

# Design of a Self-Timed Data Synchronizer for Crossing Two Different Clock Domains

Hatem M. Zakaria  
Asst. Professor  
Electrical Engineering Department  
Benha Faculty of Engineering,  
Benha University

Rehab I. Nawar  
Demonstrator  
Electrical Engineering Department  
Benha Faculty of Engineering,  
Benha University

## ABSTRACT

This paper presents asynchronous switch between any two different local clock synchronous domains. The asynchronous switch will generate a slower clock from two local clock modules and moderate the high rated clock domain to slow down its clock frequency without stopping or pausing any clock of them throughout the data communication among them. The proposed design is implemented using the CMOS 45nm technology of STMicroelectronics. In this case, the delay time to change the clock is shown to be about 0.4ns. The proposed system is designed to use a small number of circuit elements. So that, the asynchronous switch has a noticeable improvement in terms of power consumption, throughput, and circuit area.

## Keywords

SOC, GALS, FIFO, PSTR

## 1. INTRODUCTION

Modern system on chip (SOC) designs faces an increasing integration densities and large chip design. Therefore, large chip designs are commonly partitioned into multiple clock domains. Each individual clock domain may be synchronous and often use asynchronous method to connect between these domains. This type of systems is commonly known as GALS (globally asynchronous locally synchronous). Synchronous clock domains based on programmable ring oscillator can be used to change each domain frequency by applying different configurations of set/reset for the C-Muller gates, [12].

GALS is designed to solve the challenges on designing over large system on chip [1]. GALS design style combines the advantage of the both synchronous and asynchronous operation [1]. The main issue is designing reliable GALS interfaces to eliminate the metastability logic domains. Earlier solutions were designed to improve throughput, reduce power consumption and reduce area of chip. More recent GALS solutions have some focus on reducing EMI, facilitating system integration and providing side channel security. The principle architecture of GALS system for dataflow structure is shown in figure (1). All GALS systems have common structure as illustrated in figure (1). There are three techniques used to transfer data safely between different synchronous blocks to handle metastability in GALS system, namely,

pausable-clock generators, FIFO buffers, and boundary synchronization.

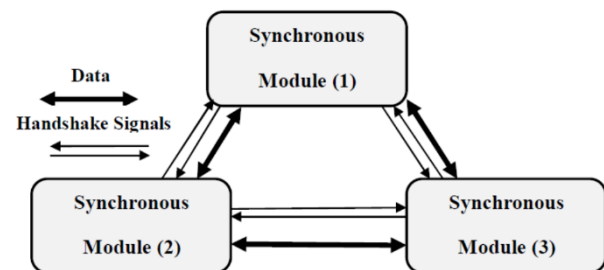


Figure (1): Principle architecture of GALS system.

### A. FIFO Solution

This solution handles the synchronization problem within FIFO buffers. Designer can use this solution to interconnect synchronous and asynchronous modules and also to construct synchronous-synchronous and asynchronous-asynchronous. This method can achieve an acceptable data throughput [2], [3], and [4]. The architecture of FIFO is shown in figure (2). Because of data cells (empty/full detector) in FIFO architecture the silicon area is costly. The advantage of FIFO is that they do not affect the locally synchronous domain's operation. Many FIFO based designs have been published recently [5], [6] and [7].

### B. GALS with Pausible Clocking

The main idea of asynchronous wrapper is to generate a stretch signal to stop the activity of both clock (receiver and transmitter). The general architecture of pausable clock generators is shown in figure (3). This method used to solve the problem of synchronization between the two clock domains [8], and [9]. This structure contains inputs and outputs between the locally synchronous modules.

### C. Boundary Synchronization

Third solution is to achieve very reliable data transfer between locally synchronous modules. This method does not affect the inner operation of the locally synchronous blocks during data synchronization at the borders of locally synchronous module [10], and [11]. Asynchronous switch has been designed by using this method of GALS system that is able to achieve many advantages.

