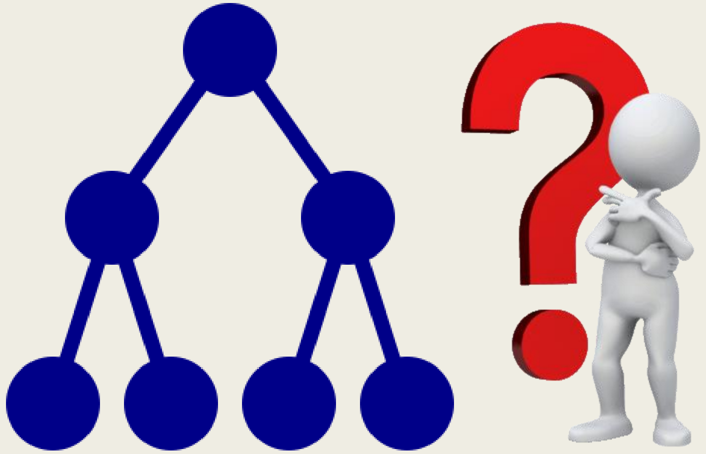
HIERARCHICAL DESIGN IN

Workcraft

Victor Khomenko, Danil Sokolov, Alex Yakovlev

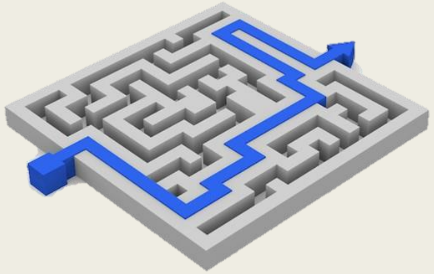
Newcastle University

Why hierarchical?



- The only way to design large systems
  - Requests from the engineers
  - Natural development of Workcraft
- Automation of repetitive tasks by tracking module dependencies
  - Reduces chance of human error
  - Reduces the amount of necessary documentation
- Promotes design reuse (within a project and between projects)
- Improves maintainability

Case study: Navigation



- Realistic systems comprise many modules
  - Multiple versions of the same module
  - Modules are stored as separate files, often without consistent naming
  - Iterative refinements of each module (CSC resolution, concurrency reduction, logic decomposition)
  - Different people working on different modules
- 100-1000s of files to navigate ☹
- Numerous relationships of various kinds between files that one needs to understand and document ☹ ☹ ☹
- Tracking dependencies and navigation can (and should!) be automated

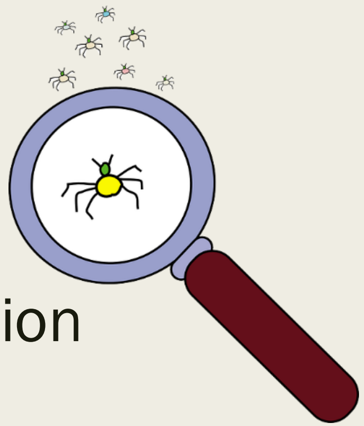
Case study: Design reuse

- Some modules are instantiated multiple times in the same design
  - Stages of multiphase buck
  - Delay controllers
  - Pipeline controllers
- Modules may be reused between designs
  - Opportunistic merge
  - Family of A2A elements (WAIT, WAITX, etc.)
- Hierarchical design helps to identify reusable modules
- Module reuse should be made easy – avoid repeating work
- User-extendible library of reusable modules

Case study: Import/export

- Exporting the whole design hierarchy as modular Verilog netlist
- Import of modular Verilog
- Option for uniquification

