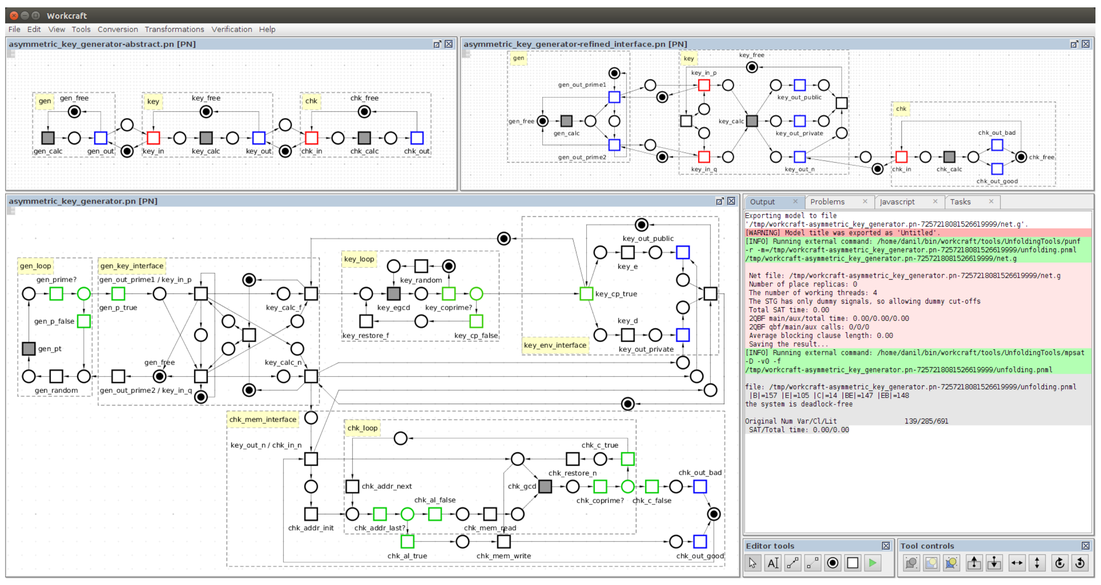
# Workcraft

http://workcraft.org/

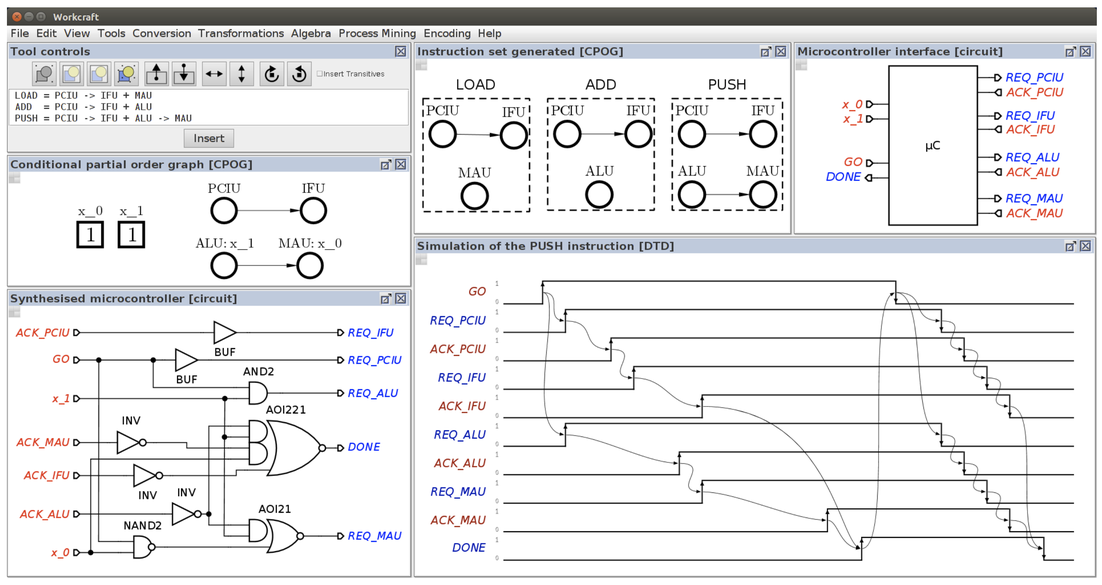
## Overview

- Verification-driven design of circuits based on formal graph models.
- Graphical frontend for visual editing, analysis, simulation and verification.
- Interoperability between different abstraction levels using Petri nets as a common language.
- Established backend tools for synthesis and model checking (*Petrify*, *MPSat*).
- Open source code and plugin-based architecture for new graph formalisms and analysis tools.

