Notice of Violation of IEEE Publication Principles

"Delayed Latching for data synchronization in GALS SOC"
by Vivek E. Khetade, Dr. S. S. Limaye
in the 2011 4th International Conference on Emerging Trends in Engineering and Technology (ICETET), 2011, pp. 247 – 252

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

This paper contains significant portions of original text from the paper cited below. The original text was copied with insufficient attribution (including appropriate references to the original author(s) and/or paper title) and without permission.

Due to the nature of this violation, reasonable effort should be made to remove all past references to this paper, and future references should be made to the following article:

"High Rate Data Synchronization in GALS SoCs"
by Rostislav (Reuven) Dobkin, Ran Ginosar, and Christos P. Sotiriou
in the IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, Issue 10, 2006, pp. 1063 – 1074

2011 Fourth International Conference on Emerging Trends in Engineering & Technology

Delayed Latching for data synchronization in GALS SOC

Vivek E. khetade ,Member IEEE
Department of electronics engineering
Shri Ramdeobaba Kamla Nehru Engineering College,
Rashtrasant Tukdoji Maharaj Nagpur university
Nagpur, India
khetadev@rknec.edu

Dr. S. S. Limaye, Sr Member IEEE
Department of electronics engineering
Jhulelal Institute of Technology
Rashtrasant Tukdoji Maharaj Nagpur university
Nagpur ,India
shyam_limaye@hotmail.com

Abstract— **Globally asynchronous, locally synchronous (GALS) systems-on-chip (SoCs) may be prone to synchronization failure. This paper presents an in-depth analysis of the problem and proposes a novel solution. The problem is analyzed considering the cycle times of the GALS module, and the complexity of the asynchronous interface controllers using Petri Net graph (PN) approach .**

When high data bandwidth is not required, matched-delay asynchronous ports may be employed. A novel architecture for synchronizing inter-modular communications in GALS, based on delayed latching (DL), is described. DL synchronization does not require pausable clocking, is insensitive to clock tree delays, and supports high data rates. It replaces complex global timing constraints with simpler localized ones. Decoupled input port and Decoupled output port for Delayed Latching are presented. The risk of metastability in the synchronizer is analyzed in a technology-independent manner. Here we present the Petri net models of the Globally Asynchronous and Locally Synchronous (GALS) architectures for speed independent (SI). The models are feed into Petrify to produce logic equations for gate level implementation of asynchronous circuit. The circuit is simulated on VCS and synthesized on Design compiler of Synopsys EDA tool .

Index Terms—**Asynchronous circuits, globally asynchronous, locally synchronous (GALS), synchronization, system-on-chip (SoC).**

I. INTRODUCTION

With the rapid progress of silicon technology, systems-on-chip (SoCs) incorporate an increasing number of modules of growing sizes, operating at faster clock frequencies. These developments make it even more difficult to distribute a single synchronous clock to the entire chip [1]. As an alternative, different methods for providing each module with its own clock are being developed. Another motivation for independently clocking different modules, is to reduce power consumption by means of dynamic voltage and frequency scaling (DVFS) [2]–[4]. When the clock frequencies of the various modules are uncorrelated with each other, and when they can change over time independent of the clocks of other modules, the resulting SoC is termed a globally asynchronous, locally synchronous (GALS) system [5], [6]. Each GALS module (a "Locally Synchronous Island") can be enclosed in an asynchronous wrapper, which facilitates inter-modular communications and generates the clock for the module [7]–[14].

Data synchronization and communication across clock domains in GALS architectures constitute a major challenge. Simple "two-flop" synchronizer typically incurs significant multicycle latency and limits the throughput. An alternative solution is provided by elastic first-in, first-out (FIFO) buffers. The most promising approach employs stoppable stretchable clocks for the GALS modules: Port controllers can pause the local clock when sampling asynchronous input. By stopping the local GALS clock during data transfers across clock domains, the possibility of metastability is eliminated [7]–[14].

The main contribution in this paper is that we describe a novel synchronization technique for GALS SoC, delayed latching (DL). It does not require pausable clocking, it is insensitive to clock tree delays, and it provides high data rate synchronization. Detailed reliability analysis for DL is presented. We present possible GALS wrappers based on DL.

