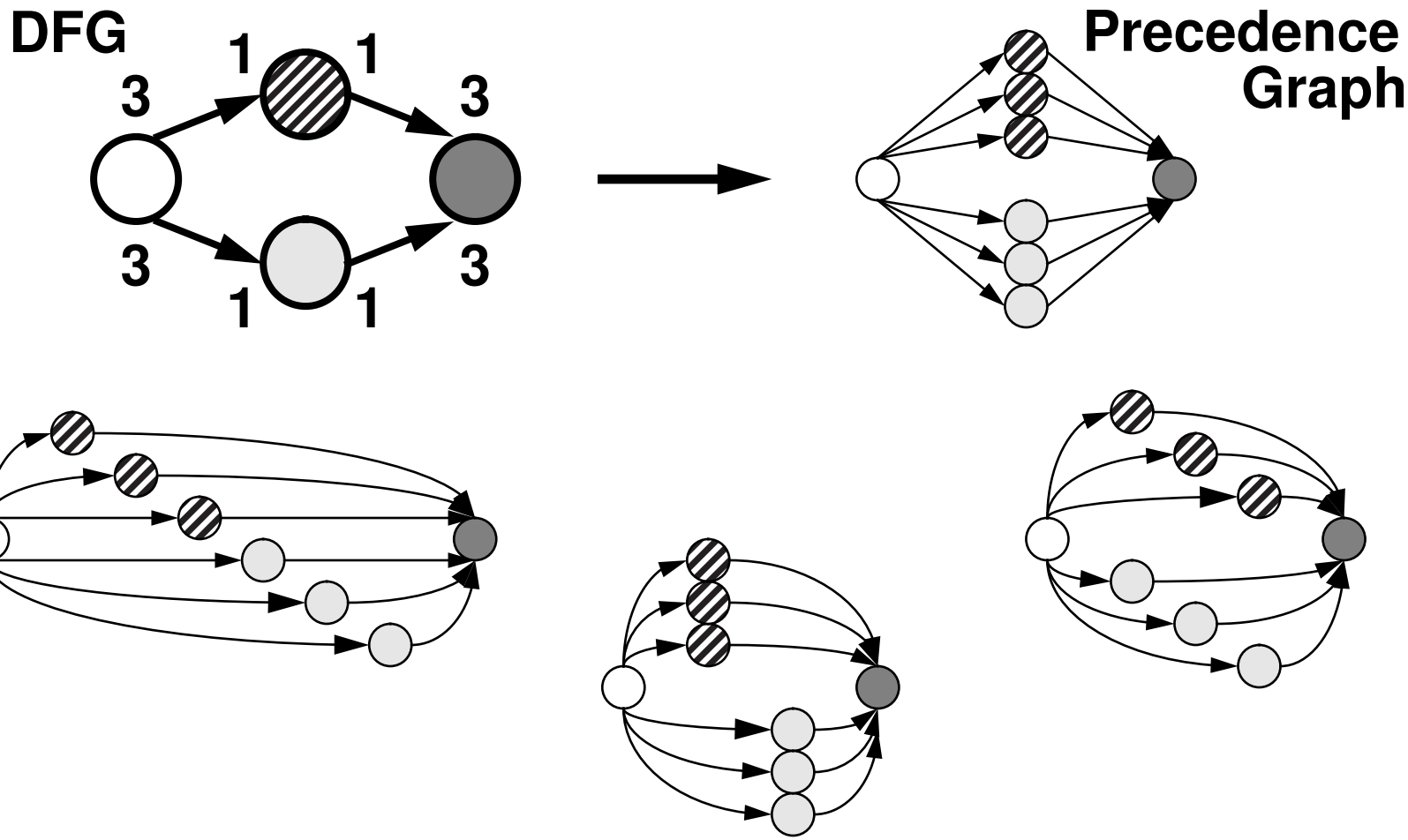
Demonstrations

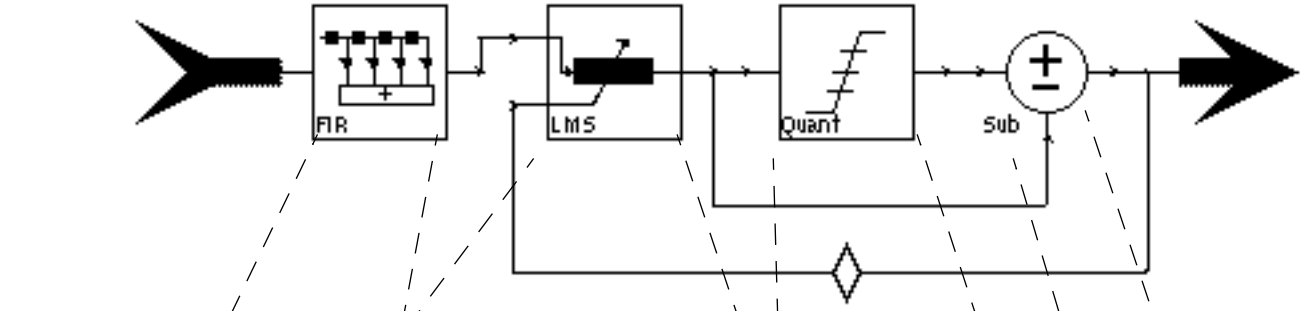
- **Retargetting from SDF to VHDL**
- **Multiple Targets within VHDL Domain**
- **Input to synthesis tools**
- **Coordinated use of synthesis and optimization with Ptolemy for design exploration**

