International Journal of Scientific Research in Computer Science and Engineering
Research Paper Vol-1, Issue-3 ISSN: 2320–7639

Metastability Analysis of Synchronizer

Ankush S. Patharkar[*1] and V. E. Khetade[2]

[*1]*Department of Electronics, Shri Ramdeobaba college of engineering and management, Nagpur, India*
[2]*Department of Electronic, Shri Ramdeobaba college of Eng. and management, Nagpur, India*

Available online at www.isroset.org

Received: 20 March 2013 Revised: 28 March 2013 Accepted: 22 May 2013 Published: 30 June 2013

Abstract- The multiple clock domain systems communicate with each other causes data loss. This data loss is due to mismatch in frequencies. For proper communication the frequencies must be synchronized. Hence synchronizer is used for data synchronization process with non-zero probability of failure. The synchronizer is also having its parameters. It suffers from metastability as data changes in between timing window due to which synchronizer failure occurs. As the metastability occurs we cannot predict the correct level at output. The proposed architecture is modeled with verilog and simulated with Xilinx ISE design suit 13.1and its parameter are verified with Quartus II 10.1. Analog behavior is studied by using Tanner. The probability of occurrence of metastability reduces with timing window width.

Keywords: Data synchronization, data loss, synchronizer, metastability.

1. INTRODUCTION

System on chip is an integrated circuit that integrates all components of electronic system on single chip. It may contain digital, analog, mixed signal and radio frequency functions on single chip substrate. System-on-chip that consists of multiple clock domain connected through asynchronous interconnect are called globally asynchronous and locally synchronous Network-on chip (GALS).

Globally asynchronous locally synchronous (GALS) is a model of computation. It is based on synchronous MoC and asynchronous MoC. It allows to relax the synchrony assumption to model and and design computer system. Frequency and phase of each synchronous module (IP core) is independent.

1.1 Synchronizer:

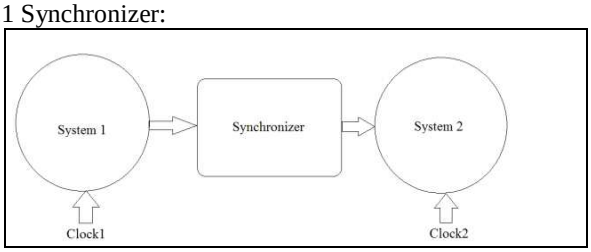

Fig1.Introduction to synchronizer in two clock domain

Consider two digital systems which are working at different clocks say system1 is working at clock1 and system2 is working at clock2. When the data is transferred from one clock domain to another clock domain data synchronization is required. These two digital systems require a block to communicate with each other. This synchronization is performed by the synchronizer. A synchronizer samples an asynchronous signal and outputs a version of the signal that has transitions synchronized to the local clock.
The significance of the synchronizer is it tries to synchronize the data transfers from one system to another. `The parameters associated to synchronizer are clock, data rate and timing window (setup and hold time).

Corresponding Author: *Ankush S. Patharkar*

1.2 Need of Synchronizer:

1) Any asynchronous input from the outside world to a clocked circuit represents a source of unreliability.
2) There is always some residual probability that the clocked circuit will sample the asynchronous signal just at the time that it is changing.
3) From a specification point of view, synchronous elements such as flip flops specify a Setup time and a Hold time.
4) The problem of metastability propagating into synchronous systems is to use a synchronizing circuit to take the asynchronous input signal, and align it to the timing regimen of the rest of the system.
5) To minimize the probability of this occurring and propagating to the output of the synchronizer.
6) This can usually be achieved with a two stages or three stages synchronizer.

1.3 Synchronizer failure and Metastability:

Metastability is a potentially catastrophic event that can occur when asynchronous inputs and flip-flops are used. As the flip flops have only two stable states i.e. 0 and 1. But all have third metastable state half way between 0 and 1. At this metastable state we cannot predict the output. The metastability occurs when setup and hold time violated

which puts the synchronizer into metastability. When an output becomes metastable, it will be between the high and the low states. After a certain amount of time, the output unpredictably reverts to either a high or a low state. Metastability can be predicted by using the equation of MTBF.

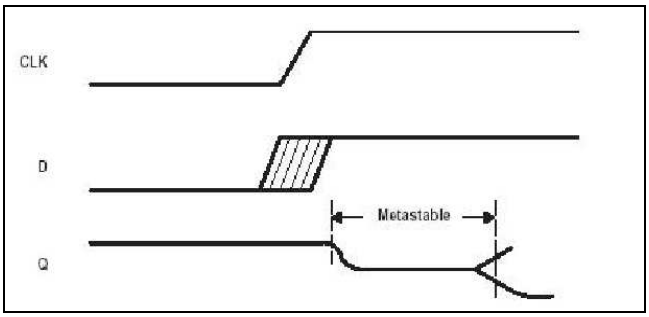

Fig2. Occurrence of metastability

If the system uses a synchronizer output while the output is still in the metastable state causes synchronizer failure.

