1- Introduction

2- Basic Structures of VHDL

3- Combinational Circuits

4- Sequential Circuits

5- Memory

6- Writing Testbenches

7- Synthesis Issues

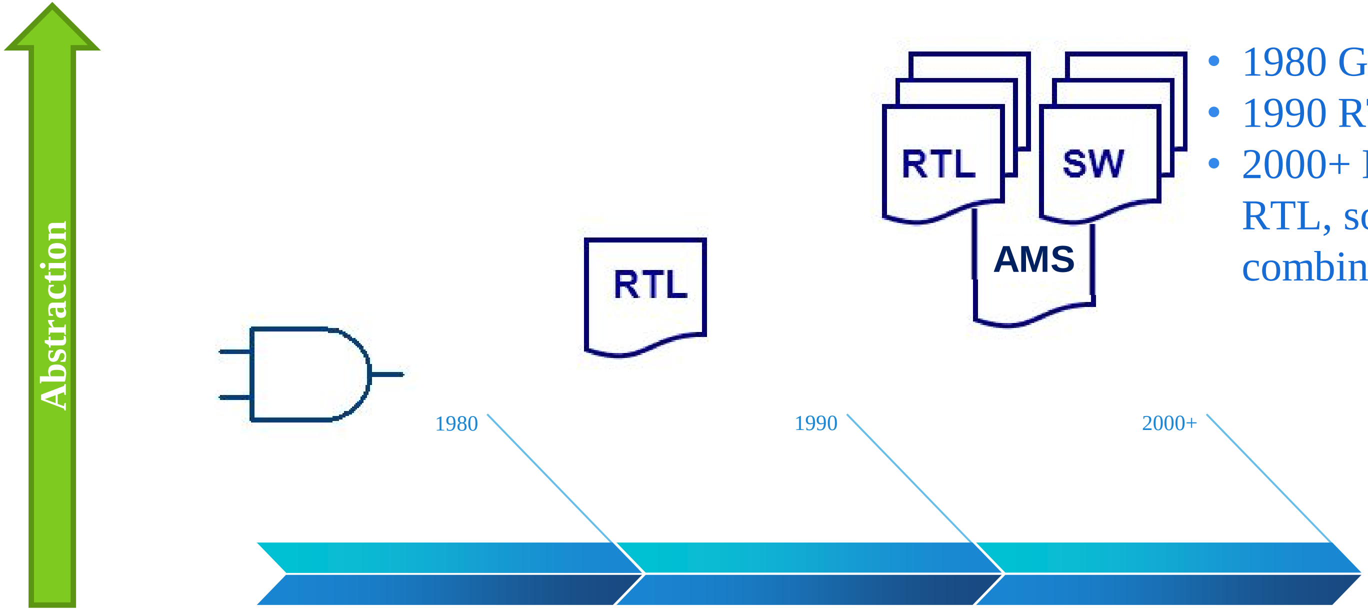
8- RTL Cores

1- Introduction

- Abstraction Levels
- HDL/RTL Design Flow
- Hardware Description Language
- What is a testbench?
- What is a waveform?
- Simulation Environment
- Synthesis Concepts

Abstraction Levels

- 1980 Gate Level
- 1990 RTL
- 2000+ Multiple Interfaced RTL, software, analog combined

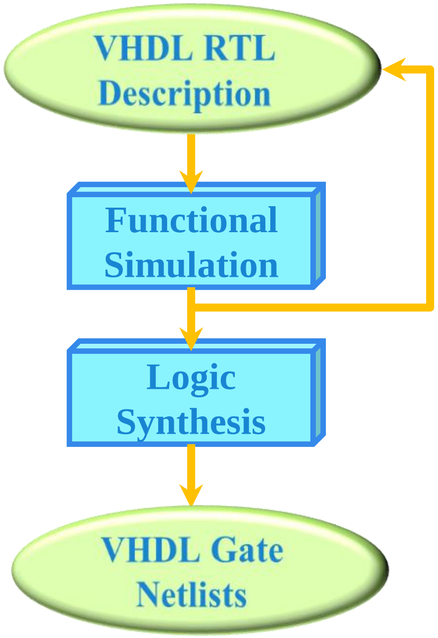


Gate Level

Register Transfer Level (HDL)

System Level

HDL/RTL Design Flow



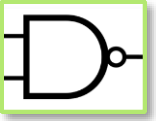
Hardware description language (HDL) design is based on the creation and use of textural based descriptions of a digital logic circuit or system.