The paper begins with a survey of related research, in Section II. In Section III proposed interface scheme for the delayed latching is given along with reliability and performance analysis .The architecture for Decoupled input port along with their controller modeling in Petri Net, simulation is described in Section IV. The architecture of decoupled output port along with controller modeling in Petri Net is given in Section V. Simulation result and conclusion and future work is described in section VI and VII along with appendix for the component detail such as Muller-C and MUTEX.

II RELATED WORK

Two principal clocking and synchronization methods have been proposed for solution of the data synchronization problem. Clock synchronization employs handshake clocks that are stopped based on inputs from other domains [7]. Stoppable local clocks have been proposed in [8]–[13]. According to that methodology, a local ring-oscillator clock generator in each synchronous "island" incorporates a set of mutual exclusion elements (MUTEX) [16] that stop the clock temporarily when new input data arrives, so as to avoid the risk of metastability.

Stoppable clocks have been introduced for GALS system in [12] and [13]. Asynchronous inter-modular communication is decoupled from the stoppable clock interface of the synchronous modules. A modified two-phase asynchronous interface wrapper for communication between

978-0-7695-4561-5/11 $26.00 © 2011 IEEE
DOI 10.1109/ICETET.2011.13

two locally synchronous modules is presented in [8]. The authors also propose FIFO buffering for performance enhancement. A four-phase version of the asynchronous GALS wrapper, which handles multiple ports and also facilitates testing, is presented in [9]. A number of GALS interconnect structures and modified wrappers are analyzed in [10], focusing on ring topology and packet based communications. An architecture for combining synchronous and asynchronous modules in a GALS system is presented in [11], employing handshake based on matched delays. Finite-state machine (FSM)-based demand and poll port controllers are also presented.

III PROPOSED ARCHITECTURE OF DELAYED LATCHING

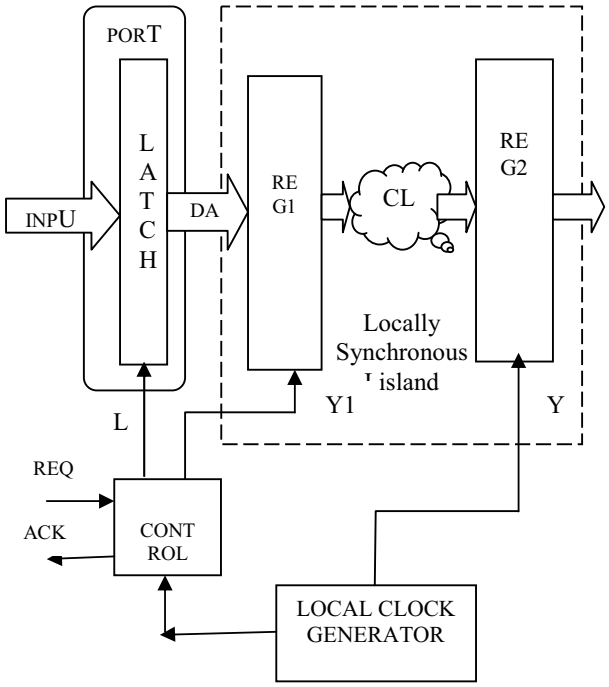


Fig 3.1. Architecture of Delayed Latching

In an Delayed Latching input port synchronizer, the asynchronous controller Fig. 3.1 controls both the input latch 'L' and Y1, the clock input to the first sampling register. Signal Y, the local clock of the module, is uninterrupted. In addition, the port issues a valid indication for each newly received data word and prevents WRITE after READ (WAR) hazards. Various modes of the DL operation are demonstrated in Fig. 3.2.

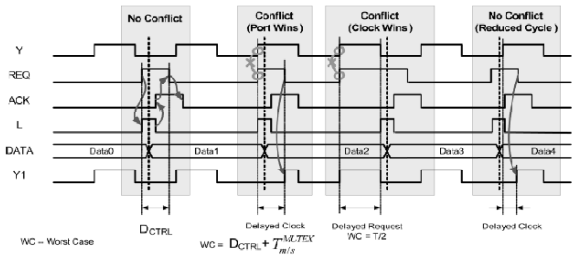


Fig 3.2 Various modes of operation

In DL, the clock of the Locally Synchronous Island is never stopped. The only measure available is to delay Y1+ when a conflict is imminent. Y- is unaffected, and only the high phase is shortened. A port request is accepted only during the low-phase of Y, latching the incoming data (L+) and delaying Y1 when needed. The conflicts between Y+ and REQ + are resolved by a MUTEX inside the control.

