International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.001 ISSN: 2581-4621

A Glitch-free Clock Multiplexer forNon-Continuously Running Clocks

[1] Dr. R. Senthamil Selvan, [2] P. Sasikala, [3] M. Suparna, [4] N. Vamsi Krishna, [5] A. Vinod Kumar
[1]*Associate Professor, Department of ECE, Annamacharya Institute of Technology and Sciences, Tirupati, A.P.*
[2,3,4,5,6]*B. Tech Student, Department of ECE, Annamacharya Institute of Technology and Sciences, Tirupati, A.P.*

Abstract
Modern system-on-chips frequentlyintegrate blocks that, based on the state of the circuit, require the triggering of two or more clock sources. Such systems bring glitch-free clock multiplexers to choose the required clock. Modern solutions have the particular issue that they require running clocks to switch from one source to another. In this project, a new clock multiplexer is introduced that gets around this restriction and makes it possible to switch the clock even if it ceases operating before switching has taken place. The inclusion of the multiplexer in a RADHARD 1.6–2.5Gbps SERDES for space uses has demonstrated the multiplexer's applicability in silicon.
Index Terms— Clock multiplexer, clock domains, System-on-Chip (SoC), glitch-free, stoppable clocks

I. INTRODUCTION

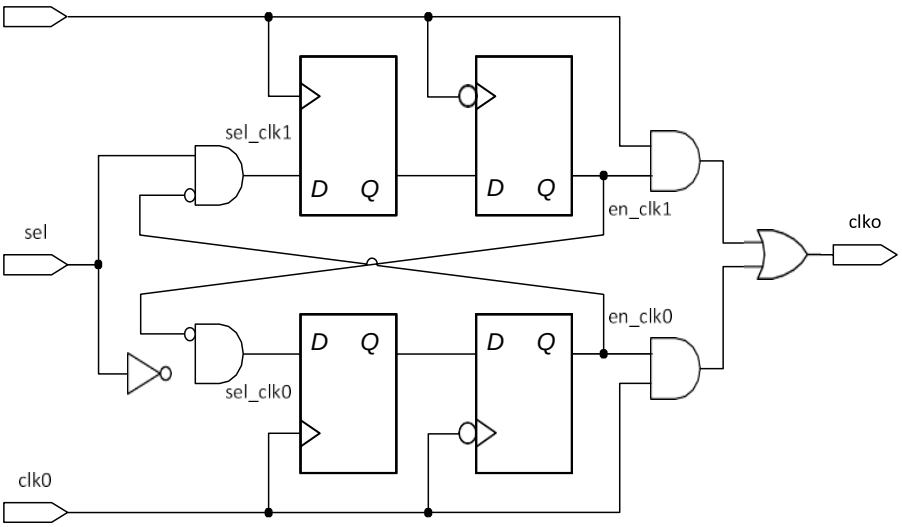
Today's sophisticated integrated circuits and systemon chips (SoC) are made up of a variety of parts. Adifficult job in the design flow is integrating thecomponents into the entire system. Individual components frequently belong to various clock domains and must function at various clock frequencies. Depending on the system status, aspecific block of a circuit may occasionally be activated by multiple clocks. Consider, for illustration, a device with an external interface thatincludes a second clock, such as a serial peripheralinterface (SPI). The system may have a componentthat, when the SPI is operational, must be triggered by the SPI clock; however, when the system isoperating normally, this component must betriggered by the system clock. A clock multiplexer,also known as a clock switch, is needed to executethis scheme in order to choose the clock source. Toavoid output glitches that could lead to impropersystem behavior, such a clock multiplexer needs to be carefully built. There have been many suggested clock multiplexer architectures, each with a uniqueset of characteristics. The fundamental tenet of the majority of the solutions put forth is to turn off the existing clock before switching to and turning on thenew clock that will be present at the output sink. In this respect, flip-flop chains or cross-coupled flip- flops, as described in [1], are the foundation of the majority of widely used stand-alone multiplexer architectures. The remedies outlined there, however,call for switching to be carried out by perpetually running clocks. These multiplexers become stuck and fail to reflect the clock signal from the new clock source if the presently chosen clock is deactivated before the switching has been completed. In the aforementioned example, if the SPI clock is disabledbefore choosing the normal operation mode, switching back from the SPI to the normal operationmode following an SPI transaction is not feasible with conventional methods. There are other designsas well, but they may not be preferred since they require extra monitoring or controlling. In this paper, we show a novel clock multiplexer architecture that gets around these restrictions and ensures that the next clock source will be switched on even if the active clock stops working.

