

A Mesochronous Technique for Communication in Network on Chips

Mohsen Saneei

Nanoelectronics Center of
Excellence, School of Electrical and
Computer Engineering,
University of Tehran, Tehran, Iran
Email: mohsen.saneei@ece.ut.ac.ir

Ali Afzali-Kusha

Nanoelectronics Center of
Excellence, School of Electrical
and Computer Engineering,
University of Tehran, Tehran, Iran
Email: afzali@ut.ac.ir

Zainalabedin Navabi

Computer Aided Design Lab.,
Elec. and Comp. Eng. Dept.
University of Tehran, Tehran,
Iran
Email: navabi@cad.ece.ut.ac.ir

Abstract: In this paper, we propose a mesochronous scheme for communication over serial buses in network on chips (NoC). The technique, which removes metastability errors in mesochronous communications, makes use of only one strobe line with the bus. The strobe line toggles once with every frame of the data. In the suggested method, the frequencies of the transmitter and the receiver are independent with some tolerable difference. The results of HSPICE simulations in a 0.13 μ m standard CMOS technology show a 3.65 Gbps as the maximum transmission bandwidth of the technique. The idea can be applied to parallel buses without any change in the control circuit.

Index Terms—Network on Chip, Mesochronous, Serial Interconnect, Interconnect Parameter, Metastability.

I. INTRODUCTION

While gate delays scale down in nano-technology systems, the delays of the global wires typically increase or remain constant if the repeaters are used. As the computation energy costs in these systems are decreasing, the energy consumption of the on-chip and the inter-chip communications is increasing. It was estimated that in a 50 nm technology, the projected chip die edge will be around 22 mm with a clock frequency of 10 GHz and a global wire delay can be up to 6–10 clock cycles [1]-[3]. The major problems caused by the interconnection wires in these technologies include the wiring complexity, the cross talk, the global synchronization difficulty, the scalability problems, and the transmission bandwidth limitation [2]-[6]. As a result, recently On-Chip Networks (OCN) have been actively studied to reduce the wire complexity in the point-to-point interconnection and solve the problems of scalability caused by widely popular used bus-based SoC communication [2][3][4][6]. An alternative scenario consists of local synchronous cores which each communicate with the other through a network-centric architecture with asynchronous or mesochronous communication. These systems are called NoCs. In these systems both parallel and serial interconnection may be used. The serial communication has the advantage of much lower wiring complexity to make the implementation of OCN possible and practical [5][6]. The bandwidth of OCN transmission will be limited by the serial communication clock frequency. Consequently, high-speed designs become the

critical demand in the on-chip serial communication architecture [6].

One of synchronization paradigms for future chips is globally- asynchronous locally synchronous (GALS) with many different clocks [4][7]. In this scheme, every core is a synchronous block which has its own local clock. The communication protocol between cores is asynchronous which makes use of *request* and *acknowledge* signals for the handshaking. In this scheme, the global wires will span multiple clock domains, and synchronization failures in communicating between different clock domains will be rare but unavoidable events [4]. Therefore, although synchronous blocks are fast but the communication between the cores are slow due to the delay of the wires and the overhead of the request and acknowledge signals.

A type of the GALS methodology is called mesochronous clocking technique. This method of asynchronous communication reduces the handshaking overhead to just a strobe signal. In the nano-technology era where the computational requirements of new applications of SoCs (such as multimedia) and their on-chip communication needs renders mesochronous clocking as a better solution as a more reliable and robust methodologies [8]-[11]. The mesochronous scheme is typically used for the communication between two cores that have the same frequency with arbitrary phases [9]. The problem with this communication protocol is metastability which happens if the sampling edge of the clock occurs when the input data is changing [9]. Some research efforts have been suggested to overcome this problem. In [9], a scheme called STSS was proposed which uses mesochronous clocking in each PE. For an error-free parallel data transfer, the technique automatically selects the proper clock edge for sampling the data. The edge selection is based on the detection of the metastability. In [10], a technique which is called globally updated mesochronous design style (GUM) was suggested. The idea of the GUM design style is to split the VLSI system into blocks, add clock delay circuitry for calibration to each block, design them separately, and finally integrate them into one or several physical chips. Prior to the full operation, a short calibration phase is carried out within the application to establish the block synchronization [10]. In mesochronous clocking scheme proposed in [11], a strobe signal is distributed along each link where at each edge of the strobe signal, a new data is sent on the link. Therefore, if the

strobe is connected at one of the incoming ports of each block, it can be used as the local clock for that block [11]. The strobe signal can replace the clock distribution network.

