

A Low-delay Configurable Register for FPGA using GDI Technique

T Adithya Nag Venkat¹, Nandyala Venkata Sai Samartha Aditya², Dr. Aarthy M^{*3}

Submitted: 29/01/2024 **Revised:** 07/03/2024 **Accepted:** 15/03/2024

Abstract: The fundamental building block of FPGA (Field Programmable Gate Array) is the CLB (Configurable Logic Block). The sequential component of a CLB is a Configurable register. The goal of this paper is to show how to reduce the area of a low delay configurable register without significantly affecting its latency. To achieve this, we employed the GDI (Gate Diffusion Input) approach, which is well known for drastically reducing the transistor count. The GDI logic operates slightly different than a 2x1 mux. The primary issue with this technique is that it produces weak signals (weak 0s and weak 1s), while reducing the number of transistors. In this paper, we cleverly addressed the issue.

Keywords: FPGA, CLB, Configurable register, latency, GDI

1. Introduction

In an FPGA, the configurable logic block serves as the fundamental core circuit. Cornerstones of the intricate FPGA implementation relies on both its combinational and sequential logic [2] [3] [4]. Configurable registers are essential to the execution of the sequential logic function. To ensure programmability, it is necessary to configure the customisable registers as latches or registers based on the demands and must have characteristics such as global initialization, synchronous as well as asynchronous reset. The addition of capture and write back functions is also important to achieve quick calculation.

1.1. Configurable Register

For better routing, performance, and power, CLB enables accessing the register inputs independently. A consequence of this trend leads to a greater impact on the adjustable registers timing parameters. Previously, to realize various operations, the adjustable registers worked in conjunction with other configurable logic cells using D flip-flops as master-slave configuration [2]. However, excessive control circuitry increased the total circuit's delay, took up more space, with reduction in usage of combinational logic [7]. The Fig. 1 depicts the configurable register implementations in which the input D and reset signal are combined using a combinational circuit in this approach to create the desired signal [3]. Due to the series

connection between the register input and the control signal, the master circuit's state is changed by the D input more slowly, increasing setup time. To mitigate this negative effect [3], the relevant combinational logic can be implemented using a transistor-level unbalanced circuit, as shown in Fig. 2. However, this approach makes the D input load heavier and cannot significantly shorten setup time.

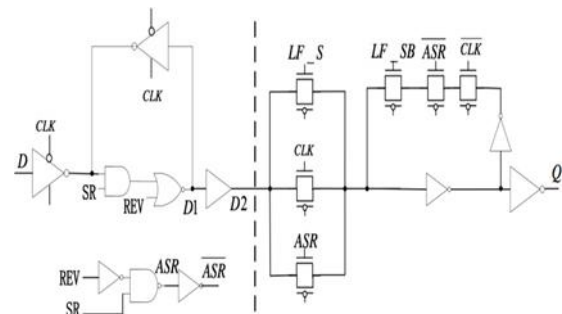


Fig. 1. Slow setup time Configurable register [3]

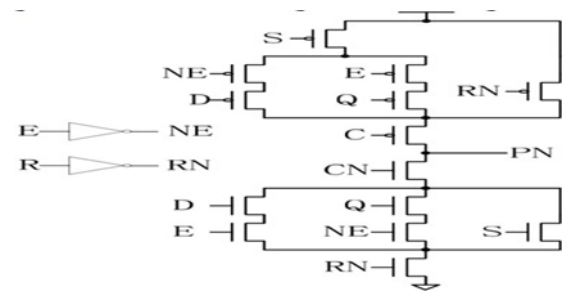


Fig. 2. Improved transistor-level circuit [3]

1.2. Low Delay Configurable Register

The master-slave D flip-flop is still the design's foundation due to its overall stability and adaptability. The desired signal is subsequently added to carry out the responsibilities of latches, registers, global initialization, both asynchronous and synchronous resets, capture, and write-back. The transmission gate on the key nodes is

