Design of a Multi-mode Clock Divider Based on Dual Finite State Machine

Tianhao Luo[1]

10235304406@stu.ecnu.edu.cn[1]

School of Information and Electronic Engineering, East China Normal University, Shanghai 200241, China [1]

Abstract. Modern System-on-Chip (SoC) devices drive various modules that require clock signals with different frequencies and duty cycles. Still, existing digital clock dividers often lack flexibility in complex mode switching and precise duty cycle control. To meet the requirement, this study proposes a multi-mode clock divider based on a three-stage dual-channel Finite State Machine (FSM). The design uses 3-bit binary coding for selecting the target frequency division ratio and 2-bit coding for choosing the specific duty cycle adjustments. Moreover, this paper adopts a right-shift algorithm to calculate the thresholds efficiently. In even division cases, the system employs a single rising-edge FSM to save power, while the system uses dual-channel FSMs and logical "OR" operation to achieve half-cycle precision in odd division conditions. Verified by ModelSim simulation with a 100MHz input clock and 8500ns duration, the divider managed to output desired frequencies, with both odd and even divisions showing little duty cycle deviation, despite some malfunctions in specific situations. This reliable design provides a solution for SoC clock management, supporting dynamic frequency scaling and low-power strategies, and offers a practical reference for sequential logic-based timing control.

Keywords: Clock Divider; Multi-mode; Duty-Frequency Adjustment; Verilog.

1 Introduction

With the rapid development of modern technology, System-on-Chip (SoC) features various functional modules, such as microprocessors, memory, and controllers, within a single silicon die [1]. Various working situation often requires different clock frequencies on a single chip. Therefore, the Clock Management Unit (CMU) has become the core of the SoCs [2]. For example, the accuracy of the signal output relies heavily on the clock frequency. Therefore, the clock divider not only keeps the output frequency precise and working for various purposes, but also plays an important role in Dynamic Frequency Scaling (DFS) and low-power design strategies [3].

Among the various ways for implementing clock dividers, the Finite State machine (FSM) plays a vital role due to its logical flexibility and control precision. It has great advantages in achieving

ICMEEA 2026, April 24-26, Singapore, Singapore

DOI 10.4108/eai.24-4-2026.2364880

non-50% duty cycles, odd-integer division, and fractional division. Moreover, it can achieve precise state encoding as well as lower energy consumption with a certain optimized design [4]. Therefore, exploring a universal, multi-mode clock divider architecture based on FSMs is of significant value for improving the flexibility and reliability of SoC clock networks.

Current academic and industrial research on clock division systems has achieved significant accomplishments, primarily focusing on high-performance analog division and programmable digital division.

In the field of high-performance clock generation, a multi-output clock generator combining an analog Phase-Locked Loop (PLL) with an open-loop fractional divider was proposed. It achieved a wide output range from 500kHz to 150MHz via an open-loop fractional divider [5]. As for the digital domain, Suhas designed a Programmable Clock Management Unit (PCMU) based on clock gating and division techniques, emphasizing bus integration and system-level power management [6].

However, despite these achievements, there is a lack of basic research focusing on pure digital logic that enables users to change the division mode and duty ratio whenever they want. Existing digital dividers often lack the flexibility of FSMs in handling complex state jumps and mode switching. This research aims to bridge this gap by proposing a multi-mode clock divider design based on FSMs, utilizing fine-grained state control to achieve a low-latency, high-reliability universal architecture.

This paper proposes a design framework of an FSM-based clock division architecture and demonstrates its performance through a complete case study. The rest of this paper is organized as follows: Section 2 introduces the theoretical basis of the clock divider and FSM design principles. Section 3 details the FSM design logic for odd and even division. Section 4 presents the implementation of the universal configurable divider. Section 5 analyzes the simulation waveforms and performance parameters. Section 6 concludes the paper.

2 Theoretical Background: Clock Divider and Finite State Machine

2.1 Principle of Clock Divider

The clock divider is a fundamental unit in digital circuits, essential for frequency synthesis and timing control. Its main purpose is to convert a high-frequency input clock signal into a lower-frequency output signal for dealing with different circuit functions. According to the research, the most universal and effective method for implementing clock division requires digital counters as well as logic comparison circuits [7].

