A self-adaptive full asynchronous bi-directional transmission channel for network-on-chips*

Guan Xuguang(管旭光)†, Yang Yintang(杨银堂), Zhu Zhangming(朱樟明), and Zhou Duan(周端)

(Institute of Microelectronics, Xidian University, Xi'an 710071, China)

Abstract: To improve two shortcomings of conventional network-on-chips, i.e. low utilization rate in channels between routers and excessive interconnection lines, this paper proposes a full asynchronous self-adaptive bi-directional transmission channel. It can utilize interconnection lines and register resources with high efficiency, and dynamically detect the data transmission state between routers through a direction regulator, which controls the sequencer to automatically adjust the transmission direction of the bi-directional channel, so as to provide a flexible data transmission environment. Null convention logic units are used to make the circuit quasi-delay insensitive and highly robust. The proposed bi-directional transmission channel is implemented based on SMIC 0.18 μm standard CMOS technology. Post-layout simulation results demonstrate that this self-adaptive bi-directional channel has better performance on throughput, transmission flexibility and channel bandwidth utilization compared to a conventional single direction channel. Moreover, the proposed channel can save interconnection lines up to 30% and can provide twice the bandwidth resources of a single direction transmission channel. The proposed channel can apply to an on-chip network which has limited resources of registers and interconnection lines.

Key words: network-on-chip; bi-directional transmission channel; full asynchronous; dual-rail encoding; threshold gates

DOI: 10.1088/1674-4926/31/8/085008 **EEACC:** 1265B

1. Introduction

As semiconductor technology scales down, more IP cores are being integrated onto a single chip to implement more complicated system functions. But with the increasing working speed and larger scale of on-chip systems, conventional single clock operating mode faces a lot of challenges, such as poor reusability of modules, power consumption increment of the clock tree, large clock tree area, clock skew and EMI. Because of these problems, the complexity of designing very deep submicron integrated circuits is greatly enhanced. So problems caused by the clock have become crucial issues which need to be solved first in ultra-large integrated circuits.

To reduce power consumption and increase communication performance, extensive research has been conducted into network-on-chip (NoC)[1−4] systems. The NoC approach particularly suits communication-dominant on-chip systems. Asynchronous NoCs have been proposed to eliminate the clock for global communication[5, 6], providing better power efficiency and higher modularity compared to synchronous NoCs. However, limitations on area and power consumption in network-on-chips make the connection lines and channel resources more precious. Traditional connections between NoC routers comprise a pair of channels, one output channel and one input channel. But in most cases, not all the routers have data to send and receive. It is often observed that the output channel may be flooded with out-going traffic while the input channel remains idle[7], which greatly wastes the connection lines and buffer resources. Thus, the performance of the channel is lowered. To overcome these shortcomings, the proposed self-adaptive full asynchronous bi-directional transmission channel can resolve these problems. It can efficiently utilize connection lines together with buffer resources, and provides better transmission performance at the same time, which can meet the requirements of network-on-chips with limited resources.

2. Network-on-chip and transmission channels

The operating mode of network-on-chips refers to the data transmission mode in computer networks. Data can be transmitted to the corresponding target module by route switching, which substitutes the conventional data transmission mode in bus-based architecture. So it has a high concurrent transmission capacity and expansibility[8]. Owing to transmission data in asynchronous on-chip networks are reached by handshake signals other than the clock, so problems caused by the clock can be eliminated, and modularity is also greatly enhanced[9]. Although on-chip networks have various topologies, 2D mesh topology has gained more consideration by designers due to its simplicity, and Figure 1 shows its structure. An NoC mainly consists of computational processing elements (PE), network interfaces (NI), and routers. The latter two comprise the primary communication architecture. The network interface is used to packetize data or unpack data. Only packetized data can be sent to routers to traverse the NoC. The packet is forwarded hop by hop on the network until it reaches the destination. For

* Project supported by the National Natural Science Foundation of China (Nos. 60725415, 60971066), the National High-Tech Research and Development Program of China (Nos. 2009AA01Z258, 2009AA01Z260), and the National Science & Technology Important Project of China (No. 2009ZX01034-002-001-005).

† Corresponding author. Email: guanxuguang_5@126.com

Received 1 February 2010, revised manuscript received 18 March 2010

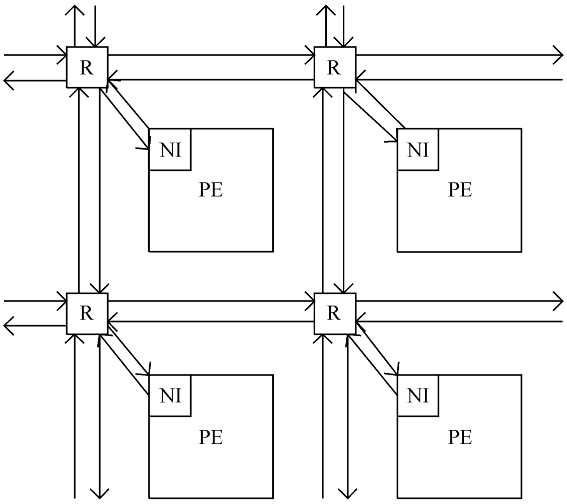


Fig. 1. Structure diagram of a 2D mesh network-on-chip.

Table 1. Dual-rail quasi delay-insensitive encoding.

D	D0	D1
Data 0	1	0
Data 1	0	1
Null	0	0
Invalid	1	1

the destination router, the packet is first received and stored at an input buffer, and sent into the PE after unpacking the packet.

Single rail operating mode is widely used in conventional NoC design schemes[10−12], because it is similar to synchronous operating mode and can effectively reuse existing synchronous units. But it also has unavoidable shortcomings. For example, extra circuits are needed to generate control signals, and delay-matching is needed to exert an influence on the control path. Furthermore, static timing analysis is required to perform to the whole system, and inappropriate design in control circuit can make the whole circuits go into an unknown state and produce errors. Additionally, as the characteristic size of integrated circuits shrinks and system working speed increases, designers will face more challenges in handling glitches and circuit errors. Also, particles, cross-talk and noise make the on-chip interconnections likely to show soft-errors.

