Low Power Asynchronous Wrapper for Network on Chips with Dynamic Frequency Scaling and Power Cut Off

Xuguang Guan, Zhangming Zhu, Duan Zhou, Yintang Yang
Institute of Microelectronics, Xidian University,
Xi’an, 710071, China
guanxuguang_5@126.com, zmyh@263.net, zhouduan3788@126.com, yangyt@xidian.edu.cn
doi : 10.4156/jdcta.vol5.issue5.8

Abstract

Large power consumptions and frequency non-tunable are two shortcomings existing in conventional asynchronous wrapper of network on chips. This paper proposes a low power variable frequency asynchronous wrapper. Through detecting the full/empty state of virtual channels, the wrapper can adjust the working speed of the stoppable clock module, so as to control the working speed of synchronous module, which lowers the network burden as well as power consumption of synchronous module. Meanwhile, power cut off technique is used to lower the leakage power in null cycle of null convention logic. Dual-rail operation mode and null convention logic make the whole circuits strong in robustness. The proposed asynchronous wrapper is implemented based on SMIC 0.18μm standard CMOS technology. Results have shown that sender and receiver wrapper can support synchronous module working at 2.44GHz and 2.7GHz respectively under circumstances of four virtual channels and four bits width. Dynamic power consumption is reduced by 12.8%, while static power consumption is reduced by 69.8%. The proposed wrapper is suitable for low power high robustness interconnection applications due to its high speed low power and delay-insensitiveness.

Keywords: Asynchronous Circuits, Network on Chips, Null Convention Logic, Frequency Scaling, Power Cut Off

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 frequency and larger scale of on chip systems, conventional single clock operation mode faces a lot of challenges, such as poor reusability of modules, power consumption increment of clock tree, large area of clock tree, clock skew and EMI etc. Because of these problems, the complexity of designing very deep submicron integrated circuits is greatly enhanced. So problems brought by clock have become the crucial issues to be solved firstly 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 NoC. Asynchronous circuit, which uses handshake protocols instead of clocks to control circuit behavior, is a potential design manner in future VLSI design area. The 2007 edition of the International Technology Roadmap for Semiconductors (ITRS) [7] shows that asynchronous circuits account for 11% of chip area in 2008, compared to 7% in 2007, and estimates that asynchronous circuits will account for 22% of chip area within the next 5 years, and 30% of chip area within the next 10 years. The advantages of asynchronous circuits include no clock tree, high power efficiency, flexible timing requirement, robust circuit operation, and low noise [8].

Another area of interest in low power research is the leakage power. Dynamic power has been the major factor in CMOS digital circuit power consumption. Recently with the dramatic decrease of characteristic size of the CMOS transistor, leakage power consumption gradually takes a large proportion of the total power consumption [9]. So how to decrease the leakage power consumption becomes the first problem to solve in deep sub-micron integrated circuits designs.
Self-timed wrappers proposed recently in [10] [11] [12] are bundled data wrappers. Although bundled data encoding design can broadly reuse traditional synchronous units, it has unavoidable limitations as delay-matching, weak in anti-EMI, extra control circuits and glitches. And these asynchronous wrappers do not have the function of virtual channel allocation and have no advantages on power consumption. So this paper aims to solve the above problems in two aspects. Firstly, the wrapper proposed can detect the full/empty state of the virtual channels (VC), if the VCs are about to get full, it will regulate the stoppable clock to slow the clock frequency down, so the working frequency of the synchronous processing element (PE) can become lower, thus the power consumption can be lowered. Secondly, the asynchronous transmission circuits are modified with power cut off functions. So the leakage power consumption can be lowered. The proposed wrapper is more preferable to network on chips which are sensitive to power consumption.

