INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

DESIGN AND ANALYSIS OF AREA EFFICIENT NOVEL MESOCHRONOUS FIFO BUFFER

Sirasapalli Lakshmi Nivas Teja[1], Simhadri Likith[2], Dr.G. Anand kumar[3]
[1] Mtech, Dept. of ECE, NIT Jaipur, Jaipur, Rajasthan., India
Email Id: nivasteja1997@gmail.com
[2] Btech, Dept. of ECE, Gayatri vidya Parishad College of engineering, Vizag, A.P., India
Email Id: likith.simhadri@gmail.com
[3] PhD scholar, Dept. of ECE, Gayatri vidya Parishad College of engineering, Vizag, A.P., India
Email Id: ganand@gvpce.ac.in

Abstract:

Every memory in which an information word is engraved in first as well as it moves toward out first once the memory is read is a first-in first-out FIFO memory. Tranquil Clocking systems switched with effusively synchronous clocking, for instance Mesochronous clocking, to promote system composability and allow timing closure. Mesochronous clock signals are often used for time signalling operations in synchronous memory systems. By using the same clock source, under the identical clock frequency with different phase difference. In such cases, clock synchronization is desirable for transmission of data across the modules. In this project, we are employing a novel mesochronous FIFO buffer which can knob together clock synchronization. Clock synchronization is compulsory for the organization of happenings as well as to reserve the state of properties. Even if the transmitter and receiver are detached by means of elongated link whose latency is too prodigious to adequately privileged the mark of the functional frequency, the suggested methodology will work aptly. In such popular circumstances like this, the suggested methodology of mesochronous First in First out buffer could be sustained to multiple-cycle linkage latencies in modular manner in addition to with slight adjustments to the baseline architecture. The proposed method is synthesis as well as verified in Xilinx ISE 14.7 version tool.

I. INTRODUCTION

The modern SoC (System on Chip) realized in extremely scaled technologies appears low wires as

well as process PVT variations. Due to these difficulties, synchronous abstraction is becoming increasingly unworkable over vast chip areas, necessitating significant design determination to accomplish timing closure in various author proposals. Since synchronous operations as well as their associated timing constraints are confining, segregating the SoC into universally asynchronous as well as locally synchronous domains in the previous approach only partially alleviates the issue.

FIFO is abbreviated as First in First Out, which partakes a read/write memory arrangement; it could spontaneously preserve the trajectory of the instruction in which data writes (data_in) into the module as well as reads the data out in the same way.

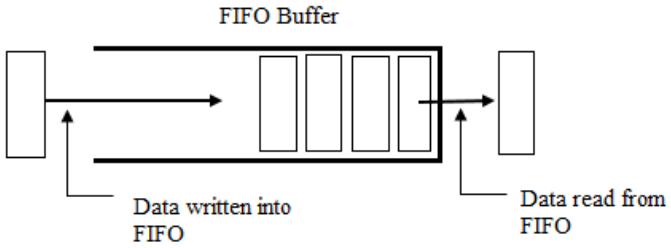


Fig.1 FIFO buffer

Buffering as well as flow control among hardware as well as software are typical uses for FIFOs in electronic circuits. A FIFO is mainly made up of a series of read plus write pointers, as well as storage and control logic, in its hardware form. Static random access memory (SRAM), flip-flops, latches, also any additional appropriate storage device may be used.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES
ISSN : 2454-9924

FIFO full-empty:

For the synchronization process in hardware FIFO buffer is used. The FIFO buffer has two indicators:

[1] Read pointer/Register to read address
[2] Write pointer/Register to write address

FIFO empty:

Whenever the read address register matches the write address register, then the FIFO I said to be empty.

FIFO full:

Whenever the read address Least Significant Bits equivalent the write address LSBs as well as the additional MSBs are distinct, the FIFO is full.

Contemporaneous read/write FIFOs, established on the controller indicators for writing as well as reading, alienated obsessed by double individuals:

- Synchronous FIFOs
- Asynchronous FIFOs

Synchronous FIFO Buffer:

- A synchronous FIFO refers to a FIFO architecture, the values of data written successively into a memory array employing a clock signal, as well as the data values remaining read out successively from the memory array utilizing the identical clock signal.

Asynchronous FIFOs:

- Asynchronous FIFO refers to various clocks employed to control the read as well as write operations. So the crossing clocking domain concept is used in this FIFO for the communication between read and write operations.

Clock domain crossing:

Asynchronous FIFO utilizes different clocks to perform the operation of reading as well as writing. Asynchronous FIFOs introduce the meta-stability issues the clocking cross domain concept is used here. In digital electronic design clock domain crossing is used to avoid the meta-stability state.

