Questions about homework 3?

ES 4: Testing and debugging VHDL

Steven Bell

6 October 2021

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

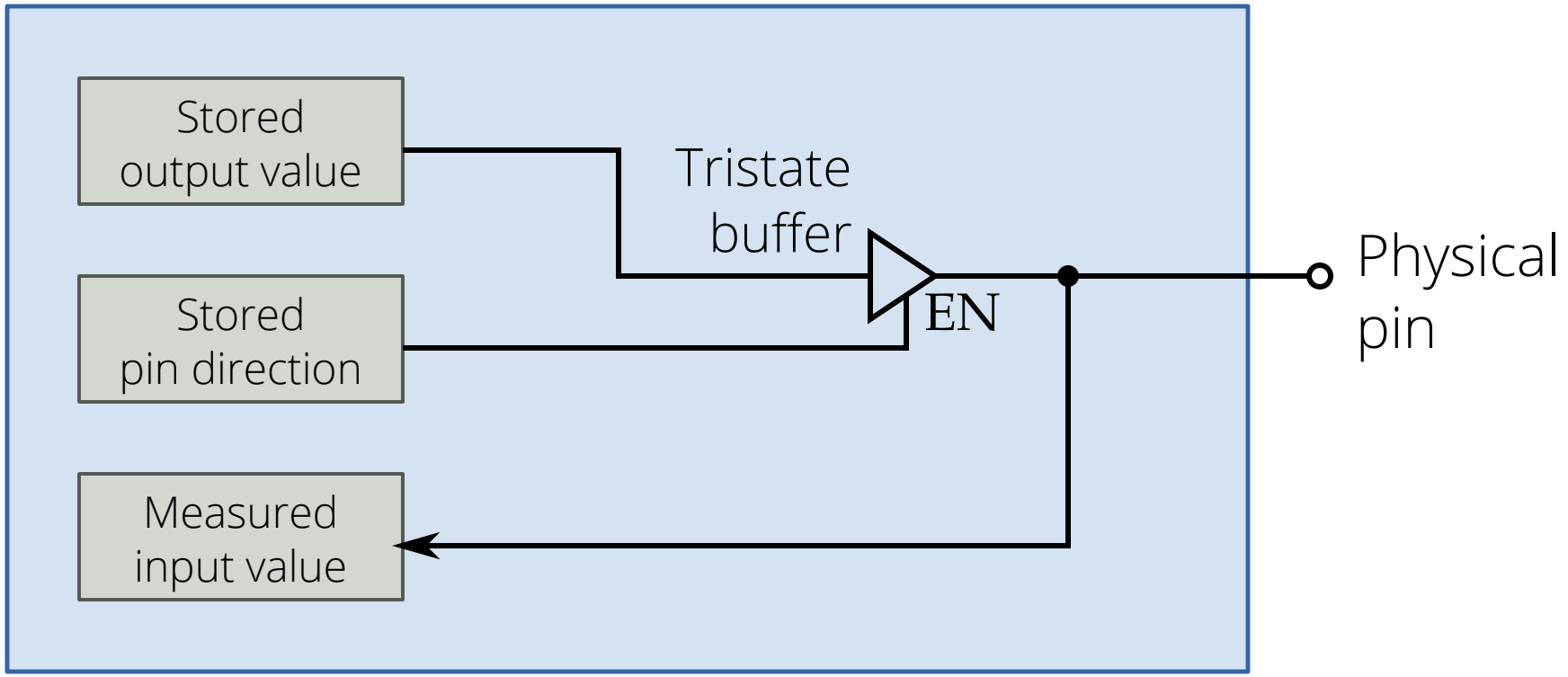
From the reading check

ahhh this is getting confusing.

Will we be doing any more VHDL practice?

