Adding Slow-Silent Virtual Channels for Low-Power On-Chip Networks

Hiroki Matsutani (Keio Univ, Japan)
Michihiro Koibuchi (NII, Japan)
Daihan Wang (Keio Univ, Japan)
Hideharu Amano (Keio Univ, Japan)
I am very sorry...

- My flight was canceled on April 6.
- I was waiting for rebooking at airport for seven hours, but I couldn’t get a ticket. I got a fever.
- I arrived at Newcastle on April 7.
- I couldn’t find my baggage; I wore only a shirt.
- My hotel reservation was canceled w/o asking; I didn’t have a place to sleep...
- I went to another hotel to book a room in my shirt sleeves in the rain. The fever was gone up.
- Ms. Jerder kindly did her presentation on Apr 8.
- I would like thank her and ASYNC/NOCS program committee.
Adding Slow-Silent Virtual Channels for Low-Power On-Chip Networks

Hiroki Matsutani (Keio Univ, Japan)
Michihiro Koibuchi (NII, Japan)
Daihan Wang (Keio Univ, Japan)
Hideharu Amano (Keio Univ, Japan)
Introduction: Area and power

- Due to the finger process technology,
  - Area constraint is relaxed
  - But power density becomes more serious
- Adding extra hardware resources (e.g., VCs)
  - We can get a performance margin; so
  - We can reduce voltage and frequency to reduce power

Issues to be tackled in this presentation

- Adding extra hardware increases the leakage power
- How much resource is required to minimize total power
Outline: Slow-silent virtual channels

- Network-on-Chip (NoC)
- On-Chip Router
  - Architecture and its power consumption

- Slow-silent virtual channels
  - Voltage and frequency scaling
  - Run-time power gating of virtual channels
  - Adaptive VC activation

- Evaluations (1VC, 2VC, 3VC, and 4VC)
  - Throughput
  - Power consumption (with PG & voltage freq scaling)
  - How many VCs are required to minimize power
Network-on-Chip (NoC)

• Processor core
  - Largest component
  - Various low-power techniques are used
    e.g., Standby current 11μA [Ishikawa, IEICE’05]

• On-chip router
  - Area is not so large
  - Always preparing (active) for packet injection

The next slides show “Router architecture” and “Its power”
On-Chip Router: Architecture

- 5-input 5-output router (data width is 64-bit)

Each physical channel has 2 VCs
Each VC has a FIFO buffer (4 x 64 bits)

HW amount is 34 kilo gates and 64% of area is used for FIFO
On-Chip Router: Pipeline

- A header flit goes through a router in 3 cycles
  - RC (Routing computation)
  - VSA (Virtual channel / Switch allocation)
  - ST (Switch traversal)

E.g., Packet transfer from router A to C

A packet consists of a header and 3 data flits.
On-Chip Router: Power consumption

- Place-and-routed with 90nm CMOS
- Post layout simulation at 200MHz

Packet switching power is large → Voltage freq scaling

Power consumption of a router when *n* ports are used [mW]

A router consumes more power as the router processes more packets
On-Chip Router: Power consumption

Power consumption when *no* port is used → standby power

- Leakage (55.0%)
- Dynamic (45.0%)
- Channels (49.4%)

Leakage of channel buf is the largest → Runtime power gating

Packet switching power is large → Voltage freq scaling
Outline: Slow-silent virtual channels

- Network-on-Chip (NoC)
- On-Chip Router
  - Architecture and its power consumption

- Slow-silent virtual channels
  - Voltage and frequency scaling
  - Run-time power gating of virtual channels
  - Adaptive VC activation

- Evaluations (1VC, 2VC, 3VC, and 4VC)
  - Throughput
  - Power consumption (with PG & voltage freq scaling)
  - How many VCs are required to minimize power
Slow-Silent Virtual Channels

- Adding extra VCs
  - Performance improves
- Voltage & frequency scaling (VFS)
  - Set the reduced voltage and frequency
  - In response to the performance margin
  \[ f \propto \frac{(V - V_{th})^\alpha}{CV} \]
  \[ P_{\text{switching}} = a \cdot C \cdot f \cdot V^2 \]

- Problem
  - Adding extra VCs increases leakage power
  - It may overwhelm VFS

We focus on run-time power gating of VCs to reduce leakage
Power Gating of virtual channels

- Run-time power gating of virtual channels
  - No packets in a VC $\rightarrow$ Sleep (turn off the power supply)
  - Packet arrives at the VC $\rightarrow$ Wakeup (turn on the power)
Power Gating of virtual channels