In a counter-based division architecture, the division factor N determines the proportional relationship between the output signal frequency and the input signal frequency. In some complex cases, the factor N needs to be proceeded before designing the circuit [8]. The output frequency can be defined as:

$$f_{out} = \frac{f_{in}}{N}. \tag{1}$$

2.2 Concept of FSM

FSM is a mathematical model used for designing sequential logic circuits, describing system transitions between a finite number of states. In digital system design, FSMs are widely applied to implement control units. FSMs are primarily categorized into two types based on their output logic dependencies: Mealy machines and Moore machines [9].

The distinguishing characteristic of a Moore machine is that its output signals only depend on the current state, with no direct relationship with the current input signals, of which the mathematical expression can be defined as:

$$Y = F(S) . \tag{2}$$

where S is the current state of the FSM.

On the other hand, the output signals depend on both the current state and the current input signals in a Mealy FSM. Its mathematical expression can be defined as:

$$Y = F(S, X) . \tag{3}$$

where S stands for the current state of the FSM and X stands for the input signals.

3 Proposed FSM and Clock Divider Design Methodology

3.1 Design of Configurable Division Ratio

To achieve adjustable clock division, the design adopts a control mechanism based on 3-bit binary coding, which allows users to dynamically adjust the current clock division, as shown in Table 1

Table 1. Definition of Division Ratio Coding.

Inputs	Division Mode	Description
000	STOP	output shutdown
001	BYPASS	output the original clock signal
010	2 DIV	Frequency divided by 2
011	3 DIV	Frequency divided by 3
100	4 DIV	Frequency divided by 4
101	5 DIV	Frequency divided by 5
110	6 DIV	Frequency divided by 6
111	7 DIV	Frequency divided by 7

3.2 Design of Configurable Duty Cycle

In addition to frequency adjustability, the system integrates a 2-bit coded duty cycle control method, enabling users to directly input certain binary codes to adjust the clock output duty cycle. The detailed coding scheme is shown in Table 2

Table 2. Definition of Duty Cycle Coding.

Inputs	Duty Cycle	Description
00	\	Invalid Input
01	25%	25% Duty Cycle
10	50%	50% Duty Cycle
11	75%	75% Duty Cycle

Based on these two input bits and the current division factor "N", the system can dynamically calculate the target threshold for the clock output high level.

3.3 FSM Architecture Design

Design of Three-Stage Dual-Channel FSM Architecture. To solve the challenge of achieving a 50% duty cycle in odd-integer division, this design proposes a three-stage dual-channel FSM Architecture. The first stage of the FSM manages state register reset and update action, ensuring synchronous state transitions. The second stage computes the next state based on the current state and inputs, ensuring clear logic to debug. Finally, the third stages implements independent output decoding logic, ensuring the clock signal integrity.

This architecture introduces a "dual-channel" processing mechanism. One of the FSM triggers state transitions on the positive edge of the system clock, while the other triggers state transitions on the negative edge of the system clock.

Both channels share the same state transition logic but are phase-shifted by 0.5T in the time domain. By combining the outputs of these two channels, a phase difference of half a clock cycle can be precisely synthesized. This enables high-precision duty cycle control for odd division without the need for frequency doubling.

Implementation of 1/4 Cycle Algorithm. To efficiently implement 25%, 50%, and 75% duty cycles in hardware, the system must quickly calculate the 1/4 value of the division factor N. This design greatly reduces the complexity of the circuit. In binary arithmetic, shifting a value to the right by "k" bits is equivalent to dividing by 2^k[10]. Therefore, calculating 1/4 situation only requires shifting the binary representation of N to the right by 2 bits, that is, "N >> 2" in Verilog description. This method requires no logic gate resources, relying solely on wiring, which significantly saves hardware area and reduces latency.

Even Division State Assignment and Detailed Design. In even division mode, the output waveform can only depend on the rising edge of the system clock. Therefore, to reduce dynamic power consumption, the design enables only the rising edge channel. Based on the right-shift algorithm, the high-level duration cycle count for even division is set as follows. In 25% duty cycle case, the threshold equals (N >> 2) – 1. In 50% duty cycle case, the threshold equals (N >> 1) – 1. In 75% duty cycle case, the threshold equals ((N × 3) >> 2) – 1.

For instance, assuming that the division factor N equals 4 with the target duty cycle of 75%, the threshold will be ((4 × 3) >>2) – 1 = 2 (010 in binary format). The corresponding state transition table is shown in Table 3.

