wire controls the clock signal to pass through the state machine and output is detected only at desired state, reducing the power consumption. Clock gating technique is used to conserve power in digital circuits by selectively disabling the clock signal to certain components or portions of a circuit when they are not in use. Applying clock gating to a sequence detector can help in reducing power consumption when the detector is not actively searching for a specific sequence. Fig. 6a shows the technology schematic of the sequence detector circuit after applying clock gating on it. It represents the circuit in terms of logic elements optimized to the sequence detector circuit. Fig. 6b depicts the output waveform of a sequence detector after applying clock gating. The input signal is represented as data, the output signal as output_seq, the clock signal as clk, and the reset signal as reset signal.

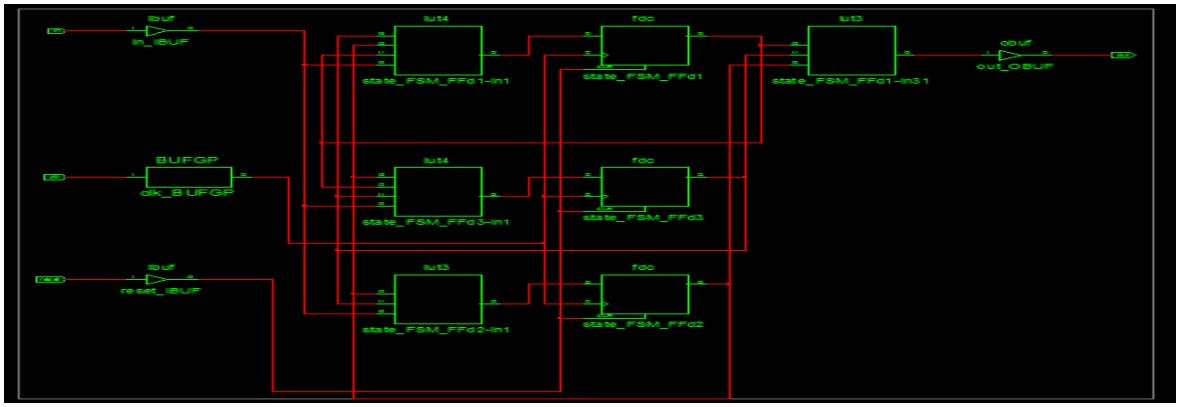

Figure 6(a): Schematic of Sequence detector circuit with clock gating

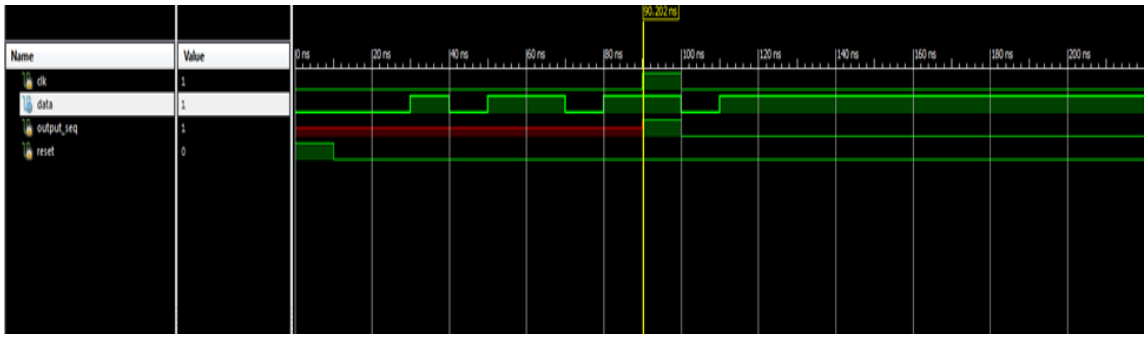

Figure 6(b): Output waveform of FSM after clock gating

5.2 Applying FSM partitioning and Clock gating on Traffic Signal Controller

Consider a traffic controller designed for an intersection between a main highway and a country road as shown in Fig. 7. The traffic signal prioritizes the main highway due to its continuous car flow, keeping the primary traffic light green. Occasionally, cars from the country road approach the traffic signal. In such cases, the country road signal should turn green for a sufficient duration, allowing the cars to proceed. Once there are no cars on the country road, the signal transitions from green to yellow and