Design Methodology for High−Level Synthesis

Chapter 9

Source: Gajski, Dutt, Wu, Lin

"High−Level Synthesis"

Kluwer Academic Publishers, 1992

DESIGN METHODOLOGY REQUIREMENTS

1. The syntax and semantics of the input and output descriptions.

2. The set of algorithms for translating input into output descriptions.

3. The set of components to be used in the design implementation.

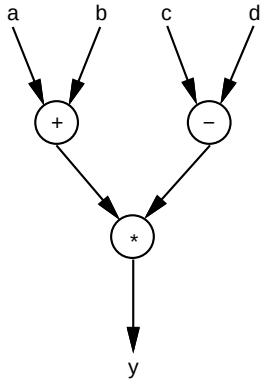
4. The definition and ranges of design constraints.

5. The mechanism for selection of design styles, architectures, topologies and components.

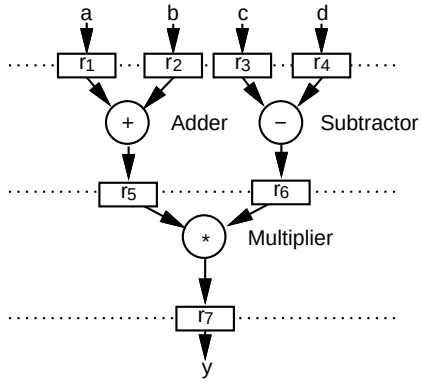
6. Control strategies (usually called scenarios or scripts) that define synthesis tasks and the order in which they are executed.

TRIVIAL SYNTHESIS SYSTEM

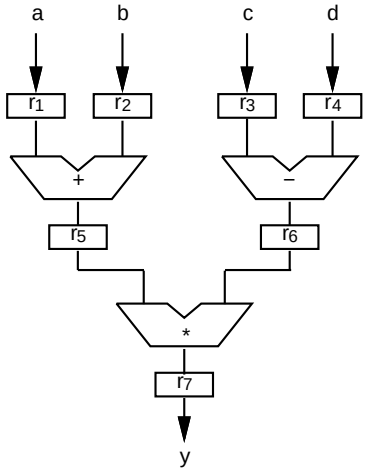
Assumptions: Sample/clock cycle
Computation/2 clock cycles
Operation/clock cycle
Same bit width



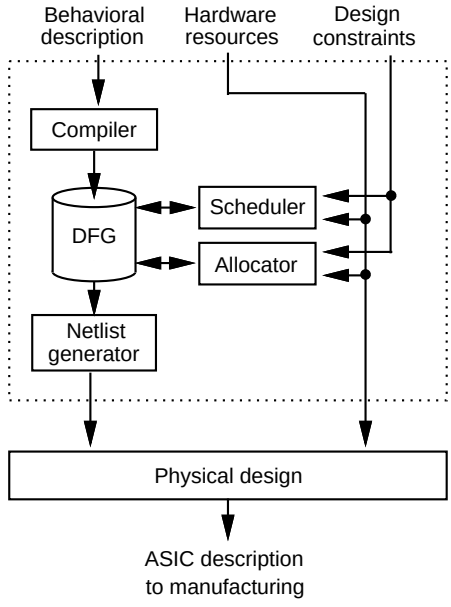
DFG



Annotated DFG



Datapath

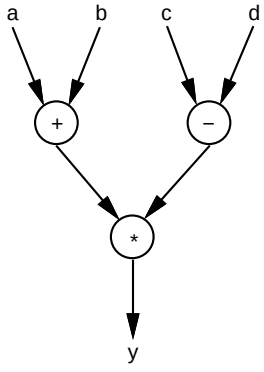


Synthesis system

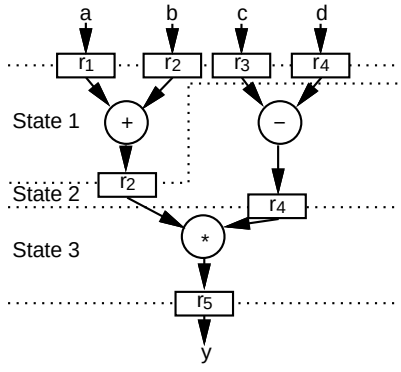
PRACTICALITY OF ASSUMPTIONS

1. All units are not of the same bit width or same propagation delay.
2. Dataflow architechure is too expensive.
3. I/O rates do not match architecture.
4. Synchronous I/O is not always available.

