EE3302 DIGITAL LOGIC CIRCUITS

VHDL

- What is VHDL?

V*H I S C* → *Very High Speed Integrated Circuit*

H*ardware*

D*escription*

L*anguage*

IEEE Standard 1076-1993

History of VHDL

- Designed by IBM, Texas Instruments, and Intermetrics as part of the DoD funded VHSIC program
- Standardized by the IEEE in 1987: IEEE 1076-1987
- Enhanced version of the language defined in 1993: IEEE 1076-1993
- Additional standardized packages provide definitions of data types and expressions of timing data
  - IEEE 1164 (data types)
  - IEEE 1076.3 (numeric)
  - IEEE 1076.4 (timing)

Traditional vs. Hardware Description Languages

- Procedural programming languages provide the *how* or recipes
  - for computation
  - for data manipulation
  - for execution on a specific hardware model
- Hardware description languages *describe* a system
  - Systems can be described from many different points of view
    - Behavior: what does it do?
    - Structure: what is it composed of?
    - Functional properties: how do I interface to it?
    - Physical properties: how fast is it?

Usage

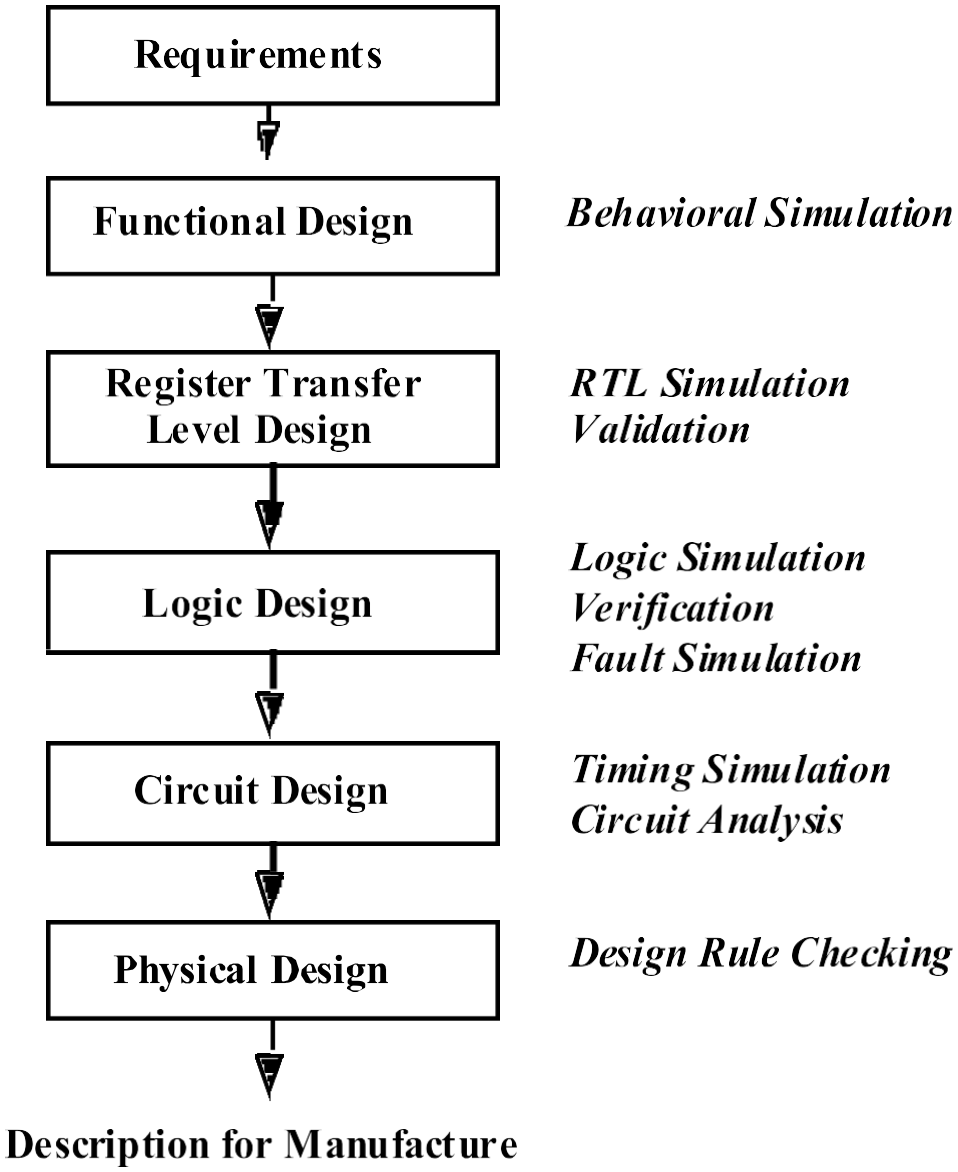
- Descriptions can be at different levels of abstraction
  - Switch level: model switching behavior of transistors
  - Register transfer level: model combinational and sequential logic components
  - Instruction set architecture level: functional behavior of a microprocessor

- Descriptions can used for
  - Simulation
    - Verification, performance evaluation
  - Synthesis
    - First step in hardware design

Why do we Describe Systems?