The various clock domains will have dissimilar clock frequencies as well as dissimilar phase or both. The synchronization of single bit to clock domain with greater frequency can be proficient by registering the signal via flip flop. To ignore the meta-stability state in the terminus area, at least two stages of re-synchronization flip-flops are comprised in the terminus area.

Meta-stability of Synchronous Circuits:

The absolute period of the every clock cycle diverges slightly, as well as sooner or later prominent to the Metastability state, adjacent ample to respectively supplementary switches. Recurrently Metastability will ensue with the amalgamation of customary demonstration strategies. Logical ‘0’ or ‘1’ level surrounded by the period compulsory for appropriate circuit manoeuvre, at the Metastable federations the circuit is impotent to fluster into a steady. One communal approach to establish metastability is to resource dual clocks which fluctuate precisely marginally in frequency to the information in addition to clock involvements.

We use a synchronizer to synchronize the signals in order to prevent a meta-stability condition. To prevent the meta-stability condition, most people use a two flop or two stage synchronizer. The synchronizer’s entire perseverance is to minimize the catastrophe likelihood. Remember that the synchronizer doesn’t synchronize the information moderately, it synchronizes the controller pointers.

By giving the delay to evade Metastability in asynchronous systems. Synchronizer will afford the satisfactory delay to overwhelm this state.

In asynchronous systems, metastability is unavoidable, but smart design may typically avoid the issue of breaching setup and hold times. The metastability properties of a device are determined by the process technology utilised in its construction as well as the surrounding environments. They've become more common as operational frequencies have increased.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

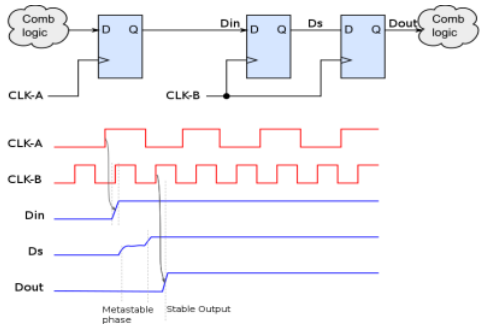


Fig. Timing diagram of Two-Level Synchronization

Multiple level synchronization may be utilized to eradicate meta-steadiness. A twofold flop synchronizing route demonstrates auxiliary misery. The latency of the principal flop could be encouragingly enriched because of the meta-steadiness.

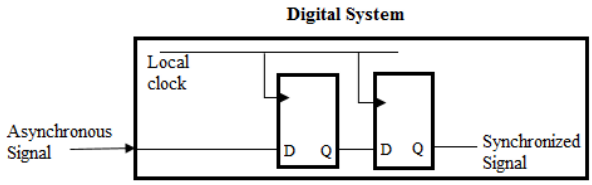


Fig.2 Block Illustration of Two-Level Synchronization

The ultimate goal of this article is to incorporate the advantages of both loosely coupled as well as tightly coupled methods. In particular, a unique buffering architecture is proposed that incorporates Mesochronous synchronization as well as magnificently syndicates area reliability, high performance, as well as maintenance for multi-cycle link delays.

II. RELATED WORK

There are twofold foremost approaches for proficient synchronization as well as buffering transversely mesochronous interfaces.

Those are loosely combined execution, in this synchronization and buffering happen separately as well as tightly combined execution, those are united as well as attached and hooked on a distinct structure. In this project our definitive theme is to amalgamate the reimbursements of mutually loosely coupled and tightly coupled methodologies. In the existing method, a new buffering structure that adds mesochronous synchronization is implemented, it produces area efficiency, and high performance as well as it supports multiple cycle linkage latencies.

The recommended mesochronous twofold clock FIFO buffer structural architecture is the foremost to undertake information synchronization meanderingly.

This optimized architecture provides the benefits which are provided by synchronizer: 1.Mesochronous is the first design to prove lossy manoeuvre under whichever safeguarding distance>1, 2. Negligible buffering necessities for full throughput, 3. which supports multi cycle links without other restrictions.

Mesochronous Synchronization and Buffering:

There are various methodologies to send the data across two mesochronous clock domains. N flip-flop synchronizer is the most suitable approach. In this n register and n flip-flops were employed in equivalent in the communication purview as well as twofold counters which are enlarged in every single cycle. A 2-flop synchronizer can't assure that the outcome information will persist steady all the way through an entire receiver clock. The three flip-flop synchronizer is mandatory to manoeuvre under the several skew phase as well as deprived of employing the phase sensor.

