Warmup

A co-worker has implemented the circuit below in VHDL.
How do you check that it is correct?

Submit your answer on **pollev.com/stevenbell699**

ES 4: Testing and debugging VHDL

Steven Bell

20 February 2019

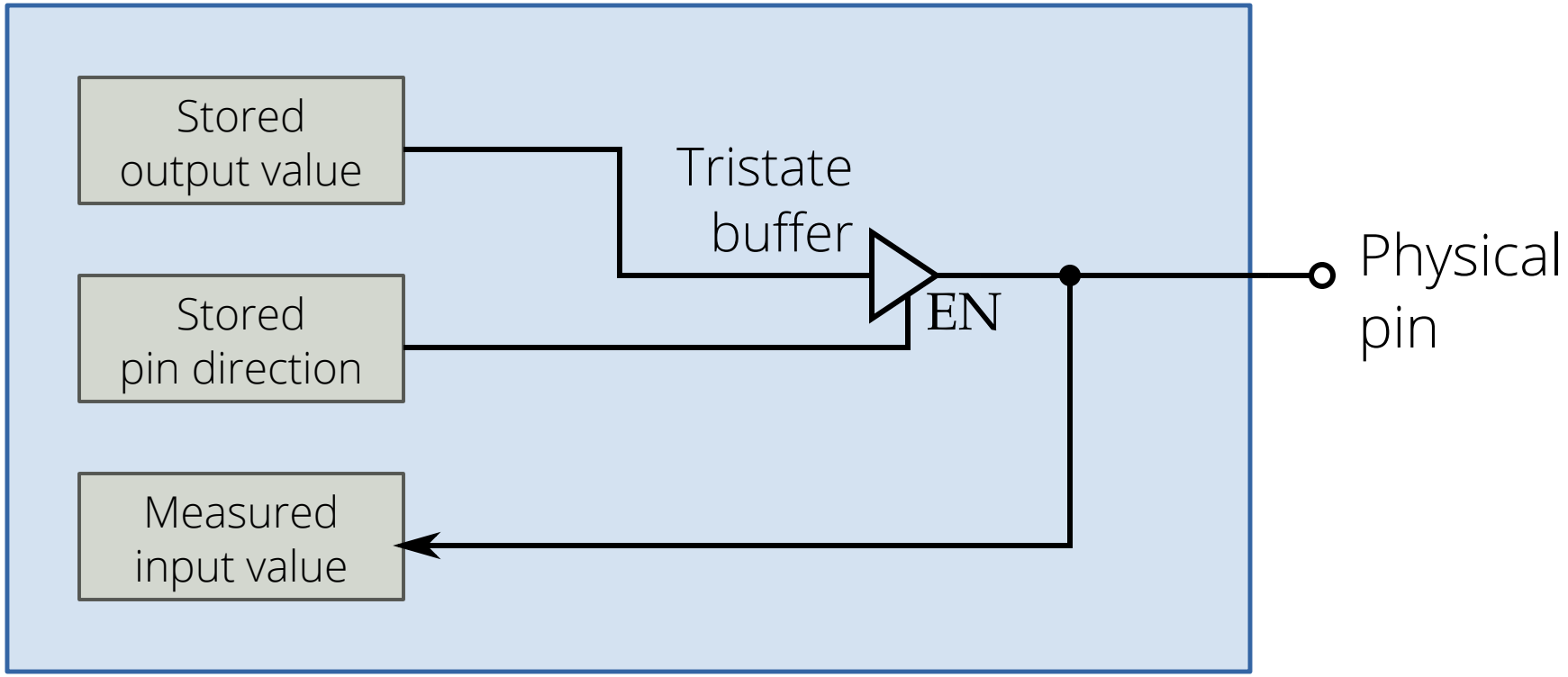
Tufts UNIVERSITY

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

- Use the VHDL `process` structure for combinational logic and print-style debugging
- Write a testbench for combinational logic
- Explain the difference between synthesizable and non-synthesizable VHDL constructs

Quiz question: tristates

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



Quiz question: clocks

We'll talk about this after the midterm.

Demo time: why is VHDL useful?

Using submodules (aka structural modeling)

```
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"

Component declaration

Instantiations

Logistics stuff

Labs are being graded

HW 1 is graded; HW 2 is in process

CELT is coming on Monday

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!

Process block

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

Process block

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

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.

Practice!

http://vhdlweb.com:8000

Testbenches

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

Useful bits in a testbench

`wait` Wait forever (halt)

```
wait for 5 ns
```

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

`report` Print something to the console log

Practice!

http://vhdlweb.com:8000

What is one question you have from today?

Submit your answer on **pollev.com/stevenbell699**

For tomorrow (Thursday is the new Monday)

1. Read the book (5.1-5.2) and complete the pre-class quiz
2. Homework 4 is due next Monday (2/25) via `provide es4 hw4`
3. Lab report 2 due next Wednesday (2/27)