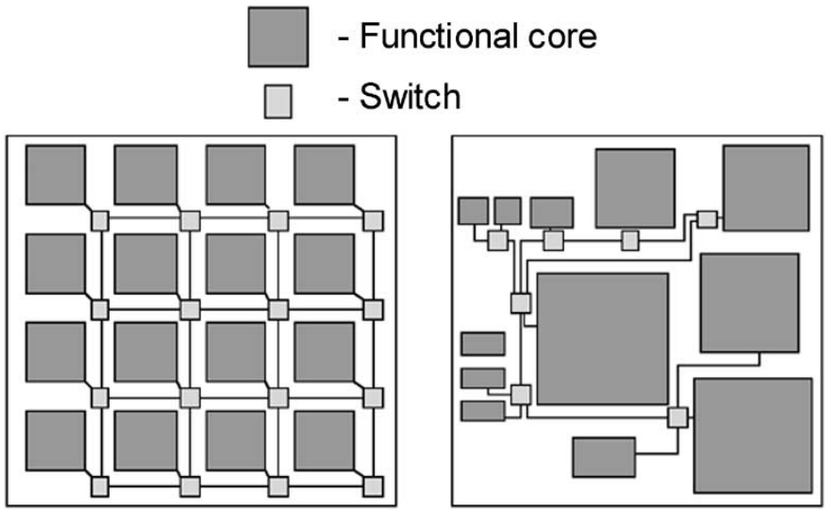
Automated Deadlock Verification in Register Transfer Level Designs of Communication Fabrics

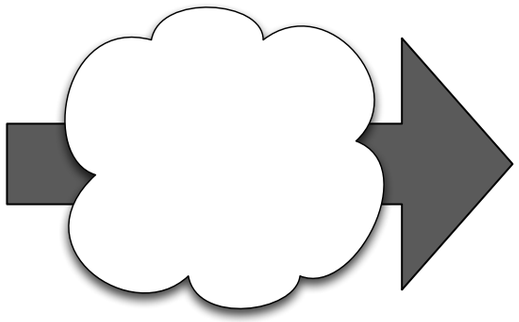
Sebastiaan J.C. Joosten
Julien Schmaltz

Open Universiteit
www.ou.nl

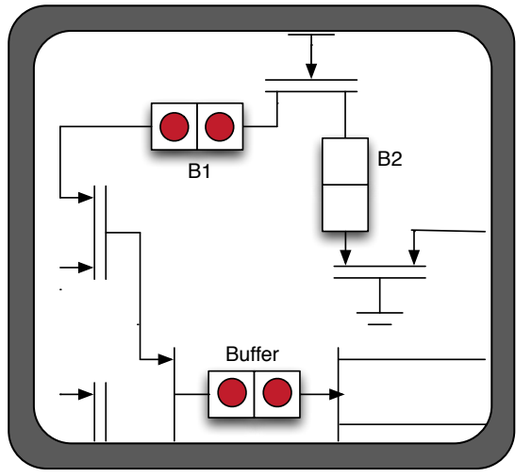
Automated Deadlock Verification in Register Transfer Level Designs of Communication Fabrics