The project is organized as follows: Section II lists the most pertinent architectures and current approaches. Section III describes the overall design and purpose of the innovative clock multiplexer as well as its practical application. Section IV presentsthe experimental findings and the multiplexer's use in a chip architecture. Section V brings the report toa close.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.001 ISSN: 2581-4621

II. RELATED WORK

A well-known problem in the construction of complex systems with various clock domains is clock multiplexing. As a result, many different clockmultiplexer designs have been suggested and protected by patents.



Clk1

Figure 1. Conventional glitch-free clock multiplexer (cf. [1])

In order to avoid glitches, all architectures share thepractice of deactivating the present clock before activating the next clock. A list of basic clock multiplexer designs is provided in [1]. The most sophisticated solution in this summary, shown in Figure 1, is meant to manage clocks that are unrelated to one another (i.e., fully asynchronous). In this project, we will use this architecture as our reference plan. The fundamental elements of the solution are two flip-flop chains that are cross-coupled and function as two-flop synchronizers (cf.[2]), as well as some logic to combine the input clocks and their enable signals to produce the outputclock. The use of two-flop synchronizers prevents the generation of glitches and metastability due to asynchronous inputs of the possibly unrelated clocks, while cross-coupling prevents both clocks from being enabled at the same time. This is how themethod operates: The values of all flip-flops (FFs) are logical 0 in their starting states. Let's then presume that clk0 will be chosen, in which case sel will be set to 0. As a result, signal sel clk0 changes to logical 1, which spreads through the two FFs andcauses en clk0 to do the same. As a result, clk0 is given to clko. To activate clk1 if sel is set to logical-1, sel clk0 must first be set to 0. However, unless the0 has spread from sel clk0 to en clk0, sel clk1continues to be logical-0. When sel clk1 is set to 1, it propagates to en clk1 and then sends clk1 to clko if en clk0 eventually became logical-0. When switching to a different clock source, it is obvious that both clocks must be running in order to first deactivate the existing clock and then activate the new clock.

The same idea of creating activation signs using cross-coupled flip-flops underlies numerous additional solutions [3]–[8]. They only vary in a few
small details, like accelerating the next clock's activation, etc. They are therefore very close to the reference designs in terms of functionality. In particular, if the active clock stops, they all become stuck and are unable to conduct the switching.

There are also methods that call for extra control circuitry or distinct activation signals, such as those suggested in [9] and [10]. As a result, the preventionof activating multiple clock signals necessitates higher organizational layers and more control. As shown, for example, in [11], there are architectures that use timers to monitor the switching behavior ofthe input clock sources. The solution, however, necessitates a second monitor clock, which might not be accessible.

III. NOVEL CLOCK MULTIPLEXER

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.001 ISSN: 2581-4621

A. *General Architecture & Functionality*

Based on the drawbacks of prior approaches and in accordance with the aforementioned application situation, our novel clock multiplexershall be designed with the following characteristics:

- Even if the currently chosen clock becomesinactive, the multiplexer must enable switching from the selected clock to another clock source.
- Only the selection signal and the time sources may be inputs into the multiplexer. Therefore, no additional display clock is needed.
- Because the multiplexer is a standalone solution, no extra circuitry is needed to control the activation and deactivation of the clock sources.

Our clock multiplexer is designed to pick one clock signal from two, similar to the reference implementation in [1].