2. Asynchronous Interconnect and Null Convention Logic with Power Cut Off

The working scenario of network on chip 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 high concurrent transmission capacity and expansibility [13]. Owing to transmission data in asynchronous on chip networks are reached by handshake signals other than clock, so problems brought by clock can be eliminated, also modularity is greatly enhanced [14]. Although on chip networks have varies of topologies, 2D mesh topology has gained more consideration from designers due to its simplicity, and Fig.1 is its structure. NoC mainly consists of computational processing elements (PE), network interfaces (NI), and routers. The latter two comprise the primary communication architecture.

![Figure 1. Structure diagram of 2D mesh network on chip](image)

While supply noise, electromagnetic coupling and capacitance coupling can cause propagation delay variations of signals, and these effects are often the most noticeable in the system-interconnect of large network on chips. Asynchronous logic and QDI circuits [15] in particular are attractive for their tolerance of such variation and are gaining acceptance in the NoC space for its tolerance which can translate into power-saving and area advantages.

QDI working manner uses dual-rail delay insensitive encoding [16], which guarantees the robustness and transmission efficiency of the circuit. Specific encoding pattern 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 the state is used to separate two data cycles. That is, each transmission cycle must go through a null state before getting into the next transmission cycle. In dual-rail working manner, data also represent request signals, so design of control circuits can be avoided. Furthermore,
dual-rail working manner employs a flow control mode named “back pressure” [17], so its normal working is not susceptible to delay variations. These advantages make it very popular to asynchronous designers recently [18].

Table 1. Dual-rail quasi-delay insensitive encoding

<table>
<thead>
<tr>
<th></th>
<th>D0</th>
<th>D1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data 0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Data 1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Null</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Invalid</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The basic elements in QDI circuits are null convention logics [19], and Fig.2 shows their symbols. Null convention logic gates have n inputs, and m is the threshold value. That means, when m out of these n inputs become high, the output becomes high; only in case of all inputs become low, the output turns to low. Different to the conventional ones, the proposed null convention logic gates support low power sleep function, and the power can be cut off during sleep period. Thus the leakage power can be lowered.

![Symbol of null convention logic gates](a) Conventional logic gate          (b) Proposed logic gates with power cut off

Figure 2. Symbol of null convention logic gates

By implementing power cut off into asynchronous QDI circuits, the primary drawbacks of synchronous power cut off technique can be eliminated, namely, sleep signal generation, storage element data loss during sleep mode, and sleep transistor sizing. Here we adopt early completion technique [20] to enhance the performance of the QDI circuits. And this point will be discussed in the following section.

Here we give the specific structure of null convention logic gate with power cut off, only the structure of TH22 gate is given for an example, as shown in Fig.3.

![Proposed TH22 gates structure with power cut off](a) Conventional logic gate          (b) Proposed logic gates with power cut off

Figure 3. Proposed TH22 gates structure with power cut off

Transistors in dashed circles are high threshold transistors, others are low threshold transistors. When the TH22 gate goes into normal working manner, sleep signal is low, and the circuit has the same function with the original one. But the difference is that when TH22 gate ends its working states, it doesn’t have to wait for signals a and b to go low. Instead, sleep signal realizes this function. This working manner has two advantages. Firstly, the working speed can be enhanced due to sleep signal, and sleep signal is generated by early completion module. Secondly, this structure can reduce leakage power by disconnecting the power supply from the circuit during the sleep mode while maintaining high performance in the active mode. Low threshold transistors offer fast speed but cost high leakage. In contrast, high threshold transistors suffer from reduced speed, but leak less current when turned off.
So the combination of these two kinds of transistors can preserve proper performance as well as suppress the subthreshold leakage.

Owing to the unique advantages of QDI circuits, it is very suitable to use QDI circuits to construct the interconnection circuits in NoC. So how to combine the QDI circuit with synchronous circuits in NoC becomes the key point to realize asynchronous network on chips. Fortunately, asynchronous wrapper can realize this function. We have proposed the first QDI asynchronous wrapper in Ref.[21], but it doesn’t support virtual channel allocation, that is, it only can be used in NoCs which don’t have virtual channels. Furthermore, it does not support dynamic frequency scaling and power cut off, so the dynamic and static power could be large. The QDI asynchronous wrapper proposed can support all the functions mentioned above, and has the advantage of low power consumption.