- Start with RTL description
- Perform functional simulation until satisfactory results
- Perform logic synthesis once VHDL description is finalized
- Synthesis produces some form of a netlist

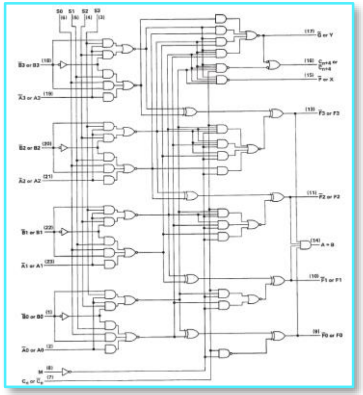
Hardware Description Language

- A picture is worth a thousand words.

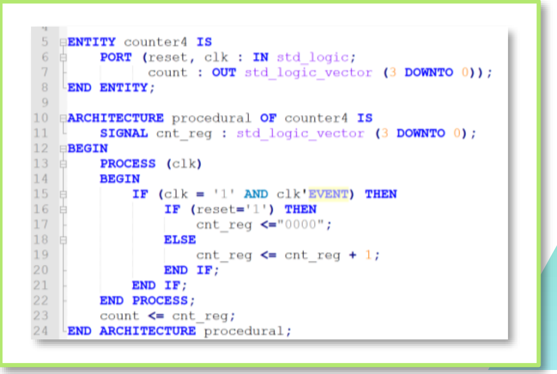
NAND è un costrutto con due input, uno output. L'output diventa ...



- A hardware description is worth a thousand pictures.



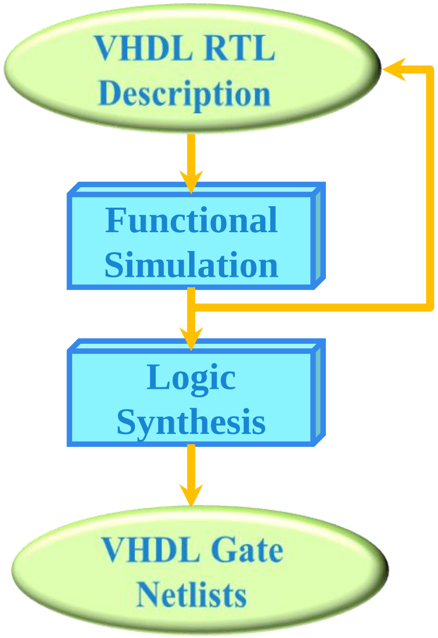
VHDL Description Language



```
ENTITY counter4 IS
    PORT (reset, clk : IN std_logic;
            count : OUT std_logic_vector (3 DOWNTO 0));
END ENTITY;

ARCHITECTURE procedural OF counter4 IS
    SIGNAL cnt_reg : std_logic_vector (3 DOWNTO 0);
BEGIN
    PROCESS (clk)
    BEGIN
        IF (clk = '1' AND clk'EVENT) THEN
            IF (reset='1') THEN
                cnt_reg <="0000";
            ELSE
                cnt_reg <= cnt_reg + 1;
            END IF;
        END IF;
    END PROCESS;
    count <= cnt_reg;
END ARCHITECTURE procedural;
```

