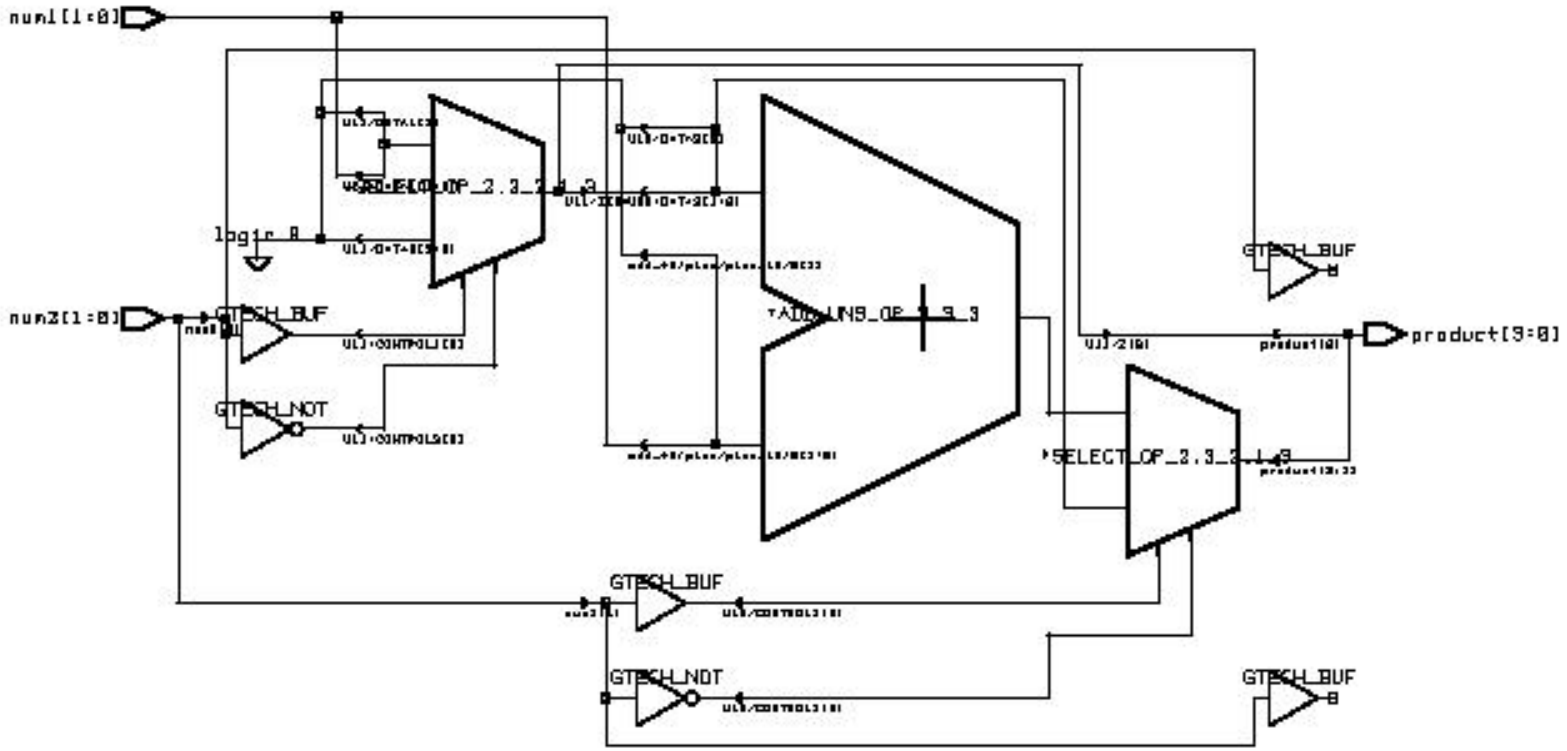
Hardware Description Languages

A Brief Introduction

Schematic Capture

- Schematic capture is the process of creating a graphical representation of the desired electronic circuit
  - Focus on design implementation
  - Less attention to overall function of the design
  - Less effective for large designs with complex operations and hierarchies

Schematic Capture Example



HDLs

- HDLs facilitate design specification through language semantics
- Focus is on the high level system functionality
- Logical partitioning of design components may be represented in HDL semantics
- Designers think about how the design will work rather than its low level implementation
- An electronic circuit may be derived from the HDL model description through design synthesis