Therefore, it is necessary to adopt a kind of highly robust encoding scheme to diminish or avoid the errors above. Here, dual-rail quasi delay-insensitive encoding is used[13]; the specific encoding scheme is shown in Table 1. Each bit data is represented by two lines. “01” represents logic 0, while “10” represents logic 1. “00” state is a null state, and this state is needed to separate two data cycles. That is, each data transmission must go through a null state before getting into the next transmission cycle. In dual-rail operating mode, data also represent request signals, so the design of control circuits can be avoided. Furthermore, dual-rail operating mode employs a flow control mode named “back pressure”[14], so its normal working is not susceptible to delay variations. These advantages have made it very popular with asynchronous designers recently[15].

Single direction transmission channels are widely used in modern 2D mesh network-on-chips to transmit data[16−18]. A pair of transmission channels is used between two adjacent routers. One is used to input data, while the other is used to output data, as shown in Fig. 2.

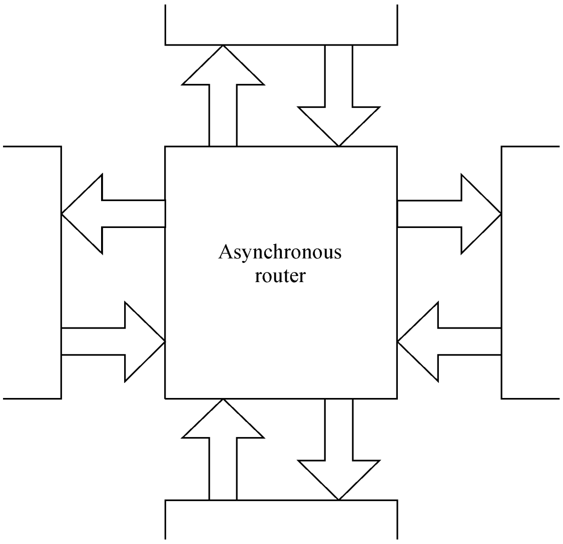


Fig. 2. Conventional transmission channels in a 2D-mesh network-on-chip.

To improve the channel bandwidth utilization rate, this paper proposes a self-adaptive bi-directional transmission channel. To the best of our knowledge, there have been no previous reports of full asynchronous dual-rail bi-directional channels. The bi-directional transmission channel can provide four transmission modes. They are single channel forward transmission, single channel backward transmission, dual channel forward transmission and dual channel backward transmission respectively, as shown in Fig. 3. The working state is controlled by an empty/full detection circuit and steering circuit. The free channel can dynamically change its direction to help the other channel transmit data, thus the throughput is improved. If both of the routers have data to send, then one channel is allocated to one router, and this circumstance is just the same as in single channel transmission mode; if one router completes its transmission, then the channel can change its direction immediately to help the other channel to transmit data. Each channel in our paper can transmit data in both directions, so it has high flexibility and is suitable for various communication flow types.

Comparisons between the working mode and working time are given for the single direction and bi-directional channels, as shown in Fig. 4. Tsingle and Tdouble represent the transmission time of the two channels respectively. Owing to parallel transmission mode in the bi-directional channel, the transmission time used is much less than that for the single direction channel. This can be very effective in improving blocking and congestion problems caused by on-chip burst mode data transmission.

3. Implementation and analysis of the bi-directional transmission channel

The design of the bi-directional transmission channel needs to meet three major targets. Firstly, data must be sampled

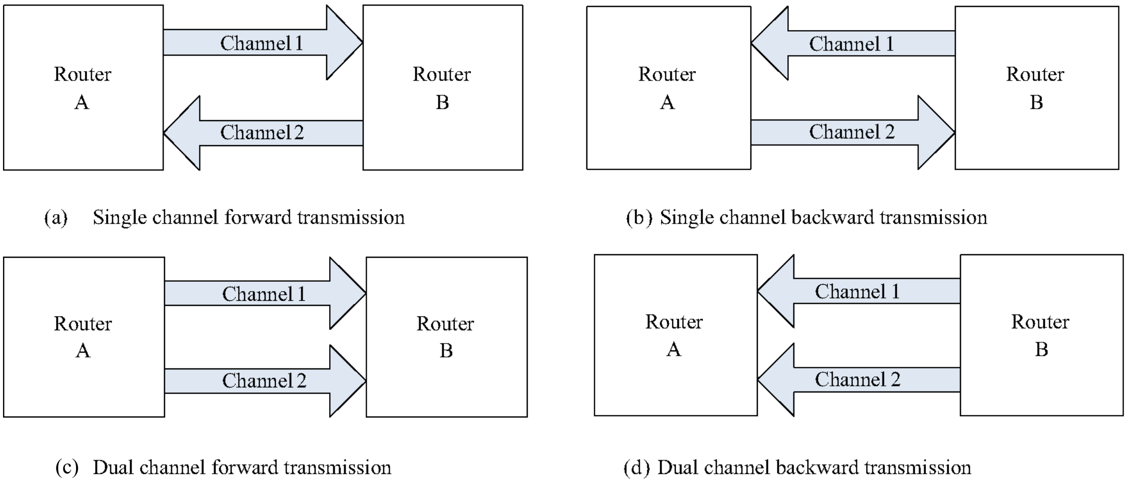

Router A
Channel 1
Channel 2
Router B
(a) Single channel forward transmission
(b) Single channel backward transmission
(c) Dual channel forward transmission
(d) Dual channel backward transmission

Fig. 3. Four kinds of transmission mode in the proposed bi-directional transmission channel.

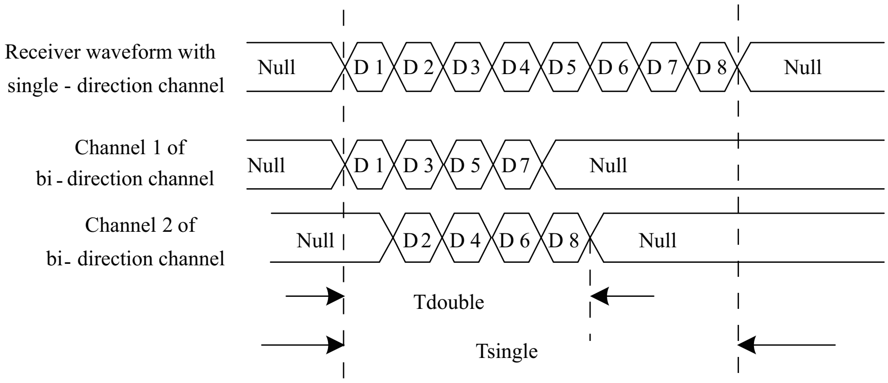


Fig. 4. Comparisons of runtime for the two transmission channels.

safely, and this is the foundation of correct operations of the circuit. Secondly, the empty/full state must be detected. If the empty/full state is improperly detected, data overlapping is likely to occur and cause errors. Thirdly, conflicts between steering signals must be avoided. Once a conflict occurs, data in channels will transmit at both directions and cause errors. Figure 5 is the specific implementation of the proposed bi-directional transmission channel; the whole circuit adopts dual-rail quasi-delay insensitive encoding to improve the robustness and immunity of the technology process.