DL is unaffected by clock cycle changes that can be caused for instance due to dynamic frequency or voltage scaling [2]–[4]. There is also no restriction on stopping the clock during periods of inactivity.

3.1 DL Synchronization Reliability and Performance

The worst case operation occurs when at the conflict between REQ+ and Y+, REQ+ wins. In this case the high-phase of Y1+ is maximally shortened, shown in Fig. 3.2, in the "Port Wins" case. The shorter cycle leaves less time for computing in the combinational logic immediately following the first register. The implementation must assure that the remaining high phase is long enough, according to the restrictions on the minimal high phase width for FFs or registers of the target library. The high phase of the clock is shortened by an amount equal to the latency of the asynchronous control DCTRL and the MUTEX resolution latency.

We require that the SoC be at least M -safe, where a selected value for M could be 100 years (other values may also be used). To achieve that, the safety of each synchronizer in a SoC with about K = 100 synchronizers must be at least K times larger, namely M= 10 000 years [25]. We note that, in a standard SoC (a digital IC based on standard cells and designed using standard EDA tools) the shortest clock cycle is typically about 100–160 FO4 inverter delays [26]. The nominal FO4 inverter delay depends mostly on the process technology. Thus, the fastest high phase (50% of the clock cycle) is about 50 inverter delays long. In order to assess the worst-case MTBF in the following equation, we assume that τ and 'W' are one and two FO4 inverter delays, respectively, [24], $F_D = F_C$ (worst-case analysis), the clock cycle $T=100\tau$ and, thus, $F_C = F_D = 1/100\tau$. We can determine the required metastability resolution time in terms of a number of gate delays, by solving

$$\mathrm{MTBF} = \frac{e^{T_{m/s}^{\mathrm{MUTEX}}/\tau}}{W \cdot F_C \cdot F_D} = \frac{e^N}{2 \times \frac{1}{100} \times \frac{1}{100}} \times \tau \geqslant 10^4 \quad \text{year.}$$

For $10^{-11} < \tau < 10^{-10}$ s (the range of FO4 gate delays in present and foreseeable technologies,) (15) For $T = 100\tau$, this implies that at least one half of a symmetric clock cycle should be allowed for resolution. For slower SoCs, e.g., where the fastest clock cycle is 160τ [26], a quarter clock cycle suffices to achieve this MTBF. For most aggressive designs [such as high-speed processors or high-speed application- specified integrated circuit (ASIC) modules], where $10\tau < T < 50\tau$ a different approach based on multicycle resolution time or on multisynchronous clocking [27] is required.

3.2 DL Constraints

We require that to guarantee minimal high clock phase

$$\frac{T}{2} - D_{\mathrm{CTRL}} - T_{m/s}^{\mathrm{MUTEX}} > T_{\mathrm{HP}}^{\mathrm{Min}}.$$

Leading to a constraint on the asynchronous control delay

$$D_{\mathrm{CTRL}} < \frac{T}{2} - T_{\mathrm{HP}}^{\mathrm{Min}} - T_{m/s}^{\mathrm{MUTEX}}. \quad (1)$$

Another constraint applies to D_L, the delay of the combinational logic that follows REG1. When the rising edge of Y1 is delayed, the effective computation time in that logic stage becomes shorter than the clock cycle. Therefore, the following should be satisfied.

$$D_L < T - D_{\mathrm{CTRL}} - T_{m/s}^{\mathrm{MUTEX}}. \quad (2)$$

D_{CTRL} contains additional buffering delays when wide data path is required.