Table 3. Example State of 4 DIV, 75% Duty Cycle.

Current State			Next State			Output
Q3	Q2	Q1	Q3*	Q2*	Q1*	Y
0	0	0	0	0	1	1
0	0	1	0	1	0	1
0	1	0	0	1	1	1
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0

Odd Division State Assignment and Detailed Design. The challenge in odd division lies in generating a pulse width of half a cycle. This design achieves this by simultaneously activating both the rising and falling edge channels and applying a logical "OR" operation to their outputs. To approximate the set duty cycle in odd division, the counting thresholds for the two channels are separately adjusted. For example, in 25% duty cycle case, the threshold of the positive edge channel equals 0, while the threshold of the negative edge channel is $(N >> 2) - 1$. In 50% duty cycle case, both the positive and negative edge channels are set to the same threshold. Due to the 0.5T phase difference, the overlap of identical pulse widths automatically extends the total width by 0.5T, perfectly realizing a pulse width of $(N-1)/2 + 0.5 = N/2$. In 75% duty cycle case, the positive edge channel threshold is set to last (N-1) T, while the negative edge channel threshold is set to $((N \times 3) >> 2) - 1$.

Similarly, considering the 5 DIV, 25% duty cycle case, the rising edge channel high voltage level lasts 1T; thus, the positive edge threshold is 0. Meanwhile, the negative edge threshold is 0. Detailed states of positive-edge channel FSM and negative-edge channel FSM are listed in Table 4 and Table 5.

Table 4. Example state of a positive-edge channel FSM.

Current State (positive-edge FSM)			Next State (positive-edge FSM)			Output
Q3	Q2	Q1	Q3*	Q2*	Q1*	Y
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0

Table 5. Example state of a negative-edge channel FSM

Current State (negative-edge FSM)			Next State (negative-edge FSM)			Output
Q3	Q2	Q1	Q3*	Q2*	Q1*	Y
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	0
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0

Though two state transition plot appears to be the same, positive-edge FSM domains the interval from 0T to 1T, while the negative-edge FSM domains the clock period from 0.5T to 1.5T. After the logical "OR" operation, the final output high level covers the time interval from 0.5T to 1.5T, resulting in a pulse width of 1.5T. So, the duty cycle created by the system is 1.5T/5T=30%. Given the physical limitation that digital logic cannot generate pulse widths other than multiples of 0.5T, this is the design closest to the theoretical 25%.

4 Case Study: Design and Implementation

The implementation of the system can be divided into five core functional modules: top-level control, frequency division branching, duty adjustment cycle, FSMs sub module, and output generation logic

4.1 Top-Level Control

This is the entry point of the entire system, responsible for initializing all internal registers and input/output signals. After the initialization, the system first checks the reset signal: if the active-low reset signal is activated, the system will clear all FSM state registers and set the output clock to logic low, and keep detecting the reset signal until it is released. When the reset signal is released, the system decodes the input signals "sel_div" and "sel_duty": if "sel_div" is set to "STOP" state or "sel_duty" is set to invalid duty input, the system disables the working state, sets both FSMs to "STOP" state and sets the output to logic low; if the enable condition is met, the system steps into the frequency division mode branching module. The entire workflow, which operates synchronously to the input clock, is shown in Figure 1.

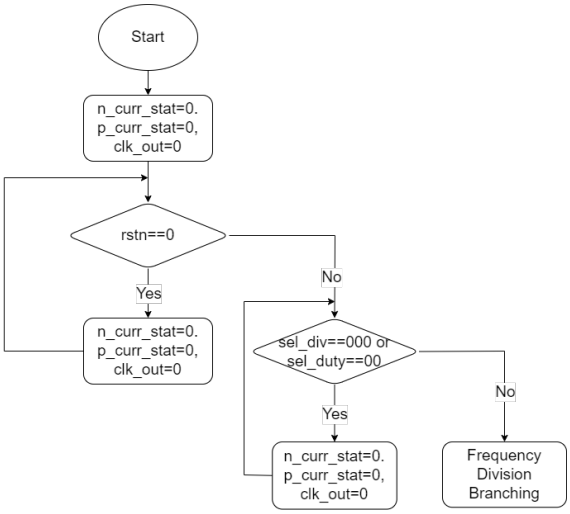

Fig. 1. Top-level Control Flow (Reset + Enable).