In all of the above schemes, the transmitter and the receiver have the same frequency but with arbitrary phases. In [5], a serial communication scheme that the frequencies of the transmitter and receiver can have minor differences was proposed. In this scheme, the transmitter and the receiver have separate ring oscillators which are controlled by a strobe pulse. In this method, the transmitter sends a strobe pulse with every frame of the data [5]. The scheme in [6] is similar to [5] but with this difference that, instead of the counter, a shift-register controls the ring oscillator. This change has improved the speed of the transmitter and the receiver [6]. In both of these two techniques, the frequency of the external clock (strobe) which is sent from the transmitter to the receiver is equal to the number of data frame per second.

In this work, we introduce a mesochronous scheme in which the frequency of the strobe signal sent from the transmitter to the receiver is equal to the half of the number of data frames per second. We describe the scheme in Section II while Section III discusses the range of tolerable difference between the receiver and the transmitter frequencies. We explain the procedure of extending the scheme to the parallel buses in Section IV. The results are discussed in Section V with the summary and conclusion presented in Section VI.

II. PROPOSED SERIAL INTERFACE

In this section, we describe the proposed scheme for interfacing two cores in the serial communication. There are only two signals between each two cores: data and strobe. Every frame of data has n bit which are transferred serially on the data line. The strobe signal toggles one time for every frame of data.

In the transmitter, a ring oscillator is used to generate the serial transfer clock (CLK_t) to synchronize the serial transmission of the data. This ring oscillator starts its oscillation after rising or falling edge of strobe and generates transmitter clock (CLK_t). Then transmitter shift register (parallel to serial converter) shifted out with CLK_t and send data on the line. Therefore, the edge of the first bit of data in the transmitter is occurred after edge of the strobe.

Since the lengths of the data and the strobe lines are approximately the same, the data and the strobe signals have the same delay on the links between the two cores. In the receiver, the strobe signal activates the enable signal of the receiver ring oscillator that generates the receiver clock (CLK_r). If the receiver clock generator is the same as the transmitter clock generator, the edge of the first bit of data in the receiver and the rising edge of CLK_r occur simultaneously but the falling edges of CLK_r occur at the middle of the data bit times and the receiver samples the data correctly.

Transmitter and receiver need a control circuit and a counter to generate the proper signal for the *enable* input of their ring oscillators. The counter counts the number of generated clocks by the ring oscillator and when the proper number of clocks is generated, clears the output of controller (the *enable* of the transmitter ring oscillator). Figure 1 shows the block diagram of this scheme. The controller sets its output (*enable* signal) to '1' at both of the positive and the negative edges of the strobe input and reset its output to '0' at the positive edge of disable input. This circuit does nothing at the negative edge of the disable input.

Figure 2 shows the block diagrams of the transmitter and the receiver, and Figure 3 shows the complete waveforms of the proposed scheme. As evident from the figure, the data is transferred on the line after the strobe rising or falling edge. Since the lengths of the data and the strobe lines are approximately the same, these lines have the same delay. Thus, the strobe reach to the receiver sooner than the data with the amount equal to $t_{strobe \rightarrow data}$. The strobe signal enables the clock generator at the receiver and a shift register samples the data line at every clock of the receiver clock generator. Consequently, the positive edge of CLK_r will occur sooner than the first bit of the data with the amount equal to $t_{clock \rightarrow data}$ while the negative edge of CLK_r will take place in the middle of the data bit time. This indicates that we should use the negative edge of the clock for sampling the data line in the receiver. After receiving a complete frame of the data, the *disable* signal is activated stopping the ring oscillator. Then the output port read the data from the receiver shift register, activate the clear signal, and prepare the receiver for receiving the next data frame.

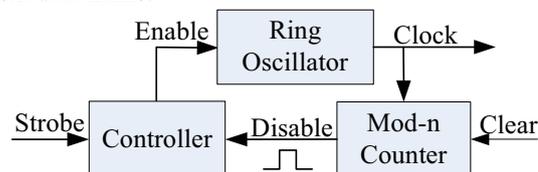


Figure 1. Complete circuit of the clock generator.

