
AUTOMATED VERIFICATION OF ASYNCHRONOUS CIRCUITS USING CIRCUIT PETRI NETS

**Ivan Poliakov, Andrey Mokhov, Danil Sokolov,
Ashur Rafiev, Alex Yakovlev**

Asynchronous Systems Group

School of Electronic, Electric and Computer Engineering

Newcastle University

Outline

- Motivation
- Circuit Petri nets
- Environment composition
- Verification method
- Worckraft framework
- Benchmarks
- Conclusions

Introduction

Approach

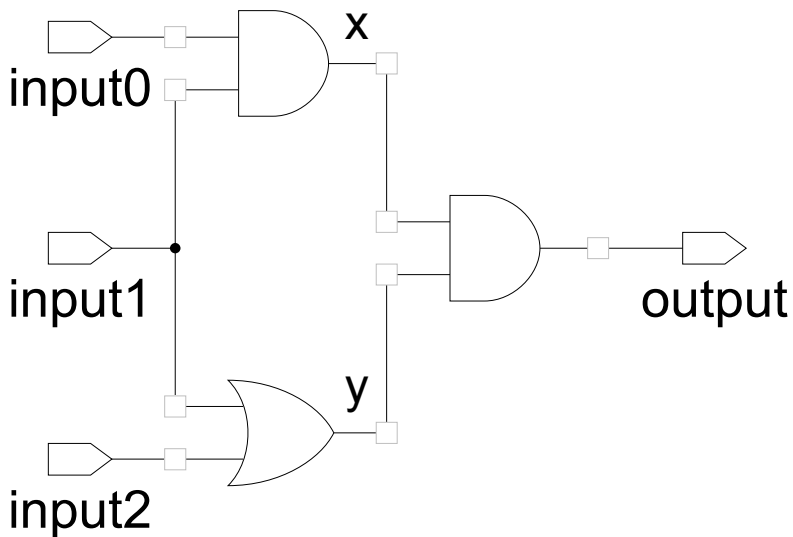
Representing the circuit and its composition with the environment as a special type of a Petri net.

Motivation

An alternative, Petri nets based approach to verification of asynchronous circuits (most of the currently used methods employ state graphs and BDDs) meant to exploit recent advances in Petri net model checking methods, particularly those based on unfoldings.

Circuits

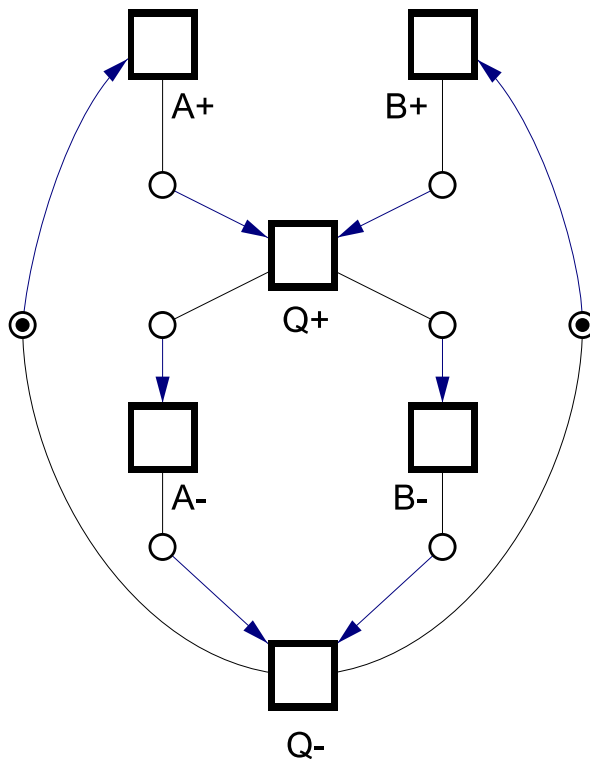
- A **circuit** C is a triple $C = \langle V, F, s_0 \rangle$ [Roig 97]
 - V is a set of signals
 - F is a mapping $V \rightarrow f$, f is a logical function (gate) driving the signal
 - s_0 is the initial state of the signals



$x = \text{input0 AND input 1}$
 $y = \text{input1 OR input 2}$
 $\text{output} = x \text{ AND } y$

Signal Transition Graphs (STG)

