BDL

- *a visual notation…*

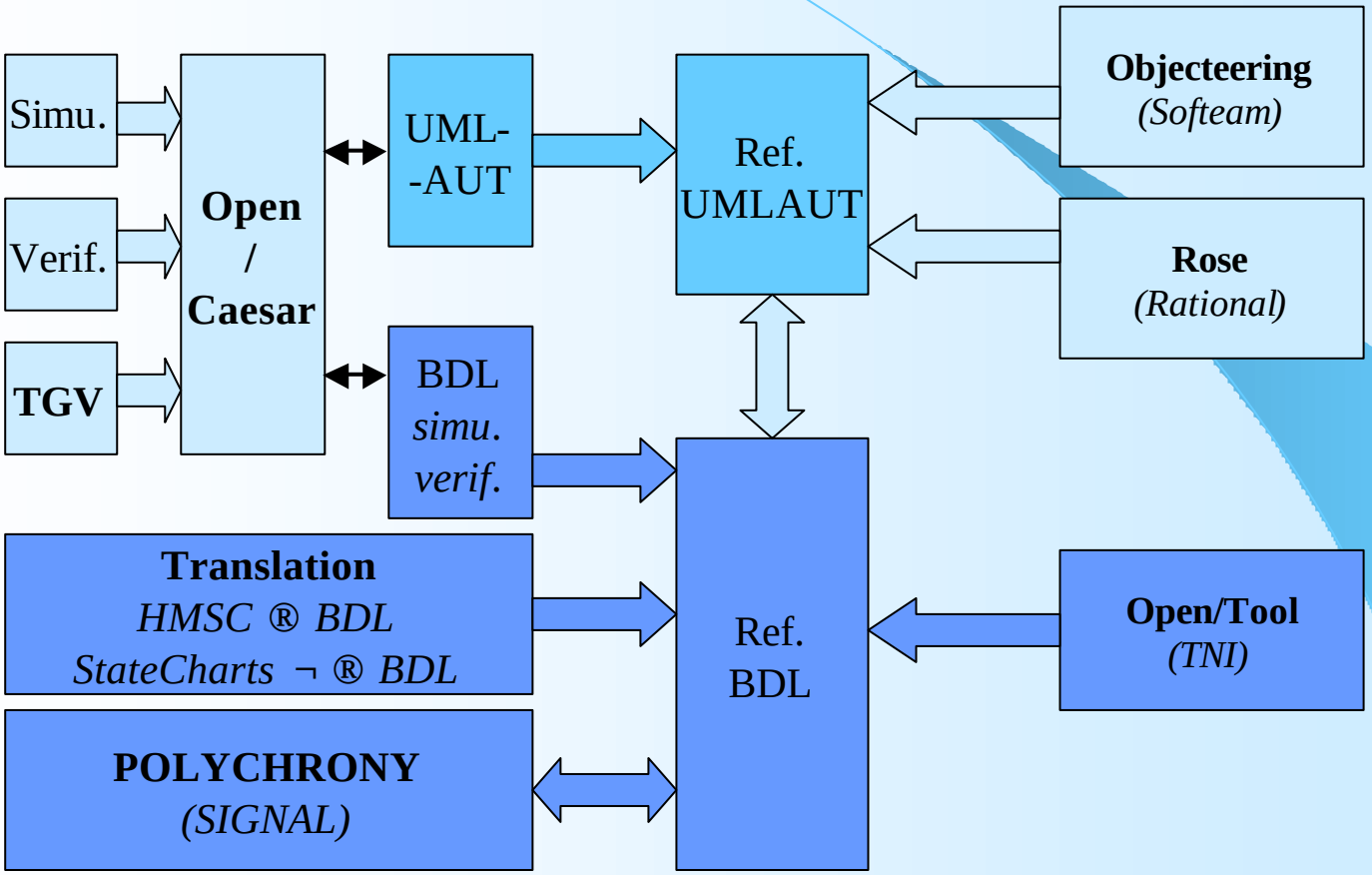
  architectural aspects borrowed from statecharts

