Modeling for Transaction Generator

Updated: December 8, 2009
Department of Computer Systems
Tampere University of Technology

Contents

1 INTRODUCTION

Increasing parallelism in processing and growing number of Intellectual Properties (IPs) in System-on-Chips (SoCs) implies increasing amounts of data to be transmitted via Network-on-Chips (NoCs). Multitude of NoCs have been developed and proposed in literature which all have their own characteristics.

For a NoC developer to estimate what gain would a certain novel feature bring or a SoC developer to choose from various NoCs and configurations a proper testing and comparison method is needed.

Transaction Generator is a SystemC simulator which models the application, platform and the mapping between them. Communication via NoC is handled through OCP channels and thus any network model can be used provided it has an OCP TL2 interface.

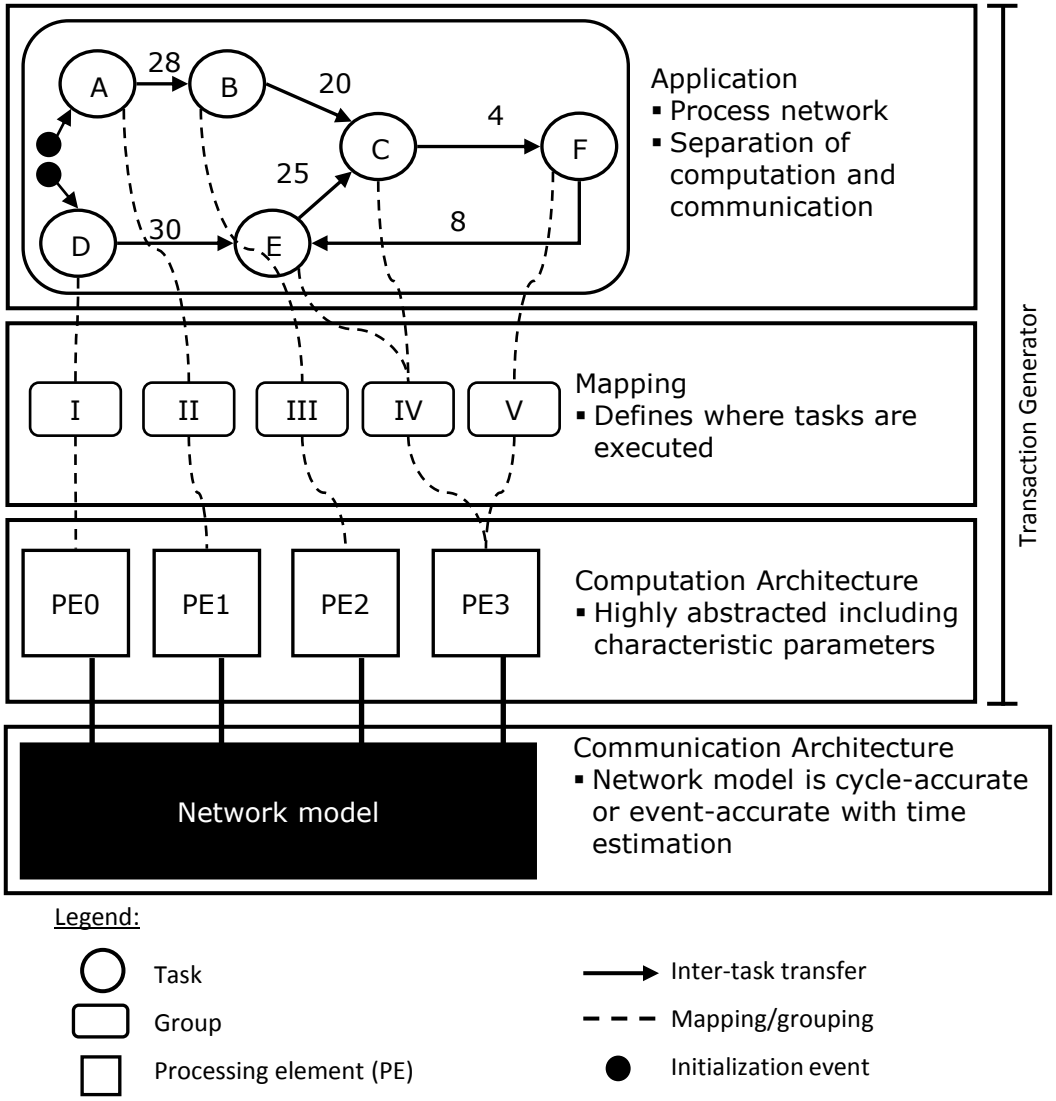

Figure 1: *Conceptual view of the utilised system model*

2 OVERVIEW OF THE USED MODEL

2.1 Application

Application model defines the computational and communicational load. It resembles Kahn Process Network (KPN). In figure 1 vertices represent computation tasks that communicate with each other via channels represented by edges.

Task model incorporates only the external behavior of application tasks. E.g. no actual computation is made but the clock cycles that would have been used is modelled. Tasks are triggered to execute when they receive a full data token to a certain port or ports. Actions taken once triggered can depend on the amount of data received and task's internal state.