then red, while the main highway signal returns to green. The controller utilizes a sensor to detect cars waiting on the country road, sending a signal labelled X. When cars are present, X is set to 1; otherwise, it is set to 0. The transitions from S1 to S2 to S3 and from S4 to S0 involve adjustable delays to accommodate the necessary timing. The original FSM is divided into two smaller state machines in the instance of the traffic signal controller code, each of which is in charge of managing the traffic signals for a single direction of traffic flow. The signals used to communicate between the two smaller state machines show when each machine is in a condition that permits traffic to flow in a specific direction. The traffic_signal_controller module uses two smaller modules, north_south_fsm and east_west_fsm, to implement the partitioned FSM. Each of these smaller modules is responsible for controlling the traffic signals for the north-south and east-west directions, respectively. The modules "north_south_fsm" and "east_west_fsm" share similar architectures, comprising state registers and a combinational circuit that determines the next state and output signal values based on the current state and input signals. These individual state machines interact with each other through signals named "ns_to_ew" and "ew_to_ns," which indicate the availability of traffic flow in specific directions when each machine is in a particular state.

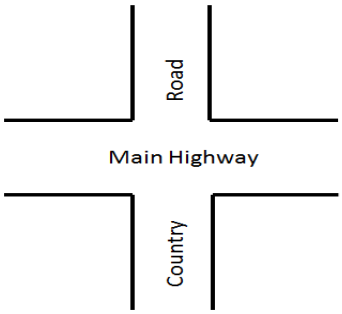

Figure 7: Intersection of two roads for a typical highway and a country road

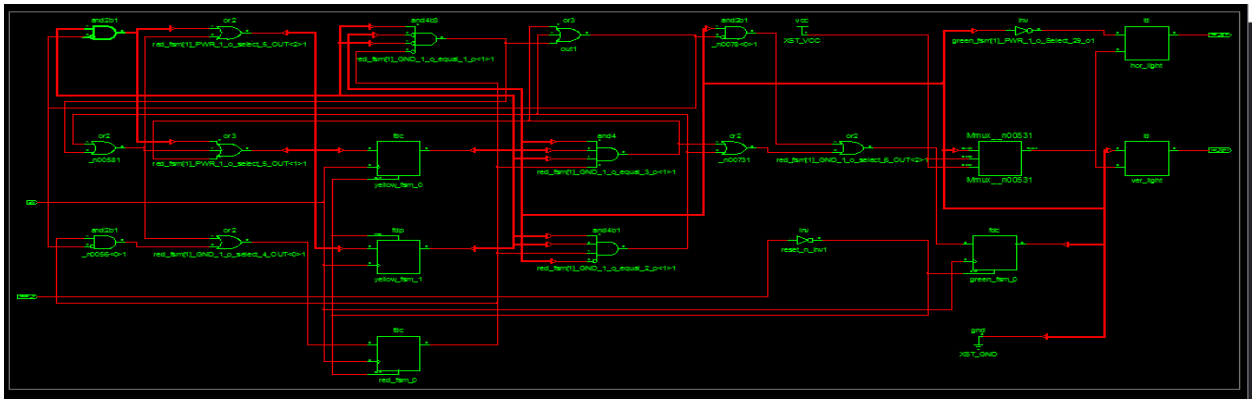

Figure 8(a): Technology Schematic Circuit for Traffic Signal Controller after FSM Partitioning