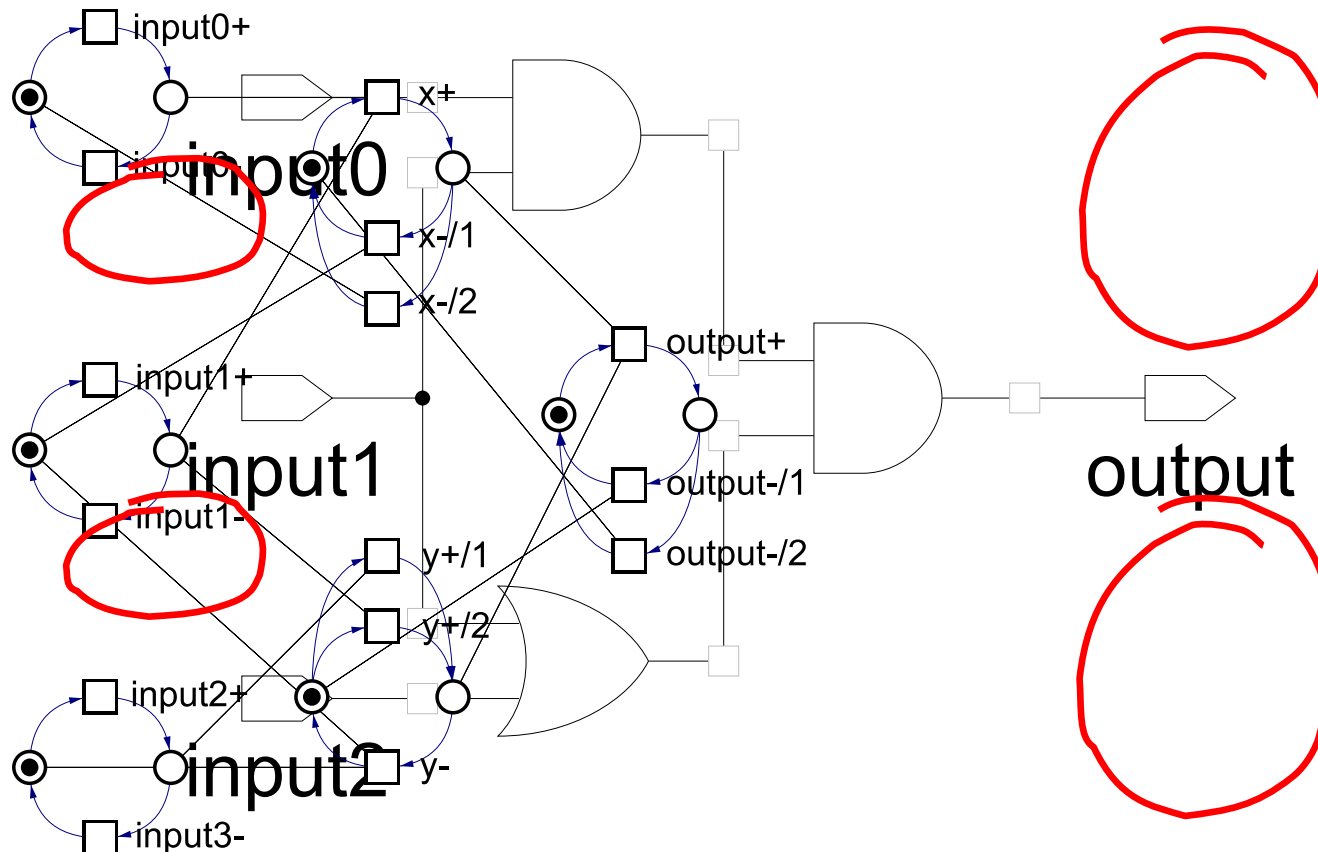
- A **signal transition graph** (STG) is a Petri net where each transition is labelled with a signal level change.



