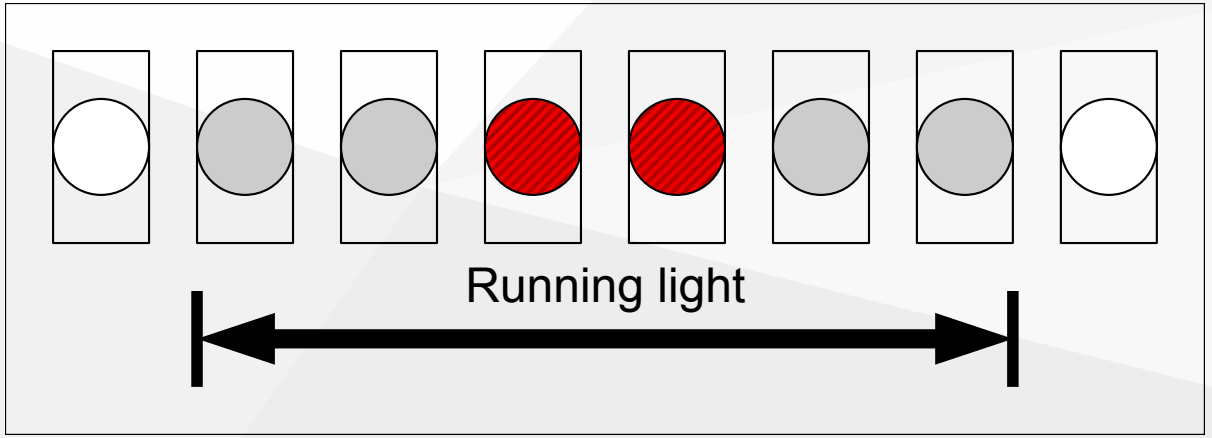
RISC-V Lab

Ex1: Introduction



Content

1. Organisation
2. Project overview
3. Exercise 1: HDL Entry & Simulation
4. Group registration

Organisation

- **4** students per team
- `project` : **self defined** HW (FPGA) / SW (RISC-V)
  e.g.: real time video & audio, many-core X, networking...

Formals

- module = lecture + project
- 6 ECTS (<=> *27.5h / 12 lecture weeks = 14h/week)
- project ET / CS
- see module description for exam details

(To) Do's

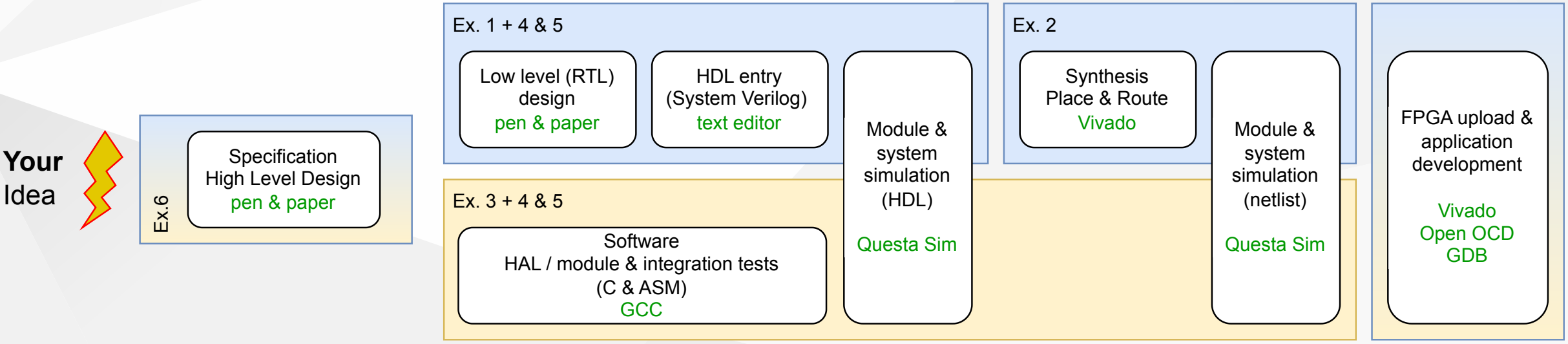
1. 6 assignments **completed in time** (=one week, deadline 2 weeks)
   walk you through the design flow & build base of your project
2. your project's **self specified** functionality demonstrated **on FPGA**
3. **take part** in mid- and final presentations (incl. video clip)
4. project specification & documentation
   ("on the go", effort minimized)