¹ School of Electronics Engineering, Vellore Institute of Technology, Vellore, INDIA

Email: tadithya2001@gmail.com
ORCID ID : 0009-0004-6061-8440

² School of Electronics Engineering, Vellore Institute of Technology, Vellore, INDIA

Email: adityanandyala@gmail.com
ORCID ID : 0009-0000-2619-6572

³ Assistant Professor Senior, School of Electronics Engineering, Vellore Institute of Technology, Vellore, INDIA

ORCID ID : 0000-0002-0617-260X

* Corresponding Author Email: aarthy.m@vit.ac.in

utilised to regulate the register mode. To prevent the control circuit from impacting the register timing settings, the control and the D input signal are isolated. The primary circuit design of the programmable register is shown in Fig3, and it can be accessed via three ports.

The diagram shows that the primary circuit can be entered through ports (a, b, and c). A conventional D input signal is input through port a. Synchronous and asynchronous reset signals are input through ports b and c, respectively. Global initialization and write back signals that require asynchronous computation are also available through port c.

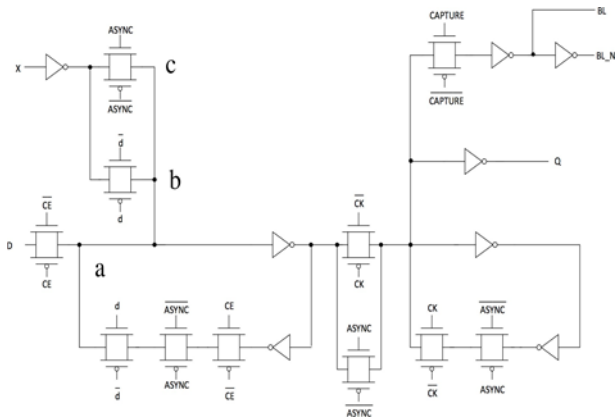
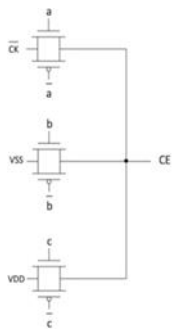


Fig. 3. Primary Circuit design of Low Delay Configurable Register [1]



$$a = \overline{LATCH} \cdot \overline{ASYNC} \cdot \overline{SYNC}$$

$$b = LATCH$$

$$c = \overline{LATCH} \cdot (ASYNC + SYNC)$$

Fig. 4. Control Circuit of Low Delay Configurable Register [1]

The following control circuit, which has three input ports, can choose one of them to setup the circuit by using the key nodes of transmission gates to switch it into synchronously / asynchronously overwritten, register mode, or latch mode, and then execute various operations with the control signals. The following four approaches are listed below:

- **Latch Mode:** This mode chooses Port A among the three ports. At this stage, when the CE, SYNC, and ASYNC signals are all at logic0, the circuit is modified to resemble Fig. 5, that achieves a transparent latch.

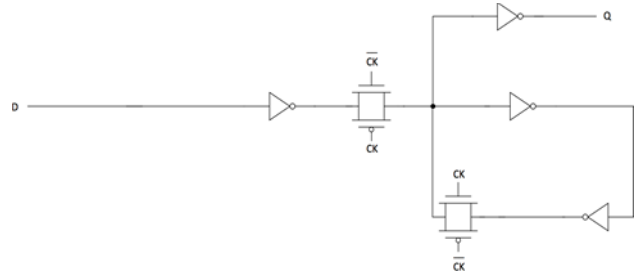


Fig. 5. Latch Mode of Low Delay Configurable Register [1]

Register Mode: The Register Mode selects Port A from the three available ports. The customizable register circuit is now simplified to look like Fig. 6, which depicts a fundamental clock falling edge trigger register. The CE is set with the inverted clock signal, the SYNC and ASYNC signals are all at logic0.