The circuit in Fig. 5 can be divided into four parts. They are the input/output register queue of the routers, data distribution, data merge and bi-directional transmission channels. Control circuits are not shown in Fig. 5. The white transmission path is the forward transmission channel of router A, while the gray transmission path is the forward transmission channel of router B. The arrow in the channel represents the data flow. The thin arrow represents the data flow of router A, while the thick arrow represents the data flow of router B. It can be seen that the transmission paths of routers A and B are also connected to the receiver port, and input/output signals are selected through null convention logic DEMUX and MUX at sender and receiver. Sxy (x = 1–4, y = 1, 2) are the control signals of DEMUX and MUX, and they control data sending to either channel. The “control” signal is generated by a null convention logic sequencer, and its function is to enable the selection signals of MUX and DEMUX in a round-robin manner. Certainly, some modifications are needed to the sender and receiver sequencers before they can meet the requirements of single direction or bi-directional transmission. The sender sequencer is controlled by request signals (reqA or reqB) and steering signals (Enii and Enxy) as well as the “control” signal. A request signal is generated by the last register of the router’s output queue, and a high request signal represents that output data are ready to transmit. Enii and Enxy are steering signals; they control the working rhythms and transmission directions of the whole circuits. En11 represents the forward transmission enabling signal of router A; En12 represents the backward transmission (thin arrow in gray path) enabling signal of router B; En22 represents the forward transmission enabling signal of router B; En21 represents the backward transmission (thick arrow in white path) enabling signal of router A. Signals En11 and En21 must be mutually exclusive, as well as En22 and En12. Only in this way can the uni-directivity of the null convention logic registers be guaranteed, thus the channel can transmit in one direction in a period of time. The control signal is the “AND” of write signals of router A and router B. That is, the control signal rises only when two routers want to send data simultaneously, and this will make the router stop sending data into backward channels. In comparison, the receiver sequencer is only controlled by steering signals and acknowledgement signals of the router’s input register queue. Steering signals also control the receiver sequencer to

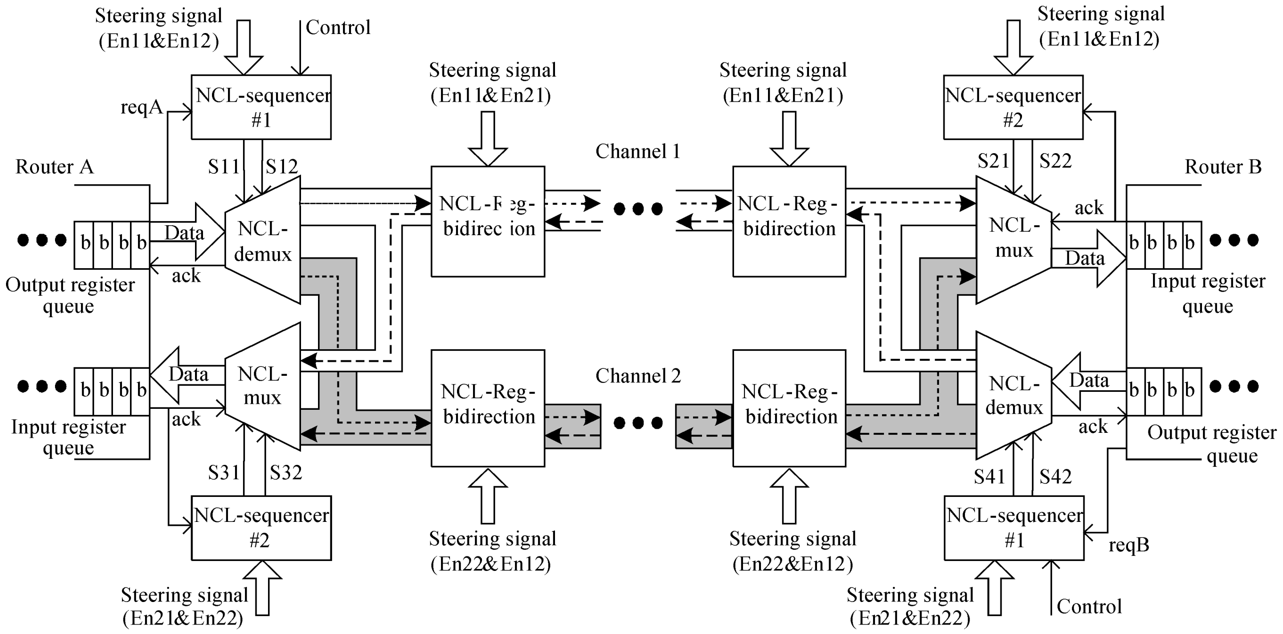


Fig. 5. Specific implementation of the proposed bi-directional transmission channel.