3. Specific Implementation of Low Power Asynchronous Wrapper

The whole system can be generally divided into following three parts: QDI asynchronous wrapper, stoppable clock generators and virtual channels of the router, as shown in Fig.4. Wrapper R is responsible for receiving data from asynchronous virtual channels, while Wrapper S is responsible for sending data to asynchronous virtual channels. Stop R and Stop S signals are sent respectively by Wrapper R and Wrapper S, and their functions are stopping the clock generator module so as to keep the data from or to the PE stable. DFS control block is used to control the output frequency of clock generators according to full/empty status of input VC channels of the router. The advantages of dynamic frequency scaling lie in two aspects. Firstly, clock generator can lower its output frequency if one or more VCs become full, so the working frequency of PE is lowered, thus the power consumption of the synchronous module (PE) can be decreased. Secondly, if one or more VCs become full, it indicates the network gradually becomes busy, that is, the network traffic burden is heavy, so to lower the working frequency of PE can make the data flow be sent into the network much less, which can play a positive role in easing the network traffic burden. It should be noted that only the sender wrapper has the DFS control, and F/E signals are sent by VCs from input VC channels. Receiver wrapper doesn’t need a DFS control, because what it needs to do is receiving data from the network as soon as possible so as to alleviate the burden of network.

3.1. Sender asynchronous wrapper

Specific structure of sender QDI asynchronous wrapper is shown in Fig.5. It can convert output data of the synchronous module to dual-rail QDI output data and allocate them into virtual channels automatically. For simplicity, we merely draw 1 bit conversion here, and more bits conversion is easy to reach through adding bit width at virtual channels and D flip-flops. Stoppable and controllable clock module is used to stop the clock when synchronous PE has data to send, and T[0] to T[4] are four

Figure 4. Diagram of the interface between PE and network router
regulation signals which are sent by DFS control module. While DFS control module can change its outputs according to full/empty status of VCs. VC allocator controls the NCL-demux to distribute the incoming data into one of the VCs. And VCs send back acknowledgement signals to indicate that data have been successfully received.

Specific working procedure of sender QDI asynchronous wrapper is as follows. At the beginning, C elements and threshold gates are reset. After reset, signal c, stretch, D0 and D1 are low while signal b is high. Full 1, Full 2, Full 3 and Full 4 are low, which indicate VCs are all empty, so stoppable and controllable clock module can work in full speed mode. And now, the synchronous PE module begins to work properly with wrapper waiting for write signal. When sender wants to send data, write signal goes high. As the output of #3 AND gate is low, the output of #1 C element, c, keeps low and output of #1 AND gate, a, goes high. Signal b keeps high since ack1 to ack4 are high, so #4 AND gate is high. And now inputs to #3 C element are all high, so signal stretch goes high and controls stoppable and controllable clock module to stop the clock. At the same time, signal stretch controls D flip-flops to sample the synchronous output data. And VC allocator controls NCL-demux to send the sampled data into one of the VCs. If VC has successfully received the data, it will send back an acknowledgement signal, that is, the corresponding ack signal will go low. And now #4 AND gate goes low, and causes D flip-flops to reset. So null period can be sent into the corresponding VC, by now, the allocation period is over. On the other hand, low in #4 AND gate will cause the output of #2 C element, b, to go low. This will cause the output of #3 AND gate, rs, to go high. It can be found that rs port is the reset port of #3 C element. And high in rs signal will cause #3 C element to reset, so stretch signal goes low. Clock is released when signal stretch going low, which makes synchronous module go into work again. By now, a whole sender cycle has completed. It should be noted that, if one or more VCs become full, that is, one or more full signals become high, and then DFS control module can regulate the stoppable and controllable clock to lower its output clock frequency. Thus, the power consumption of synchronous PE can be lowered.

