Introduction to VHDL

By: Dr. M. SHIPLE

Overview

Tools

PROJECT

VHDL

ISE

Frequency Counter

Eyes on VHDL History

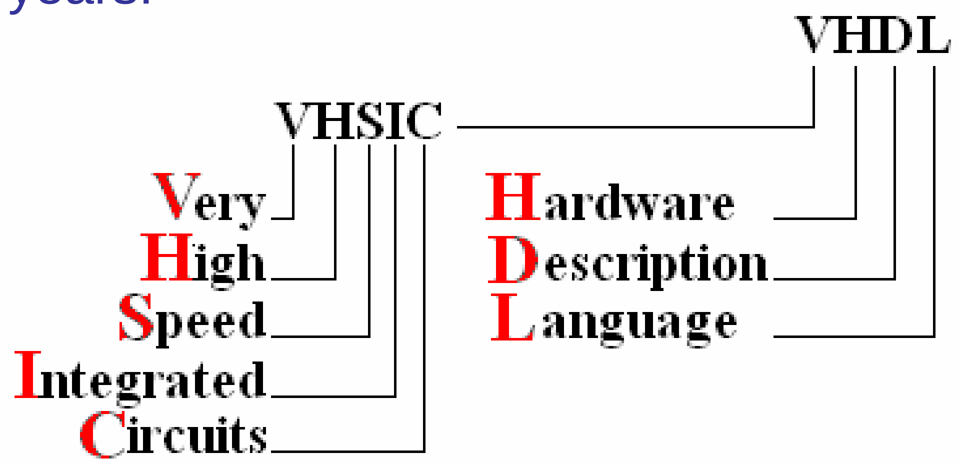
- VHDL is a hardware description language that can be used to model a digital system.
- 1981: the U.S. Department of Defense and the IEEE initiated HDL project.

The standard revised every five years:

–IEEE Std 1076-1987

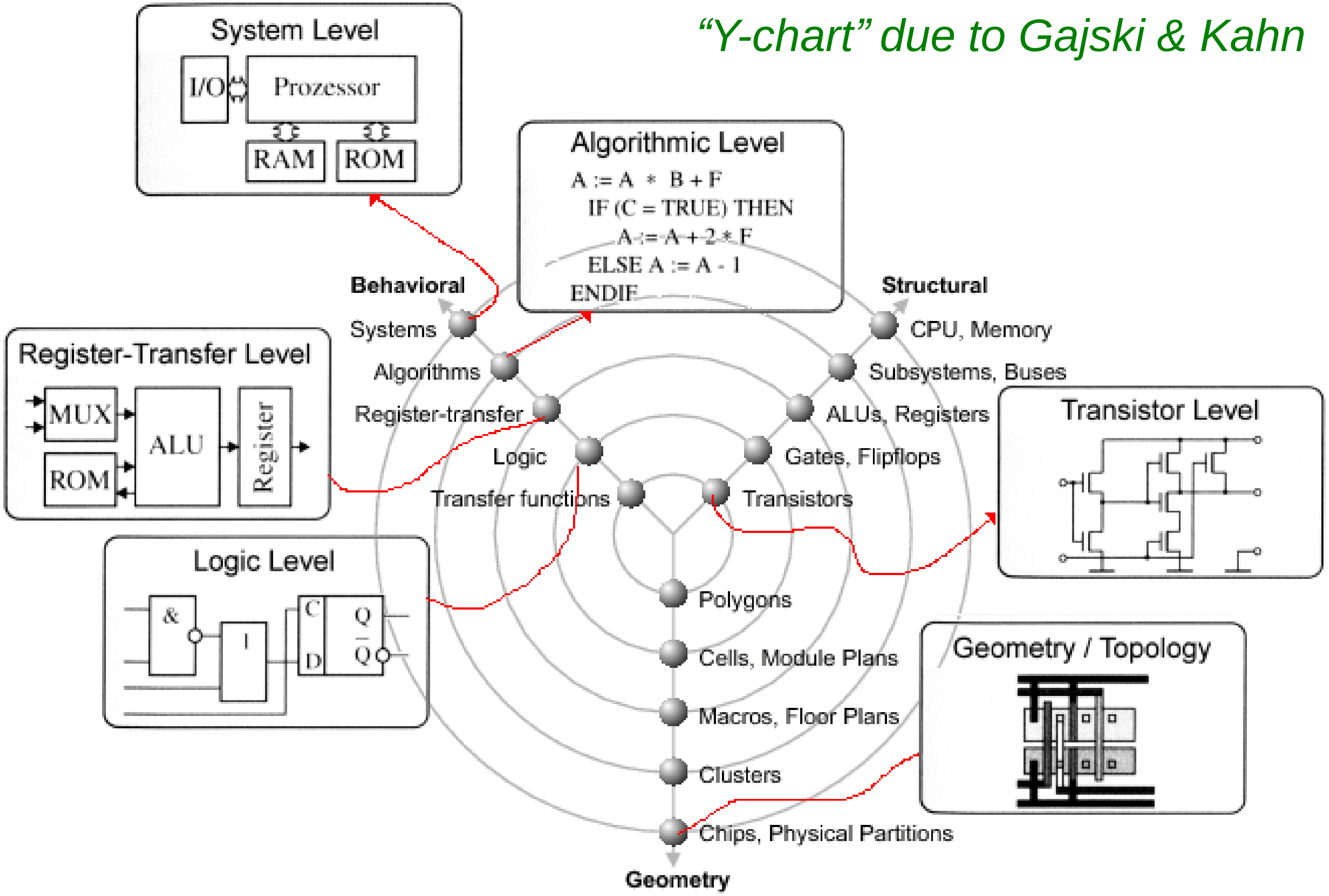
–IEEE Std 1076-1993

–IEEE Std 1164-1993

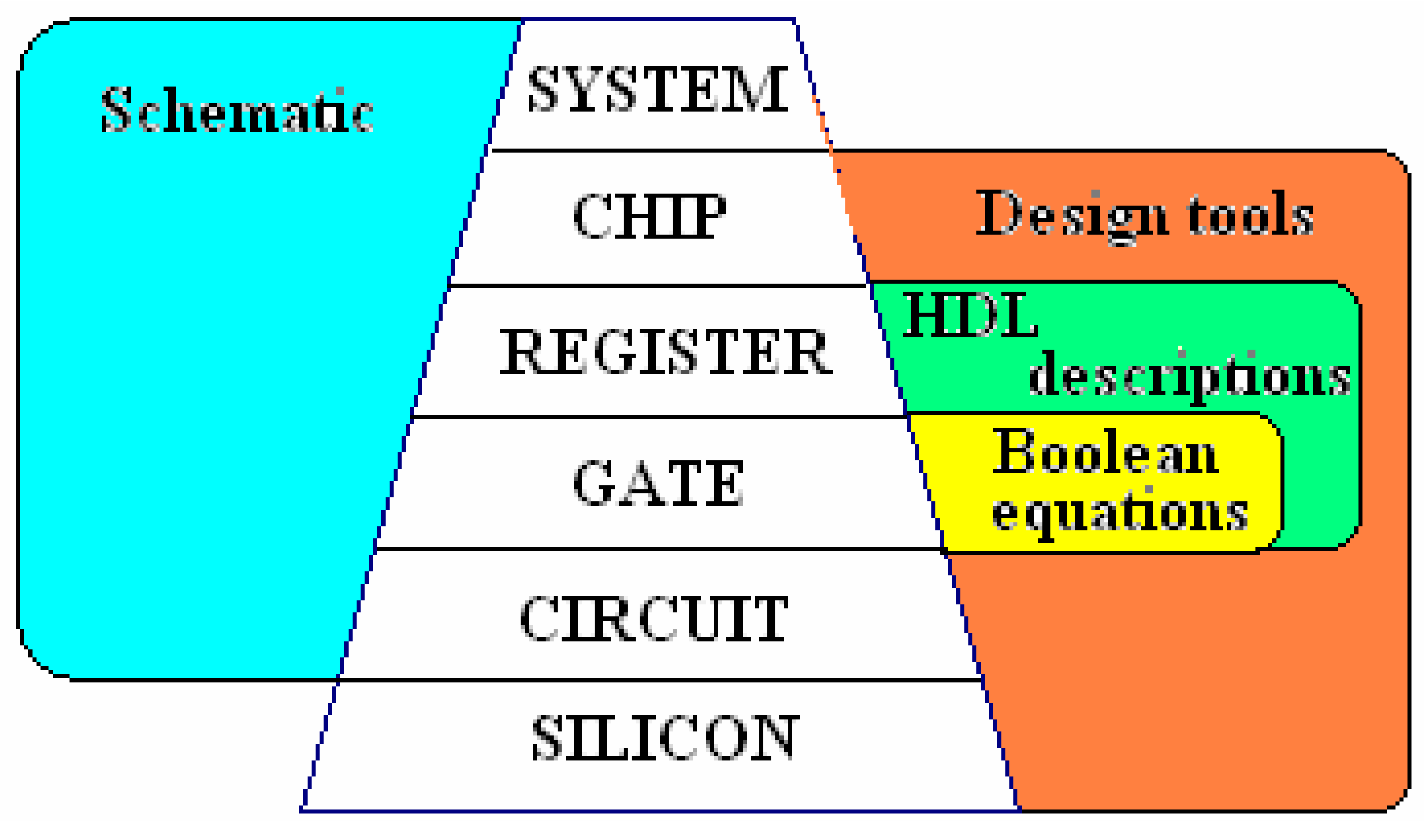


Different Description Levels

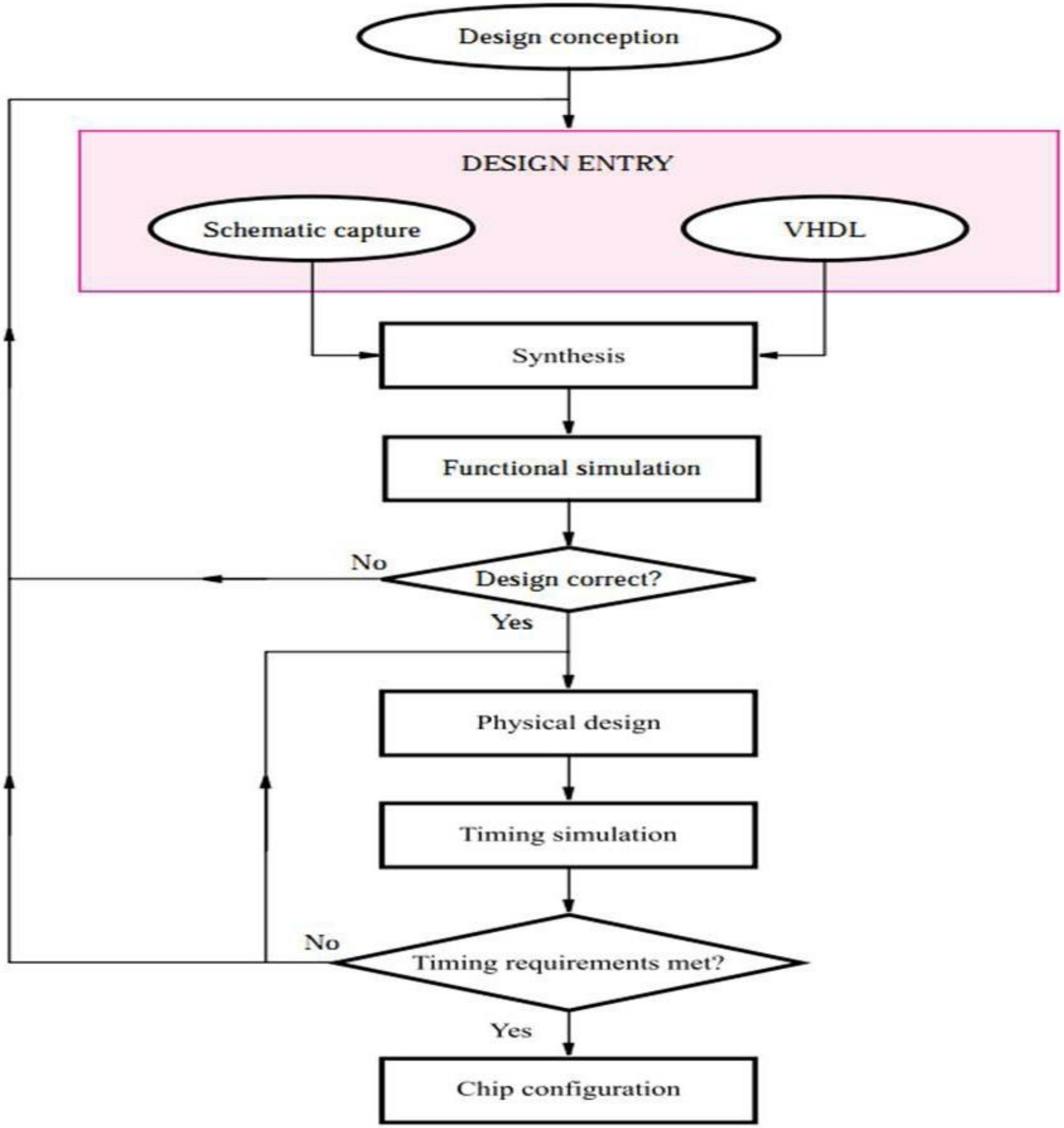
“Y-chart” due to Gajski & Kahn



Available tools



Design Flow


Design conception
DESIGN ENTRY
Schematic capture
VHDL
Synthesis
Functional simulation
No
Design correct?
Yes
Physical design
Timing simulation
No
Timing requirements met?
Yes
Chip configuration

