ES 4: Testing and debugging VHDL

Steven Bell

3 October 2023

Tufts UNIVERSITY

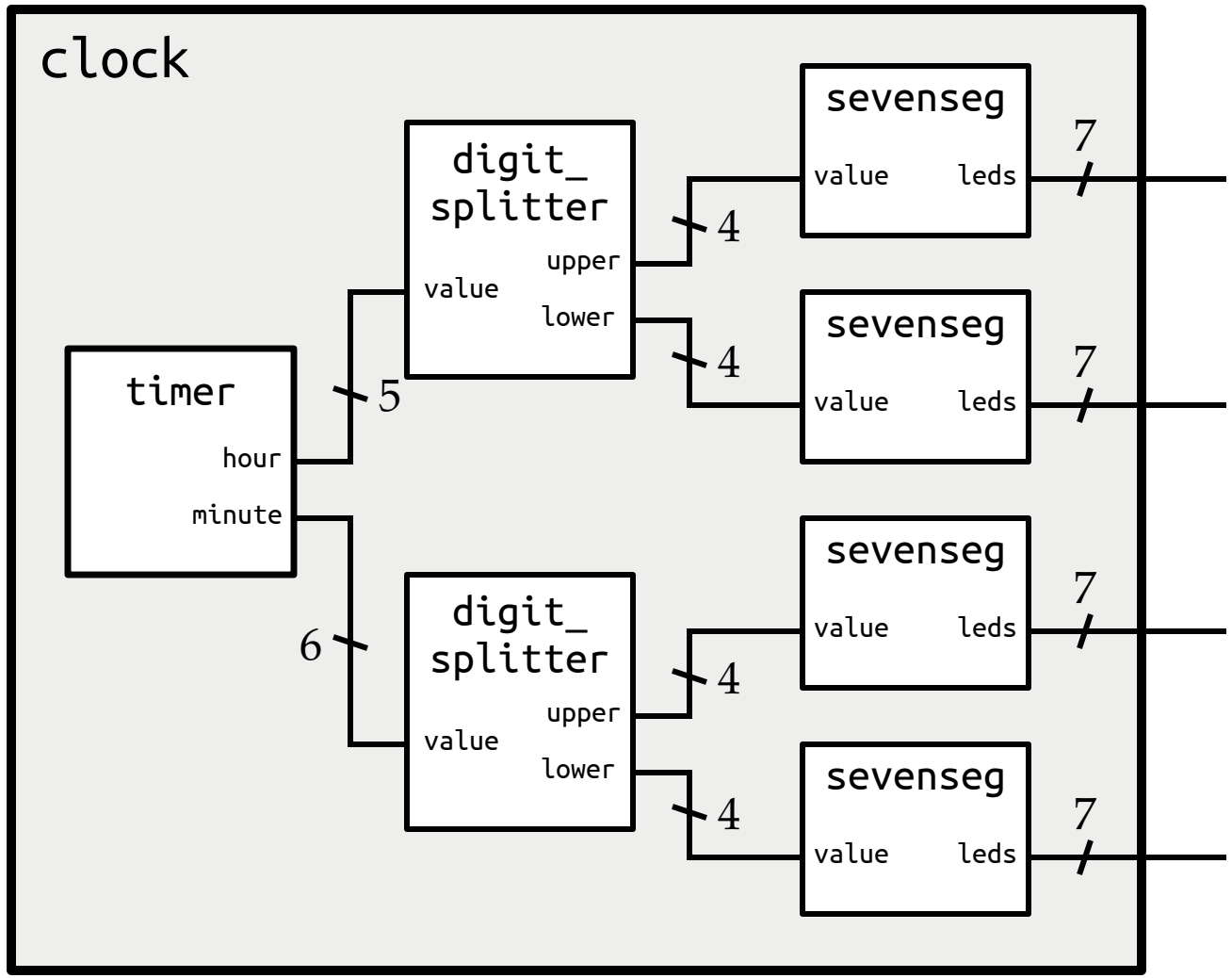
By the end of class today, you should be able to:

- Explain what it means when people say code in a VHDL `process` block executes "sequentially"
- Instantiate a sub-module in VHDL
- Write a testbench for combinational logic


Introduction to Digital Logic
Computer Programming
Verilog
VHDL
Fields of Enlightenment
Memory
Sequential Logic
Process Block Perils
Swamps of Arcane Syntax
State Machines
YOU ARE HERE
Flip-flop Pass
Computer Architecture
Plains of Combinational Logic
Desert of "How things used to be done"

What's the point of structural modeling?

