Verifying NoC communication architectures with ACL2

Julien Schmaltz, Amr Helmy, Laurence Pierre

Radboud University Nijmegen, The Netherlands
TIMA - VDS Group, Grenoble, France

ACL2 2007, Nov. 15-16

Platform-Based Design and Networks on a Chip

- Platform-Based Design:
  - Re-use of parameterized modules (*Intellectual Properties*)
  - High-level of abstraction
  - *Communication*-centric: from buses to networks
- System Verification:
  - Proof of each component
  - Proof of their interconnection
- State-of-the-Art:
  - Model checking or theorem proving of instances of systems
  - Often at RTL and below
- The GeNoC Approach
  - A generic model for reasoning about NoCs

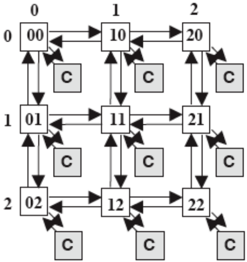
The GeNoC Model

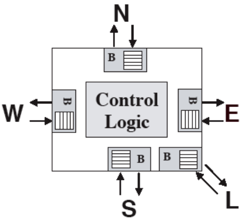
- Meta-model
    - Network topology and size
    - Routing algorithms
    - Switching techniques
- High-level of Abstraction
    - Abstract view of *Transport* (4) and *Network* (3) of the OSI model
- Encoded in the ACL2 theorem prover
    - Functional formalism
    - Parameterized proofs

An NoC Example: the Hermes NoC

- Result from cooperation between Catholic University of Rio Grande do Sul and LIRMM
- Public distribution
- FPGA implementation
- Protocol: credit based control flow
- Routing: deterministic minimal XY routing in a 2D-mesh
- Scheduling: wormhole switching

XY Routing and Wormhole Switching





- XY minimal deterministic routing
- Frame structure:
  - Header flit (Route Information)
  - Control flit (Number of Flits)
  - Data flits (Payload)

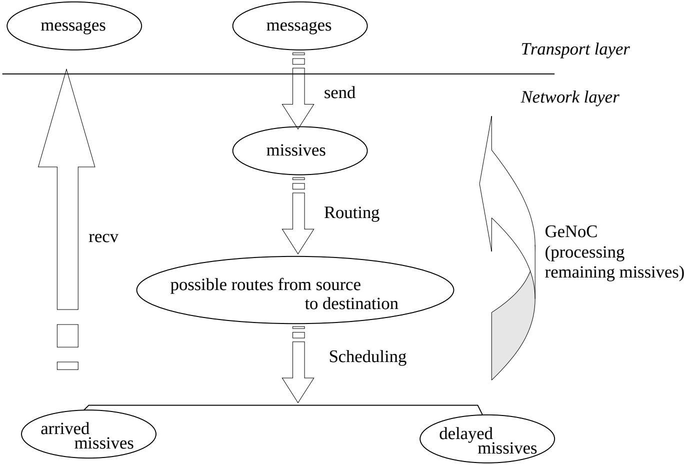
DATA	NOF	HD

GeNoC and Hermes

- Initial model
  - Nodes associated with coordinates
  - Several messages cannot occupy a node simultaneously
  - Messages are atomic
  - No explicit notion of time
- Extensions [NOCs 2007]
  - Nodes have coordinates as well as ports
  - A node can be occupied by several messages
  - Support for non-atomic messages (flits)
- Application to Hermes
  - Modeling and validation using the extensions
  - Simulation in ACL2 $\neq$ Simulation VHDL
  - Messages blocked at origin if conflicts