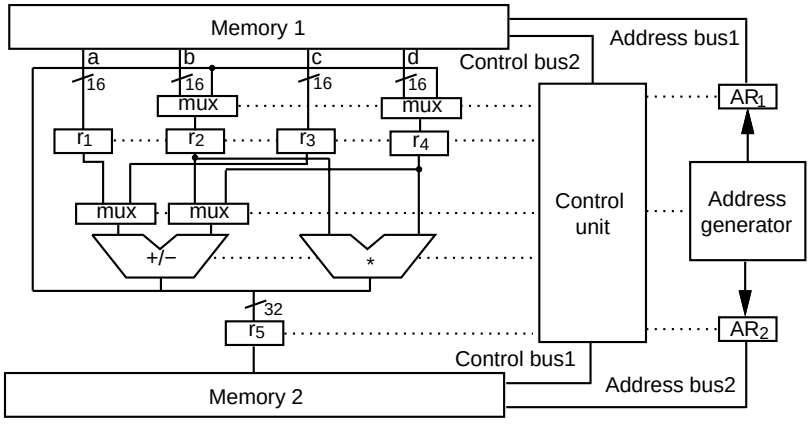
EXAMPLE WITH MEMORIES



DFG

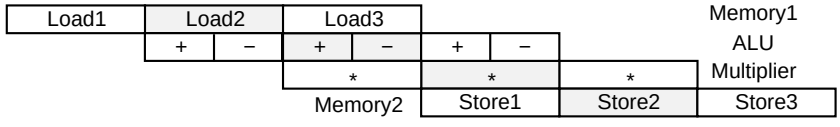


Annotated DFG



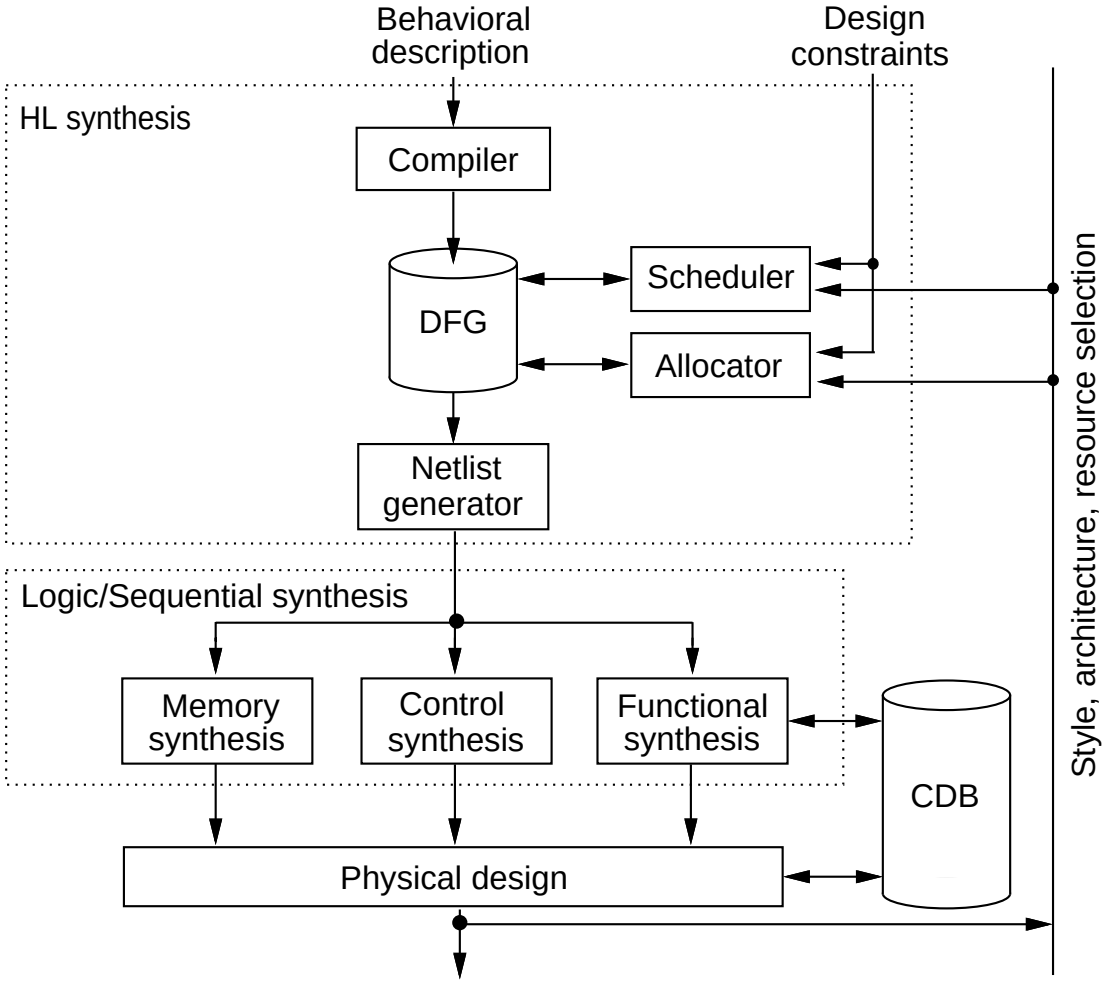
FSMD implementation

Resource utilization



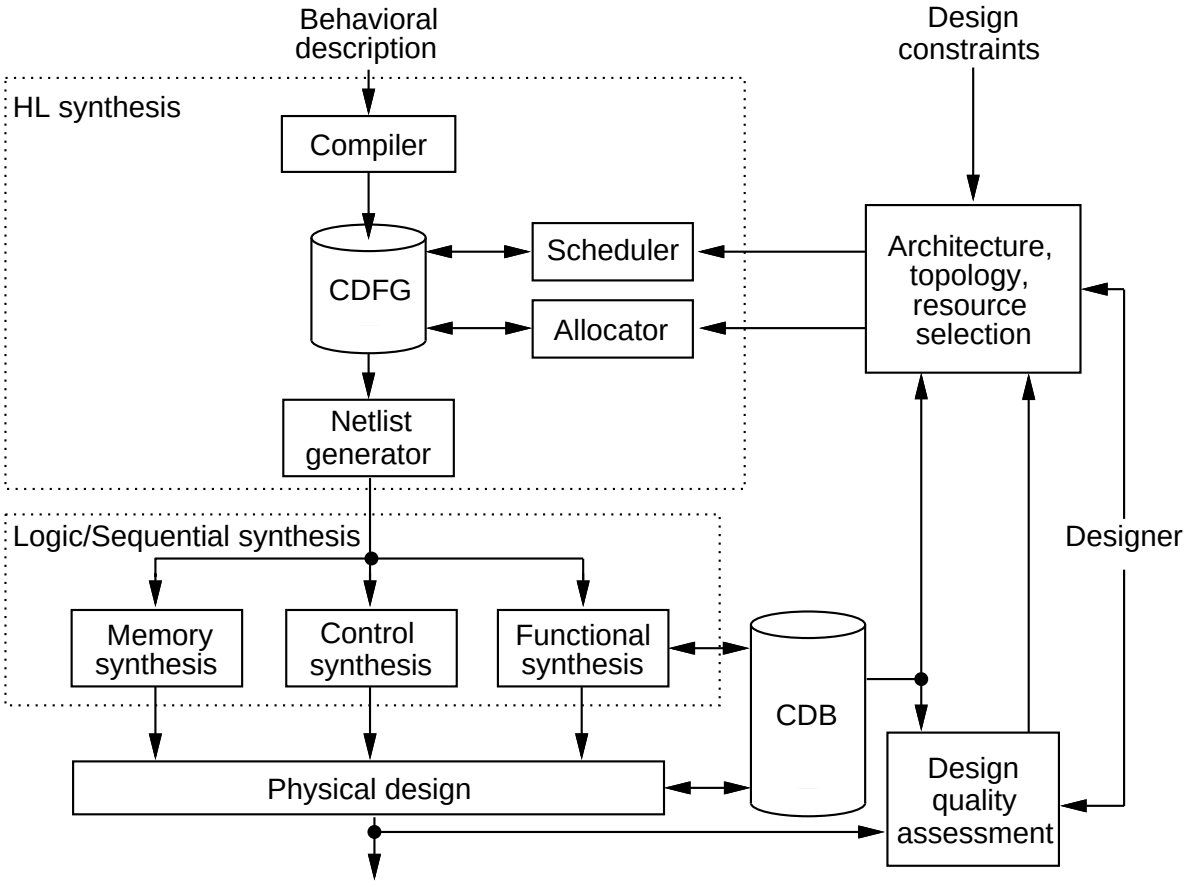
Improved resource utilization

EXAMPLE SYNTHESIS SYSTEM



Synthesis system with a component database and user controlled resource selection