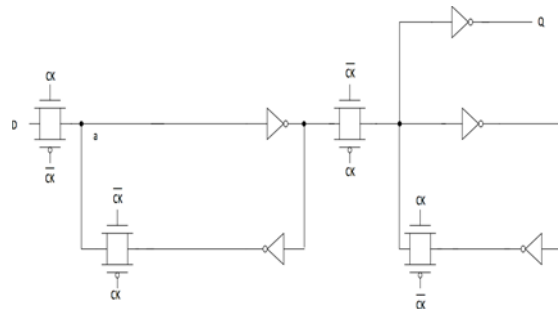


Fig. 6. Register Mode of Low Delay Configurable Register [1]

- **Synchronous Overwrite Mode:** In this mode, the circuit is streamlined by setting CE and SYNC to logic1, and selecting port b from the available three ports, as shown in Fig. 7. The only difference between this circuit and the master-slave D flip-flop is that the input port is switched from D to X.

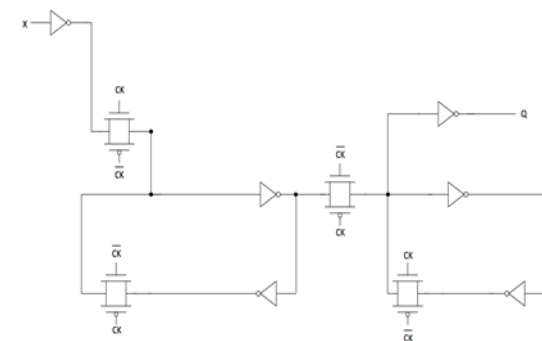


Fig. 7. Synchronous Overwrite Mode of Low Delay Configurable Register [1]

- **Asynchronous Overwrite Mode:** The asynchronous overwrite mode simplifies the circuit by selecting Port C from the three ports, setting CE and ASYNC to logic1, as illustrated in Fig. 8. The flip-flop's master and slave circuits are now disconnected, and three inverters capable

of asynchronously overwriting the output connect the input X to the output port Q. The master and slave circuits are disabled to avoid their stored values from interfering with the asynchronous overwrite input X signal and leading to a logic error.

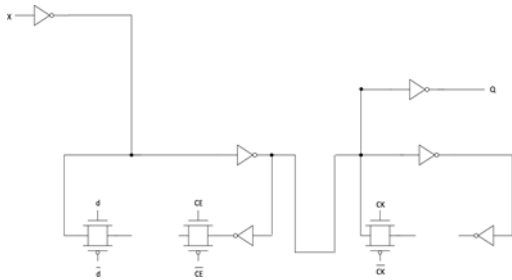


Fig. 8. Asynchronous Overwrite Mode of Low Delay Configurable Register [1]

1.3. GDI Technique

The GDI, or Gate Diffusion Input, approach provides reduction in digital circuit size, power consumption, and propagation latency while retaining a minimal level of complexity in the logic architecture. The GDI approach depends on a simple cell, as seen in Fig. 9.

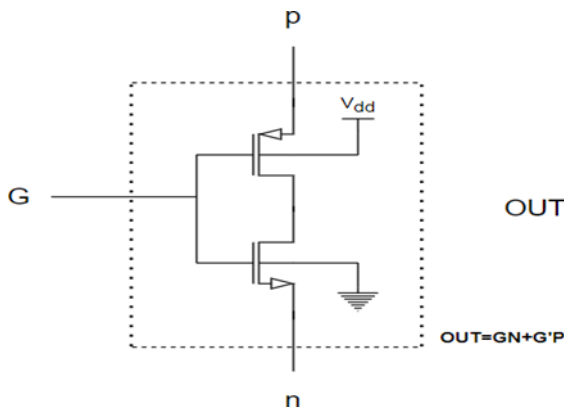


Fig. 9. GDI basic cell [9]

The GDI basic cell initially resembles a typical CMOS inverter, but there are some notable differences:

- 1) There are three inputs to the GDI cell: G represents the common gate input for both nMOS and pMOS, P represents the source/drain input for pMOS, and N represents the source/drain input for nMOS.
- 2) In contrast to a CMOS inverter, it can be freely biased because the majority of both nMOS and pMOS are linked to N or P [9]. Each GDI unit functions as a 2x1 mux with G as the selection line. This logic is used throughout this paper.