Figure 5. Specific structure of sender QDI asynchronous wrapper
Figure 6. NCL-demux with sleep function

Figure 6 shows the specific structure of NCL-demux with sleep function. It has several acknowledgement signals; each acknowledgement signal is applied to each input group. Here we take 1 bit NCL-demux module for example. Initially, all the TH33 gates are in sleep mode to reduce the static power consumption. When data arrive, all the TH33 gates begin to work. If S1 is high, then data D0, D1 can go through threshold gate and get to Out0_VC1 and Out1_VC1. The following register of VC1 will send back an ack1 signal if data are successfully received. And this message also causes the sleep generator to send the sleep signal again, so as to make all TH33 gates go into sleep mode. Out0_VC1 and Out1_VC1 signals go to zero as soon as sleep signal arrives, which equals to send null signals to the next stage. This working manner can make the TH33 gate go into null cycle ahead in advance, and don’t have to wait for all selection signals to go low, which saves the time going into null cycle for TH33 gate.

Figure 7. Structure of sleep generator module for NCL-demux

The structure of sleep generator module is shown in Fig.7. It can make threshold gates go into sleep mode in null cycles. At the beginning, VC1~VC4 are all empty, so ack1~ack4 are all high, which indicates ackall signal is high. And now data do not yet come, that is, D0 and D1 are low, so the output of inverted C element, sleep signal, is high. And now the NCL-demux module is in sleep mode. When data come, one of D0 and D1 will rise, so OR gate rises. And now the inverted C element reaches its threshold, so sleep signal goes low, which releases the sleep mode of NCL-demux module. When data have been successfully transmitted into the virtual channel, one of the ack goes low, so ackall goes low, while D0 and D1 go low due to reset in D flip-flops. And now the inverted C element reaches its reset condition, that is, the output of the inverted C element, sleep, goes high again. And the NCL-demux goes into sleep mode, waiting for next data.
The virtual channel with sleep function and full/empty detection is shown in Fig.8. Here we also take 1 bit virtual channel for example. And Fig.8 is first half of the virtual channel which joints with the NCL-demux module. sl is the sleep port of TH22 gate. It can be found that every stage has a sleep generator to control the sleep working mode of present stage. The virtual channel stages use a completion detection mode named "early completion" [20]. Early Completion utilizes the inputs of registeri-1 along with the ACKi request to registeri-1 to generate the acknowledgement signal to registeri-2. Now this acknowledgement signal to registeri-2 can be used to sleep the TH22 gates in stage-1 without compromising quasi delay-insensitivity, since stage-1 will only be put to sleep when both its inputs are null and it is requesting null. Full signal is generated by four ACK signals which are adjacent to each other. When virtual channel is full, the status of stage i+1 to stage i-2 is empty, full, empty, full, so full signal can be easily generated. And this full signal will be sent to DFS control signal to control the clock generator to lower its output frequency.

DFS control block is the most important module in sender asynchronous wrapper. Its specific implementation is shown in Fig.9.
The DFS control block has four inputs and five outputs, so it has five stages regulation function. And the specific function of DFS control module is shown in Table 2. When four full signals are all low, which indicates the virtual channels are all empty, a0 goes high. And now the stoppable and controllable clock generator is working in full speed. When one of the full signals is high, it indicates one virtual channel is full, and then a2 goes high. Now the stoppable and controllable clock generator is working in 2-level clock speed. If four full signals are all high, which indicates the four virtual channels are all full, then a4 goes high. At this time the stoppable and controllable clock generator is working in the lowest frequency so as to reduce the working frequency of PE module, thus the sending rate of the PE is lowered.