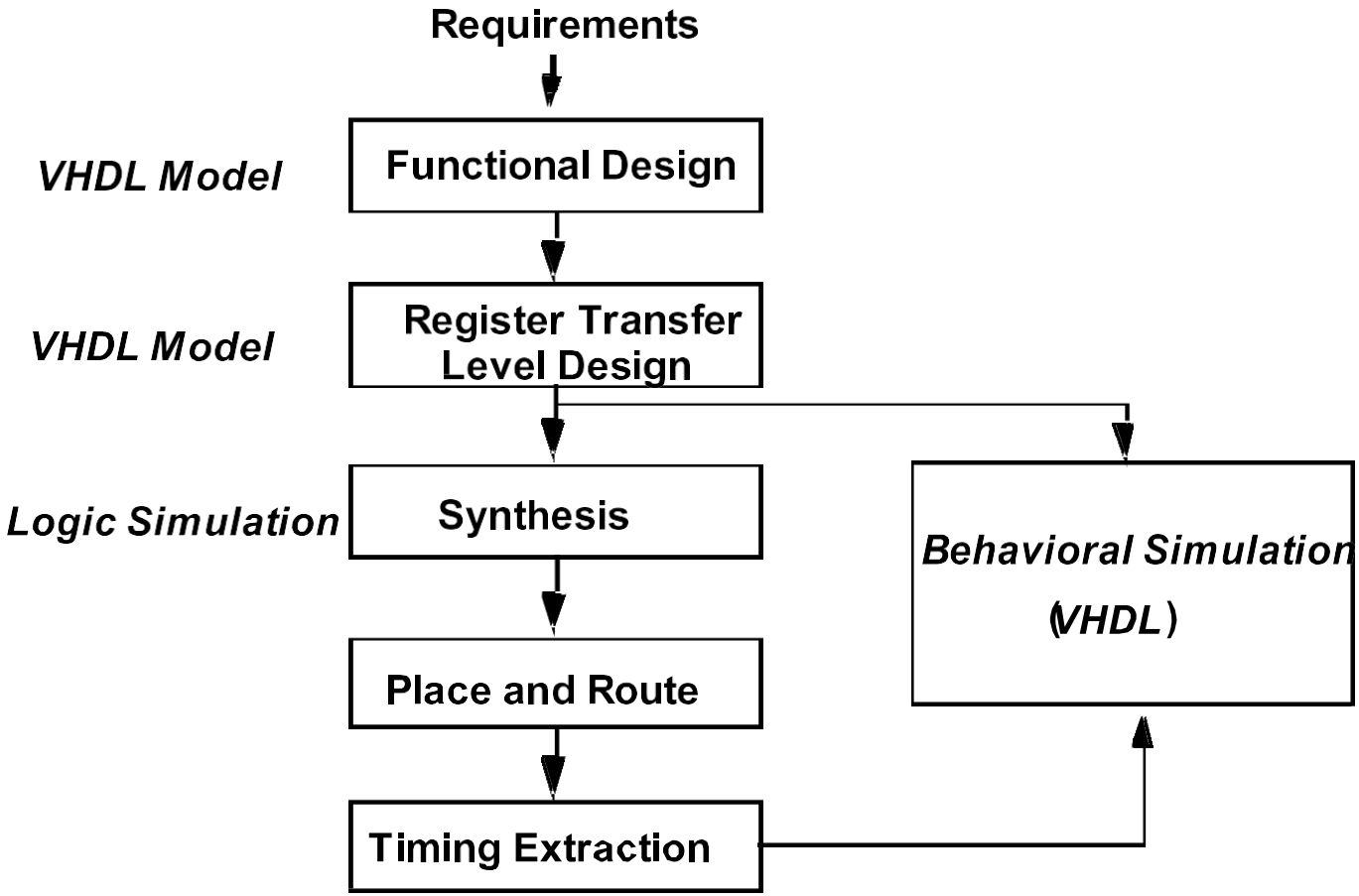
- Design Specification
  - unambiguous definition of components and interfaces in a large design
- Design Simulation
  - verify system/subsystem/chip performance prior to design implementation
- Design Synthesis
  - automated generation of a hardware design