In this case, at metastable condition we can send any of two counter reset signals. We need a 4-flop synchronizer to confirm accurate manoeuvre at any phase difference amongst dual clocks. At every single clock cycle, the synchronized data is committed to read at the receiver end. A push or pull flow control is used on the mesochronous interface, so that transmission of data from the transmitter stops when the receiver buffer is full. When FIFO is full, it displays the full signal indicated; the incoming data is valid or not indicated by the push signal which is driven by the transmitter. In the loosely as well as tightly coupled cases, 3-flop synchronizers are sufficient, during reset condition clocks are de-asserted. Instead of it, in case of asynchronous reset as in Star sync a 4-flop synchronizer remains desirable.

Amended synchronizer in loosely as well as tightly merged architectural structures, which is a reset mechanism.

Architecture and operation:

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

The existing method Mesochronous FIFO is shown in figure. At contributor purview, information which necessities to be present coordinated is dropped in recollection. The tail indicator locates the transmitter-synchronous to the memory position where the novel data word is written, while the head indicator locates the receiver-synchronous to the location where a data word will be received to read out. En-queue (write) and de-queue (consume) events are synchronized among both sides using a couple of contrasting-way one-bit 4-flop synchronizers.

Whenever the transmitter transfers a synchronized data word, the data is written to the memory location indicated by utilizing the tail pointer. The pop signal is proclaimed, as well as the incremented head pointer positioned at the location where the subsequent data word is located, until the receiver has actually consumed the data.

In the queue, the receiver utilizes the status counter and it will count the synchronized data items. The counter increased; whenever the original data word is acknowledged, as signified by an inward push signal commencing the Tx synchronizer is displayed in below figure. While a data word expanded, the counter decremented to reveal the alteration in push state. In this approach the two significant goals are:

1). Data remain explicitly synchronized via the implicit synchronization of the push trials,

2). First in First Out guidelines were well-maintained in the buffer.

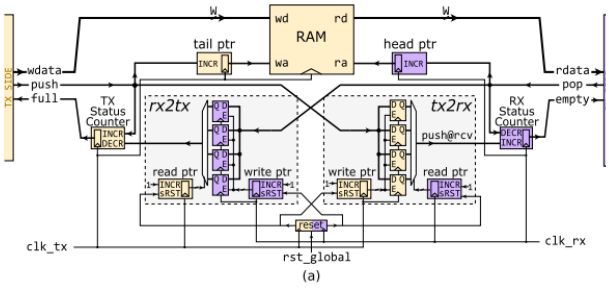


Fig.3. Existing Mesochronous FIFO Buffer

The next step is to synchronize the queues start that transmitter so that transmitter and receiver synchronization at the queue doesn't overflow. To reach it the transmitter correspondingly utilizes a status counter as displayed in figure 4(a), to preserve its substances presently deposited in the queue. Either the counter will be incremented or else decremented an element is en-queued or de-queued from the queue individually. De-queue events are synchronized, they have to be coordinated to the source block over and done with the distinct diffident synchronizer. On de-queue the receiver emphasizes the pop indication of receiver to transmitter. Once signal synchronization is done, the transmitter diminishes its prominence counter commendably left in excess of in sync with the decrement in buffer state. The accelerative latency of the synchronizing as well as en-queuing a novel data element among first as well as third cycle, depending on the cycles which are spread with synchronous counter.

Then again, the reset of the write pointer will be delayed, which reduces forward latency. The progressive latency i.e. amount of cycles required to synchronize the pop events, is also among one as well as third cycle. However, in the worst case forward as well as backward latencies do not occur simultaneously. The write & read pointers, which are powered by the tx side's synchronized reset signal, behave similarly. As a result, if one side has a three-cycle latency, the other side will have a one-cycle latency. At four cycles, the number of the forward as well as backward delay is unchanged.

Support for Multi-cycle Basis-Synchronous Contacts: The Mesochronous FIFO design block reveals Transmission as well as Receiver are physically same, so there are no timing delays.

Whenever this argument is flawed, such as when the transmitter as well as receiver segregated by a lengthy physical connection with a delay that extends the goal functioning frequency, when this proclamation fails, together with when the transmitter as well as receiver are disconnected by a long corporal connection whose delay exceeds the aim functioning frequency, the recommended Mesochronous First In First Out could be protracted to maintain multi-cycle link delays in a scalable system with negligible changes to the reference design.

One simple solution is to put the memory as well as the receiver end of the header indicator, so

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

that if a word is processed, the next data object can be recovered instantly in the next loop. The subsequent data element would arrive at the receiver with a (RB + RF) sequence latency if memory besides head baton remained to be found, formerly the multiple cycle linkage: the RF sequences for subsequent data term to attain at receiver.

