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)

Usage

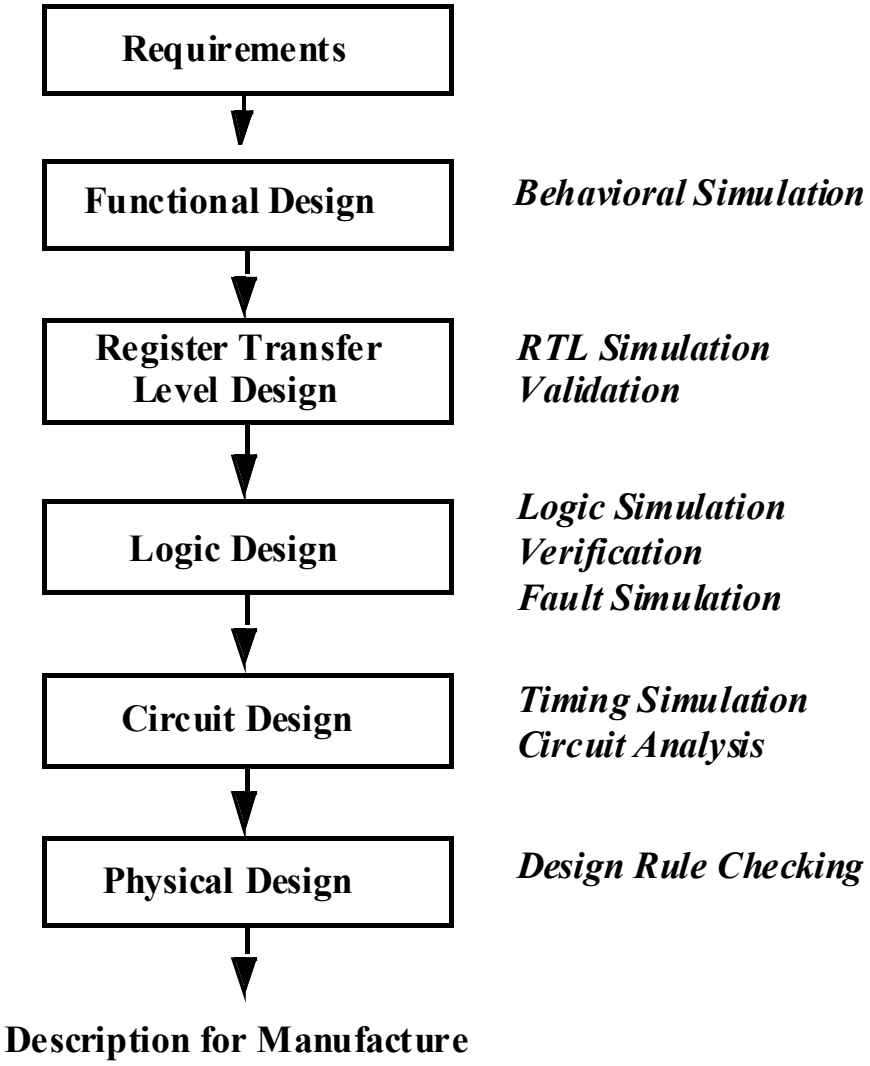
- Descriptions can be at different levels of abstraction
  - Switch level: model switching behavior of transistors
  - Register transfer level: model combinational and sequential logic components
  - Instruction set architecture level: functional behavior of a microprocessor

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

Why do we Describe Systems?