2.2 Mapping

Mapping defines on which hardware resource tasks are executed. Tasks are classified dependig on which kind of resource they can be executed allowing automated design space exploration. An optional software platform can be mapped to a resource e.g. if the resource is modelling a CPU running an operating system.

2.3 Platform

Model for resources is simplified allowing higher simulation speed. Resources are characterized by their performance to execute operations per clock cycle, operating frequency, maximum operating frequency, area, power consumption etc.

3 MODELING FOR TRANSACTION GENERATOR

Workload model for Transaction Generator is described in eXtensible Markup Language (XML) which has the benefit of being readable for humans also. SystemC code is generated automatically from the XML source file for simulation. Figure 2 presents the major tags and figure 3 expands the <*task*> tag from figure 2.

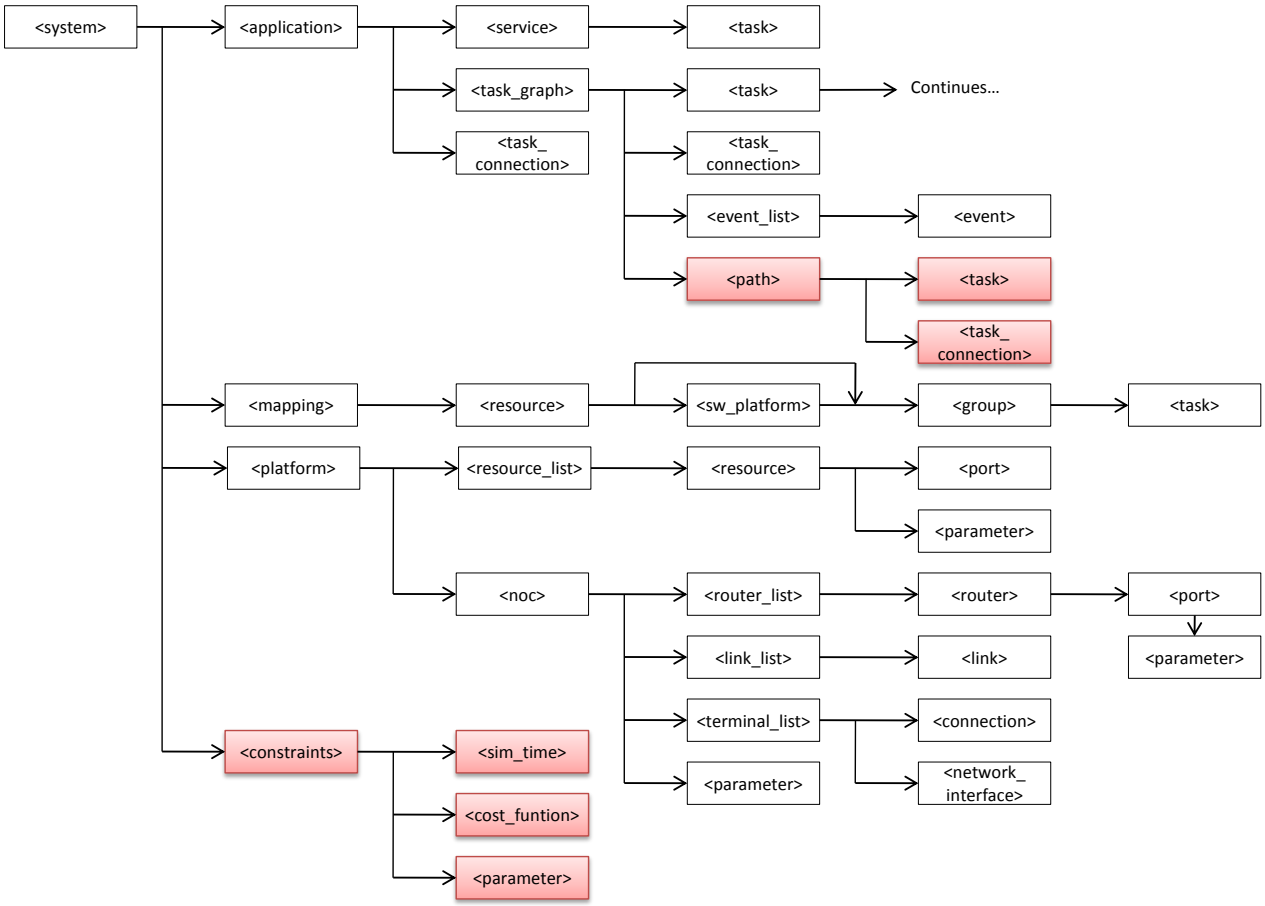

Figure 2: *XML tags used in modeling, part 1*

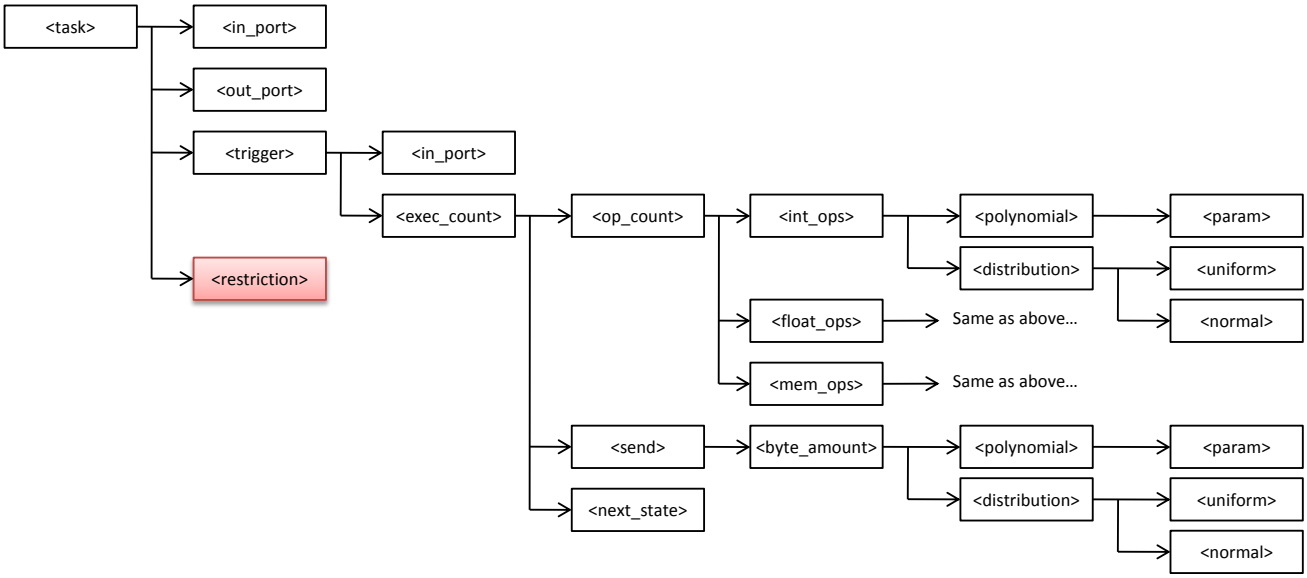

Figure 3: *XML tags used in modeling, part 2*

3.1 General modeling rules

Few important issues regarding the modeling syntax and semantics.

- Order of unrelated XML tags that belong to the same hierarchy level is free, but related tags e.g. <send> are executed in the order they appear in the file
- Id numbers of <task> tags in <task_graph> and <event> tags must be unique
- Id numbers of <task> tags not in <task_graph> are used as pointer and must refer to an id number of <task> tag in <task_graph>
- Id numbers of input and output ports used inside triggers must belong to the same task

3.2 XML document root

Example code below shows the fundamental elements that must be present in the source file. Following sections describe the tag structure for <application>, <mapping> and <platform> tags in detail. Order of these tags inside <system> is free but they must be defined only once.