HDL/RTL Design Flow



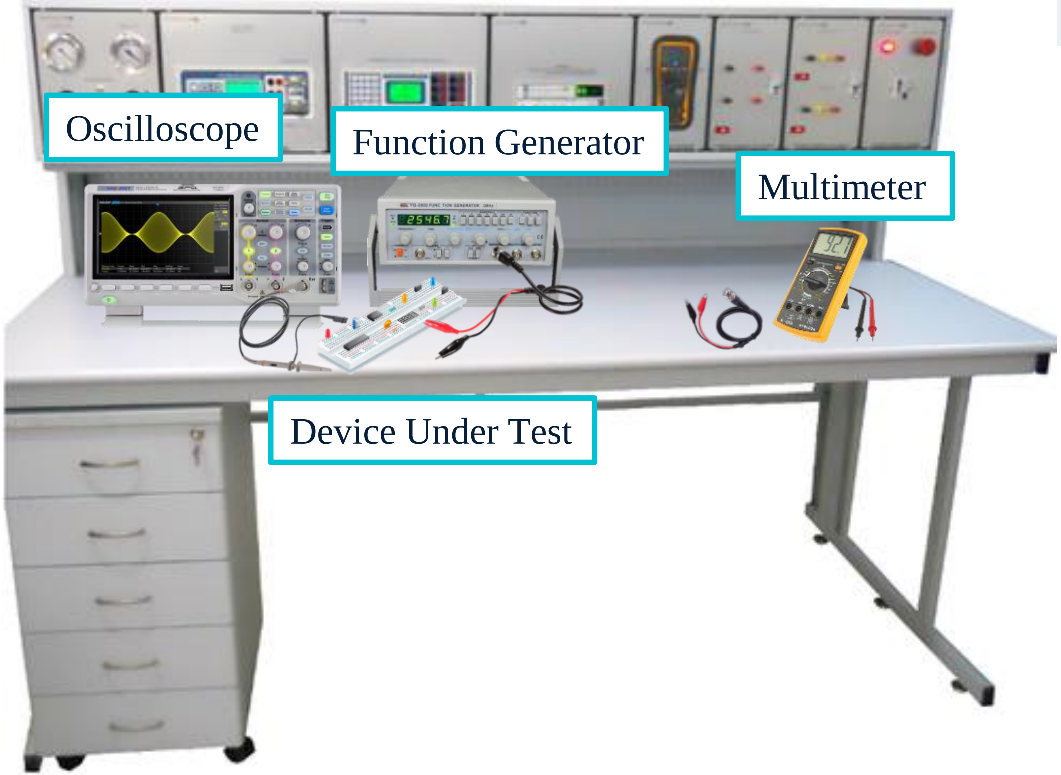
Hardware description language (HDL) design is based on the creation and use of textual based descriptions of a digital logic circuit or system.

A functional simulation simulates the hardware description to verify its logical correctness.

- Functional simulation requires a testbench

1- Introduction

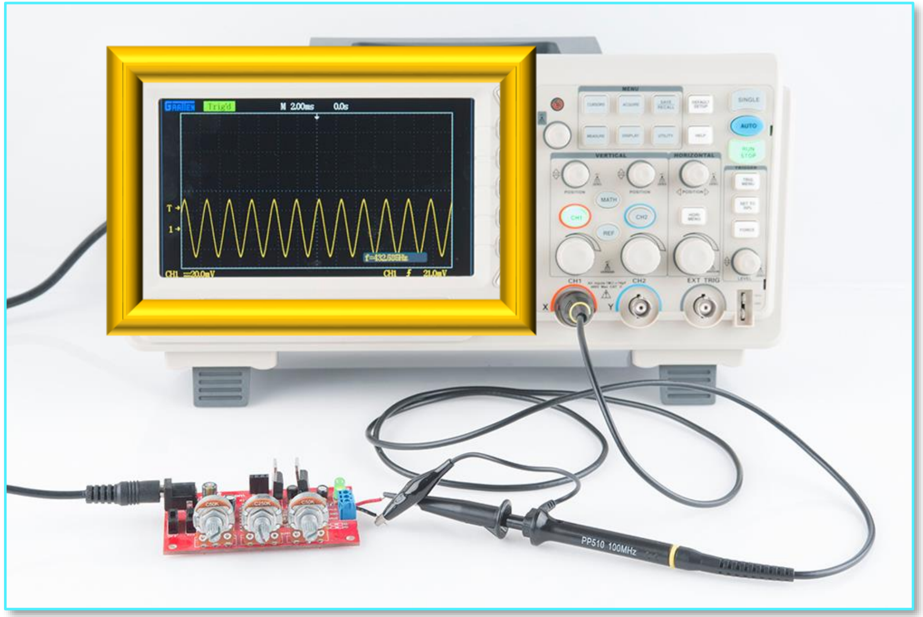
What is a Testbench?