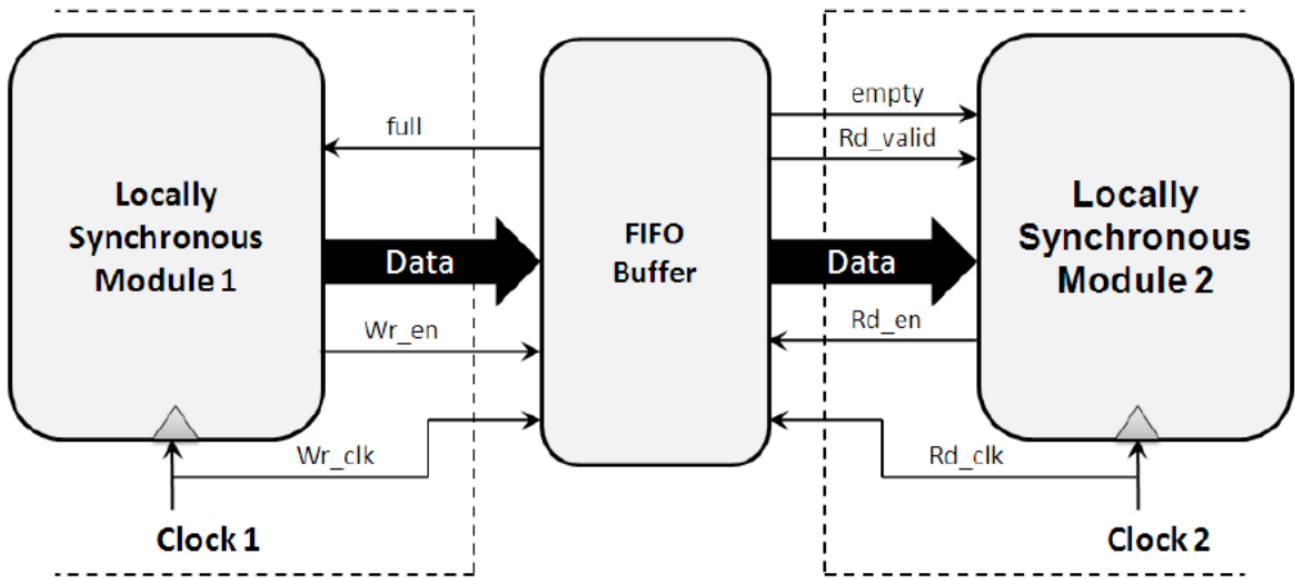


Figure (2): Typical FIFO Based GALS System.

