Introduction to High–Level Synthesis

Chapter 1

Source: Gajski, Dutt, Wu, Lin
"High–Level Synthesis"
NEED FOR HIGH-LEVELS OF ABSTRACTION

- VLSI complexity requires hierarchy
- VLSI technology reached maturity
- First silicon and first specification
- Shorter design cycle
- Better exploration of design space
- Algorithms outperform designers

Two schools of thought:

1. capture-and-simulate
2. describe-and-synthesize
LEVELS OF ABSTRACTION

- **PHYSICAL DOMAIN**
  - Transistor layouts
  - Cells
  - Chips
  - Boards, MCMs

- **STRUCTURAL DOMAIN**
  - Processors, Memories, Buses
  - Registers, ALUs, MUXs
  - Gates, flip-flops
  - Transistors

- **BEHAVIORAL DOMAIN**
  - System synthesis
  - Register–transfer synthesis
  - Logic synthesis
  - Circuit synthesis
  - Transistor functions

- **PROCESSOR SYNTHESIS**
  - Flowcharts, algorithms
  - Register transfers
  - Boolean expressions
if IR(3) = '0' then
  PC := PC + 1;
else
  DBUF := MEM(PC);
  MEM(SP) := PC + 1;
  SP := SP - 1;
  PC := DBUF;
end if;

BEHAVIOR

STRUCTURE

FLOORPLAN
DEFINITION OF SYNTHESIS

Behavior–to–structure

- Circuit synthesis
- Logic synthesis
- Register–transfer synthesis
- System synthesis

Structure–to–layout

- Cell layout generation
- Module layout generation
- Chip floorplanning
- System partitioning and placement
Several descriptions for the same behavior
Several styles for the same description
Different abstractions for the same design
if CNT ≠ LIM then
    EN <= ENIT;
else
    EN <= '0';
end if;

Level sensitive

if ENIT = '1' and not ENIT'stable then
    EN <= '1';
elseif CNT = LIM then
    EN <= '0';
end if;

Edge sensitive
DIFFERENT STYLES FOR THE SAME DESCRIPTIONS

Transmission gates

AND–OR–INVERT gate
DIFFERENT CONSTRUCTS FOR THE SAME BEHAVIOR

1 state  \( \text{(no status register)} \)
if \( x = 0 \) then \( y = a+b \) else \( y = a-b \)

2 states  \( \text{(with status register)} \)
if \( x = 0 \) then status = 1
if status = 1 then \( y = a+b \) else \( y = a-b \)