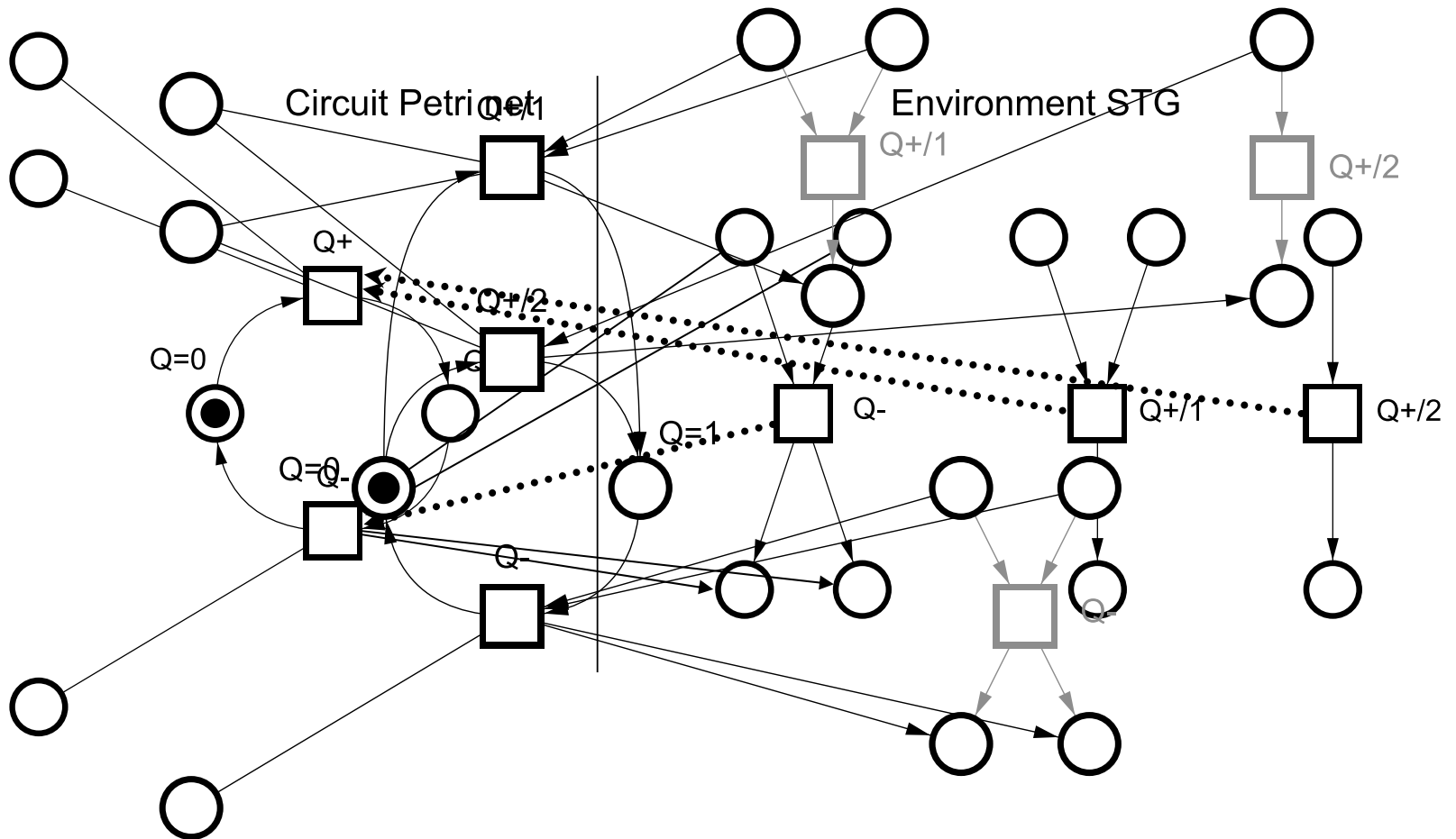
STG of C-element specification

Circuit Petri nets

- A **circuit Petri net** R associated with a circuit C is a type of STG that is constructed from the circuit.



Circuit and environment composition



Verification

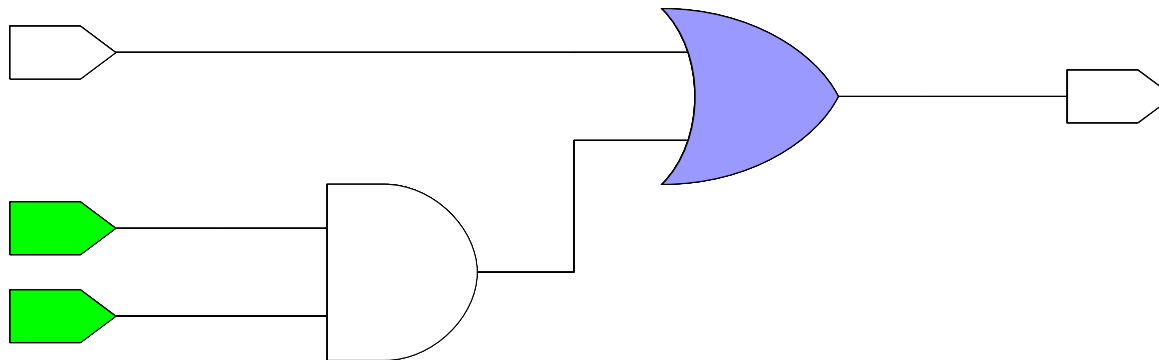
A circuit is considered speed-independent under a given environment, if

- It is hazard-free,
- It conforms to the environment, i.e. produces only those changes of output signals that do not conflict with the environment's STG.

(“environment conformance” definition will be provided later, and should not be confused with Dill’s definition)

Hazards

A **hazard** is defined to be an **unexpected change of the input signal** of a gate, such that it causes an **enabled** (positively or negatively excited) **gate to become disabled** (i.e. to return into a stable state without firing).