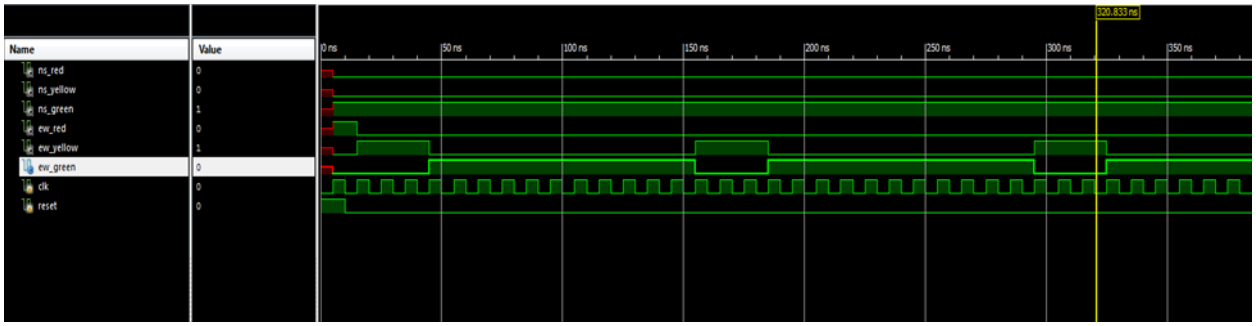

Figure 8(b): Output waveform of conventional Traffic Signal Controller

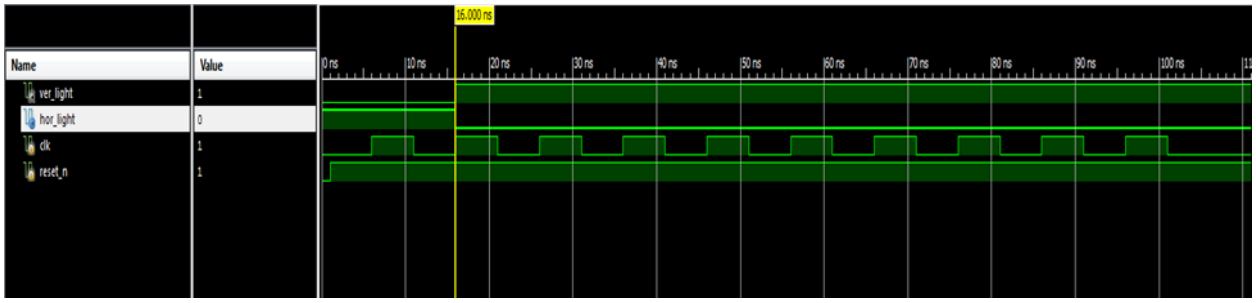

Figure 8(c): Output waveform of partitioned Traffic Signal Controller

Without using any FSM Low Power Technique, Fig. 8(b) displays the simulated behavioural model of the Traffic Signal Controller circuit with the signals ns_red, ns_yellow, ns_green, ew_red, ew_yellow, ns_green, clk (the clock signal), and a reset signal. The result

waveform after executing RTL code is displayed. Figure 8(c) depicts the traffic signal controller circuit's simulated behavioural model after using FSM partitioning. The variables used for representation are ver_light, hor_light, clk for the clock signal, and reset_n for the reset signal.

Figure 9(b) depicts the traffic signal controller circuit's simulated behavioural model after clock gating technique has been applied to it. It demonstrates that there are three signals—red, yellow, and green, that are activated in accordance with input signals given by the clock signal, or clk, and an enable signal that is added to mask the clock signal.