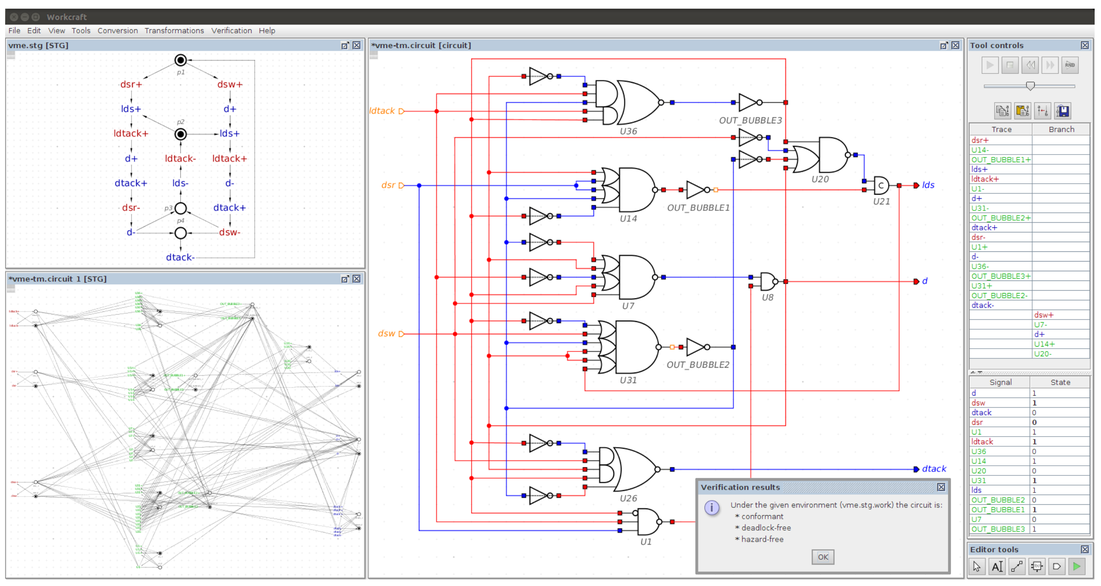
# Workcraft in Action



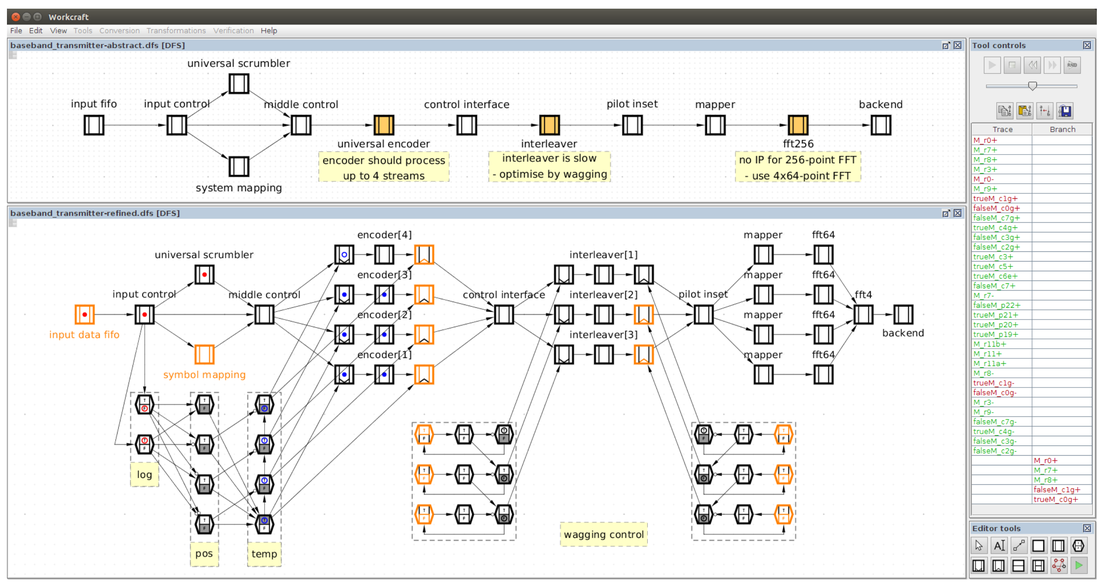

- Modelling and verification of concurrent algorithms with *Petri Nets*.




- Designing instruction decoders with Conditional Partial Order Graphs.