Digital System Design Flow



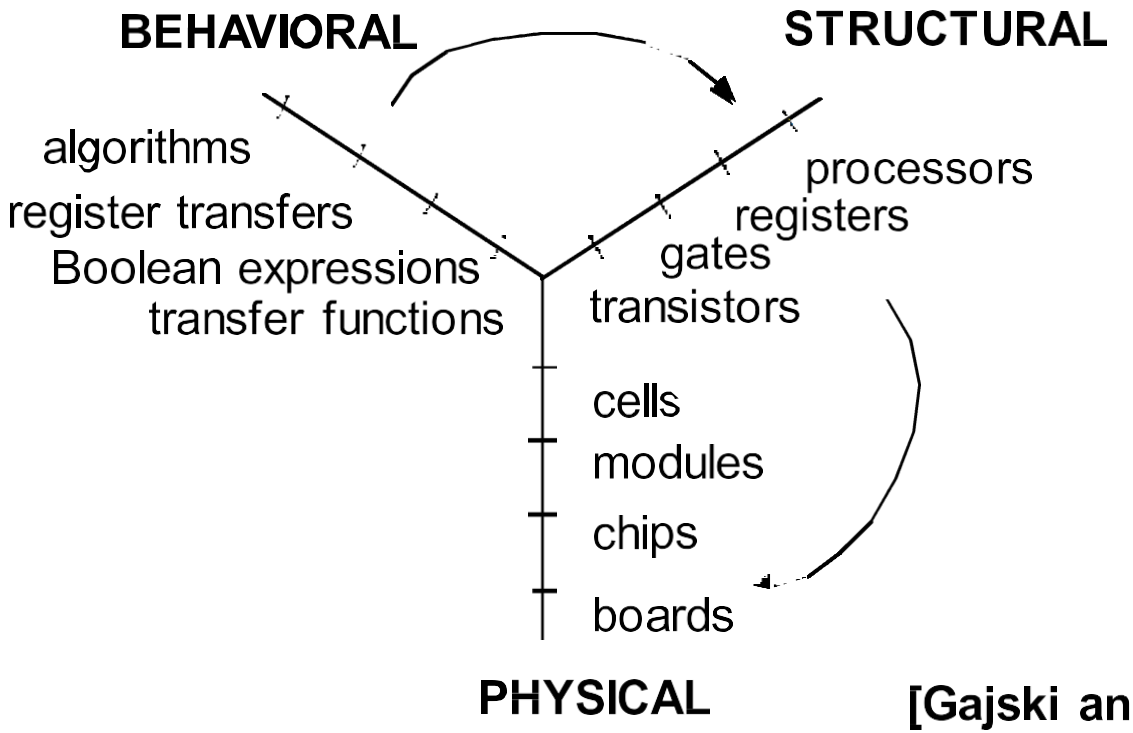
- Design flows operate at multiple levels of abstraction
- Need a uniform description to translate between levels
- Increasing costs of design and fabrication necessitate greater reliance on automation via CAD tools
  - $5M - $100M to design new chips
  - Increasing time to market pressures

A Synthesis Design Flow



- Automation of design refinement steps
- Feedback for accurate simulation
- Example targets: ASICs, FPGAs