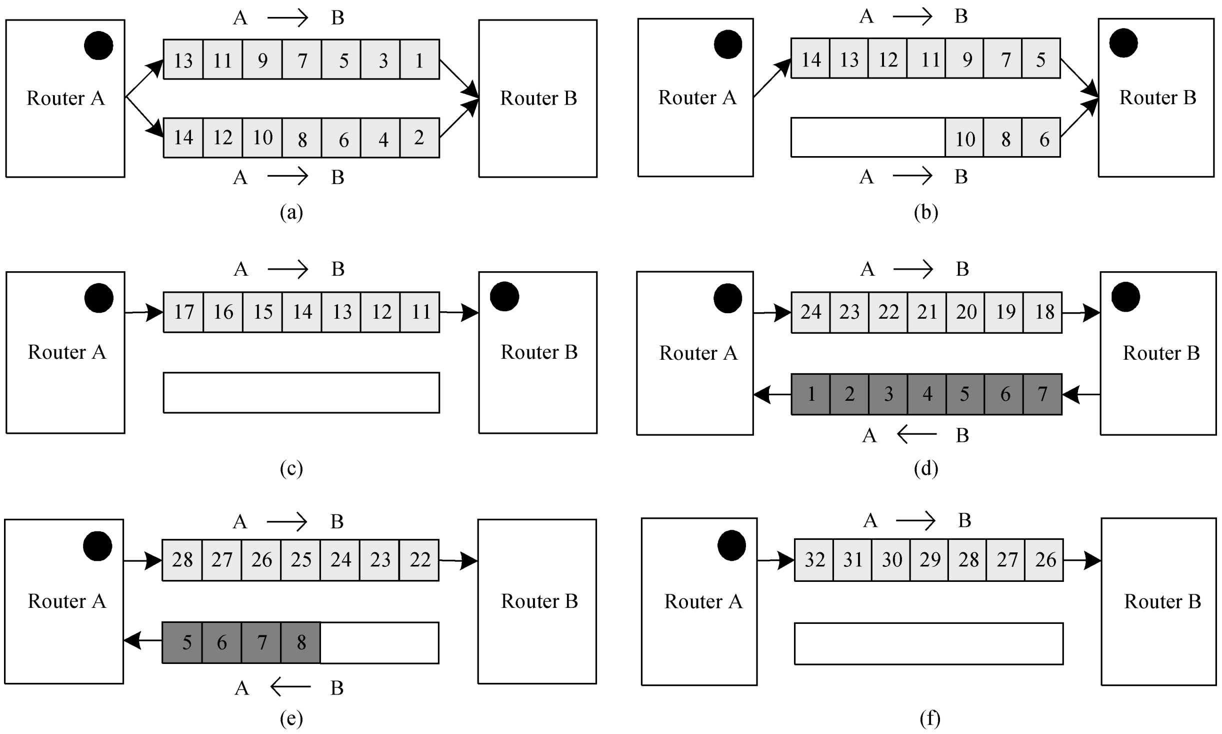


Fig. 6. Working procedure of a self-adaptive bi-directional transmission channel.

receive data from the corresponding channel. If the router has successfully received the data, it sends back an acknowledgement signal, which tells the MUX and sequencer to be ready to transmit the next piece of data.

Figure 6 shows the working procedure of the self-adaptive bi-directional transmission channel. The black point in the router indicates that the router has data to send; the white channel shows that the channel has no data; the gray channel represents the out-going data of router A; the dark gray channel represents the out-going data of router B. Arrows beside the channel show the direction of the channel, and the figure represents the sequence of data. Figure 6(a) shows that only router A has data to send, so both channels are used to transmit data in parallel mode. As can be seen from the diagram, data are sent to both channels in a round-robin manner, and router B needs to receive data between two channels alternately, so as to guarantee the sequence of the data. If router B now wants to send data, the “control” signal will go high, and router A stops sending data into channel 2, as shown in Fig. 6(b). We should note that channel 2 needs to keep the direction unchanged un-

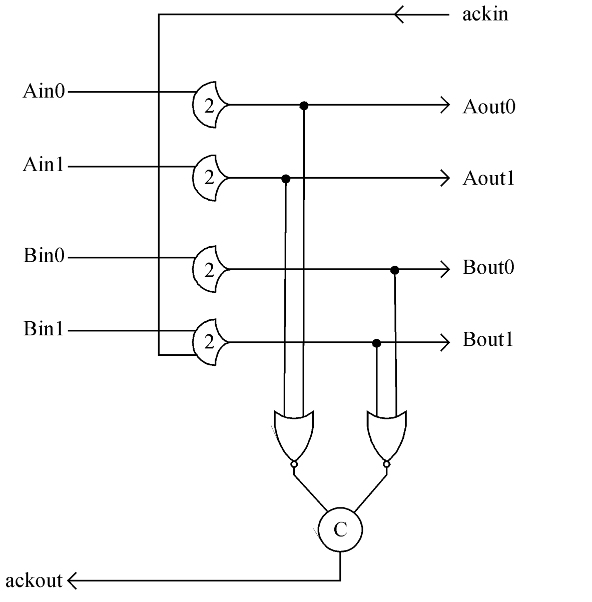


Fig. 7. Conventional dual-rail delay-insensitive asynchronous register.

til the data in channel 2 are all successfully received by router B, as shown in Fig. 6(c). The reason for doing so is to prevent router B sending data to channel 2 before the data in channel 2 are entirely received by router B. Otherwise, it may cause errors induced by data overlap. After router B has received the data, channel 2 begins to change its direction. At the same time, router B begins to send data, while router A begins to receive data, as shown in Fig. 6(d). Channel 2 needs to keep its direction unchanged until router B has completed sending the data, and router A will continue receiving the remaining data from router B until channel 2 is empty, as shown in Fig. 6(e). Now, if router A still has data to send, then channel 2 will change its direction to A → B, as shown in Fig. 6(f). Router A can keep on sending data to channel 2, and both of the channels go to the state of Fig. 6(a) again to proceed with dual channel transmission mode.

Figure 7 is the implementation of a conventional dual-rail delay-insensitive asynchronous register with data storage and completion detection. The Ackin signal indicates that the next stage has completed data storage, and asks the present stage for the next data, while the Ackout signal indicates that the present stage has completed data storage, and asks the previous stage for the next data. The Ackout signal is generated by the completion detection circuit of register outputs Aout0, Aout1, Bout0 and Bout1. The acknowledgement signal can not only control the normal working order of the register, but also can implement flow control through “back pressure”. This kind of register can only transmit in one direction, and $2n + 1$ lines are needed when transmitting n bits data.

Figure 8 is the proposed dual-rail delay-insensitive bi-directional asynchronous register. Null convention logic threshold gates with enable ports are used here, and the transmission direction is controlled by the enable signal. When En1 is high, the upper two threshold gates begin to work, and the data flow is from A to B. When En2 is high, the lower two threshold gates begin to work, and the data flow is from B to A. To guarantee that the data transmitted are correct, threshold gates in different directions cannot work simultaneously. That is, En1 and En2 cannot go high at the same time.