| Table 2. Function table of DFS control block |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                | a0   | a1   | a2   | a3   | a4   |
| All empty      | 1    | 0    | 0    | 0    | 0    |
| 1 VC full      | 0    | 1    | 0    | 0    | 0    |
| 2 VC full      | 0    | 0    | 1    | 0    | 0    |
| 3 VC full      | 0    | 0    | 0    | 1    | 0    |
| All full       | 0    | 0    | 0    | 0    | 1    |

### 3.2. Receiver asynchronous wrapper

Specific structure of receiver QDI asynchronous wrapper is shown in Fig.10. It can convert output data of the virtual channels to synchronous input data and guarantee them to be sampled by synchronous PE module. For simplicity, we merely draw 1 bit conversion here, and more bits conversion is easy to realize through adding bit width at virtual channels and D flip-flops. As has been stated before, the receiver asynchronous wrapper doesn’t need dynamic frequency scaling, and what it needs to do is receiving data as soon as possible. The receiver asynchronous wrapper also has stretch signal, which is used to stop the clock generator when data sampling, preventing the occurrence of metastability.

![Specific structure of receiver asynchronous wrapper](image)

**Figure 10.** Specific structure of receiver asynchronous wrapper

Specific working procedure of circuit in Fig.10 is as follows. At the beginning, C elements and threshold gates begin to reset. After reset, signal a, stretch, D, e1 and e3 are low. Synchronous module begins to work properly, and receiver wrapper waits for read signal. When synchronous module wants to receive a new data, signal read will go high, and the output of asymmetric TH23, signal stretch, begins to go high with signal a being low and signal b being high. Stoppable clock module is stopped and synchronous module is ready to receive new data. If there is no data in virtual channels, the output of #4 NOR gate will be high with signal D remaining low. #2 C element waits for
the arrival of input data, that is, it waits for signal req to go low. When data come, one of c_t and c_f goes high, and now the output of #4 NOR gate, req signal, begins to go low. #2 C element reaches its threshold with output signal D going high, which controls D flip-flops to sample input data. The inputs of #2 NOR gate are reverse signals, thus its output goes low. This makes the output of #3 NOR gate, signal Ack, go low, and requests for null signal go into NCL-mux module. It is notable that only data should be sampled other than null, namely avoiding D flip-flop sampling at null period. The output of #4 NOR gate goes high after null signal arrives at c_t and c_f, which makes signal b high. So stretch signal goes low and clock generator is released due to the reset of asymmetry TH23 gate. Low in signal stretch would further lead to low in signal D, signal read, and high in signal Ack. Now the circuits come back to the initial state, and are ready to move on to the next working cycle.

Specific implementation of NCL-mux is shown in Fig.11. NCL-mux is the module connecting with VC channels and the asynchronous receiver wrapper. Data from which VC channels can get through the NCL-mux is decided by VC allocator. D signal is from #2 C element in Fig.10. sleep1~sleep4 signals are generated by registers in VC1~VC4 using early completion operation mode. ackr1~ackr4 are acknowledgement signals to the next stage registers of corresponding virtual channels. When data reach corresponding input port (A0, A1, B0, B1, C0, C1, D0, and D1), sleep signals go low, and TH33 gates are ready to go into working mode. If now virtual channel 1 is selected, S1 is high. Due to two inputs of the inverted C element are low, its output is high. So data from VC1 now can get through TH33 gate, and be outputted by OR gate with four inputs. And now the NCL-mux is waiting for D flip-flops in Fig.10 to begin sampling. D signal goes high and the sample process starts, and at the same time, high in D signal makes ackr1 signal in NCL-mux go low, which informs the next stage register in VC1 that the data have been sampled successfully. The next stage register will send out a sleep1 signal upon receiving ackr1 signal, which makes TH33 gates go into sleep mode again. By now, a work cycle of the NCL-mux is complete.

Figure 11. NCL-mux with sleep function
4. Simulation Results and Analysis

The whole circuits of sender wrapper and receiver wrapper are implemented using SMIC 0.18μm technology. The numbers of virtual channels of sender and receiver are both four. And the virtual channel width is 4 bits. Fig.12 is part of the SPICE simulation waveforms.

