Politecnico di Milano

Application-Specific Design Customization for Network on-Chip Architectures

Gianluca Palermo – gpalermo@elet.polimi.it
Motivations

- Most of the on-chip communication architecture are developed for one application.
- Network on chip architectures are characterized by a wide range of design solutions.
- Application specific design time specialization is an important step to exploit the features of the NoC paradigm.
- Need of methodologies and design tools.
Outline

- Introduction
- Application Specific NoC Design
- Proposed Design Flow
- Topology Customization on STNoC
- Experimental Results
- Conclusions and Future Works
In the future SoCs designer will continue to:

- Increase functionality putting many applications on a chip
- Increase speed of cores

Semiconductor firms trend is to propose systems based on few advanced cores tightly interconnected on a single chip.

This richness of the computational resources places tremendous demands on the communication resources as well.
To date, the shared-bus scheme (either single bus or multi-bus) has been the system communication architecture of choice
- Large capacity load
- Bus does not scale with the number of processing units

The scalability and success of switch-based networks and packet-based communication in parallel computing has inspired the researchers to propose the **Network-on-Chip (NoC)** architecture as a viable solution to the on-chip communication problems.
NoC Design Space

- Network on-Chip architectures are characterized by a large design space
  - NoC-level:
    - Topology
    - Routing Algorithm
    - Data-path width
    - Mapping of cores
    - ...
  - Router-Level
    - Input/Output queue
    - Buffer size
    - Arbitration Mechanism
    - ...

- NoC are developed for either a single application or for as a platform for small class of applications

- For application specific design it is important to find the best architectural configuration
Previous Works

- **Random stimuli for NoC evaluation**
  - [Bononi et al. – DATE06], [Pande et al. – TCOM05]
  - QoS [Santi et al. ISCAS05]

- **Mapping of cores onto Network tiles**
  - Bandwidth constraints [Murali and DeMicheli – DATE04]
  - Energy-aware [Hu and Marculescu ASP-DAC03]
  - Evolutionary algorithms [Ascia et al. CODES-ISSS04]
  - Multi-use cases [Murali et al. DATE06/ASP-DAC06]

- **Application specific customization**
  - STBus [Murali and DeMicheli – DATE05]
  - Buffer allocation [Hu et al. – TCAD06]
  - Long link insertion [Orgas and Marculescu – TVLSI06]
STNoC

- Developed by STMicroelectronics
- Packet switched network
- Family of topologies between Ring and Spidergon
  - Low complexity (area/power advantages)
  - Symmetry (easy to route)
  - Low diameter (performance advantages)

Complexity

Ring | Spidergon | 2D-mesh | 2D-torus
Design Flow

Application

Application Core Graph

Design Space Explorer (STShell)

SystemC Models

Simulation Environment (GRAPES)

NoC Design Space

NoC Instance

SystemC Model

NoC-Compiler

NoC Libraries

Application Specific NoC Configuration

Latency

Area/Power
The Design Space Exploration (DSE) is a wide problem and it is an important phase in all the designs of innovative systems.

Existing exploration engines are not retargetable: Designed for specific target problems or use scripting languages (e.g., Python, Perl).

STShell is an automatic exploration tool for abstract exploration problems.

Exploration kernel that provides API for the independent development of problem drivers (used to wrap the exploration problems) and exploration algorithms.
Trend in using only one joined description for the three main blocks (DS-Explorer, NoC-Compiler, Simulation Environment)

Examples of module descriptions:

```xml
<module name="ipo">
    <parameter name="type" value="IP"></parameter>
    <parameter name="path" value="lib/libIPexample.so"></parameter>
</module>

...

<module name="ro">
    <parameter name="type" value="NoCRouter"></parameter>
    <parameter name="latency" value="2"></parameter>
    <parameter name="maxConnections" value="4"></parameter>
    <parameter name="bufferDepth" value="4"></parameter>
    <parameter name="path" value="lib/libNoCRouter.so"></parameter>
</module>

...

<connection name="co">
    <parameter name="source" value="ipo"></parameter>
    <parameter name="destination" value="ro"></parameter>
    <parameter name="maxBw" value="1600"></parameter>
</connection>
```
One of the first steps in the NoC design flow is the topology selection.

In application-specific systems the communication behavior is **not symmetric**.

Standard topologies already used for macro-networks cannot be useful as in the past.

**Target communication core graph:**

---

**Macro Networks**

**NoC**
Motivation Example (1)
Motivation Example (2)
Proposed Solution

- Topology customization and mapping are correlated
  - Huge design space!

- Problem decomposition:
  - Mapping of core onto NoC tiles
  - Topology Customization
Proposed Approaches for STNoC

- STNoC supports a family of topology ranging from ring to spidergon

- Key points for the optimization
  - Initial Mapping of the cores on the NoC tiles
  - Type of Customization

- 4 different approaches have been implemented
Design Space Description

Architecture definition with 2 new keyword:

- ALL

- VOLATILE

```xml
<connection name="co">
    <parameter name="source" value="a"></parameter>
    <parameter name="destination" value="ALL"></parameter>
    <parameter name="maxBw" value="1600"></parameter>
    <parameter name="VOLATILE" value="true"></parameter>
</connection>
```
Routing Algorithm

- For custom topology one of the problem is the routing since the network MUST BE deadlock free.