  detail notations borrowed from sequence diagrams
- *… having a mathematical specification semantics…*

  compositional

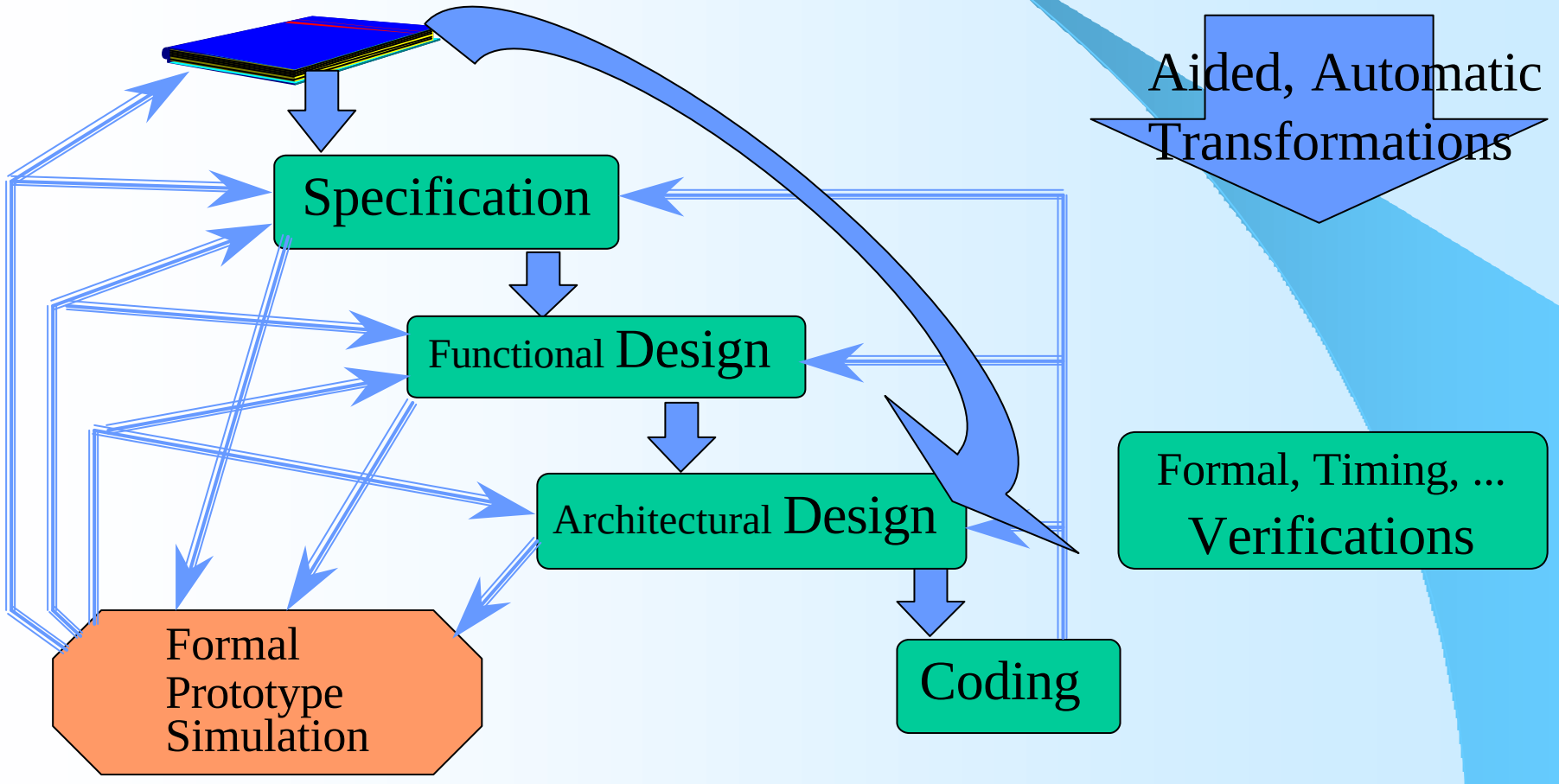
  allows for changes in the architecture
- *… and supporting correct-per-construction deployment*