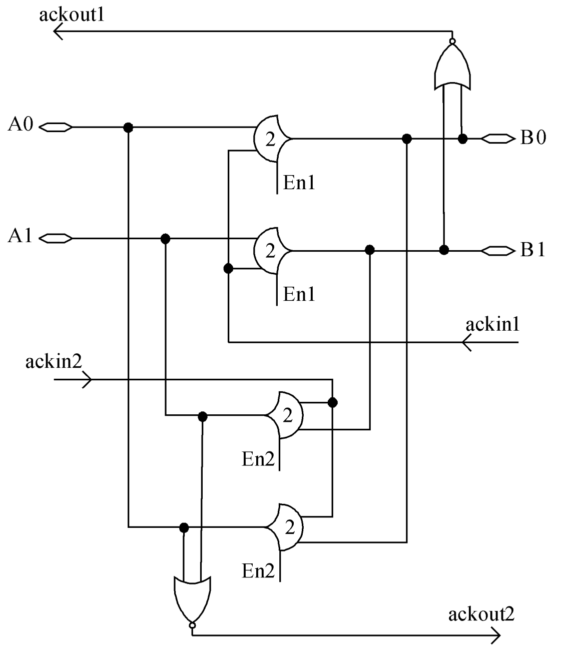


Fig. 8. Proposed dual-rail delay-insensitive bi-directional asynchronous register.

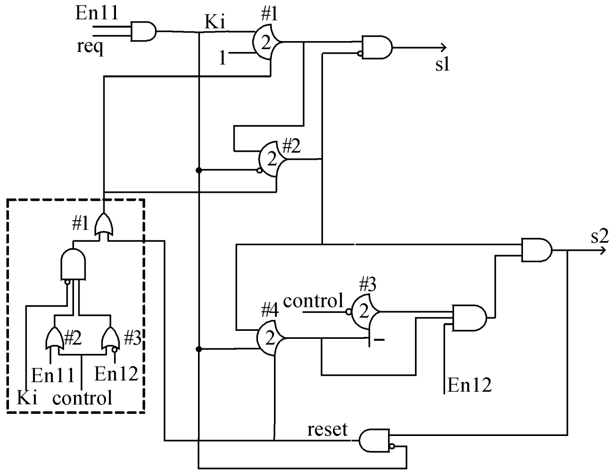


Fig. 9. Implementation of the sender sequencer.

Figure 9 is the specific circuit of the sender sequencer. The sequencer has two functions according to control signals. Here we take the sender sequencer of router A as an example. The first case is dual channel transmission mode. The “control” signal is low, while En11 and En12 signals are high. The output of the three input AND gate in the dashed frame is low, so the “reset” signal can propagate through the #1 OR gate, and now the sequencer selects the output through the input times of Ki. Ki is controlled by the En11 and req signals, and Ki is allowed to go high only when both En11 and req are high. The last input can be preserved in the circuit due to the hysteresis of threshold gates until Ki changes, which makes s1 and s2 become high alternately. s1 goes high when Ki becomes high for the first time, while s1 and s2 go low after Ki becomes low. When Ki

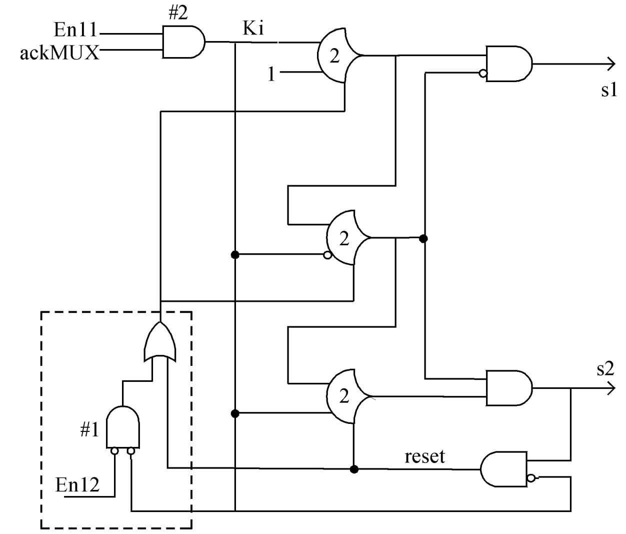


Fig. 10. Implementation of the receiver sequencer.

becomes high for the second time, s2 goes high, which indicates the sending of data to channel 2. The second case is single channel transmission mode. The “control” signal begins to go high; En11 is high and En12 is low. So the outputs of the #2 and #3 OR2 gates are both high, and s2 goes low due to En12 being low. Now the output of the #1 OR gate will be totally decided by the Ki signal. When Ki is high, s1 goes high normally; when Ki goes low, the #1 OR gate will go high, and this will make the #1 and #2 threshold gates reset. This indicates that the following data can only be sent into channel 1, and the channel begins to carry out single channel transmission mode. The sender sequencer of router B is just the same as the circuit in Fig. 9, and the only difference is the change of En11 and En12 into En22 and En21.

Figure 10 is the specific implementation of the receiver sequencer. Here we take the sender sequencer of router A as an example. Also, the sequencer has two functions. Firstly, the Ki signal is controlled by En11 and the acknowledgement signal of MUX. That is, Ki can go high only when the forward transmission channel is open and the in-going registers of the router have free spaces. In dual channel transmission mode, both En11 and En12 are high, and the output of the #1 AND gate is low, so the “reset” signal can get to the enable port of the threshold gates. The “reset” signal is only triggered after s2 is enabled as well as Ki being low. This guarantees that the whole circuit reverts to its original state only after s1 and s2 are triggered. Secondly, in single channel transmission mode, En11 is high and En12 is low. So when Ki is low, the output of the #1 AND gate in the dashed frame goes high. This will cause the upper two threshold gates to be reset, that is, s2 will not be triggered in the next period. The receiver sequencer of router B is just the same as the circuit in Fig.10, and the only difference is the change of En11 and En12 into En22 and En21.