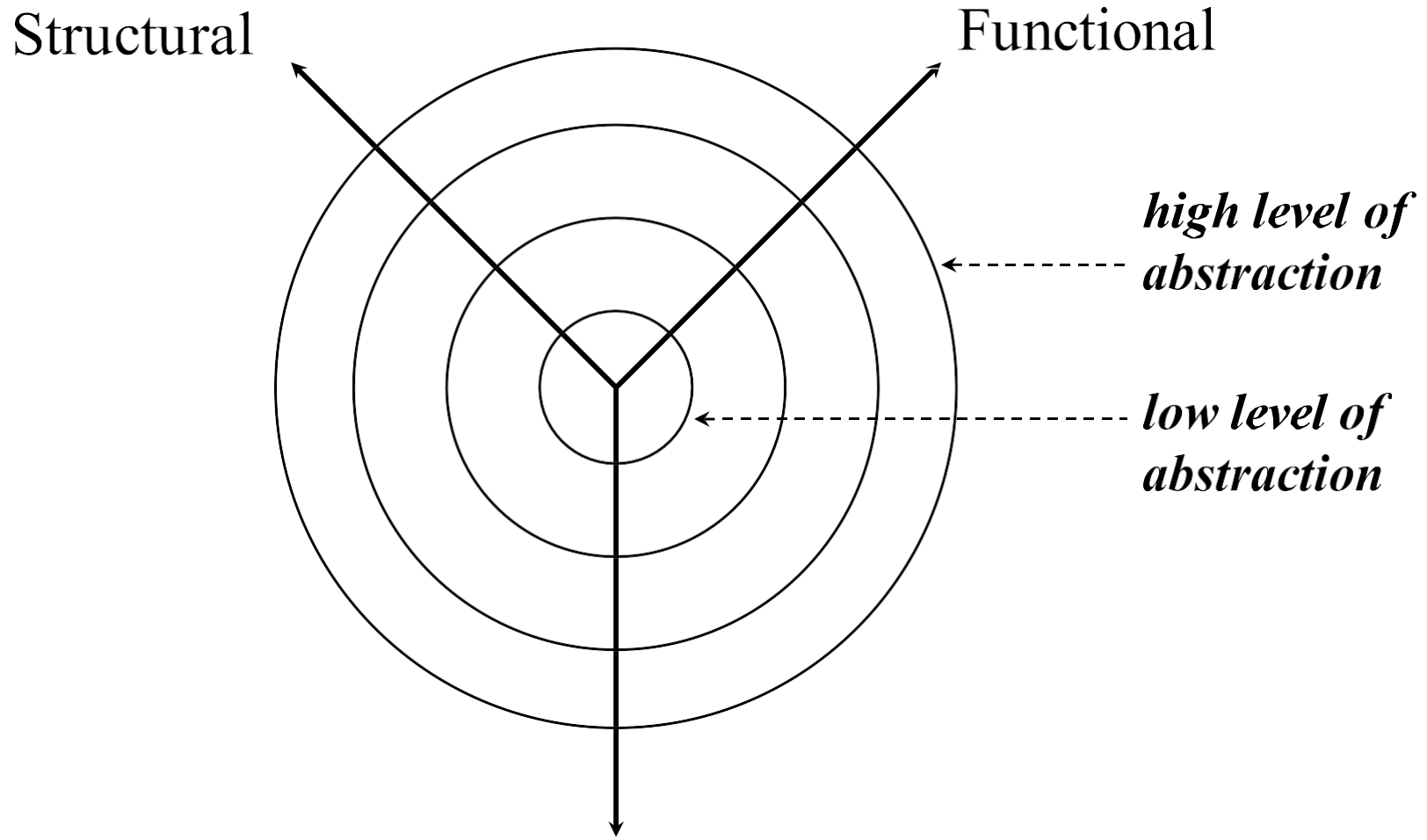
The Role of Hardware Description Languages



[Gajski and Kuhn]