Listing 1: *Root*

```
<?xml version='1.0'?>
<!DOCTYPE system_description>

<system xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="...">
  <xsm_version value="4"/>

  <application>
    .
    .
    .
  </application>

  <mapping>
    .
    .
    .
  </mapping>

  <platform>
    .
    .
    .
  </platform>

</system>
```

3.3 Application

Application model must have at least one <task_graph> which contains task models, the connections between them and events to start the application. Optional tags are <service> which can be used to group tasks belonging to a same "service" together and <task_connection>.

<task_connection> tags are used to connect events to tasks and tasks together to form the graph that represents application's communicational dependencies.

Listing 2: *Application*

```
<application>
  <task_graph>

    <task name="Task_0" id="0" class="general">
      <in_port  id="7"/>
      <out_port id="0"/>
        .
        .
    </task>
        .
        .
    <task name="Task_n" id="9" class="general">
        .
        .
    </task>

    <task_connection src="0" dst="1"/>
        .
        .
    <task_connection src="8" dst="9"/>


    <event_list>
      <event out_port_id="6" amount="1" trigger_type="periodic"
             name="Event_0" id="4" time="0.1" prob="1"/>
          .
          .
      <event out_port_id="8" amount="1" trigger_type="one-shot"
             name="Event_n" id="5" time="0.05" prob="1"/>
    </event_list>

  </task_graph>
</application>
```

3.3.1 Tasks

<task> tags inside <task_graph> are used to model the behavior of tasks. Tasks may contain any number of input or output ports and triggers which defines how to react to incoming data tokens. Tasks communicate through unidirectional ports with each other.

Id numbers of tasks cannot be used for event id numbers and vice versa. Also id numbers of input and output ports must be unique in the model.