HDL Example

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity mux is
    port(rst, sLine: in std_logic;
         load, result: in std_logic_vector( 3 downto 0 );
         output: out std_logic_vector( 3 downto 0 )
    );
end mux;

architecture mux_arc of mux is
begin
    process( rst, sLine, load, result )
    begin
            if( rst = '1' ) then
                output <= "0000";-- do nothing
            elsif sLine = '0' then
                output <= load;-- load inputs
            else
                output <= result;-- load results
            end if;
    end process;
end mux_arc;
```

Popular HDLs

- VHDL
  - VHSIC Hardware Description Language
  - VHSIC – Very High Speed Integrated Circuit
  - Bears a close resemblance to the Ada programming language
- Verilog
  - Bears a close resemblance to the C programming language

Focus on VHDL

- For the purposes of discussion, we will focus on designing logic using VHDL

Example VHDL Model

```
entity half_adder is
port(
   x: in Bit ;
   y: in Bit ;
   Sum: out Bit ;
   Carry: out Bit) ;
end half_adder;
```

```
architecture logic of
half_adder is
begin
 P1: process(x, y)
  begin
     Sum <= x xor y after 5 ns;
     Carry <= x and y after 5
ns;
 end process;
end logic;
```

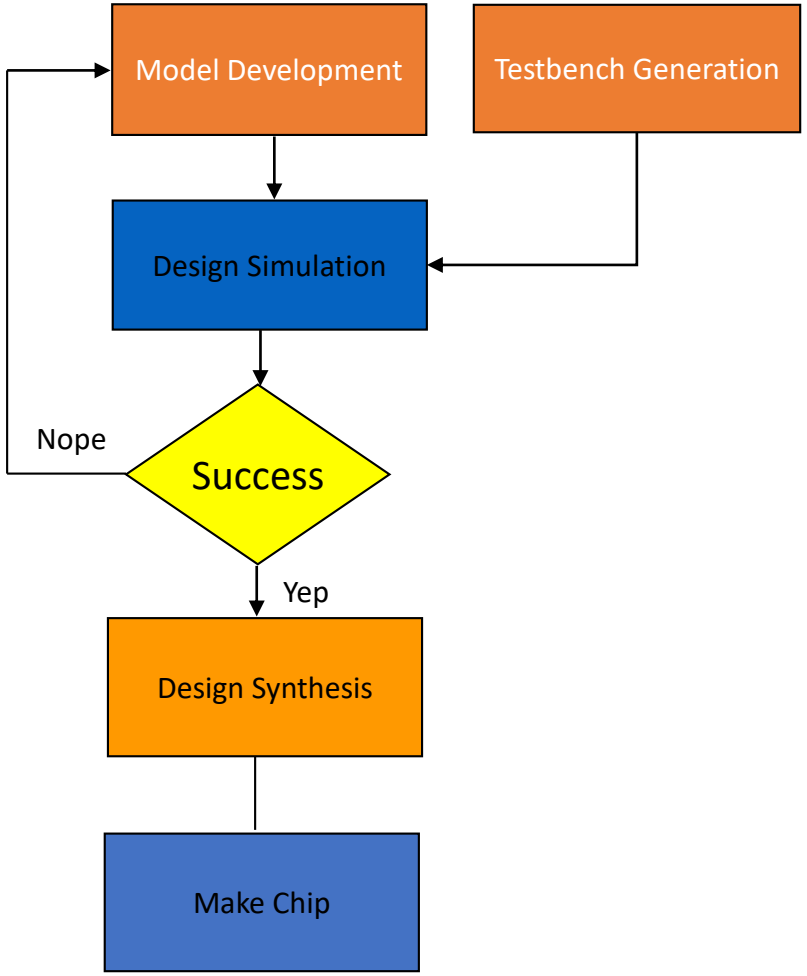
Behavioral Modeling