2. ANALYSIS OF EXISTING SYNCHRONIZER

Synchronizers are classified as follows,

1) Two flop synchronizer.
2) Data delay synchronizer.
3) Dual data delay synchronizer.
4) Clock delay synchronizer.
5) FIFO synchronizer.
6) Clock edge synchronizer.
7) Low latency clock edge synchronizer.

2.1 Comparison of various synchronizers

Synchronizer	Latency (cycles)	Throughput
Two flop	1-2	1/Latency
Data delay	0-1	1
Dual data delay	0-1	1
Clock delay	0-1	1
FIFO	0-1	1
Clock edge	5	1
Low latency clock edge	0-1	1

2.2 Two flop synchronizer:

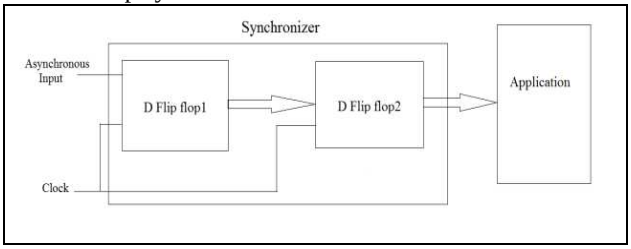

Fig3. Two flop synchronizer with two D flip flop

The reliable synchronizer design consisting of cascaded two D flip flop in which asynchronous input is applied from another application. As we know when setup and hold time violated then metastability occurs. The probability that a flip-flop stays in the metastable state decreases exponentially with time. Therefore, any scheme that delays using the signal can be used to decrease the probability of failure. If the clock period is greater than metastability resolution time plus FF2 setup time, FF2 gets a synchronized version of ASYNCIN. Multi-cycle synchronizers (using counters or more cascaded flip-flops) are even better.

Probability of the failure is given as,

$$p(failure) = p(enter\ MS) \times p(time\ to\ exit > S) = T_W F_C \times e^{-S/\tau}$$

Rate of entering metastability computed above to drive the rate of expected failure is,

$$Rate(failures) = T_W F_C F_D \times e^{-S/\tau}$$

The inverse of failure rate is MTBF.

$$MTBF = \frac{e^{S/\tau}}{T_W F_C F_D}$$

The MTBF can be calculated by using Quartus II software for the synchronizer.

2.3 Circuit:

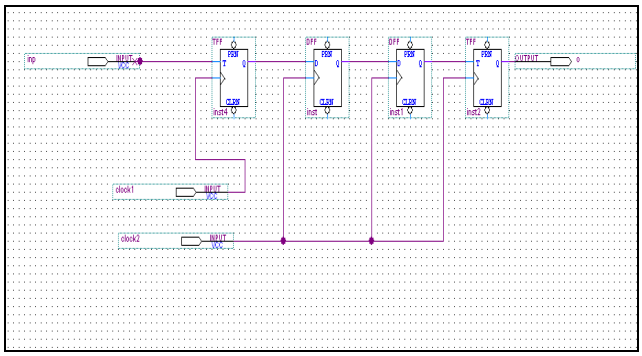

Fig4. Synchronizer chain

2.4 Observed Simulation result

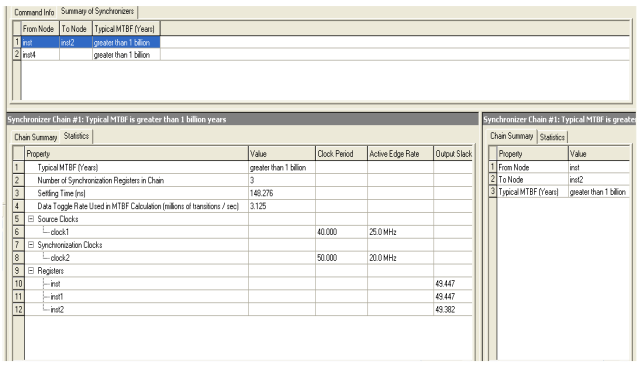

Fig5. Simulation result

3. Data rate modeling for analyzing the synchronizer:

Data rate is defined as the maximum data bytes transferred by the transmitter at its operating frequency. As the synchronizer is used between two clocks domain application the sender transmits the data at particular data rate. The data rate may vary depends upon the application which is to be applied to synchronizer. There may be many application transmits data with their data rate applied to only one synchronizer. So in order to study how synchronizer handles data rate, for this the data rate model using various blocks is shown below,