Figure 11 is the steering signal generation circuit, and it controls the proper work of the sequencer and bi-directional registers. Here we take the steering signal generation circuit of channel 1 as an example. It can be found that the circuit includes three parts. They are the forward transmission regulator, backward transmission regulator and output module of

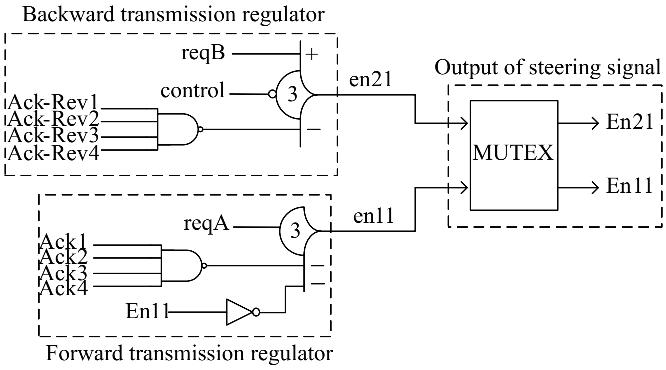


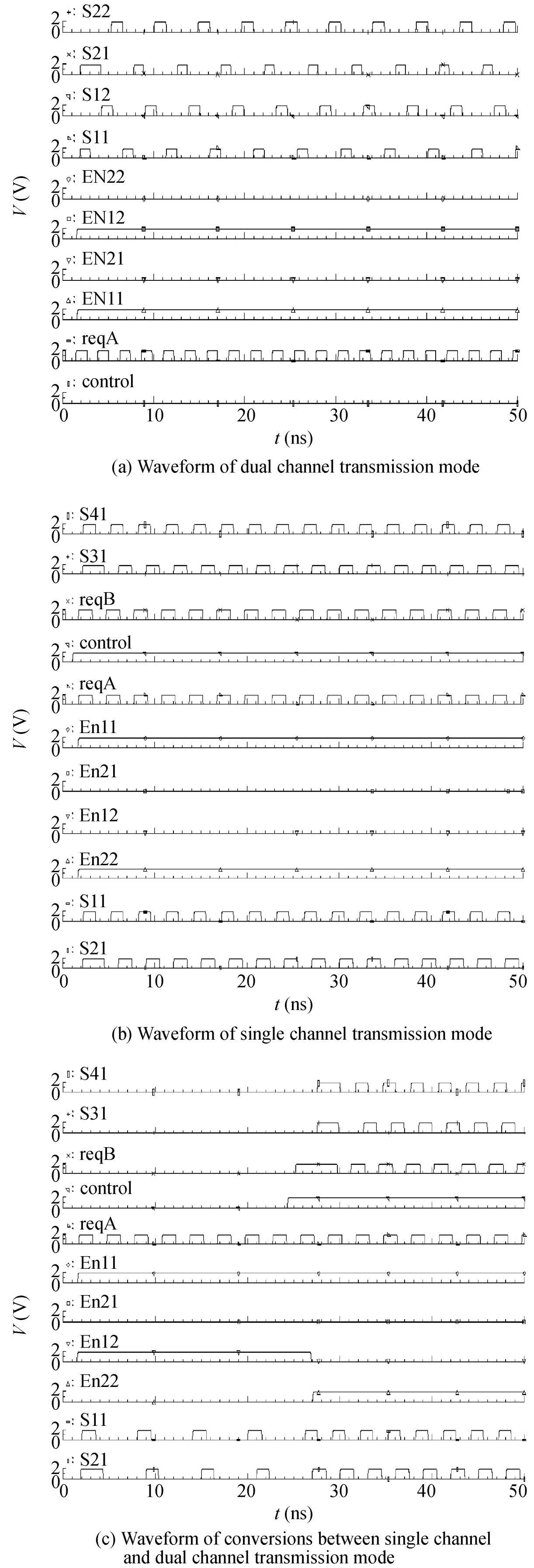
Fig. 11. Steering signal generation circuit.

steering signals. Asymmetric threshold gates and mutually exclusive units are used here. Ack1, Ack2, Ack3 and Ack4 are the forward acknowledgement signals of the registers, while Ack-Rev 1, Ack-Rev 2, Ack-Rev 3 and Ack-Rev 4 are the backward acknowledgement signals of the registers. When they are all high, it means there are no data in the channels. If at least one of the acknowledgement signals is low, it can be regarded as meaning data are still in the channel. A rise in reqA causes en11 to go high, but en11 may not output immediately. It depends on whether backward data transmission is still in process. en11 will wait at the input of MUTEX until en21 turns low. The advantage of this method is obvious, because it can avoid data overlapping. Likewise, the backward transmission regulator goes high only after reqB and control signals are both received, and en21 can go through MUTEX once en11 goes low. We should note that going low in en21 and en11 is restricted by the full/empty state of the channel, that is, directions are allowed to change only when data in the channels are successfully received by the corresponding router. This guarantees that data can be received intact. The direction generation circuit of channel 2 is just the same as Fig. 11, and the only difference is the exchange of the signals to channel 2.

4. Simulation results and analysis

All the self-adaptive bi-directional channel circuits are implemented using SMIC 0.18 μm CMOS standard technology. Simulations on dual channel transmission mode, single channel transmission mode and conversion mode were made respectively. Figure 12 shows part of the SPICE simulation waveform.

As can be seen from Fig.12(a), only router A is sending data, so the control signal is low, while En11 and En12 signals are high, which indicates that both channels are sending data to router B. S11 and S12 are the control signals of sender DEMUX; S21 and S22 are the control signals of receiver MUX. Each time data comes, the reqA signal rises. S11 and S12 signals go high alternately, sending data into channel 1 and channel 2. Likewise, S21 and S22 in the receiver also go high alternately, which controls the MUX to take data from channel 1 and channel 2. In the single channel transmission waveform of Fig. 12(b), sender A and sender B both have data to send, so the “control”, En11 and En22 signals are all high, while the En21 and En12 signals are low. That is, the direction of channel 1 is A → B; the direction of channel 2 is B → A. S11, S21, S31 and



(a) Waveform of dual channel transmission mode

(b) Waveform of single channel transmission mode

(c) Waveform of conversions between single channel and dual channel transmission mode

Fig. 12. Simulation waveform of the self-adaptive bi-directional transmission channel.

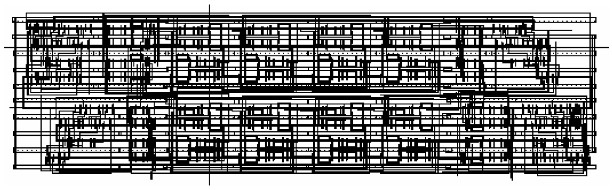


Fig. 13. Layout of the proposed bi-directional transmission channel.