Case study: Hierarchical verification

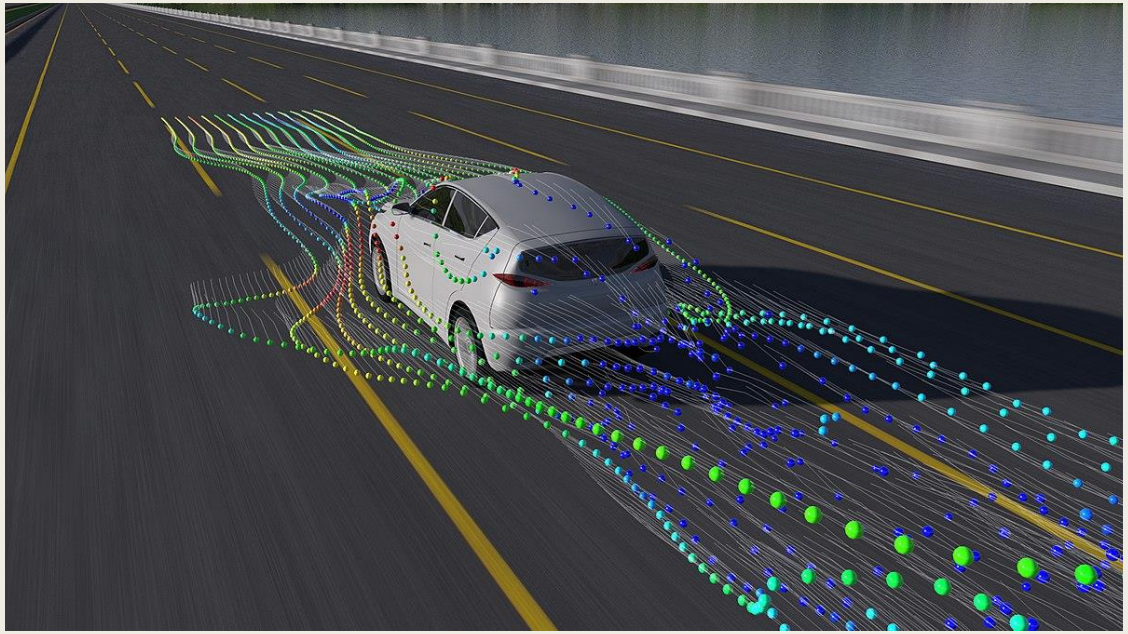


- Tracking dependencies enables automatic formulation of verification obligations
  - Saves manual effort
  - Reduces risk of human error
  - More efficient due to the most abstract models picked up automatically
- Custom properties can be specified at the most appropriate levels of hierarchy

- Abstract models can be inserted into hierarchy
  - Way of capturing the knowledge about the system
  - Improves the efficiency of verification
  - E.g. token ring stage (e.g. phase of a multiphase buck) can be abstracted as a buffer

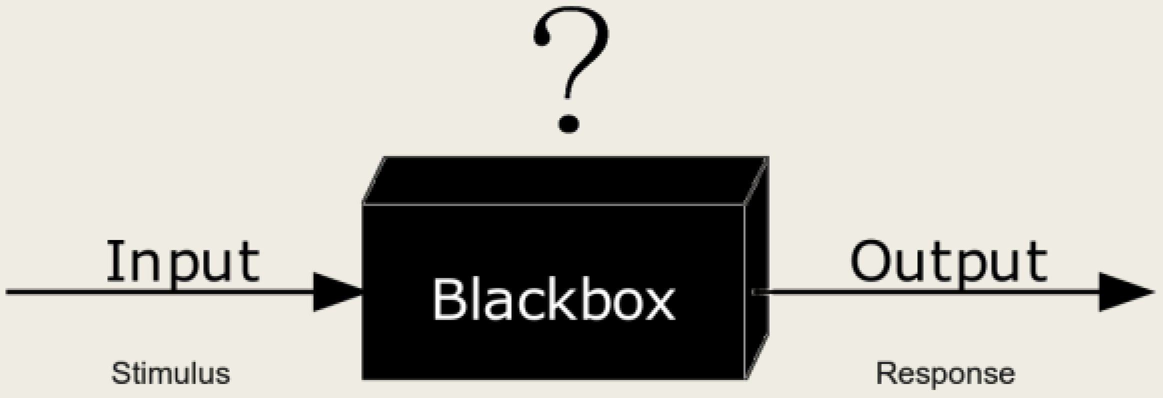
Case study: Hierarchical simulation

- Simulating an execution at different levels of hierarchy
  - Outputs of modules which are not implement can be provided manually
  - Synchronised simulation of several modules
- Local views of a violation trace from formal verification