4.2 Frequency Division Branching

On the basis of the encoded signal, this module adjusts the frequency division and the decoding of the division factor. First, the system judges whether it is in bypass mode. In this case, the system directly outputs the original clock signal. If it is not in bypass mode, the system decodes the division factor N according to the "sel_div" and sets the even/odd frequency division flag. Based on this flag, the workflow diverges into two branches: even frequency division and odd frequency division. Each branch proceeds to the corresponding duty cycle adjustment module, as shown in Figure 2.

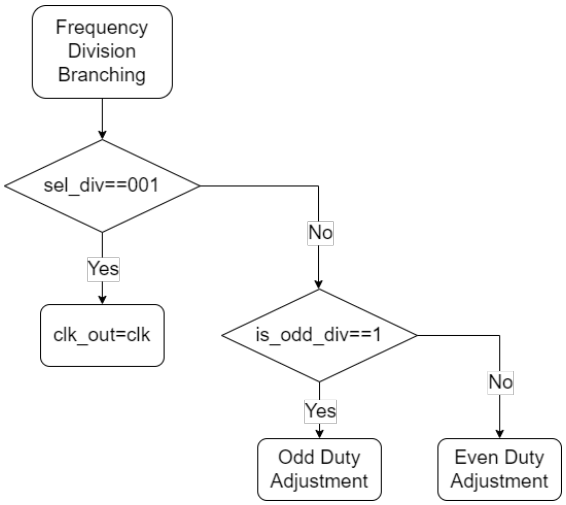

Fig. 2. Frequency Division Mode Branching.

4.3 Duty Adjustment Cycle

This module completes the calculation of the duty cycle threshold given the input signal "sel_duty". For even frequency division, the system only calculates a single threshold of the positive edge FSM, and the threshold value is determined by the selected division factor. For odd frequency division, the system calculates two independent thresholds to optimize the duty cycle accuracy. Different threshold allocation strategies are adopted for different target duty cycles. After the threshold calculation is completed, the system checks the boundary of the threshold value to ensure that the threshold is within a valid range, and then outputs the threshold to the FSM sub module. The detailed process is shown in Figure 3.

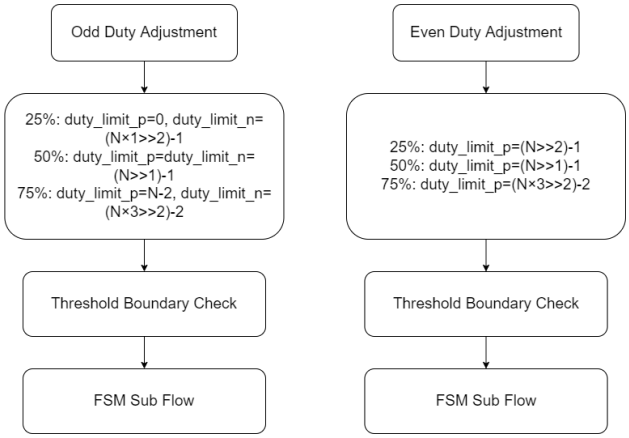

Fig. 3. Duty Adjustment Logic.

4.4 FSM Sub Module

This module is the core unit of the system, including two FSMs: positive edge FSM and negative edge FSM. For the positive edge FSM, its process can be divided into three stages. Firstly, the state register is updated to the next state. Next, if the current state value in the FSM register reaches the value of the division factor N, the next state is cleared to "0". Otherwise, the current state value is increased by 1. Finally, the output logic is generated by comparing to the duty cycle threshold, and the output signal of the positive edge FSM is determined. The negative-edge FSM has the same three-stage execution logic as the positive-edge FSM. The only difference is that this register is updated at the falling edge of the input clock. For even frequency division, the system only needs the positive edge FSM. As for the odd frequency division, the system calls for both FSMs to achieve half-cycle precision counting, as shown in Figure 4 and Figure 5.

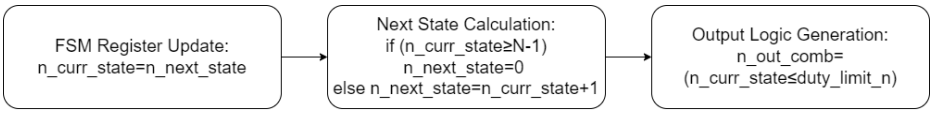

Fig. 4. Negative Edge FSM Sub Flow.