IV MODELING OF DECOUPLED INPUT PORT

Fig.4.1 shows a possible implementation of Fig. 3.1. Without a conflict, Y1+ is either not delayed or delayed by less than D_{CTRL} .R2'+ is granted only during the low-phase of Y . The MUTEX arbitrates any conflict between R2'+ and Y+ . When R2'+ wins over Y+ , the asynchronous controller is granted (R3+) . The controller employs an asymmetric matched delay Do --→Di to open the latch and then close it again (L+ -→ L-). The asymmetric delay applies a longer delay to one of the two edges. In our case, the delay is minimal for the falling edge and is longer for the rising one. The long delay matches the latch propagation delay for all corners and delay variations. After R2'- ,Y1+ triggers REG1, leading to a shortened cycle in the combinational logic following REG1 (the cycle is shortened by D_{CTRL}

If the clock wins over R2'+, R3+ happens only half a cycle later, after Y- .

The PN of the decoupled input port control is shown in Fig. 4.1. The controller delay is measured along the dashed path. The path is contained entirely inside the input port, ensuring that any reduction of the clock cycle depends solely on the input port control logic (and not on the logic and clock of the transmitter module). The MUTEX output Y1 should be buffered with a low-skew net when wide data path is required. In this case, the additional latency must be taken into account.

Generation of the valid signal is performed as follows. For each granted data transfer the control issues signal that latches the data and concurrently sets valid to logic high, indicating a new ready data word inside the data latch. Once latching is accomplished, the controller deasserts R2 (R2-) and along with valid + releases the MUTEX. The next incoming data transfer request (R2+) is blocked (by means of the c-element) until the data is sampled by REG1. Valid is released with the next rising edge of Y1, also enabling arbitration of the next incoming data transfer request. The SR latch of the valid signal is free of conflicts: L and Y1 are guaranteed to be mutually exclusive thanks to the MUTEX.