- A testbench contains equipment for input generation
- It includes equipment for response observation and analysis
- Usually output is a waveform

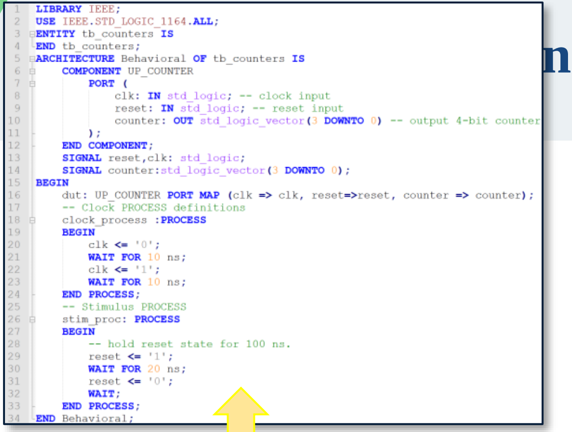
1- Introduction

What is a Waveform?

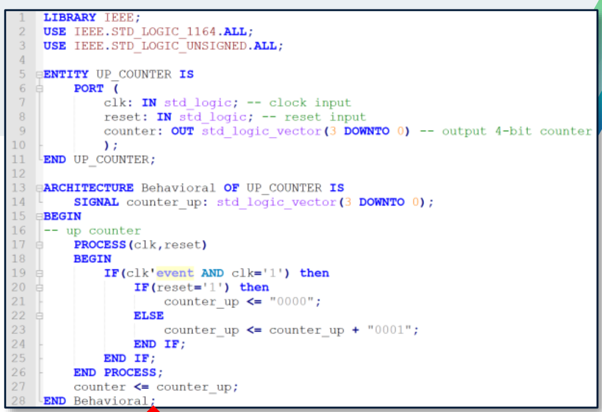


In electronics, the **waveform** of a signal is the shape of its graph as a function of time, independent of its time and magnitude scales and of any displacement in time.

nt



```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY tb_counters IS
END tb_counters;
ARCHITECTURE Behavioral OF tb_counters IS
    COMPONENT UP_COUNTER
        PORT (
            clk: IN std_logic; -- clock input
            reset: IN std_logic; -- reset input
            counter: OUT std_logic_vector(3 DOWNTO 0) -- output 4-bit counter
        );
    END COMPONENT;
    SIGNAL reset,clk: std_logic;
    SIGNAL counter:std_logic_vector(3 DOWNTO 0);
BEGIN
    dut: UP_COUNTER PORT MAP (clk => clk, reset=>reset, counter => counter);
    -- Clock PROCESS definitions
    clock_process :PROCESS
    BEGIN
        clk <= '0';
        WAIT FOR 10 ns;
        clk <= '1';
        WAIT FOR 10 ns;
    END PROCESS;
    -- Stimulus PROCESS
    stim_proc: PROCESS
    BEGIN
        -- hold reset state for 100 ns.
        reset <= '1';
        WAIT FOR 20 ns;
        reset <= '0';
        WAIT;
    END PROCESS;
END Behavioral;
```