Listing 3: *Task*

```
<task name="Task0" id="0" class="general">
  <in_port  id="5"/>
  <in_port  id="7"/>
  <out_port id="0"/>

  <trigger>
    <in_port id="5"/>
    <exec_count>
      <op_count>
        <int_ops>
          <polynomial>
            <param value="10000" exp="0"/>
          </polynomial>
        </int_ops>
      </op_count>
      <send out_id="0" prob="1">
        <byte_amount>
          <polynomial>
            <param value="1048576" exp="0"/>
          </polynomial>
        </byte_amount>
      </send>
      <next_state value="READY"/>
    </exec_count>
  </trigger>

  <trigger>
    <in_port id="7"/>
    <exec_count>
      <op_count>
        <int_ops>
          <polynomial>
            <param value="2000" exp="0"/>
          </polynomial>
        </int_ops>
      </op_count>
      <next_state value="READY"/>
    </exec_count>
  </trigger>

</task>
```

Listing 3 models a simple task that has two input ports and one output port. When task receives a data token from port 5 it first executes 10,000 integer operations and then sends a 1 MB data token to port 0. Received tokens from port 7 triggers task to execute 2,000 integer operations but not to send anything.

3.3.2 Triggers

Triggers are evaluated when all or one of the input ports have received a full data token. If trigger's dependency_type attribute is "and" it's evaluated after all ports have received data and if it's "or" it's evaluated when a data token is received to any of the trigger's input ports.

Triggers must have at least one input port and one exec_count tag.

Listing 4: *Trigger*

```
<trigger>
  <in_port id="7"/>
  <in_port id="8"/>
  <exec_count>
    <op_count>
      <int_ops>
        <polynomial>
          <param value="400" exp="0"/>
        </polynomial>
      </int_ops>
    </op_count>
    <send out_id="2" prob="1">
      <byte_amount>
        <polynomial>
          <param value="1024" exp="0"/>
        </polynomial>
      </byte_amount>
    </send>
    <send out_id="1" prob="0.5">
      <byte_amount>
        <polynomial>
          <param value="256" exp="0"/>
        </polynomial>
      </byte_amount>
    </send>
    <next_state value="READY"/>
  </exec_count>
</trigger>
<trigger dependence_type="and">
  <in_port id="10"/>
  <in_port id="11"/>
  <in_port id="12"/>
  <exec_count>
    .
    .
  </exec_count>
</trigger>
```

First trigger is evaluated whenever there is a data token in port 7 or 8. It executes 400 integer operations and after that sends 1024 bytes to port 2 and 256 bytes to port 1 with 50% probability.

Second trigger is evaluated only after all ports 10, 11 and 12 have received data tokens. If dependence_type attribute is not defined "or" is assumed.

3.3.3 Execution counts

Amount of operations task executes and bytes it sends can be different depending how many times it has been executed. Conditions of exec_counts are evaluated in the order they appear in the source file and the first one to have a condition that evaluates as true is executed. Rest of the exec_count conditions after the first one to return true are not evaluated.

Attribute mod_period is used to model periodical behavior e.g. for tasks that does every tenth time it's executed something differently. Attributes min and max are used to select a range of this period and using mod_phase is identical to using same value for both min and max.

Listing 5: *Execution counts (1)*

```
<trigger>
  <exec_count mod_phase="0" mod_period="3">
    ...
  </exec_count>
  <exec_count mod_phase="1" mod_period="3">
    ...
  </exec_count>
  <exec_count> <!-- default -->
    ...
  </exec_count>
</trigger>
```

Trigger in the first example executes the fisrt exec_count when task's execution count modulo three is zero. Second exec_count is executed when the modulo returns 1 and the third exec_count otherwise.

Listing 6: *Execution counts (2)*

```
<trigger>
  <exec_count min="1" max="4" mod_period="5">
    ...
  </exec_count>
  <exec_count> <!-- default -->
    ...
  </exec_count>
</trigger>
```

Second trigger example demonstrates the use of min and max. First exec_count is executed when when task's execution count modulo five is greater tha zero and the second exec_count when the result is zero.

Listing 7: *Execution counts (3)*

```
<trigger>
  <exec_count mod_phase="0">
    ...
  </exec_count>
  <exec_count mod_phase="1">
    ...
  </exec_count>
   .
   .
   .
  <exec_count mod_phase="n">
    ...
  </exec_count>
</trigger>
```

It's possible also to define a different behavior for every time the task is executed by not defining attribute mod_period.

3.3.4 Polynomial and distributional amounts

Listing 8 shows few different ways to determine the amount of operations to execute on HW resource or the amount of bytes to send when a certain trigger is being evaluated. In the example code amount of integer operations is $10x^2 + 100x + 400$ where x is the amount of bytes received. Number of floating point operations is a random number in range of 30 to 90 with uniform distribution. Amount of memory operations is randomly chosen using normal distribution. If the mean attribute is not defined for a normal distribution as in the second <send>, the amount of bytes received is used as a mean.