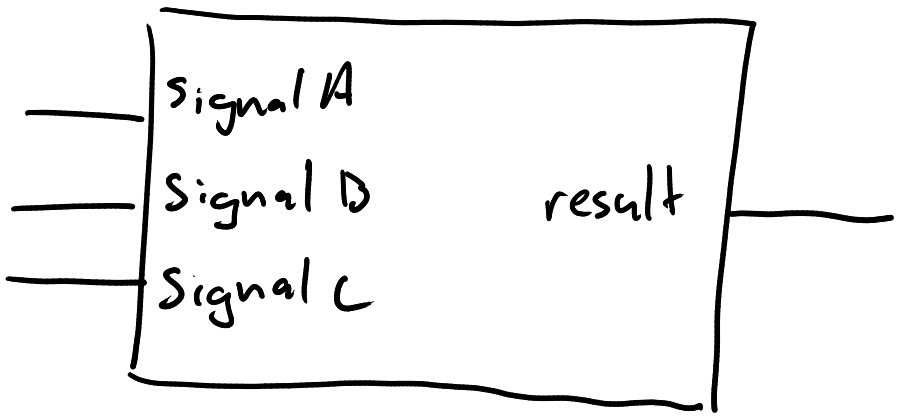
(aka hierarchical modeling)
(aka component instantiation)
(aka *composition* in software-speak)



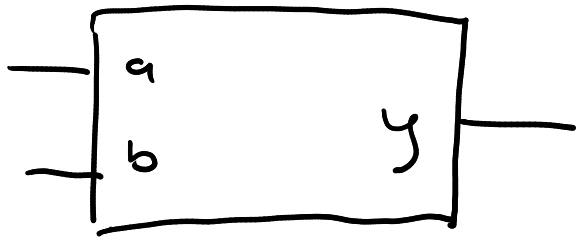
See the full videos on the course website, also Lab 5.

Instantiating a submodule in VHDL (entity declarations)

```
entity multigate is
  port (
    SignalA : in std_logic;
    SignalB : in std_logic;
    SignalC : in std_logic;
    result : out std_logic
  );
end;
```




```
entity andgate is
  port (
    a : in std_logic;
    b : in std_logic;
    y : out std_logic
  );
end;
```



Instantiating a submodule in VHDL (architecture)

```
architecture synth of multigate is
    component andgate is
        port(
            a : in std_logic;
            b : in std_logic;
            y : out std_logic
        );
    end component;
    signal ab : std_logic;
begin
        and1 : andgate port map(signalA, signalB, ab);
        and2 : andgate port map(ab, signalC, result);
end;
```

Inside an "architecture"
(but before "begin")

Component declaration
(because VHDL can't #include)