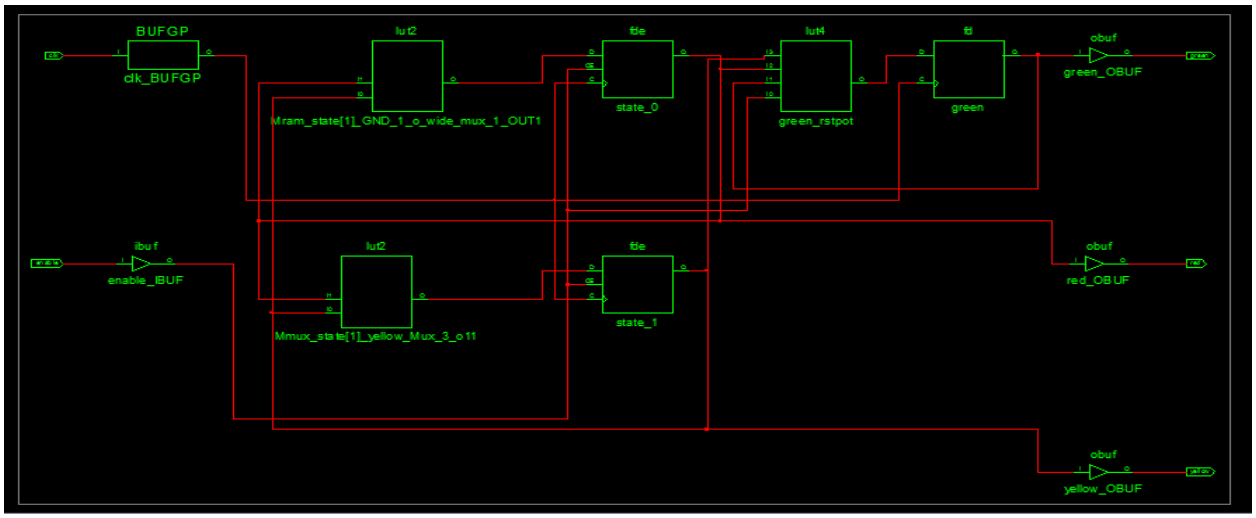

Figure 9(a): Technology Schematic diagram after applying Clock Gating on Traffic Signal Controller

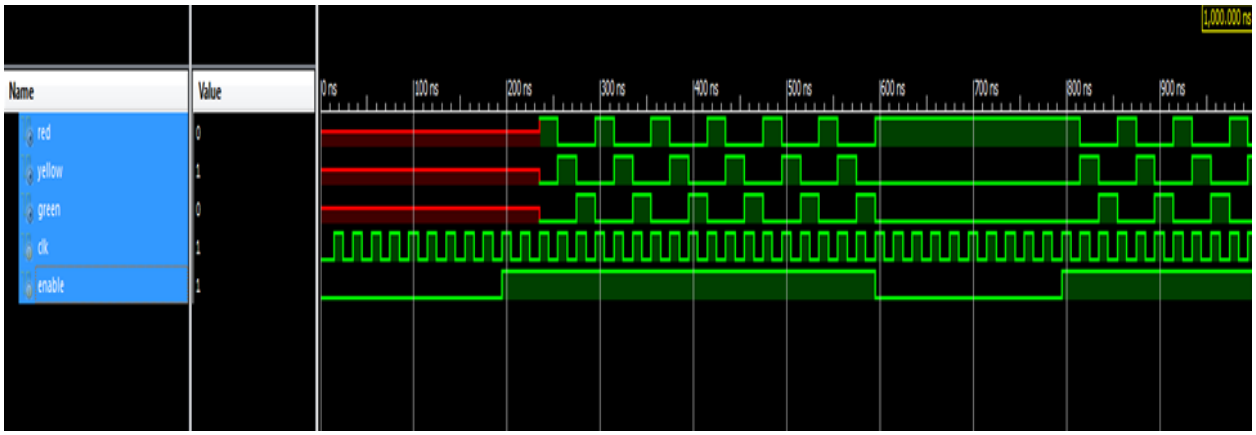

Figure 9(b): Output Waveform of Traffic Controller after clock gating

6.Results and Discussions

The results for sequence detector and traffic signal controller after applying low power techniques are shown in Table 2. Dynamic power of the conventional sequence detector has been calculated as 0.3mW. Dynamic power in decomposed FSMs after partitioning has been calculated as 0.083mW and dynamic power after applying clock gating is 0.2mW. Power saving achieved in sequence detector circuit has been 72% and 33% on applying FSM partitioning and clock gating techniques respectively. Similarly for the conventional traffic signal controller circuit, the dynamic power has been calculated as 8uW. Dynamic Power in FSM after partitioning is calculated as 1uW and after applying clock gating is found to be 2uW. Power

