Reliable NoC in the Many Core Era

Topic: NoC monitoring and error detection

Author: Hou, Jie

Supervisor: Adan Kohler
• Agenda

1. Why do we need monitoring for NoC?

2. A generic event based NoC monitoring system

3. What is Æthereal?

4. An event based monitoring system for Æthereal NoC

5. Transaction monitoring for NoC

6. The impact of NoC monitoring
1 Why do we need monitoring for NoC?

• Two main functional parts of NoC based large scale SoCs
  • Computation
  • Communication

• Why do we need observability?
  • It is important for system debugging and adaptivity

• Observability
  • Computation Observability (e.g. ARM’s ETM)
  • Communication Observability
Agenda

• Agenda

1. Why do we need monitoring for NoC?

2. A generic event based NoC monitoring system

3. What is Æthereal?

4. An event based monitoring system for Æthereal NoC

5. Transaction monitoring for NoC

6. The impact of NoC monitoring
2 A generic event based NoC monitoring system

- The monitoring system uses hard-ware probes and it can provide services
  - Capturing runtime information and functional data
  - Generating information based on events

A generic monitoring system (source C.Ciordas)
2 A generic event based NoC monitoring system

2.1 Generic Event model

• Classifications of events

The fifth type: Monitoring Service Internal Events

NoC event space (source C. Ciordas)
2 A generic event based NoC monitoring system

2.2 Generic Probe Architecture

- It contains three parts: Sniffer(S), Event Generator (EG), Monitoring Network Interface (MNI)

(source C.Ciordas)

MSA: Monitoring Service Access
Agenda

1. Why do we need monitoring for NoC?
2. A generic event based NoC monitoring system
3. What is Æthereal?
4. An event based monitoring system for Æthereal NoC
5. Transaction monitoring for NoC
6. The impact of NoC monitoring
3 What is Æthereal?

• Æthereal is an example of a specific NoC proposed by Philips.

• It defines the protocol, switching and routing schemes as well as a design flow.

• It provides two types of communication services
  • Guaranteed throughput service (GT)
  • Best effort service (BE)

• Difference between GT and BE
  • GT services have higher priority than BE services
Agenda

1. Why do we need monitoring for NoC?
2. A generic event based NoC monitoring system
3. What is Æthereal?
4. An event based monitoring system for Æthereal NoC
5. Transaction monitoring for NoC
6. The impact of NoC monitoring
4 An event based monitoring system for Æthereal NoC

4.1 Event Model for Æthereal NoC

What is an event for Æthereal?

Event:

- **Identifier**: 8 bit
- **Timestamp**: 16 bit
- **Producer**: 8 bit
- **Attributes**: describes the type of an event
- **Defines at which time the event is generated**
- **Describes which element generates this event**
- **Useful payload of an event**
4 An event based monitoring system for Æthereal NoC

• 4.1 Event Model for Æthereal NoC

User Configuration Events
- Connection opened event (attrs: connection identifiers, connection type, connection ports)
- Connection closed event (attrs: connection identifiers)

User Data Events
- GT sniff event (attrs: GT flit, the identifier of the queue where the information was sniffed)
- BE sniff event (attrs: BE flit, the identifier of the queue where the information was sniffed)
4 An event based monitoring system for Æthereal NoC

• 4.1 Event Model for Æthereal NoC

NoC Configuration Events

- Reserve slot event (attrs: the reserved slot number and its value)
- Free slot event (attrs: the released slot number)

NoC Alert Events

- Queue filling event (attrs: how many queues a router has, the states of each queue)
- End-to-end credit 0 event (attrs: connection identifier)

Monitoring Service Internal Events: Synchronization
4 An event based monitoring system for Æthereal NoC

4.2 An event based monitoring system for Æthereal

(source C.Ciordas)
4 An event based monitoring system for Æthereal NoC

4.3 Æthereal Probe Architecture

- The EG generates timestamped events.
- Currently Æthereal NoC uses 8 bits for identifier, 16 bits for timestamp, and 8 bits for producer.

source C.Ciordas
4.4 Æthereal Probe’s programming model

When can a probe be configured?
• For Æthereal, they can be configured at runtime because the probes have a memory mapped slave port.

Monitoring connection setup
• Because packets that contain generated events must be transported to MSA, the connection between MNI and MSA must be set like normal standard Æthereal connections.