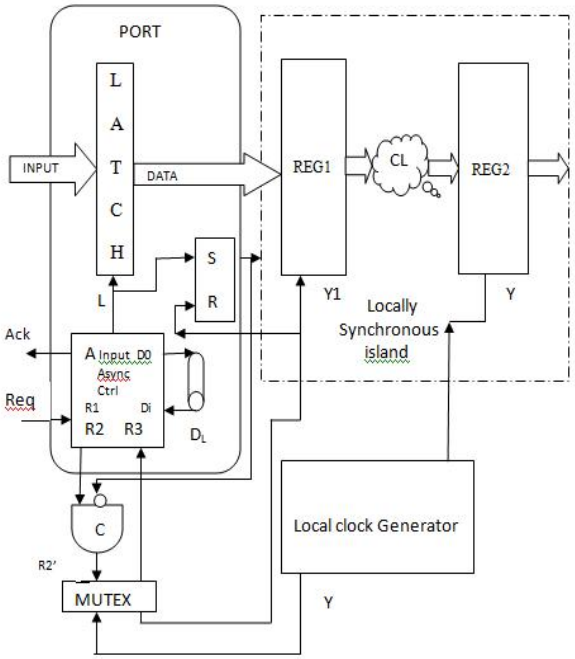


Fig 4.1 GALS module decoupled input port

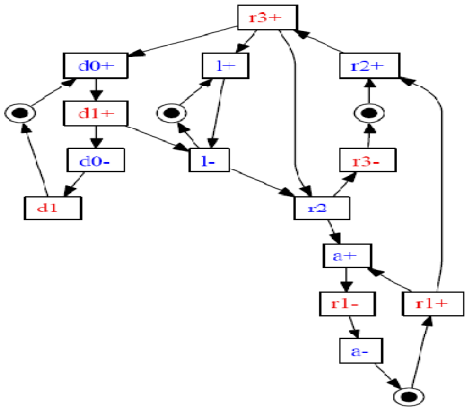


Fig4.2 Petri Net model of decoupled input port controller

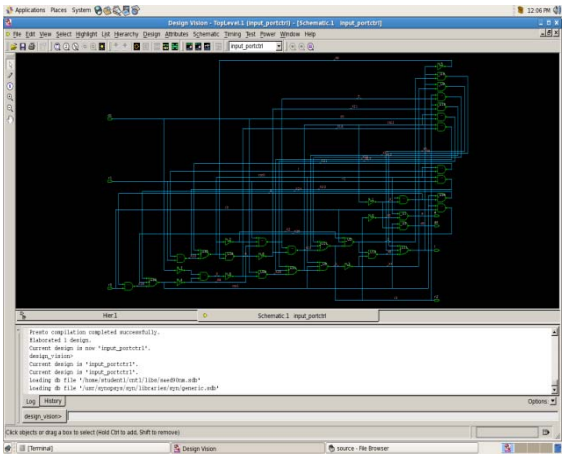


Fig4.3 Schematic of input port controller from design compiler

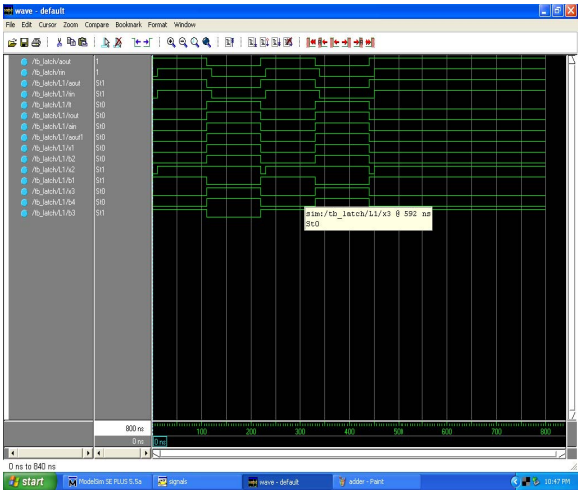


Fig 4.4 Simulation result of Decoupled input port controller

V MODELING OF DECOUPLED OUTPUT PORT

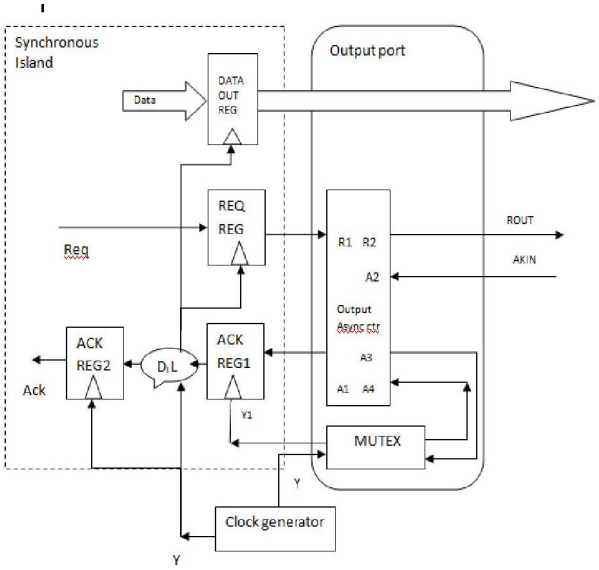
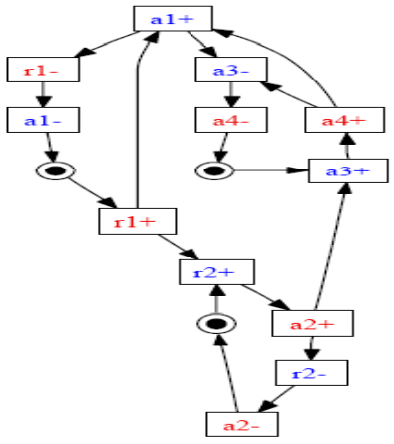


Fig5.1 GALS module decoupled output port

Decoupled output ports of GALS modules are subject to the same issues as the input ports which were discussed in the previous sections. The difference is that with an output port it is the incoming ACK signal which must be synchronized. An output port circuit is shown in Fig. 5.1 and the PN of its control is shown in Fig.5.2. The internal acknowledge (A1) is decoupled from the external asynchronous handshake.