Listing 8: *Execution and send amounts*

```
<op_count>
  <int_ops>
    <polynomial>
      <param value="400" exp="0"/>
      <param value="100" exp="1"/>
      <param value="10"  exp="2"/>
    </polynomial>
  </int_ops>
  <float_ops>
    <distribution>
      <uniform min="30" max="90"/>
    </distribution>
  </float_ops>
  <mem_ops>
    <distribution>
      <normal mean="50" standard_deviation="5"/>
    </distribution>
  </mem_ops>
</op_count>

<send out_id="2" prob="1">
  <byte_amount>
    <polynomial>
      <param value="1024" exp="0"/>
    </polynomial>
  </byte_amount>
</send>

<send out_id="2" prob="1">
  <byte_amount>
    <distribution>
      <normal standard_deviation="100"/>
    </distribution>
  </byte_amount>
</send>
```

3.3.5 Events

Events are used to trigger the execution of task models either once or periodically. At least one event must exist to start the workload model to execute as all tasks are initially waiting for tokens to trigger them and thus the model wouldn't do anything without an event to trigger at least one of the tasks.

Id numbers of events cannot be used for task id numbers and vice versa.

Listing 9: *Events*

```
<event_list>

  <event id="7"
         out_port_id="1"
         amount="1"
         trigger_type="periodic"
         name="Event0"
         time="0.1" prob="0.3"/>

  <event id="8"
         out_port_id="2"
         amount="20"
         trigger_type="one-shot"
         name="Event1"
         time="1.0"
         prob="1"/>

   .
   .

</event_list>
```

In the example "Event0" sends 1 byte data token to port 1 every 0.1 seconds with 30% probability. First byte is sent at the moment the simulation starts. "Event1" is sending 20 bytes to port 2 when simulation has been run for 1.0 seconds with 100% probability.

3.4 Mapping

Mapping section binds software platforms to HW resources, group of tasks to either software platforms or directly to the resources and finally tasks to groups. Attributes "contents" and "position" are used in automatic design space exploration to determine whether or not not the optimization algorithm can e.g. move groups to different resources or alter the contents of groups by moving tasks.

Listing 10: *Mapping*

```
<mapping>

  <resource      name="cpu0"         id="0" contents="mutable">
    <sw_platform position="movable" id="0" contents="mutable">
      <group     position="movable" id="0" contents="mutable"
                 name="group0">

        <task position="movable" id="0" name="Task0"/>
        <task position="movable" id="1" name="Task1"/>
        <task position="movable" id="2" name="Task2"/>

      </group>
      <group     position="movable" id="1" contents="immutable"
                 name="group1">

        <task position="immovable" id="3" name="Task3"/>

      </group>
    </sw_platform>
  </resource>

  <resource name="acc1"           id="1" contents="immutable">
    <group  position="immovable" id="2" contents="immutable"
            name="group2">

      <task position="immovable" id="4" name="Task4"/>

    </group>
  </resource>

  .
  .
  .

</mapping>
```

In the example "cpu0" models a general purpose processor with a software platform and two groups of tasks. First group could be altered in any way in the exploration but the second could only be moved to another resource but not changed. Resource "acc1" has no software platform and can't be altered in any way.

3.5 Platform

Platform section defines the type of hardware resources, the type of NoC and how resources are connected to the NoC.

Listing 11: *Platform*

```
<platform>
  <resource_list>

    <resource id="0" name="cpu0" frequency="100" type="Generic_CPU">
      <port terminal="0"/>
    </resource>

      .
      .

    <resource id="9" name="acc9" frequency="100" type="Accelerator_x">
      <port terminal="9"/>
    </resource>

  </resource_list>

  <noc type="hibi">

    <router_list>
      <router width="32" id="0" name="Hibi_segment" frequency="80"
              type="Hibi_segment">
        <port name="hibi_p1" id="0" type="Hibi_if" address="0x100000"/>
          .
          .
        <port name="hibi_p9" id="9" type="Hibi_if" address="0x900000"/>
      </router>
    </router_list>

    <terminal_list>
      <connection port="0" router="0" name="hibi_p" id="0"/>
        .
        .
      <connection port="9" router="0" name="hibi_p" id="9"/>
      <network_interface type="Hibi_if"/>
    </terminal_list>

  </noc>
</platform>
```

3.5.1 Resources

Resource's type attribute maps it to an external HW library which defines its characteristics e.g. how many integer and floatingpoint operations it can execute in one clock cycle. Port tags determine how many interfaces the resource has to connect itself to a NoC or directly to another resource.

Values of terminals in port tags must be unique.

Listing 12: *Resources*

```
<resource_list>

  <resource id="0" name="cpu0" frequency="80" type="Generic_CPU">
    <port terminal="0"/>
  </resource>

  <resource id="1" name="cpu1" frequency="120" type="IP_CPU_y">
    <port terminal="1"/>
    <port terminal="2"/>
  </resource>

    .
    .

  <resource id="42" name="acc42" frequency="20" type="Accelerator_z">
    <port terminal="42"/>
  </resource>

</resource_list>
```

3.5.2 NoC

Example code models a HIBI bus with two segments bridged together. Router list defines all the routers or bus segments as in this case and the ports they have. Type attribute tells which network model to use in simulation. Terminal list connects routers' ports to resources' terminals and link list determines the connections between routers.