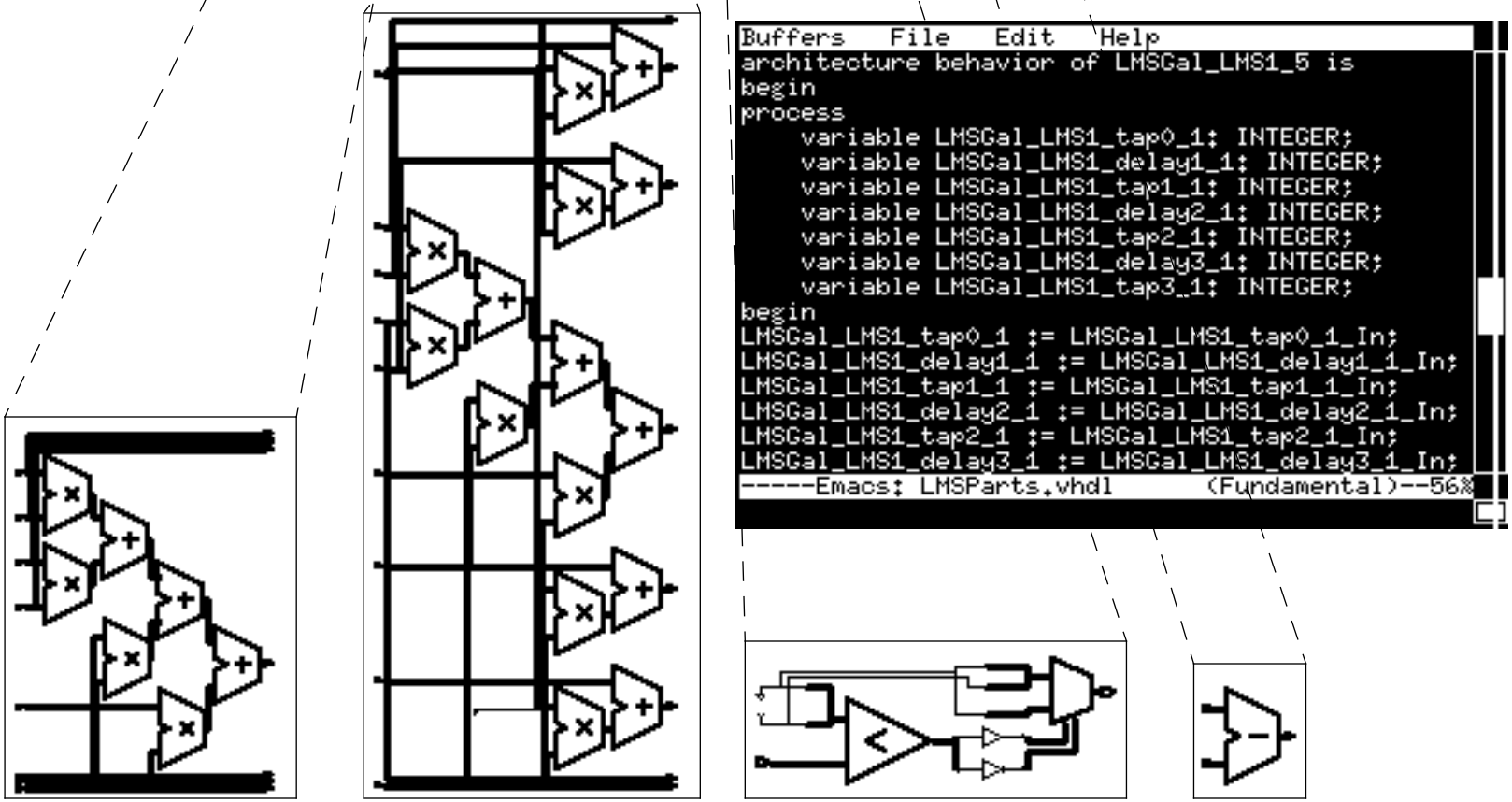
Design Exploration



- Precedence graph exposes concurrency, constraints
- Various options for execution order are possible
- Future goal to explore groupings into execution units