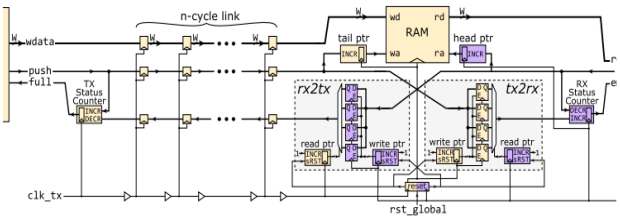


Fig.4. Restructuring of the existing Mesochronous

FIFO

To fit in the single cycle, the length among the transmitter as well as receiver is too lengthy. To perform the forward & backward data as well as flow control signal, the link will be splitted into two multiple register platforms.

The empty & full conditions of the FIFO on the receiver as well as transmitter sides, we have to select the Rx as well Tx status counters correspondingly. Rather than transmitting as well as synchronizing push & pop activities, the suggested approach allows the state to also be defined globally upon one another.

III. METHODOLOGY

The novel Mesochronous FIFO buffer is implemented in the proposed method, by considering the baseline architecture of existing methodology, modified to multi cycle design, which can knob clock synchronization as well as short-term data loading. FIFO synchronizer will transfer the true data from transmitter to receiver. To avoid metastability, synchronizers are implemented with pointers, D flip-flops as well as multiplexers etc. The hardware complexity will become more for the proposed architecture.

In FIFO synchronizers are there to transfer the true data from transmitter Tx to receiver Rx. Pointers, multiplexers, delay flip flops as well as counters are there in synchronizer to avoid meta-stability state. The total hardware of the architecture will increase because of consuming more area.

In scalable manner, the proposed a novel Mesochronous FIFO buffer combines the buffering as well as Mesochronous clock synchronization. Without being expressly synchronized, data is securely transmitted. Merely only-bit push or else pop controller signals on either side of the FIFO are used for synchronization. As shown in figure 3, we will adjust the synchronizer circuit in this proposed method to minimize the design's delay and power. To escape the meta-stability condition in FIFO, the proposed design only uses a Delay flip-flop as well as a multiplexer. The delay flip-flop produced enough delay to prevent the state of meta-stability. As a result, the current approach employs a delay flip-flop, multiplexer, clock.

As a result, these synchronizers are needed in the proposed method to communicate among the Tx and Rx sides in order to correctly transfer data with some delay. In comparison to the current system, this implementation of Synchronizers would shrink the hardware complexity of the design.

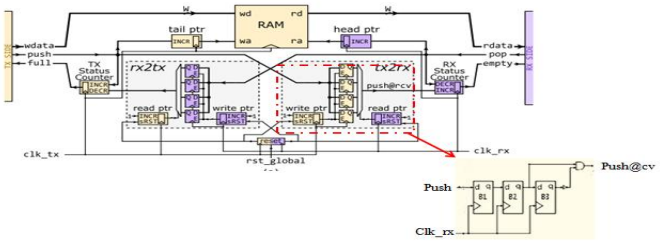


Fig.5. Proposed Mesochronous dual clock FIFO

The data word is essential to be synchronized when it is received from the transmitter, where the tail pointer positioned in the memory the data should be written in that location. Now the tail pointer is incremented, the forward data tx2rx proposed Mesochronous [see Fig. 5] is proclaimed when the push signal will push the data simultaneously. Where the header pointer positioned in the memory addresses the data to be read from the receiver. The pop signal is proclaimed, as soon as the receiver receives the data, as well as we can see the increment in header point will be changed to the next position to find next data.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

Unless a novel push occurrence has been synchronized, the receiver doesn't read the data word from the reorganized head pointer position, which indicates that reorganized data to be present as well as safe to be read out.

A very subsequent period is to coordinate the chain's phase from the transmitter as well as assurance that the queue doesn't overspill. The transmitter will utilize the state counter, to accomplish this as displayed in fig. 4(a).

Either the counter is enlarging or else lessening, whenever the element is en-queue or else de-queued from the queue accordingly.

When the receiver proclaims the pop signal of the receiver to transmitter (rx2tx) synchronizer on de-queue. As soon as the signal remains synchronized, the transmitter diminishes its status counter, efficiently outstanding in sync through the buffer's state.

IV. RESULTS AND DISCUSSION

In the proposed method when data is entered into the FIFO it will be encrypted the push operation will perform, while performing decryption the data will pop in the data out.

