The HeartBeat model
A platform abstraction enabling fast prototyping of real-time applications on NoC-based MPSoC on FPGA

Francesco Robino, Johnny Öberg

KTH Royal Institute of Technology

ReCoSoC 2013
Overview of the talk

- Motivation
  - What is the problem?
  - Benefits solving the problem
- Our solution
  - Modeling techniques
  - Our idea
  - From idea to implementation
  - Experimental evidences and examples
- Conclusion
Motivation: Embedded systems modeling

- Embedded systems are getting very complex
- We want to abstract, reduce details in the system model

We want System Design Automation to automatically add details
  - it implements the embedded application on multi-processor platforms
  - the automated synthesis of the whole system must be fast!
Motivation: Embedded system architecture

- Embedded architectures are getting very complex
- Details are increasing!

- Sea-of-cores / processors
- NoC-based MPSoC
The problem

- Huge abstraction gap
- Huge design space
  - Platform allocation
  - Processes binding
  - Processes schedule
- Platform:
  - complex to program
  - complex to generate

Filling the gap through design automation: benefits

- Reduces time to market, reduces errors, reduces complexity...
A MoC defines **communication and execution rules between nodes executing the operations**.

- The MoC rules can be described through formal notation (math)
- Embedded application modeled as a set of communicating processes
The synchronous MoC

- We use the *synchronous MoC*
- Used in synchronous languages (i.e. Lustre) to guarantee real-time
- Combinatorial application in the synchronous MoC:

```
Tag   Event   Value
0 1 2 3
4 7 6 9
0 1 2 3
3 6 5 8
```

Application with memory in the synchronous MoC:

```
Tag   p_{+1}   dly
0 1 2 3
0 4 7 6
```

From synchronous MoC to NoC-based MPSoC

[Diagram showing the transition from synchronous MoC to NoC-based MPSoC, with tags and time intervals indicated.]
From synchronous MoC to NoC-based MPSoC

- We define an intermediate layer, the HeartBeat model:
  - fills the abstraction gap adding details about timing and architecture
  - enables the MPSoC to expose the same semantics of the MoC
  - critical path (RTL, sea of gates) ⇐ HB period (MoC, sea of cores)

\[
t_{HB} \geq \max_{m=0}^{M} (t_{\epsilon_i} + t_{c_i})
\]

The HeartBeat model (for this simple example)
A HW/SW co-design flow based on the HeartBeat model

NoC system Generator
Target FPGA
Software synthesis of C wrappers for MoC support
HeartBeat
XML platform configuration and process(es) mapping
Process(es) functionality C file(s)
Application view Platform view
Graphical User Interface
Front-end flow
Back-end flow
Allocation
Binding
Scheduling

SW synthesis technique

- Application view:
  synthesis of software to access NoC API and synchronize on HB tick

begin
  initialize processes
  execute PE processes
  HB tick received?
    Y
    execute user specified functionality
    N
    previous process communication concluded?
      Y
      send result to next process
      N
      message received?
        Y
        execute user specified functionality
        N
        previous process communication concluded?
          Y
          send result to next process
          N
          message received?
            Y
            execute user specified functionality
            N
            previous process communication concluded?
              Y
              send result to next process
              N
              message received?
                Y
                execute user specified functionality
                N
                previous process communication concluded?
                  Y
                  send result to next process
                  N
                  message received?
                    Y
                    execute user specified functionality
                    N
                    previous process communication concluded?
                      Y
                      send result to next process
                      N
                      message received?
                        Y
                        execute user specified functionality
                        N
                        previous process communication concluded?
                          Y
                          send result to next process
                          N
                          message received?
                            Y
                            execute user specified functionality
                            N
                            previous process communication concluded?
                              Y
                              send result to next process
                              N
                              message received?
                                Y
                                execute user specified functionality
                                N
                                previous process communication concluded?
                                  Y
                                  send result to next process
                                  N
                                  message received?
                                    Y
                                    execute user specified functionality
                                    N
                                    previous process communication concluded?
                                      Y
                                      send result to next process
                                      N
                                      message received?
                                        Y
                                        execute user specified functionality
                                        N
                                        previous process communication concluded?
                                          Y
                                          send result to next process
                                          N
                                          message received?
HW synthesis technique

Platform view:
synthesis of NoC-based MPSoC with HeartBeat support
- A HW timer generates HB ticks
- The HB period (timer period) can be defined by the user, following the Heartbeat model
Case study

- A neural network, 4 layers, each layer 100 neurons
- Each layer on a separate PE (4 layers, 4 PEs, $2 \times 2$ NoC)
- Generation of files for prototype synthesis on FPGA $< 1$ second
Case study

- Optimize the application: find the minimal HB period of the system

The HeartBeat model (for this simple example)

\[ t_{HB} \geq \max_{m=0}^{M} (t_{ci} + t_{ei}) \]

- \( t_{ei} \) has been found through extensive emulations. Its value is 5.97 ms
- \( t_{ci} \) can be found following models proposed in referenced papers
- The HeartBeat period is totally dominated by \( t_{ei} \), it is an order of magnitude larger than \( t_{ci} \)
- Concluding the maximal HB frequency of the system is \( \frac{1}{5.97} \approx 150 \) Hz
## Results and related work

<table>
<thead>
<tr>
<th>Design flow</th>
<th>MAMPS</th>
<th>Our work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target platform</td>
<td>2 processors</td>
<td>4 processors</td>
</tr>
<tr>
<td>Platform interconnection</td>
<td>$2 \times 1$ NoC</td>
<td>$2 \times 2$ NoC</td>
</tr>
<tr>
<td>Creating system model</td>
<td>manual(SDF)</td>
<td>manual(SY)</td>
</tr>
<tr>
<td>Creating application model</td>
<td>manual</td>
<td>semiautomatic</td>
</tr>
<tr>
<td>Generating architecture model</td>
<td>1 second</td>
<td>$\sim 100$ millisec</td>
</tr>
<tr>
<td>Mapping the design</td>
<td>1 minute</td>
<td>manual</td>
</tr>
<tr>
<td>Generate Xilinx/Altera project</td>
<td>16 seconds</td>
<td>$\sim 500$ millisec</td>
</tr>
</tbody>
</table>

**Table: SDA flows steps rapidity**

<table>
<thead>
<tr>
<th></th>
<th>COMPSOC</th>
<th>Open-Scale</th>
<th>MAMPS</th>
<th>Our work</th>
</tr>
</thead>
<tbody>
<tr>
<td>MoC entry</td>
<td>HSDF</td>
<td>KPN</td>
<td>SDF</td>
<td>Sync</td>
</tr>
<tr>
<td>NoC arch.</td>
<td>Mesh</td>
<td>Mesh</td>
<td>Mesh</td>
<td>Mesh</td>
</tr>
<tr>
<td>NoC dim.</td>
<td>2D</td>
<td>2D</td>
<td>2D</td>
<td>2D,3D</td>
</tr>
<tr>
<td>API generation</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Prototype</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>OS avail.</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

**Table: SDA flows properties comparison**
The HeartBeat model is an intermediate platform model bridging the abstraction gap between the synchronous MoC and a NoC-based MPSoC platform.

The HeartBeat model enables a SDA flow to synthesize an embedded application onto a NoC-based MPSoC on FPGA, provided that the timing constraints of the HeartBeat model are met.

The design flow reduces the complexity of designing embedded systems for NoC-based MPSoCs through a GUI.

The automated flow speeds up the design process of embedded systems.

More info and tutorials

https://forsyde.ict.kth.se/noc_generator/