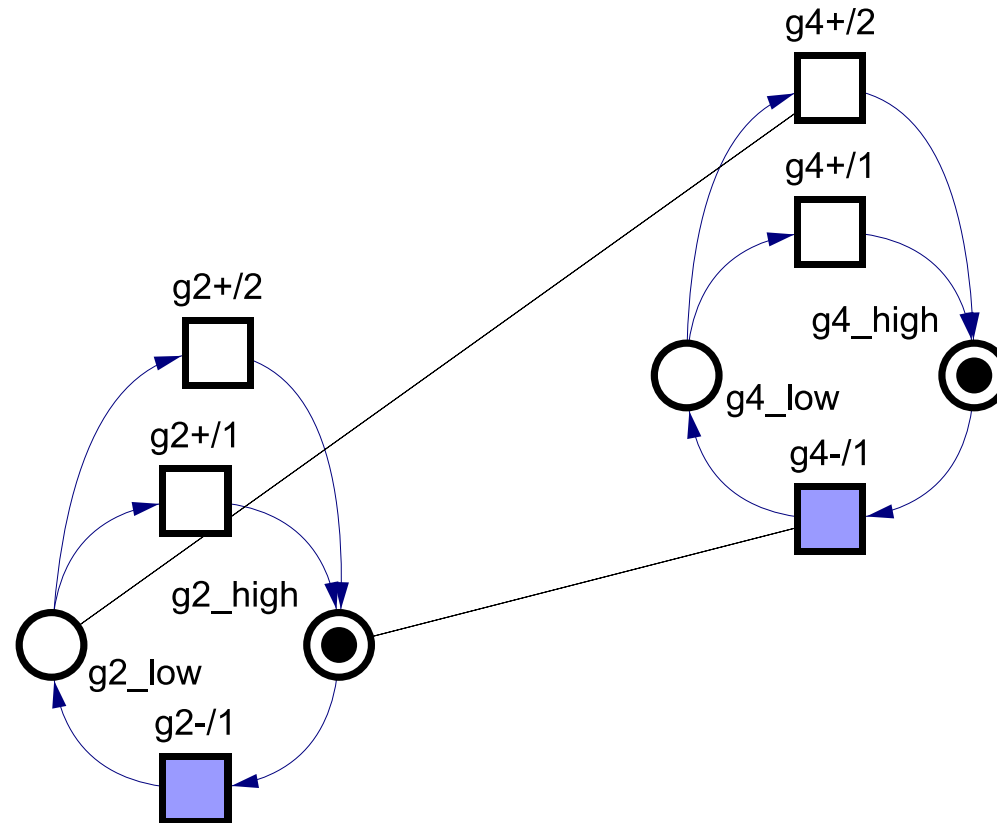


Detection of potential hazards


- A circuit is said to be **free from potential hazards** if the circuit Petri net constructed from it does not violate *semi-modularity* property:


The Petri net is semi-modular if, once each place in the preset of a transition has become marked with a token (enabling the transition), no other transition can remove any of these tokens, thus disabling the transition until it has fired.