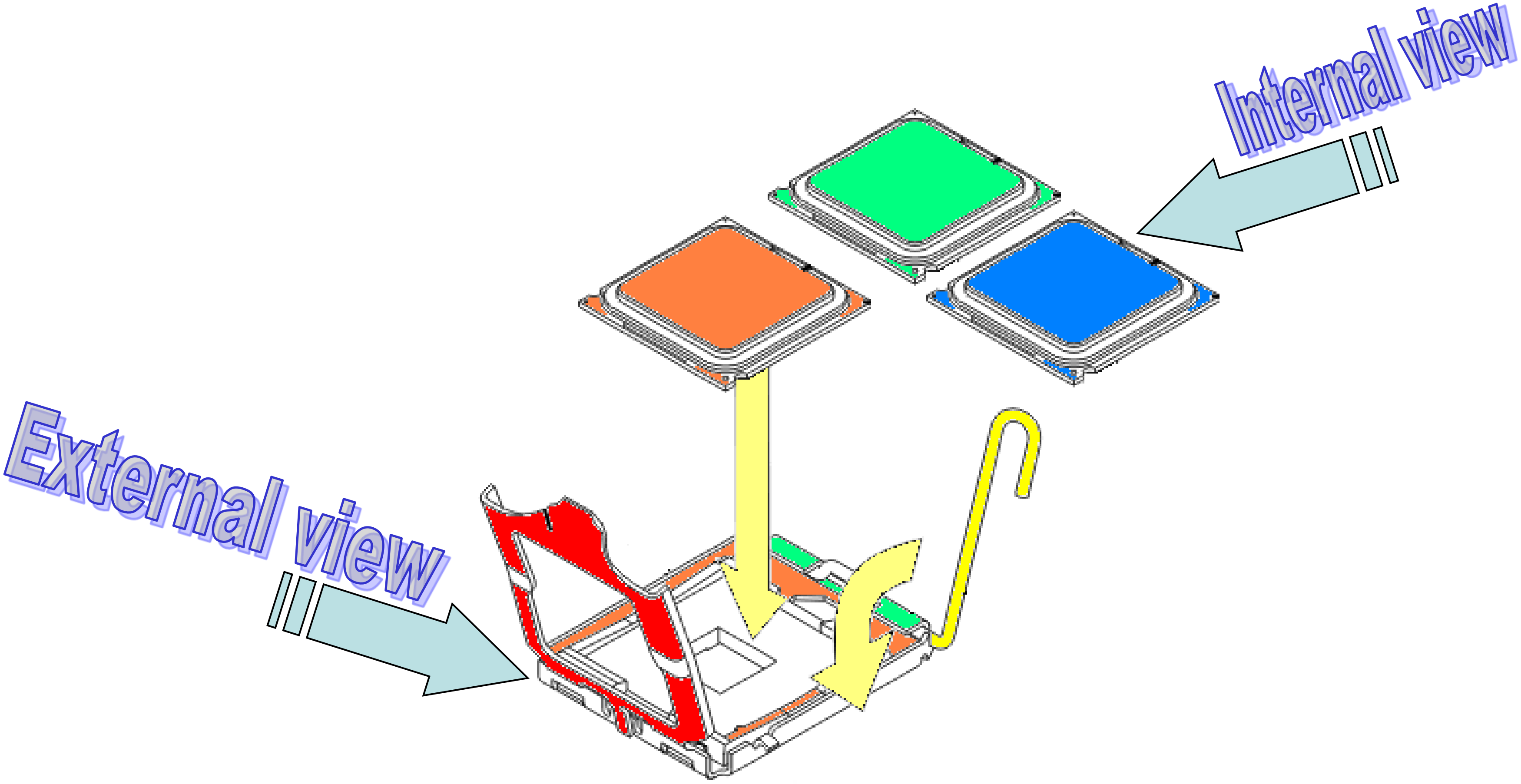
Hardware Abstraction

External view

HDL

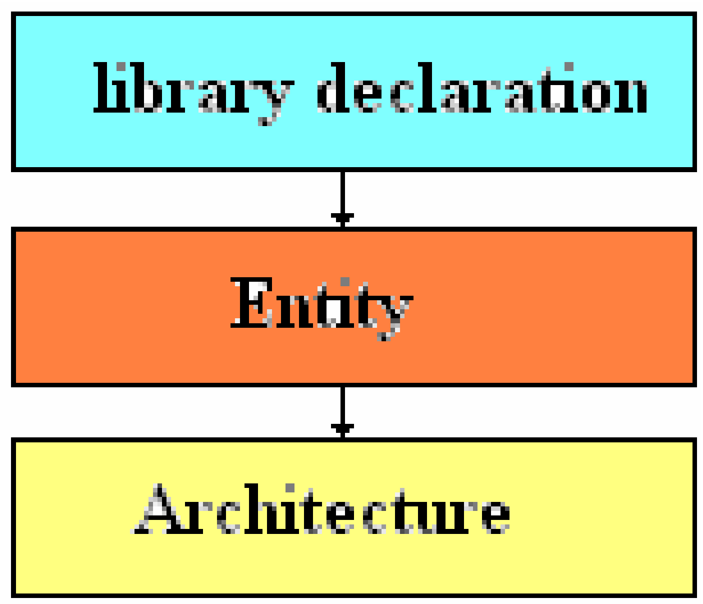
Internal view

Multiple Architecture



VHDL structure

- VHDL used to **model digital systems**



library declaration

Entity

describes the external view

Architecture

contains the internal description of the entity

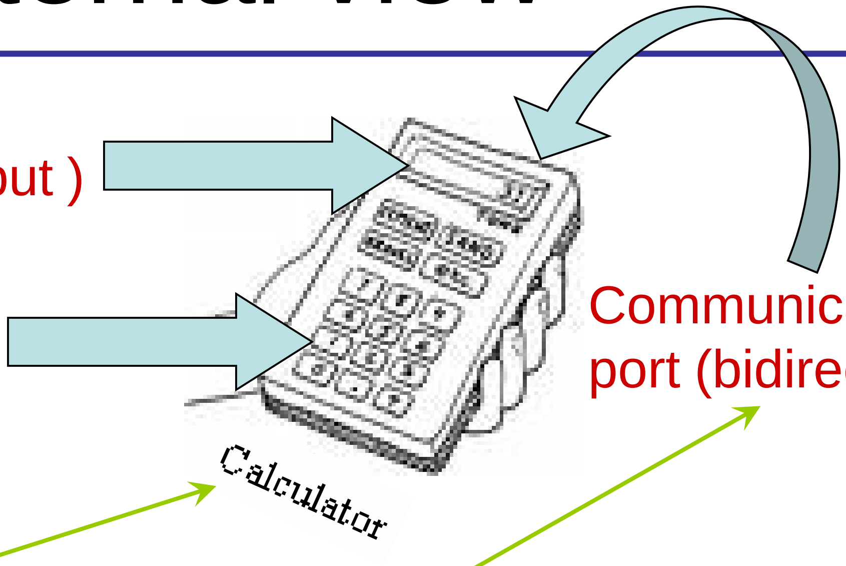
Library

Libraries provide a set of packages, components, and functions that simplify the task of designing hardware

```
LIBRARY library_name;
USE library_name.package_name.package_parts;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
```

