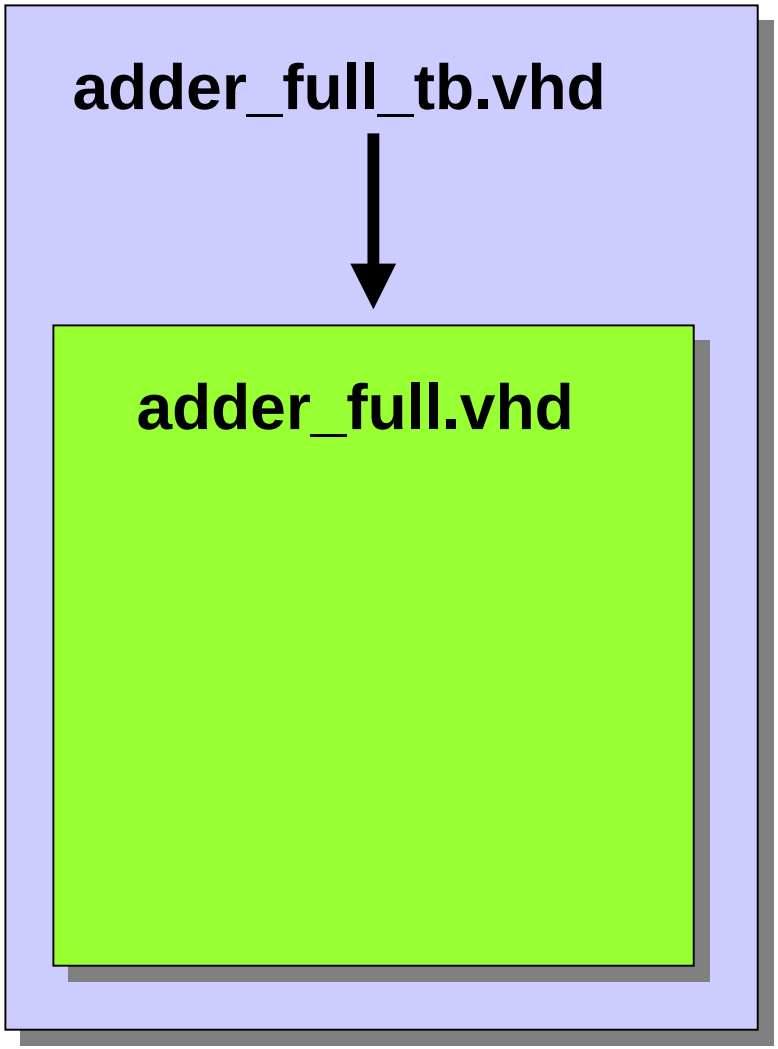
EECS 318 CAD
Computer Aided Design

LECTURE Simulator 1:
Synopsys Simulator

Instructor: Francis G. Wolff
wolff@eecs.cwru.edu
Case Western Reserve University

This presentation uses powerpoint animation: please viewshow

The adder file test bench hierarchy



adder_full.vhd: full 1-bit adder

```
LIBRARY IEEE;
use IEEE.std_logic_1164.all;
```

```
ENTITY adder_full IS
    PORT (x, y, Cin: IN std_logic; Sum, Cout: OUT std_logic
); END;
```

```
ARCHITECTURE adder_full_arch OF  adder_full  IS

BEGIN
        Sum   <= ( x XOR y ) XOR Cin;
        Cout  <= ( x AND y ) OR (Cin AND (x OR y));
END;
```

```
CONFIGURATION  adder_full_cfg OF  adder_full IS
        FOR adder_full_arch
        END FOR;
END CONFIGURATION;
```

VHDL analyzer: vhdlan

Unix command: vhdlan -NOEVENT <filename.vhd>

- **Must be done to every vhdl file in the design**

For example:

> vhdlan -NOEVENT adder_full.vhd

Synopsys 1076 VHDL Analyzer Version 2000.06--May 24, 2000

VHDL Simulator: vhdlsim

Unix command: vhdlsim <vhdl_configuration_name>

- **Starts the text based vhdl simulator**

For example:

Simulator command line prompt #

> vhdlsim adder_full_cfg

Synopsys 1076 VHDL Simulator Version 2000.06-- May 24, 2000

Copyright (c) 1990-2000 by Synopsys, Inc.
ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys, Inc. and may be used and disclosed only as authorized in a license agreement controlling such use and disclosure.