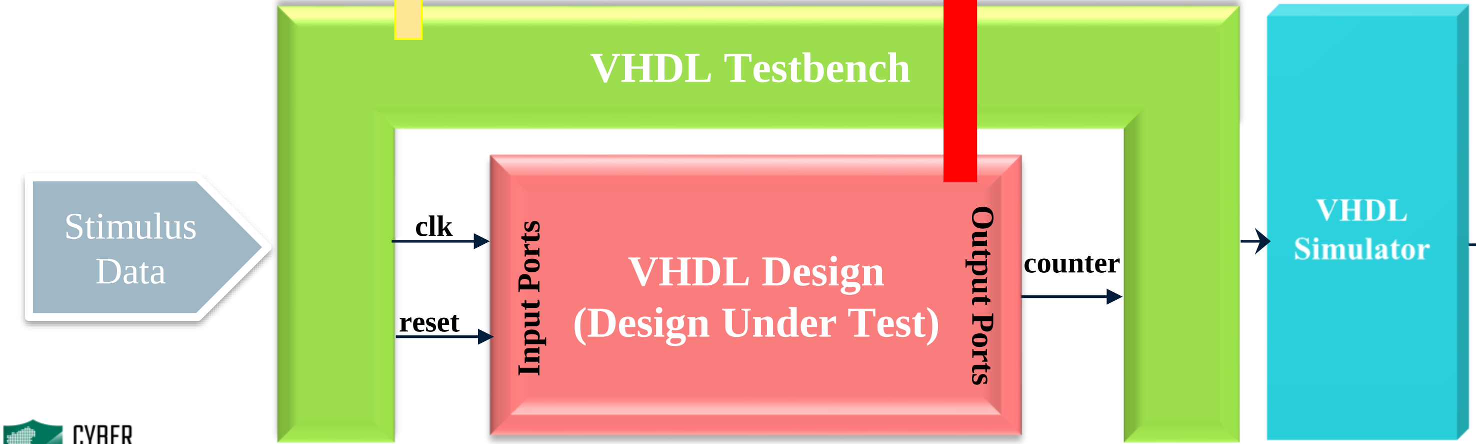


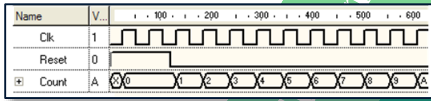
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY UP_COUNTER IS
    PORT (
        clk: IN std_logic; -- clock input
        reset: IN std_logic; -- reset input
        counter: OUT std_logic_vector(3 DOWNTO 0) -- output 4-bit counter
        );
END UP_COUNTER;