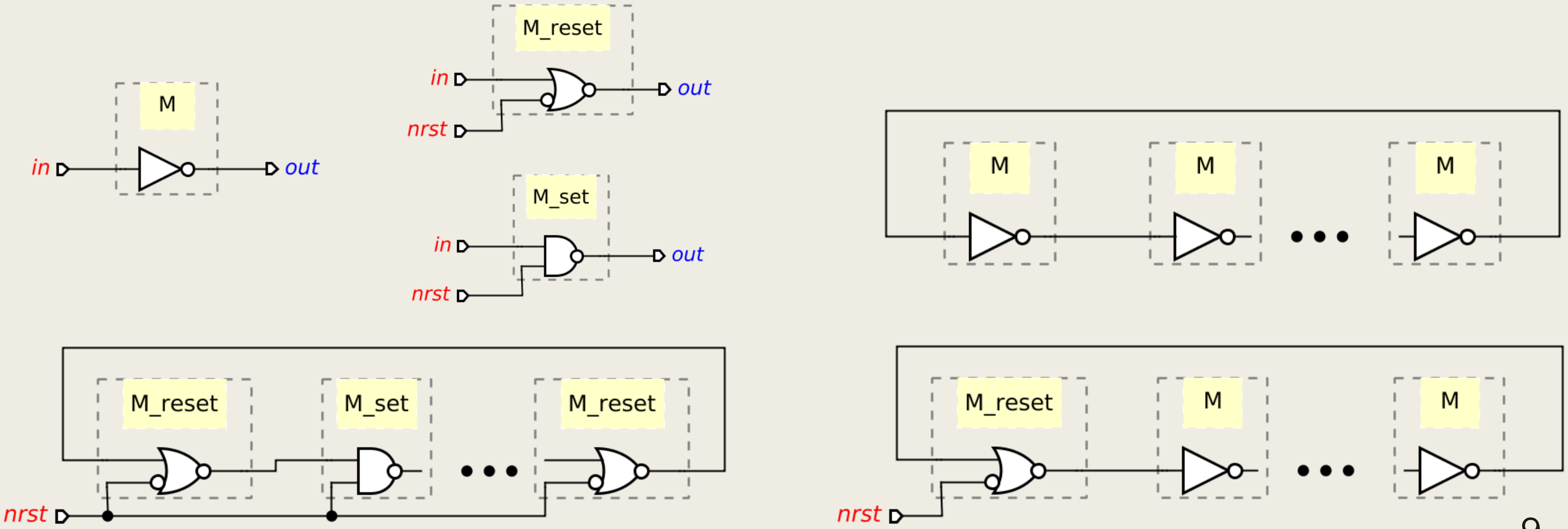
Case study: Hierarchical synthesis

- Push-button synthesis – saves manual effort
- Optimisation due to caching synthesis results and preservation of intermediate STGs for CSC resolution and technology mapping
- Re-synthesis of module compositions
- Decomposition of monolithic modules using DesiJ



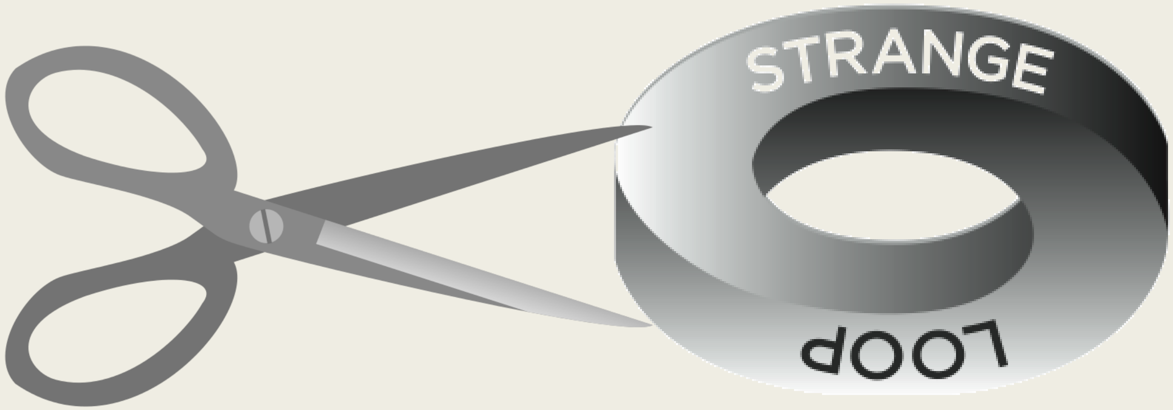
Case study: Hierarchical reset

- Saving manual effort
- Reusing existing reset circuitry for modules
- Optimise reset logic by tracking dependencies (when looking at a module in isolation, one cannot rely on inputs being reset, but context can help)



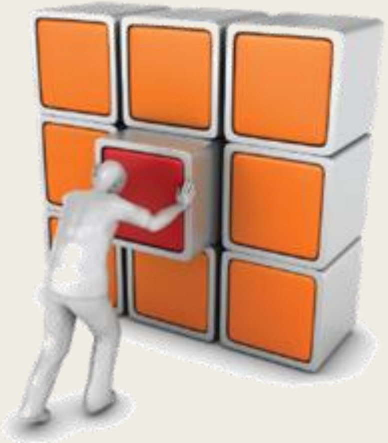
Case study: Hierarchical loop breaking

- Saving manual effort
- Optimising loop breaking due to tracking dependencies
- Taking global loops into account
- No need to rely on PrimeTime loop breaking
- Higher ATPG coverage