#

VHDL Simulator list components: ls

vhdlsim list command: ls [-type] [-value]

- **lists the vhdl component types and data values**

After reading in the adder_full.vhd design, a list will show

```
# ls
ADDER_FULL      STANDARD        ATTRIBUTES
STD_LOGIC_1164  _KERNEL


# ls -type
ADDER_FULL          COMPONENT INSTANTIATION STATEMENT
STANDARD            PACKAGE
ATTRIBUTES          PACKAGE
STD_LOGIC_1164      PACKAGE
_KERNEL             PROCESS STATEMENT
#
```

VHDL Simulator change directory: cd and pwd

vhdlsim cd command: cd <component_path>
cd ..
pwd

- cd - change design hierarchy (cd .. go up a level)
- pwd - display present working directory

```
# cd ADDER_FULL
# pwd
/ADDER_FULL

# ls -type
X               IN PORT  type = STD_LOGIC
Y               IN PORT  type = STD_LOGIC
CIN             IN PORT  type = STD_LOGIC
SUM             OUT PORT type = STD_LOGIC
COUT            OUT PORT  type = STD_LOGIC
_P0             PROCESS STATEMENT
```

Alternately, using full paths
 # ls -type /ADDER_FULL

VHDL Simulator assign signal: assign

vhdlsim command: assign [-after <time>] <value> <signal>

- **assign a value to a signal**

```
# ls -value
X                'U'
Y                'U'
CIN              'U'
SUM              'U'
COUT             'U'
# assign '1' X
# ls -value
X                '1'
Y                'U'
CIN              'U'
SUM              'U'
COUT             'U'
```

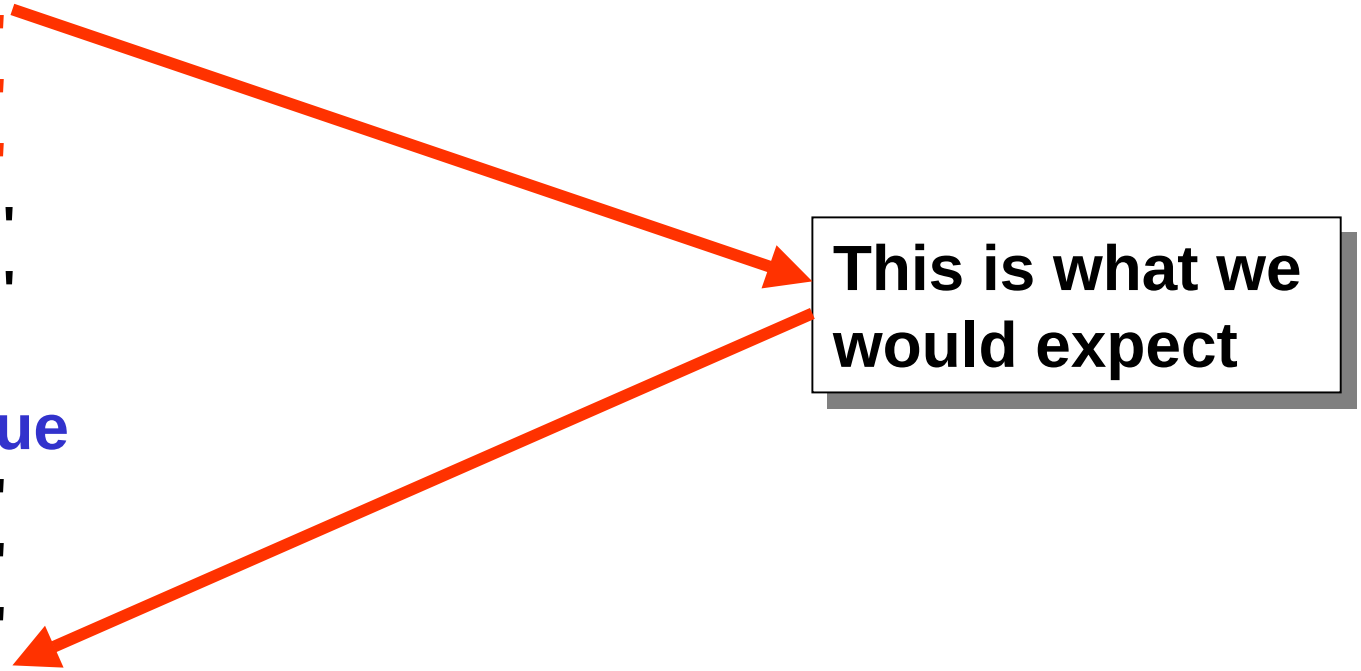
Alternately, using full paths
 # assign '1' /ADDER_FULL/X

VHDL Simulator run simulation: run

vhdlsim command: run [<time nanoseconds>]

- **Use Control-C to cancel a simulation**

```
# assign '1' X
# assign '1' Y
# assign '0' Cin
# ls -value
X       '1'
Y       '1'
CIN     '0'
SUM     'U'
COUT    'U'
# run
# ls -value
X       '1'
Y       '1'
CIN     '0'
SUM     '0'
COUT    '1'
```



VHDL Simulator include

vhdlsim command: include [-e] <filename.vhdlsim>

- **Reads and executes vhdlsim commands from a file**
- **-e will displays the lines as it reads them in**

For example, the file adder_full.vhdlsim contains:

```
cd ADDER_FULL
assign '1' X
assign '1' Y
assign '0' Cin

ls -value >adder_full.run
run
ls -value >>adder_full.run
exit
```