2. Design Methodology

2.1 GDI implementation

We managed the mode of operation in this paper using a different strategy [1].The mode of operation will be

controlled by three signals: Y (which toggles between latch and register modes), Z (which toggles between asynchronous and synchronous reset), and R (which allows the reset value to function in line with Z).

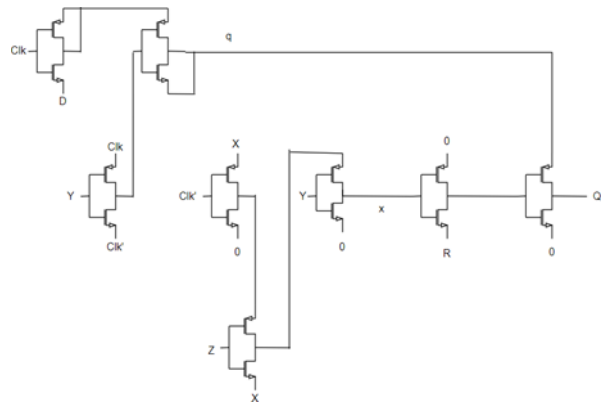


Fig. 10. GDI Circuit

Additionally, we constructed the circuit using the GDI method, which reduced the required number of transistors by half. For our design strategy, no control circuit has been developed. Instead, we have employed the following alternative control scheme shown in Table1.

Table 1.
GDI Circuit Modes

Y	Z	R	Mode
1	X	X	Latch
0	X	0	Register
0	0	1	Synchronous Reset
0	1	1	Asynchronous Reset

Clk

Fig. 11. GDI Circuit Latch Mode

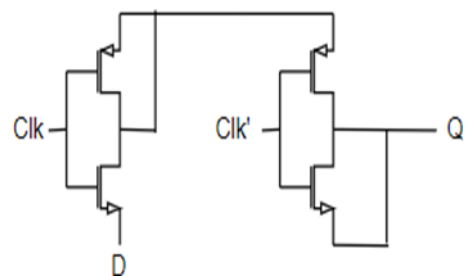


Fig. 12. GDI Circuit Register Mode

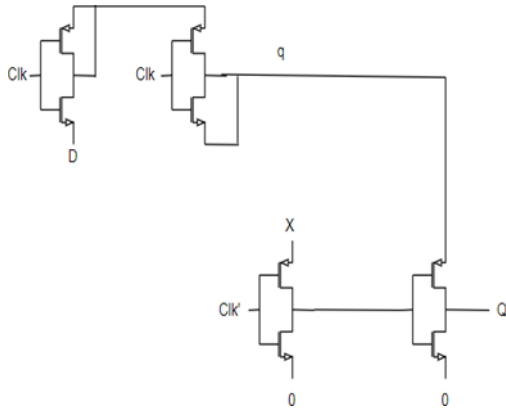


Fig. 13. GDI Circuit Synchronous Overwrite Mode

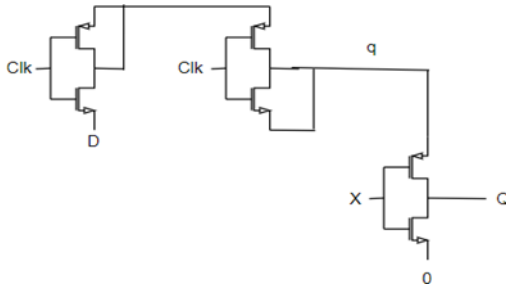


Fig. 14. GDI Circuit Asynchronous Overwrite Mode

2.1 Improved circuit

The primary problem was that, on sometimes, GDI logic yielded weak logics (weak 0 when P and G are both 0 and weak 1 when N and G are both 1) [8], [9]. In order to overcome these weak signals, the P of one GDI unit is connected to the N of the second GDI unit, and vice versa. The complimentary signals are applied to each G pin individually as shown in the Fig. 15. If we consider the TGL is not necessary (for instance, if N is linked to 0, there is no need to connect a pMOS in parallel to it as it always outputs a strong 0), we remove it. As a result, the circuit's current transistor count ranges from 50% to 100% of the original circuit's transistors.

