CENG232
Verilog Tools Recitation
Icarus & GTKWave

Outline

- Hardware description languages
- Testbench idea
- Verilog
  - Language overview
  - Compiler: Icarus (iverilog and vvp)
  - Waveform viewer: GTKWave (gtkwave)

Hardware Description Lang.

- Designing schematics wire by wire is:
  - Hard
  - Error-prone
  - Gets more complex as project gets bigger
- HDL
  - Describe your circuits as textual ”code”, just like any programming language
  - Divide larger projects into smaller modules
  - Harder to visualize, but easier to manage

Hardware Description Lang.

- Two popular languages:
  - VHDL
    - Complex at initial glance
    - Provides better control when synthesizing to physical elements
    - Preferred in industry
  - Verilog
    - Simpler syntax
    - Easier to learn

Testbench idea

- HDLs provide you a way to write modules
  - with their inputs and outputs
- Now, how to test it?
  - Create a test-module, that instantiates your own module, writes to and reads from your module's inputs/outputs
  - Connect inputs and outputs of the test module to your module
  - Run the testbench

Testbench idea

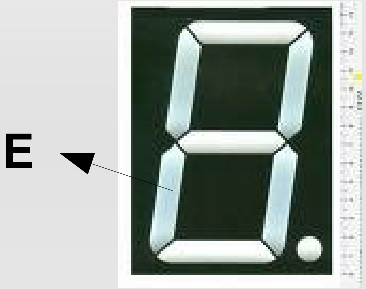
- Same as plugging a chip to a physical test board which tests various inputs and blinks green light if everything's ok
- You don't have to blink a green light:
  - HDLs provide a way to write to screen:

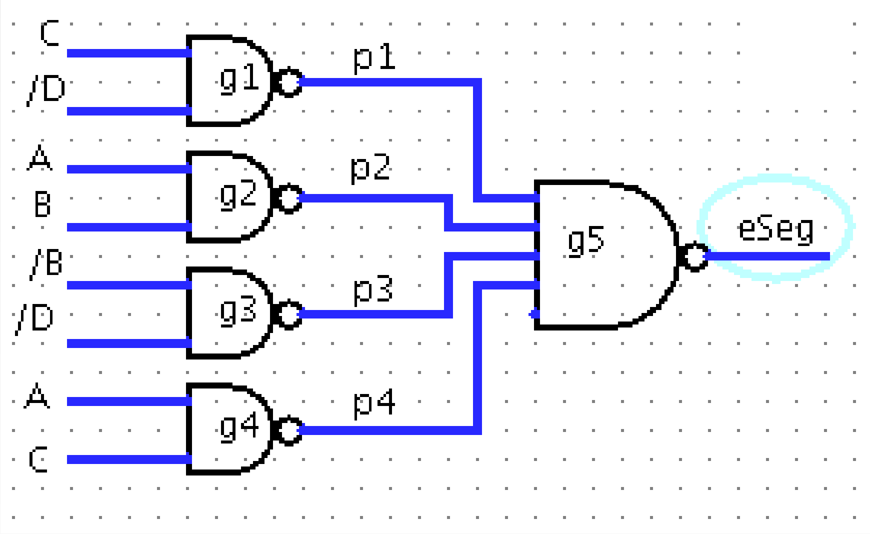
```
TEST1 passed
TEST2 failed
TEST3 passed
...
```

Verilog

```
module binaryToESeg;
    wire    eSeg, p1, p2, p3, p4;
    reg A, B, C, D;

    nand #1
        g1 (p1, C, ~D),
        g2 (p2, A, B),
        g3 (p3, ~B, ~D),
        g4 (p4, A, C),
        g5 (eSeg, p1, p2, p3, p4);
endmodule
```





Tools

- Icarus
  - Provides iverilog and vvp commands (install package ”verilog” in ubuntu)
    - iverilog: Compiler
    - vvp: Simulator. Generates a VCD file.
- GTKWave
  - Viewer for VCD files
  - You can see all waveform changes that occured in the simulation.
  - Install package gtkwave in ubuntu

Tools: Icarus (iverilog)

- iverilog is the compiler
  - Source files are named *.v
  - A .v file can contain multiple modules
  - A project can consist of multiple files

- iverilog -o hello hello.v
  - Compiles source ”hello.v” into the file ”hello”

```
iverilog -o output file1.v file1_testbench.v
```

  - Compiles multiple files into a single project

Tools: Icarus (vvp)

- vvp is the simulator
- Run with the name of the compiled file:

```
/home/kerem$ iverilog -o hello hello.v
/home/kerem$ vvp hello
```

- There are some special commands inside the source-file that affect simulation:
  - $stop, $dumpfile, $dumpvars, $monitor, $display, ..