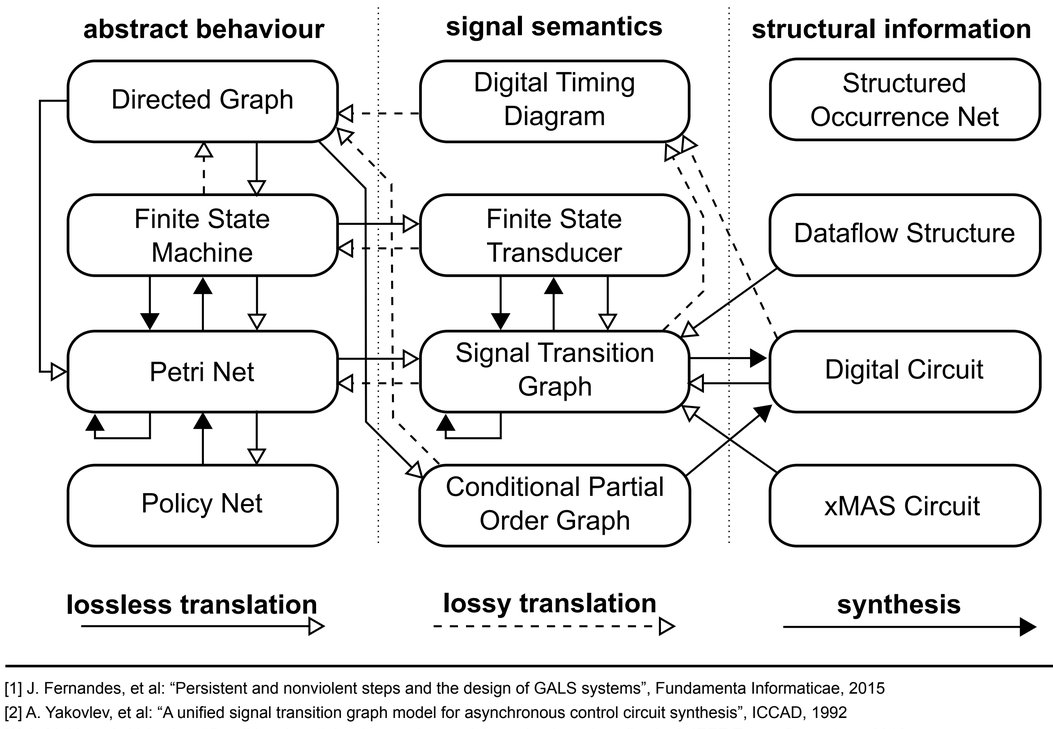



- Specification and synthesis of speed-independent controllers based on *Signal Transition Graphs* and *Circuit Petri nets*.




- Modelling self-timed pipelines with Spread-token and Counterflow execution semantics using *Dataflow Structures*.

# Supported Models




| Model | Editing | Simulation | Verification | Synthesis | |
|---|---|---|---|---|---|
| Directed Graph | ✓ | ✓ | ✓ | ○ | abstract behaviour |
| Finite State Machine | ✓ | ✓ | ✓ | ✓ | |
| Petri Net | ✓ | ✓ | ✓ | ✓ | |
| Policy Net [1] | ✓ | ✓ | ✓ | ○ | |
| Digital Timing Diagram | ✓ | ✗ | ○ | ○ | signal semantics |
| Finite State Transducer | ✓ | ✓ | ✓ | ✓ | |
| Signal Transition Graph [2] | ✓ | ✓ | ✓ | ✓ | |
| Conditional Partial Order Graph [3] | ✓ | ◐ | ✗ | ✓ | |
| Structured Occurrence Net [4] | ✓ | ✓ | ✓ | ○ | structural information |
| Dataflow Structure [5] | ✓ | ✓ | ✓ | ✗ | |
| Digital Circuit [6] | ✓ | ✓ | ✓ | ○ | |
| xMAS Circuit [7] | ✓ | ✓ | ◐ | ✗ | |

✓ – fully supported ◐ – partially implemented ✗ – work in progress ○ – not applicable


[1] J. Fernandes, et al: “Persistent and nonviolent steps and the design of GALS systems”, Fundamenta Informaticae, 2015
[2] A. Yakovlev, et al: “A unified signal transition graph model for asynchronous control circuit synthesis”, ICCAD, 1992
[3] A. Mokhov, A. Yakovlev: “Conditional partial order graphs: model, synthesis and application”, IEEE Trans. Computers, 2010
[4] M. Koutny, B. Randell: ”Structured occurrence nets: a formalism for aiding system failure prevention and analysis techniques”, Fundamenta Inf., 2009
[5] I. Poliakov, et al: “Automated verification of asynchronous circuits using circuit Petri nets”, ASYNC, 2008
[6] D.Sokolov, et al: “Analysis of static data flow structures”, Fundamenta Inf., 2008
[7] F. Burns, et al: “GALS synthesis and verification for xMAS models”, DATE, 2015
[8] D.Sokolov, et al: "Benefits of asynchronous control for analog electronics: multiphase buck case study", DATE, 2017 **(paper 12.6.1)**



EPSRC
Engineering and Physical Sciences Research Council

Asynchronous design for Analogue electronics (A4A), Dataflow Computation a la Carte,
UNderstanding COmplex system eVolution through structurEd behaviouRs (UNCOVER),
SElf-timed DATapath synthEsis (SEDATE), Globally Asynchronous Elastic Logic Synthesis (GAELS).

Newcastle University