Channel Slicing: a Way to Build Fast Routers for Asynchronous NoCs

Wei Song and Doug Edwards
The University of Manchester
15/09/2009
Content

- **Asynchronous NoCs**
- Channel Slicing
  - Motivation
  - Sliced sub-channels
  - Flow control
- An asynchronous wormhole router
  - Implementation details
  - Performances
Network-on-Chip (NoC)

RT: router  
NI: Network Interface  
PE: Processor Element 

PE: Processor Element  
NI: Network Interface  
RT: router
Synchronous/Asynchronous

• Synchronous
  – Fast
    • Intel 80-tile 4GHz 65nm
    • DSPIN 408MHz 130nm
  – Small
    • DSPIN 0.161mm
  – Power Consuming
    • 10.39mW (250MHz)
  – Sensitive to variation
  – Complex clock tree

• Asynchronous
  – Slow !!
    • ASPIN 714MHz 90nm
    • ANoC 220MHz 130nm
  – Large
    • ANoC 0.211mm
  – Power Efficient
    • 3.69mW (160MHz)
  – Tolerance to variation
  – No clock tree
Content

• Asynchronous NoCs
• **Channel Slicing**
  – Motivation
  – Sliced sub-channels
  – Flow control
• An asynchronous wormhole router
  – Implementation details
  – Performances
Asynchronous Pipelines

- **CHAIN** (*Bainbridge’02*)
  - 4 phase 1-of-4 pipelines
- **QoS NoC** (*Felicijan’04*)
  - 8-bit, Four 4 phase 1-of-4 pipelines
- **ANoC** (*Beigne’05*)
  - 32-bit 16 4 phase 1-of-4 pipelines
- **SpiNNaker** (*Plana’07*)
  - Several 1-of-4/2-of-7 pipelines
- **ASPIN** (*Sheibanyrad’08*)
  - 32-bit 16 dual-rail pipelines / bundled-data
- **MANGO** (*Bjerregaard’05*) & **QNoC** (*Dobkin’09*)
  - Bundled-data
Completion Detection

Advantages: data on all sub-channels are synchronized, ease the time division multiple access (TDMA) techniques, such as virtual channel and TDMA

Drawbacks: low speed (66% on CD)
ChSlice: implementation

2-bit

\[
\begin{align*}
\text{C} & \quad \vdots \quad \text{C} \\
\text{C} & \quad \text{CD} & \quad \text{C} & \quad \text{CD}
\end{align*}
\]

2-bit

\[
\begin{align*}
d_i & \quad 16 \\
d_o & \quad \vdots
\end{align*}
\]

\[
\begin{align*}
d_i & \quad \text{d}_i0 \\
d_o & \quad \text{d}_o0
\end{align*}
\]

\[
\begin{align*}
\text{ack}_i & \quad \text{ack}_i0 \\
\text{ack}_o & \quad \text{ack}_o0
\end{align*}
\]

2-bit

\[
\begin{align*}
d_i & \quad \text{d}_i15 \\
d_o & \quad \text{d}_o15
\end{align*}
\]

\[
\begin{align*}
\text{ack}_i & \quad \text{ack}_i15 \\
\text{ack}_o & \quad \text{ack}_o15
\end{align*}
\]
How to do it in a router?
Flow control

The diagram illustrates the concept of flow control in data transmission. It shows the progression of data over time, with sub-channels being used to route different data packets. The vertical axis represents time, and the horizontal axis shows the progression of data. The head routing data is indicated in the diagram, highlighting the sequence and timing of data transmission.
Content

• Asynchronous NoCs
• Channel Slicing
  – Motivation
  – Sliced sub-channels
  – Flow control
• An asynchronous wormhole router
  – Implementation details
  – Performances
Router: structure

5 input ports

5 output ports

ctl

arbiter

ack_i_0

d_i_0

ack_i_4

d_i_4

ack_o_0

d_o_0

ack_o_4

d_o_4

80 16

80 16

80 16
Router: data path
Re-Synchronization

- input buffer
  - ib_d
  - ic_d
- crossbar
- output buffer
  - oc_d
  - ob_d
- ip_d
- op_d

States:
- normal frame
- faulty frame

Signals:
- ip_d
- ip_a
- ib_d
- ib_a
- ib_pa
- ib_eof
- ib_dec
- ip_eof
- ib_fin

Operation:
- rt_dec
- rt_err
- acken
- ch_fin

Acknowledgment:
- normal frame
- faulty frame
Routing Decision

Advanced Processor Technology Group
The School of Computer Science
Router: layout

• Faraday 130nm Technology
• 32-bit, 5 ports, XY routing algorithm
• 0.3x0.3mm (12.6K gates, 0.050mm²)
• Typical corner (25 °C 1.2V)
• Cycle period 2.2 ns (1.82GByte/s per port)
• Equivalent to 450MHz
## Compare with other routers

<table>
<thead>
<tr>
<th></th>
<th>Tech (nm)</th>
<th>Period (ns)</th>
<th>Period (Hz)</th>
<th>Pipeline Style</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sliced Wormhole</td>
<td>130</td>
<td>2.2</td>
<td>450M</td>
<td>4-phase 1-of-4</td>
<td>Standard cell</td>
</tr>
<tr>
<td>Synchronized Wormhole</td>
<td>130</td>
<td>2.8</td>
<td>360M</td>
<td>4-phase 1-of-4</td>
<td>Standard cell</td>
</tr>
<tr>
<td>ANoC</td>
<td>130</td>
<td>4.0</td>
<td>250M</td>
<td>4-phase 1-of-4</td>
<td>Customized Cell Lib</td>
</tr>
<tr>
<td>ASPIN</td>
<td>90</td>
<td>0.88</td>
<td>1.13G</td>
<td>Dual-Rail / Bundled-Data</td>
<td>Customized FIFO</td>
</tr>
<tr>
<td>QNoC</td>
<td>180</td>
<td>4.8</td>
<td>208M</td>
<td>Bundled-data</td>
<td>Delay line</td>
</tr>
<tr>
<td>MANGO</td>
<td>120</td>
<td>1.26</td>
<td>790M</td>
<td>Bundled-data</td>
<td>Delay line</td>
</tr>
<tr>
<td>DSPIN</td>
<td>130</td>
<td>2.45</td>
<td>408M</td>
<td>Synchronous circuit</td>
<td></td>
</tr>
</tbody>
</table>
Speed vs. Data Width

Sliced Wormhole

QNoC

Advanced Processor Technology Group
The School of Computer Science
Speed and Area

![Graphs showing speed and area vs. data width for channel sliced and synchronized systems.](image-url)
Question?