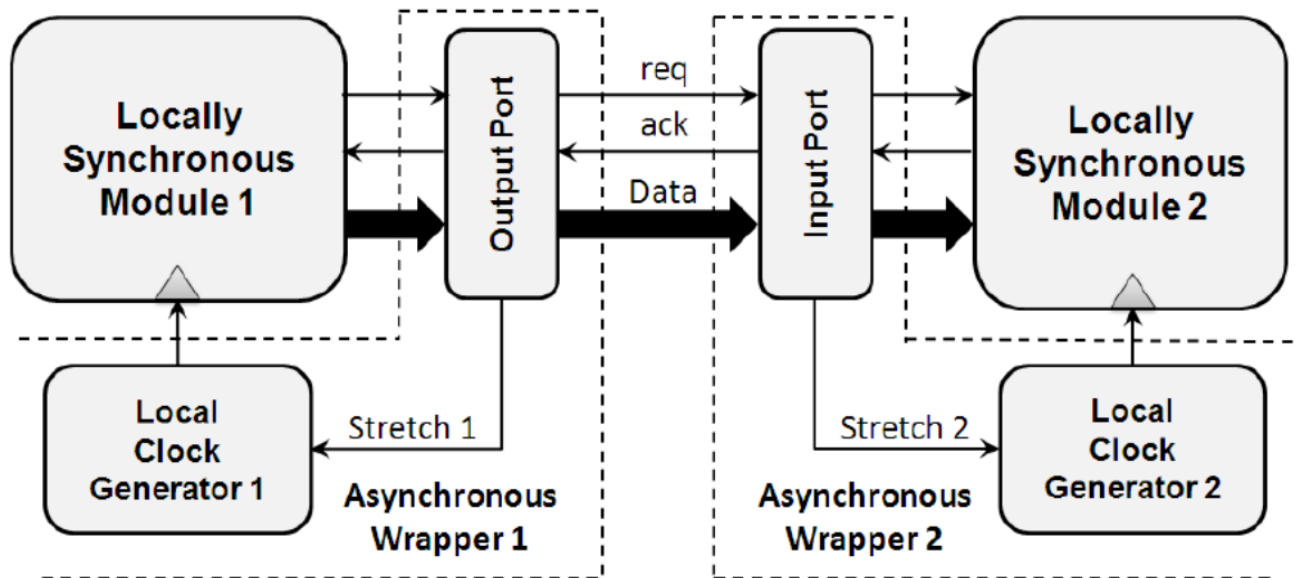


Figure (3): GALS System with Pausible Clocking.

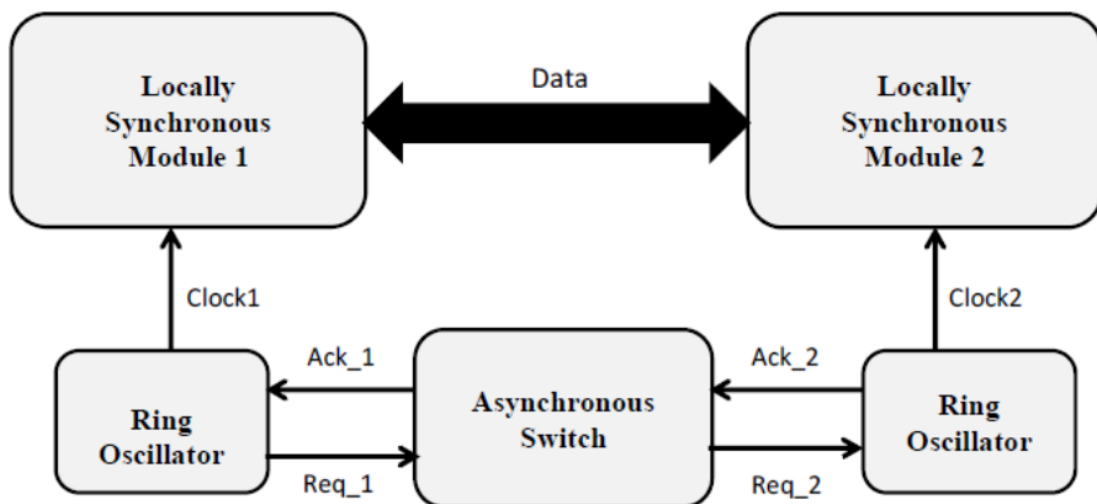


Figure (4): GALS System Based on Asynchronous Switch.

## 2. THEORY OF ASYNCHRONOUS SWITCH

In this section, a new method is presented for interfacing synchronizing different clock frequencies in a GALS system. The proposed GALS system uses the ring oscillator as the main generating source of local clock in each synchronous domain. This method is based on pausing the clock on one side and interfacing a handshake on the other side. The main advantage of this method is that it uses asynchronous circuit in securing data communication between different clocks and handling the problem of metastability. The significant in this system is that it avoids stopping data communication between different clock domains during their synchronization phase of data. In Figure (4) a simplified block diagram of transferring data based on GALS system is depicted. It uses ring oscillator and asynchronous switch to safely synchronize the data transfer by asynchronous handshaking acknowledgement (Ack) and request (Req) signals. The com signal is used in data exchange mode. The detailed interconnection between the ring oscillator and asynchronous switch are depicted in following section.