Instantiations

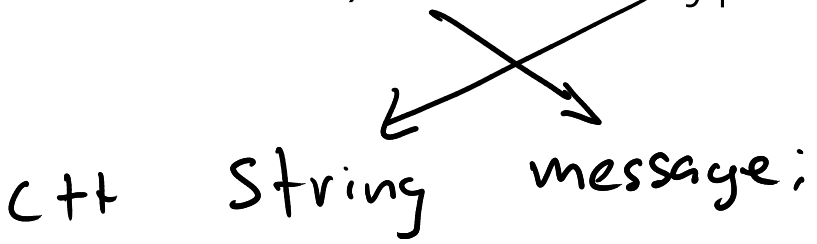
Anatomy of an instantiation

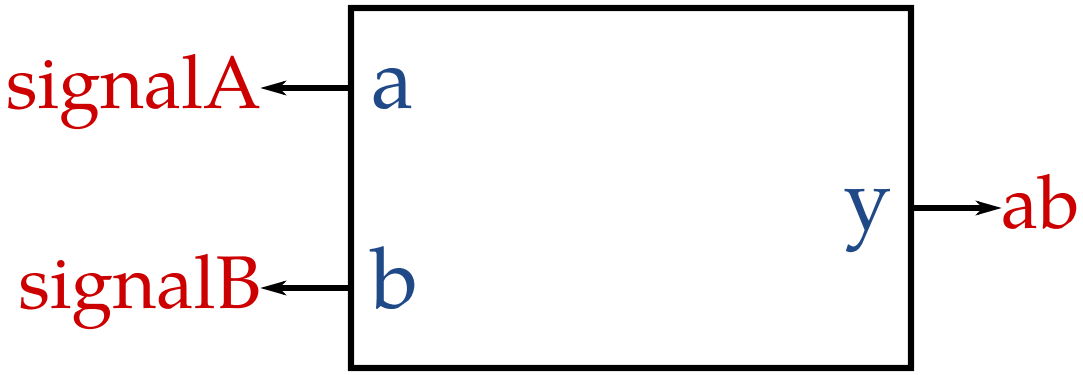
```
and1   :   andgate  port map(signalA, signalB, ab);
```

Instance name (like a variable name)

Module name (like a class or type name)

Port mapping (what connects to what?)





Mapping ports explicitly by name:

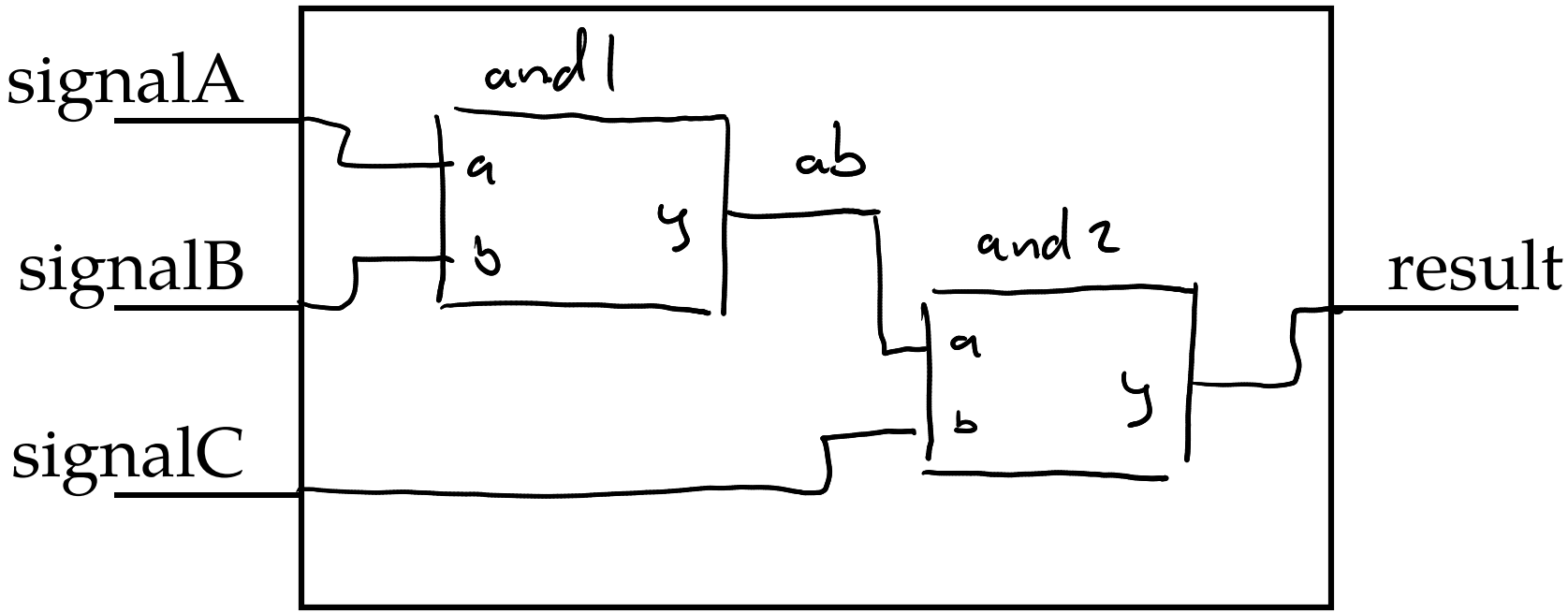
```
and1 : andgate port map(a=>signalA, b=>signalB, y=>ab);
```

(Good practice when you have many ports!)