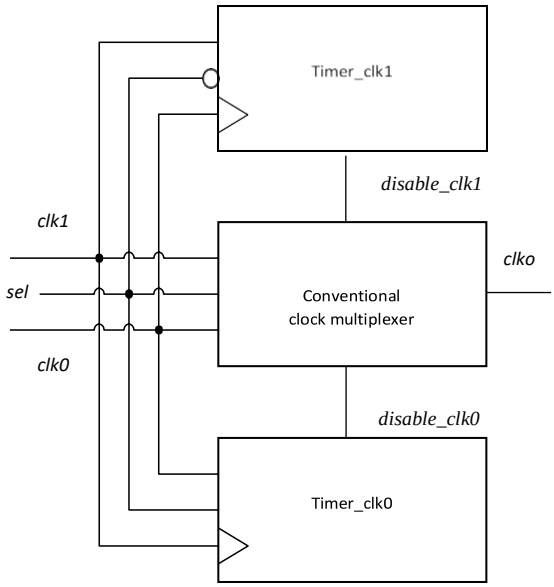


Figure 2. Block diagram of the multiplexer

As a result, the multiplexer receives the selection signal sel, the two clock sources clk0 and clk1, and the output clock clko as input signals. The fundamental design of our inventive multiplexer is shown in Figure 2. As seen in the figure, we employtimers to monitor the clock signals for activity while using a traditional clock multiplexer, such as the standard design from [1].
The fundamental operation is as follows:

- When both clocks are active and a new clock source is selected, our multiplexer acts similarly to the traditional reference design. As a result, after a delay that is at least equal to the smallest period of one of the two input clock impulses, the switch to the next clock is made. The same holds true if a clock is switched from being active to being dormant.
- With the change from an inactive to an active clock, our method is innovative. For each clock source, we therefore add a timer that monitors the activity of the clock signal and shows whether or notthe corresponding clock is running. As a result,Timer clk1 is triggered to watch clk1 for action andTimer clk0 is triggered to watch clk0. Additionally,an activation indication is sent to each timer. Only the clock, which is not presently selected by the input selection, has a timer activated. As a result, thetimer Timer clk0 is enabled if the clock source clk1 is chosen next, and vice versa. Finally, if a timeout is detected, each timer produces a disable signal (disable clk0/1), which is asserted. This signal compels the multiplexer to deactivate the presently chosen clock so that the following clock

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.001 ISSN: 2581-4621

can be turned on.

As previously mentioned, the design is based on a standard clock multiplexer with cross-coupled flip-flop chains. We use FFs with asynchronous reset andadded the two input signals disable clk0 and disable clk1 to dessert the clock enable signals (en clk0 bzw. en clk1). The extra input signals are connected to thetimer outputs as previously shown in Figure 2, as mentioned.

In essence, synchronizers with asynchronous restart are also used to implement timers. As shown in Figure 3, a transition detection circuitry made up ofa delay cell and an XOR-gate can be used to identifythe activity of the clock signals. However, this extracircuit is only required if the clock signal's strength is unknown or unpredictable when the clock is off. Otherwise, based on the clock's degree of inactivityand the active value of the reset input of the synchronizer FFs, as illustrated later, one can omit this circuitry or use an inverter.

Last but not least, the timers are incorporated into the architecture by connecting the multiplexer's inputs as follows: The clock indication for the timerTimer clk0 watching clk0 is clk1. The timer's din isconnected to the selection input sel, and the reset input is connected to clk0, as follows:

- A transition detection circuit must be added if the clock's active status is unknown or subject to change.
- If not, let's suppose that the active values of the clock signal and reset are respectively c 0, 1 and r 0,

1. An converter must be added if r a = 1. If not, the clock can be connected straight to the reset input.

The deactivate signal disable clk0 for clk0 is Timer clk0's output. Similar to this, timer Timer clk1 gets the clock signal from clk0. Finally, Timer clk1's output is the signal disable clk1. The inverted selector's input sel is connected to the timer's din, and clk0 is connected to the reset input.