Listing 13: *NoC*

```
<noc type="hibi">

  <router_list>
    <router width="32" id="0" name="Hibi_segment0" frequency="25"
            type="Hibi_segment">

      <port name="hibi0_p0" id="0" type="Hibi_if" address="0x1000000"/>
      <port name="hibi0_p1" id="1" type="Hibi_if" address="0x2000000"/>
      <port name="hibi0_p2" id="2" type="Hibi_if" address="0x3000000"/>

    </router>
    <router width="32" id="1" name="Hibi_segment1" frequency="25"
            type="Hibi_segment">

      <port name="hibi1_p0" id="0" type="Hibi_if" address="0x4000000"/>
      <port name="hibi1_p1" id="1" type="Hibi_if" address="0x5000000"/>
      <port name="hibi1_p2" id="2" type="Hibi_if" address="0x6000000"/>

    </router>
  </router_list>

  <terminal_list>
    <connection port="0" router="0" name="hibi_p" id="0"/>
    <connection port="1" router="0" name="hibi_p" id="1"/>
    <connection port="2" router="0" name="hibi_p" id="2"/>
    <connection port="0" router="1" name="hibi_p" id="3"/>
    <connection port="1" router="1" name="hibi_p" id="4"/>
    <connection port="2" router="1" name="hibi_p" id="5"/>
    <network_interface type="Hibi_if"/>
  </terminal_list>

  <link_list>
    <link id="0"
          src_router="0" dst_router="1" src_port="2" dst_port="2"/>
  </link_list>

</noc>
```

4 TAGS

4.1 <system>

```
<system>
```

Root tag of the model.

Tags	Count	Notes
<application>	1	
<mapping>	1	
<platform>	1	
<constraints>	0	Read currently from a different file

Attributes	Type	Description
none		

4.2 <application>

```
<system>
  <application>
```

Tags	Count	Notes
<service>	0+	
<task_graph>	1+	
<task_connection>	0+	

Attributes	Type	Description
none		

4.3 <service>

```
<system>
  <application>
    <service>
```

Tags	Count	Notes
<task>	1+	

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
name	xs:string	Optional

4.3.1 service's <task>

```
<system>
  <application>
    <service>
      <task>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, refers to <task> in <task_graph>

4.4 <task_graph>

```
<system>
  <application>
    <task_graph>
```

Tags	Count	Notes
<task>	1+	
<task_connection>	1+	
<event_list>	1+	
<path>	0+	

Attributes	Type	Description
none		

4.5 <task>

```
<system>
  <application>
    <task_graph>
      <task>
```

Tags	Count	Notes
<in_port>	1+	
<out_port>	0+	
<trigger>	1+	
<restriction>	0+	Not implemented at the moment

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, must be unique in the goup of task_graph's <task> and <event> tags
name	xs:string	Optional
class	xs:string	Required

4.5.1 <in_port>

```
<system>
  <application>
    <task_graph>
      <task>
        <in_port>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, unique in group of task's <in_port> and <out_port> tags

4.5.2 <out_port>

```
<system>
  <application>
    <task_graph>
      <task>
        <out_port>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, unique in group of task's <in_port> and <out_port> tags

4.5.3 <trigger>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
```

Tags	Count	Notes
<in_port>	1+	
<exec_count>	1+	

Attributes	Type	Description
dependence_type	xs:string	Optional. Either “or” or “and”, default “or”

4.5.4 trigger's <in_port>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
          <in_port>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, id must be one of task's own in_port

4.5.5 <exec_count>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
          <exec_count>
```

Tags	Count	Notes
<op_count>	1	
<send>	0+	
<next_state>	1	

Attributes	Type	Description
min	xs:nonNegativeInteger	
max	xs:nonNegativeInteger	
mod_period	xs:nonNegativeInteger	
mod_phase	xs:nonNegativeInteger	

4.5.6 <op_count>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <op_count>
```

Tags	Count	Notes
<int_ops>	0..1	At least one of the ops tags must be present
<float_ops>	0..1	At least one of the ops tags must be present
<mem_ops>	0..1	At least one of the ops tags must be present

Attributes	Type	Description
none		

4.5.7 <int_ops>, <float_ops>, <mem_ops>, <byte_amount>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <op_count>
               <int_ops>
               <float_ops>
               <mem_ops>
             <send>
               <byte_amount>
```

One and only one of the tags must be present.

Tags	Count	Notes
<polynomial>	0..1	Mutually exclusive with <distribution>
<distribution>	0..1	Mutually exclusive with <polynomial>

Attributes	Type	Description
none		

4.5.8 <polynomial>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <op_count>
               <int_ops>
                 <polynomial>
               <float_ops>
                 <polynomial>
               <mem_ops>
                 <polynomial>
             <send>
               <byte_amount>
                 <polynomial>
```

Tags	Count	Notes
<param>	1+	

Attributes	Type	Description
none		

4.5.9 <param>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <op_count>
               <int_ops>
                 <polynomial>
                   <param>
               <float_ops>
                 <polynomial>
                   <param>
               <mem_ops>
                 <polynomial>
                   <param>
             <send>
               <byte_amount>
                 <polynomial>
                   <param>
```

Tags	Count	Notes
none		

Attributes	Type	Description
exp	xs:nonNegativeInteger	Required
value	xs:double	Required

4.5.10 <distribution>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <op_count>
               <int_ops>
                 <distribution>
               <float_ops>
                 <distribution>
               <mem_ops>
                 <distribution>
             <send>
               <byte_amount>
                 <distribution>
```

One and only one of the tags must be present.