Instantiating a submodule in VHDL (architecture)

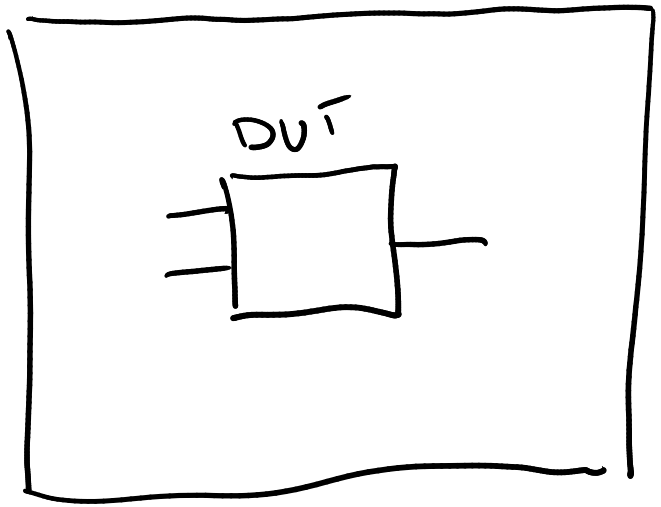
```
signal ab : std_logic;

and1 : andgate port map(signalA, signalB, ab);
and2 : andgate port map(ab, signalC, result);
```



Testbenches

A testbench is an entity with no ports,
and non-synthesizable code to test a submodule.



Who needs testbenches?

FPGA bugs are really hard to find compared to software bugs, because you have poor visibility.

And ASIC bugs are even worse!

Way more than half of digital design effort is in verification.

VHDL is concurrent

Within a module, all assignments happen simultaneously

VHDL is concurrent

Within a module, all assignments happen simultaneously

But we need some way to execute tests… like, sequentially.

Process block

```
process (SENSITIVITY)
begin
    stuff goes here
end process;
```

(We'll cover sensitivity in two weeks)

Process block

```
process (SENSITIVITY)
```
(We'll cover sensitivity in two weeks)

```
begin


end process;
```

Within a process block, statements execute from top to bottom (rather than "all at once").

(but don't get too excited yet)

Useful bits in a testbench

A process with no sensitivity will execute immediately and continuously

`wait` Wait forever (halt)

```
wait for 5 ns
```

space here!

```
assert CONDITION report "MESSAGE" severity SEVERITY
```

`report` Print something to the console log

Practice!

On VHDLweb:

Write code to print "Hello, world!" once

Write code to print "Hello!", and "Goodbye!" 200ns later

Write code to check if Y = 1 (you'll need to set it yourself)

Write code to check if Y = A and B

The big caveat

Inside a process:

statements are executed sequentially,

BUT

signals are not updated until time passes.

Practice: swap two ~~variables~~ signal values

Use the "testbench playground" on VHDLweb

Practice writing testbenches

VHDLweb "andgate" and "3-variable" problems

For next time

(I screwed up the deadlines)

1. Read the book (5.1-5.2) and complete the reading check (by 10/10)

2. Lab report 3 due next week, Lab 2 graded by this weekend

3. HW 2 & 3 due next week

4. Exam 1 is next Tuesday

Printing debugging info

```
report "hello, world!";
signal a : std_logic_vector(3 downto 0);
report "A is " & to_string(a);
```

Printing debugging info

```
report "hello, world!";
signal a : std_logic_vector(3 downto 0);
report "A is " & to_string(a);
```