Coupling TDM NoC and DRAM Controller for Cost and Performance Optimization of Real-Time Systems

Manil Dev Gomony\textsuperscript{1}, Benny Akesson\textsuperscript{2} and Kees Goossens\textsuperscript{1}

\textsuperscript{1} Eindhoven University of Technology, The Netherlands
\textsuperscript{2} Czech Technical University in Prague, Czech Republic
Multi-processor platforms

- Network-on-Chip is used as an interconnect
  - *For scalability as opposed to bus-based interconnects*
- Main memory (DRAM) is a shared resource
  - *For cost and communication reasons*
In real-time systems

• Memory clients come with real-time requirements on memory bandwidth and/or latency
  – *Real-time NoCs and Memory Subsystems*

• Minimize cost and maximize performance!
Real-time NoC

- Statically scheduled Time Division Multiplexed (TDM) NoC
  - *Single global TDM schedule*
- Provide end-to-end guarantees on latency and bandwidth between a source and destination
  - *Dedicated virtual channel/circuit*
Real-time memory subsystems

• Provide guarantees on memory bandwidth and latency to a memory client
  – *Real-time memory controller*: Fixed set of memory access parameters (burst size, page-policy)
    • Atomizers (AT) split larger transactions to smaller sized *service units*
  – Predictable arbiter for resource sharing
Motivation

- Existing **TDM NoCs** and real-time **memory subsystems** are optimized independently

- **Multiple arbitration points!**
- **Reducing to a single arbitration**
  - *Reduces worst-case latency*
  - *Destination NI needs a single output port*
Our contributions

• Novel methodology to couple any existing TDM NoC with a real-time memory subsystem
  – Compute the different NoC configurations for minimal area and/or power consumption

• Trade-off between area and power consumption
  – For different NoCs and memory devices

• Comparison of coupled and decoupled architectures by synthesizing the designs
  – In 40nm technology
Outline

• Introduction
• Coupling TDM NoC and memory subsystem
• Dimensioning the NoC
• Experiments
• Conclusions
Coupling NoC and memory subsystem

- Remove the bus-based arbitration in the memory subsystem
- Move atomizers and decoupling buffers to the client side
- Perform memory arbitration in the NoC
  - Lose flexibility in selecting different arbitration for memory!

![Diagram of Coupling NoC and memory subsystem]
Need to address..

- Different clock domains, transaction granularities
  - What should be the buffer size?
  - How to select NoC interface width and operating frequency?
  - How to guarantee real-time performance to clients?

![Diagram of a memory subsystem with multiple clients, routers, and arbitration units connected through a network-on-chip (NoC).]
NoC configuration

- NoC transports *flits* and the memory controller executes service units
  - *Configure NoC flit size = service unit size*
- Buffer of size equal to the service unit is required at the destination NI
  - *Complete service unit need to be buffered*
Bandwidth matching

- NoC link bandwidth must be same as the memory subsystem in a service cycle

$$f_n \times \frac{IW_n}{8} \times \frac{SC^{cc}_n - \delta_{ov}}{SC^{cc}_n} = f_m \times \frac{SU^{bytes}}{SC^{cc}_m}$$
Clock alignment

- Clock edges of NoC and memory subsystem must be **aligned** at the service cycle boundaries
  - *Single clock source*
Coupled architecture - operation

TDM allocation

SC

Client1 Client2 Client3 Client4 Client5

Clock

clkn
ck

Router 1

Client1

Client2

na

na

na

Client1

Router 2

na

na

Client3

Client4

na

na

Router 3

Client1

Client2

Client3

Client4

Client5

Client1

Request path

Client2

Response path

Memory

Command

data

Request path

Response path

TDM allocation

Global TDM schedule

NoC

DRAM

Real-time memory controller

Buffers

Arb

AT

Na

28-Mar-14

Manil Dev Gomony / Eindhoven University of Technology
Outline

• Introduction
• Coupling TDM NoC and memory subsystem
• Dimensioning the NoC
• Experiments
• Conclusions
The service cycle duration of a given memory device can be computed using state-of-the-art methods.