External view



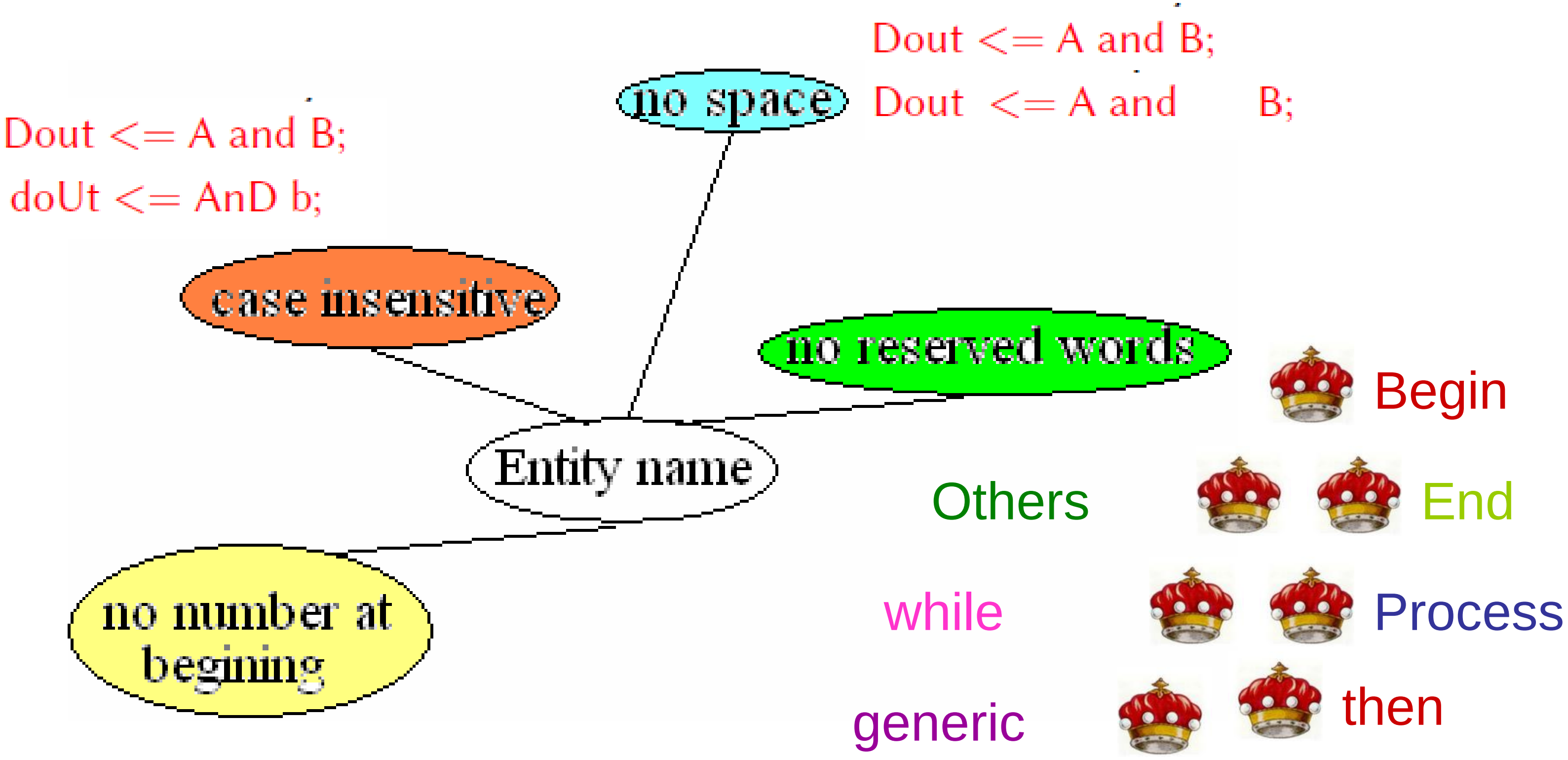
Display (output)

Dip switches (Input)

Communication port (bidirectional)

```
ENTITY some_entity IS
PORT (  port_name1 : port_mode signaltype;
        port_name1 : port_mode signaltype
      );
END some_entity;
```

Reserved words



```
Dout <= A and B;
Dout  <= A and      B;
```

```
Dout <= A and B;
doUt <= AnD b;
```