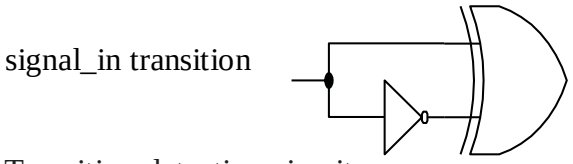


Figure 3. Transition detection circuitry

The design for the resulting clock multiplexer is shown in Figure 4.The circuit operates as follows: the timers are constantly reset while both clocks are running. Thus,the circuit operates in the same manner as the conventional reference design because their outputsare always logical-0. Let's now examine the scenarioin which the currently chosen clock, let's say clk0, stops. Then, to choose clk1, sel is assigned to logical-1. Similar to the traditional clock multiplexer, en clk0 stays logical-0 because clk0 is not operating. But concurrently with this, Timer clk0's input is set to logical 1, activating the timer. After two clk1 cycles, the logical 1 has spread to disable clk0, which resets both en clk0 and the lowest FFs of the traditional multiplexer. As a result, the logical- 1 can spread through the conventional multiplexer's higher FFs, activating clk1 and causingit to be present at the multiplexer's output clko.

c. Timer Dimensioning

The size of the timers, which must be meticulously adjusted to the ratio of the clock frequencies, is a crucial component of the multiplexer's design. The amount of flip-flops must be adjusted in particular toprevent an early and thus unintended reset of the clock enable signal. To avoid problems with metastability, we recommend using at least two FFs.

Consider the following to determine the appropriatequantity of flip-flops: Let clkf be the quickest clockand clks be the slower of the two clocks, clk0 and clk1. Let's now think about the relation f between theclocks' frequencies as specified in 1.

$$\widetilde{f} = \frac{f(clk_f)}{f(clk_s)} > 1 \qquad (1)$$

The lowest integer I N with 1 I 1 f I is the minimumnumber of FFs needed for the clock timer clks, whilethe minimum number of FFs needed for the clock timer clkf is 2. Recall that I 2. It is possible

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Special Issue: 1 Volume No.7 April – 2023
DOI:10.46647/ijetms.2023.v07si01.001 ISSN: 2581-4621

to use a binary counter if the percentage is too high.

IV. EXPERIMENTAL RESULTS & APPLICATIONOF THE MULTIPLEXER

We used CADENCE ADE to carry out analoguesimulations to verify the multi-operation. plexer's As a result, we used IHP 130 nm technology to build the circuit. The patterns produced by these simulations are shown in Figure 5.