- Each architecture is evaluated by applying one of the following routing:
  - Min path:
    - Is used only if it is deadlock free.
  - Cross First:
    - The diagonal direction can be used only on the FIRST hop.
  - Cross Last:
    - The diagonal direction can be used only on the LAST hop.
Fully-From-Ring (FFR)

- Starting mapping on RING
- FULLY customization of the diagonal connections
- Complexity for N-nodes architecture:
  - Mapping: N!
  - Topology customization: N!
- Advantages:
  - Initial mapping
  - Short added links
  - Limited number of added links
- Disadvantages:
  - Huge design space
  - Custom topology
Fully-From-Spidergon (FFS)

- Starting mapping on SPIDERGON
- FULLY customization of the diagonal connections
- Complexity for N-nodes architecture:
  - Mapping: N!
  - Topology customization: N!
- Advantages:
  - Good uses of the added links
- Disadvantages:
  - Initial mapping
  - Huge design space
  - Custom topology
Starting mapping on SPIDERGON

Customization of the diagonal connections only with the SUBSET of spidergon

Complexity for N-nodes architecture:
- Mapping: $N!$
- Topology customization: $2^N$

Advantages:
- Initial mapping
- Added links are a subset of spidergon (not a custom topology)
- Small design space for the customization (fast exhaustive exploration)

Disadvantages:
- Small design space
- Routing
- Mapping techniques
Subset-From-Ring (SFR)

- Starting mapping on RING
- Customization of the diagonal connections only with the SUBSET of spidergon
- Complexity for N-nodes architecture:
  - Mapping: N!
  - Topology customization: $2^N$
- Advantages:
  - Added links are a subset of spidergon (not a custom topology)
  - Small design space for the customization (fast exhaustive exploration)
- Disadvantages:
  - Small design space
  - Routing
  - Initial mapping
Better solutions have been obtained by using the FFR and SFS approaches w.r.t. FFS and SFR

- **FFR – Fully From Ring:**
  - Uses fully custom link to solve mapping problem due to the ring topology
- **SFS – Subset From Spidergon:**
  - Reduces the number of useless diagonal connections
- **FFS – Fully From Spidergon:**
  - Contains all the solution of SFS but the DS is huge
- **SFR – Subset From Ring:**
  - The diagonal connection are not well used since the mapping have been performed without them
MPEG4
Power/Area Comparison

December 5, 2006  Gianluca Palermo
MPEG4
Weighted Distance/Area Comparison

Area [mm²]

Weighted Average Network Distance [#hops]
FFR – Results Examples

Results Examples

TGFF12

MPEG4
**FFR – Some Examples**

Target Application

<table>
<thead>
<tr>
<th></th>
<th>RING</th>
<th>CT7</th>
<th>CT15</th>
<th>SPIDERGON</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>-</td>
<td>+8.9</td>
<td>+19.2</td>
<td>+41.4</td>
</tr>
<tr>
<td>POWER</td>
<td>-</td>
<td>-38.6</td>
<td>-40.4</td>
<td>-24.4</td>
</tr>
<tr>
<td>LATENCY</td>
<td>-</td>
<td>-29.7</td>
<td>-32.2</td>
<td>-30.7</td>
</tr>
<tr>
<td>AB [MB/s]</td>
<td>3192</td>
<td>1900</td>
<td>1798</td>
<td>1876</td>
</tr>
</tbody>
</table>

(a) **RING**  
(b) **CT7**  
(c) **CT15**  
(d) **SPIDERGON**
Problem Extension to Multi-Use Cases

- Complex system architectures

- Target application:
  - Not unique (set of applications)
  - Single application characterized by different communication behaviors

=> Same architecture but different edges in core graphs

Use Case 1

Use Case 2

Use Case n
The design flow is based on a multi-objective exploration engine

Problem solved adding exploration cost functions

- Ex. Exploration with 3 cost functions
  (CF³ is only architectural dependent, e.g. Area)
  - Single use case cost function vector:
    \[ \langle CF^1, CF^2, CF^3 \rangle \]
  - Double use-case cost function vector:
    \[ \langle CF^1_{uc1}, CF^1_{uc2}, CF^2_{uc1}, CF^2_{uc2}, CF^3 \rangle \]
Results for FFR
What about latency?

- Some hints about latency are inserted in the flow
  - Probability of contentions
  - Network distances
  - Simple latency model

- Evaluation using simulations:
  - Time consuming (remember that is a wide design space)
  - Needs real traffic pattern:
    - System-level platform of the architecture
    - Traffic pattern generators specific for the target applications
    - Communication characteristics (Single word, Burst..., binomial, pareto, exponential...)

- Queue network theories have been already used in the NoC field by Marculescu et al. CMU for buffer allocation in application specific NoC
Conclusions and Future Works

- Application specific design flow for network oriented architectures is an hot topic

- We presented our approach focused on the topology customization on the STNoC architecture

Future works:
- Fast latency estimations
- Extension of the approach for other topologies

- Definition of a semi-standard description of the target architecture design space