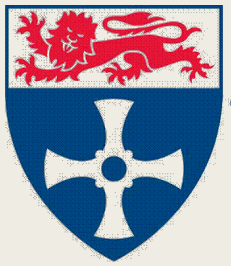


Newcastle University

Workcraft: A Framework For Interpreted Graph Models

Victor Khomenko

Interpreted Graph Models

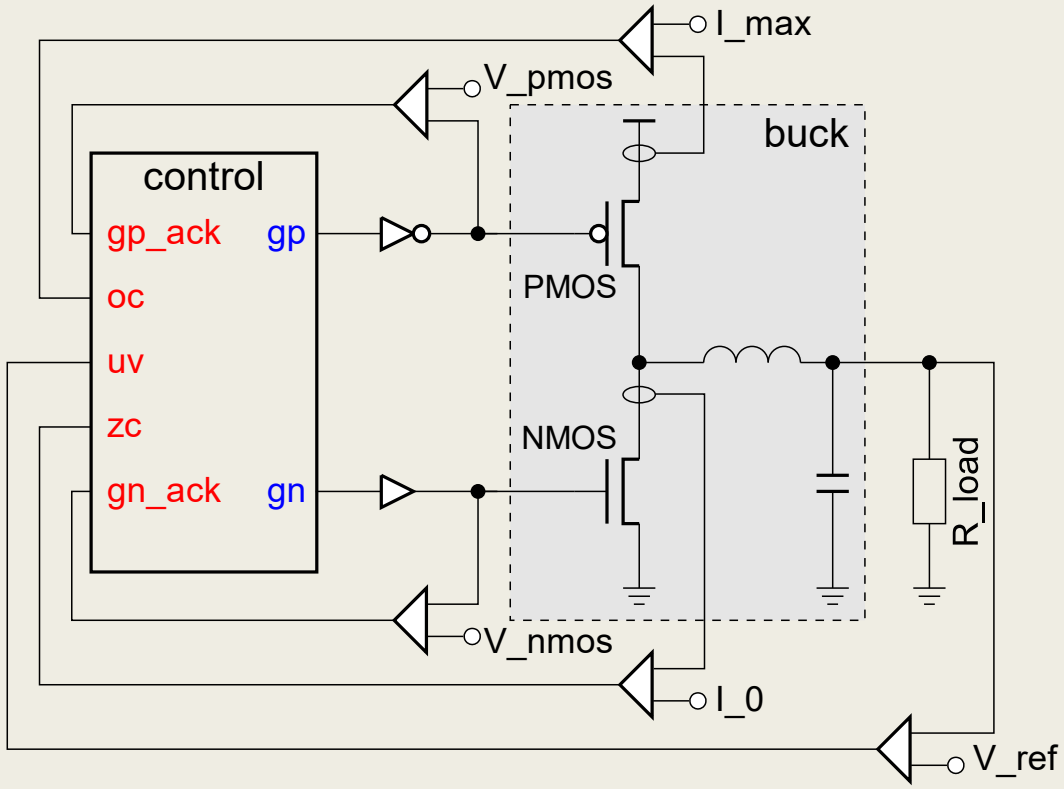
Static graph structure + dynamic aspect (tokens, signals, etc.) defining the behaviour, e.g.:

- Petri Net / Signal Transition Graph
- Finite State Machine
- Digital Circuit
- + many other

Visual editing, co-design, simulation, formal verification, synthesis

Demo 1: Basic buck controller

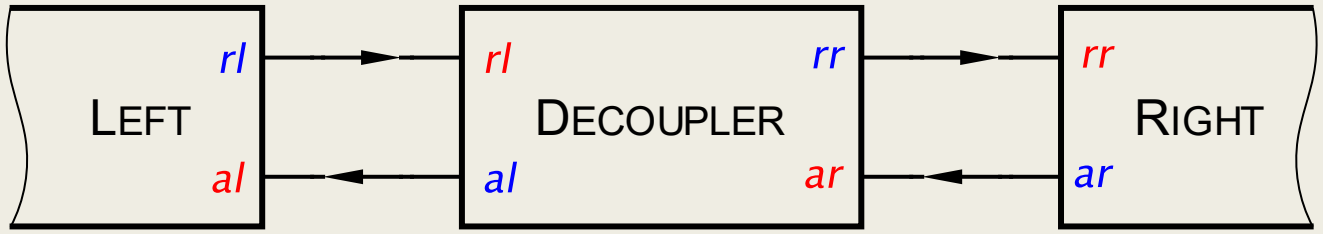
- QDI design flow:
  - STG specification
  - Validation via simulation
  - …