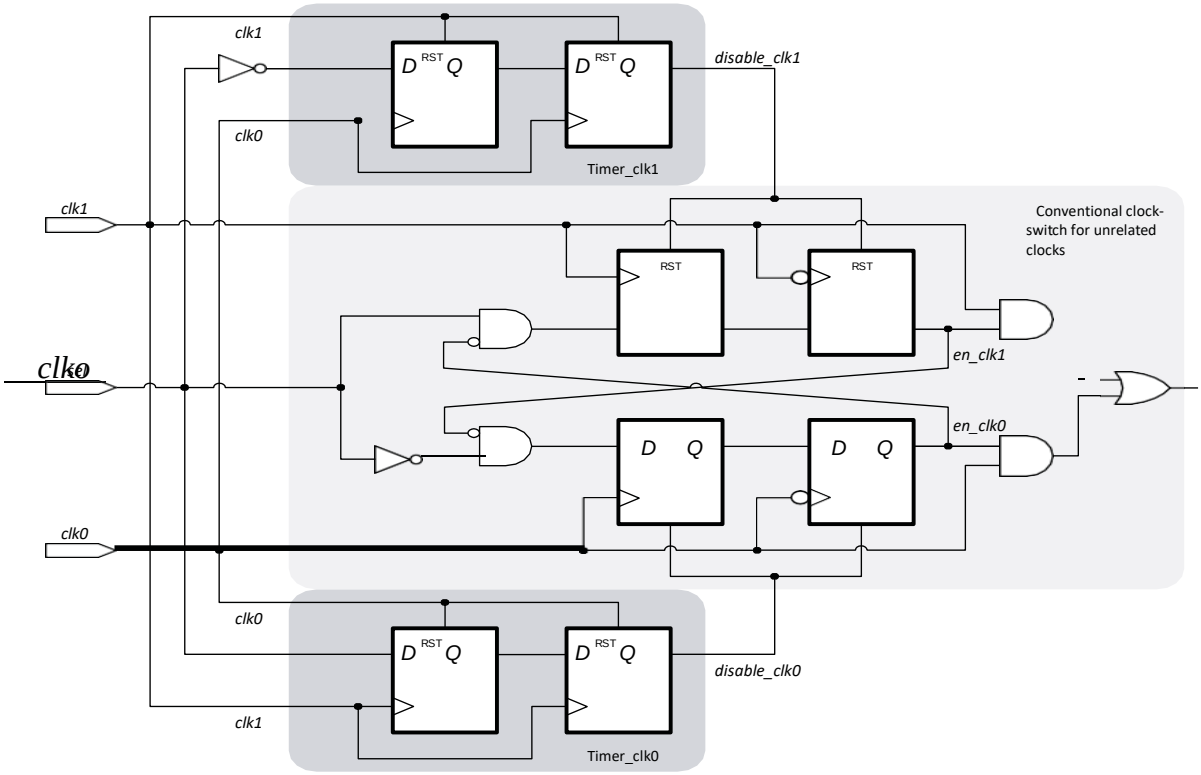


Figure 4. Novel clock multiplexer

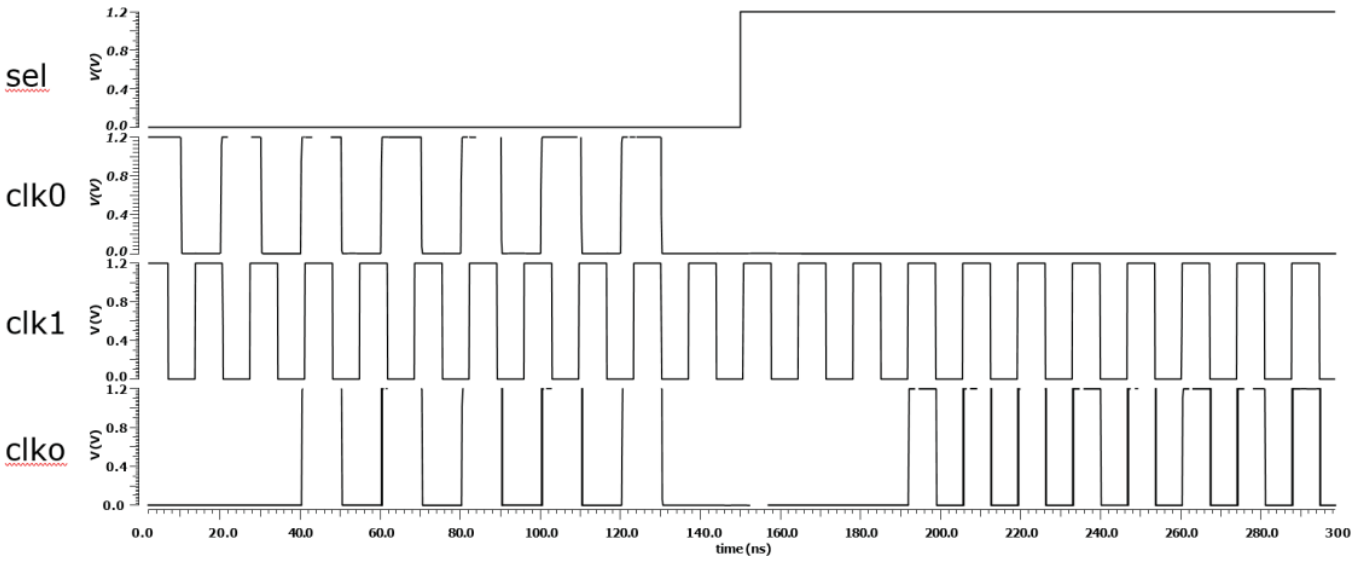


Figure 5. Analog simulation of the clock multiplexer

In addition, for space uses, we have integrated the clock multiplier in a 1.6-2.5 Gbps serializer/deserializer (SERDES) chip. The design has a collection of configuration registers that are

