Low-Latency Quasi-Synchronous Transmission Technique for Multiple-Clock-Domain IP Modules

Jhao-Ji Ye, You-Gang Chen, I-Chyn Wey, and An-Yeu (Andy) Wu

Graduate Institute of Electronics Engineering, and Department of Electrical Engineering,
National Taiwan University, Taipei 106, Taiwan, R.O.C.

Abstract—Data transmission on multiple clock domains will face reliable problems. The conventional globally asynchronous locally synchronous (GALS) technique can resolve the problem but has a high latency problem. In this paper, we present a novel asynchronous transmission technique called quasi-synchronous with an adaptive phase mechanism to reduce the transmission latency. Compared with the conventional GALS techniques, the proposed technique saves 50%~83% of latency. It is implemented on standard-cell library by using TSMC 0.18um 1P6M CMOS technology.

I. INTRODUCTION

Recently, System-on-Chip (SoC) design is a hot issue, but it will suffer some problems, for example, wire connection complexity, cross talk, data synchronization, multiple clock sources, and etc. In this paper, we will focus on the multiple clock-sources issue. A global clock source will limit the chip design performance and each IP module can not run at maximum speed because of the clock skew problem. Therefore, multiple clock sources is a trend for future SoC design. It is a challenge for transmitting data reliably between multiple clock domains. A conventional synchronous transmission method will cause data losing between multiple clock domains. Another method suitable for transmitting data between different clock domains is an asynchronous technique. Fig. 1 shows the condition of transmitting data between different clock domains by asynchronous and synchronous methods. In the Fig. 1 (a), the transmitter and receiver are operated at separated clock source, which are different in clock phase. In this case, the asynchronous method can transmit data reliably, but the synchronous method will lose data. In Fig. 1(b), the transmitter and receiver are operated at different clock rate and the synchronous method will lose data more seriously when the relative clock rate ratio increases. In several asynchronous transmission technologies, adaptive synchronization [1] and STARI [2] are used to compensate clock skew in a system driven by a global clock source. Pipeline synchronization [3], efficient self timed interfaces [4] can be used in unrelated clock domains of interfaces. However, these asynchronous based interfaces will suffer from a data-

losing problem due to metastability. A technology can eliminate the failure of metastability by using clock pausing mechanism, which is called globally asynchronous locally synchronous (GALS) technique. This technique can transmit data between two separated clock domains including different clock phase and clock rate.

Fig. 1 Asynchronous method vs. synchronous method

II. PROPOSED ASYNCHRONOUS TECHNIQUE ARCHITECTURE

2.1 Design issues in GALS

Fig. 2 GALS block diagram

Fig.2 shows a GALS block diagram [5]. The GALS technique can transmit data reliably by using handshake based protocol for multiple clock domains. However, when the transmitter transmits data streams to the receiver, it needs more latency to receive one data in the receiver because of the

handshake based protocol. Moreover, the high latency issue will degrade the average data throughput compared with synchronous based design. Besides, the high latency characteristic will cause high energy consumption because the transceiver must cost more time to finish total data transmission. Thus, high latency, low data throughput and high energy consumption are the factors which cause the GALS technology is not popular on existing SoC designs.

2.2.1 Proposed quasi-synchronous technique

In order to resolve the problems described above, we propose a new asynchronous technique to improve the natural limitation of GALS design and preserve the characteristics of transmitting data reliably. Fig.3 shows the idea of the proposed circuit. We replace a pair of asynchronous handshake lines by one forward control line. Due to not to consider the return timing of the ack signal, it can reduce the latency of the conventional asynchronous technology. The data is transmitted by the value transition of the control signal and the control line is generated by the clock source of the transmitter. The relationship of control line and data is shown in Fig.3 (b). When the control line has a level transition, one data is transmitted. Therefore, the transmitter transmits one data per cycle and the receiver receives a data per cycle.

Fig. 3 The idea of the proposed technique

The proposed asynchronous technique architecture is shown in Fig.4. The control line triggers the mechanism of the receiver to receive data reliably. Because the latency of transmission time is reduced, it can improve the original data throughput and save some energy consumption.