The BDL Workbench



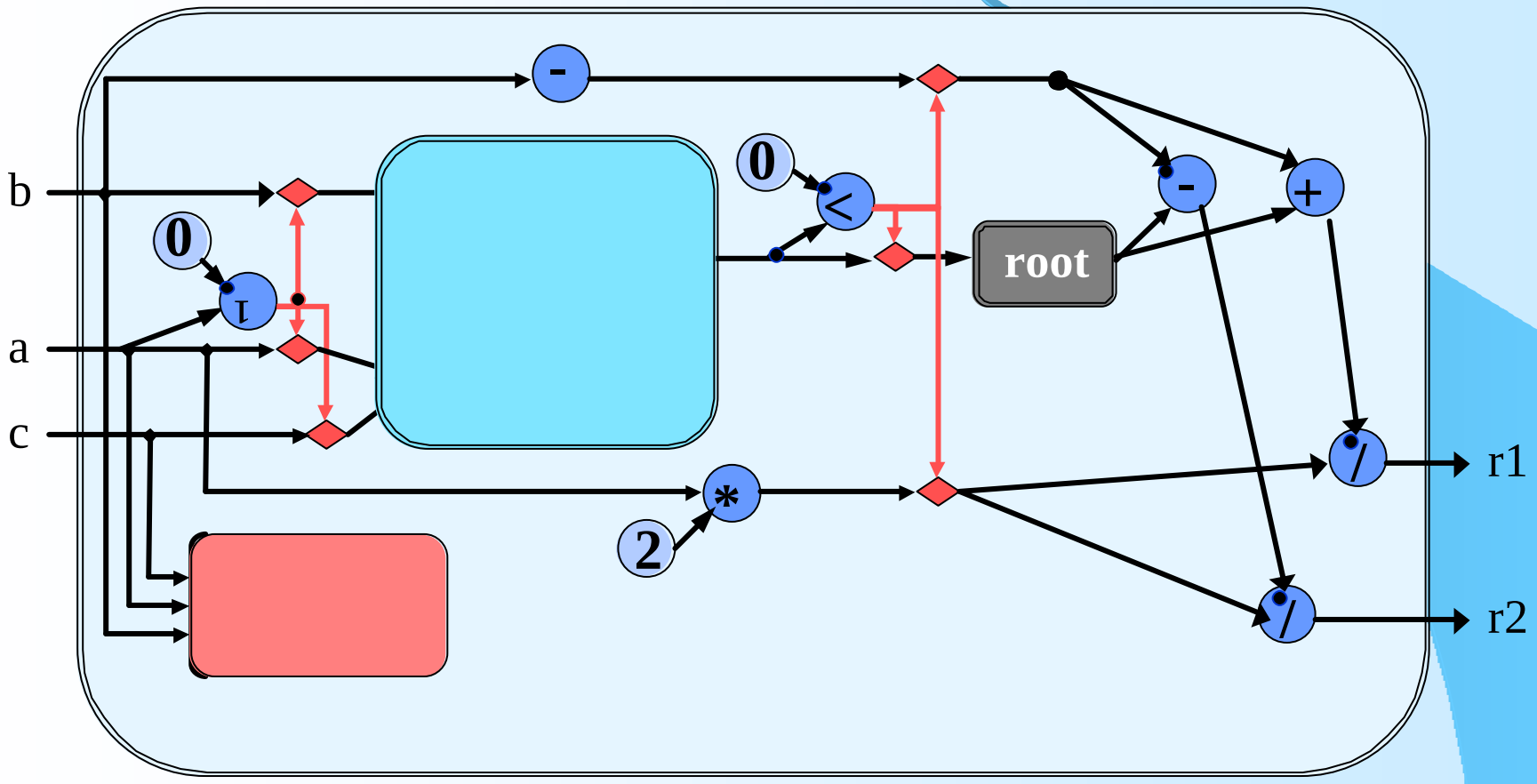
Polychrony

- Real time and embedded system design



The core model

Synchronized data flow



Polychrony

Some functionalities

State transformations

- Event/Boolean conversion
- Modifying interface for event signals
- Flattening expansion: hierarchy of clocks is reduced to one level
- Boolean State variables: defined at the master clock
- Cycle detection