RTL design



Our approach



(candidate) deadlock configurations

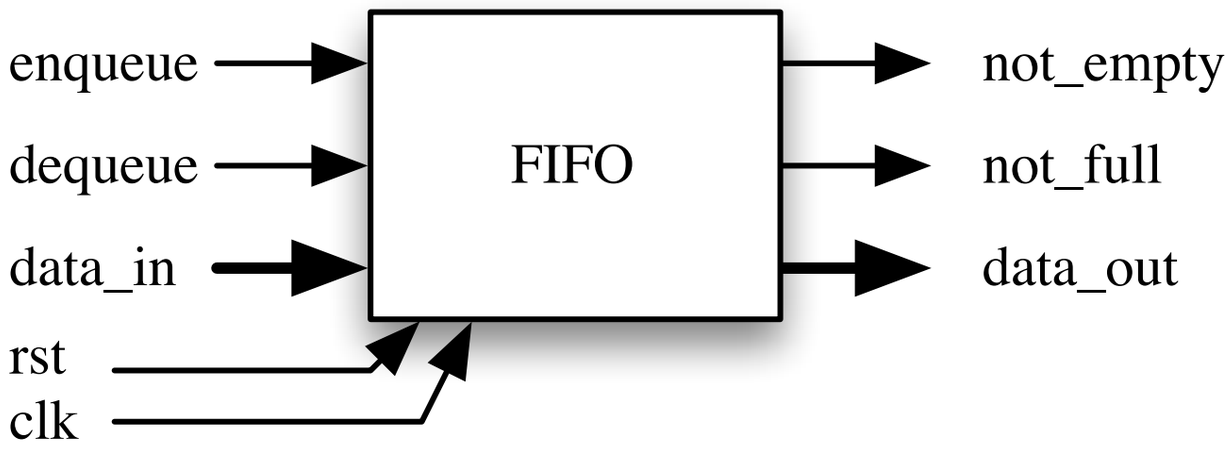
Sebastiaan J.C. Joosten

Our approach

- Sound for deadlock freedom / Complete for finding deadlocks
- Fast because:
  - abstract from queues (using Verilog module structure)
  - use off-the shelve SAT solvers
  - find static deadlock configurations (just one state!)

Encoding deadlocks = Encoding persistency

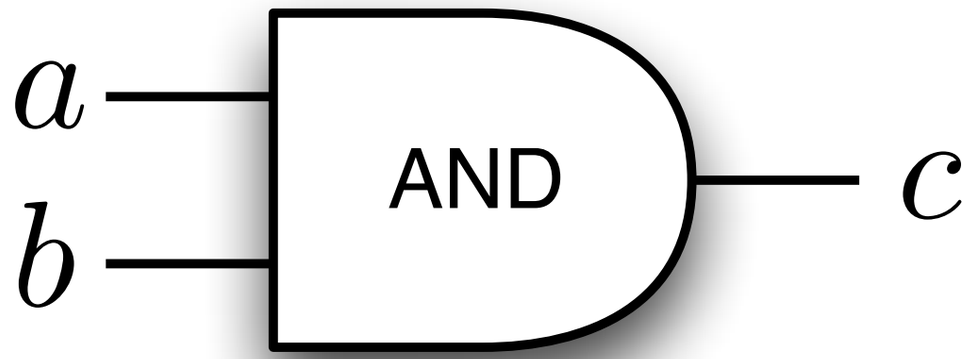
- A dead queue is one that never releases its packet



$$\Diamond\Box\,(\neg \texttt{dequeue} \wedge \texttt{not\_empty})$$

Encoding persistency

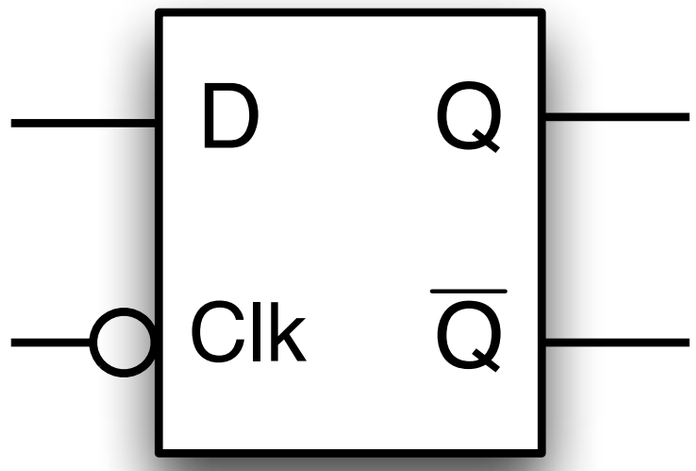
- Persistency can be propagated over the network