ARCHITECTURE Behavioral OF UP_COUNTER IS
    SIGNAL counter_up: std_logic_vector(3 DOWNTO 0);
BEGIN
-- up counter
    PROCESS(clk,reset)
    BEGIN
        IF(clk'event AND clk='1') then
            IF(reset='1') then
                counter_up <= "0000";
            ELSE
                counter_up <= counter_up + "0001";
            END IF;
        END IF;
    END PROCESS;
    counter <= counter_up;
END Behavioral;
```

- Not a physical testbench
- This is a virtual testbench
- A virtual testbench performs simulation
- A physical testbench performs testing



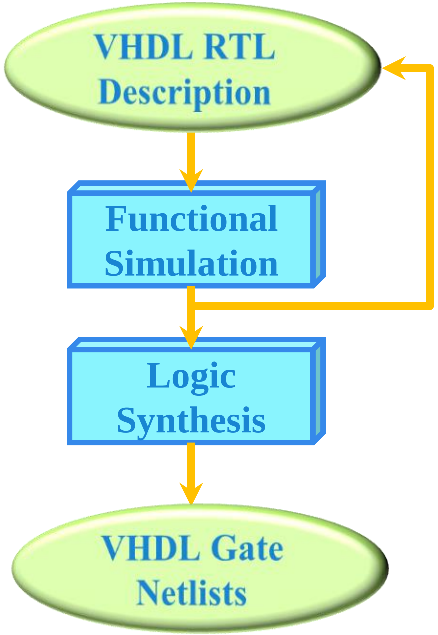


Output Results

Simulation

- Simulation for design validation, done before a design is synthesized
- Also referred to as RT level, or Pre-synthesis Simulation
- Simulation at RT level is accurate to the clock level
- The advantage: its speed compared with simulations at the gate or transistor levels

HDL/RTL Design Flow



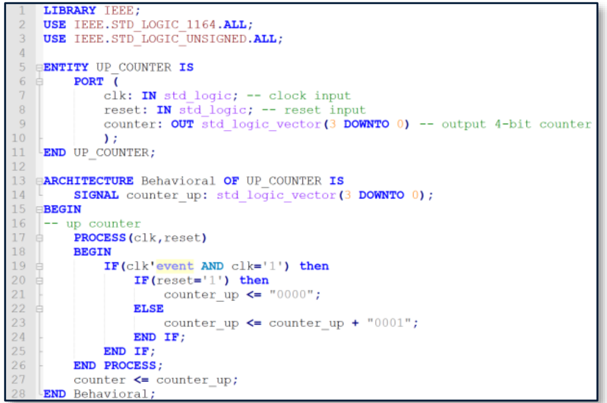
Hardware description language (HDL) design is based on the creation and use of textural based descriptions of a digital logic circuit or system.

A functional simulation simulates the hardware description to verify its logical correctness.

Synthesis is the process of taking some higher level description down to a lower level description.

1- Introduction

Synthesis Concepts



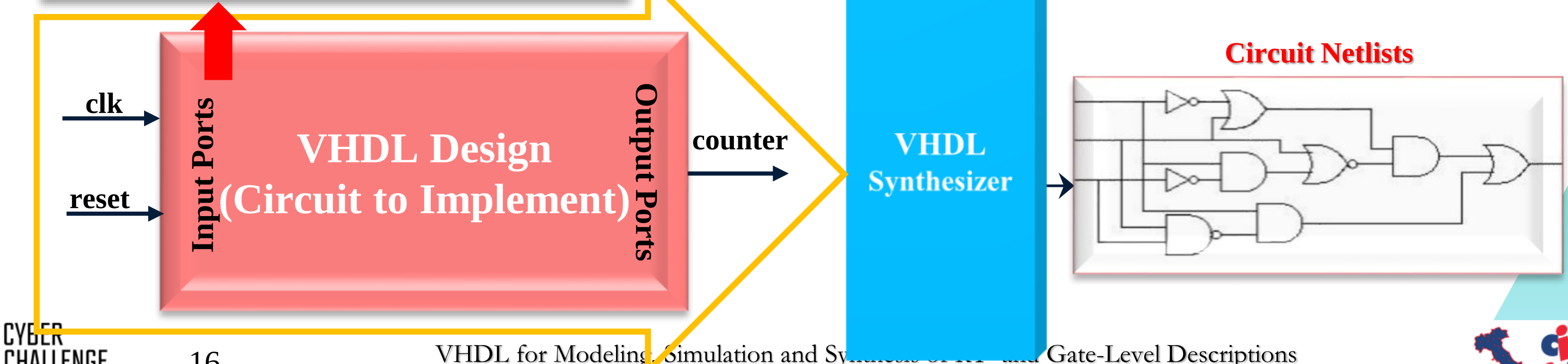
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY UP_COUNTER IS
    PORT (
        clk: IN std_logic; -- clock input
        reset: IN std_logic; -- reset input
        counter: OUT std_logic_vector(3 DOWNTO 0) -- output 4-bit counter
        );
END UP_COUNTER;

ARCHITECTURE Behavioral OF UP_COUNTER IS
    SIGNAL counter_up: std_logic_vector(3 DOWNTO 0);
BEGIN
-- up counter
    PROCESS(clk,reset)
    BEGIN
        IF(clk'event AND clk='1') then
            IF(reset='1') then
                counter_up <= "0000";
            ELSE
                counter_up <= counter_up + "0001";
            END IF;
        END IF;
    END PROCESS;
    counter <= counter_up;
END Behavioral;
```

VHDL Description

- A synthesizable description is fed to the VHDL synthesis tool
- The synthesis tool produces a netlist in one form or the other



1- Introduction