Graph normalization

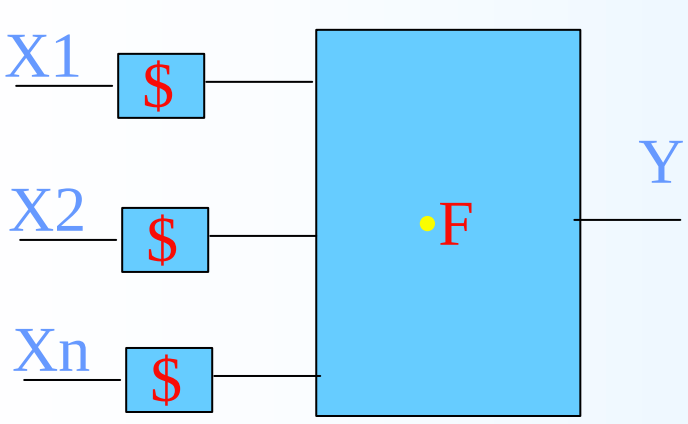
- Function:
  - Unifying signal defined by the same expression
  - Solving constraints
- Based on:
  - Rewritting of Boolean signal expressions

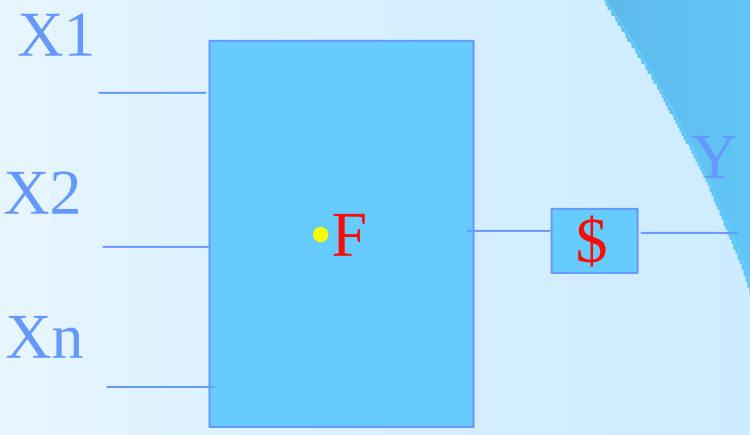
RETIMING

- Minimization of the state variables
- Solving constrainsts

  UPSTREAM retiming

  DOWNSTREAM retiming





PARTITIONING

- User partitioning
- Input directed partitioning
- Output directed partitioning
- Control/Calculation partitioning
- State variable/other signals partitioning

USER PARTITIONING

Reorganization

- Using PRAGMA: RunOn
- Method returns
  - a set of subgraphs using the value of the pragma
  - Clocks are assigned to subgraphs
  - The interface of each subgraph is built

I/O directed PARTITIONING

Reorganization

- Definition:
  - X,Y are in the same set iff they depend on the same inputs
- A Program P is rewritten in
  - (| (| P1 | P2 | ….|Pn |) | SCHEDULER() |)
- How?:
  - a set of subgraphs is built, clocks are assigned to subgraphs, interface of each subgraph is built
  - The graph of the node calls is built: SCHEDULER

CONTROL/CALCULATION partitioning

Reorganization

- A Program P is rewritten in
  - (| Pcontrol() | Pcalculations() |)
    - Pcontrol: contains clocks and boolean definitions
    - Pcalculations: contains numerical definitions
- How?:
  - Two subgraphs are built according on types of signals
  - Clocks are assigned to subgraphs
  - The interface of each subgraph is built
  - The graph of the node calls is built