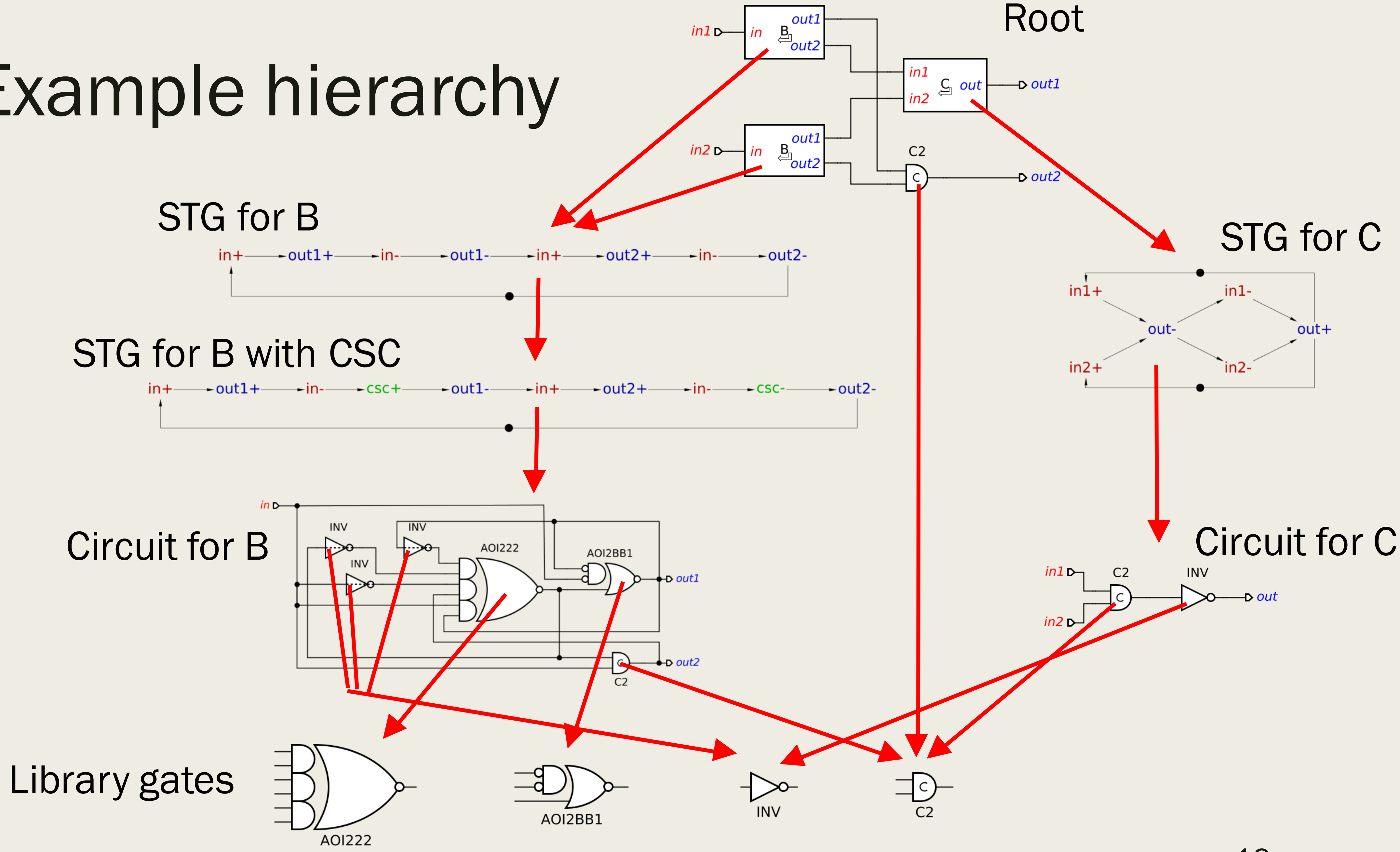


Representation in Workcraft

- Two fundamental relations between modules
  - Module A **instantiates** module B (A must be a circuit)
  - Module C **refines** module D
  - Special low-level modules: complex gates, library gates, MUTEX, WAIT
- No cycles
- Design can be represented as a rooted directed acyclic graph



Example hierarchy



Vision

- Productivity (10x increase)
  - Tracking dependencies
  - Automation of recurring tasks
  - Higher synthesis success rate
  - Systematic design reuse
- Confidence (10x reduction of human error rate)
  - Enhanced verification flow exploiting module dependencies
- Better circuits (up to 50% improvement of latency and area)
  - Optimisation of reset and test circuitry