Tags	Count	Notes
<uniform>	0..1	Mutually exclusive with <normal>
<normal>	0..1	Mutually exclusive with <uniform>

Attributes	Type	Description
none		

4.5.11 <uniform>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <op_count>
               <int_ops>
                 <distribution>
                   <uniform>
               <float_ops>
                 <distribution>
                   <uniform>
               <mem_ops>
                 <distribution>
                   <uniform>
             <send>
               <byte_amount>
                 <distribution>
                   <uniform>
```

Tags	Count	Notes
none		

Attributes	Type	Description
min	xs:positiveInteger	Required
max	xs:positiveInteger	Required

4.5.12 <normal>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <op_count>
               <int_ops>
                 <distribution>
                   <normal>
               <float_ops>
                 <distribution>
                   <normal>
               <mem_ops>
                 <distribution>
                   <normal>
             <send>
               <byte_amount>
                 <distribution>
                   <normal>
```

Tags	Count	Notes
none		

Attributes	Type	Description
mean	xs:positiveInteger	Optional, if not specified the input amount is used as mean
standard_deviation	xs:double	Required, value must be greater than zero

4.5.13 <send>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
           <exec_count>
             <send>
```

Tags	Count	Notes
<byte_amount>	1	

Attributes	Type	Description
out_id	xs:nonNegativeInteger	Required, must be one of task's own output ports
prob	xs:double	Required, probability for sending, value from zero to one

4.5.14 <next_state>

```
<system>
  <application>
    <task_graph>
      <task>
        <trigger>
          <next_state>
```

Tags	Count	Notes
none	0	

Attributes	Type	Description
value	xs:string	Required, possible values “FREE” and “READY”

4.5.15 <restriction>

```
<system>
  <application>
    <task_graph>
      <task>
        <restriction>
```

Not implemented at the moment.

Tags	Count	Notes
none		

Attributes	Type	Description
none		

4.6 <task_connection>

```
<system>
  <application>
    <task_connection>
    <task_graph>
      <task_connection>
```

Tags	Count	Notes
none	0	

Attributes	Type	Description
src	xs:nonNegativeInteger	Required, source port's id
dst	xs:nonNegativeInteger	Required, destination port's id

4.7 <event_list>

```
<system>
  <application>
    <task_graph>
      <event_list>
```

Tags	Count	Notes
<event>	1+	

Attributes	Type	Description
none		

4.7.1 <event>

```
<system>
  <application>
    <task_graph>
      <event_list>
        <event>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, must be unique in group of <event> and task_graph's <task> id numbers
time	xs:double	Required, time between sends
prob	xs:double	Required, probability for sending
out_port_id	xs:nonNegativeInteger	Required, must be one of task's own output ports
amount	xs:positiveInteger	Required, bytes to send
trigger_type	xs:string	Required, possible values "periodic" and "one-shot"
name	xs:string	Optional

4.8 <path>

```
<system>
  <application>
    <task_graph>
      <path>
```

Not implemented at the moment.

Tags	Count	Notes
<task>	0+	
<task_connection>	0+	

Attributes	Type	Description
none		

4.8.1 path's <task>

```
<system>
  <application>
    <task_graph>
      <path>
        <task>
```

Not implemented at the moment.

Tags	Count	Notes
none		

Attributes	Type	Description
none		

4.8.2 path's <task_connection>

```
<system>
  <application>
    <task_graph>
      <path>
        <task_connection>
```

Not implemented at the moment.

Tags	Count	Notes
none		

Attributes	Type	Description
none		

4.9 <mapping>

```
<system>
  <mapping>
```

Tags	Count	Notes
<resource>	1+	

Attributes	Type	Description
none		

4.9.1 <resource>

```
<system>
  <mapping>
    <resource>
```

Tags	Count	Notes
<group>	1+	Mutually exclusive with <sw_platform>
<sw_platform>	1+	Mutually exclusive with <group>

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
contents	xs:string	Required, possible values “immutable” and “mutable”
name	xs:string	Optional

4.9.2 <sw_platform>

```
<system>
  <mapping>
    <resource>
      <sw_platform>
```

Tags	Count	Notes
<group>	1+	

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
position	xs:string	Required, possible values “immovable” and “movable”
contents	xs:string	Required, possible values “immutable” and “mutable”
priority	xs:nonNegativeInteger	Optional

4.9.3 <group>

```
<system>
  <mapping>
    <resource>
      <group>
      <sw_platform>
        <group>
```