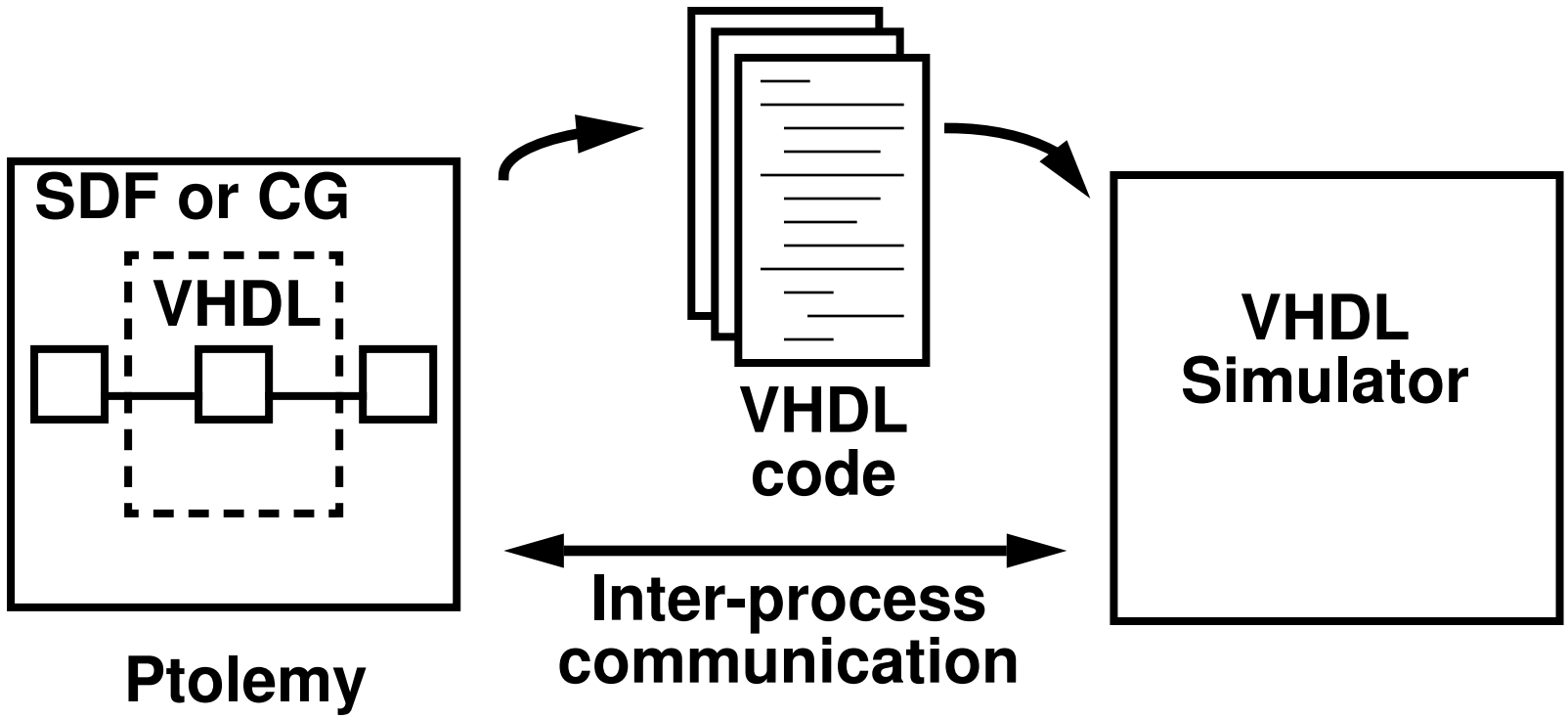
Generating VHDL for Synthesis





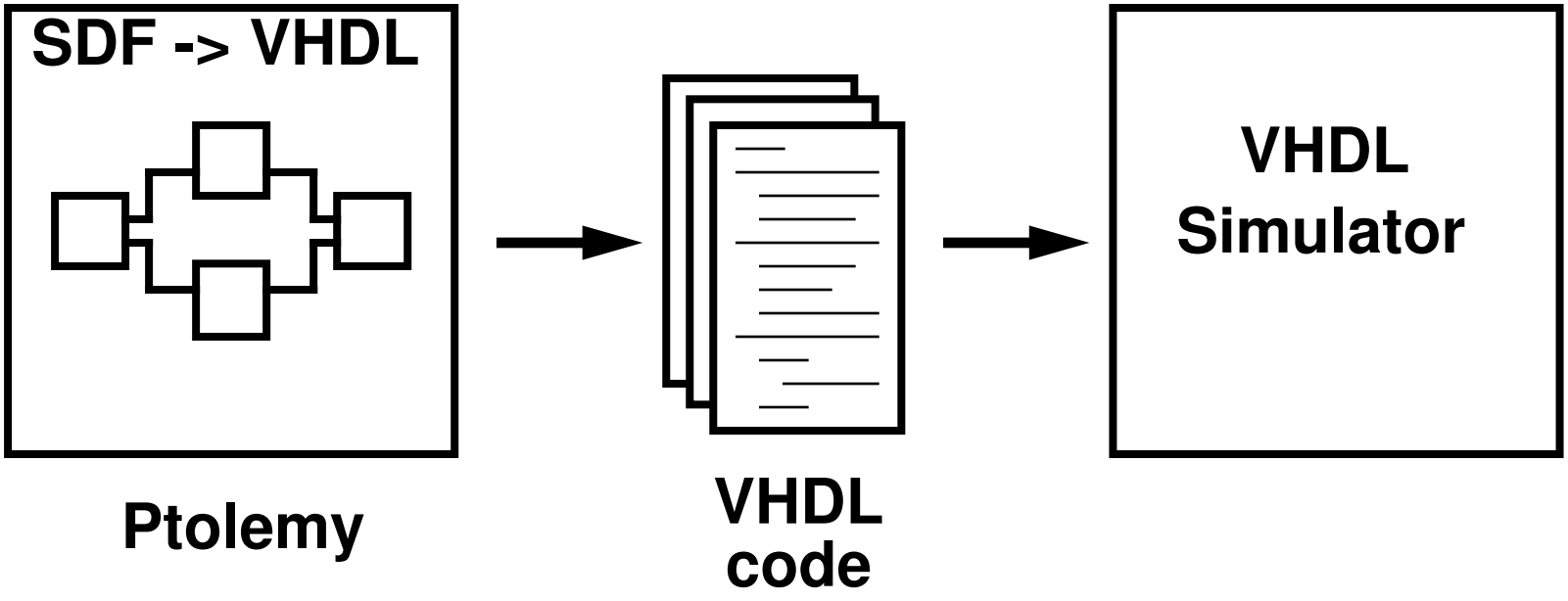
```
Buffers   File   Edit   Help
architecture behavior of LMSGal_LMS1_5 is
begin
process
    variable LMSGal_LMS1_tap0_1: INTEGER;
    variable LMSGal_LMS1_delay1_1: INTEGER;
    variable LMSGal_LMS1_tap1_1: INTEGER;
    variable LMSGal_LMS1_delay2_1: INTEGER;
    variable LMSGal_LMS1_tap2_1: INTEGER;
    variable LMSGal_LMS1_delay3_1: INTEGER;
    variable LMSGal_LMS1_tap3_1: INTEGER;
begin
LMSGal_LMS1_tap0_1 := LMSGal_LMS1_tap0_1_In;
LMSGal_LMS1_delay1_1 := LMSGal_LMS1_delay1_1_In;
LMSGal_LMS1_tap1_1 := LMSGal_LMS1_tap1_1_In;
LMSGal_LMS1_delay2_1 := LMSGal_LMS1_delay2_1_In;
LMSGal_LMS1_tap2_1 := LMSGal_LMS1_tap2_1_In;
LMSGal_LMS1_delay3_1 := LMSGal_LMS1_delay3_1_In;
-----Emacs: LMSParts.vhdl      (Fundamental)--56%
```