![Figure 12. Simulation waveform of (a) sender asynchronous wrapper (b) receiver asynchronous wrapper](image)

In Fig.12 (a), wr signal is the write signal of the synchronous PE, and rs signal is the reset signal of #3 C element in Fig.5. S1, S2, S3 and S4 are four selection signals of VC allocator. Ackall is the reset signal of D flip-flops. It can be found that one write signal arrives at each time; stretch signal goes high to stop the clock in order to guarantee the sampling process stable. Data are sent to virtual channels in the order of VC1 to VC4. When data have been successfully received by virtual channels, rs signal goes high to reset #3 C element, thus the stretch signal can go low, and the clock generator can resume to working status. Up to now, a cycle of the sender asynchronous wrapper is complete. It also can be found that with data sent into virtual channels, the virtual channels are gradually becoming full, so the clock frequency becomes lower to reduce sending speed of the synchronous PE. Thus, the dynamic frequency scaling can be realized. In Fig.12 (b), rd signal is the read signal of the synchronous PE; D is the trigger signal of D flip-flops, Ackr1~Ackr4 are four acknowledgement signals from four virtual channels. It can be found that one read signal arrives at each time; stretch signal goes high to stop the clock in order to guarantee the reading process stable. Data are read out following the sequence of VC1 to VC4, which has the same sequence with the sender asynchronous wrapper. When data have successfully been received by synchronous PE, req goes high and ack goes low, which make the asymmetric TH23 gate reset, thus the stretch signal goes low, and the clock generator is released. Up to now, a cycle of the receiver asynchronous wrapper is complete.

Here we also test the frequency regulating scope of stoppable and controllable clock generator, as shown in Table 3.

| Table 3. Frequency regulating range of stoppable and controllable clock generation module |
|-----------------------------------------------|----------------|-------------|----------------|----------------|----------------|
| Frequency | 1 stage | 2 stage | 3 stage | 4 stage | 5 stage |
| 1.43GHz | 1.02GHz | 787.4MHz | 555MHz | 362MHz |

It can be found that the stoppable and controllable clock generator has a range of five stage, and the frequency coverage is from 362MHz to 1.43GHz. The frequency regulating range is so large that it can effectively lower the power consumption and data sending rate of the synchronous PE.

To test the performance and robustness of the wrapper proposed, simulations were made under three technology models (tt, ss, ff). Results are been shown in Table 4. Here we define delayF as the delay from rise of read/write signal to rise of signal stretch, and define delayALL as the delay from rise of read/write signal to fall of signal stretch. Pdynamic represents the average dynamic power dissipation.
of the circuit, while $P_{\text{static}}$ stands for average static power dissipation of the circuit. It should be noted that $\text{delay}_\text{ALL}$ comprises two aspects: conversion delay and allocation delay, while the power consumption is the total power consumed by the wrapper and virtual channels.