DOI:10.46647/ijetms.2023.v07si01.001 ISSN: 2581-4621

accessed via SPI and are implemented by triple- modular-redundancy (TMR) flip-flops [12]. Since the registers must be refreshed to prevent the accumulation of single-event effects, the SPI clock is required to trigger them in SPI mode, whereas theTX input clock triggers the FF in normal operation mode. So that we could alternate between the TX and the SPI clock, we added the splitter. Only while an SPI transmission is in progress does the SPI clockcome into play. Lastly, we used an Advantest V93kSoC test system to conduct functional tests. Our testing has demonstrated that the design and consequently the combination function as required.

V. CONCLUSIONS

We have introduced a novel clock multiplexer for unrelated possible clocks in this article. This specifically means that the multiplexer allows switching even if the clock that is currently operating stops, whereas cutting-edge solutions get stuck and stop responding. We use timers to monitorthe action of the clocks in order to achieve this behavior. The multiplexer is an easy-to-implement standalone option. Furthermore, no extra clock or external control are needed to check the activity of the clocks. Analog simulations and functional tests of a SERDES chip, which contains two instances ofthe multiplexer, have both been effective in successfully verifying the design.

VI. REFERENCES

[1] R. Mahmud. (2003, Jun.) Techniques to make clock switching glitch free. [Online]. Available: https://www. eetimes.com/document.asp?doc_id=1202359

[2] R. Ginosar, "Fourteen Ways to Fool Your Synchronizer," in *Proceedings of the 9th International Symposium on Asynchronous Circuits and Systems*. Washington, DC, USA: IEEE Computer Society, 2003, pp. 89–. [Online]. Available: http://portal.acm.org/citation.cfm?id= 785169.785383

[3] M. E. Heimann, "Glitch-free clock multiplexer," US Patent 5 357 146A, 1994. [Online]. Available: https: //patents.google.com/patent/US5357146A/en

[4] J. P. Miller and M. S. Vacanti, "Glitchless clock switch circuit," US Patent 6 275 546B1, 2001. [Online]. Avail- able: https://patents.google.com/patent/US6275546B1

[5] S. P. Young, "Clock multiplexer circuit with glitchless switching," US Patent 6 429 698B1, 2002. [Online]. Available: https://patents.google.com/patent/US6429698

[6] B. S. Haroun, H.-C. Lin, and T. Foo, "Glitch free clock multiplexing circuit with asynchronous switch control andminimum switch over time," US Patent6 784 699B2, 2004. [Online]. Available: https://patents.google.com/patent/US6784699

[7] A. Klindworth, "Secure asynchronous clock multiplexer,"US Patent 6 535 048B1, 2003.

[8] F. Boutaud, "Glitch free clock select switch," USPatent6 600 345B1, 2003. [Online]. Available: https: //patents.google.com/patent/US6600345B1

[9] G. A. Lapiana, "Glitch free asynchrnous clockmultiplexer," US Patent 20 180 145 689, 2018. [On-line]. Available: http://www.freepatentsonline.com/y2018/ 0145689.html

[10] C. Walker, R. Parry, and J. A. Drummond-murray, "Glitch free clock multiplexer circuit," US Patent 6 265 930, 2001. [Online]. Available: http://www.freepatentsonline. com/6265930.html

[11] (2019, Feb.) Glitch free clock multiplexer(mux).RTLery. [Online]. Available: http://rtlery.com/ components/glitch-free-clock-multiplexermux

[12] V. Petrovic and M. Krstic, "Design Flow for Radhard TMR Flip-Flops," in *Design and Diagnostics of Electronic Circuits Systems (DDECS), 2015 IEEE 18th International Symposium on*, April 2015, pp. 203–208.