Co-Simulation of VHDL Within SDF



- Future goal to simulate VHDL within dataflow
- Ptolemy/SDF for flexible test environment & data visualization
- Ptolemy/CG for heterogeneous system design

Generating VHDL for Simulation



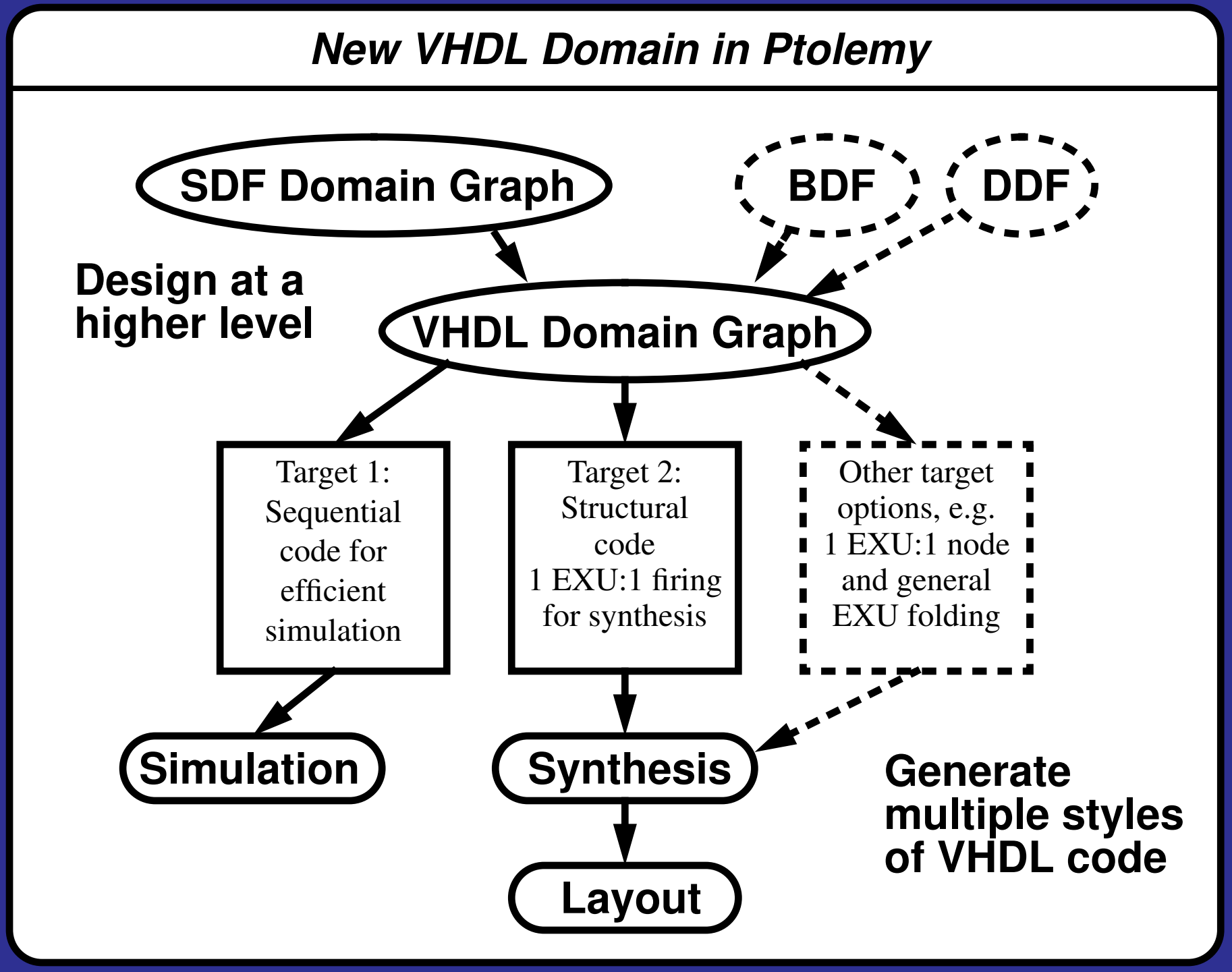
- Retarget from SDF to VHDL, or begin in VHDL
- Generate sequential VHDL based on a valid SDF schedule
- Pass code to external simulator

Design of VHDL Domain Stars

- **VHDL Targets: Polymorphic use of library Stars**
  - — **Sequential VHDL: one process, no internal signals, for efficient simulation**
  - — **Structural VHDL: multiple entities, connected by signals, for input to synthesis tools**

```
defstar {
        name { FIR }
        domain { VHDL }
        desc {
An n-tap FIR filter, with tap values tap0 through
tapn, having states for storing the last n-1 input values.
        }
        codeblock (std) {
$ref(output) $assign(output) $ref(input) * $ref(tap0) + $ref(delay1) *
  $ref(tap1) + $ref(delay2) * $ref(tap2) + $ref(delay3) * $ref(tap3) + ...;
        ...
$ref(delay3) $assign(delay3) $ref(delay2);
$ref(delay2) $assign(delay2) $ref(delay1);
$ref(delay1) $assign(delay1) $ref(input);
        }
}
```

New VHDL Domain in Ptolemy



Motivation

Current Goals:

- **Retarget from SDF to VHDL, then simulate and synthesize**
- **Enable smooth transition from algorithm design to hardware synthesis**
- **Elevate level of design up from coding in VHDL**
- **Ensure equivalence of simulation and synthesized design: correct-by-construction**

Future Goals:

- **Enable high-level design exploration**
- **Begin with SDF, eventually extend to limited dynamic behavior (BDF, DDF)**

Outline

- **Motivation**
- **New VHDL Domain in Ptolemy**
- **VHDL for Simulation**
- **VHDL for Synthesis**
- **Design Exploration**
- **Demonstrations**

VHDL CODE GENERATION FOR SIMULATION AND SYNTHESIS

Michael C. Williamson

Research Advisor: Prof. Edward A. Lee

Ptolemy Review

University of California at Berkeley

March 10, 1995

e-mail: cameron@eecs.berkeley.edu