EXAMPLE SYNTHESIS SYSTEM



Synthesis system with automatic iterative improvement

GENERIC SYNTHESIS SYSTEM

Completeness

1. All levels of design
2. Different target architectures

Extensibility

1. Addition of new algorithms and tools
2. Addition of new architecture styles
3. Addition of new libraries

Controllability

1. Control of tools
2. Control of design exploration
3. Quality metrics of design assessment

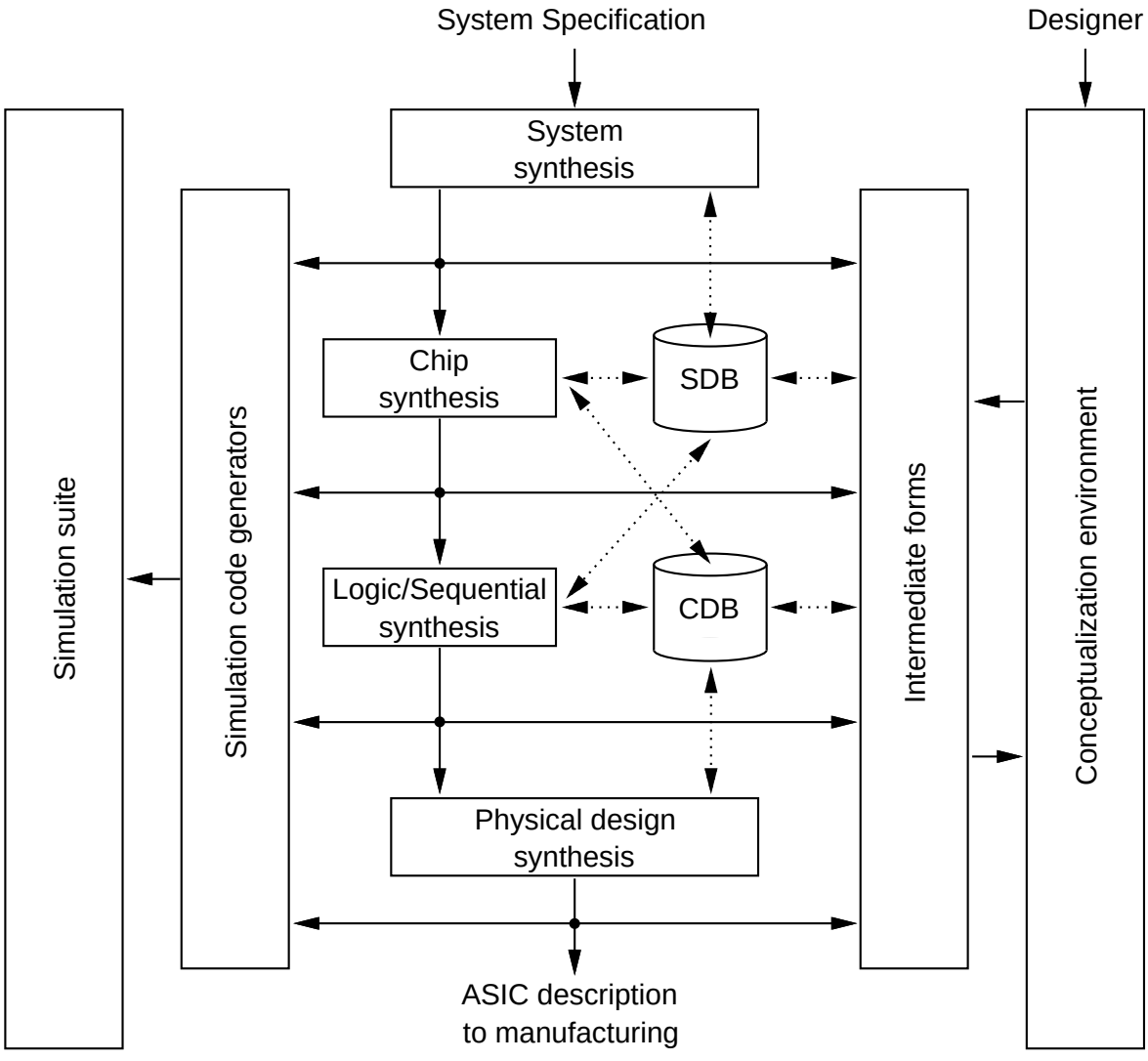
Interactivity

1. Partial design definition
2. Modification during and after synthesis

Upgradability

1. Capture−and−simulate to describe−and−synthesize
2. Mixing of strategies

HYPOTHETICAL SYNTHESIS SYSTEM

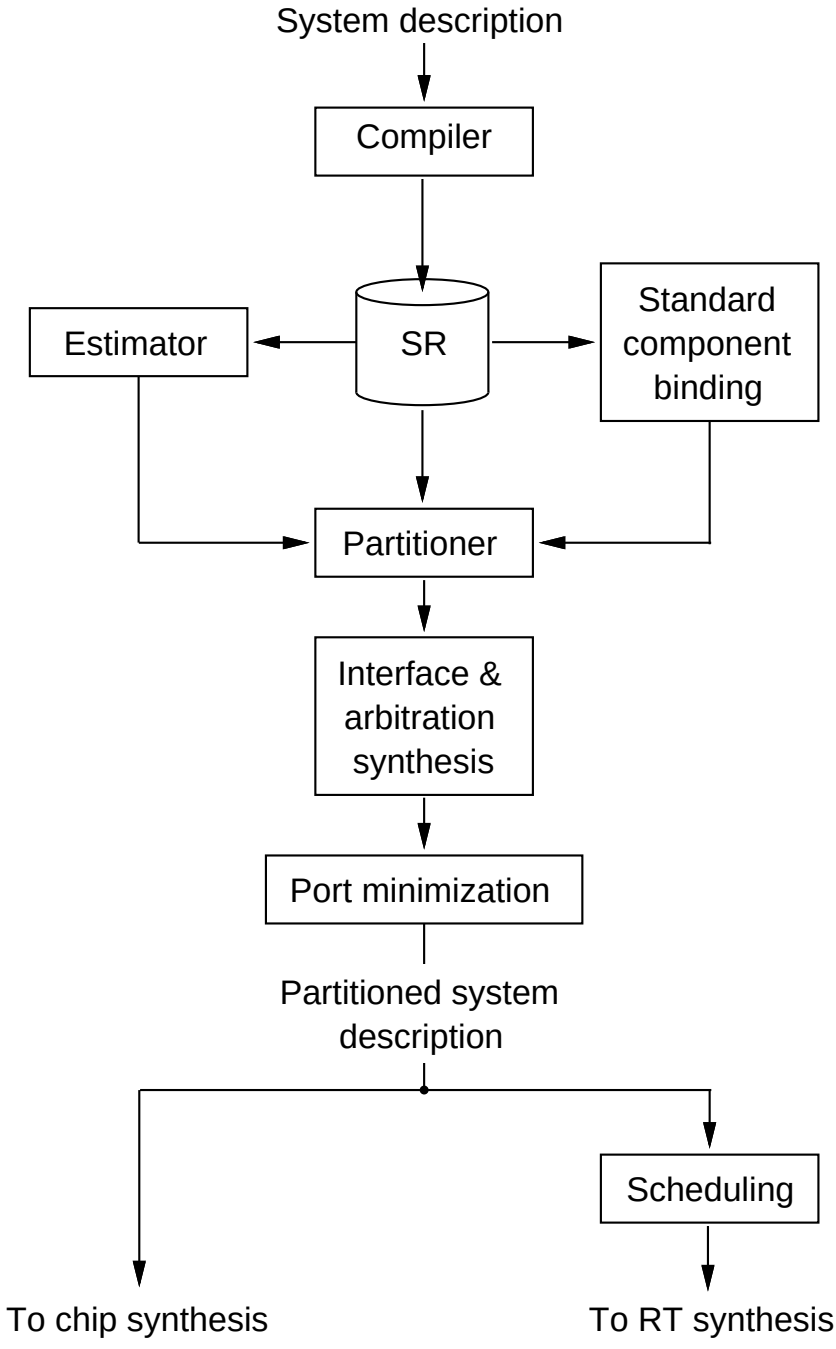


1. Supports capture−and−simulate and describe−and−synthesize methodologies.

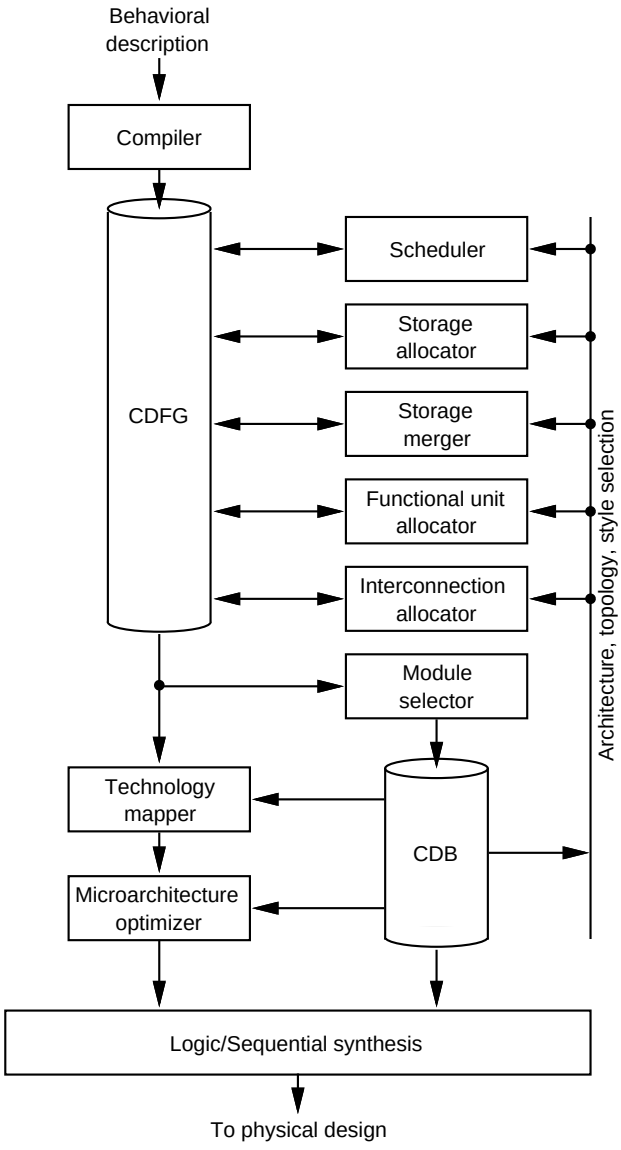
2. Separation of synthesis and simulation.