$$\Diamond\Box a \wedge \Diamond\Box b \rightarrow \Diamond\Box c$$

Encoding persistency

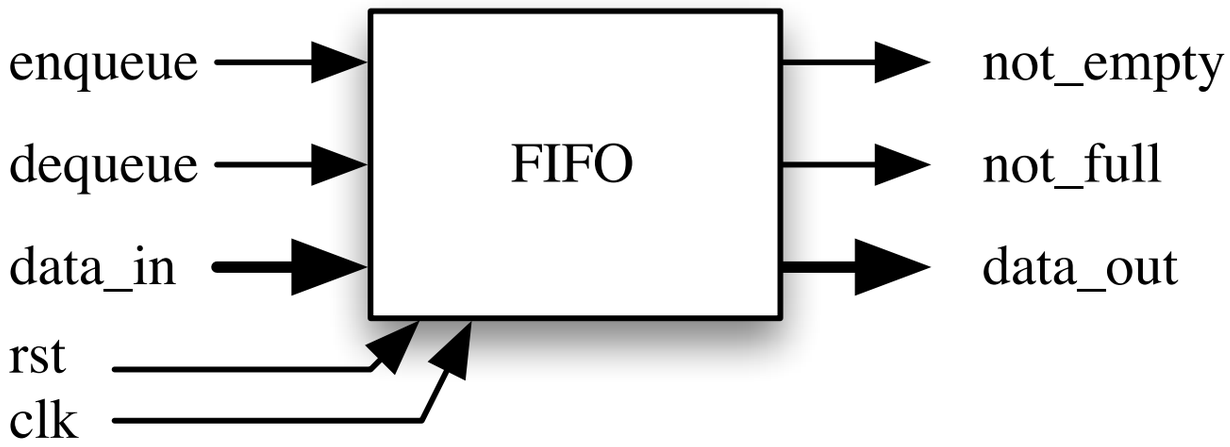
- Persistency can be propagated over the network



$$\Diamond\Box D \rightarrow \Diamond\Box Q$$

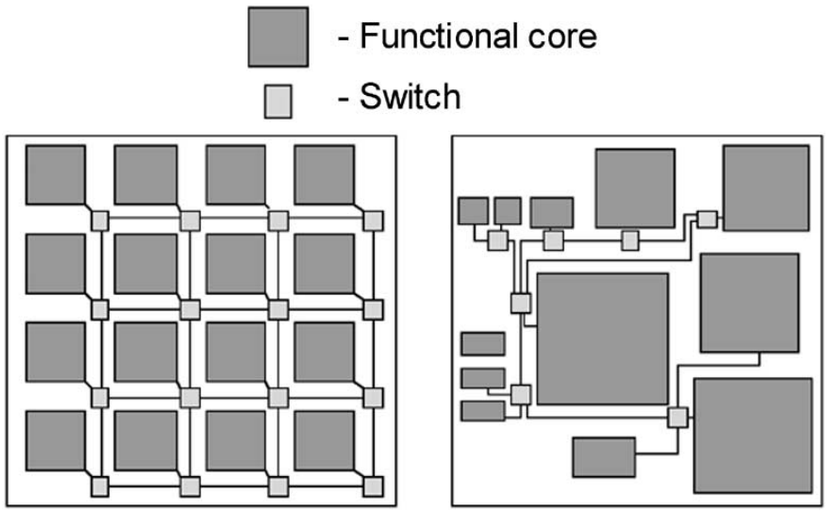
Encoding persistency

- Persistency can be propagated over the network

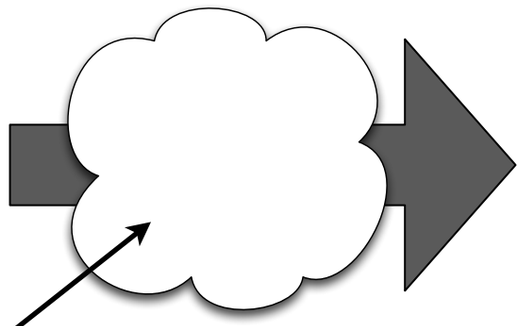


$$\Diamond\Box \texttt{enqueue} \rightarrow \Diamond\Box \texttt{not\_empty}$$

Automated Deadlock Verification in Register Transfer Level Designs of Communication Fabrics