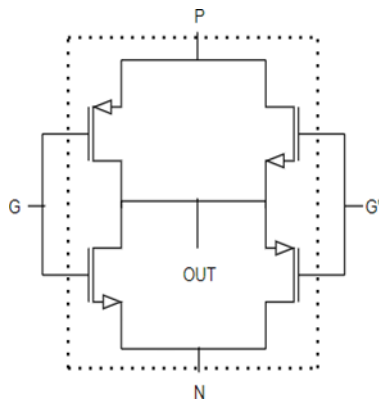


Fig. 15. Improved GDI Block

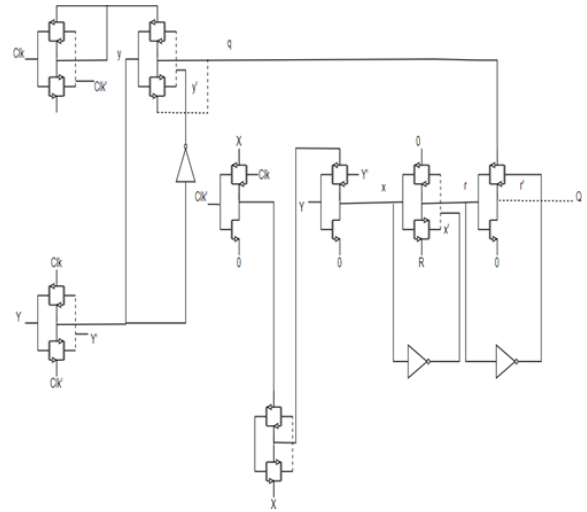


Fig. 16. Improved Circuit

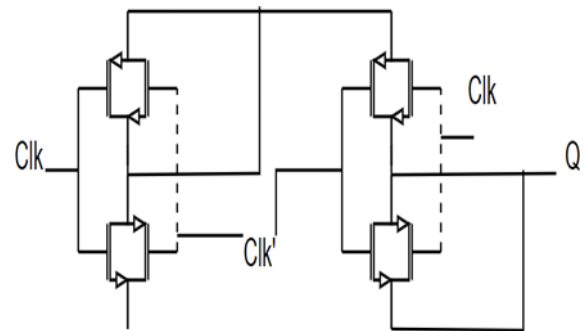


Fig. 17. Improved Circuit Latch Mode

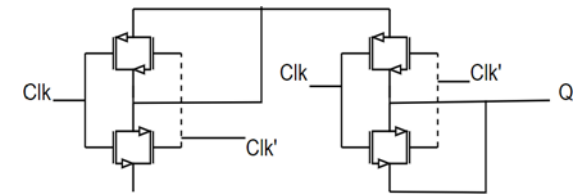


Fig. 18. Improved Circuit Register Mode

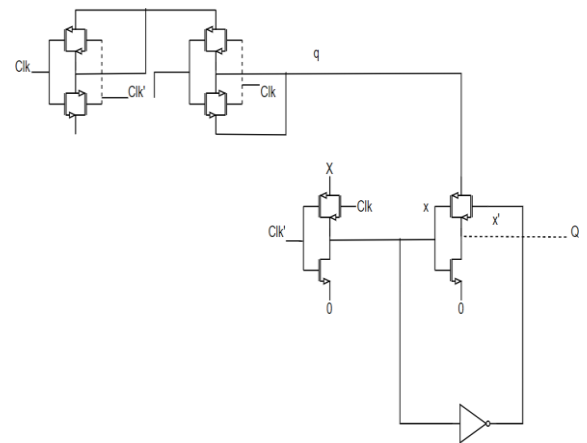


Fig. 19. Improved Circuit Synchronous Overwrite Mode

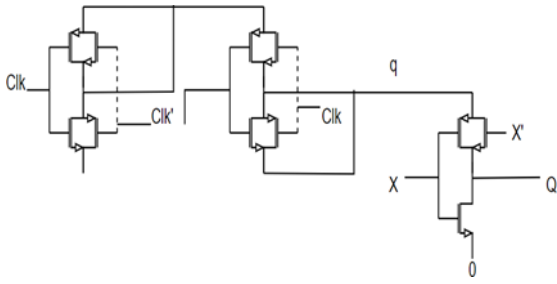


Fig. 20. Improved Circuit Asynchronous Overwrite Mode

3. Results

3.1. Transistor count

Low Delay Configurable Register:

Main Circuit: 19 pMOS 19 nMOS(11 TGL units and 8 inverters)

CE circuit: 3 pMOS 3 nMOS(3 TGL units).

Intermediate States: 25 pMOS and 25 nMOS (3 pMOS and 3 nMOS for the circuit constructions of ‘a’ and ‘c’ respectively to be utilized in the CE circuit