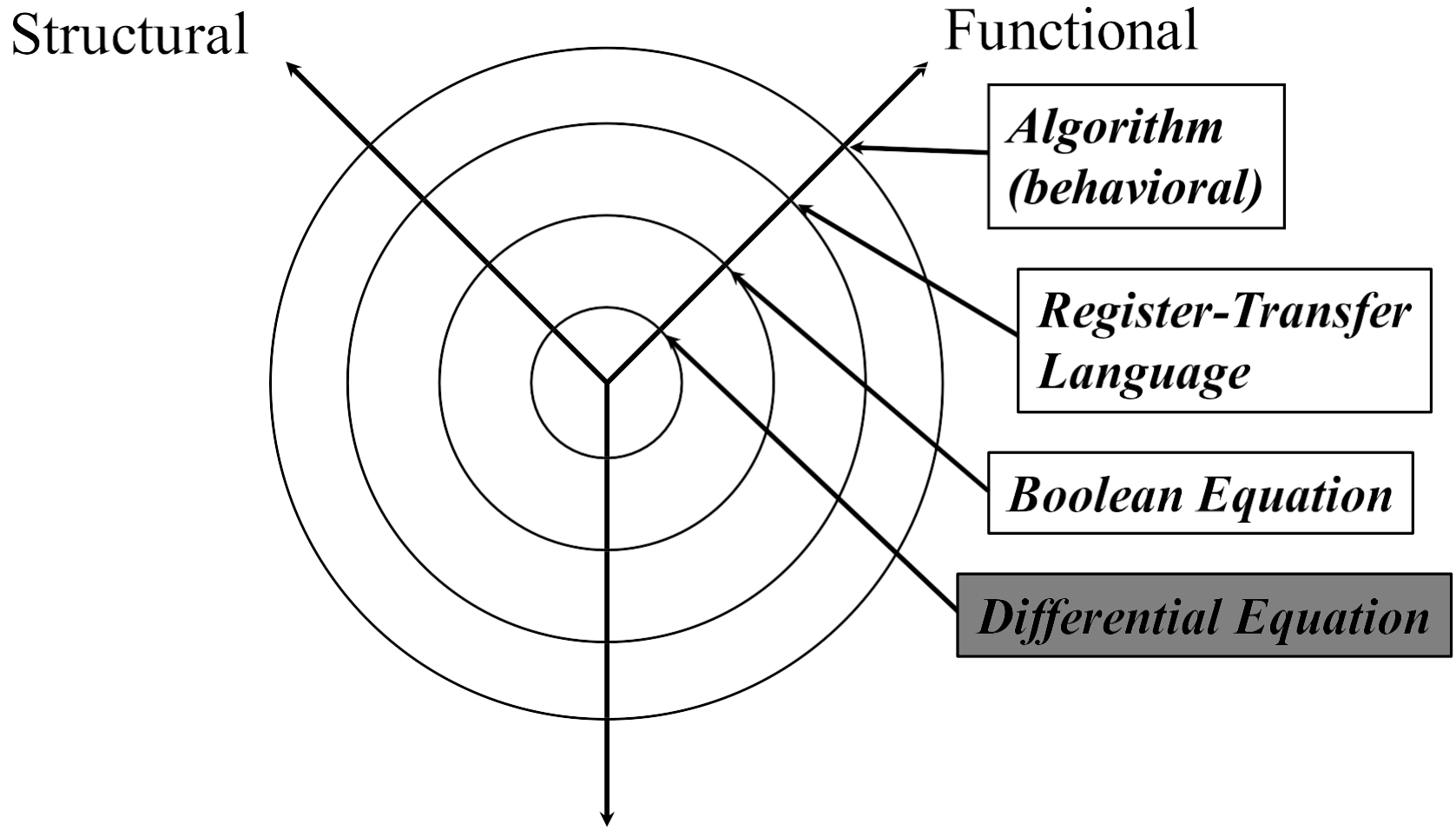
- Design is structured around a hierarchy of representations
- HDLs can describe distinct aspects of a design at multiple levels of abstraction