The controller latency of the decoupled output port control, , should be verified according to (1) and (2).

$$D_{\mathrm{CTRL}}^{\mathrm{OUT}} = D[A4+ \rightarrow A1+ \rightarrow A3-].$$



INPUTS: r1,a2,a4
OUTPUTS: r2,a3,a1

Fig 5.2 Petri Net model of decoupled output port Controller

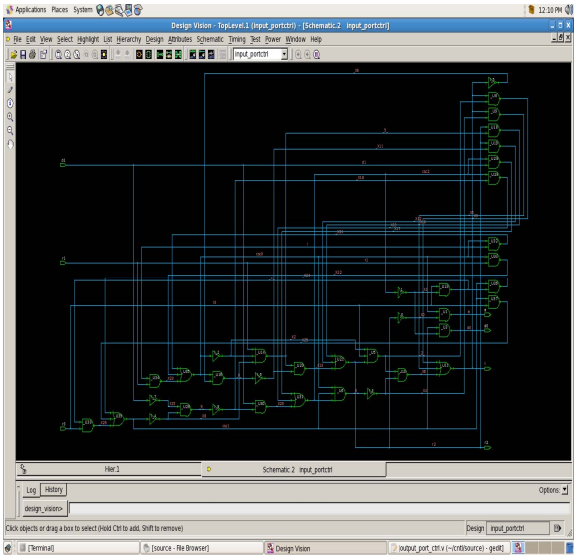


Fig5.3 schematic of input port controller from design compiler

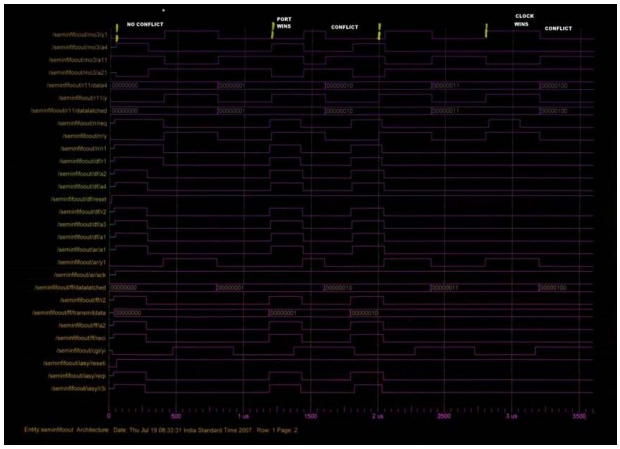


Fig5.4 Simulation result of decoupled output port controller

VI. SIMULATION

The circuits of section IV and V were synthesized for an asynchronous controller part using Petrify [30], converted to Verilog , synthesized by Synopsys design compiler CMOS libraries [31]and [32] and by gate level simulation with wire load model on Synopsys VCS tools.

The simulation and synthesis result of Petri Net is presented in Fig 4.4 and Fig 5.4 for input port controller and output port controller respectively .

To preserve timing correctness, careful layout should be performed. The sampling latch, the first register REG1, the asynchronous control, and the MUTEX must be placed closely together in order to avoid the impact of wire propagation delay on the critical path. These requirements are expected to be met easily, since the wrapper contains only a single port and is not connected to any other parts of the module.

The overhead of the DL controller is expected to be less than 100 gates. For example, the decoupled input port controller logic complexity is equivalent to 36 2-input NAND gates. For a typical SoC module of 100K gates, the DL controller overhead is only 0.1%. Another 0.1% overhead may be incurred by the latches of the input port.

VII. CONCLUSION AND FUTUREWORK

We have addressed the problem of synchronization failures due to clock delays in locally generated, arbitrated clocks of GALS SoCs. The problem has been analyzed based on clock delays, cycle time, and complexity of the asynchronous port controllers. The analysis employs a timed STG approach in order to identify potential conflicts spanning asynchronous and synchronous circuit.

A novel architecture for synchronizing inter-modular communications in GALS, based on DL, has been presented. DL synchronization does not require pausable clocking, is insensitive to clock tree delays, and supports high data rates. It replaces the complex global timing constraints on clock delays by simpler, more localized ones. Three different DL ports have been described, two for input and one for output. Their operation has been demonstrated and analyzed by simulations. We also present a technology-independent analysis of the metastability risk in the synchronizer, and its effect on the synchronizer architecture

VIII Appendix A

Muller C element

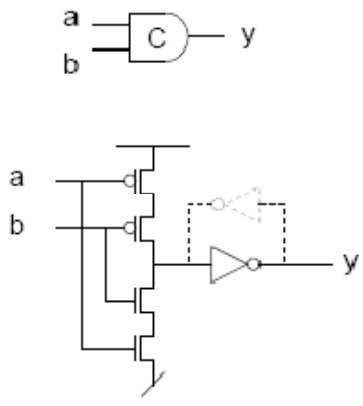


Fig 8.1 Muller-C element

It is a state-holding element much like an asynchronous set-reset latch. When both inputs are 0 the output is set to 0, and when both inputs are 1 the output is set to 1. For other input combinations the output does not change. Consequently, an observer seeing the output change from 0 to 1 may conclude that *both* inputs are now at 1; and similarly, an observer seeing the

output change from 1 to 0 may conclude that *both* inputs are now 0.