and other 19 pMOS and 19 nMOS for the circuit constructions of other miscellaneous circuits such as X, y). Total = 94 transistors

GDI circuit:

8 pMOS 8 nMOS(8 GDI units)

Total = 16 transistors.

Improved circuit:

16 pMOS 19 nMOS (3 inverters, 5×2 GDI units connected as TGL and 3 nMOS connected as TGL to 3 pMOS of the 3 GDI units which are not paired as TGL)

Total = 35 transistors.

3.2 Outputs of Improved Circuit

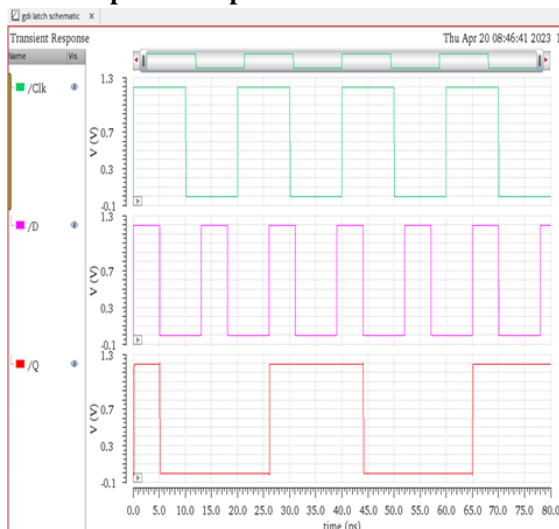


Fig. 21. Latch Mode Output

We can see that we got a high level triggered output for the specified data signal. Thus the circuit functions as a latch.

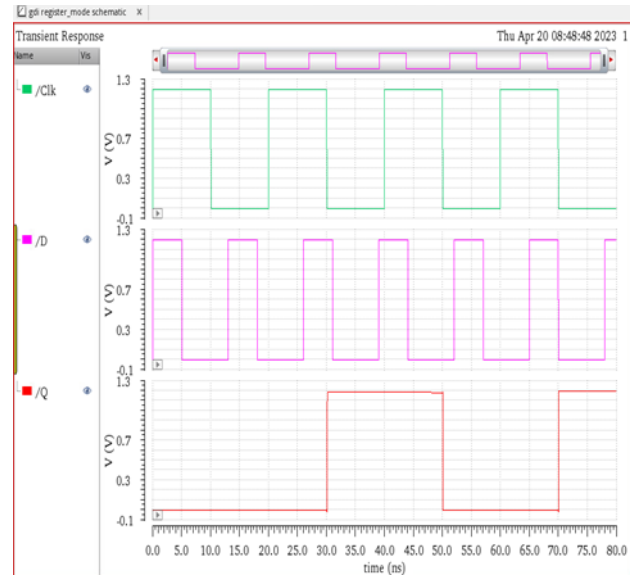


Fig. 22. Register Mode Output

We can observe that we received a negative edge triggered output for the given data signal. Thus the circuit is behaves as a register.