saving of 87.5% and 75% has been achieved on applying FSM partitioning and clock gating techniques respectively on traffic signal controllers. The power saving using different techniques for sequence detector and traffic signal controller has been given in Fig. 10. After comparing both techniques, we have found that FSM Partitioning saves more power in circuits. FSM partitioning is more effective as one single sub-machine is clocked at any particular time and the other one is shut down for that specific time period, which eventually results in more power reduction and less power dissipation of the overall circuit.

Table 2: Comparative Analysis of FSM Partitioning and Clock Gating Techniques for 11011-Sequence Detector and Traffic Signal Controller

FSM Circuit Used	Dynamic Power Value in Original Circuit	Power Reduction Methods used	Dynamic Power Value in Modified Circuit
11011-Sequence Detector	0.3mW	Clock Gating	0.2mW
		FSM Partitioning	0.083mW
Traffic Signal Controller	8uW	Clock Gating	2uW
		FSM Partitioning	1uW

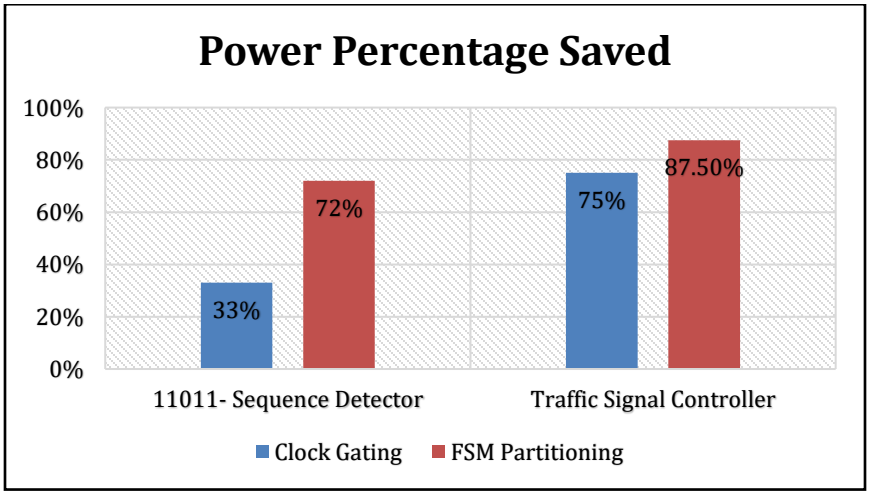

Figure 10: Power Percentage saved through FSM Partitioning and Clock Gating Techniques for 11011-Sequence Detector and Traffic Signal Controller

7.Conclusion

The present work focused on the experimental study of two low-power techniques for FSMs, namely FSM partitioning and clock gating. The example circuits used were a typical 5-bit sequence detector and a traffic signal controller. The focus was on the reduction of dynamic power dissipation, and both techniques yielded good results. FSM partitioning was found to perform better in both circuits compared to clock gating. However, the difference in terms of percentage power saved was more evident in the case of the sequence detector circuit when incorporated with clock gating.

Acknowledgement: Not Applicable.
Funding Statement: The author(s) received no specific funding for this study.
Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

[1] V. Taraate, "FSM designs," *In Digital Design from the VLSI Perspective: Concepts for VLSI Beginners*, pp. 177-191, 2022

[2] M. T. Morazán, "Deterministic Finite-State Machines," *In Programming-Based Formal Languages and Automata Theory: Design, Implement, Validate, and Prove*, pp. 87-131, 2023.

[3] M. T. Morazán, "Nondeterministic Finite-State Machines," *In Programming-Based Formal Languages and Automata Theory: Design, Implement, Validate, and Prove*., pp. 133-165, 2023.