<table>
<thead>
<tr>
<th>DRAM device</th>
<th>Memory frequency ($f_m$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPDDR-266</td>
<td>133</td>
</tr>
<tr>
<td>LPDDR-416</td>
<td>208</td>
</tr>
<tr>
<td>LPDDR2-667</td>
<td>333</td>
</tr>
<tr>
<td>LPDDR2-1066</td>
<td>533</td>
</tr>
<tr>
<td>DDR3-800</td>
<td>400</td>
</tr>
<tr>
<td>DDR3-1600</td>
<td>800</td>
</tr>
</tbody>
</table>

The interface width and operating frequency of the NoC need to be selected based on area vs. power trade-off.
NoC dimensioning

• Given a memory device with frequency $f_m$, service unit size of $SU^{\text{bytes}}$ with service cycle $SC_m^{cc}$

• Determine all $(f_n, IW_n)$ combinations which satisfies the hardware constraints

  **Step 1:** Compute all possible values of $f_n$ that are integer multiples and common fractions of $f_m$

  **Step 2:** Select the values of $f_n$ such that the clocks will be aligned at the boundaries of the service cycles

  $$(SC_m^{cc} \times f_n) \mod f_m = 0$$

  **Step 3:** Compute the values of $IW_n$ corresponding to the different $f_n$ using the bandwidth matching equation

  $$f_n \times \frac{IW_n}{8} \times \frac{SC_n^{cc} - \delta_{ov}}{SC_m^{cc}} = f_m \times \frac{SU^{\text{bytes}}}{SC_m^{cc}}$$
Introduction
Coupling TDM NoC and memory subsystem
Dimensioning of TDM NoC
Experiments
Conclusions
Experimental setup

- RTL-level implementations of
  - *Router and NI of two different TDM NoC types*
    - *Packet switched : Aelite*
    - *Circuit switched : Daelite*
  - *TDM arbiter*
  - *Bus using the Device Transaction Level (DTL) protocol*
    - *Comparable to AXI and OCP protocols*

- Power/area estimation using the Cadence Encounter RTL compiler
  - *40 nm nominal Vt CMOS standard cell library*
Area vs. power trade-off

Graph showing the relationship between Area (mm²) and Total power (mW), with operating frequency and interface width as parameters. The graph includes data points and lines for various memory technologies, such as LPDDR-266, LPDDR-416, LPDDR2-667, LPDDR2-1066, DDR3-800, and DDR3-1600, each with different data rates (MB/s). The y-axis represents total power in mW, and the x-axis represents area in mm². The graph highlights the trade-offs between area and power for different operating frequencies and interface widths.
• 16 memory clients → 16 ports in destination NI
• Four-stage NoC tree consisting of 15 routers and NIs
• Service unit size 64 B → NI buffer size

Area/power savings

- Over 11%
- Over 17%

<table>
<thead>
<tr>
<th>Memory Type</th>
<th>Area Savings (%)</th>
<th>Power Savings (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPDDR-266</td>
<td>Over 17%</td>
<td></td>
</tr>
<tr>
<td>LPDDR-416</td>
<td>Over 17%</td>
<td></td>
</tr>
<tr>
<td>LPDDR2-667</td>
<td>Over 17%</td>
<td></td>
</tr>
<tr>
<td>LPDDR2-1066</td>
<td>Over 17%</td>
<td></td>
</tr>
<tr>
<td>DDR3-800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDR3-1600</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manil Dev Gomony / Eindhoven University of Technology
Worst-case latency savings

- 16 clients with same bandwidth allocated to all clients
  - Assuming the same TDM allocation for the NoC in both architectures

<table>
<thead>
<tr>
<th>Memory</th>
<th>NoC frequency (MHz)</th>
<th>Interface width (bits)</th>
<th>Worst-case latency (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Decoupled</td>
</tr>
<tr>
<td>LPDDR-266</td>
<td>266.0</td>
<td>15</td>
<td>4.81</td>
</tr>
<tr>
<td>LPDDR-416</td>
<td>416.0</td>
<td>15</td>
<td>3.08</td>
</tr>
<tr>
<td>LPDDR2-667</td>
<td>399.6</td>
<td>16</td>
<td>2.51</td>
</tr>
<tr>
<td>LPDDR2-1066</td>
<td>355.3</td>
<td>22</td>
<td>2.45</td>
</tr>
<tr>
<td>DDR3-800</td>
<td>480.0</td>
<td>23</td>
<td>2.11</td>
</tr>
<tr>
<td>DDR3-1600</td>
<td>400.0</td>
<td>17</td>
<td>1.83</td>
</tr>
</tbody>
</table>

Over 44 %
Outline

- Introduction
- Coupling TDM NoC and memory subsystem
- Dimensioning of TDM NoC
- Experiments
- Conclusions
Conclusions

• Existing NoCs with a global TDM schedule can be coupled with a real-time memory controller by
  – Configuring NoC flit size equal to the service unit size
  – Selecting NI buffer size equal to the service unit size
  – Using a single clock source

• We proposed a methodology for computing the NoC parameters for power vs. area trade-off

• For a system with up to 16 clients, coupling the NoC with the memory subsystem saves over
  – 17% area
  – 11% power consumption
  – 44% worst-case latency

• Give up flexibility in selecting different arbiters for memory!
Questions?