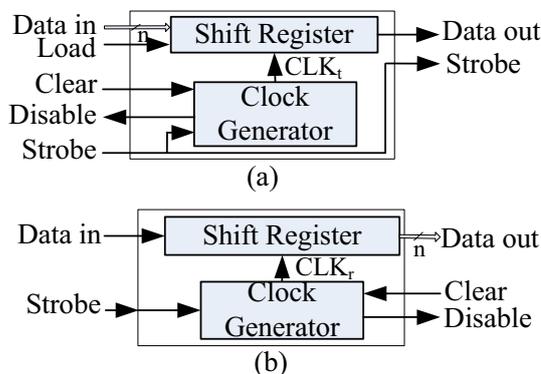


Figure 2. Block diagram of (a) transmitter, (b) receiver.

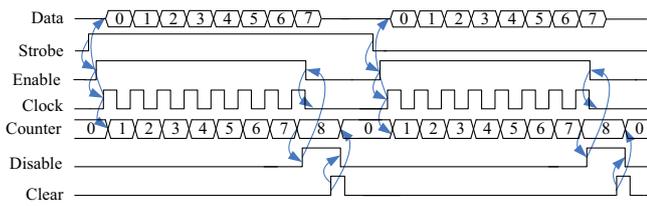


Figure 3. Complete waveforms of the proposed scheme.

Referring to Figure 3, the delay between the edge of the strobe signal and the start of the data transfer is equal to the sum of three terms as

$$t_{\text{strobe} \rightarrow \text{data}} = t_{\text{strobe} \rightarrow \text{enable}} + t_{\text{enable} \rightarrow \text{clock}} + t_{\text{clock} \rightarrow \text{data}} \quad (1)$$

Here, $t_{\text{strobe} \rightarrow \text{enable}}$ is the delay from the edge of the strobe to the positive edge of the enable (the delay of the controller), $t_{\text{enable} \rightarrow \text{clock}}$ is the delay from the positive edge of the enable to the positive edge of the clock (the delay of the ring oscillator), and $t_{\text{clock} \rightarrow \text{data}}$ is the delay from the positive edge of the clock to the edge of the first data bit (the delay of the parallel to the serial converter or the transmitter shift register).

III. MAXIMUM RECEIVER CLOCK TOLERANCE

In the proposed scheme, the transmitter and the receiver ring oscillators are the same and should have the same frequency. However, due to process and other variations which lead to some differences in the receiver and the transmitter clock frequencies. Our technique allows some difference in the clock frequencies. The tolerable range of the difference between the periods of the two ring oscillators depends on the number of bits to be transferred per frame. Let T be the period of the transmitter clock and the period of the receiver clock be in the range of $[T_{\min}, T_{\max}]$. To prevent the read error caused by the metastability, we need to know when the timing relation is not correct. We use the setup and the hold times of the receiver shift register to describe the timing alignment of the clock edge. There is a time window with the total length equal to the sum of the setup and the hold times in which the input data is not stable [9]. If we sample the data line in this window, we read wrong data due to the metastability problem. In the proposed structure, we add a small delay in the path of the strobe signal in the receiver to adjust the first negative edge of receiver clock in the middle of the first data bit time. Next, we should make sure that the edge of the last receiver clock do not occurs in the failure zone. Figure 4 shows these conditions. In Figure 4(a), the frequency of the Receiver ring oscillator is lower than the frequency of the transmitter ring oscillator and in Figure 4(b), the frequency of the receiver ring oscillator is higher than the frequency of the transmitter ring oscillator. In both of these figures, all negative edge of the receiver clocks are in the correctly region and the receiver samples input data correctly. In this figure, T_s and T_h are the setup and hold times of the receiver shift register.