Non-semi-modularity

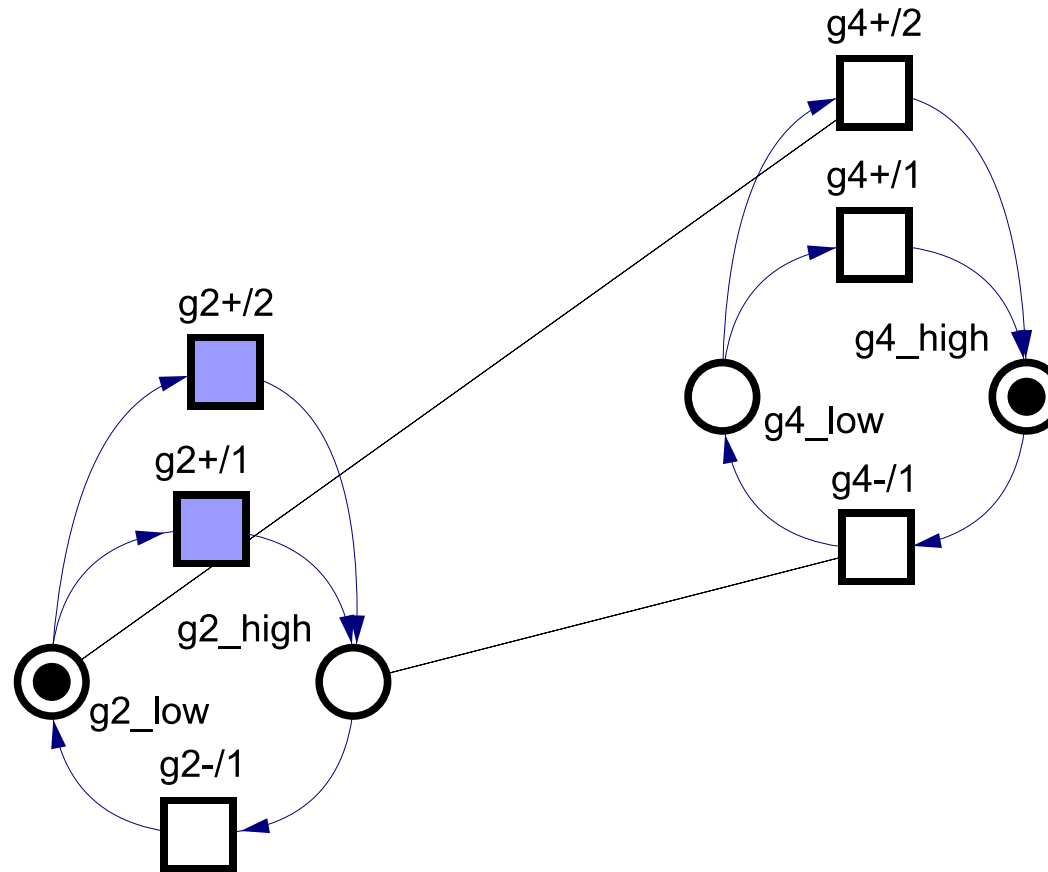


Detection of potential hazards

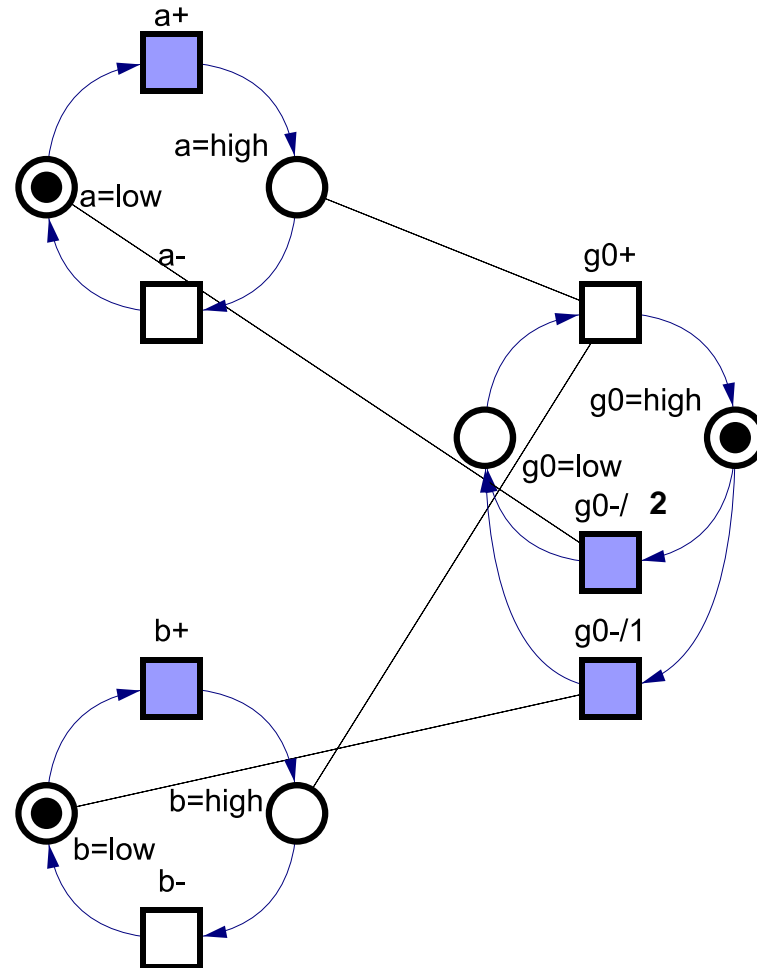
 If the circuit Petri net is semi-modular, then there are no potential hazards in the original circuit.

 However, if the Petri net is not semi-modular, this does not necessarily indicate the presence of a potential hazard.

Signal semi-modularity (1)



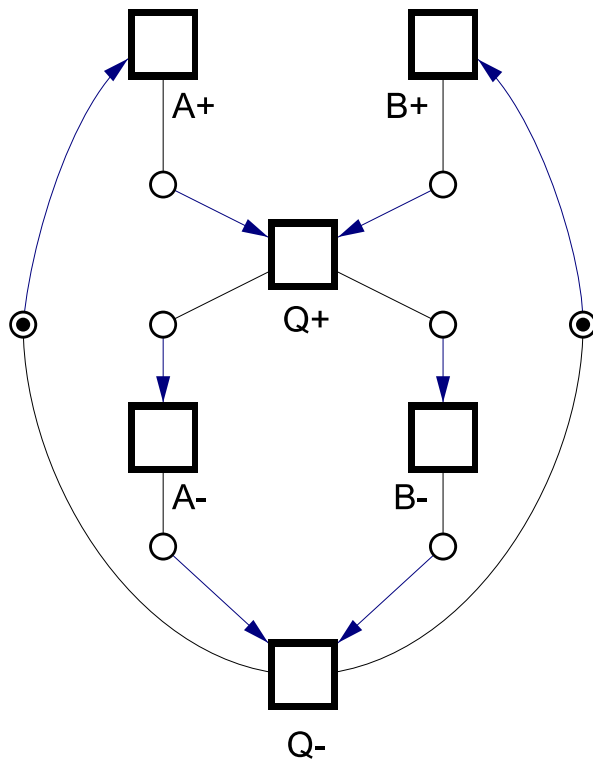
Signal semi-modularity (2)



Detection of potential hazards

- ✓ If the circuit Petri net is not semi-modular, but all non-semi-modular states are signal semi-modular, then the circuit the Petri net is built from is considered hazard-free.

Environment conformance (1)



C-element interface

Example 1: does AND gate conform to C-element interface?