## 3. CIRCUIT DISCUSSION

### 3.1 Ring Oscillator (self-timed ring)

Self-timed ring is used to generate the clock frequency based on the number of states (C-Muller's) in the ring and the initialization of C Muller (i.e. set/reset configuration) as shown in figure (5) and this model is connected with figure(7). We can use PSTR (programmable self-timed ring) instead of self-timed ring [12]. In this case, we can control in the number of stages and generate a wide range of frequencies [12]. This modeling takes the Charlie effect into consideration to give identical steady state behavior and both are oscillating in evenly spaced mode. Moreover, Charlie effect has a correct behavior of the ring model. In this model a ring stage propagation delay is represented when taking into account the Charlie [13, and 14] and Drafting effects [15, 16, and 17]. The Charlie effect can be defined by the following phenomena "the closer the input event; the longer the propagation time" which appears in the input stage transistors. The Drafting can be defined by the following phenomena "the closer the successive transitions; the shorter the propagation time" which appears on the output stage capacitance. The analytical formulation of Charlie model is expressed by:

$$Charlie(s, y) = \underbrace{D_{mean} + \sqrt{D_{Charlie}^2 + (s - s_{min})^2}}_{Charlie} - \underbrace{\frac{y}{A}}_{Drafting} \quad (1)$$

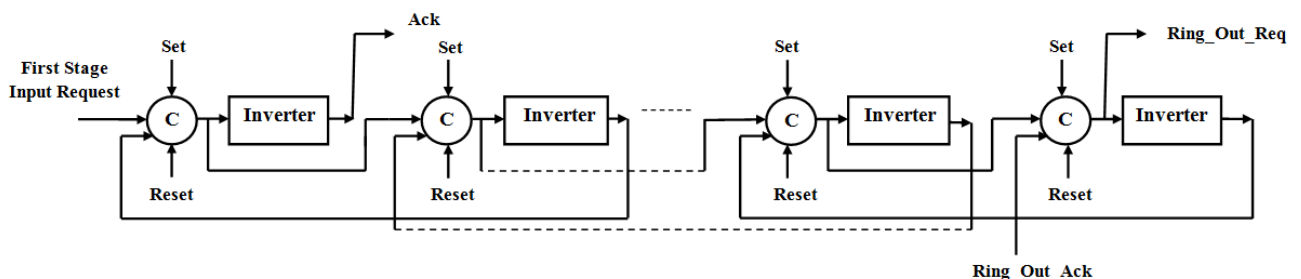


Figure (5): A Self-Timed Ring with Set/Reset.

$$D_{mean} = \frac{D_{rr} + D_{ff}}{2} \text{ and } s_{min} = \frac{D_{rr} - D_{ff}}{2}$$

- $s...$  the half separation time between inputs.
- $D_{ff}...$  the static forward propagation delay.
- $D_{rr}...$  the static reverse propagation delay.
- $D_{charlie}...$  the amplitude of the Charlie effect.
- $y...$  the time between the previous output commutation and the mean input time.
- $A ...$  the duration of the Drafting effect.
- $B ...$  the amplitude of the Drafting effect.

### 3.2 Asynchronous Switch

The asynchronous switch is designed to manage the connection of Ring-out-Req and Ring-out-Ack signals with the handshaking Req-ANOC and Ack-ANOC signals as depicted in figure (6). If com signal equals 1(i.e. data exchange), in this case the two communicating clock domains will be connected together through the asynchronous switch. The output from this switch is the slower clock frequency from the two clock domains. Once com signal equals 0(i.e. no data exchange) two ring oscillator return to work again with their own clock frequencies.

## 4. CIRCUIT DESIGN

The asynchronous switch is shown in figure (7). To facilitate testing of the switch operation, a normal clock has been applied in one side communicating module and ring oscillator in the other side as shown in figure (7). The inputs of the switch are two handshaking signals of the ring oscillator (Req and Ack) and normal clock frequency, the output of this switch are two handshaking signals to close the ring oscillator feedback to generate its corresponding clock frequency and the other output is anew clock signal. The new clock is generated only when com signal equals 1 (i.e. data exchange). To avoid any glitch or truncated clock period, the last clock cycle of communicating domain has already been completed before/after the start/stop of the data synchronization of two clock domains. As C-Muller elements are used as a state holding element, C2 will start to generate any output only when both inputs are equal to 1. When the last clock cycle of our communicating domains is completed, the data synchronization will only start. The inputs of C2 are two different domains (normal clock and output (Ack) of the ring oscillator). After that, C2 output will be the slower clock of our two communicating domains with minor delay according to the Charlie effect of C2.