STATES/OTHERS Partitioning

- A Program P is rewritten in
  - (| P_states() | P_others() |)
    - P_states: contains memorizations definitions
    - P_others: contains others definitions
- How?:
  - Two subgraphs are built according on the criterion
  - Clocks are assigned to subgraphs
  - The interface of each subgraph is built
  - The graph of the node calls is built

INTERFACE SYNTHESIS

Abstraction

- IO dependences
  - Transitive Closure reduced to I/O

    X --> Y at Hi
- Clocks Projection
  - I/O signal clocks
  - I/O Dependence clocks

OPTIMIZATIONS

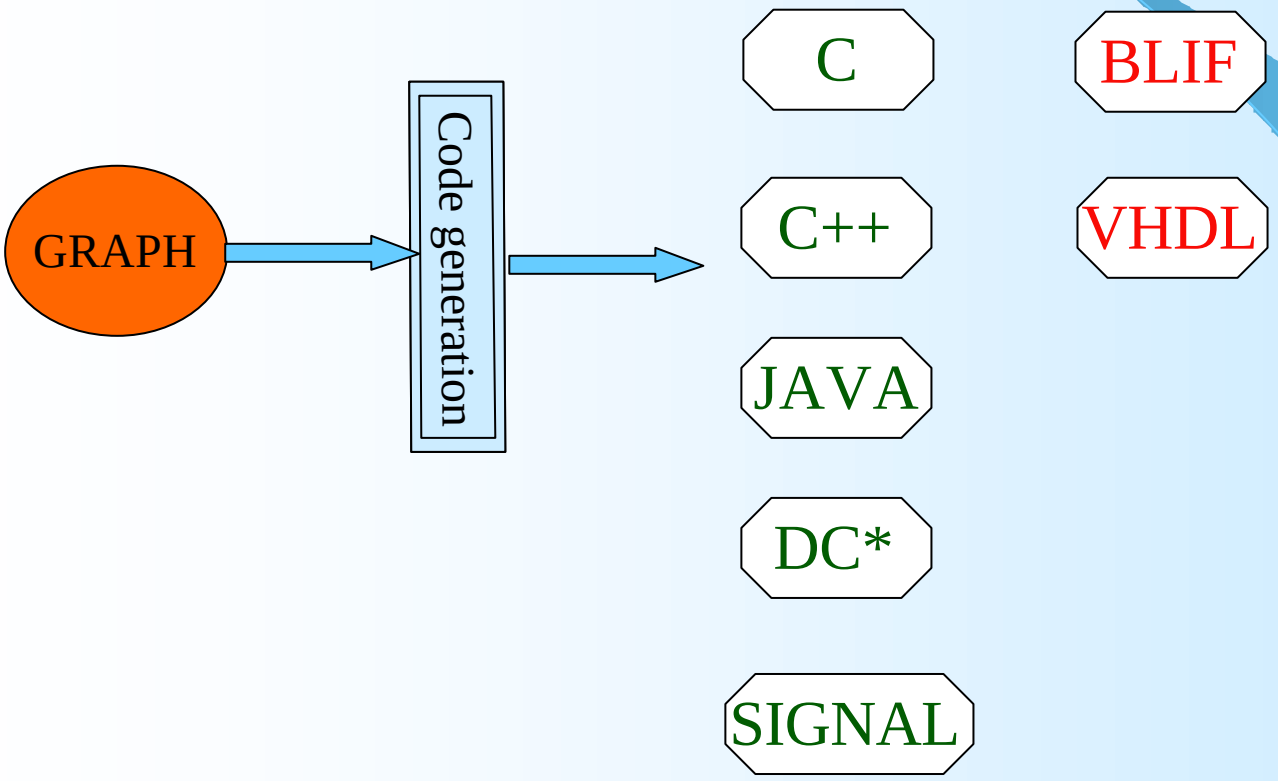
- Mutliplexing delayed variables
  - (| ZX := X $ 1 | ZZX := ZX $ 1 |)

    can be implemented using an array of 3 elements without copy
- Multiplexing by signal aggregation
  - x := y when c
  - x and y can be implemented in one variable