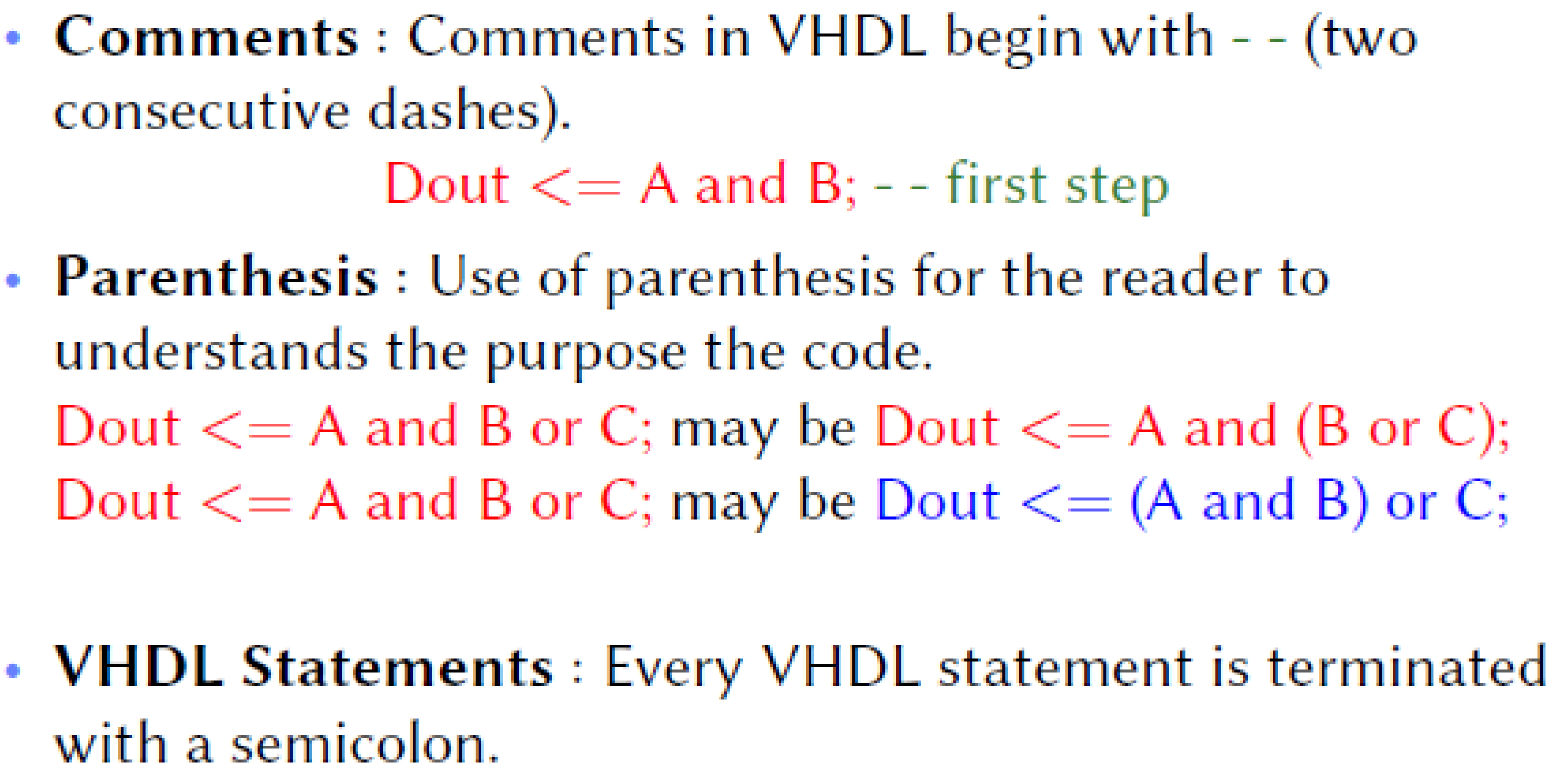
Begin

Others End

while Process

generic then

VHDL Invariants



- **Comments** : Comments in VHDL begin with - - (two consecutive dashes).

  Dout <= A and B; - - first step

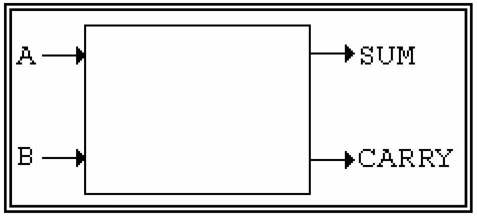
- **Parenthesis** : Use of parenthesis for the reader to understands the purpose the code.

  Dout <= A and B or C; may be Dout <= A and (B or C);

  Dout <= A and B or C; may be Dout <= (A and B) or C;

- **VHDL Statements** : Every VHDL statement is terminated with a semicolon.

Entity Example



port names

entity name

port mode (direction)

port type

```
entity HALFADD is
port (A,B               : in     std_logic ;
      SUM, CARRY        : out    std_logic );
end HALFADD;
```

Semicolon

punctuation

Definitions

```
entity HALFADD is
port (A,B               : in    std_logic ;
      SUM, CARRY        : out   std_logic );
end HALFADD;
```

In

Out

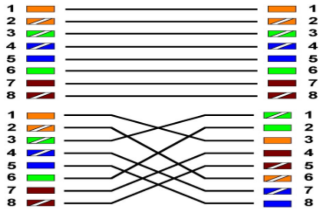
Inout

Buffer

Std_logic

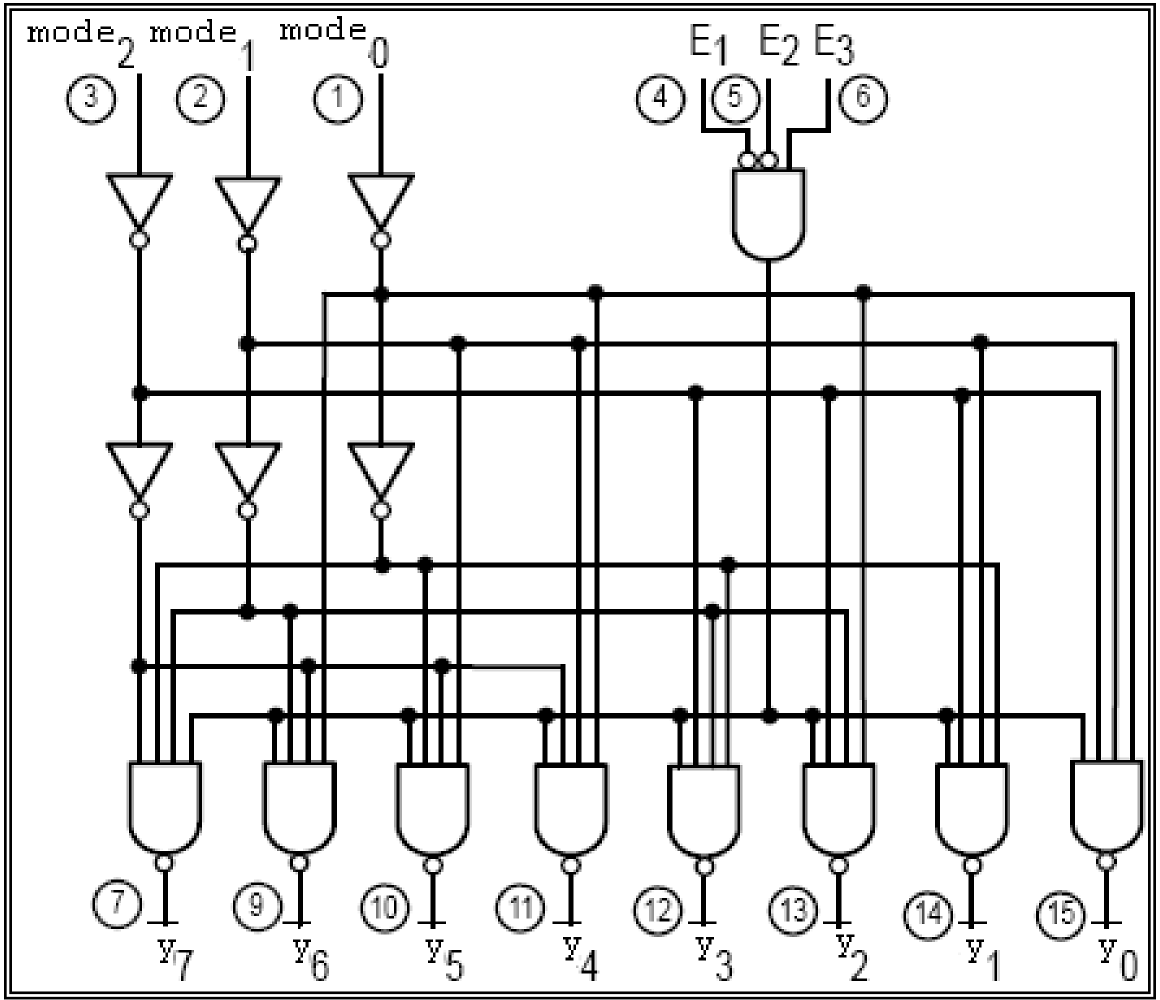
Std_logic_vector (7 downto 0)