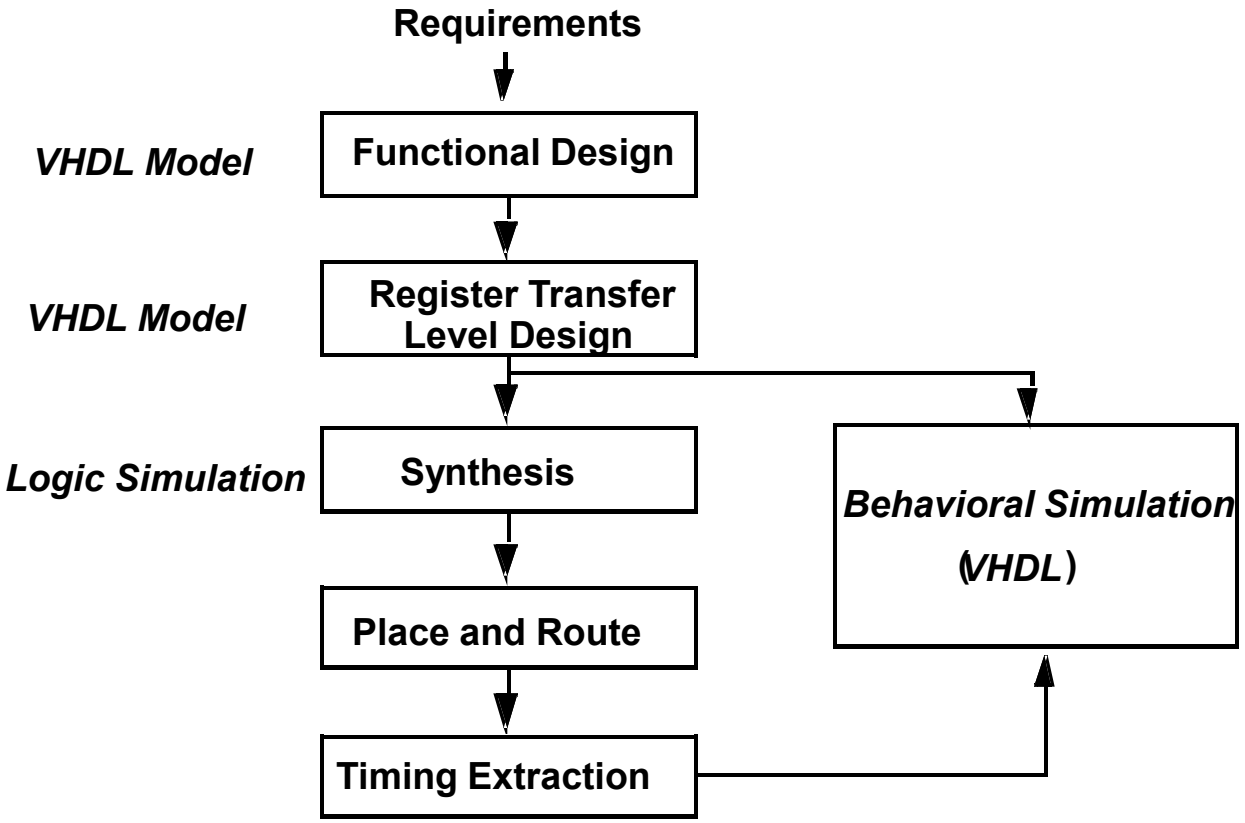
- Design Specification
  - unambiguous definition of components and interfaces in a large design
- Design Simulation
  - verify system/subsystem/chip performance prior to design implementation
- Design Synthesis
  - automated generation of a hardware design