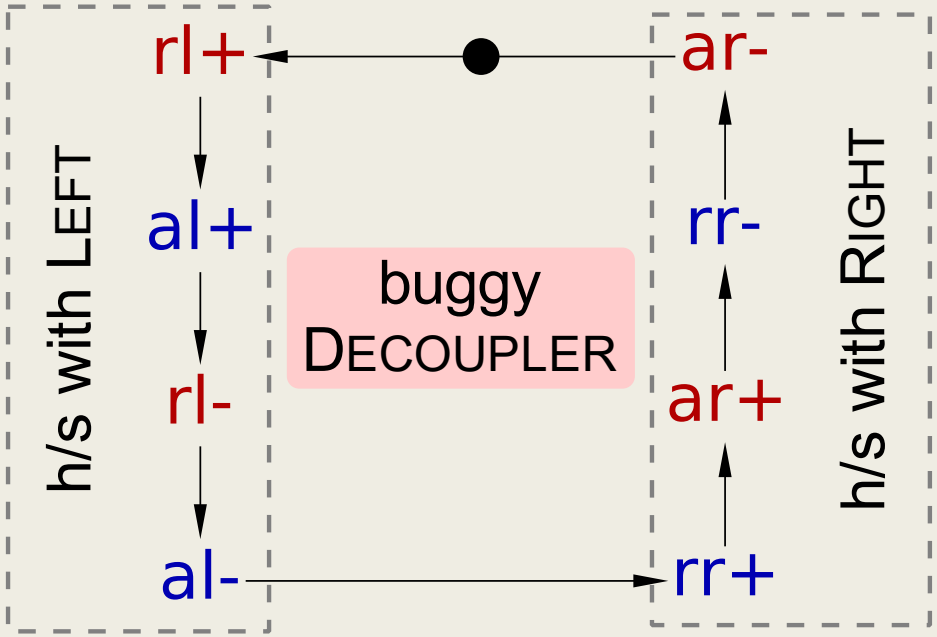


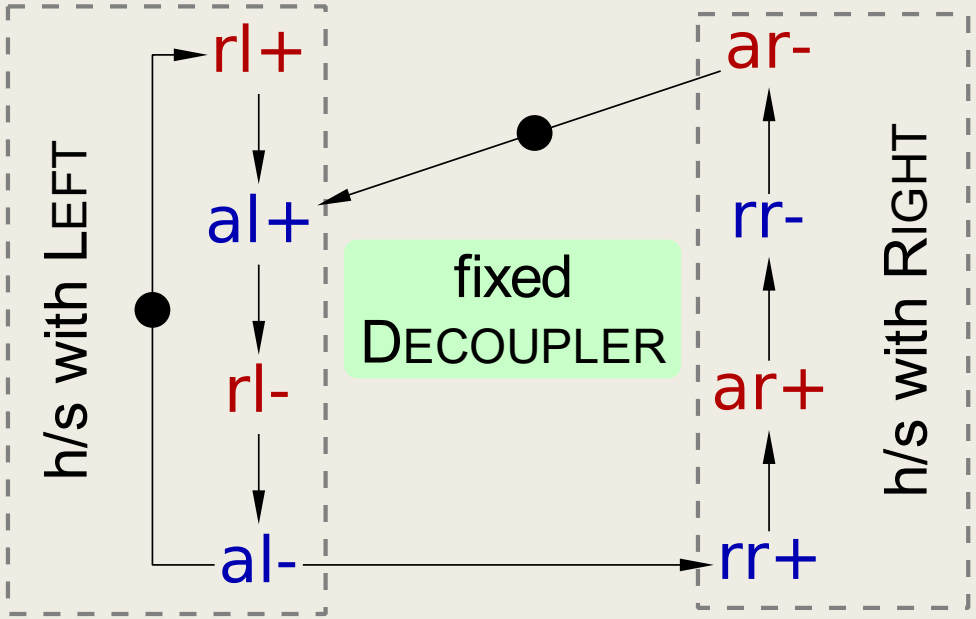
Demo 1: Formal Verification of STG

- Standard properties:
  - Consistency
  - Deadlock freeness
  - Output persistence
  - Input properness
  - Output determinacy
  - Mutex place implementability
  - Custom properties, e.g. `$S"gp" & $S"gn"`

Demo 1: Handshakes verification







Demo 1: Synthesis

- Complex-gate
- Generalised C-elements
- Standard-C
- Technology mapping

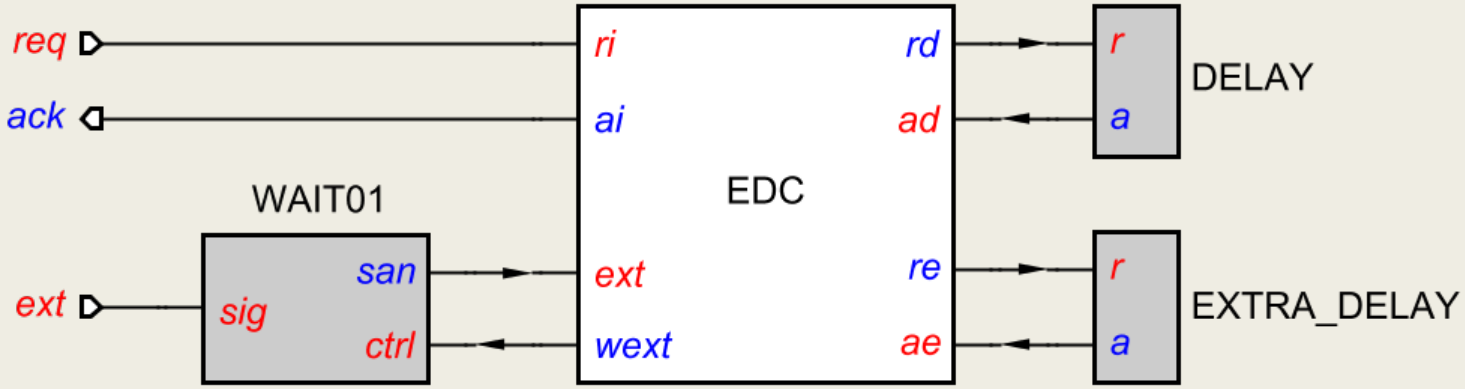
Demo 1: Circuit verification

- Tools are complicated so bugs in synthesised circuits are not impossible; also the designer might have manually altered the circuit
- Verification needs the environment (e.g. the original STG)
- Properties: Conformation, deadlock freeness, output persistence, custom

Demo 2: Arbitration

- Traditional approach: Manually factor out mutex to the environment (laborious + error-prone)
- Workcraft approach: Tag an STG place as mutex
  - Mutex protocol verification (both early- and late-release protocols are supported)
  - No false positives due to output-persistence violation
  - Catches premature withdrawal of requests
  - Automatically factors out mutex during synthesis
  - Circuit verification is supported