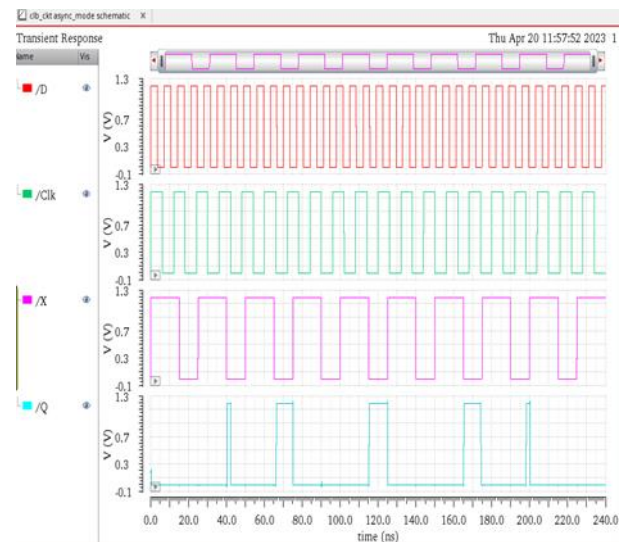


Fig. 23. Synchronous Overwrite Mode Output

We can notice that the output Q processes the data signal synchronously with respect to X and Clock Signal. Therefore the circuit works as a synchronous reset register.

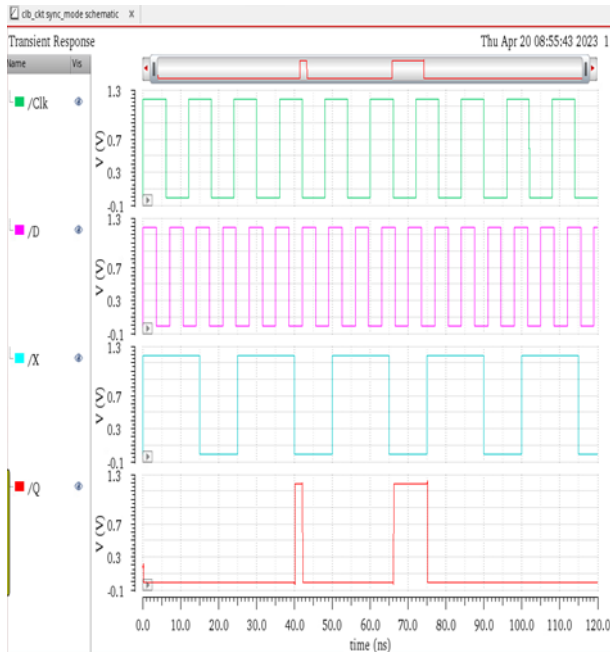


Fig. 24. Asynchronous Overwrite Mode Output

We can find that the output Q takes the data signal asynchronously with respect to X. Therefore the circuit operates as an asynchronous reset register.

3.3 Delay Calculation

Dealy time	[1]	GDI
$T_{Clk} - T_Q$	20.22ps	42.77ps
$T_D - T_Q$	39.56ps	45.77ps

Table 2. Latch mode Delay comparison

Dealy time	[1]	GDI
$T_{Clk} - T_Q$	19.36ps	21.69ps
$T_D - T_Q$	27.12ps	27.07ps

Table 3. Register mode Delay comparison

Dealy time	[1]	GDI
$T_{Clk} - T_Q$	53.64ps	27.297ps
$T_D - T_Q$	54.64ps	26.51ps
$T_X - T_Q$	63.28ps	44.84ps

Table 4. Synchronous overwrite mode Delay comparison

Dealy time	[1]	GDI
$T_{Clk} - T_Q$	53.65ps	58.9ps
$T_D - T_Q$	54.65ps	63.583ps
$T_X - T_Q$	37.56ps	94.33ps