- Run-time power gating of virtual channels
  - No packets in a VC → Sleep (turn off the power supply)
  - Packet arrives at the VC → Wakeup (turn on the power)

Link shutdown has been studied for on- & off-chip networks, but prior work uses SRAM buffers [Chen, ISLPED’03] [Soteriou, TPDS’07]

→ We use small registered FIFOs for light-weight NoC routers
Power Gating: Various overheads

- **Area overhead**
  - Power switches

- **Performance overhead**
  - Wakeup delay
  - Pipeline stall is caused
  
  → Frequent on/off should be avoided

- **Power overhead**
  - Driving power switches
  - Short sleeps adversely increases dynamic power
  
  → Frequent on/off should be avoided
Power Gating: Various overheads

- **Area overhead**
  - Power switches

- **Performance overhead**
  - Wakeup delay
  - Pipeline stall is caused
    \(\rightarrow\) Frequent on/off should be avoided

- **Power overhead**
  - Driving power switches
  - Short sleeps adversely increases dynamic power
    \(\rightarrow\) Frequent on/off should be avoided

Control that gradually activates VCs in response to workload
Power Gating: VC activation policy

- Virtual channel (VC) level power gating

- Virtual-channel selection:
  - All packets use VC#0 when they are injected to NoC
  - VC number is increased when the packet conflicts

Only VC#0 is used if workload is low
**Power Gating: VC activation policy**

- Virtual channel (VC) level power gating

- Virtual-channel selection:
  - All packets use VC#0 when they are injected to NoC
  - VC number is increased when the packet conflicts

All VCs are activated if workload is high

High peak performance of VCs with the least leakage power
Power Gating: Routing design

- A virtual-channel layer
  - A virtual network consisting of VCs with the same VC#
- Deadlock-freedom [Duato, TPDS'93] [Koibuchi, ICPP'03]
  - Moving upper to lower layers VC#0 → VC#1 → VC#2 → VC#3
  - Only bottom layer must guarantee deadlock-freedom

All VC layers except for the bottom can employ any routing, as far as the bottom guarantees deadlock-free by itself.
Outline: Slow-silent virtual channels

- Network-on-Chip (NoC)
- On-Chip Router
  - Architecture and its power consumption
- Slow-silent virtual channels
  - Voltage and frequency scaling
  - Run-time power gating of virtual channels
  - Adaptive VC activation
- Evaluations (1VC, 2VC, 3VC, and 4VC)
  - Throughput
  - Power consumption (with PG & voltage freq scaling)
  - How many VCs are required to minimize power
Evaluations of slow-silent VCs

- Preliminary
  - Leakage modeling of PG
  - Breakeven point of PG

- Evaluation items
  - Original throughput
  - Power consumption w/o PG and VFS
  - Power consumption w/ PG and VFS

- Which is the best?
  - 1VC, 2VC, 3VC, and 4VC

- Process technology
  - ASPLA 90nm CMOS
  - 1.00V (baseline)

- Simulation parameters
  - Topology: 2-D Mesh (8x8)
  - Routing: DOR (XY routing)
  - Buffer size: 4-flit (WH switching)
  - # of VCs: 1VC, 2VC, 3VC, 4VC
  - Latency: 3-cycle per 1-hop

- Traffic patterns
  - Uniform + NPB traces (BT, SP, CG, MG, IS)
Preliminary: Leakage power modeling

- **Power gating model** [Hu,ISLPED’04]
  - $E_{\text{overhead}}$: Power consumed for turning PS on/off
  - $E_{\text{saved}}$: Leakage power saving for an $N$-cycle sleep

How many cycles are required to sleep for compensating $E_{\text{overhead}}$?

We calculate the breakeven point of PG based on the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>1.0 V</td>
</tr>
<tr>
<td>Switching factor</td>
<td>0.12</td>
</tr>
<tr>
<td>Leakage power</td>
<td>52 uW</td>
</tr>
<tr>
<td>Dynamic power (200MHz)</td>
<td>78 uW</td>
</tr>
<tr>
<td>Dynamic power (500MHz)</td>
<td>194 uW</td>
</tr>
<tr>
<td>Power switch size ratio</td>
<td>0.1</td>
</tr>
<tr>
<td>Power switch cap ratio</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Based on the post layout simulation of on-chip router (90nm CMOS)
Preliminary: Leakage power modeling

- Power gating model [Hu, ISLPED’04]
  - $E_{\text{overhead}}$: Power consumed for turning PS on/off
  - $E_{\text{saved}}$: Leakage power saving for N-cycle sleep

How many cycles are required to sleep for compensating $E_{\text{overhead}}$?