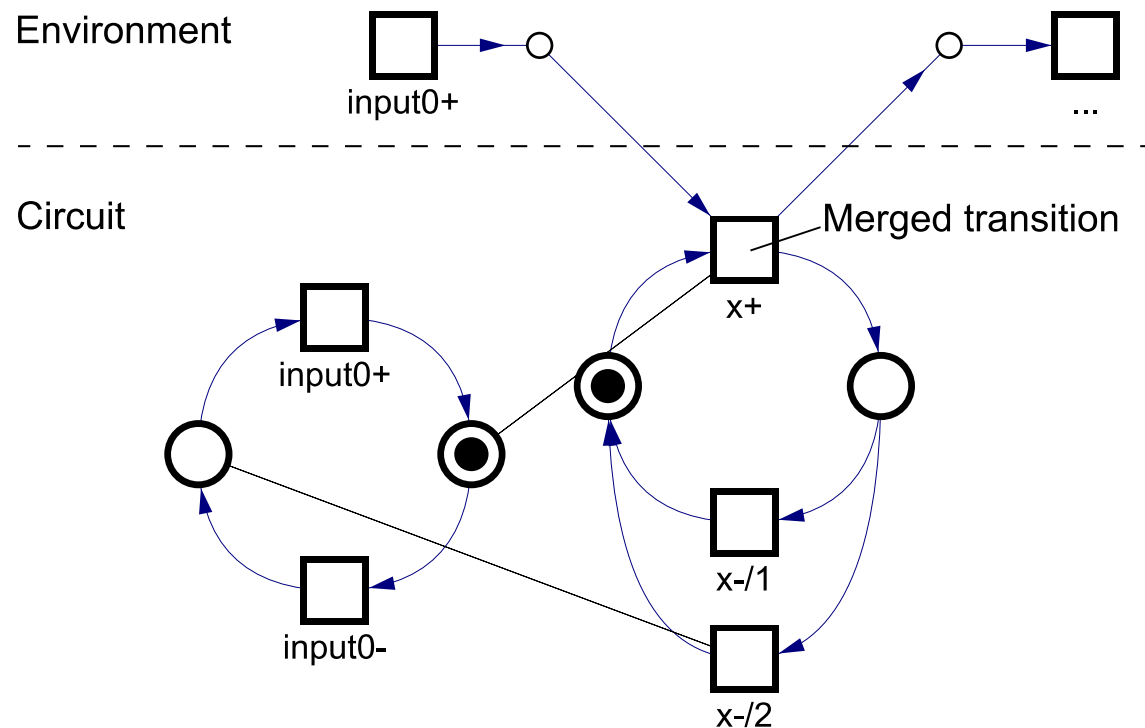
NO: after $\langle A+, B+, Q+, A- \rangle$ AND gate is ready to reset Q, while C-element interface is expecting B- to happen first.

Example 2: does XOR gate conform to C-element interface?

NO: after $\langle A+, B+ \rangle$ the system is deadlocked.

Environment conformance (2)

- ☹ The environment STG, when composed with the circuit PN, **restricts** the net from producing signal changes that are not expected by the environment.



Environment conformance (3)

☺ These situations can be detected, however, by solving a reachability problem:

If there exists a marking m in the compound PN, such that for some signal transition T that is present both in the environment STG and the circuit there are tokens in all of the places in $\bullet T$ in the circuit, but no tokens in any places in $\bullet T$ in the environment, then the circuit does not conform to that environment.

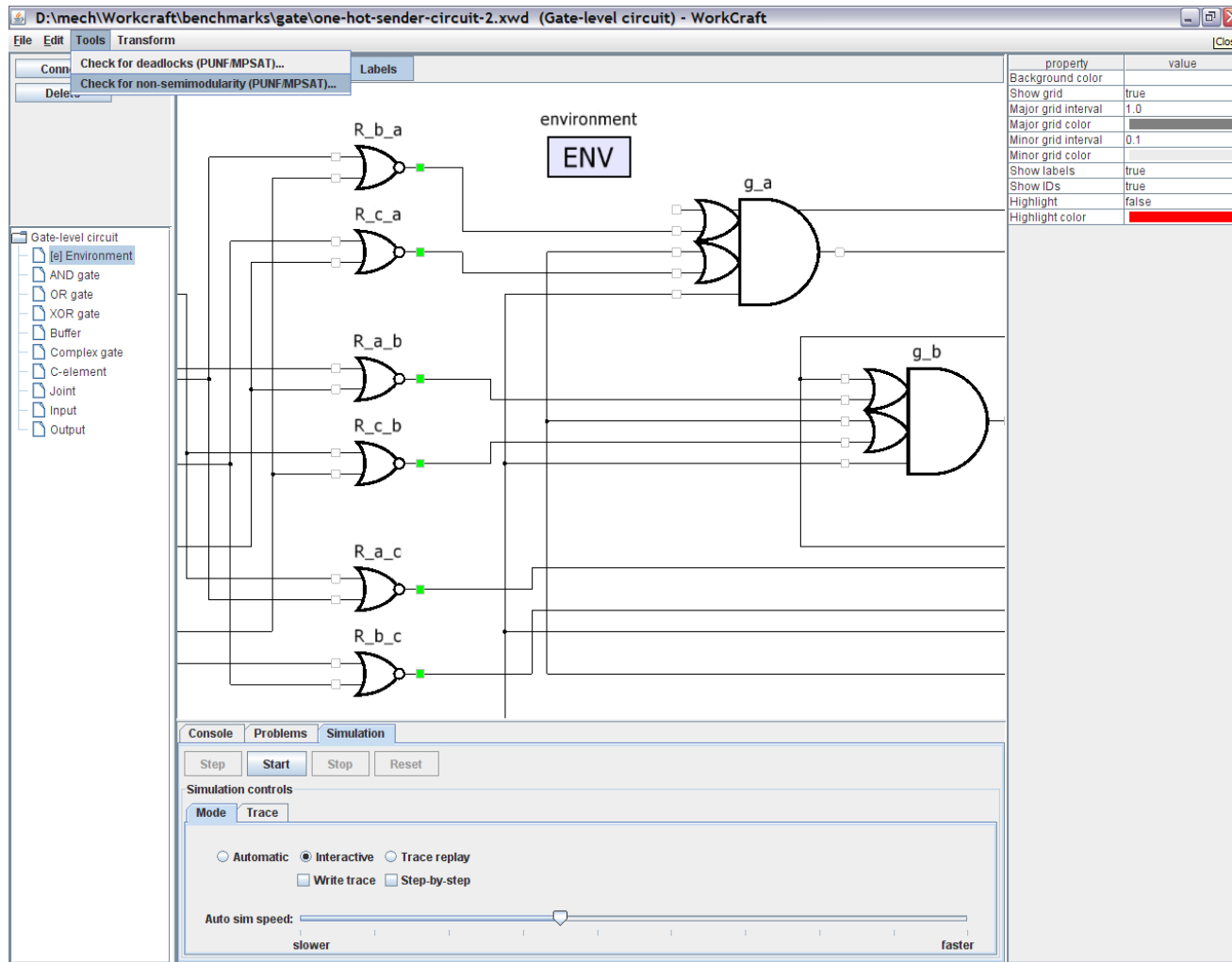
Environment conformance (4)