Domains and Levels of Modeling



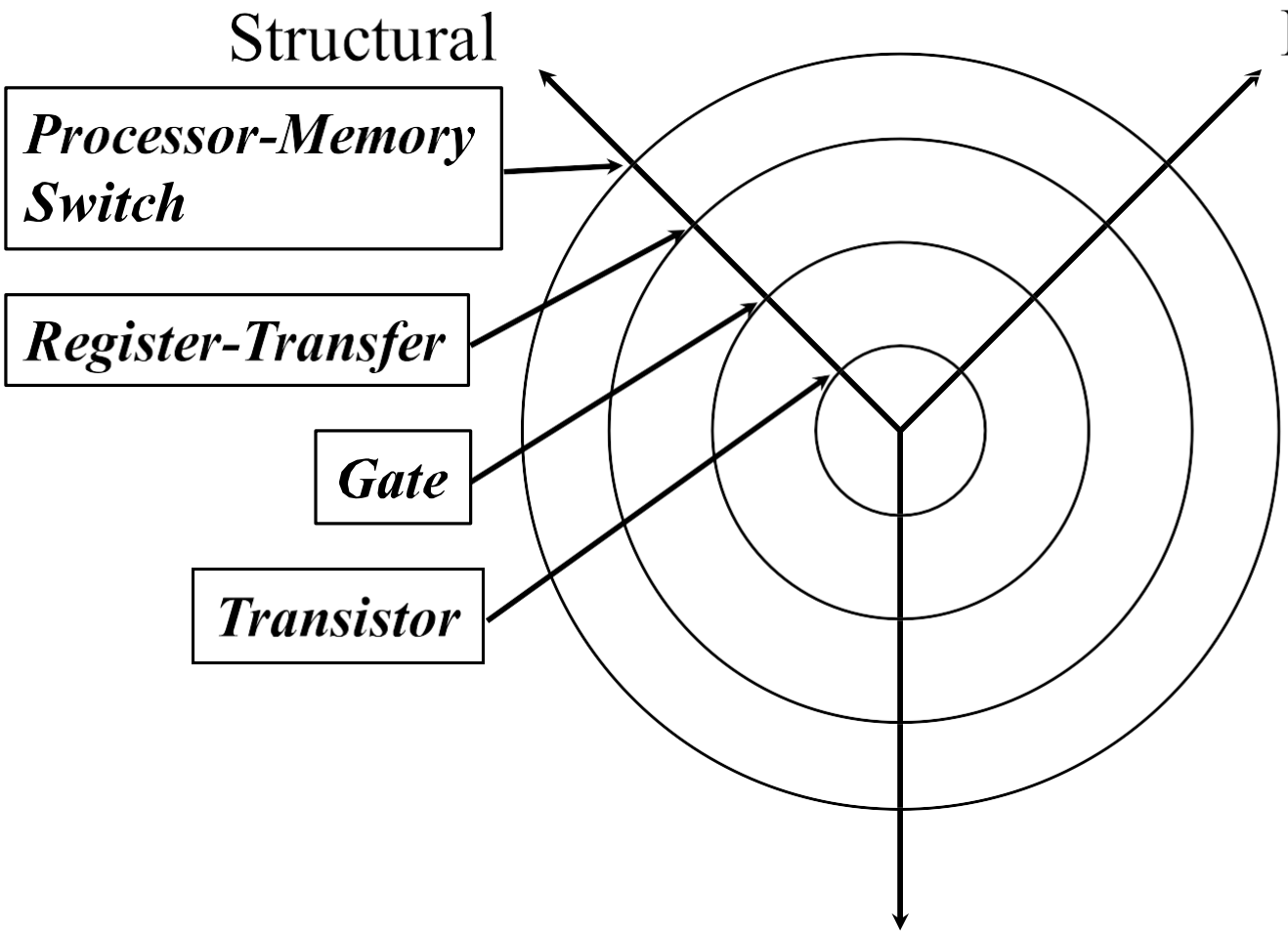
"Y-chart" due to Gajski & Kahn

Domains and Levels of Modeling



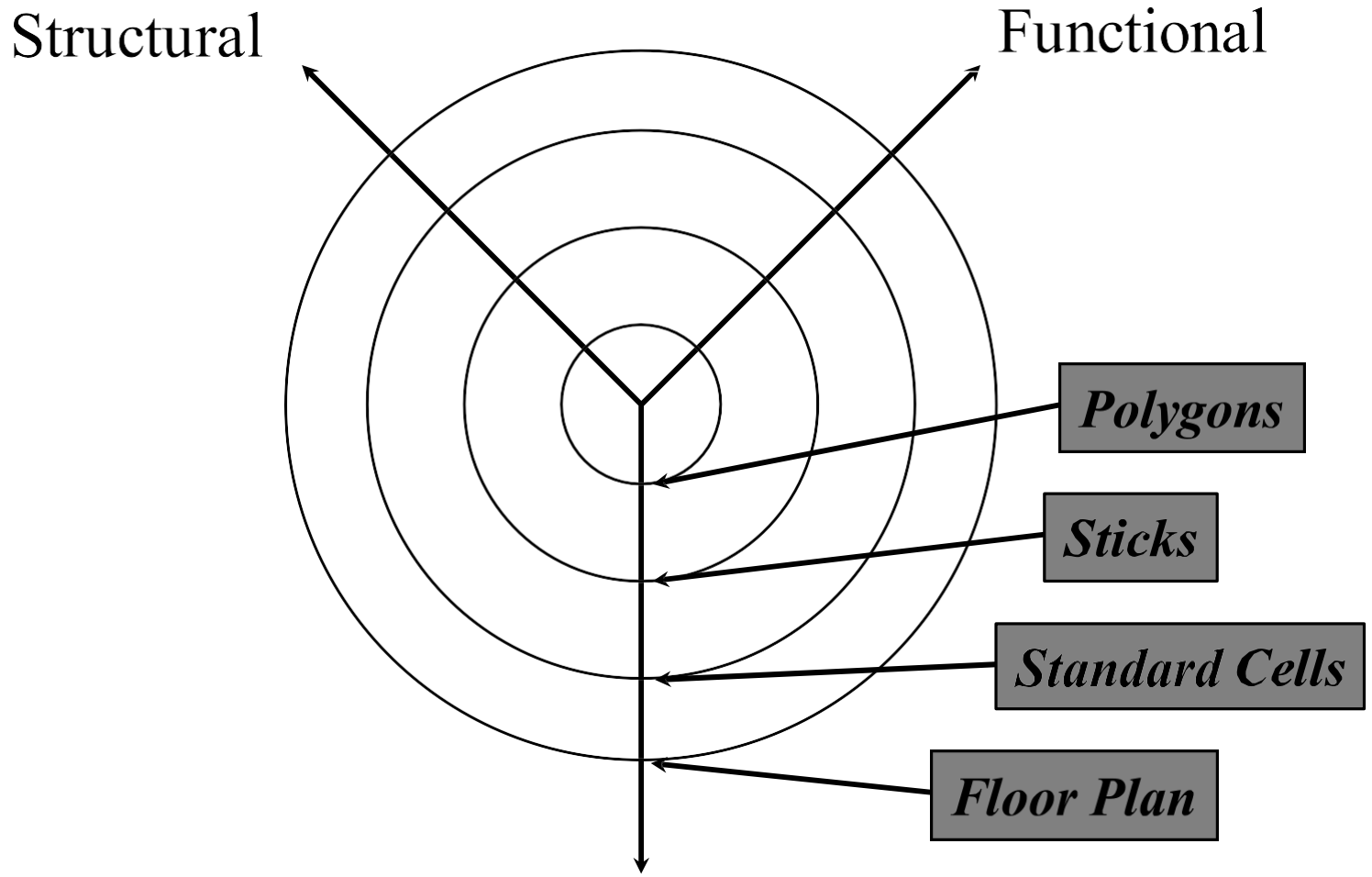
"Y-chart" due to Gajski & Kahn

Domains and Levels of Modeling



"Y-chart" due to Gajski & Kahn

Domains and Levels of Modeling



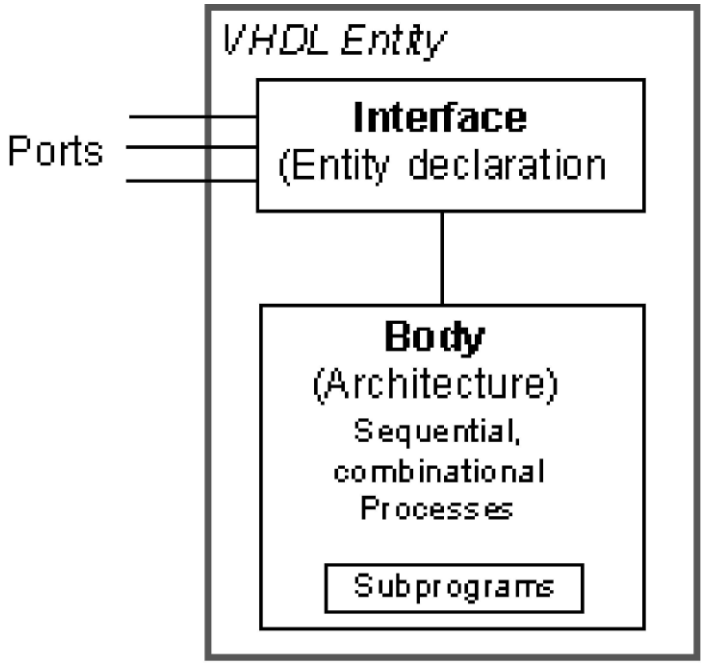
"Y-chart" due to Gajski & Kahn

Basic VHDL Concepts

- Interfaces
- Modeling (Behavior, Dataflow, Structure)
- Test Benches
- Analysis, elaboration, simulation
- Synthesis

Basic Structure of a VHDL File

- Entity
  - Entity declaration: interface to outside world; defines input and output signals
  - Architecture: describes the entity, contains processes, components operating concurrently



Entity Declaration

entity NAME_OF_ENTITY **is**
port (*signal_names*: **mode** *type*;
signal_names: **mode** *type*;
:
signal_names: **mode** *type*);
end [NAME_OF_ENTITY] ;

MVL - 9			
Uninitialized	'U'	Weak 1	'H'
Don't Care	'-'	Weak 0	'L'
Forcing 1	'1'	Weak Unknown	'W'
Forcing 0	'0'	High Impedance	'Z'
Forcing Unknown	'X'		

- NAME_OF_ENTITY: user defined
- signal_names: list of signals (both input and output)
- mode: in, out, buffer, inout
- type: boolean, integer, character, std_logic

Architecture

- **Behavioral Model:**

architecture architecture_name **of** NAME_OF_ENTITY
is

-- Declarations

…..