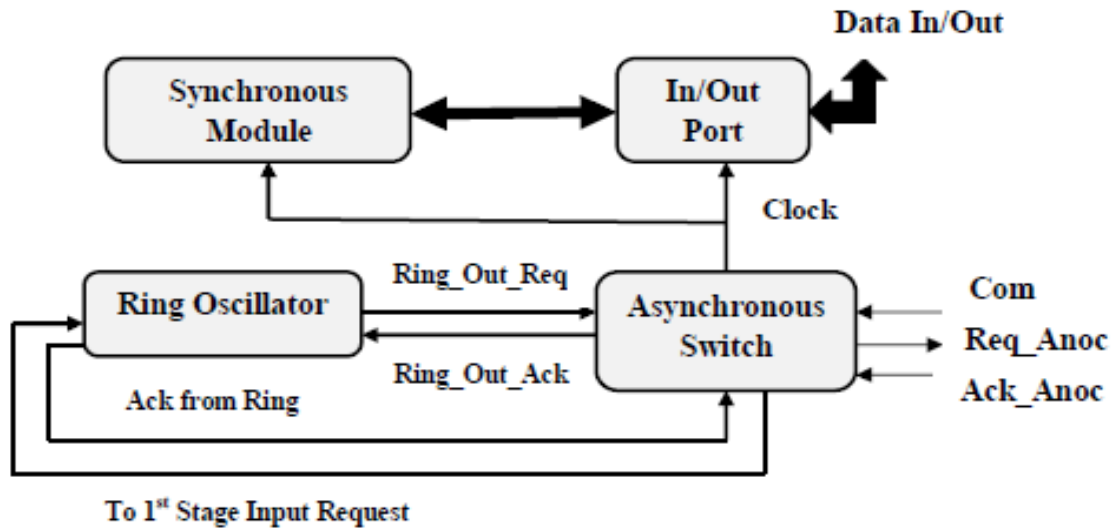


Figure (6): Asynchronous Switch Interconnections with the Ring Oscillator.

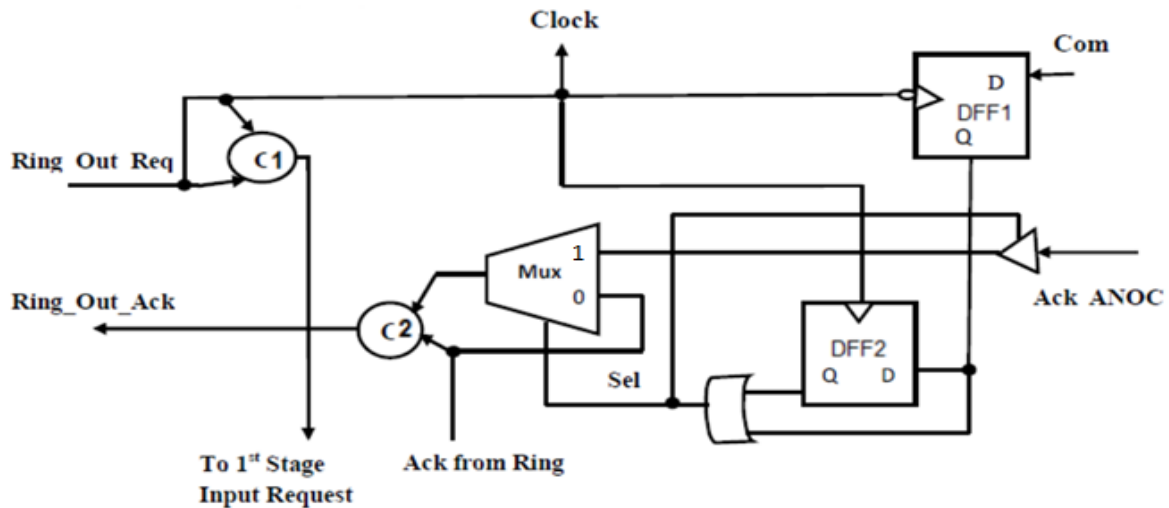


Figure (7): Asynchronous Switch.