Demo 2: Example: Extendible delay

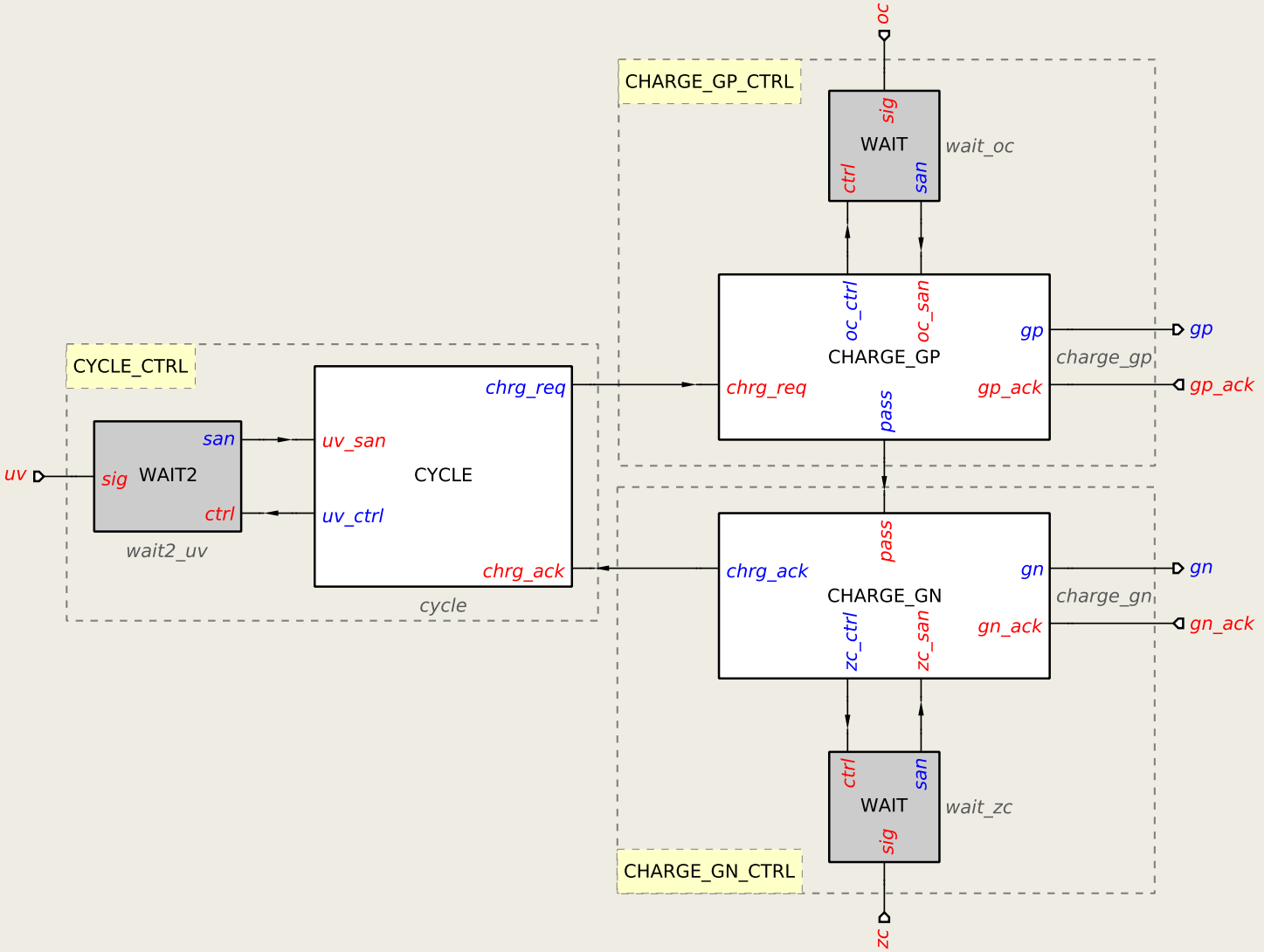


The delay between req and ack is DELAY (if ext did not arrive) and DELAY+EXTRA_DELAY (if ext arrived)

Demo 3: Hierarchical designs

- Large monolithic STGs are impractical (both for humans and synthesis tools), so architectural decomposition is essential
- Can always compose the design into one STG if necessary
- N-way conformation must hold: Whenever one component produces output s, each component receiving s as input must have it enabled
- (Handshakes Wizard handles a special case of N-way conformation.)