### Table 4. Performance test results of sender and receiver asynchronous wrapper with virtual channels

<table>
<thead>
<tr>
<th></th>
<th>$\text{delay}_F$</th>
<th>$\text{delay}_\text{ALL}$</th>
<th>$P_{\text{dynamic}}$</th>
<th>$P_{\text{static}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>sender</td>
<td>$303\text{ps}$</td>
<td>$245.15\text{ps}$</td>
<td>$200.6\text{ps}$</td>
<td>$280.3\text{ps}$</td>
</tr>
<tr>
<td></td>
<td>$4.13\text{ns}$</td>
<td>$3.39\text{ns}$</td>
<td>$2.83\text{ns}$</td>
<td>$1.67\text{ns}$</td>
</tr>
<tr>
<td></td>
<td>$1.986\text{mW}$</td>
<td>$2.12\text{mW}$</td>
<td>$2.407\text{mW}$</td>
<td>$2.2\text{mW}$</td>
</tr>
<tr>
<td></td>
<td>$2.08\mu\text{W}$</td>
<td>$1.32\mu\text{W}$</td>
<td>$6.44\mu\text{W}$</td>
<td>$2.56\mu\text{W}$</td>
</tr>
<tr>
<td>receiver</td>
<td>$245.15\text{ps}$</td>
<td>$3.39\text{ns}$</td>
<td>$2.83\text{ns}$</td>
<td>$1.67\text{ns}$</td>
</tr>
<tr>
<td></td>
<td>$223.8\text{ps}$</td>
<td>$223.8\text{ps}$</td>
<td>$2.12\text{mW}$</td>
<td>$3.72\text{mW}$</td>
</tr>
<tr>
<td></td>
<td>$187.3\text{ps}$</td>
<td>$187.3\text{ps}$</td>
<td>$2.407\text{mW}$</td>
<td>$5.274\text{mW}$</td>
</tr>
<tr>
<td></td>
<td>$6.44\mu\text{W}$</td>
<td>$1.32\mu\text{W}$</td>
<td>$6.44\mu\text{W}$</td>
<td>$1.528\mu\text{W}$</td>
</tr>
</tbody>
</table>

As can be seen from Table 4, the circuit can properly work with preferable performance under different variations of technology. Variations of technology have little impact on the circuit, namely the circuit has a better robustness.

At last we compare the proposed QDI asynchronous wrapper with the conventional one on supported synchronous module working frequency, dynamic power consumption, static power consumption as well as some functions. Specific results are shown in Table 5.

### Table 5. Performance comparisons with conventional QDI asynchronous wrapper

<table>
<thead>
<tr>
<th></th>
<th>$\text{Sender Max. Freq. supported}$</th>
<th>$\text{Receiver Max. Freq. supported}$</th>
<th>$P_{\text{dynamic _ALL}}$</th>
<th>$P_{\text{static _ALL}}$</th>
<th>$\text{Freq. tunable?}$</th>
<th>$\text{Power cutoff?}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>$2.44\text{GHz}$</td>
<td>$2.7\text{GHz}$</td>
<td>$5.84\text{mW \text{(25% full)}}$</td>
<td>$2.848\text{uW}$</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Ref.[21]</td>
<td>$2.87\text{GHz}$</td>
<td>$2.59\text{GHz}$</td>
<td>$6.7\text{mW}$</td>
<td>$9.451\text{uW}$</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

As can be seen from Table 5, the supported frequency of proposed QDI asynchronous wrapper is similar with the conventional one. But the proposed wrapper is more effective in power consumption and frequency scalable. It should be noted that the power consumptions here include both the power consumptions of wrapper and virtual channels. The dynamic power consumption of proposed wrapper is $87.2\%$ of conventional one under the situation of $25\%$ virtual channels are full. While the static power consumption of proposed wrapper is $30.1\%$ of conventional one. Thus the dynamic frequency scaling and power cutoff can save a lot of power both in dynamic and static mode. Furthermore, the leakage power in deep sub-micron is becoming larger and larger with the degradation of characteristic size, so it is foreseeable that the power cut off technique can save more power in much smaller characteristic size.

### 5. Conclusions

This paper proposes a newly developed asynchronous wrapper with dynamic frequency scaling and power cutoff. Through detecting full/empty status of the virtual channels, the sender asynchronous wrapper can adjust its working speed, so as to reduce the outgoing data rate which can lower the network burden. For virtual channels, power cutoff technique is used to reduce the static power consumption, null convention logics can go into sleep mode when they are inactive, so as to lower the leakage current. Results have shown that the proposed asynchronous wrapper have greatly reduced both dynamic and static power consumption. Hope this paper can make a valuable contribution to the area of low-power NoC.

In the future, we will focus on DVFS control technology in more details, and flexible low power design scheme, so that the power of PE can be completely turned off, thus can further lower the power consumption.
6. Acknowledgement

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

7. References