Fig. 4 The architecture of the proposed technique

Table 1 The flow control mode

[Tx/Rx] condition	Mode
1	1x
2	2x
3,4	4x
5,6,7,8	8x

2.2.2 Transmitter architecture

For multiple clock domains, there are three cases for transmitting data, the transmitter clock rate is slower than the receiver clock rate, the transmitter clock rate is equal to the receiver clock rate and the transmitter clock rate is faster than the receiver clock rate. For the first two cases, receiver can receive data normally, but in the third case, data may lose because the operating frequency of the receiver is slower than the operating frequency of the transmitter. A flow control mechanism is used to slow down the data rate of the transmitter and makes sure that data can be transmitted reliably. This mechanism is used to control “control” signal rate and supports different modes for the corresponding state. In the beginning, we divide the transmitter clock rate by the receiver clock rate and get the integer value; it is the parameter to determine the flow control mode. Table 1 shows the flow control mode with the corresponding integer value.

2.2.3 Receiver architecture

In the receiver, if a synchronous circuit must synchronize an asynchronous input, such as using a single D-type flip-flop, the circuit may enter an unstable state when the clock edge arrives too close to the data arriving from an asynchronous circuit; this is called a metastable state. To solve this situation, we can stop clock when the D flip-flop of the receiver is ready to receive data and active clock when input data is stored on the D flip-flop of the receiver. We use a mutual exclusion mechanism to achieve gating clock. When the *control* signal has a transition of value, the receiver will start to receive asynchronous data from the transmitter. If *control* signal occurs a transition, *xor_out* signal will pull high, through mutual exclusion arbitrating, data will be stored in the first D flip-flop, then, *xor_out* signal will pull low, at this same time, mutual exclusion will allow *clkB* to go through and trigger the second D flip-flop to receive the data which is stored in the first D flip-flop.

2.2.4 Adaptive phase mechanism

When the transceiver works on high clock rate, the phase drift of the control signal will cause the receiver to receive data abnormally due to wire load effect. Thus, we proposed an adaptive phase mechanism to resolve this situation. The block diagram of the adaptive phase mechanism is shown in Fig. 5. The phase/ frequency detector (PFD) compares the phase difference of the two input signals and predict whether the difference exists or not. If it does, the control circuit will send a control signal to inform the delay line to delay *clkB* signal. After several cycle of this action recursively, the phase difference of the two signal, *lineA* and *clkB* achieve close zero, and then let *clk_out* signal pass through a NOT gate, we will get the new *clkB* signal whose phase lags *lineA* signal half cycle and the receiver can receive data correctly.

Fig. 5 The adaptive phase mechanism

The adaptive phase range will be affected by the resolution of PFD, design a less dead zone and stable signal output (no glitch output) phase detector is what we expected. Fig. 6 shows the classical digital phase detector [7]. *DN* and *UP* signal is instantly triggered by reference clock and DCO clock and reset by amplified signal of *S1* and *S2*. When implemented, the dead-zone will be decided by the *NAND* gate. It means the charge-time is not enough, moreover, the *G1* and *G2* can not be reset completely, and the dead-zone scene is generated. We suggest pulse-amplified stages *R1*, *R2* can be added to the output of *G3* that dead-zone can be reduced.

Fig. 6 Phase frequency detector

III. PERFORMANCE COMPARISONS

In this session, we will compare the performance with conventional GALS and basic synchronous circuits. And all results are based on TSMC 0.18um process technology. We compare the reliability in multiple clock domains, it has two situations: different clock rate and different clock phase. Fig. 7(a) shows the comparison of basic synchronous circuits operated by different clock rate. As shown in Fig. 7 (a), the x-axis means the phase difference range of the transmitter and the receiver for one cycle, the proposed technique has no data lost condition; however, synchronous methods will lose data in some phase difference region. Fig. 7 (b) shows the comparison under relative phase difference and the x-axis shows the relative clock rate ratio of the transmitter and the receiver. Table 2 is quantitative values of the two situations. From the table 2, we can find that the data loss probability of the proposed asynchronous technique is 0%. Therefore, it is reliable for transmitting data during multiple clock domains.