> overwrite file

>> append to file

VHDL Simulator include using full path names

For example, adder_full.vhdlsim using full path names:

```
assign '1' /ADDER_FULL/X
assign '1' /ADDER_FULL/Y
assign '0' /ADDER_FULL/Cin

ls -value  /ADDER_FULL  >adder_full.run
run
ls -value  /ADDER_FULL  >>adder_full.run
exit
```

VHDL Simulator trace

vhdlsim command: trace <signals>

- **Traces vhdl signals on Synopsys Waveform Viewer**
- **To best view signals a time element must be added**
- **Use View => Full Fit in order to fully view the signals**

For example,

```
cd ADDER_FULL
assign -after 5 '1' X
assign -after 5 '1' Y
assign -after 5 '0' Cin
trace X Y Cin Sum Cout
ls -value
run
ls -value
exit
```

VHDL Simulator: abstime, step, next, status

vhdlsim command: abstime

- display the current absolute simulation time so far

vhdlsim command: step [<n steps>]

- step through each vhdl statement, default n=1

vhdlsim command: next [n steps]

- step through each vhdl statement within current arch

vhdlsim command: status

- show current simulation status

VHDL Simulator: where, environment, restart

vhdlsim command: where

- displays where the process and event stacks

vhdlsim command: environment

- displays the simulator environmental variables

vhdlsim command: restart

- restart the simulation using all previous commands
- Clean restart: restart /dev/null

VHDL Simulator: help

vhdlsim command: help [<simulator_command>]

- **simulator command help: help ls**

 # help step
Subject: STEP
Syntax: STEP [n]

STEP executes the next "n" lines of VHDL source code. If you omit the argument "n", it executes a single line.

STEP enters functions and procedures.

STEP does not count or stop on lines that are monitored by an OFF monitor.

VHDL Simulator: unix shell, exit, quit, slist

vhdlsim command: !<unix command>

- **Execute a unix shell command: !ls**

vhdlsim command: exit

- **exit the simulator**

vhdlsim command: quit

- **quit the simulator**

vhdlsim command: slist [entity name]

- **display the current source or entity name read in**

adder_full_tb.vhd: full adder test bench

```
LIBRARY IEEE;
use IEEE.std_logic.all;
```

```
ENTITY adder_full_tb IS
        PORT (x, y, Cin:        IN  std_logic;
                Sum, Cout:      OUT std_logic);
END;
```

adder_full_tb.vhd: architecture

```
ARCHITECTURE adder_full_tb_arch OF adder_full_tb IS
    COMPONENT adder_full
      PORT (x, y, Cin: IN std_logic; Sum, Cout: OUT std_logic);
    END COMPONENT;
    SIGNAL x, y, Cin: std_logic;
BEGIN
  x <= '0', '1' after 50 ns, '0' after 100 ns;
  y <= '1', '1' after 50 ns, '0' after 100 ns;
  Cin <= '0', '1' after 50 ns;
  UUT_ADDER: adder_full PORT MAP(x, y, Cin, Sum, Cout);
END;
```

```
CONFIGURATION  adder_full_tb_cfg OF  adder_full_tb IS
          FOR adder_full_tb_arch END FOR;
END CONFIGURATION;
```

VHDL Simulator: test bench

```
Unix> vhdlan -NOEVENT adder_full.vhd
Unix> vhdlan -NOEVENT adder_full_tb.vhd
Unix> vhdlsim adder_full_tb_cfg

# ls
ADDER_FULL_TB   STANDARD      ATTRIBUTES
STD_LOGIC_1164  _KERNEL
# cd ADDER_FULL_TB
# ls
SUM             _P0           _P2           ADDER_FULL    Y
COUT            _P1           UUT_ADDER     X             CIN
# ls -type
SUM                 OUT PORT   type = STD_LOGIC
COUT                OUT PORT   type = STD_LOGIC
UUT_ADDER           COMPONENT INSTANTIATION
ADDER_FULL          COMPONENT
X                   SIGNAL     type = STD_LOGIC
```

VHDL Simulator: run 10 ns

```
# ls -value
SUM             'U'
COUT            'U'
X               'U'
Y               'U'
CIN             'U'
# run 10
10 NS
# ls -value
SUM             '1'
COUT            '0'
X                '0'
Y                '1'
CIN             '0'
```

VHDL Simulator: run 60 ns

```
# run 60
70 NS
# ls -value
SUM              '1'
COUT             '1'
X                '1'
Y                '1'
CIN              '1'
# quit
```

VHDL Simulator GUI: vhdldbx

Unix command: vhdldbx <vhdl_configuration_name> &

- **Starts the VHDL GUI version of vhdlsim**
- **Does everything vhdlsim does via menus**

- **Use the trace command to view signals**
  - **First mark the variable with the mouse**
  - **Then traces -> signals**