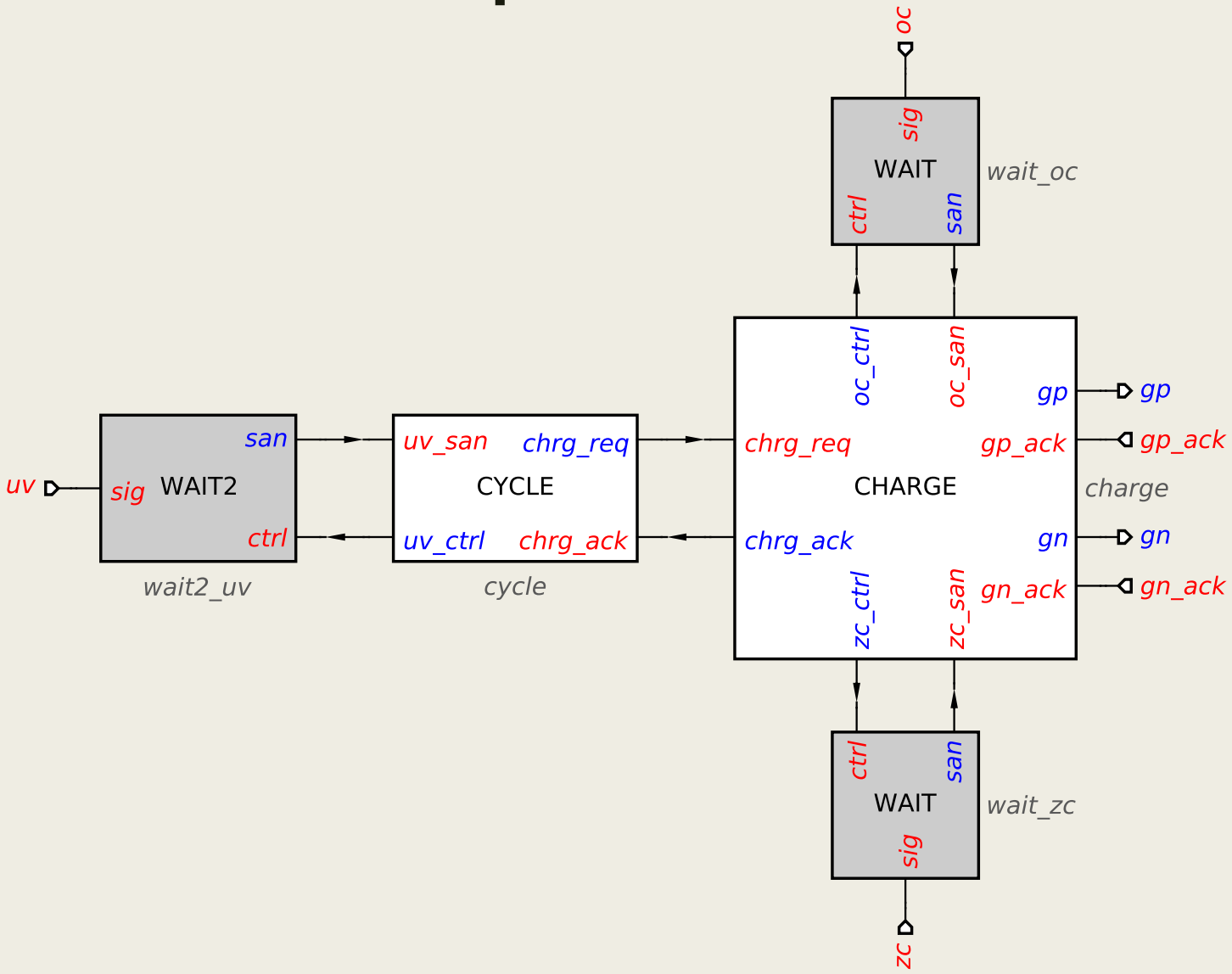
Demo 3: Example: Hierarchical buck



Demo 4: Circuit initialisation

- Initialisation phase does not have to be QDI
- May rely on the initial states of some inputs
- Substitute some of the gates with ones containing an extra input, or use reset pins for latches
- Insert additional gates to explicitly initialise the internal and output signals
- Creative process with multiple optimisation targets (avoiding critical paths, circuit size, gate size, which latches have reset pins, etc.)

Demo 4: Example: Hierarchical buck



Workcraft team

Danil Sokolov

Victor Khomenko

Alex Yakovlev

Download, tutorials, and much other stuff available at

workcraft.org