Power consumption is reduced as sleep duration becomes long.

Breakeven point is 7 cycle (200MHz)
Breakeven point is 16 cycles (500MHz)
Preliminary: Leakage power modeling

- **Power gating model** [Hu, ISLPED’04]
  - $E_{\text{overhead}}$: Power consumed for turning PS on/off
  - $E_{\text{saved}}$: Leakage power saving for N-cycle sleep

How many cycles are required to sleep for compensating $E_{\text{overhead}}$?

Power consumption is reduced as sleep duration becomes long.

Breakeven point is...
- PG(200MHz): 7 cycles
- PG(300MHz): 10 cycles
- PG(400MHz): 13 cycles
- PG(500MHz): 16 cycles

<table>
<thead>
<tr>
<th>Power consumption [mW]</th>
<th>Sleep duration (N) [cycle]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>5</td>
</tr>
<tr>
<td>0.15</td>
<td>10</td>
</tr>
<tr>
<td>0.1</td>
<td>15</td>
</tr>
<tr>
<td>0.05</td>
<td>20</td>
</tr>
</tbody>
</table>

No power gating (PG)  
PG router (200MHz)  
PG router (300MHz)  
PG router (400MHz)  
PG router (500MHz)
Evaluations of slow-silent VCs

- Preliminary
  - Leakage modeling of PG
  - Breakeven point of PG

- Evaluation items
  - Original throughput
  - Power consumption w/o PG and VFS
  - Power consumption w/ PG and VFS

- Which is the best?
  - 1VC, 2VC, 3VC, and 4VC

- Process technology
  - ASPLA 90nm CMOS
  - 1.00V (baseline)

- Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>2-D Mesh (8x8)</td>
</tr>
<tr>
<td>Routing</td>
<td>DOR (XY routing)</td>
</tr>
<tr>
<td>Buffer size</td>
<td>4-flit(WH switching)</td>
</tr>
<tr>
<td># of VCs</td>
<td>1VC, 2VC, 3VC, 4VC</td>
</tr>
<tr>
<td>Latency</td>
<td>3-cycle per 1-hop</td>
</tr>
</tbody>
</table>

- Traffic patterns
  - Uniform + NPB traces (BT, SP, CG, MG, IS)
Evaluations: Uniform (64-core) 1/4
Evaluations: Uniform (64-core) 2/4

Power (without PG & VFS)
Evaluations: Uniform (64-core) 3/4

Static voltage and frequency scaling

<table>
<thead>
<tr>
<th></th>
<th>Freq [MHz]</th>
<th>Voltage [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1VC</td>
<td>500.0</td>
<td>1.00</td>
</tr>
<tr>
<td>2VC</td>
<td>301.8</td>
<td>0.77</td>
</tr>
<tr>
<td>3VC</td>
<td>238.8</td>
<td>0.70</td>
</tr>
<tr>
<td>4VC</td>
<td>224.8</td>
<td>0.68</td>
</tr>
</tbody>
</table>

1) We re-characterized low-voltage libraries (0.68-0.77V) by Cadence SignalStrom

2) We confirm our design works at these reduced voltages
### Evaluations: Uniform (64-core) 4/4

<table>
<thead>
<tr>
<th>Static voltage and frequency scaling</th>
<th>Freq [MHz]</th>
<th>Voltage [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1VC</td>
<td>500.0</td>
<td>1.00</td>
</tr>
<tr>
<td>2VC</td>
<td>301.8</td>
<td>0.77</td>
</tr>
<tr>
<td>3VC</td>
<td>238.8</td>
<td>0.70</td>
</tr>
<tr>
<td>4VC</td>
<td>224.8</td>
<td>0.68</td>
</tr>
</tbody>
</table>

The same results can be seen in all-to-all traffics (e.g., IS)
Evaluations: BT traffic (64-core) 1/4