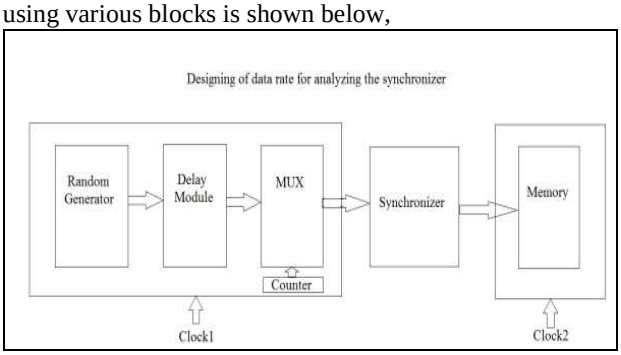

Fig6. Data rate modeling

It consisting of random number generator, delay module, multiplexer, counter, synchronizer and memory. Application 1 (random generator, delay module, mux, counter) is working at clock1 while synchronizer and application 2(memory) is working at clock 2. The random generator is nothing but the LFSR which generates random data bytes. Delay module delays these bytes at particular interval. Mux and counter generate the data rate. This variable data rate is given to the synchronizer, it tries to synchronize the data with memory which are working at different clock.
Here the three cases are studied shown below,

3.1 Case 1)→ Sender with frequency 25 MHz and receiver with frequency 50MHz without synchronizer. Here in this case there will be no data loss as receiver is working at more frequency.

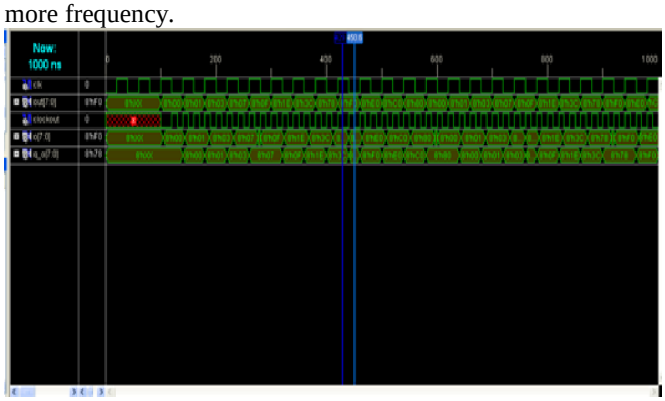

Fig7. Simulation result shows no data loss

3.2 Case 2)→ Sender with frequency 25MHz and receiver with frequency 20MHz without synchronizer. Here in this case there will be data loss as frequency of receiver is less.

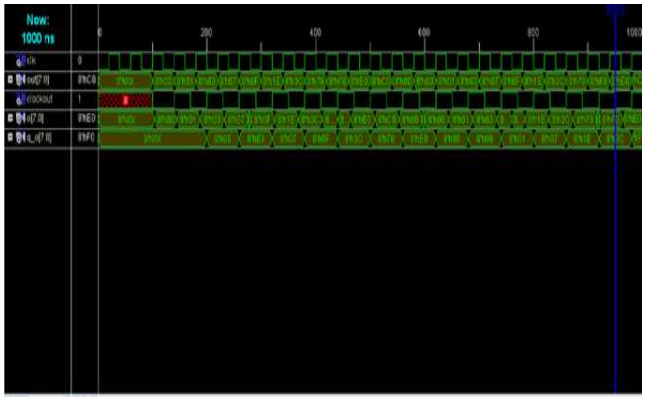

Fig8. Simulation result shows data loss

Following graph shows the comparative percentage of data loss with frequency at receiver side.

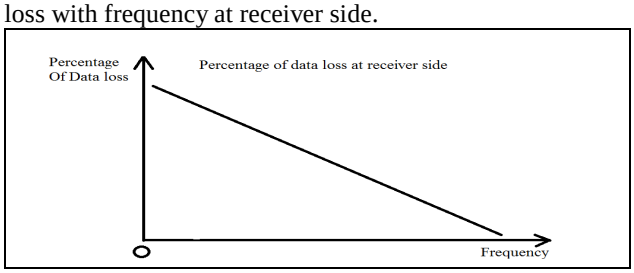

Fig9. Graph shows percentage of data loss

Thus the percentage of data loss decreases with increase in frequency. As above two cases shows if sender is having frequency say F1and receiver have F2 and F2>F1 then there is no data loss. But F1>F2 causes significant data loss depending upon the difference in F1 and F2.