Referring to the figure, the condition of the error free sampling in the receiver may be expressed as

$$n.T - T_s > (n-1/2).T_{\max} \quad (2)$$

$$(n-1)T + T_h < (n-1/2).T_{\min} \quad (3)$$

where n is number of bits per frame. Using these inequalities, one may write

$$f_{r,\min} < f_r < f_{r,\max} \quad (4)$$

where

$$f_{r,\min} = \left(\frac{(n-1/2)}{(n-T_s \cdot f_t)} \right) \cdot f_t \quad (5)$$

$$f_{r,\max} = \left(\frac{(n-1/2)}{(n-1+T_h \cdot f_t)} \right) \cdot f_t \quad (6)$$

Here, $f_t (=1/T)$ and f_r are the clock frequencies of the transmitter and the receiver, respectively.

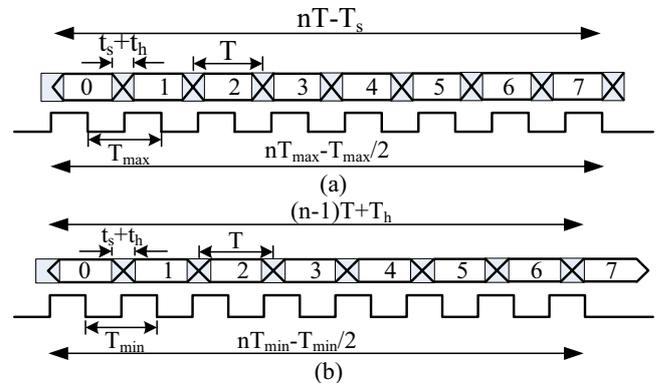


Figure 4 Timing diagram of receiver. (a) Receiver ring oscillator is slower than transmitter ring oscillator. (b) Receiver ring oscillator is faster than transmitter ring oscillator.

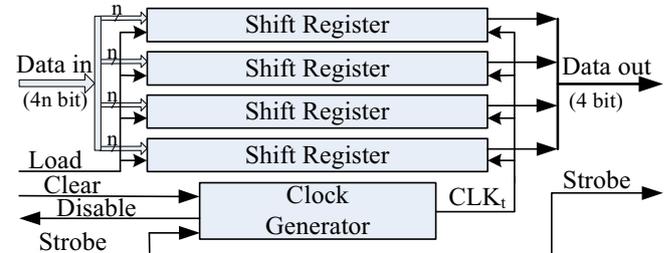


Figure 5. The block diagram of the proposed scheme for a 4-bit parallel bus.

IV. PARALLEL MESOCHRONOUS DATA TRANSFER

The proposed technique may be easily adapted to the parallel communication between cores in SoCs. For this purpose, it is sufficient that we use several shift register in the transmitter and the receiver with the same clock. Figure 5 shows the block diagram of the transmitter for a 4-bit data bus. The block diagram of the receiver is similar.

V. RESULTS AND DISCUSSION

To evaluate the performance of the proposed technique, the complete circuit shown in Figure 2 has been simulated at the

transistor level using a 0.13 μm standard CMOS technology. The simulations were performed for a serial and a 4-bit parallel buses. Figure 6 shows the waveform of the data (the first output of the receiver shift register), the strobe, the clock, and the enable of the ring oscillator at the receiver of the serial bus. In this figure, the receiver has read the sequence of "10101010". The results show that the transmitter and the receiver can operate up to 4.05GHz as the clock frequency. About one periods of the clock are required for reading a frame of the received data into the receiver shift register without the metastability error. The same amount of the time is required for loading new data in the transmitter shift register. Therefore, the transmission bandwidth will be 3.65 Gbps with 8-bit frames on the serial bus. This frequency increase to 14.6 Gbps by using a 4-bit parallel bus. The number of transistors used for the single strobe transceiver for 8-bit frames was equal to 396 (without the line drivers).

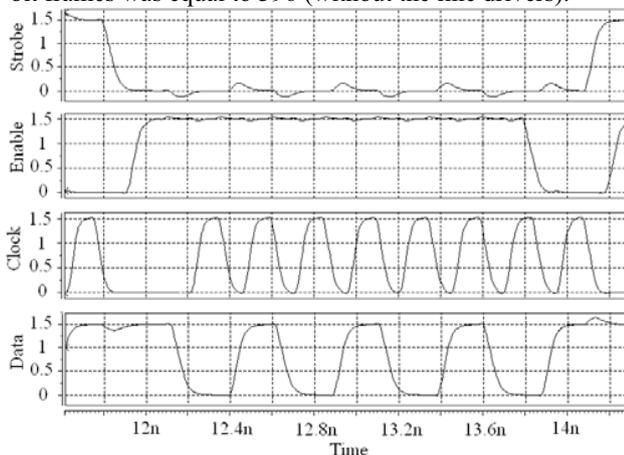


Figure 6. Output waveforms of the proposed scheme.