3. Hierarchical interactive synthesis.

SYSTEM SYNTHESIS METHODOLOGY



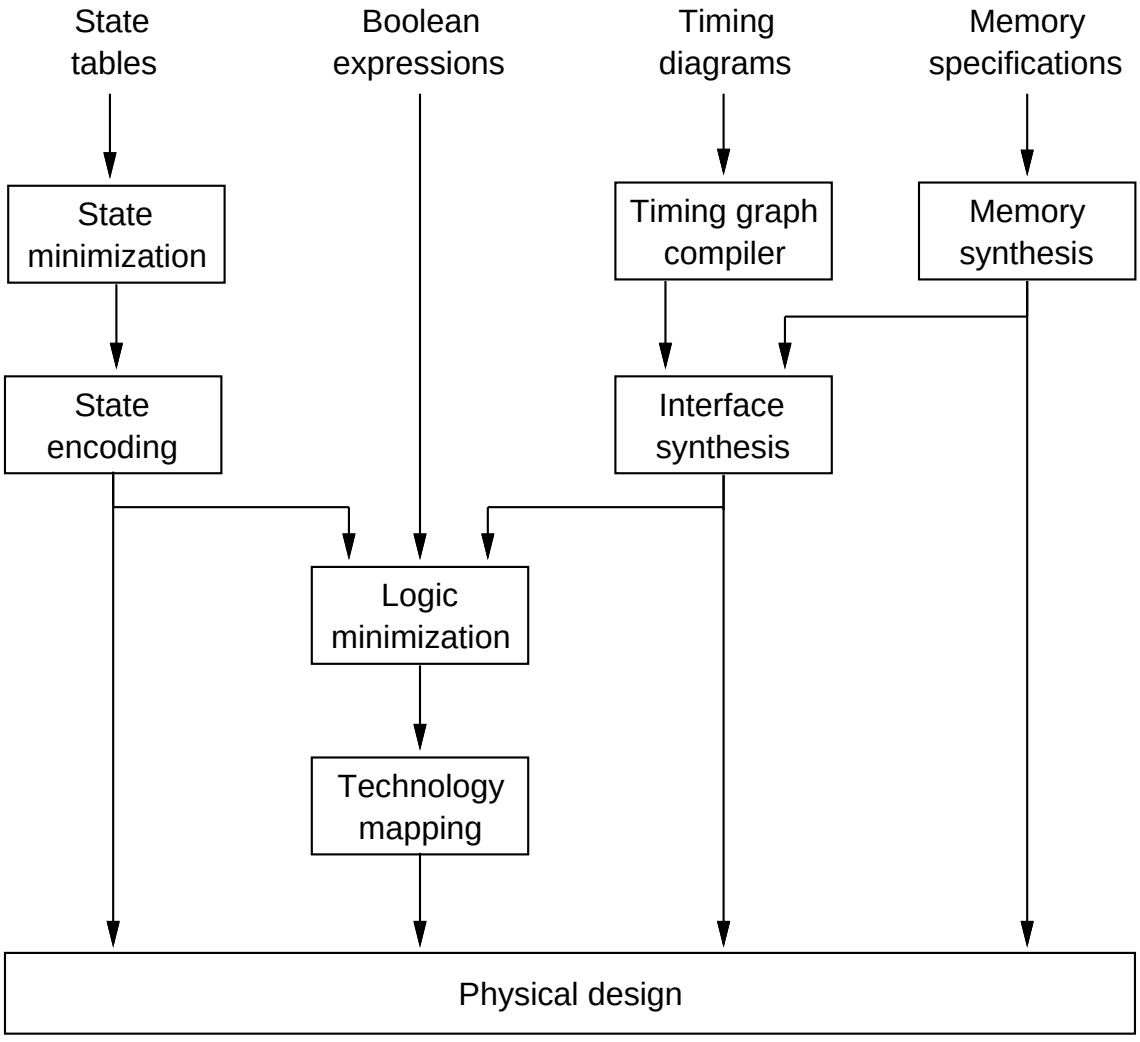
CHIP SYNTHESIS METHODOLOGY



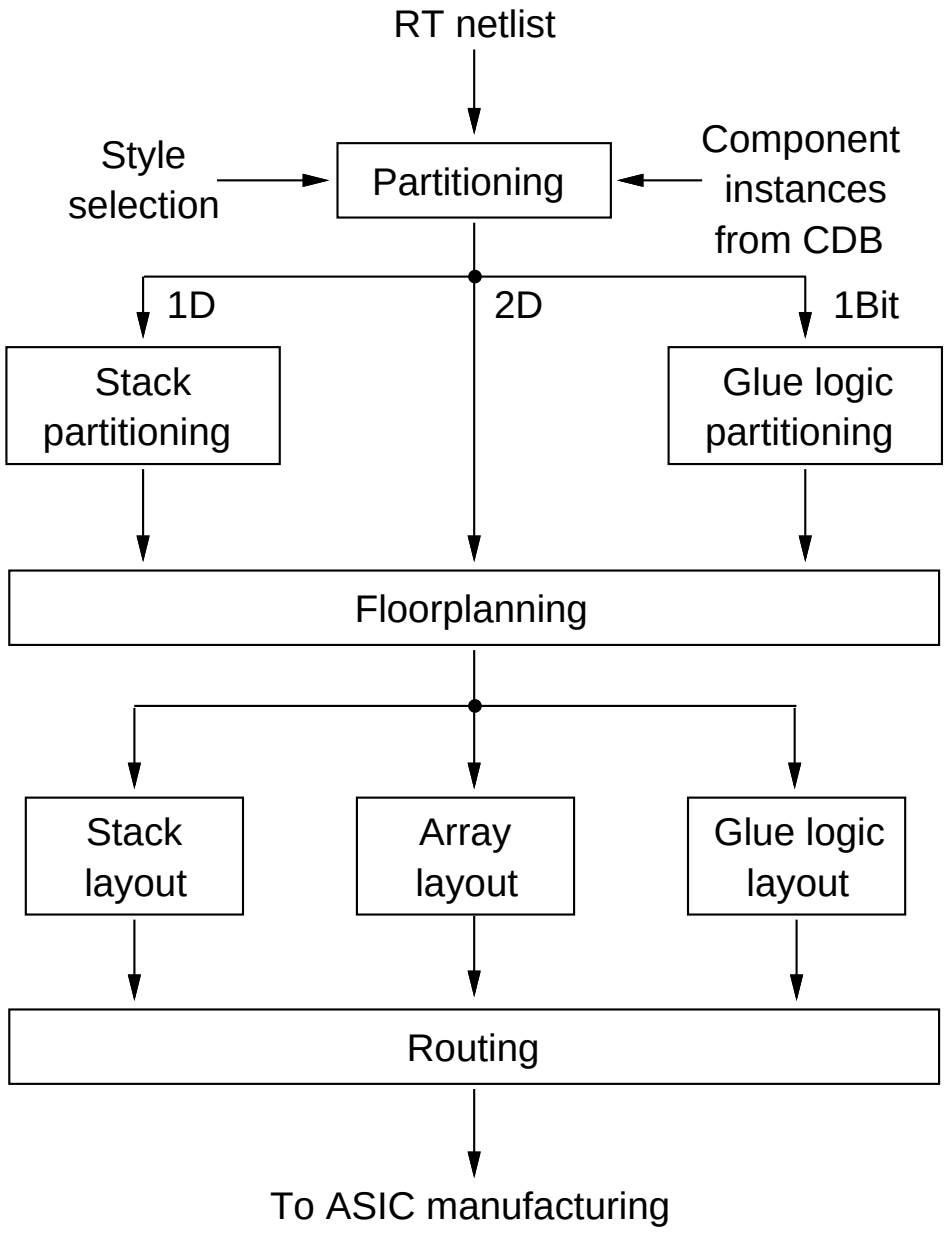
Technology mapping strategies:

1. **Top−down**
2. **Meet−in−the−middle**
3. **Bottom−up**

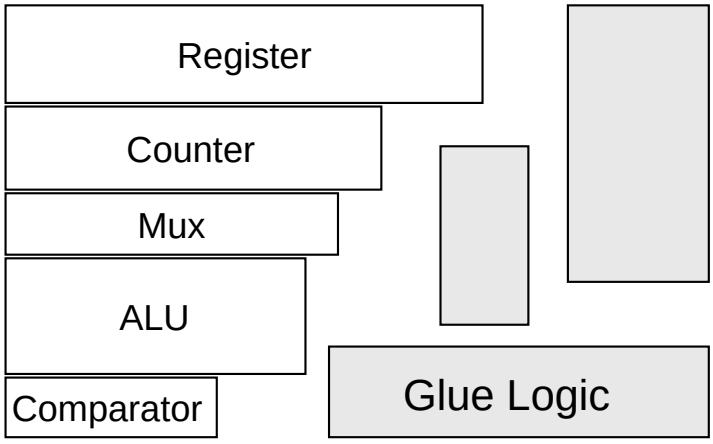
LOGIC−SYNTHESIS SYSTEM



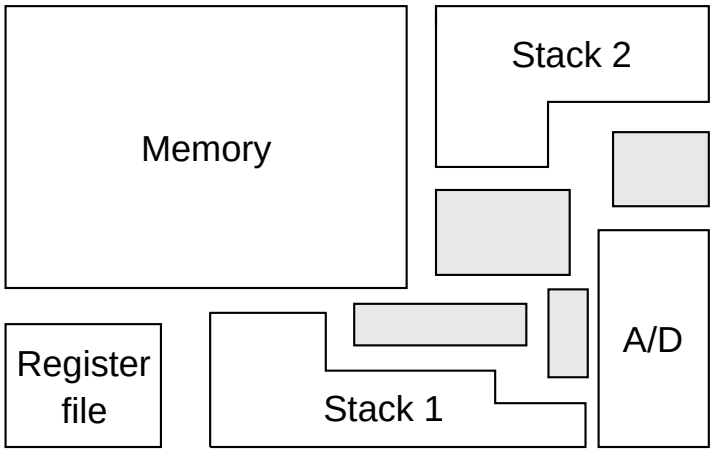
PHYSICAL DESIGN METHODOLOGY



PHYSICAL DESIGN METHODOLOGY



Datapath floorplan



Chip floorplan

SYSTEM DATABASES

Phase 1: Collection of tools

Phase 2: Tool integration

Phase 3: Common data model

Phase 4: Design views and consistency checks

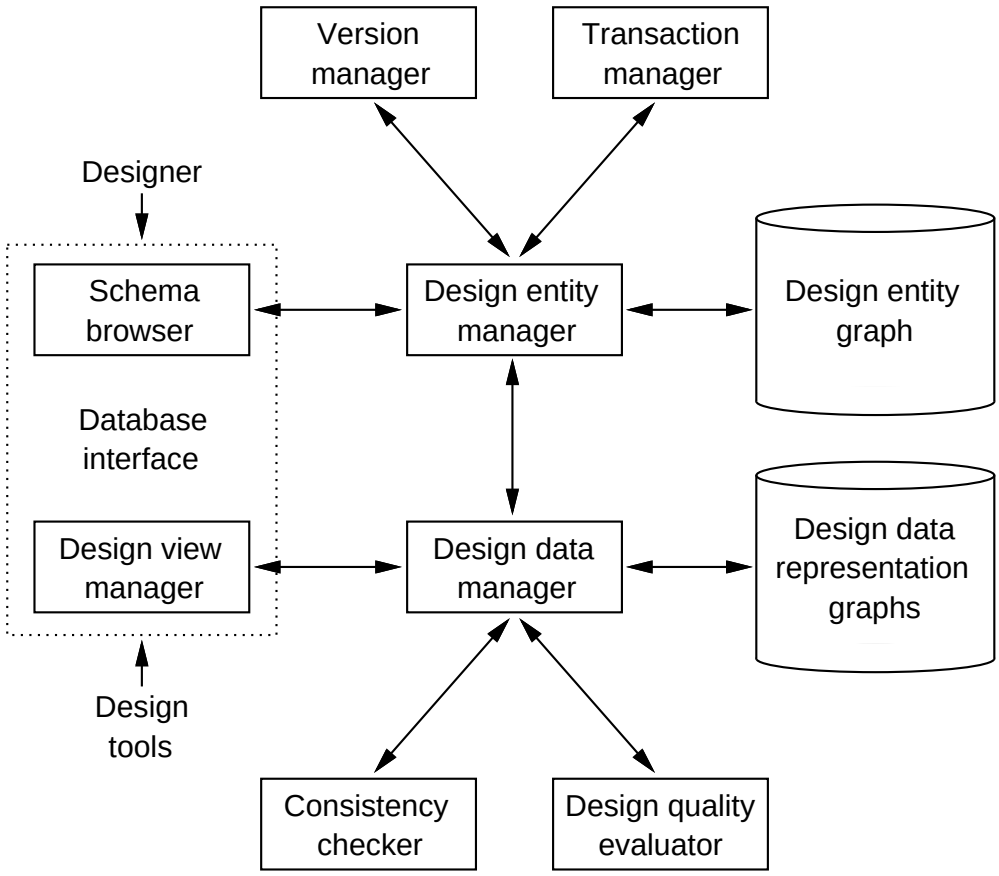
DATABASE ARCHITECTURE

Design entity graphs

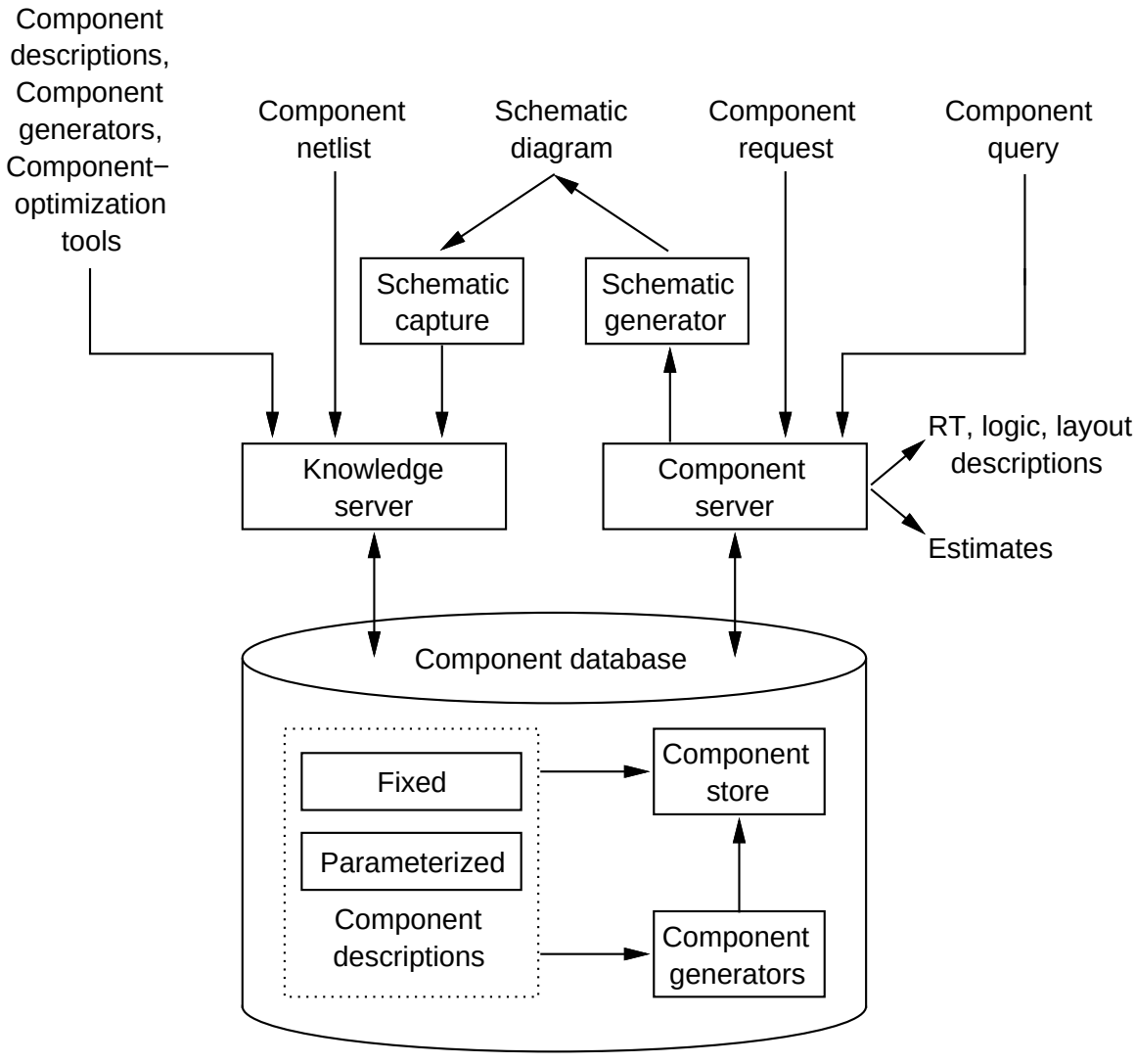
hierarchy, version control, configuration management