Probe setup
• The setup contains: event selection, attributes selection, start-time selection and enable/disable probe selection.
• Agenda

1. Why do we need monitoring for NoC?

2. A generic event based NoC monitoring system

3. What is Æthereal?

4. An event based monitoring system for Æthereal NoC

5. Transaction monitoring for NoC

6. The impact of NoC monitoring
What is a transaction?
- For NoCs, the interconnected IPs use transactions to communicate with each other.
- It can contain one or more messages, which can be read message or write message.

Why do we need transaction level monitoring for NoC?
- To increase the operational speed of system level debugging.

A transaction monitoring system can be achieved by replacing the EG with a Transaction Monitor.
• Æthereal packet format and flit format
The NIs can convert messages into packets, a packet usually contains several flits.

Æthereal Packet format

<table>
<thead>
<tr>
<th>GT</th>
<th>5</th>
<th>2</th>
<th>2.2,1,0,1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>payload 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>payload 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GT</td>
<td>payload 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>payload 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>payload 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GT</td>
<td>payload 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>payload 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eop</td>
<td>payload 7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Æthereal Flit format

<table>
<thead>
<tr>
<th>id</th>
<th>credit</th>
<th>qid</th>
<th>path</th>
<th>word 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>payload 0</td>
<td></td>
<td></td>
<td>word 1</td>
</tr>
<tr>
<td>eop</td>
<td>payload 1</td>
<td></td>
<td></td>
<td>word 2</td>
</tr>
</tbody>
</table>

(source IES06)
Transaction monitoring for NoC

- Transaction monitor (TM) Architecture

- TM uses master port (MP) and slave port (SP) to connect to MNI.

- It has four working modes, and each processing block corresponds to one mode.

- The processing blocks can be chosen via enable/configuration block.

(sourceIES06)
5 Transaction monitoring for NoC

• Working modes of TM

① Raw Mode
• It uses \textit{GT/BE filtering} block.
• After this process, all flits of one service type are monitored.

② Connection based mode
• It uses \textit{connection filtering} block.
• It uses the output of raw mode.
• After this process, we can get all the flits belong to one virtual connection
3. Transaction based mode
   • It uses the *depacketization* block.
   • After this process, we can distinguish the different messages that belong to the same connection.
   • The problem is to identify the messages and detect the start of the message. For Æthereal, it has hardware modules for depacketization.

<table>
<thead>
<tr>
<th>MP</th>
<th>MH</th>
<th>PH</th>
</tr>
</thead>
</table>

4. Transaction event based mode
   • It uses *abstraction* block.
   • For “unimportant” messages, after this process only a transaction event is generated. This event just contain the features of the message.
Agenda

1. Why do we need monitoring for NoC?
2. A generic event-based NoC monitoring system
3. What is Æthereal?
4. An event-based monitoring system for Æthereal NoC
5. Transaction monitoring for NoC
6. The impact of NoC monitoring
The impact of NoC monitoring

- Monitoring elements need to be considered during the NoC design flow
- Typical NoC design flow

It is split in four steps:

- Selection step: the router network together with bordering NIs are generated
- Mapping step: binding IPs to NI ports based on generated topology
- Path selection: paths are allocated for all the specified communication flows.
- Slot allocation: Based on TDMA method, every communication flow gets its own time slot

(source ISCAS06)
6 The impact of NoC monitoring

• Separate physical interconnect for User NoC and Monitoring NoC

The monitoring NoC:

• It can transport traffics (event traffic, programming traffic) between probes and MSA.

• It can have similar topology as the user NoC.

• Obviously, it would not interfere with user NoC!
6 The impact of NoC monitoring

- Separate physical interconnect for User NoC and Monitoring NoC

The design flow is applied twice:

- User NoC: the same as typical design flow
- Monitoring NoC: taking into account the monitoring requirements and the desired debug IPs.
- For monitoring NoC: only path selection and slot allocation need to be done again.

(source ISCAS06)
6 The impact of NoC monitoring

- Shared existing NoC for both user data and monitoring data

- User traffic and monitoring traffic share all the resources of existing user NoC

- The mapping of probes is based on the nearest NI principle

- Path selection and slot allocation: considering user data and monitoring data together

  - *The user traffic can be interrupted by monitoring traffic!*

(source ISCAS06)
6 The impact of NoC monitoring

- Shared existing NoC for both user data and monitoring data

Two possibilities:

- Everything fits into the existing NoC. Lowest area cost.

- Combined communication requirements can not fit into the existing NoC.

Worst case: no solution!

(source ISCAS06)
Do you have any question?

THANK YOU!