3.3 Case 3)→ Sender with frequency 25MHz and receiver with frequency 20MHz with synchronizer. The synchronizer is having three parameters i.e. clock, data rate and timing window (setup and hold time). Due to setup and hold time violations, the synchronizer goes in metastable state.

Following tables shows timing window violation which causes metastability,

1) Sender frequency=25MHz
   Receiver frequency=20MHz
   Setup time=20 ns
   Hold time=25 ns

Data (D)	Clock	Setup/Hold violation
232	235	20
232	235	25
280	285	20
280	285	25
324	335	20
324	335	25
368	385	20
368	385	25
412	435	25
464	485	25
732	735	20

732	735	25
784	785	20
784	785	25
828	835	20
828	835	25
872	885	20
872	885	25
920	935	20
920	935	25
964	985	25

2) Sender frequency=25 MHz
Receiver frequency=20MHz
Setup time=4ns
Hold time=5ns

Data (D)	Clock	Setup/Hold violation
232	235	4
232	235	5
732	735	4
732	735	5
784	785	4
784	785	5

From above tables we conclude that as the timing window decreases the violation decreases which leads to decrease in metastability.

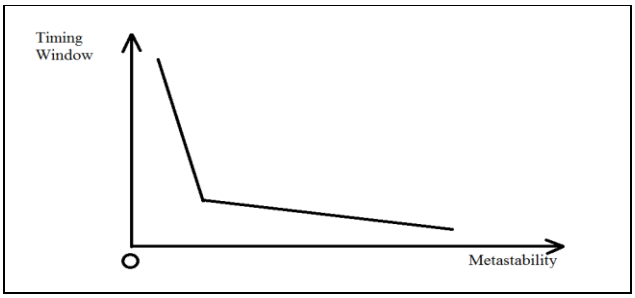

Fig10. Graph shows reduction in Metastability

4. ANALOG MODELING FOR ANALYZING METASTABILITY

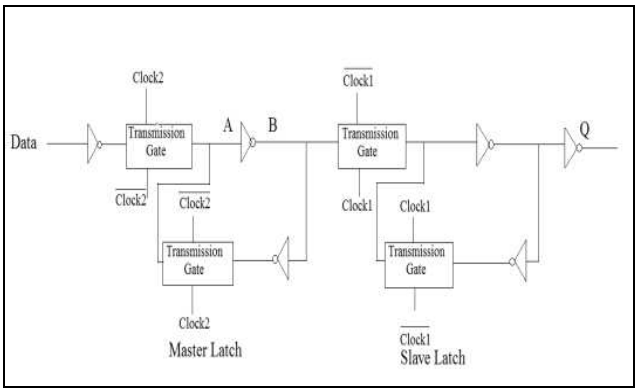

Fig11. Analog modeling

Analog modeling consisting of master latch, slave latch and transmission gates as shown above. In metastability, the voltage levels of nodes A and B of the master latch are roughly mid-way between logic 1 (VDD) and 0 (GND). Exact voltage levels depend on transistor sizing and are not necessarily the same for the two nodes.

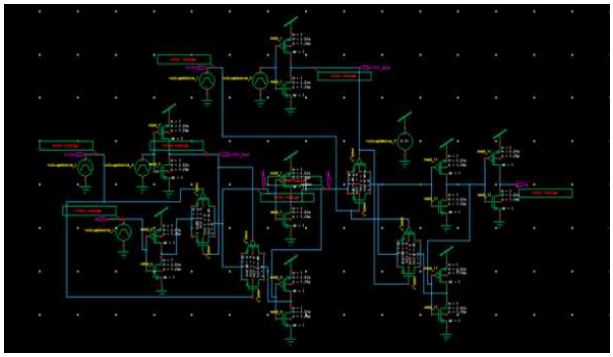

Fig12. Circuit realization

4.1 RESULT ANALYSIS

As the clock becomes low the voltage at node A becomes high and voltage at node B becomes low. As node A is high and input Data (D) changes from 0 to 1, the voltage at node A drops and voltage at node B rises. Again D changes from 1 to 0 and still node A is high, shows voltage drop at node A. This condition occurs only when node A is high and output is low.