In the compound circuit/environment Petri net:

- ✓ If the net produced by composition of environment STG with the circuit PN obtained from gate-level circuit is strongly live and
- ✓ If there are no reachable markings leading to potential unexpected signal change as explained in previous slide

then the circuit conforms to the environment.

Workcraft framework (UI)



The screenshot displays the Workcraft software interface for a gate-level circuit. The main workspace shows a circuit diagram with several components: six NOT gates (R_b_a, R_c_a, R_a_b, R_c_b, R_a_c, R_b_c), two AND gates (g_a, g_b), and an environment block (ENV). The circuit is connected to a console and simulation controls at the bottom.

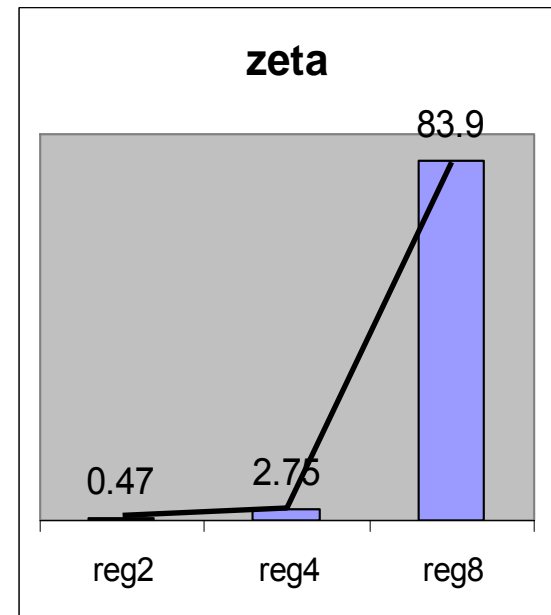
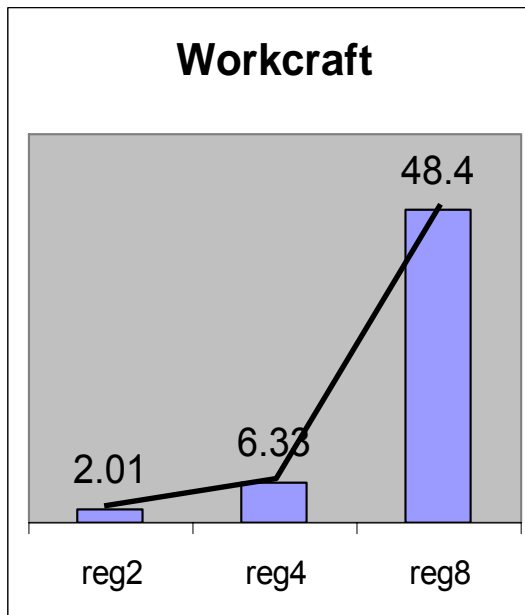
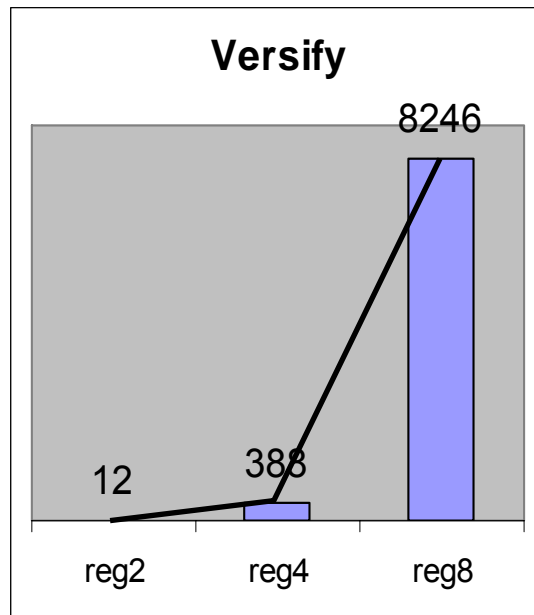
The console and simulation controls include:

- Buttons: Step, Start, Stop, Reset
- Simulation controls: Mode (Trace), Automatic, Interactive (selected), Trace replay, Write trace, Step-by-step
- Auto sim speed: A slider ranging from slower to faster.