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

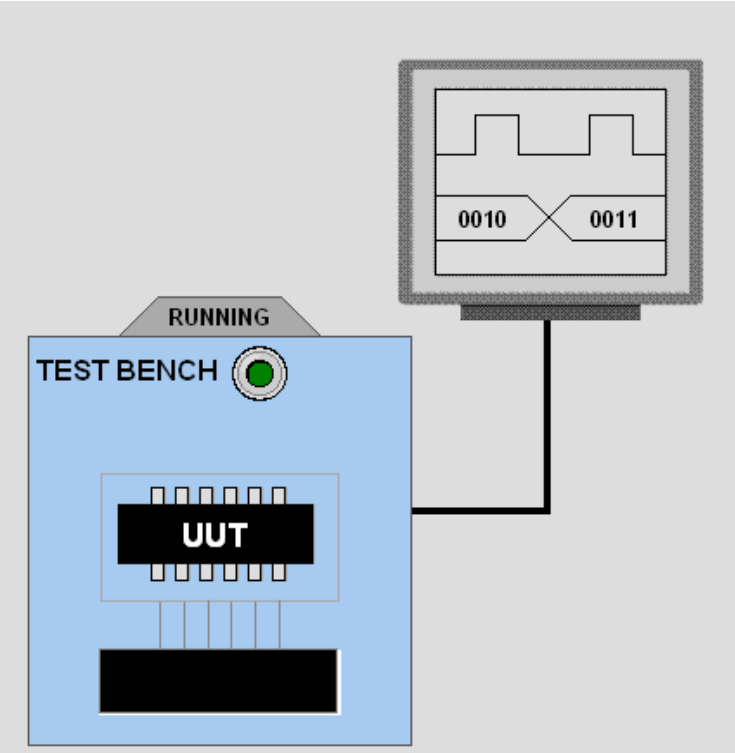
Basic VHDL Concepts

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

Test Benches

- Testing a design by simulation
- Use a *test bench* model
  - an architecture body that includes an instance of the design under test
  - applies sequences of test values to inputs
  - monitors values on output signals
    - either using simulator
    - or with a process that verifies correct operation

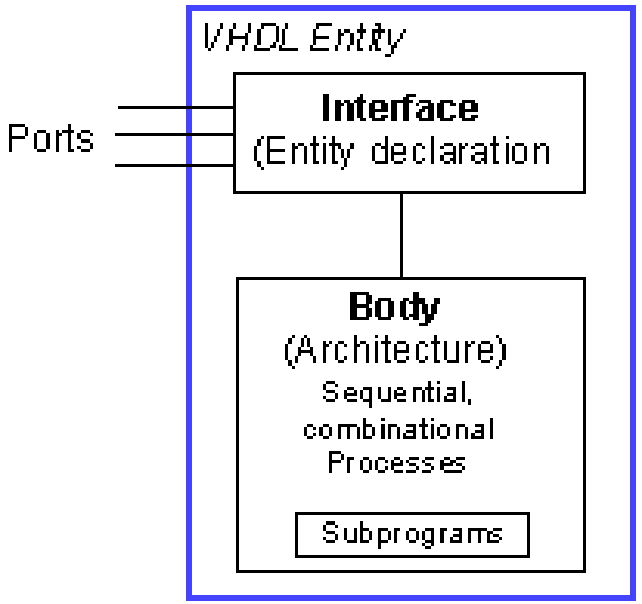
Test Bench




```
module main;
reg  a, b, c;
wire sum, carry;

fulladder add(a,b,c,sum,carry);
initial
begin
   a = 0; b = 0; c = 0;
   #5
   a = 0; b = 1; c = 0;
   #5
   a = 1; b = 0; c = 1;
   #5
   a = 1; b = 1; c = 1;
   #5
end
endmodule
```

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

…..

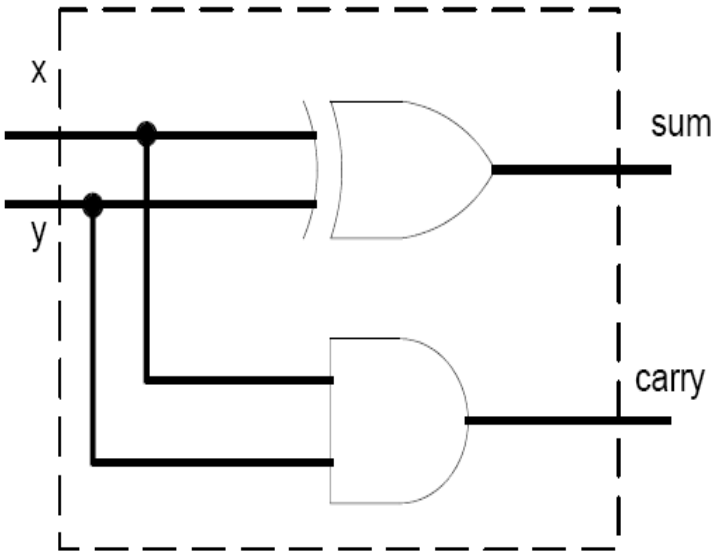
…..