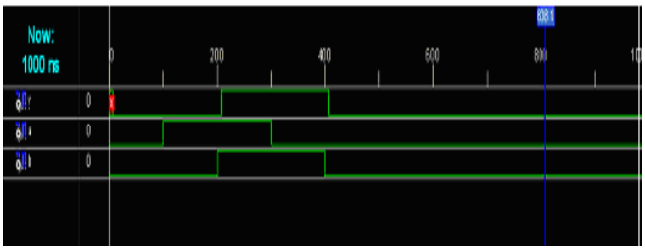


Fig 8.2 simulation of Muller C Element

MUTEX

The rising edge that appears first at the input of the MUTEX will go on to pass through the MUTEX to its respective output, and the rising edge on the other signal will pass through only after the first signal has gone low. If both signals appear at the inputs of the MUTEX at the same time, it utilizes a 'coin-tossing' strategy to let just one of the rising edges to pass through. Fig 6.3 shows the internal structure of the MUTEX and fig 6.4 shows simulation result for mutex

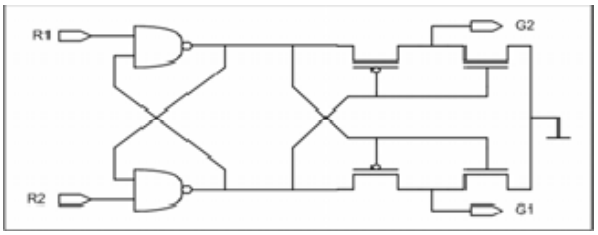


Fig 8.3 internal structure of the MUTEX

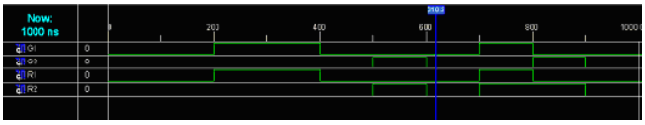


Fig 8.4 simulation result for mutex

REFERENCES

[1] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proc. IEEE*, vol. 89, no. 5, pp. 665–692, May 2001.

[2] G. Semeraro, D. H. Albonesi, S. G. Dropsho, G. Magklis, S. Dwarkadas, and M. L. Scott, "Dynamic frequency and voltage control for a multiple clock domain microarchitecture," in *Proc. IEEE/ACM Int. Symp. Microarch.*, 2002, pp. 356–367.

[3] L. S. Nielsen, C. Niessen, J. Sparsø, and C. H. van Berkel, "Low-poweroperation using self-timed and adaptive scaling of the supply voltage," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 4, pp.391–397, Dec. 1994.

[4] W. R. Daasch, C. H. Lim, and G. Cai, "Design of VLSI CMOS circuits under thermal constraint," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 49, no. 8, pp. 589–593, Aug. 2002.

[5] D. M. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D. dissertation, Comput. Sci. Dept., Stanford Univ., Palo Alto, CA, 1984

[6] D. Bormann and P. Cheung, "Asynchronous wrapper for heterogeneous systems," in *Proc. ICCD*, 1997, pp. 307–314.

[7] J. Kessels, A. Peeters, P. Wielage, and S. J. Kim, "Clock synchronization through handshake signaling," in *Proc. ASYNC*, 2002, pp. 59–68.

[8] S. Moore, G. Taylor, R. Mullins, and P. Robinson, "Point to point GALS interconnect," in *Proc. ASYNC*, 2002, pp. 69–75.

[9] S. Oetiker, F. K. Gürkaynak, T. Villiger, H. Kaeslin, N. Felber, and W.Fichtner, "Design flow for a 3-million transistor GALS test chip," in*Proc. ACiD Workshop*, 2003, ACiD workshop slides.