To see the effect of the number of bits per frame, we have used (5) and (6) to obtain the minimum and maximum frequency of the receiver. The results which are given in Table 1 show that as the number of the bits increases the receiver tolerance decreases.

Table 1
The permissible range of

n	4	8	9	10	16
$f_{r,min}$	0.875	0.938	0.945	0.95	0.968
$f_{r,max}$	1.166	1.071	1.063	1.055	1.033

VI. SUMMARY AND CONCLUSION

In this work, a mesochronous scheme for the serial communication was proposed. The technique can be easily adopted to parallel buses without any change in the control circuits. The interface consists of a strobe signal and a serial or parallel bus. The strobe signal changed only one time per every frame of the data. The transmitter and the receiver had separated ring oscillators for the clock generation. These oscillators were activated at the start of each frame and deactivated at the end of the frame. The number of the clock

pulses generated by these oscillators in every frame of data was equal to the number of bits in the frame. The scheme allowed some difference in clock frequencies of the transmitter and the receiver. The receiver frequency for 8-bit frames could vary from 93.8% to 107.1% of the transmitter frequency. The maximum frequency of the receiver or transmitter was obtained to be 4.05GHz in a 0.13 μm standard CMOS technology. This corresponded to a transmission bandwidth of 3.65Gbps.

REFERENCES

- [1] International Technology Roadmap for Semiconductors (ITRS 2003) (<http://public.itrs.net/>).
- [2] D. Bertozzi and L. Benini, "Xpipes: A Network-on-Chip Architecture for Gigascale Systems-on-Chip," IEEE Circuit and Systems Magazine, Second Quarter 2004, pp.18-31.
- [3] L. Benini, G. De Micheli, "Network on Chips: A New SoC Paradigm," IEEE Computer, Jan. 2002.
- [4] L. Benini, G. De Micheli, "Network on Chips: A New Paradigm for System on Chip Design," Proceedings of the 2002 Design Automation and Test in Europe Conference and Exhibition (DATE'02).
- [5] S. Kimura, T. Hayakawa, T. Horiyama, M. Nakanishi, and K. Watanabe, "An On-Chip High Speed Serial Communication Method Based on Independent Ring Oscillators," IEEE International Solid-State Circuits Conference (ISSCC 2003), pp. 390-391, 2003.
- [6] I. C. Wey, L. H. Chang, Y. G. Chen, S. H. Chang, and A. Y. Wu, "A 2Gb/s High-Speed Scalable Shift-Register Based On-Chip Serial Communication Design for SoC Applications," IEEE International Symposium on Circuits and Systems, ISCAS 2005, 23-26 May 2005 pp.1074 – 1077, Vol. 2.
- [7] E. Beigné, F. Clermidy, P. Vivet, A. Clouard, M. Renaudin, "An Asynchronous NOC Architecture Providing Low Latency Service and its Multi-level Design Framework," Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'05), 2005.
- [8] M. Nazm-Bojnordi, M. Semsarzadeh, A. Banaiyan, and A. Afzali-Kusha, "A Simple, Low-Cost and Low-Power Switch Architecture for NoCs," The 17th International Conference on Microelectronics (ICM) 2005. 13-15 Dec. 2005 Page(s):194 - 197
- [9] F. Mu and C. Svensson, "Self-Tested Self-Synchronization Circuit for Mesochronous Clocking," IEEE Trans. on Circuits and Systems, vol. 48, No. 2, Feb. 2001, pp. 129-140.
- [10] I. Söderquist, "Globally Updated Mesochronous Design Style," IEEE Journal of Solid-State Circuits, VOL. 38, No. 7, July 2003, pp. 1242-1249.
- [11] B. Mesgarzadeh, C. Svensson, and A. Alvandpour, "A New Mesochronous Clocking Scheme for Synchronization in SoC," IEEE Int. Symposium on Circuits and Systems (ISCAS04), 2004.