```
entity ADDER is
generic(n: natural :=2);
port(A:    in std_logic_vector(n-1
downto 0);
     B:    in std_logic_vector(n-1
downto 0);
     carry:           out std_logic;
     sum:  out std_logic_vector(n-1
downto 0)
);
end ADDER;


architecture behave of ADDER is
signal result: std_logic_vector(n
downto 0);
begin

    result <= ('0' & A)+('0' & B);
    sum <= result(n-1 downto 0);
    carry <= result(n);
end behave;
```

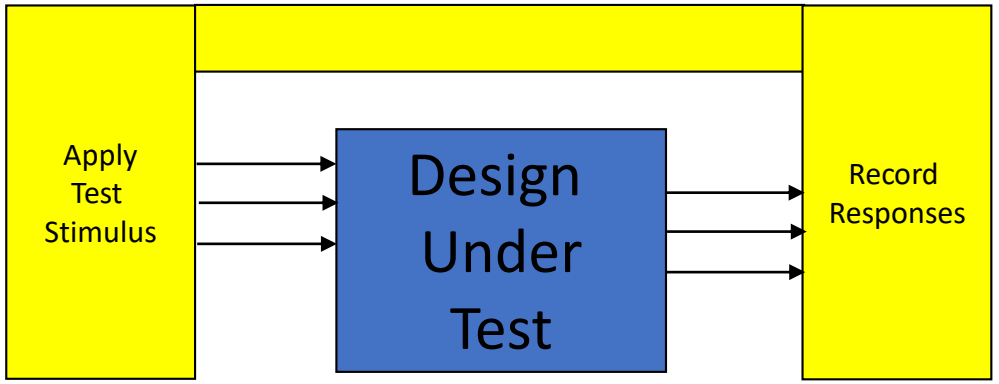
This is a 2 bit adder design because n is set to 2.

The adder's behavior is specified by the assignment statements in the architecture.

Simplified VHDL Design Flow

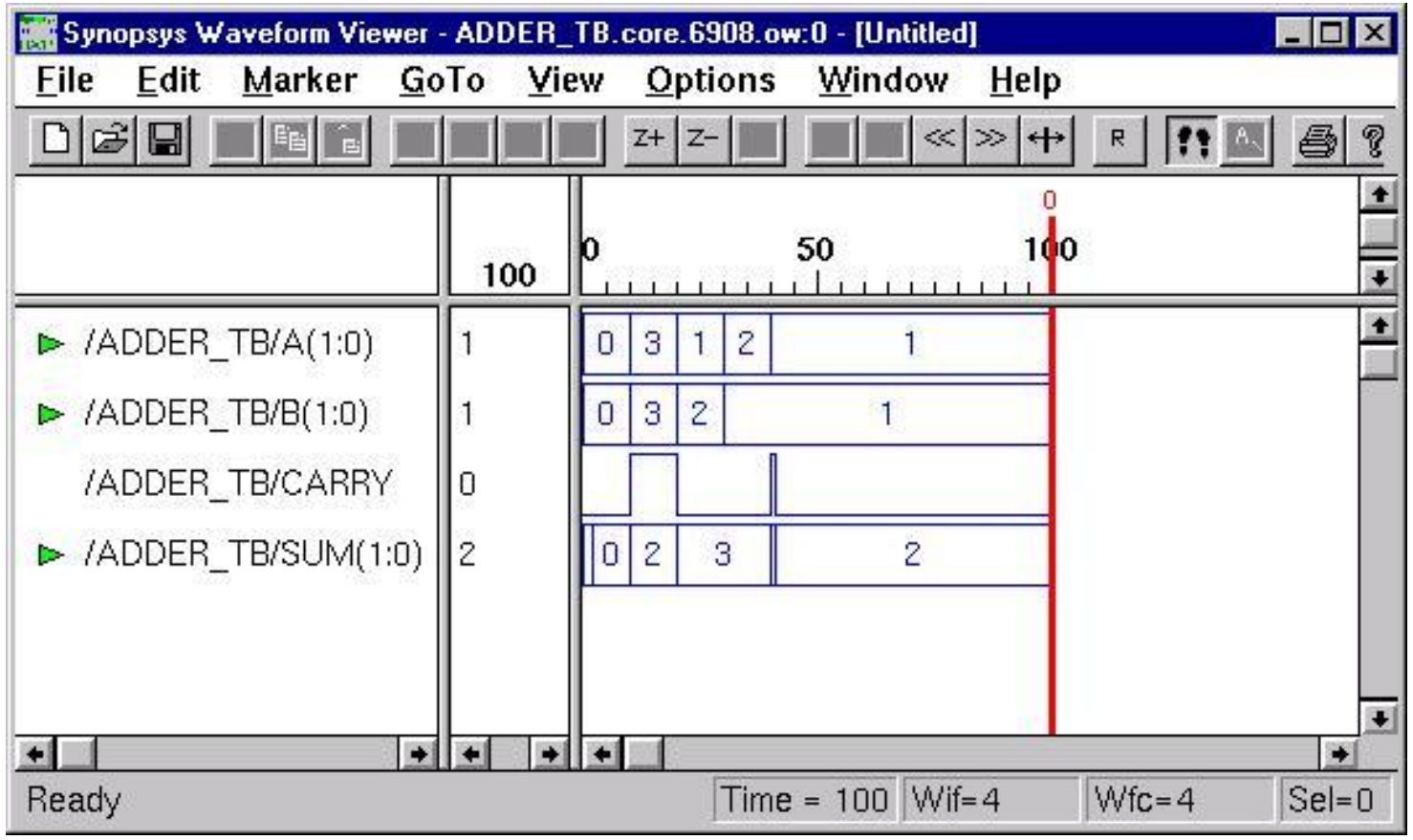


Design Simulation

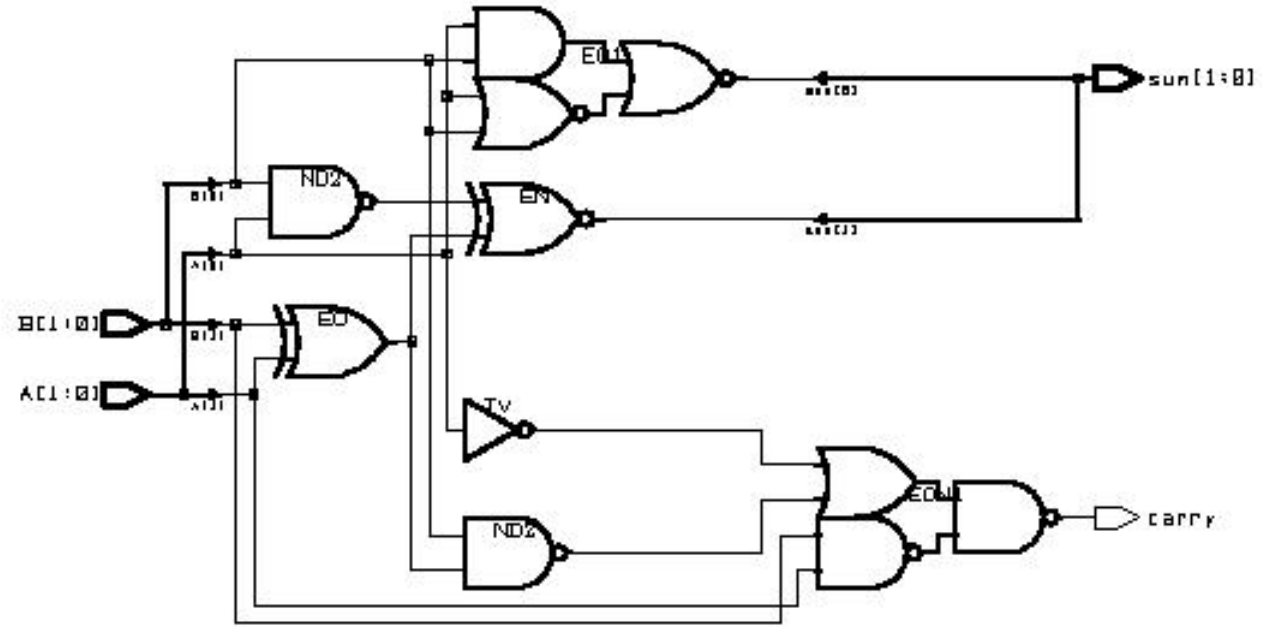


Simulation is the process of testing a design to determine if it functions in the manner for which it was designed.

Analyzing Simulation Data



Design Synthesis



Synthesis is the process of generating a schematic representation from a textual description of the logic function written in the HDL.

Summary

- Register Transfer Languages are important in helping to model a logic function without having to construct a physical implementation
- RTLs help to focus the attention of the designer on functional rather than implementation details
- Synthesis tools may be used to create the final schematic of the design
- VHDL and Verilog are the most popular of the HDLs in the logic design community