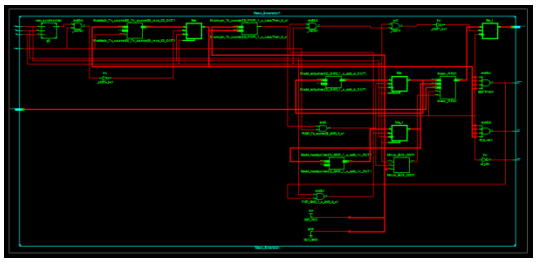
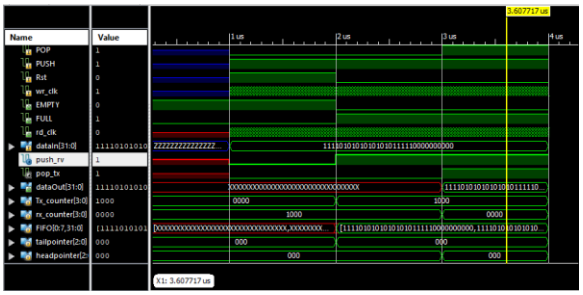


Fig.6. RTL schematic of proposed Mesochronous FIFO buffer

Simulation results:

Proposed Mesochronous FIFO Buffer:



Evaluation table for Area, Delay and power:

	Area	**Delay(ns)**	**Power(W)**
Proposed	92	25.418	0.014
Existing	97	33.227	0.014

V. CONCLUSION

In the scalable manner, Mesochronous clock synchronization as well as buffering are mingled by the dual clock FIFO. A new Mesochronous dual clock FIFO buffer implemented. Delay flip flop as well as some basic gates are positioned in proposed architecture to eliminate the meta-stability state in FIFO. The sufficient delay will be provided by the D-flip flop to eliminate meta-stability state. To transfer the data safely from transmitter to receiver side of Mesochronous interface without synchronization explicitly. By relating with the traditional method, the area will be reduced by minimizing the synchronizer circuit.by utilizing Xilinx ISE 14.7 version tool, the proposed methodology is synthesized as well as verified.

VI. REFERENCES

[1] T. Chelsea and S. M. Nowick, Robust interfaces for mixed-timing systems, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 8, pp. 857–873, Aug. 2004.

[2] A. M. S. Abdelhadi and M. R. Greenstreet, Interleaved architectures for high-throughput synthesizable synchronization FIFOs, in *Proc. 23rd IEEE Int. Symp. Asynchronous Circuits Syst. (ASYNC)*, May 2017, pp. 41–48.

[3] P. Teehan, M. Greenstreet, and G. Lemieux, A survey and taxonomy of GALS design styles, *IEEE Des. Test Comput.*, vol. 24, no. 5, pp. 418–428, Sep./Oct. 2007.

[4] J. Ax, N. Kucza, M. Vohrmann, T. Jungeblut, M. Porrmann, and U. Rückert, Comparing synchronous, mesochronous and asynchronous NoCs for GALS based MPSoCs, in *Proc. IEEE MCSoC*, Sep. 2017, pp. 45–51.

ISSN : 2454-9924

[5] R. Ginosar, Metastability and synchronizers: A tutorial, *IEEE Des. Test Comput.*, vol. 28, no. 5, pp. 23–35, Sep./Oct. 2011.

[6] R. W. Apperson, Z. Yu, M. J. Meeuwsen, T. Mohsenin, and B. M. Baas, A scalable dual-clock FIFO for data transfers between arbitrary and haltable clock domains, *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1125–1134, Oct. 2007.

[7] D. Verbitsky, R. R. Dobkin, R. Ginosar, and S. Beer, StarSync: An extendable standard-cell mesochronous synchronizer, *Integration*, vol. 47, no. 2, pp. 250–260, Mar. 2014.

[8] D. Ludovici, A. Strano, D. Bertozzi, L. Benini, and G. N. Gaydadjiev, Comparing tightly and loosely coupled mesochronous synchronizers in a NoC switch architecture, in *Proc. NOCS*, May 2009, pp. 244–249

[9] D. Ludovici, A. Stranoy, G. N. Gaydadjievx, L. Beniniyy, and D. Bertozziy, Design space exploration of a mesochronous link for cost-effective and flexible GALS NOCs, in *Proc. IEEE DATE*, Mar. 2010, pp. 679–684.

[10] S. Saponara, F. Vitullo, R. Locatelli, P. Teninge, M. Coppola, and L. Fanucci, LIME: A low-latency and low-complexity on-chip Mesochronous link with integrated flow control, in *Proc. 11th EUROMICRO Conf. Digit. Syst. Design Archit., Methods Tools*, Sep. 2008, pp. 32–35.

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924