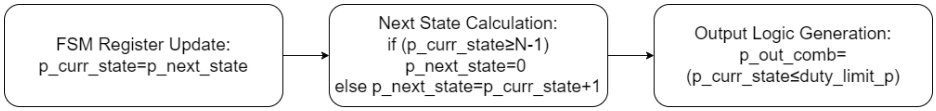

Fig. 5. Positive Edge FSM Sub Flow.

4.5 Output Generation Logic

This module generates the final output clock signal according to the even/odd frequency division type and the output of the FSM sub flow. Before generating the output, the system first checks the reset and enable signals again: if the reset signal is activated or the system is disabled, the output clock is forced to be low level. For even frequency division, the final output clock is directly the output signal of the positive edge FSM; for odd frequency division, the system performs an "OR" operation on the output signals of the positive and negative edge FSMs to realize half-cycle precision output, and then generates the final output clock, as shown in Figure 6.

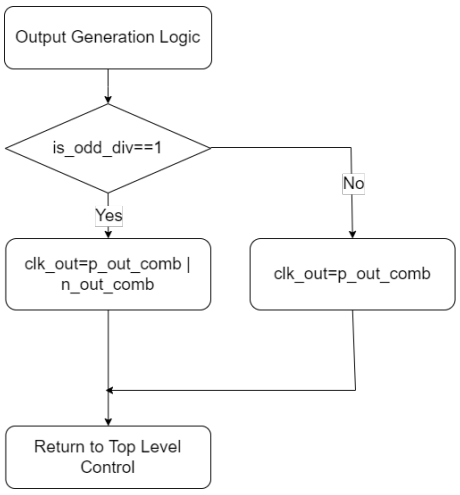

Fig. 6. Output Generation Logic.

5 Results and Discussion

5.1 Simulation Environment Description

The functional verification of this system is implemented based on ModelSim simulation software. It provides a high-precision and visual platform for validating the logic correctness of the designed circuit. Verification settings are as follows. Firstly, the testbench defines the time scale as 1ns / 1ps, where 1ns is the basic time unit for simulation, and 1ps is the simulation precision. Besides, the input clock signal is generated by flipping the clock level every 5ns in the testbench, resulting in a clock period of 10ns. That equals a 100MHz clock signal. Moreover, the total simulation time is set to 8500ns, which is long enough to cover the complete functional test of all working scenarios and ensures that each mode runs for multiple output clock periods to collect stable waveform data

5.2 Simulation Environment Description

Frequency Division Verification. The output frequency and period for each division mode are measured and summarized in Table 6. Representative waveforms are shown in Figure 7:

Table 6. Results of Output Division.

Div Mode	Output Period (ns)	Output Frequency (MHz)
000	0	0.0
001	10	100.0
010	20	50.0
011	30	33.3
100	40	25.0
101	50	20.0
110	60	16.7
111	70	14.3

Notes: Waves from top to bottom are: input clock signal, “rstn” signal, “sel_div” signal, “sel_duty” signal, output clock signal, and internal counter register.

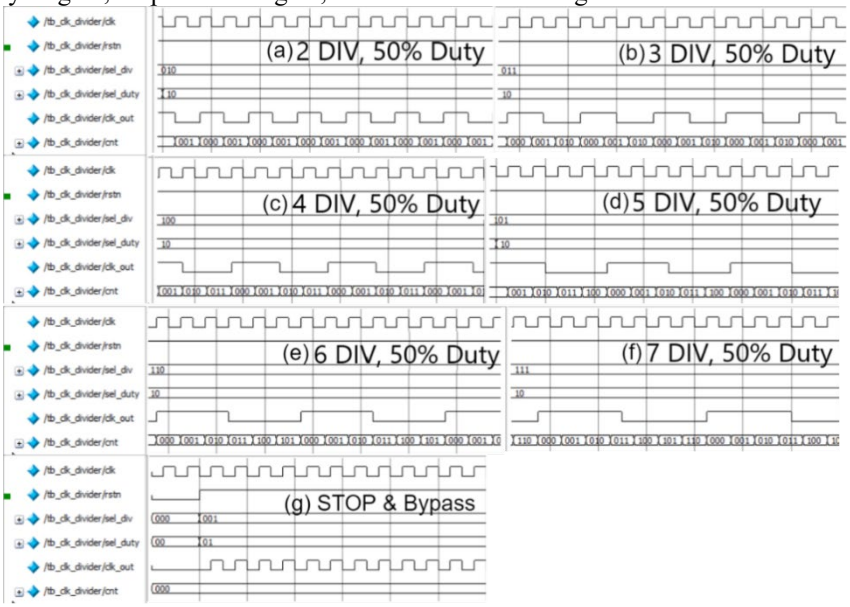

Fig. 7. Division Measurement Collection.

