# GETTING STARTED WITH THE EMP FRAMEWORK – PART 4

ANDY ROSE, IMPERIAL COLLEGE LONDON

# STATING THE OBVIOUS

- You will already be bored with waiting for firmware to build

# STATING THE OBVIOUS

- You will already be bored with waiting for firmware to build
- IPBB & EMP has a solution - obviously

# RECALL

- To create a Vivado project, we did:

```
ipbb proj create vivado my_algo my-algo-repo:an-algo -t top.dep

cd proj/my_algo

ipbb vivado project
```

# WE COULD ALSO…

- Replace `include -c emp-fwk:boards/kcu105`
- With `include -c emp-fwk:boards/testbench`

- And create a “board” that is actually a testbench running on the PC

# WARNING

- Vivado works out the dependencies of sources on one another
- ModelSim does not
- Sometimes have to play with line ordering to make Modelsim happy

# CREATE A TESTBENCH

- Open a new depfile, top.sim.dep
- And add the line `include -c emp-fwk:boards/testbench`
  - Since our “board” is actually going to be a simulation
- Add the line `src emp_payload.vhd`
  - Seems reasonable, since we want to simulate our payload

# CREATE A TESTBENCH

- And add the voodoo

```
src -c emp-fwk:components/datapath emp_data_types.vhd
src -c ipbus-firmware:components/ipbus_core ipbus_package.vhd
src -c emp-fwk:components/ttc emp_ttc_decl.vhd
```

- This is only because the testbench framework is currently being updated and should not be necessary

# CREATE A TESTBENCH

- We need to configure our testbench, the same way we need to configure the EMP infrastructure, using a declaration file
- There is a handy declaration for us to copy

```
cp src/emp-fwk/projects/examples/testbench/firmware/hdl/tb_decl.vhd
   src/my-algo-repo/an-algo/firmware/hdl
```

- Add the line to your dep file: `src tb_decl.vhd`

# CREATE A TESTBENCH

- Add more voodoo

```
src -c emp-fwk:components/framework emp_device_types.vhd
src -c emp-fwk:boards/testbench emp_device_decl.vhd
src -c emp-fwk:components/framework emp_framework_decl.vhd
```

- This is only because the testbench framework is currently being updated and should not be necessary

# NOTE

- We do not need to

```
src -c emp-fwk:components/payload ../ucf/emp_simple_payload.tcl

addrtab -c emp-fwk:components/payload emp_payload.xml
```

- Since constraining the area has no meaning for simulations
- We will not use the IPbus interface

# CREATE A TESTBENCH

```
ipbb proj create sim my_algo_sim my-algo-repo:an-algo -t top.sim.dep

cd proj/my_algo_sim

ipbb sim setup-simlib

ipbb sim ipcores

ipbb sim make-project
```

# CREATE A TESTBENCH

```
ipbb proj create sim my_algo_sim my-algo-repo:an-algo -t top.sim.dep

cd proj/my_algo_sim

ipbb sim setup-simlib

ipbb sim ipcores

ipbb sim make-project
```

Creates a project space for a simulation project

# CREATE A TESTBENCH

```
ipbb proj create sim my_algo_sim my-algo-repo:an-algo -t top.sim.dep

cd proj/my_algo_sim

ipbb sim setup-simlib

ipbb sim ipcores

ipbb sim make-project
```

We need compiled simulation libraries for Xilinx IP

# CREATE A TESTBENCH

```
ipbb proj create sim my_algo_sim my-algo-repo:an-algo -t top.sim.dep

cd proj/my_algo_sim

ipbb sim setup-simlib

ipbb sim ipcores

ipbb sim make-project
```

Make the project

# RUN THE TESTBENCH

- Now all that is left to do is run the simulation!

```
vsim -c work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'run 5us' -do 'quit'
```

# RUN THE TESTBENCH

- Now all that is left to do is run the simulation!

```
vsim -c work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'run 5us' -do 'quit'
```

We are running in command-line mode, no GUI

# RUN THE TESTBENCH

- Now all that is left to do is run the simulation!

```
vsim -c work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'run 5us' -do 'quit'
```

The project we just created

# RUN THE TESTBENCH

- Now all that is left to do is run the simulation!

```
vsim -c work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'run 5us' -do 'quit'
```

Where we get out test data from
Note – this is the file we got from the hardware, so this is exactly the same data we have previously used in HW

# RUN THE TESTBENCH

- Now all that is left to do is run the simulation!

```
vsim -c work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'run 5us' -do 'quit'
```

Where we store the output to

# CREATE A TESTBENCH

- Now all that is left to do is run the simulation!

```
vsim -c work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'run 5us' -do 'quit'
```

We want to run for some simulated period and then quit
No human intervention required
Perfect of automated validation!