Table 5. Asynchronous overwrite mode Delay comparison

Power Consumption	[1]	GDI
Latch Mode	659.7 nW	285.6 nW
Register Mode	389.3 nW	9.446 uW
Synchronous overwrite Mode	861.6 nW	8.083 uW
Asynchronous overwrite Mode	783.9 nW	5.007 uW

Table 6. Power Consumption comparison

4. Conclusion

The goal of this paper is to demonstrate how to minimize the area of a low delay programmable register without drastically altering its latency by decreasing the amount of transistors. To achieve this, we used the GDI approach, however the main drawback with this technique is that it produces weak signals (weak 0s and weak 1s), while reducing the number of transistors. So in this paper the GDI circuit was implemented using the TGL approach to remove weak signals. This subsequently doubled the transistor count. To address this, redundant transistors were removed wherever possible, leading to a reduction in transistor count compared to the base paper. In addition, the modification in control strategy helped to reduce the number of transistors. The delay increased slightly, ranging from 15ps to 46ps, but it was not significant. However, the power consumption decreased for the latch mode but increased significantly for the other three modes. Ultimately, this approach proved successful in reducing the transistor count with minimal impact on delay.

References

- [1] Lu, Zhi-Yin, Jia-Feng Liu, Yun-Bing Pang, Zheug-Jie Li, Yu-Fan Zhang, Jin-Mei Lai, and Jian Wang." A Low-delay Configurable Register for FPGA." In 2019 IEEE 13th International Conference on ASIC (ASICON), pp. 1-4. IEEE, 2019.
- [2] Rose, Jonathan, Robert J. Francis, David Lewis, and Paul Chow." Architecture of field-programmable gate arrays: The effect of logic block functionality on area efficiency." IEEE Journal of Solid-State Circuits 25, no. 5 (1990): 1217-1225.
- [3] Fu, Yong, Yuan Wang, and Jin-mei Lai." The design of FPGA's high speed configurable logic units." In 2012 IEEE 11th International Conference on Solid-State and Integrated Circuit Technology, pp. 1-3. IEEE, 2012.

- [4] Chandrakar, Shant, Dinesh Gaitonde, and Trevor Bauer." Enhancements in UltraScale CLB architecture." In Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, pp. 108-116. 2015.
- [5] Rani, D. Gracia Nirmala, C. Mathumitha, R. Priyadharshini, and S. Rajaram." Design and implementation of Configurable Logic Block of an FPGA using quantum dot cellular automata." In 2016 3rd International Conference on Devices, Circuits and Systems (ICDCS), pp. 43-48. IEEE, 2016.
- [6] Sudhanya, P., SP Joy Vasantha Rani, and M. C. Lavanya." Design of Logic Blocks for Efficient Architecture of FPGA." In 2019 IEEE 1st International Conference on Energy, Systems and Information Processing (ICESIP), pp. 1-5. IEEE, 2019.
- [7] Hai-zhou, Lu, Lai Jin-mei, and Tong Jia-rong." High effective dynamic reconfigurable registers in FPGA." Journal of Fudan University (Natural Science) 48, no. 4 (2009).
- [8] Kumre, Laxmi, Ajay Somkuwar, and Ganga Agnihotri." Analysis of GDI Technique for digital circuit design." International Journal of Computer Applications 76, no. 16 (2013).
- [9] Morgenshtein, Arkadiy, Alexander Fish, and Israel A. Wagner." Gate-diffusion input (GDI): a power-efficient method for digital combinatorial circuits." IEEE transactions on very large-scale integration (VLSI) systems 10, no. 5 (2002): 566-581.
- [10] Joy, Upal Barua, Avishek Chakraborty, Preyonti Biswas, Arka Das, Swagata Sen, and Afra Tasnim." Two-Bit Magnitude Comparator Design Using Gate Diffusion Input Technique and Static CMOS Logic." In 2023 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), pp. 17-21. IEEE, 2023.