Duty Cycle Verification. The output duty and logic high period for each division mode are measured and summarized in Table 7. Representative waveforms are shown in Figure 8:

Table 7. Results of Output Duty Cycles.

Division Mode	Selected Duty Cycle	Logic High Period	Measured Duty Period	Deviation

3 DIV	75%	2	66.7%	11.1%
4 DIV	25%	1	25.0%	0.0%
	75%	3	75.0%	0.0%
5 DIV	25%	1.5	30.0%	20.0%
	75%	4	80.0%	6.7%
6 DIV	25%	1	16.7%	33.2%
	75%	4	66.7%	11.1%
7 DIV	25%	1.5	21.4%	14.4%
	75%	6	85.7%	14.3%

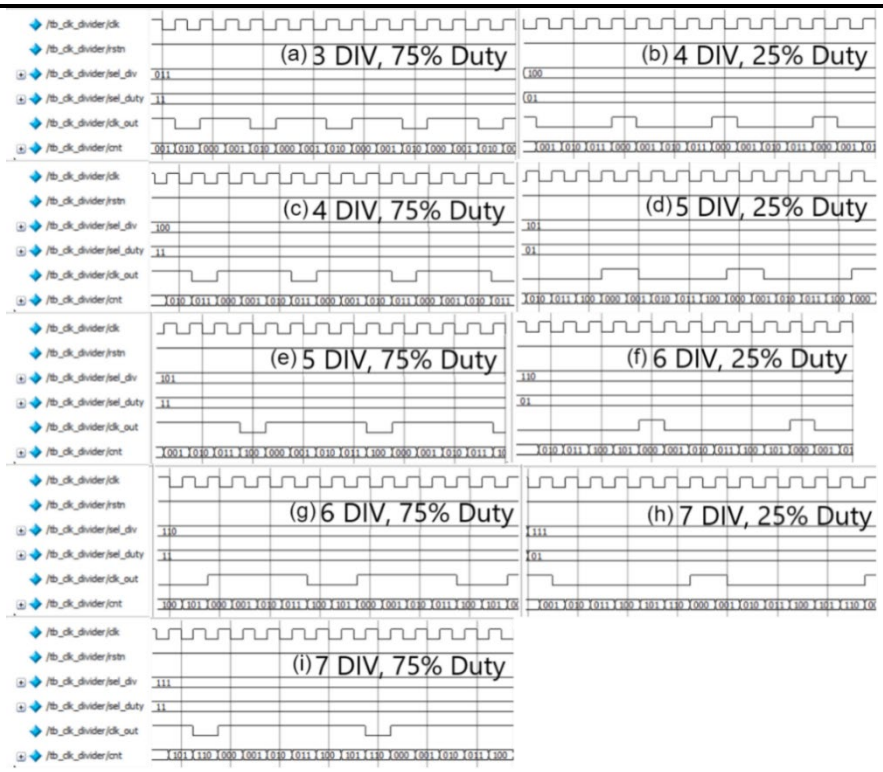

Fig. 8. Duty Measurement Collection.

Deviation Analysis. When dealing with 2 DIV 25%, 75% and 3 DIV 25% duty conditions, there is a slight malfunction in the output clock signal as shown in Figure 9. Therefore, further optimization is required.

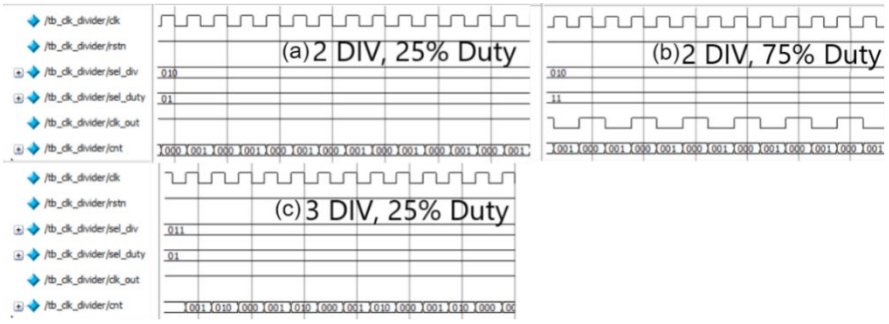

Fig. 9. Deviation Collection.