# RUN THE TESTBENCH

- Now all that is left to do is run the simulation!

```
vsim -c work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'run 5us' -do 'quit'
```

- Check the output file and make sure that it makes sense

# RUN THE TESTBENCH INTERACTIVELY

- If something is wrong in your output, you will want to look inside your algo!

```
vsim -i work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'noview *' -do 'view wave' -do 'add wave sim:/top/payload/*'
     -do 'run 5us'
```

# RUN THE TESTBENCH INTERACTIVELY

- If something is wrong in your output, you will want to look inside your algo!

```
vsim -i work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'noview *' -do 'view wave' -do 'add wave sim:/top/payload/*'
     -do 'run 5us'
```

We are running in interactive mode

# RUN THE TESTBENCH INTERACTIVELY

- If something is wrong in your output, you will want to look inside your algo!

```
vsim -i work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'noview *' -do 'view wave' -do 'add wave sim:/top/payload/*'
     -do 'run 5us'
```

We are interested in what our signals in the payload are doing and nothing else

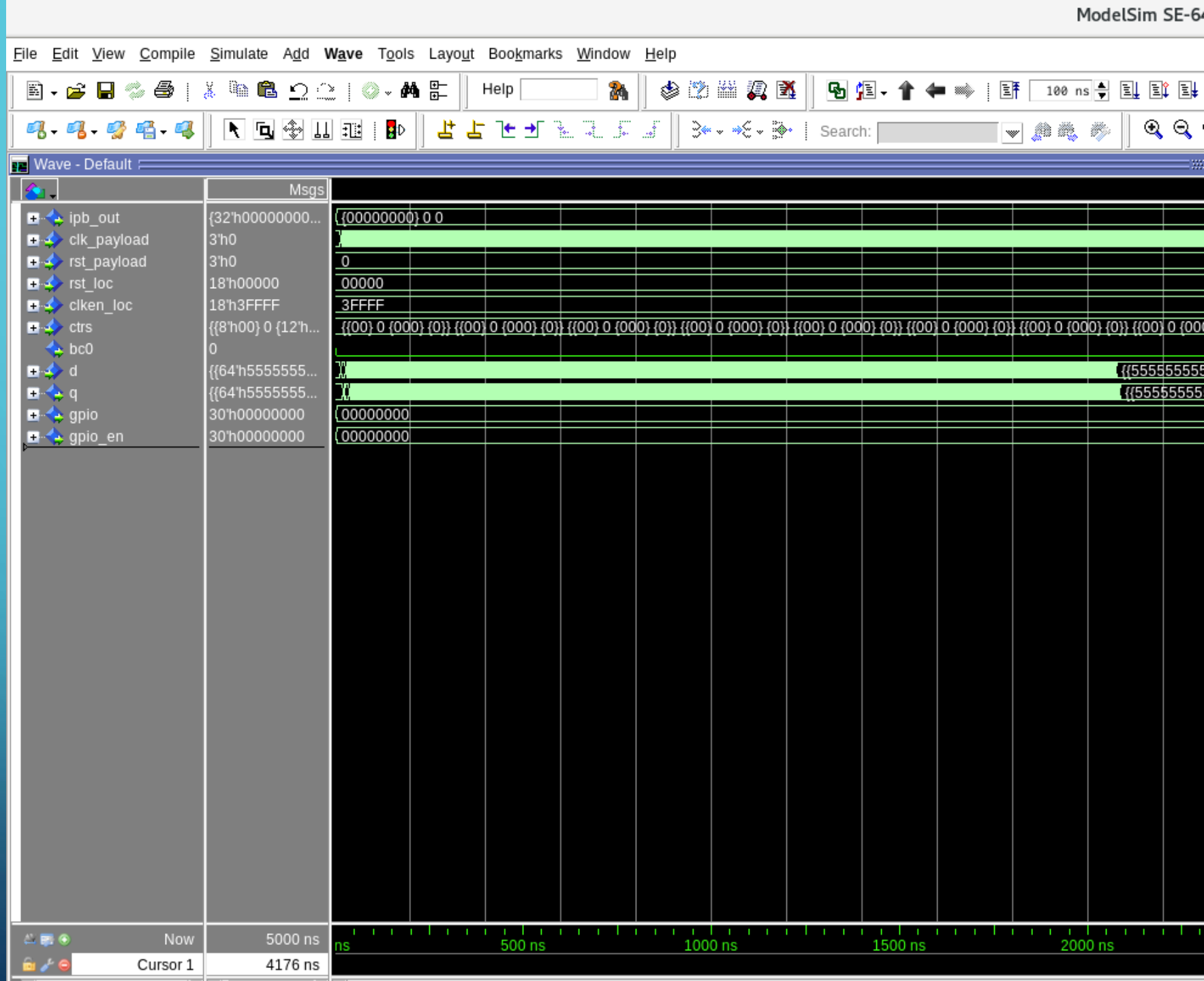
# RUN THE TESTBENCH INTERACTIVELY

- If something is wrong in your output, you will want to look inside your algo!

```
vsim -i work.top
     -Gsourcefile=/home/user/my-software/data/rx_summary.txt
     -Gsinkfile =/home/user/my-software/data/sim_tx_summary.txt
     -do 'noview *' -do 'view wave' -do 'add wave sim:/top/payload/*'
     -do 'run 5us'
```

This time run, but leave the program open!

# RUN THE TESTBENCH INTERACTIVELY



# RUN THE TESTBENCH INTERACTIVELY

- Inspect the waves and check they make sense to you!