Performance improvements of 3-VC and 4-VC are small

Original throughput
Evaluations: BT traffic (64-core) 2/4

Performance improvements of 3-VC and 4-VC are small

Power (without PG & VFS)
Evaluations: BT traffic (64-core) 3/4

Static voltage and frequency scaling

<table>
<thead>
<tr>
<th></th>
<th>Freq [MHz]</th>
<th>Voltage [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1VC</td>
<td>500.0</td>
<td>1.00</td>
</tr>
<tr>
<td>2VC</td>
<td>350.1</td>
<td>0.82</td>
</tr>
<tr>
<td>3VC</td>
<td>346.2</td>
<td>0.82</td>
</tr>
<tr>
<td>4VC</td>
<td>346.1</td>
<td>0.82</td>
</tr>
</tbody>
</table>

1) We re-characterized the low-voltage library (0.82V) by Cadence SignalStorm
2) We confirm our design works at this reduced voltage

Almost the same
**Evaluations: BT traffic (64-core) 4/4**

- **Static voltage and frequency scaling**

<table>
<thead>
<tr>
<th></th>
<th>Freq [MHz]</th>
<th>Voltage [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1VC</td>
<td>500.0</td>
<td>1.00</td>
</tr>
<tr>
<td>2VC</td>
<td>350.1</td>
<td>0.82</td>
</tr>
<tr>
<td>3VC</td>
<td>346.2</td>
<td>0.82</td>
</tr>
<tr>
<td>4VC</td>
<td>346.1</td>
<td>0.82</td>
</tr>
</tbody>
</table>

- 2-VC is the lowest leakage

The same result can be seen in neighboring traffics (e.g., SP)
How many VCs are best for LP?

→ It depends on the traffic pattern of application

- **All-to-all traffic**
  - Uniform, IS traffic
  - 3 or 4 VCs are better

- **Neighboring traffic**
  - BT, SP traffic
  - 2 VCs are enough

It depends on the traffic pattern of application.
Summary: Slow-silent virtual channels

- Slow-silent virtual channels
  - Adding extra VCs → Performance margin is available
  - We can reduce the freq and voltage
  - But adding extra VCs increases leakage power ...

- Run-time power gating of VCs
  - Adaptive VC activation

- How many VCs are required for minimizing power?
  - It depends on the traffic pattern of application
  - All-to-all traffic: 3 or 4 VCs are better
  - Neighboring traffic: 2 VCs are enough
Future work: Slow-silent fat trees

- Very “FAT” trees
  - Adding more trees & voltage frequency scaling
  - Run-time power gating
- There are a lot of types of Fat trees

• How many trees are required to minimize power?
Thank you for your attention
Backup sides
Wakeup delay: Performance impact

- Wakeup delays in literatures
  - ALU: 2 cycle [Tschanz, JSSC’03]
  - FPMAC in Intel’s 80-tile chip: 6 cycle [Vangal, ISSCC’07]

- Performance impact of wakeup delay (naïve mode)
Look-Ahead Sleep Control

- Look-ahead sleep control
  - To mitigate the wakeup delay and short-term sleeps

- Normal routing:
  - Router \( i \) calculates the output port of Router \( i \)

- Look-ahead routing:
  - Router \( i \) calculates the output port of Router \( i+1 \)

Look-ahead can eliminate a wakeup delay of less than 5-cycle

- Five-cycle margin until packet arrival
- R2 detects a packet arrival when the packet arrives at R4
- Packet will arrive after two hops
- R0 R1 R2
- R3 R4 R5
- R6 R7 R8

[Matsutani, ASP-DAC'08]
Look-ahead method: HW resources

- Routing computation of next router
  - Just changing the routing function
  - Area overhead is very small

- Wakeup signals are needed
  - Sender asserts “wakeup” signal to receiver
  - Wakeup signals becomes long
  - Negative impact of multi-cycle or repeater buffers

NRC stage: Next Routing Computation
VC activation: three grouping methods

• 4VC x 1 (# of lane is 1)
  - Starting from VC#0,
  - A packet moves VC#0 → VC#1 → VC#2 → VC#3

• 2VC x 2 (# of lanes is 2)
  - If (dst%2)=0: a packet moves VC#0 → VC#1
  - If (dst%2)=1: a packet moves VC#2 → VC#3

• 1VC x 4 (# of lanes is 4)
  - If (dst%4)=0: a packet uses VC#0
  - ...
  - If (dst%4)=3: a packet uses VC#3

The first one (used in this paper) achieves the highest performance with the least leakage power
  - If (dst%2)=1: a packet moves VC#2 → VC#3
Buffer design: Registers or SRAMs

- It depends on buffer depth, not width
  - Depth > 32-flit → Buffers are designed with SRAMs
  - Otherwise → Buffers are designed with registers

In our design:
Buffer depth is 4-flit
FIFO buffers are designed with registers