6 Conclusion

This paper shows a multi-mode clock divider design based on a three-stage dual-channel FSM. The design bridges the gap in the flexible clock divider for SoC applications. Through the design and simulation verification, the main conclusion is as follows: the proposed architecture realizes configurable frequency division and duty cycle adjustment controlled by external control signals. Furthermore, the even division employs a single rising-edge FSM to optimize power consumption, while the odd division utilizes dual-channel FSMs to achieve better performance. The simulations confirm that the output frequency meets the theoretical division ratio, verifying the correctness and reliability of the design.

This research fills the gap in pure digital FSM-based clock division systems, addressing the limitations of traditional clock dividers in handling complex mode switching and non-50% duty cycles. The proposed universal architecture provides a solution for clock management units, facilitating dynamic frequency scaling and low-power design in modern SoCs, and offers a reference framework for other researchers exploring sequential logic-based timing control.

Despite the functional performance, the design exhibits deviations in specific odd-division modes. Future research can focus on optimizing the threshold calculation algorithm to enhance the accuracy in small division factor cases, extending the division range to support higher integer or fractional divisions, and integrating advanced low-power techniques to further reduce power consumption, enabling broader application in high-frequency and energy-constrained electronic systems.

References

[1] Lin, T.L., Chen, S.J.: A Platform of Resynthesizing a Clock Architecture Into Power-and-Area Effective Clock Trees. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 39, No. 10, pp. 2475-2488 (2020).

[2] Lee, J., Han, J., Lo, C.L., Lee, J., Kim, W., Kim, S., Kang, B., Han, J., Jung, S., Nomiyama, T., Lee, J., Cho, T.B., Kang, I.: NB-IoT and GNSS All-In-One System-On-Chip Integrating RF Transceiver, 23-dBm CMOS Power Amplifier, Power Management Unit, and Clock Management System for Low Cost Solution. IEEE Journal of Solid-State Circuits, Vol. 55, No. 12, pp. 3400-3413 (2020).

[3] Tibenszky, Z., Carta, C., Ellinger, F.: A 0.35 mW 70 GHz Divide-by-4 TSPC Frequency Divider on 22 nm FD-SOI CMOS Technology. 2020 IEEE Radio Frequency Integrated Circuits Symposium (RFIC), pp. 243-246 (2020)

[4] Sharma, C., Chauhan, D.K.: High performance low power AHB DMA controller with FSM decomposition technique. 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp. 456-461 (2017)

[5] Jin, J., Jin, Y., Gan, Y.: A 500 kHz to 150 MHz Multi-Output Clock Generator Using Analog PLL and Open-Loop Fractional Divider with 0.13 μm CMOS. Electronics, Vol. 11, No. 15, p. 2347 (2022).

[6] C.D., J., Suhas, P.H., L., S., R., Ramya, J.V.: Synthesis of a Programmable Clock Management Unit Using Clock Dividers and Clock Gating using 45nm technology. 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), pp. 1-5 (2024)

[7] Bu, N., Koppad, D.: High Bandwidth Multi-Modulus Clock Divider. 2021 IEEE Mysore Sub Section International Conference (MysuruCon), pp. 114-119 (2021)

[8] Wu, P.H., Cheng, K.L., Shen, T.H., Ma, C.C., Kuo, Y.F.: Design of 2.1-4.8GHz Fractional-N Frequency Divider based on Digital Delta-Sigma Modulator and Clock Distribution Technique. 2023 Asia-Pacific Microwave Conference (APMC), pp. 174-176 (2023)

[9] Ravikumar, P., Rao, M.V.G., Nikitha, V.: Verilog Based Automated Retail System. 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0, pp. 1-5 (2024)

[10] Cui, J., Wang, S., Zhang, N., Ma, S.: An optimized probability estimation model for binary arithmetic coding. 2015 Visual Communications and Image Processing (VCIP), pp. 1-4 (2015)