This design has two D-FF's one of them is working at negative clock edge and another at positive clock edge to sure the clock cycle is completed. The inputs of OR gate are the outputs of two D-FF, the output of OR gate is sel signal (selection of MUX). If com signal changes its state from 0 to 1, the inputs of the OR gate are one and sel signal equals 1. Then, the MUX selects the normal clock connected with ring oscillator through C2 and the output of C2 is Ring\_Out\_Ack. The output of C1 is holding the Ring\_Out\_Req with the time of operating C2. As a result, the backward feedback connections (output of C1 and C2) are again closing the ring after completing the last clock cycle of the synchronized clock. Conversely, if com signal changes its state from 1 to 0, the outputs of two D-FF are 0. Thus, the output of OR gate is 0, which forces sel signal to be 0. So, the normal clock is disconnected from the MUX input and cuts the connection between the two communicating domains. Therefore, the first clock frequency of ring oscillator is return back.

## 5. SIMULATION RESULTS

The asynchronous switch shown in figure (7) is implemented on CMOS 45ns technology. Post layout simulation using timed VHDL model has been used to extract the delay information. Figure (10), shows an example where the asynchronous switch is requested to change its state from no

communication to data transfer with another synchronous module, then it changed its state to no communication again.

At point A, com signal changes its state from 0 to 1. As a result, the clock completes its last clock cycle; it starts to decide wither or not to continue at the same clock frequency according to the slower clock frequency of the two domains. In this state, the selection is equal to 1 after 1.22ns and the clock frequency from our ring oscillator is synchronized with ACK\_ANOC with small time delay (0.2ns) as shown in figure (8). The output of C2 (cout2) is connected to the ring oscillator which generate a new frequency, the delay between the inputs (ring\_out\_req) and output of C1 (cout1) equals 0.207ns as depicted in figure (9). The clock frequency from our ring oscillator is synchronized with Ack\_ANOC(normal clock) with small time delay (0.4ns) showing in figure (10). Therefore, the two domains are now well synchronized and the data transfer correctly cross the different GALS domains. At point B, com changes its state from 0 to 1. The sel signals changes to 0 and Ack\_ANOC (normal clock) disconnected from MUX. Thus, inputs of C2 come from the ring oscillator, in this state no communication between two synchronous modules. Sothat, the clock frequency returns back to its first clock as showing in figure (10) after completing its last clock cycle.

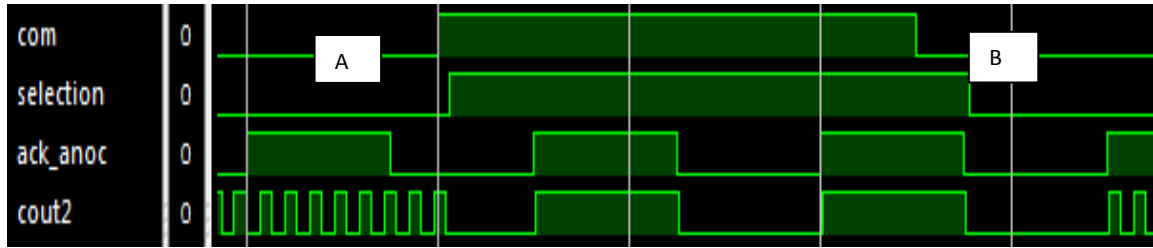


Figure (8): Timing Diagram of the output of Cmuller2.