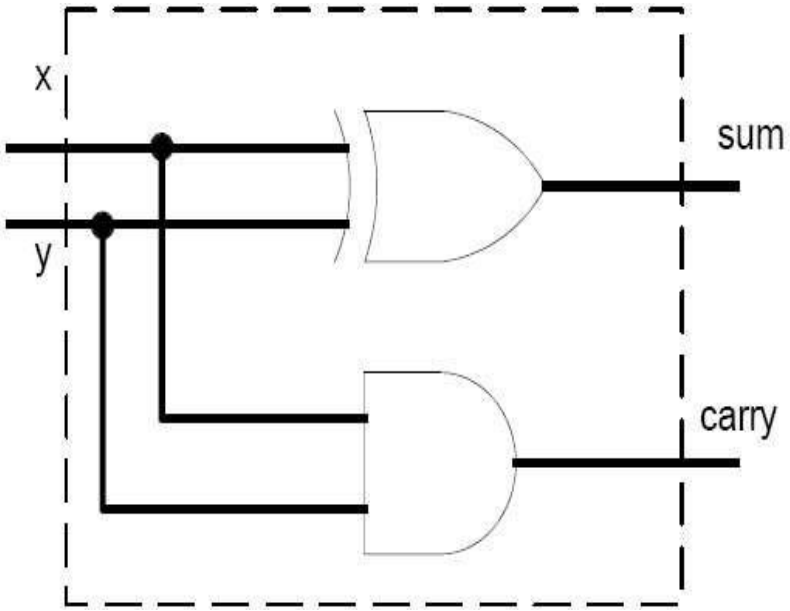
…..

begin

-- Statements

end architecture_name;

Half Adder




```
library ieee;
use ieee.std_logic_1164.all;
entity half_adder is
port(
        x,y: in std_logic;
        sum, carry: out std_logic);
end half_adder;

architecture myadd of half_adder is
    begin
        sum <= x xor y;
        carry <= x and y;
end myadd;
```

Entity Examples …

```
entity half_adder is
port(
        x,y: in std_logic;
        sum, carry: out std_logic);
end half_adder;
```

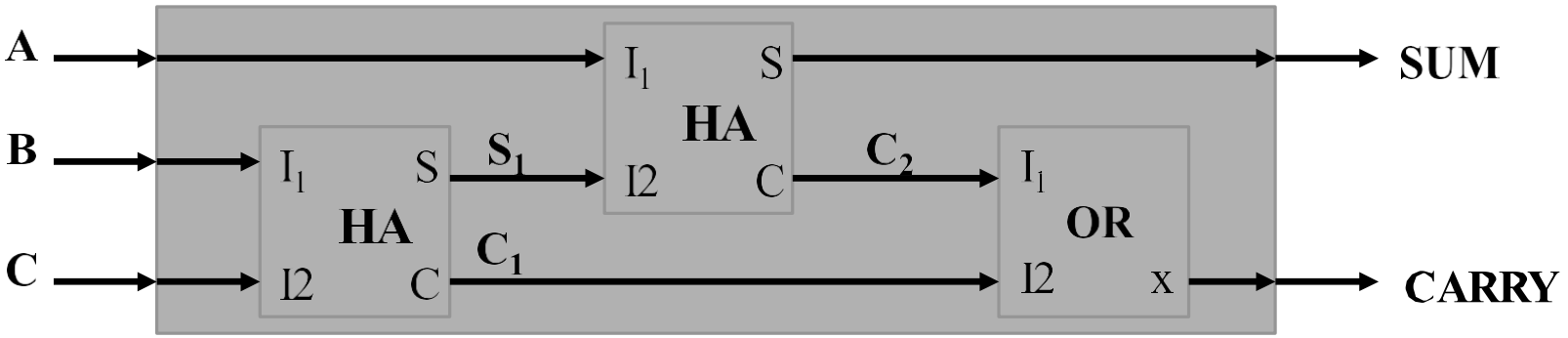
Architecture Examples: Behavioral Description

- Entity FULLADDER is
  port (A, B, C: in std_logic;
  SUM, CARRY: in std_logic);
  end FULLADDER;

- Architecture CONCURRENT of FULLADDER is
  begin
  SUM <= A xor B xor C after 5 ns;
  CARRY <= (A and B) or (B and C) or (A and C) after 3 ns;
  end CONCURRENT;

Architecture Examples: Structural Description …

- architecture STRUCTURAL of FULLADDER is
  signal S1, C1, C2 : bit;
  component HA
    port (I1, I2 : in bit; S, C : out bit);
  end component;
  component OR
    port (I1, I2 : in bit; X : out bit);
  end component;
  begin
  INST_HA1 : HA port map (I1 => B, I2 => C, S => S1, C => C1);
  INST_HA2 : HA port map (I1 => A, I2 => S1, S => SUM, C => C2);
  INST_OR : OR port map (I1 => C2, I2 => C1, X => CARRY);
  end STRUCTURAL;



… Architecture Examples: Structural Description

```
Entity HA is
PORT (I1, I2 : in bit; S, C : out bit);
end HA ;
Architecture behavior of HA is
begin
   S <= I1 xor I2;
   C <= I1 and I2;
end behavior;
```

```
Entity OR is
PORT (I1, I2 : in bit; X : out bit);
end OR ;
Architecture behavior of OR is
begin
   X <= I1 or I2;
end behavior;
```

Thank You