OPTIMIZATIONS

- Using clock calculus
  - Exclusive clocks
  - Utility clock of signals
  - Assignment clock of signals
  - Modification clock of signals

CODE GENERATION

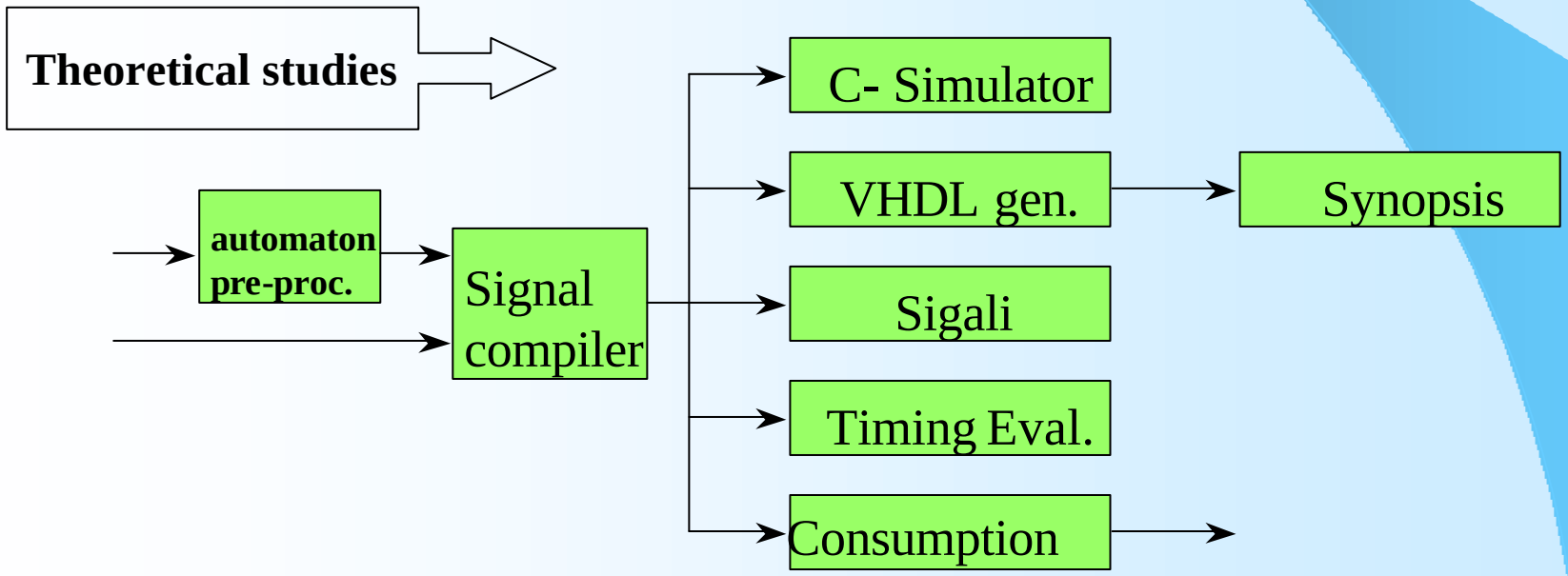


Formal Verification

- Sigali and automata
  - Translation automata CADP
  - Studying correspondence Sigali μ-calculus

- Temporal Interpretation
  - P a program
  - A a model of target architecture
    - **Set of generators**
  - we define $T_A(P)$ model of time consumption of P on A
  - Simulator P| $T_A(P)$
- Verification of duration constraints
  - max-plus Algebra

- Hardware synthesis
  - Interpretation of absence as *don't care*
  - Clock synthesis
  - Exploitation of exclusions (clock calculus)



Perspectives

- à la GNU-licensed version of Polychrony
- Software architecture
- Static properties
  - abstract interpretation (intervals in BDD)
- Validation
  - validation of each transformation
  - verification of the trace of their application
- Test generation
  - using controller synthesis techniques