Std_logic_vector (0 to 7)

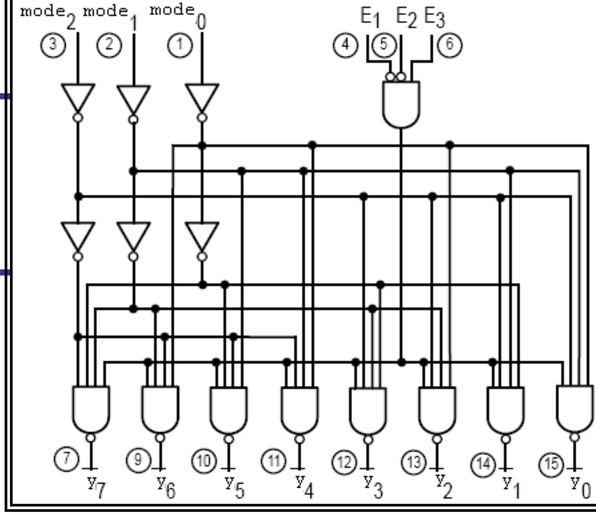


Exer page 5

Exercise

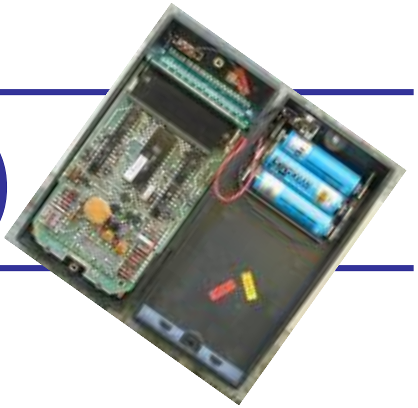

mode2
mode1
mode0
E1
E2
E3
3
2
1
4
5
6
7
9
10
11
12
13
14
15
y7
y6
y5
y4
y3
y2
y1
y0

Exercise



```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY 74ls138  IS
PORT (        y     : out  std_logic_vector (7 DOWNTO 0) ;
           mode  : IN   std_logic_vector (2 DOWNTO 0) ;
                 E1 : IN   std_logic;
                 E2 : IN   std_logic;
                 E3 : IN   std_logic) ;
END ENTITY 74ls138 ;
```

Architecture (internal view)



Architecture name

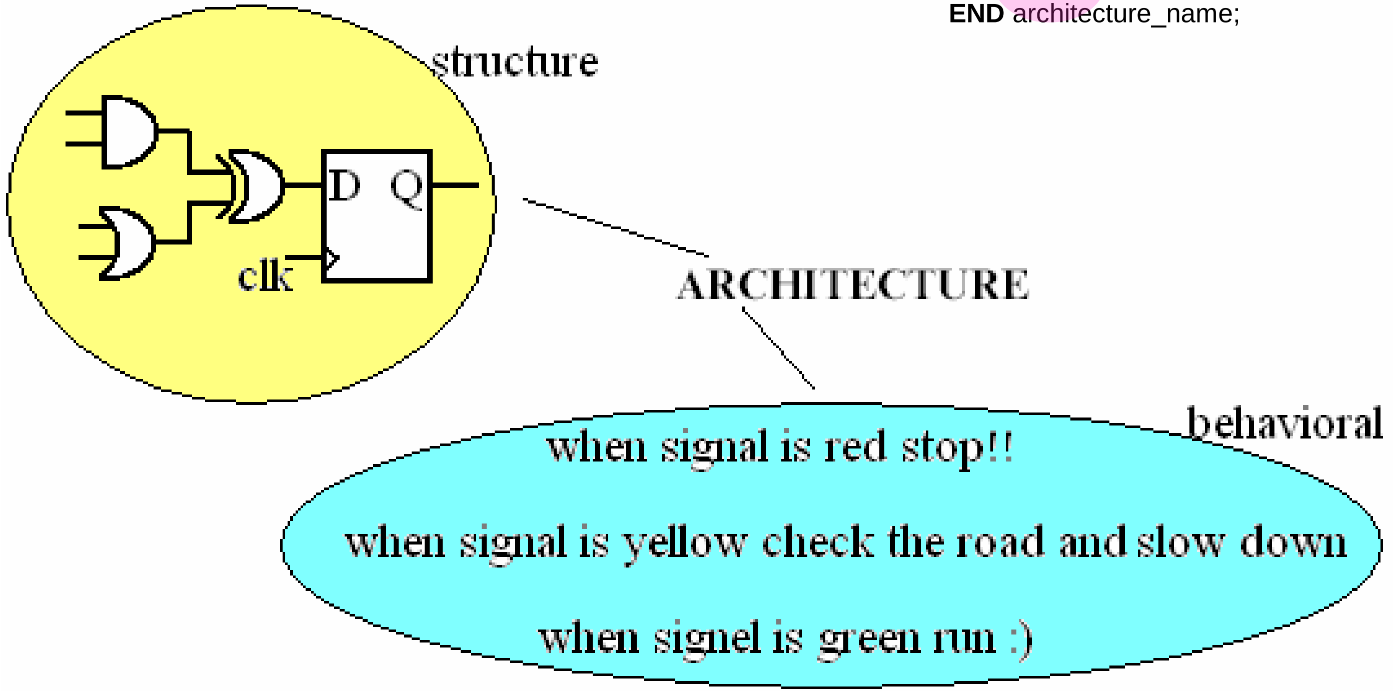
entity name

```
ARCHITECTURE architecture_name OF entity_name IS
[declarations]
BEGIN
      (code)
END architecture_name;
```