Reading check question: tristates

In	Enable	Out
0	0	Z
1	0	Z
0	1	0
1	1	1

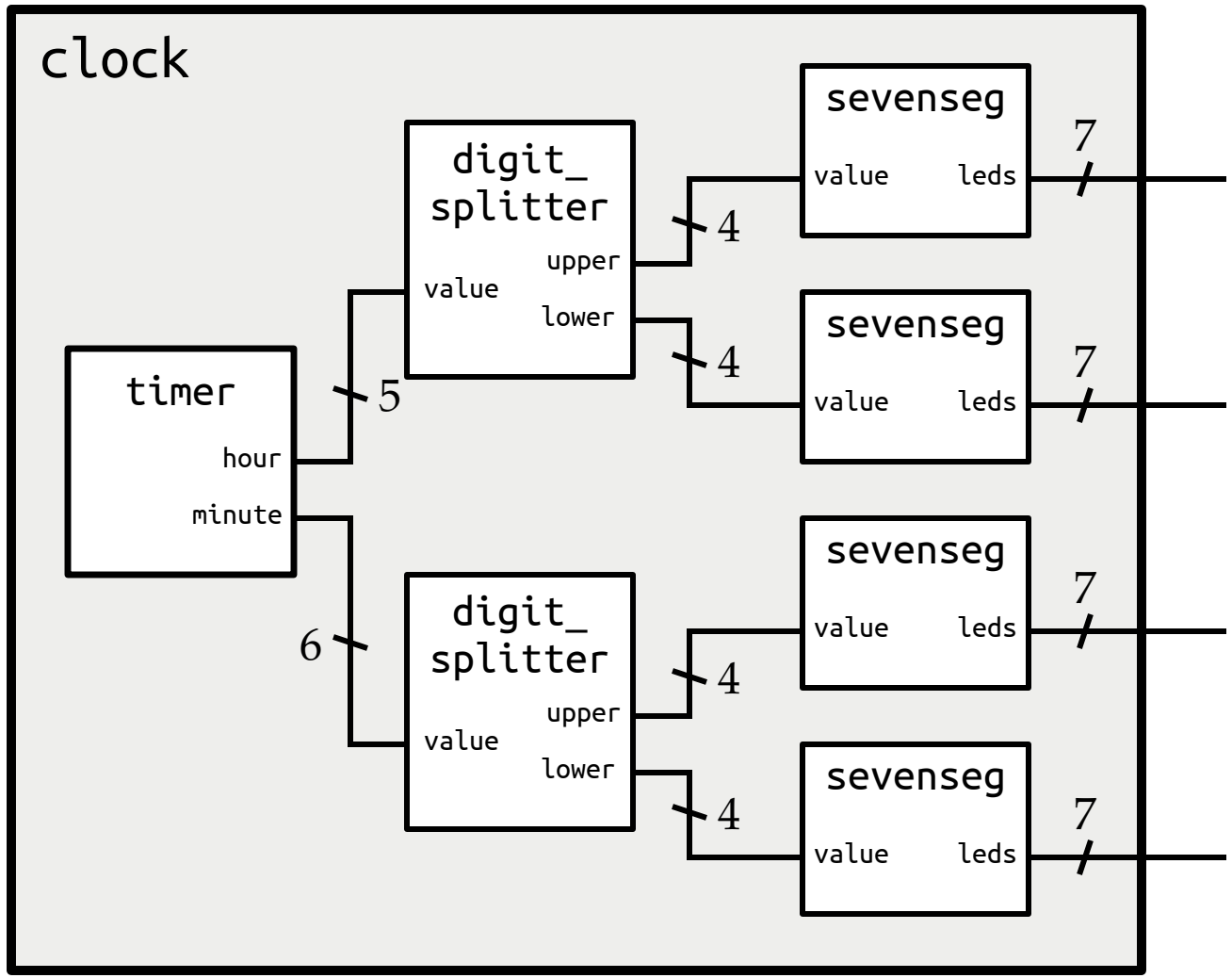


Reading check question: clocks

We'll talk about this after the midterm.

What's the point of structural modeling?

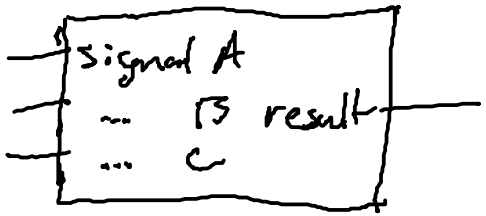
(aka structural modeling)
(aka hierarchical modeling)
(aka component instantiation)



See the full videos on the course website.

Instantiating a submodule in VHDL

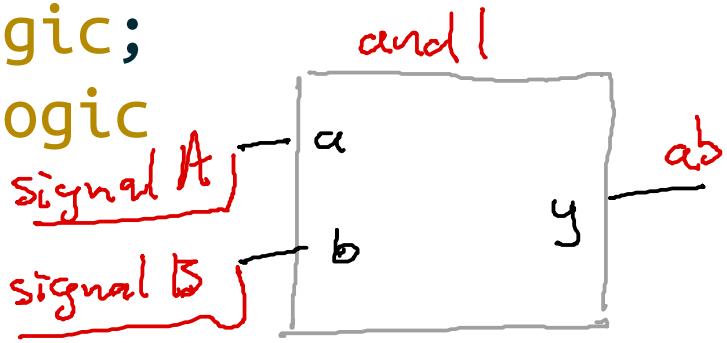
Textbook P.190



Inside an "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;
```

Component declaration



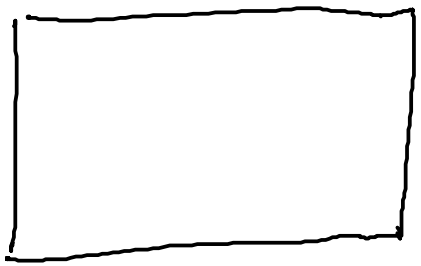
Instantiations

local name

component

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

When a signal here changes

```
process (SENSITIVITY)
begin
   -- if/case/print go here
end process;
```

Then do this stuff

```
process (a, b)
begin
   y <= a or b;
end process;
```

Process block

```
process (SENSITIVITY)
begin
   -- if/case/print go here
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
```

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

`report` Print something to the console log

Practice!

https://vhdlweb.com/problem/test_playground

The big caveat

Inside a process:

statements are executed sequentially,

BUT

signals are not updated until time passes.

Practice: swap two variables

https://vhdlweb.com/problem/test_playground

Practice writing testbenches

VHDLweb.com

What is one question you have from today?

Submit your answer on **pollev.com/stevenbell**

For next week (Monday is a holiday)

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

2. Lab report 2 due this week

3. Lab report 3 due next week

Printing debugging info

```
report "hello, world!";

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

(concatenation) (conversion to string)

Printing debugging info

```
report "hello, world!";

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

(concatenation) (conversion to string)

Only in VHDL 2008!

If sensitivity includes:

`all↕` → Combinational logic

`clk↑` → Flip-flop / register

After the midterm

`clk↑ + data↕` → Latch

Nothing → Testbench (continuous evaluation)

Something else → Bad things you probably didn't want.