[10] T. Villiger, H. Kaeslin, F. K. Gürkaynak, S. Oetiker, and W. Fichtner,"Self-timed ring for globally-asynchronous locally-synchronous systems,"in *Proc. ASYNC*, 2003, pp. 141–150.

[11] J. Muttersbach, T. Villiger, and W. Fichtner, "Practical design of globally-asynchronous locally-synchronous systems," in *Proc. ASYNC*, 2000, pp. 52–61.

[12] K. Y. Yun and R. P. Donohue, "Pausible clocking: A first step toward heterogeneous systems," in *Proc. Comput. Des.: VLSI Comput.Process., ICCD*, 1996, pp. 118–123.

[13] K. Y. Yun and R. P. Donohue, "Pausible clocking-based heterogeneous systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 7, no. 4, pp. 482–488, Dec. 1999.

[14] A. E. Sjogren and C. J. Myers, "Interfacing synchronous and asynchronous modules within a high-speed pipeline," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 5, pp. 573–583, Oct. 2000.

[15] Arnab Banerjee , Pascal T.Wolkotte,Robert D. Mullins and Simon W Moore , "An Enrgy and performance Exploration of Network-on-chips Architecture ," in IEEE Transaction on VLSI ,vol-17,no 3, March 2009 pp. 319–329.

[16] C. L. Seitz, "System Timing," in *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980, ch. 7.

[17] T. Chelcea 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.

[18] A. Chakraborty and M. R. Greenstreet, "Efficient self-timed interfaces for crossing clock domains," in *Proc. ASYNC*, 2003, pp. 78–88.

[19] A. Chakraborty and M. R. Greenstreet, "A minimal source-synchronous interface," in *Proc. ASIC/SOC Conf.*, Sep. 2002, pp. 443–447.

[20] S. Chakraborty, J. Mekie, and D. K. Sharma, "Reasoning about synchronization techniques in GALS systems: A unified approach," in *Proc. Workshop Formal Methods GALS Arch. (FMGALS)*, 2003, invited paper.

[21] T. A. Chu, C. K. C. Leung, and T. S.Wanuga, "A design methodology for concurrent VLSI systems," in *Proc. ICCD*, 1985, pp. 407–410.

[22] J. Mekie, S. Chakraborty, and D. K. Sharma, "Evaluation of pausible clocking for interfacing high speed IP cores in GALS framework," in *Proc. VLSI Des.*, 2004, pp. 559–564.

[23] S. W. Moore, G. S. Taylor, P. A. Cunningham, R. D. Mullins, and P. Robinson, "Self-calibrating clocks for globally asynchronous locally synchronous systems," in *Proc. Int. Conf. Comput. Des. (ICCD)*, 2000, pp. 73–78.

[24] C. Dike and E. Burton, "Miller and noise effects in a synchronizing flip-flop," *IEEE J. Solid-State Circuits*, vol. 34, no. 6, pp. 849–855, Jun. 1999.

[25] R. Ginosar, MTBF of multi-synchronizer SoC [Online]. Available: http://www.ee.technion.ac.il/~ran/papers/MTBFmultiSyncSoc.pdf

[26] *"International Technology Roadmap for Semiconductors (ITRS),"* (2003) [Online]. Available: www.itrs.net

[27] Y. Semiat and R. Ginosar, "Timing measurements of synchronization circuits," in *Proc. ASYNC*, 2003, pp. 68–77.

[28] R. Ho, K. W. Mai, and M. A. Horowitz, "The future of wires," *Proc. IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.

[29] N. Weste and D. Harris, *CMOS VLSI Design—A Circuits and Systems Perspective*. Reading, MA: Addison-Wesley.

[30] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev, "Petrify: A tool for manipulating concurrent specifications and synthesis of asynchronous controllers," *IEICE Trans. Inf. Syst.*, vol. E80-D, no. 3, pp. 315–325, 1997

[31] J. B. Sulistyo, J. Perry, and D. S. Ha, "Developing standard cells for TSMC 0.25 _m technology under MOSIS DEEP rules," Dept. Elect. Comput. Eng., Virginia Tech., Blacksburg, Tech. Rep. VISC-2003-01, Nov. 2003.