Design data graphs

behavior, structure, geometry, timing



COMPONENT DATABASE



CONCEPTUALIZATION ENVIRONMENT

Data and design manager

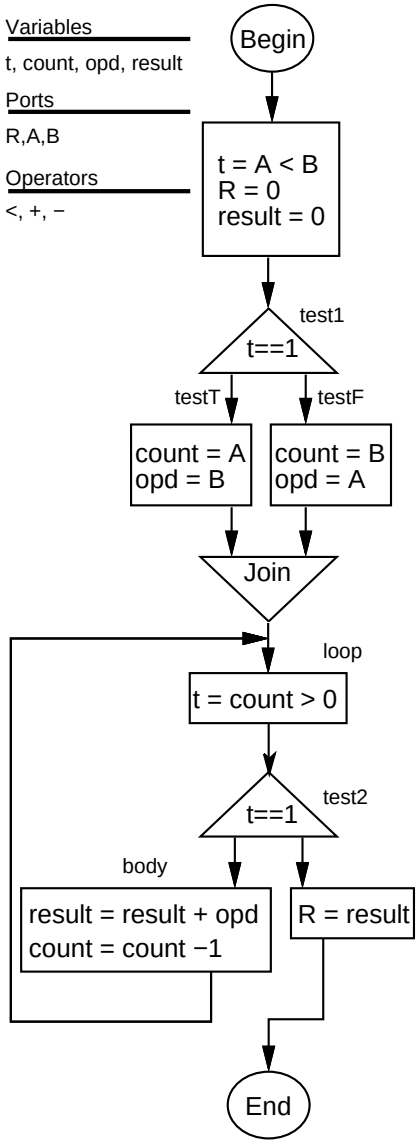
Displays and editors

Design−quality estimators

Design−consistency checkers

Synthesis algorithms

BEHAVIORAL DESCRIPTION DISPLAY

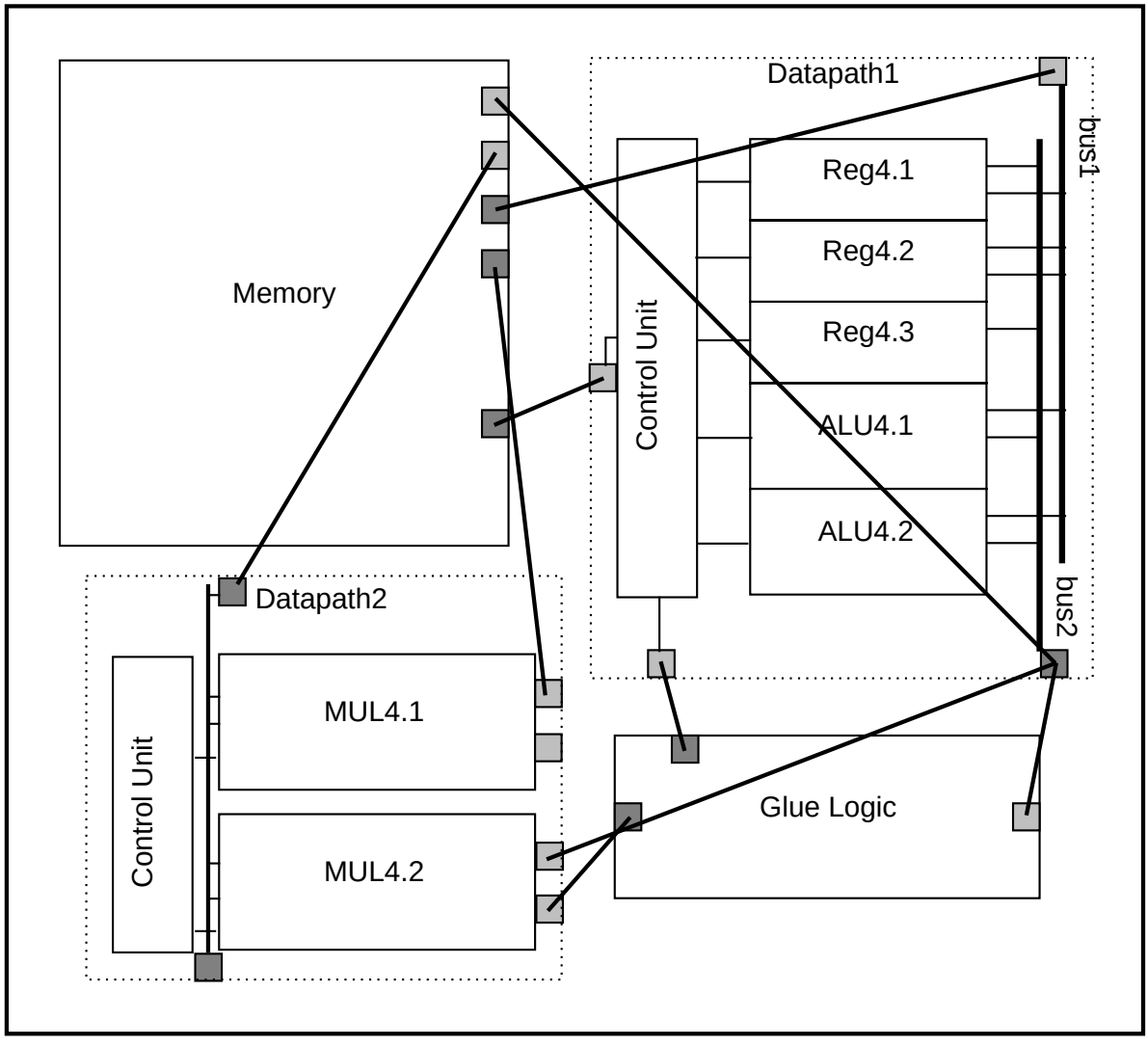


Flowchart

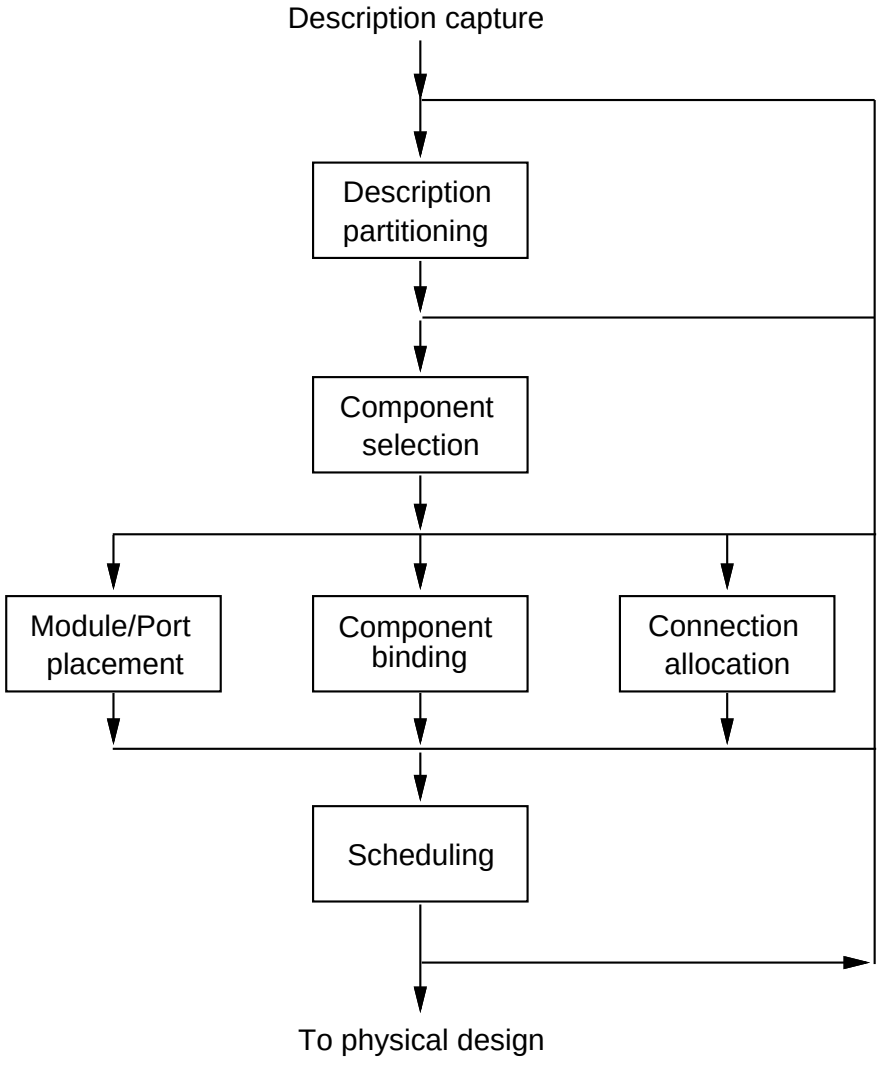
State	Condition	CondValue	Actions	NextState
BEGIN			t=(A<B), R=0, result = 0	test1
test1	(t)	("1")		testT
		("0")		testF
testT			count=A, opd=B	join
testF			count=B, opd=A	join
join				loop
loop			t=(count>0)	test2
test2	(t)	("1")		body
		("0")		1
body			result=result+opd, count=count−1	loop
1			R = result	END
END				END

State table

FLOORPLAN DISPLAY



INTERACTIVE SYNTHESIS



Possible scenarios for interactive synthesis

FUTURE DIRECTIONS

Complete synthesis systems/frameworks

Descriptions and modeling guidelines

Quality metrics and estimation

Component taxonomy and generators

Databases and environments

Design exploration strategies

Hardware/software codesign