GeNoC Definition

```
( defun GeNoC_t (M NodeSet att TrLst)
  (declare (xargs :measure (SumOfAttempts att)))
  (if (zp (SumOfAttempts att))
      (mv TrLst M)
    (let ((V (Routing M NodeSet))) ;; comp. all routes
      (mv-let (Scheduled Delayed newAtt)
              ;; scheduled = reach dest.
              (Scheduling V att)
              (GeNoC_t (ToMissives Delayed) NodeSet newAtt
                       (append Scheduled TrLst))))))
```


messages

messages

Transport layer

Network layer

send

missives

Routing

recv

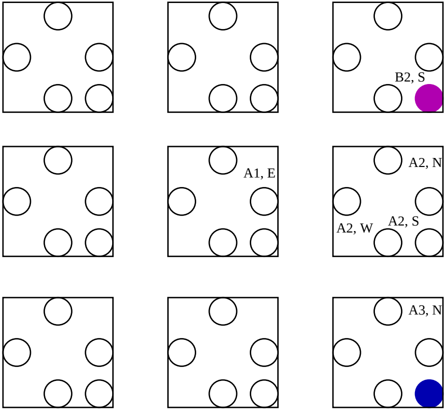
possible routes from source to destination

GeNoC (processing remaining missives)

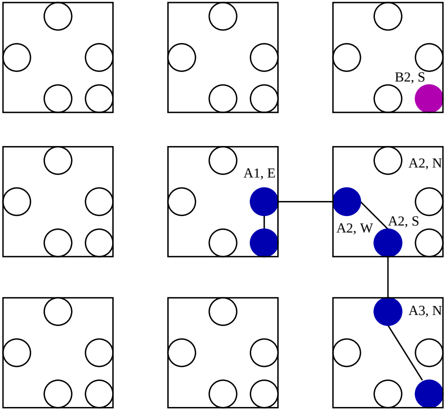
Scheduling

arrived missives

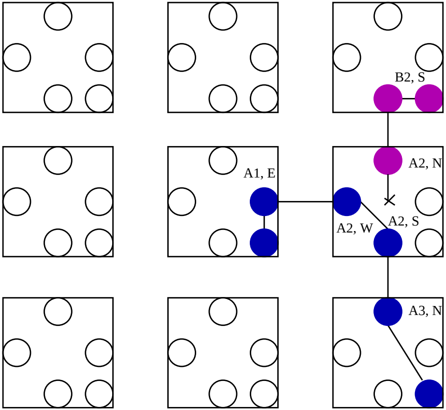
delayed missives



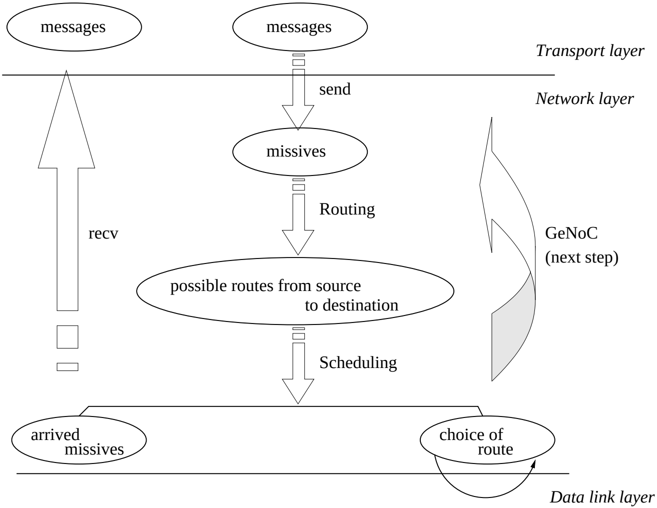
- Scheduled missives reach destination in one computation
- If conflicts, blocking missives stay home, they do not stay blocked in the network
- All missives sent at the same time



- Scheduled missives reach destination in one computation
- If conflicts, blocking missives stay home, they do not stay blocked in the network
- All missives sent at the same time



- Scheduled missives reach destination in one computation
- If conflicts, blocking missives stay home, they do not stay blocked in the network
- All missives sent at the same time


messages
messages
Transport layer
send
Network layer
missives
Routing
recv
GeNoC
(next step)
possible routes from source
to destination
Scheduling
arrived
missives
choice of
route
Data link layer

GeNoC: The Future

- Simulation ACL2 = Simulation VHDL
  - Improved step-by-step simulation
  - Global network state
- Deadlock avoidance
  - Protocol and structural
  - Control flow mechanism (credits)
- Link with RTL !
  - Data link layer (2) of OSI model