Fig.7 Transmit (a) at different clock rate,(b) under relative phase difference

Table 2 Summary table of data loss probability

Method	Data loss	Clock rate ratio (Tx/Rx)	sync. 20% Phase diff.	sync. 40% Phase diff.	sync. 60% Phase diff.	sync. 80% Phase diff.	proposed
synchronous 100MHz	0-50%						
synchronous 200MHz	0-100%	1	100%	0%	0%	0%	0%
synchronous 300MHz	0-100%	2	50%	50%	50%	50%	0%
synchronous 400MHz	0-100%	4	75%	75%	75%	75%	0%
synchronous 500MHz	0-100%	8	87.5%	87.5%	87.5%	87.5%	0%
proposed	0%	16	93.75 %	93.75 %	93.75 %	93.75 %	0%

Fig.8 shows the latency and throughput charts. Table3 is a quantitative result and shows 50%~83% latency-saving compared with the conventional GALS technique when operated in different clock rate. From the table3, the proposed asynchronous technique has 2x~6x data throughput improvement due to the single control line technique. Fig.9 represents the energy-consumption chart when the data is transmitted from the transmitter to the receiver. From the table4, the proposed asynchronous technique has 40%~82% energy-consumption saving.

Fig.8 Latency and throughput comparisons

Table 3 Summary table of Latency and throughput improvement

Clock rate	Latency saving		Throughput improvement	
	Four phase	Two phase	Four phase	Two phase
100Mhz	75%	50%	4X	2X
200Mhz	75%	50%	4X	2X
300Mhz	75%	50%	4X	2X
400Mhz	75%	50%	4X	2X
500Mhz	83%	50%	6X	3X

Fig.9 Energy saving

Table 4 Summary table of Energy saving

Clock rate	Energy Saving	
	Four phase	Two phase
100Mhz	70%	40%
200Mhz	70%	40%
300Mhz	70%	40%
400Mhz	73%	46%
500Mhz	82%	65%

IV. CHIP IMPLEMENTATION

We implement and integrate the conventional GALS technology and the proposed technique into a chip. Because the limitation of IO pad is 200 MHz, we establish a digital clock oscillator (DCO) to support clock rate from 300 to 500 MHz. Fig.10 shows the chip layout using TSMC 0.18 um CMOS technology. The characteristics of the chip are summarized in the right-hand side of Fig. The maximum frequency of the chip is 500 MHz and the die size of layout is 0.92 mm x 0.92mm. The power consumption is 8mW at the maximum speed.

Fig. 10 Chip layout and performance summary

V. CONCLUSION

In this paper, we propose a novel transmission technique which improves the conventional GALS technique natural defects on latency and energy consumption. Compared with the conventional GALS technology, it has the characteristics of low latency, high throughput and low energy consumption, it saves 50%~83% latency timing, improves 2x~6x data throughput and reduces 40%~82% energy consumption.

VI. REFERENCE

- [1] R. Ginosar and R. Kol, "Adaptive synchronization," in Proc. of ICCD, pp. 188-189, October 1998.
- [2] M.R. Greenstreet, "Implementing a STARI chip," in Proc. of 1995 Int. Conf. on Computer Design, pp.38-43, Oct. 1995.
- [3] J. N. Seizovic, "Pipeline synchronization," in Proc. of the 1st Int. Symp. on advanced research in Asynchronous Circuits and Systems, pp.87-96, 1994.
- [4] A. Chakraborty and M. R. Greenstreet, "Efficient selftimed interfaces for crossing clock domains," in Proc. of ASYNC'03, May 2003.
- [5] J. Muttersbach, T. Villiger, and W. Fichtner, "Practical design of globally-asynchronous locally-synchronous systems," in Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, pages 52-59, April 2000.
- [6] D.M. Chapiro, "Globally asynchronous locally synchronous systems," PhD. thesis, Stanford University, 1984.
- [7] Ching-Che Chung and Chen-Yi Lee, "An All-Digital Phase-Locked Loop for High-Speed Clock Generation," IEEE J. Solid-State Circuits, vol.38, pp. 347-351, Feb 2003.
- [8] A. Hatakeyama, "A 256-Mb SDRAM using a register-controlled digital DLL," IEEE J. Solid-State Circuits, vol. 32, pp. 1728-1732, Nov. 1997.