Table 2. Splitting and steering delay of the self-adaptive bi-directional channel under different technology corners (27 ℃, 1.8 V, pre-layout).

	tt	ss	ff
Splitting delay of the input (ps)	688.6	723.05	576.9
Splitting delay of the output (ps)	230.7	285	184
Steering delay (initial) (ps)	616.4	721.48	523.1
Steering delay (working) (ps)	171.08	209.9	133.2

Table 3. Splitting and steering delay of the self-adaptive bi-directional channel under different technology corners (27 ℃, 1.8 V, post-layout).

	tt	ss	ff
Splitting delay of the input (ps)	933.57	1120	777.1
Splitting delay of the output (ps)	326.76	401.48	265.5
Steering delay (initial) (ps)	801.94	954.51	678.46
Steering delay (working) (ps)	196.91	244.11	156.1

S41 are the first signals of MUX and DEMUX in Fig. 5 respectively. It can be found that S11 and S41 go high with a rise in reqA and reqB. This means that data from router A and router B are loaded into channel 1 and channel 2 respectively. At the receiver, S21 and S31 also go high, which indicates that data are taken out of channel 1 and channel 2. In the conversion waveform of Fig. 12(c), the first half of the waveform shows that router A is sending data in dual channel transmission mode. In the latter half of the waveform, router B wants to send data, so the channel has to experience a conversion process. It can be found that En11 and En12 are all high before conversion. That means that the two channels are both occupied by router A. When router B wants to send data, the “control” signal goes high. En12 goes low subsequently, that is, router A is no longer sending data into channel 2. In the meantime, En22 goes high if channel 2 becomes empty, and router B begins to occupy channel 2. After this, the operating mode is just the same as Fig. 12(b). By now, the conversion is over.

A full-custom layout is implemented using SMIC 0.18 μm 1P6M technology, as shown in Fig. 13. The circuit has successfully passed the post-layout simulation and the area of the layout is 69435.14 μm^2.

To test the performance under different technology conditions and the sensitivity to PVT variations, pre-layout simulations and post-layout simulations were done on three technology corners (tt, ss, and ff) at a temperature of 27 ℃. The results are shown in Tables 2 and 3.

The circuit can work properly after layout and shows a preferable performance. This indicates that the results of the post-layout simulation are in accordance with the pre-layout simulation. The splitting delay of the input represents the delay from the router to the first channel unit, while the splitting

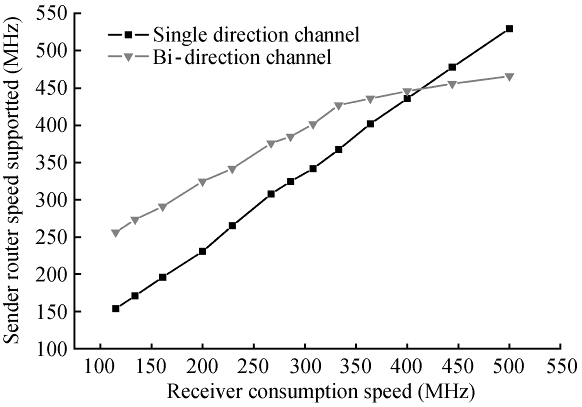


Fig. 14. Maximum supported speed versus receiver working speed (10 stages).

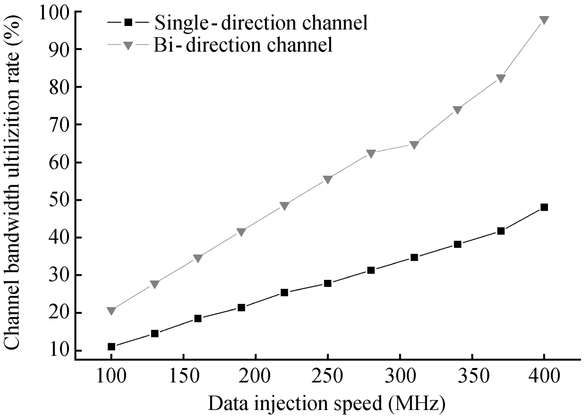


Fig. 15. Channel bandwidth utilization ratio versus injection speed (10 stages).

delay of the output represents the delay from the last channel unit to the router. The initial steering delay is the time needed from the initial state to the working state of channel units, and the working steering delay is the time needed for channel units to change from one direction to the other direction. The reason why working steering delay is smaller than initial steering delay is that steering signals only have to pass MUTEX to output in the proper working state. The delays of post-layout simulation are about 20%–30% higher than pre-layout simulation, which are in normal ranges. It can be seen that the circuit can work properly with a preferable performance under variations of technology conditions. Variations of technology conditions have little impact on the circuit, that is, the circuit has a better robustness.

Here we also compare the maximum supported sender working speed versus receiver working speed of the two channels. When the receiver speed is low, the single channel often becomes "full". This is because the buffer space of the conventional single transmission channel is smaller, and this causes the speed of sender to be much lower. In contrast, the bi-directional transmission channel has more channel buffers, so it can accommodate more data and can support transmission at a larger rate. It can be found that there is an intersection between two lines at about 450 MHz, which indicates that data in the channel can be received more quickly with the increment of receiver working speed. In the latter half part of Fig. 14, the increment of sender output speed of the bi-directional transmission channel is significantly weakened. The reason for this phenomenon is that with increasing receiver speed, the increment speed of the sender is not only limited by the buffer size of the channel. The splitting delay at the entrance of the dual channel makes it difficult for the sender speed to increase any more, thus the output speed stays at a certain value. But in practical applications, this transmission speed can meet most requirements. Moreover, the bigger buffer space in the bi-directional transmission channel can provide a stable and continuous data flow to the receiver router.

Figure 15 shows the relationship between channel bandwidth utilization and injection speed. Here we define the channel bandwidth utilization ratio as:

$$U_{\text{bandwidth}} = \frac{\text{used\_channelunits}}{\text{time} \times \text{total\_channelunits}}.$$

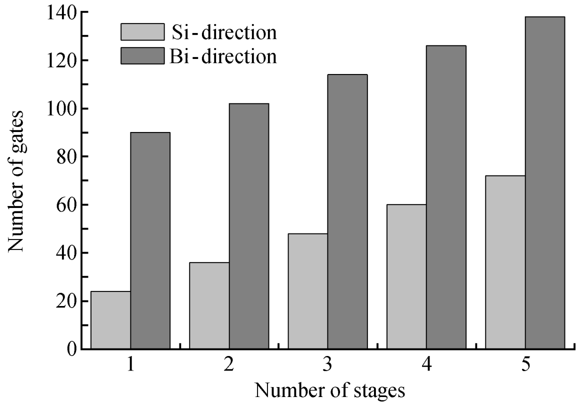


Fig. 16. Area comparison between the two channels.