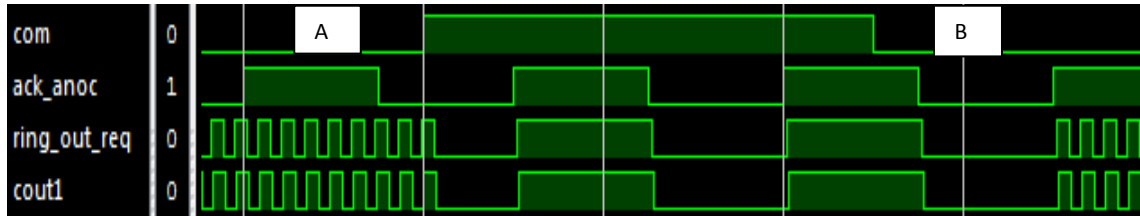


Figure (9): Timing Diagram of the output of Cmuller1.

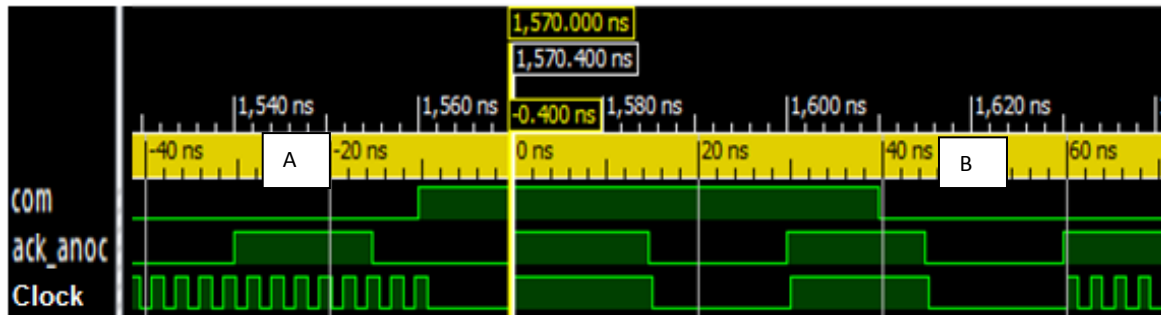


Figure (10): Timing Diagram of Asynchronous Switch.

Table (1) presents different frequencies between the locally synchronous modules and computes the delay between the input and the output of asynchronous switch. Clock1, Clock2 and New clock are showing in figure (11). When period of clock2 is smaller than clock1, the new clock is the same as that of clock1 without any delay, but there is a slight change in clock period as shown in table(1). On the other hand, when the clock1 is smaller than clock2, the new clock is the same as that of clock2 with 0.4ns delay as shown in table (1).

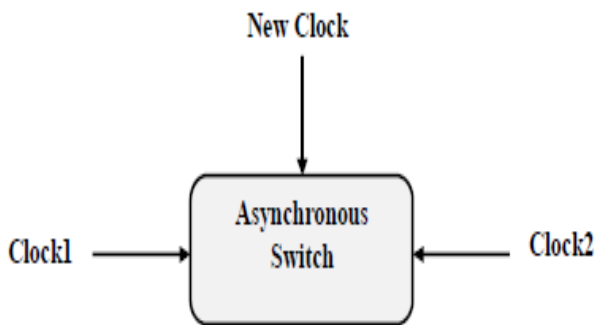


Figure (11): Block diagram of Asynchronous Switch.

Table (1): Test cases of different clock domains

Clock1	Clock2	New clock	Time delay
5.232ns	1ns	5.228ns	No delay
	2ns	5.378ns	No delay
	5ns	5.282ns	No delay
	6ns	6ns	0.4ns
	15ns	15ns	
	30ns	30ns	
1.676ns	60ns	60ns	0.4ns
	2.623ns	20ns	
	1.66ns		
	2.8ns		
	1.665ns		

## 6. CONCLUSIONS

In this design a point to point communication between the locally synchronous modules is implemented on CMOS 45nm technology and simulated using timed VHDL model (Xilinx ISE Design Suite 12.1). The area of this circuit is small due to using small number of elements (C-element, MUX, D-FF, OR, and buffer). So that, the asynchronous switch have low power consumption of GALS network and high throughput advantage of the FIFO-based GALS, as we are switching

directly (0.4ns only delay) from one frequency to the other without need to stop or reprogram the clock generator. In future work, asynchronous switch will be developed to communicate a point to multi point between different locally synchronous modules. The necessary delay to output the new clock signal will be considered.