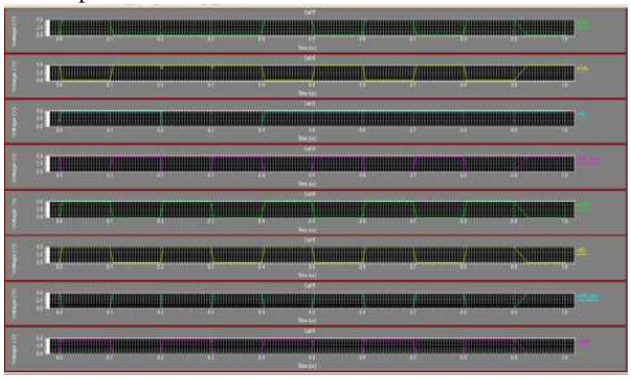

Fig13. Simulated result

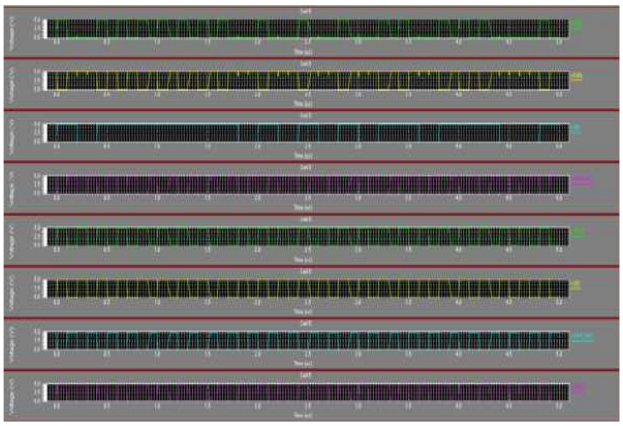

Fig14. Simulated result

However the violation due to change of input in timing window can be overcome and we can exit from metastability.

5. CONCLUSION

When many applications works on different clock then for better communication between them needs synchronizer. Synchronizer tries to synchronize the different clock domains for communication. But due to specification parameters of synchronizer causes metastability. However this metastability cannot be avoided as input to the synchronizes is asynchronous. But we can minimize the occurrence of metastability by parameters of synchronizer. Synchronizer performance is based on MTBF(mean time between failures).

As the data rate increases MTBF reduces which increase the probability of failure of synchronizer. The failure of synchronizer occurs only when data changes in timing window. However metastability can be reduced if timing window is very small that again depends upon application. Also the percentage of data loss reduces with increase in frequency at receiver side with respect to receiver frequency.

REFERENCES

[1] "Keeping Metastability from Killing Your Design," Grosse, Debora. Unisys Report. June 23, 1994. Available[online]hppt://archives.einsite.net/archives/ednmag/reg/1994/062394/13df2.htm

[2] Metastable Response in 5-V Logic Circuits," Haseloff, Eilhard. Texas Instruments Report.Feb1997available[online]http://www.ti.com/sc/docs/psheets/abstract/apps/sdya006.htm.

[3] "Metastability in Altera Devices," version 4. Altera Application Note 42. May 1999 available[online] http://www.altera.com/literature/an/an042.pdf.

[4] T. H.-Y. Meng, Synchronization Design for Digital Systems, Kluwer Academic Publishers, 1991.

[5] D. J. Kinniment, Synchronization and Arbitration in Digital Systems, Wiley, 2008.

[6] W. J. Dally and J. W. Poulton, Digital System Engineering, Cambridge University Press, 1998.

[7] C. Dike and E. Burton, ―Miller and noise effects in a synchronizing flip-flop,‖ IEEE Journal of Solid.

[8] S. Yang, M. Greenstreet, ―Computing synchronizer failure probabilities,‖ Proc. Design Automation and Test in Europe (DATE), pp. 1 – 6, 2007.

[9] R. Ginosar, ''Fourteen Ways to Fool Your Synchronizer,'' Proc. IEEE Int'l Symp. Asynchronous Circuits and Systems (ASYNC 03), IEEE CS Press, 2003, pp. 89-96.

[10] S. Beer, R. Dobkin and R. Ginosar, "Metastability measurements of several ASIC and FPGA synchronizers," Technical Report, EE Dept, Technion, Oct. 2009.

[11] Digital design principles and practices by John F. Wakerly.

[12] Timing Measurement by Semiat Ginosar .

[13] C. Dike and E. Burton, "Miller and Noise Effects in a Synchronizing Flip-Flop," IEEE Journal of Solid-State Circuits.

[14] J. Jex and C. Dike, "A fast resolving BiNMOS synchronizer for parallel processor interconnect,"-. IEEE Journal of Solid-State circuits.

[15] R. Ginosar and R. Kol, "Adaptive Synchronization".

[16] D.G. Messerschmitt, Synchronization, in T.H. Meng, Synchronization Design for Digital Systems, Kluwer Academic Publishers.

[17] D.J. Kinniment, A. Bystrov, and A.V. Yakovlev, "Synchronization Circuit Performance," IEEE.