RTL design



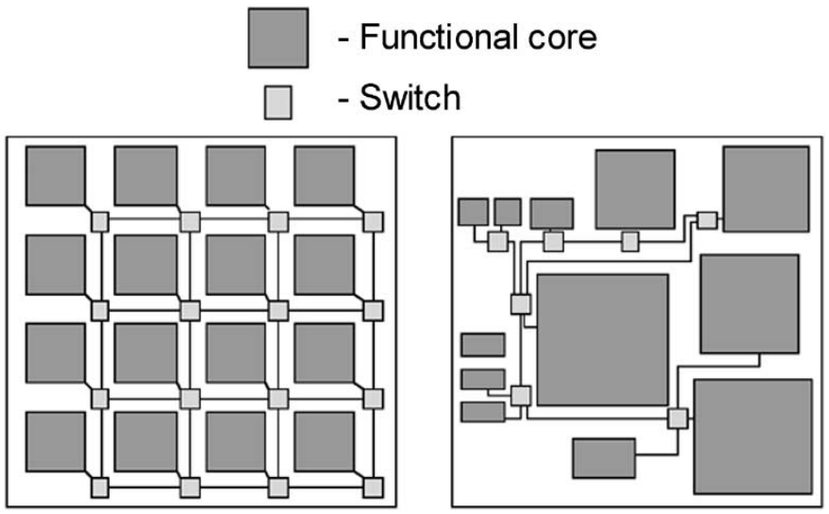
Our approach

automatic proof of deadlock freedom

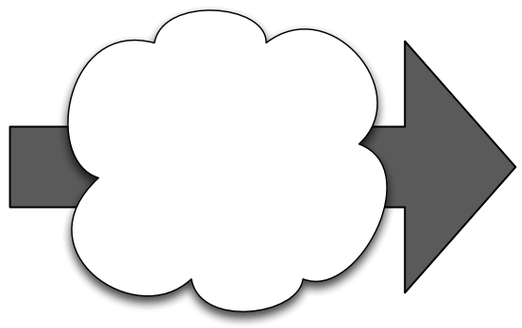
- Find essential properties of the design
- Find restrictions for compositional verification

Sebastiaan J.C. Joosten

Automated Deadlock Verification in Register Transfer Level Designs of Communication Fabrics



RTL design



Our approach

automatic proof of deadlock freedom

Sebastiaan J.C. Joosten

Check out my poster

NWO Netherlands Organisation for Scientific Research

intel

Open Universiteit www.ou.nl

Deadlock Verification in Register Transfer Level Designs of Communication Fabrics

This research is funded by NWO, Project 612.001.108 Effective Layered Verification of Networks-on-Chips

Sebastiaan J.C. Joosten
sjj@ou.nl

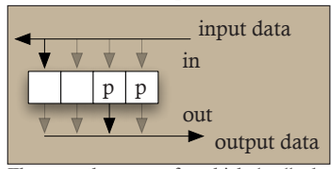
Julien Schmaltz
Julien.Schmaltz@ou.nl

Verilog files

- Synthesisable verilog with explicit clock and reset signals
- Automatic conversion by Centaur Technology

Mark queues and parse

- Provide a list of queue modules



These are the queues for which deadlocks (or absence thereof) will be found.

EMOD Module

Logic

Registers

Queues

- Abstract away from queues and registers
- Add properties instead of their implementation

- For every wire: two Boolean variables
- one for the wire itself
- one to indicate whether it is `stuck':

$$w_{\mathrm{p}} = \Diamond\Box w \vee \Diamond\Box\neg w$$

Add deadlock equation to all properties

- For each queue, check if it is dead:

$$\mathbf{dead}(q) \triangleq \Diamond\Box(\neg q_{\mathrm{o}}.\mathrm{a} \wedge q_{\mathrm{o}}.\mathrm{r})$$

- SMT formulation:

$$= \neg q_{\mathrm{o}}.\mathrm{a} \wedge q_{\mathrm{o}}.\mathrm{r} \wedge q_{\mathrm{o}}.\mathrm{a_p} \wedge q_{\mathrm{o}}.\mathrm{r_p}$$

By taking the disjunction of all these, we obtain a condition yielding a configuration if there is a deadlock.

SMT instance

The solver returns either:
- Proof of deadlock absence
- Possible deadlock

Instance	# q	# w	generation	solving
2x2 mesh	20	1952	6 sec	1 sec
4x4 mesh	80	5281	42 sec	3 sec
6x6 mesh	180	13972	866 sec	4 sec
8x8 mesh	320	24515	4618 sec	4 sec

Experimental results for networks with deadlocks

Instance	# q	# w	generation	solving
2x2 mesh	28	940	3 sec	0 sec
4x4 mesh	112	4768	68 sec	1 sec
6x6 mesh	252	13380	1660 sec	7 sec
8x8 mesh	448	23872	9148 sec	11 sec

Experimental results for deadlock-free networks

Context

- Communication fabrics constitute a key component of multicore processors and systems-on-chip.
- Detection of message dependent deadlocks in communication fabrics is a challenge due to the large number of queues and the distributed character of control.

Motivation

- Verification of deadlock freedom of Register Transfer Level designs of communication fabrics.

Contribution

- Reduce queues to an abstract entity
- Convert deadlock freedom to an SMT instance
- Implemented approach in ACL2
- Approach scales to large fabrics
- Sound but incomplete: false deadlocks may be found

Future work:

- Reduce generation time
- Reduce false deadlocks by adding invariants

Sebastiaan J.C. Joosten