begin

-- Statements

end architecture_name;

Half Adder



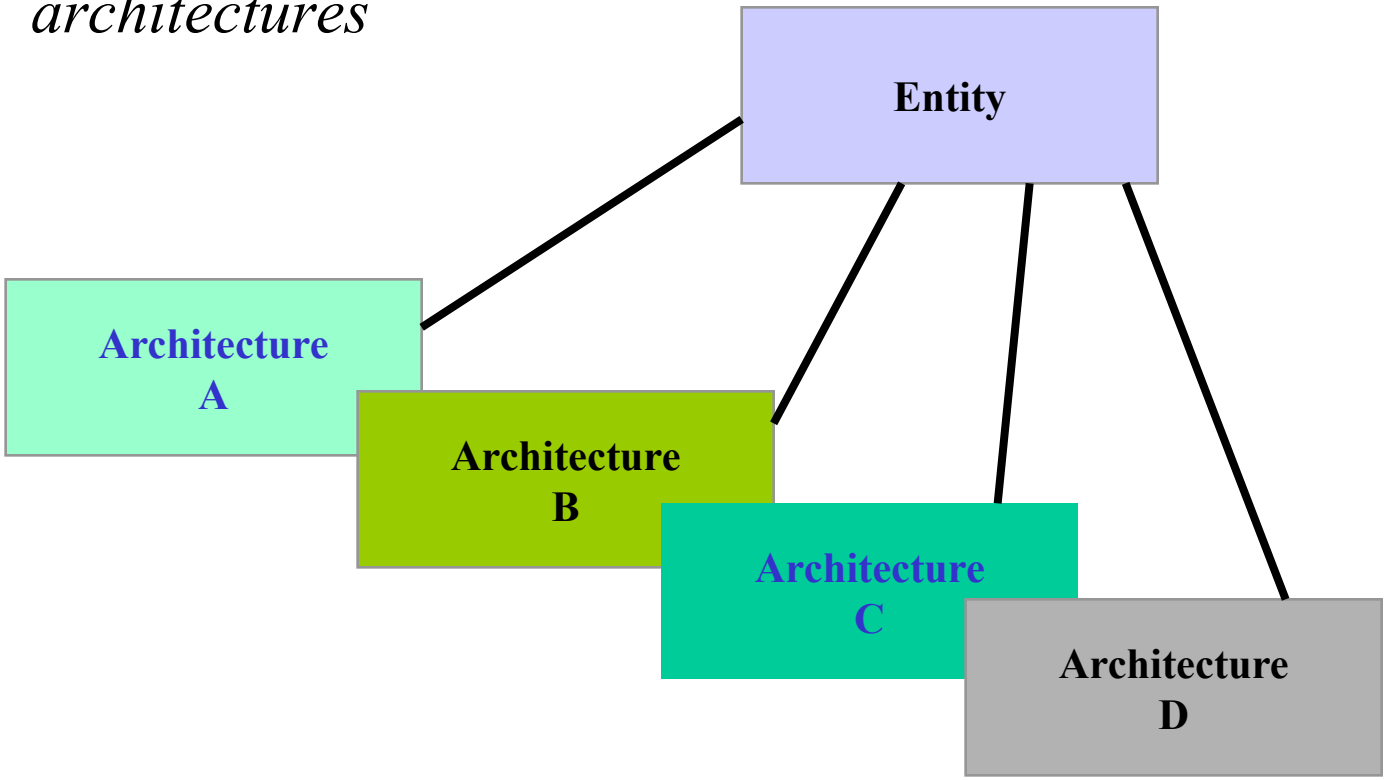

```
library ieee;
use  ieee.std_logic_1164.all;
entity half_adder is
port(
            x,y: in std_logic;
            sum, carry: out std_logic);
end half_adder;

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;
```

One Entity Many Descriptions

- A system (an *entity*) can be specified with different *architectures*



Text-books

1. Digital System Design using VHDL by C.H. Roth.
2. Circuit Design with VHDL by Volnei A. Pedroni;