Tools: Icarus (vvp)

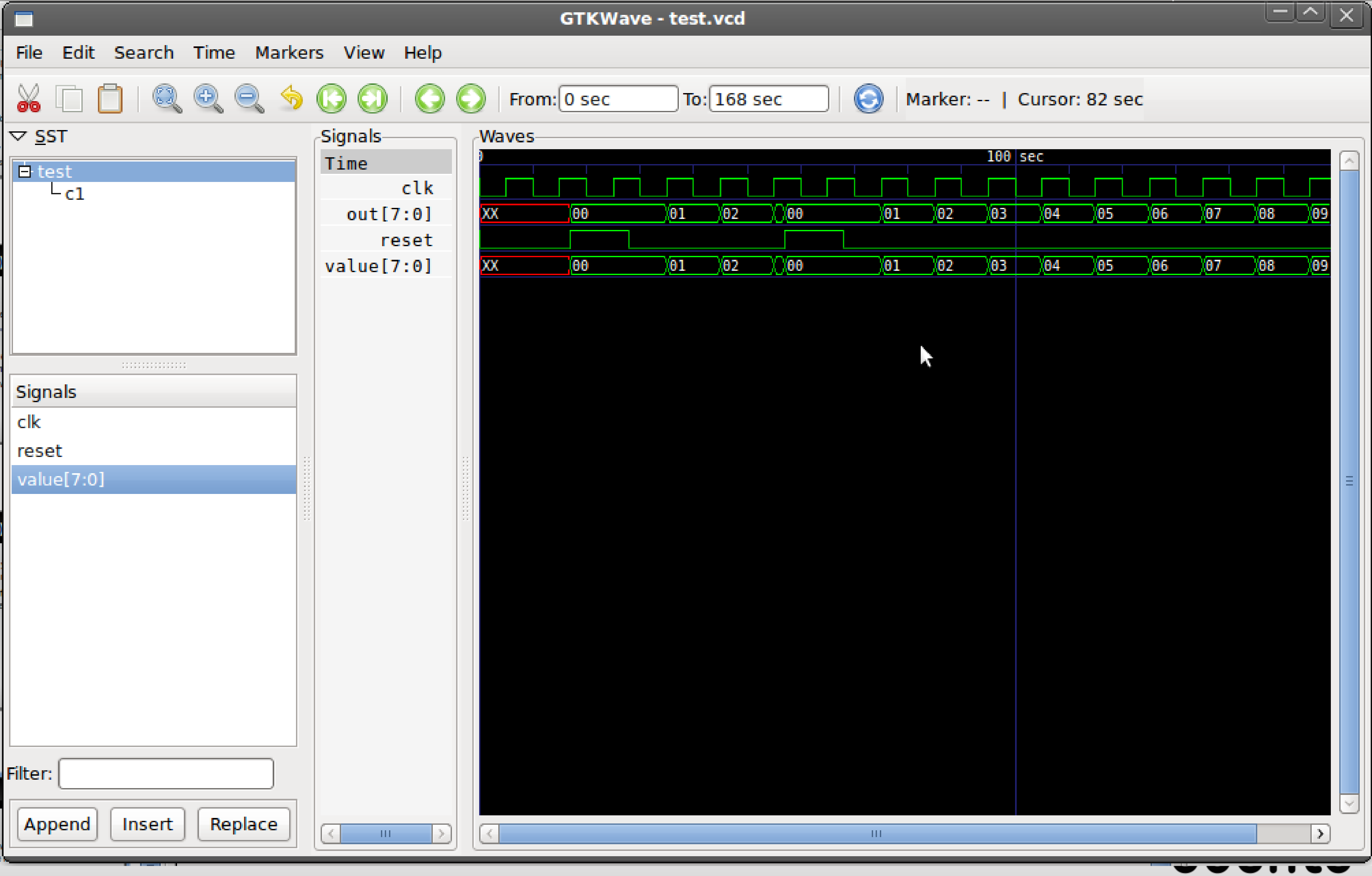
- To generate a VCD file, you need to specify $dumpfile and $dumpvars in source file
- These specify which part of the circuit will be logged, and the filename:

```
module my_testbench
....
initial
    begin
        $dumpfile("test.vcd");
        $dumpvars(0,my_testbench); // will log everything in my_testbench
    end
endmodule
```

Tools: GTKWave

- gtkwave filename.vcd
- Once you open GTKWave, you can see all the module hierarchy on the left side
- Select a module, and you will see it's signals
- You can move the desired signals to the right panel, and see their waveforms on the right

GTKWave



GTKWave - test.vcd

From: 0 sec To: 168 sec

Marker: -- | Cursor: 82 sec

SST

- test
  - c1

Signals

- clk
- reset
- value[7:0]

Filter:

Signals

- Time
- clk
- out[7:0]
- reset
- value[7:0]

Waves