Tags	Count	Notes
<task>	1+	

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, refers to <task> in <task_graph>
position	xs:string	Required, possible values “movable” and “immovable”
contents	xs:string	Required, possible values “mutable” and “immutable”
name	xs:string	Optional

4.9.4 group's <task>

```
<system>
  <mapping>
    <resource>
      <group>
        <task>
      <sw_platform>
        <group>
          <task>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required, refers to <task> in <task_graph>
position	xs:string	Required, possible values "movable" and "immovable"
name	xs:string	Optional
priority	xs:nonNegativeInteger	Optional

4.10 <platform>

```
<system>
  <platform>
```

Tags	Count	Notes
<resource_list>	1	
<noc>	1	

Attributes	Type	Description
none		

4.10.1 <resource_list>

```
<system>
  <platform>
    <resource_list>
```

Tags	Count	Notes
<resource>	1+	

Attributes	Type	Description
none		

4.10.2 <resource>

```
<system>
  <platform>
    <resource_list>
      <resource>
```

Tags	Count	Notes
<port>	1+	
<parameter>	0+	

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
name	xs:string	Required
type	xs:string	Required, type refers to a resource description in HW library
frequency	xs:nonNegativeInteger	Optional

4.10.3 resource's <port>

```
<system>
  <platform>
    <resource_list>
      <resource>
        <port>
```

Tags	Count	Notes
none		

Attributes	Type	Description
terminal	xs:nonNegativeInteger	Required, refers to a <connection>'s id in <terminal_list>

4.10.4 <parameter>

```
<system>
  <platform>
    <resource_list>
      <resource>
        <parameter>
        <port>
          <parameter>
    <noc>
      <parameter>
      <router_list>
        <router>
          <port>
            <parameter>
  <constraints>
    <parameter>
```

<parameter> tags are used to pass technology dependent parameters.

Tags	Count	Notes
none		

Attributes	Type	Description
name	xs:string	Required
value	xs:string	Required

4.10.5 <noc>

```
<system>
  <platform>
    <noc>
```

Tags	Count	Notes
<router_list>	0+	
<link_list>	0+	
<terminal_list>	1	
<parameter>	0+	

Attributes	Type	Description
type	xs:string	Required

4.10.6 <router_list>

```
<system>
  <platform>
    <noc>
      <router_list>
```

Tags	Count	Notes
<router>	0+	

Attributes	Type	Description
none		

4.10.7 <router>

```
<system>
  <platform>
    <noc>
      <router_list>
        <router>
```

Tags	Count	Notes
<port>	1+	

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
type	xs:string	Required
frequency	xs:nonNegativeInteger	Required
width	xs:positiveInteger	Required
name	xs:string	Optional

4.10.8 router's <port>

```
<system>
  <platform>
    <noc>
      <router_list>
        <router>
          <port>
```

Tags	Count	Notes
<parameter>	0+	

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
type	xs:string	Optional
name	xs:string	Optional
width	xs:positiveInteger	Optional
address	xs:string	Optional

4.10.9 <link_list>

```
<system>
  <platform>
    <noc>
      <link_list>
```

Tags	Count	Notes
<link>	0+	

Attributes	Type	Description
default_width	xs:positiveInteger	Optional

4.10.10 <link>

```
<system>
  <platform>
    <noc>
      <link_list>
        <link>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
src_router	xs:nonNegativeInteger	Required
dst_router	xs:nonNegativeInteger	Required
src_port	xs:nonNegativeInteger	Required
dst_port	xs:nonNegativeInteger	Required
name	xs:string	Optional
width	xs:potiveInteger	Optional

4.10.11 <terminal_list>

```
<system>
  <platform>
    <noc>
      <terminal_list>
```

Tags	Count	Notes
<connection>	1+	
<network_interface>	1	

Attributes	Type	Description
none		

4.10.12 <connection>

```
<system>
  <platform>
    <noc>
      <terminal_list>
        <connection>
```

Tags	Count	Notes
none		

Attributes	Type	Description
id	xs:nonNegativeInteger	Required
router	xs:nonNegativeInteger	Required, refers to router's id
port	xs:nonNegativeInteger	Required, refers to port's id
name	xs:string	Optional
address	xs:string	Optional

4.10.13 <network_interface>

```
<system>
  <platform>
    <noc>
      <terminal_list>
        <network_interface>
```

Tags	Count	Notes
none		

Attributes	Type	Description
type	xs:string	Required
name	xS:string	Optional

4.11 <constraints>

```
<system>
  <constraints>
```

Read currently from a different file.

Tags	Count	Notes
<sim_time>	1	
<cost_function>	1	
<parameter>	0+	

Attributes	Type	Description
none		

4.11.1 <sim_time>

```
<system>
  <constraints>
    <sim_time>
```

Tags	Count	Notes
none		

Attributes	Type	Description
value	xs:positiveInteger	Required, simulation time in nanoseconds

4.11.2 <cost_function>

```
<system>
  <constraints>
    <cost_function>
```

Tags	Count	Notes
none		

Attributes	Type	Description
value	xs:string	Required