## 7. REFERENCES

- [1] Chelcea T. and Nowick S., “Low-latency asynchronous FIFO's using token rings”, in Proceedings of the 6th International Symposium on Advanced Research in Asynchronous Circuits and Systems ASYNC '00, pp. 210-220, 2000.
- [2] Chakraborty A. and Greenstreet M., “Efficient self-timed interfaces for crossing clock domains”, in Proceedings of the 9th International Symposium on Asynchronous Circuits and Systems ASYNC '03, pp. 78-88, 2003.
- [3] Beigne E. and Vivet P., “Design of on-chip and off-chip interfaces for a GALS NoC architecture”, in Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems, ASYNC '06, pp.172-183, 2006.
- [4] Chelcea T. and Nowick S., “Robust Interfaces for Mixed-Timing Systems”, in IEEE Transactions on Very Large Scale Integration Systems, vol. 12, no. 8, pp 857-873, august 2004.
- [5] Sheibanyrad A. and Greiner A., “Two Efficient Synchronous-Asynchronous Converters Well-Suited for Network on Chip in GALS Architectures”, in Integration, the VLSI Journal, vol. 41, n° 1, pp 17-26, January, 2008.
- [6] Panades I. M. and Greiner A., “Bi-Synchronous FIFO for Synchronous Circuit Communication Well Suited for Network-on-Chip in GALS Architectures”, in Proceedings of the 1st International Symposium on Networks-on-Chip (NOCS'07), pp 83-92, May, 2007.
- [7] Muttersbach J., Villiger T. and Fichtner W., “Practical design of globally asynchronous locally-synchronous system”, in Proceedings of the International Symposium on Advanced Research in Asynchronous Circuits and Systems ASYNC'00, pp. 52-59, 2000.
- [8] Yun K. and Donohue R., “Pausible Clocking: A First Step toward Heterogeneous Systems”, In Proceedings of International Conference on Computer Design ICCD, pp. 118-123, 1996.
- [9] Ginosar R., “Fourteen ways to fool your synchronizer”, International Symposium on Asynchronous Circuits and Systems Async'03, pp. 1-8, 2003.
- [10] Dobkin R., Ginosar R. and Sotiriou C., “Data synchronization issues in GALS SoCs”, in Proceedings of the 10th International Symposium on Asynchronous Circuits and Systems ASYNC '04, pp 170-180, 2004.
- [11] Zakaria H., “Asynchronous Architecture for Power Efficiency and Yield Enhancement in the Decanometric Technologies: Application to a Multi-Core System-on-Chip”, PhD Thesis, Grenoble University, France, 2011.
- [12] Ebergen J. C., Fairbanks S. and Sutherland I. E., “Predicting performance of micropipelines using Charlie diagrams”, ASYNC'98, San Diego, CA, USA, IEEE, April 1998, pp. 238 - 246.
- [13] Zebilis V. and Sotiriou C. P., “Controlling event spacing in self-timed rings”, ASYNC'05, New York, USA, IEEE, March 2005, pp. 109 – 115.
- [14] Winstanley A. and Greenstreet M., “Temporal Properties of self-timed rings”, CHARM'01, London, UK, Springer-Verlag, April 2001, pp. 140 - 154.
- [15] Fairbanks S. and Moore S., “Analog micropipeline rings for high precision timing”, ASYNC'04, CRETE, Greece, IEEE, April 2004, pp. 41–50.
- [16] Winstanley A., Garivier A., and Greenstreet M., “An event spacing experiment”, in Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, ASYNC 02, pp. 47–56, 2002.