On the right side, there is a table of properties:

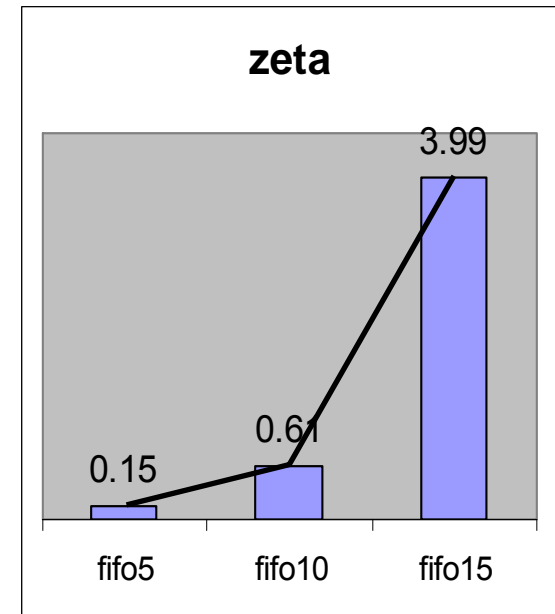
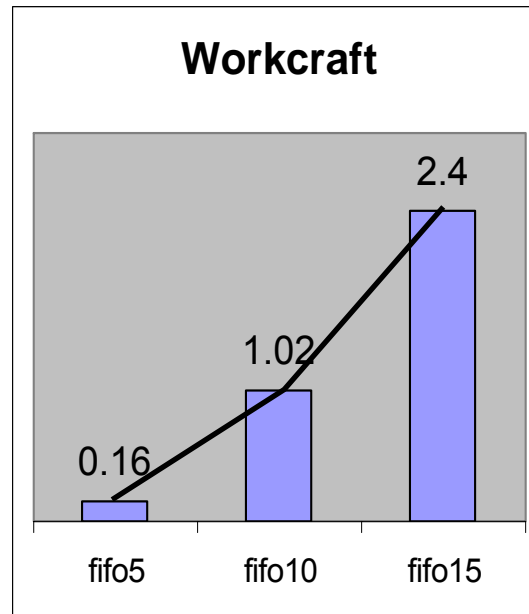
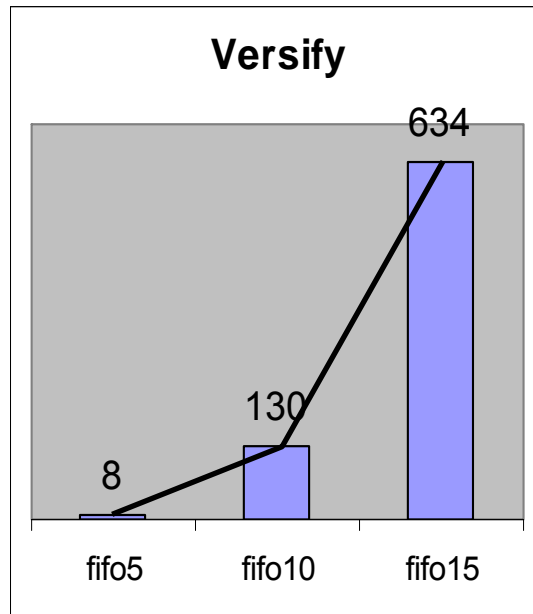
property	value
Background color	
Show grid	true
Major grid interval	1.0
Major grid color	
Minor grid interval	0.1
Minor grid color	
Show labels	true
Show IDs	true
Highlight	false
Highlight color	

Benchmarks (1)



Benchmark	States	Net size (P/T)	Unf. (evt./cutoffs)
reg2	$2.5 \cdot 10^4$	183/124	368/29
reg4	$7.6 \cdot 10^7$	337/220	2464/177
reg8	$7.1 \cdot 10^{14}$	649/416	72192/4865

Benchmarks (2)



Benchmark	States	Net size (P/T)	Unf. (evt./cutoffs)
fifo5	$2.6 \cdot 10^3$	97/58	86/1
fifo10	$1.2 \cdot 10^6$	177/108	166/1
fifo15	$5.8 \cdot 10^8$	257/158	246/1

Advantages and disadvantages

- ☺ Highly modular
- ☺ More visual
- ☺ 'Delegated model-checking' approach: using state-of-the-art model checking tools, but not bound to any particular one
- ☺ Significantly faster on certain class of benchmarks compared to well-known Versify tool (when using unfolding-based model checker)

- ☹ Unstable performance: a minor change in the initial state can lead to a drastic growth of the verification time (when using unfolding-based model checker)

Conclusions

- A workflow for verification of asynchronous circuits using Petri nets was developed
 - Implemented in Workcraft framework
 - Automatic transparent conversion into circuit PNs
 - Detection of deadlocks, potential hazards and interface non-conformance implemented using external model checking tools (PUNF/MPSAT)
 - Automatic bad trace parsing and propagation onto high-level model
 - Very high performance for certain circuit classes

End

Thank you!
Questions?