Semicolon

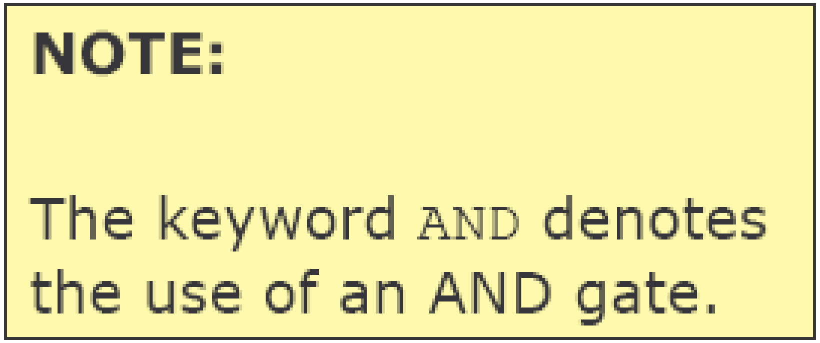
Architecture language

```
ARCHITECTURE architecture_name OF entity_name IS
[declarations]
BEGIN
      (code)
END architecture_name;
```


structure
D
Q
clk
ARCHITECTURE
behavioral
when signal is red stop!!
when signal is yellow check the road and slow down
when signel is green run :)

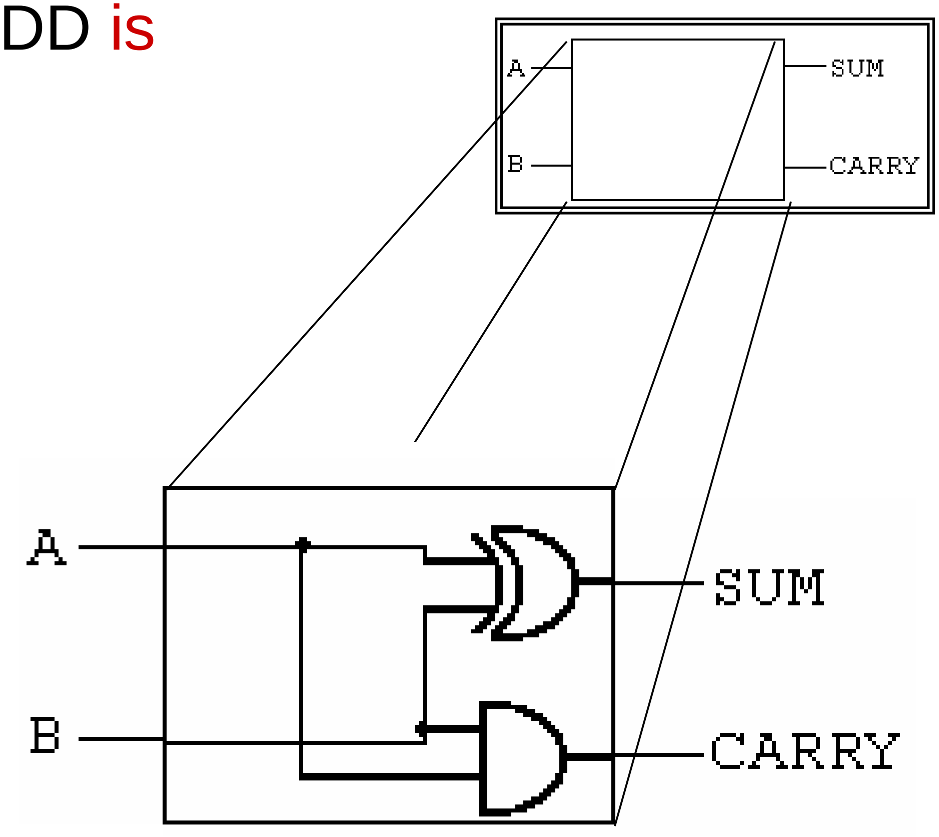
Structural mode

```
architecture struct of  HALFADD is
begin
        SUM     <= (A XOR B);
      CARRY   <=  A AND B   ;
end  struct ;
```



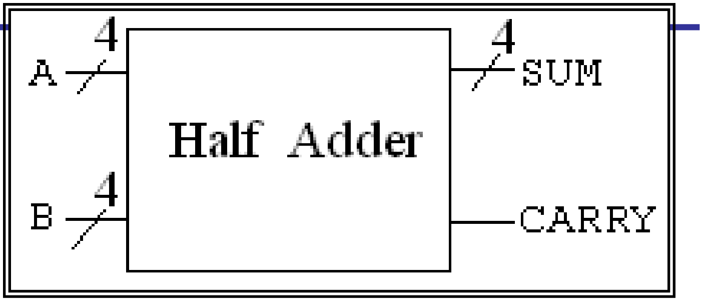
NOTE:

The keyword AND denotes the use of an AND gate.



Exer page 7

Exercise



```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

entity HALFADD is
port (             A,B : in      std_logic_vector (3 downto 0) ;
                   SUM : out     std_logic_vector (3 downto 0) ;
              CARRY  : out     std_logic );
end HALFADD;

architecture struct of  HALFADD is
begin
        SUM     <= (A XOR B);
    CARRY   <=     A AND B   ;   -- is it correct??
end  struct ;
```

END