As can be seen, the bandwidth utilization ratios of the two channels rise with the increment of injection rate, and the bandwidth utilization ratio of the bi-directional transmission channel grows much faster. This indicates that the bi-directional transmission channel can maximize the utilization of channel and connection lines resources, and has a better utilization of limited on-chip resources. In contrast, the conventional transmission channel can only transmit in one direction, so only one transmission channel is used, yet the other channel is still in an idle state. This greatly wastes on-chip asynchronous registers and interconnection line resources. Furthermore, a small buffer space can very easily cause congestion at the sender router. So the bi-directional transmission channel has great advantages in saving on-chip resources.

The areas of the two channels are compared under the same buffer stages, and the number of gates is used to reflect the area. The results are shown in Fig. 16. We find that the bi-directional transmission channel uses about twice the number of gates compared to the single direction channel. This is due to the use of extra circuits implementing full/empty detection and steering. But the bi-directional transmission channel is not necessarily used in every transmission channel, and it is mainly used at paths which are single-direction-traffic dominant, so the increment in area has little influence on the whole system. Compared to the improvement of throughput and channel utilization, sacrificing area is acceptable.

Table 4. Comparison between single direction and bi-directional transmission channels.

	Si-direction [19]	Bi-direction
$N_{\text{interconnect}}$	$2n + 1$	$n + 2$
W_{average} (four stages)	1.164 mW @ 333 MHz	2.68 mW @ 333 MHz
Asyn-register available	n	$2n$
Channel direction	n-in n-out	$2n$-inout
D_{forward}	97.26 ps	110.32 ps

Finally, comparisons between the conventional single direction transmission channel and the bi-directional transmission channel were made, as shown in Table 4. Here we compare the average number of interconnection lines ($N_{\text{interconnect}}$), the average power consumption (W_{average}), the number of asynchronous register available, the channel directions and the forward delay of the channel unit (D_{forward}) of the two channels. It can be seen that the bi-directional transmission channel is slightly inferior to the single direction transmission channel in average power consumption and forward delay of the channel unit. But the bi-directional transmission channel is far superior to the single direction transmission channel in number of interconnection lines and registers; in other words, it can greatly reuse interconnection lines and registers between routers. Moreover, the channels are more flexible in bi-directional transmission channels, and can have four kinds of transmission modes. By contrast, the single direction transmission channel can only realize one of them, so the flexibility and adaptability are very poor, and it cannot effectively use the interconnect lines as well as register resources.

5. Conclusion

This paper proposes a solution to improve the low utilization rate of transmission channels and the large number of interconnection lines between routers in conventional network-on-chips. The new bi-directional transmission channel unit can change its direction according to the state of adjacent routers, thus bi-directional transmission mode can be realized. An asynchronous design scheme guarantees the portability and modularity of the circuits, while quasi delay-insensitive encoding gives the circuit excellent robustness and technology adaptability. Post-layout simulation results have shown that the self-adaptive bi-directional transmission channel is superior to the conventional single direction transmission channel in bandwidth utilization, number of interconnection lines, utilization rate of register resources and channel direction flexibility. It is hoped that this paper can make a valuable contribution to the area of network-on-chips.

References

[1] Dally W J, Towles B, Packets R. Not wires: on-chip interconnection networks. The 38th ACM Design Automation Conf, 2001: 684

[2] Benini L, Micheli G D. Networks on chips: a new SoC paradigm. Computer, 2002, 35(1): 70

[3] Guerrier P, Greiner A. A generic architecture for on-chip packet-switched interconnections. Proc DATE, 2000: 250

[4] Kumar S, Jantsch A, Millberg M, et al. A network on chip architecture and design methodology. Proc IEEE Computer Society Annual Symposium on VLSI, 2002: 105

[5] Bainbridge J, Furber S B. Chain: a delay-insensitive chip area interconnect. IEEE Micro, 2002, 22(5): 16

[6] Lines A. Asynchronous interconnect for synchronous SoC design. IEEE Micro, 2004, 24(1): 32

[7] Akl C J, Bayoumi M A. Wiring-area efficient simultaneous bidirectional point-to-point link for inter-block on-chip signaling. Proc VLSID, 2008:195

[8] Dobkin R R, Ginosar R, Kolodny A. QNoC asynchronous router. Integration, the VLSI Journal, 2009, 42(2): 103

[9] You J, Xu Y, Han H, et al. Performance evaluation of elastic GALS interfaces and network fabric. Electronic Notes in Theoretical Computer Science, 2008, 200(1): 17

[10] Teehan P, Greenstreet M, Lemieux G. A survey and taxonomy of GALS design styles. IEEE Design & Test of Computers, 2007, 24(5): 418

[11] Sheibanyrad A, Greiner A, Miro-Panades I. Multisynchronous and fully asynchronous NoCs for GALS architectures. IEEE Design & Test of Computers, 2008, 25(6): 572

[12] Krstic M, Grass E, Gurkaynak F K, et al. Globally asynchronous, locally synchronous circuits: overview and outlook. IEEE Design & Test of Computers, 2007, 24(5): 430

[13] Sankar R, Kadiyala V, Bonam R, et al. Implementation of static and semi-static versions of a 24+8×8 quad-rail NULL convention multiply and accumulate unit. IEEE Region 5 Tech Conf, 2007: 53

[14] DeMara R F, Kejriwal A, Seeber J R. Feedback techniques for dual-rail self-timed circuits. Proc International Conference on VLSI, 2004: 458

[15] Birtwistle G, Stevens K S. The family of 4-phase latch protocols. Proc ASYNC Asynchronous Circuits and Systems, 2008: 71

[16] Bjerregaard T, Sparso J. A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip. Proc DATE, 2005: 1226

[17] Nimmy J, Ramesh R C, Varadarajan K, et al. RECONNECT: a NoC for polymorphic ASICs using a low overhead single cycle router. Proc ASAP, 2008: 251

[18] Tran X T, Thonnart Y, Durupt J. Design-for-test approach of an asynchronous network-on-chip architecture and its associated test pattern generation and application. IET Computers & Digital Tech., 2009, 3(5): 487

[19] Kakarla S, Al-Assadi W K. Testing of asynchronous NULL conventional logic (NCL) circuits. IEEE Region 5 Conf, 2008: 1