[4] V. Taraate, "FSM Design Techniques," In Digital Design Techniques and Exercises: A Practice Book for Digital Logic Design, pp. 137-151, 2021.

[5] Q. Ningning and X. Baoguo, "Research on FSM for the wireless communication", *In 2010 2nd International Conference on Future Computer and Communication*, vol.2, pp. V2-417, 2010.

[6] M. Ben-Ari and F. Mondada, "Finite state machinesm," *In Elements of Robotics*, pp. 55-61, 2018.

[7] A. Barkalov, L. Titarenko, J. Bieganowski and K. Krzywicki, "Basic Approaches for Reducing Power Consumption in Finite State Machine Circuits—A Review," *Applied Sciences*, vol.14, no.7, pp. 2693, 2024.

[8] P. Bhattacharjee and A. Majumder, "LECTOR based gated clock approach to design low power FSM for serial adder," *In 2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, pp. 250-254, 2016.

[9] G. Ashutosh, M. Jivan, G. Shraddha and P. Rahul, "Reducing Clock Power by Using the Clock Gating Technique," *In Techno-Societal 2016, International Conference on Advanced Technologies for Societal Applications*, pp. 919-926, 2016.

[10] A. Saleem and S. A. Khan. "Low power state machine design on FPGAs," *In 2010 3rd international conference on advanced computer theory and engineering (ICACTE)*, vol.4, pp. V4-442, 2010.

[11] A. Nag, S. Das and S. N. Pradhan, "Low-power FSM synthesis based on automated power and clock gating technique," *Journal of Circuits, Systems and Computers*, vol. 28, no.05, pp. 1920003, 2019.

[12] D. Mihhailov, A. Sudnitson and K. Tarletski, "Web-based tool for FSM encoding targeting low-power FPGA implementation," *In 2010 27th International Conference on Microelectronics Proceedings*, pp. 349-352, 2010.

[13] S. Li and K. Choi, "A high performance low power implementation scheme for FSM," *In 2014 International SoC Design Conference (ISOCC)*, pp. 190-191, 2014.

[14] A. Klimowicz and T. Grzes, "Combined State Merging and Splitting Procedure for Low Power Implementations of Finite State Machines," *In Advances in Systems Engineering: Proceedings of the 28th International Conference on Systems Engineering*, ICSEng 2021, December 14-16, Wrocław, Poland, pp. 190-199, 2012.

[15] P. Choudhury and S. N. Pradhan, "Power modeling of power gated FSM and its low power realization by simultaneous partitioning and state encoding using genetic algorithm," *In Progress in VLSI Design and Test: 16th International Symposium*, VDAT 2012, July 1-4, Shibpur, India, pp. 19-29, 2012.

[16] A. Bukowiec and L. Gomes, "Partitioning of Mealy finite state machines", *IFAC Proceedings Volumes*, vol.42, no.21, pp. 19-24, 2009.

[17] P. Brundavani, "A Principal Component Analysis Algorithm for Seed Enterprise Financial Performance and Scientific and Technological Innovation," *Journal of Computer Allied Intelligence* ,vol. *2*, no.2, pp.49-62, 2024.

[18] Srinivasa Sai Abhijit Challapalli, "Optimizing Dallas-Fort Worth Bus Transportation System Using Any Logic," *Journal of Sensors, IoT & Health Sciences*, vol. 2, no.4, pp.40-55, 2024.

[19] J. E. Ortiz and R. H. Klenke, "Simple traffic light controller: A digital systems design project," *In Proceedings of the IEEE SoutheastCon*, pp. 85-88, 2010.

[20] B. J. LaMeres, "Modeling Finite-State Machines," *In Quick Start Guide to VHDL*, pp. 127-141, 2023.

[21] Srinivasa Sai Abhijit Challapalli, "Sentiment Analysis of the Twitter Dataset for the Prediction of Sentiments," *Journal of Sensors, IoT & Health Sciences*, vol.*2*, no.4, pp1-15, 2024.