Support

- `rvlab/doc/_build/html/index.html` !!
- IT (workstations, EDA tools, licenses ..)
  - Adela Westedt: adela.westedt@tu-berlin.de, 24248, EN435a
  - Rene Hartman: rene.hartmann@tu-berlin.de, 25351, EN435a
- Marco Liem: marco.liem@campus.tu-berlin.de (-> ZOOM)
- Tobias Kaiser: kaiser@tu-berlin.de, 25561, EN411

Rvlab Overview

Rvlab Flow



Rvlab System

gcc

Questa Sim

Vivado

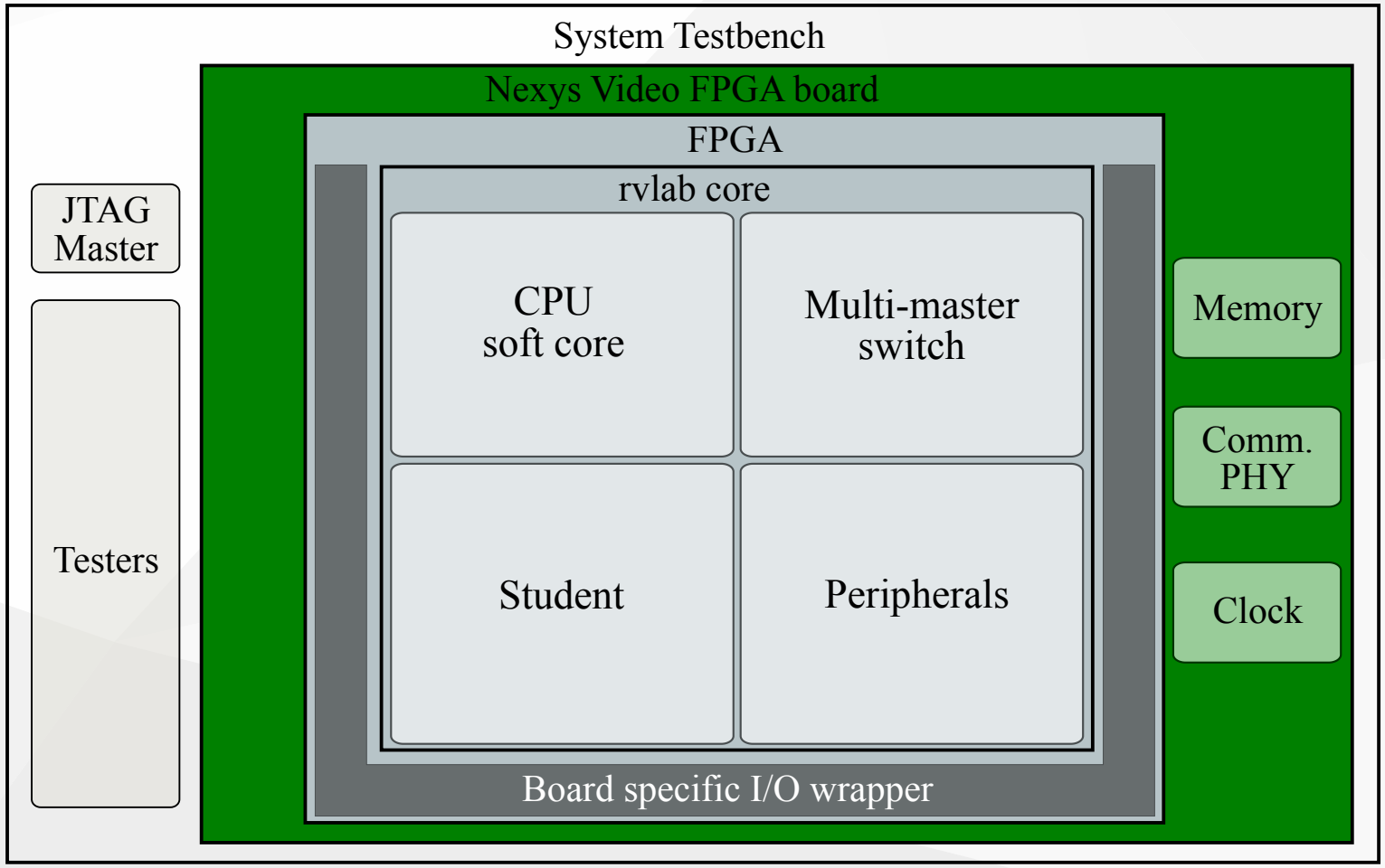
OpenOCD
GDB

Python
design flow

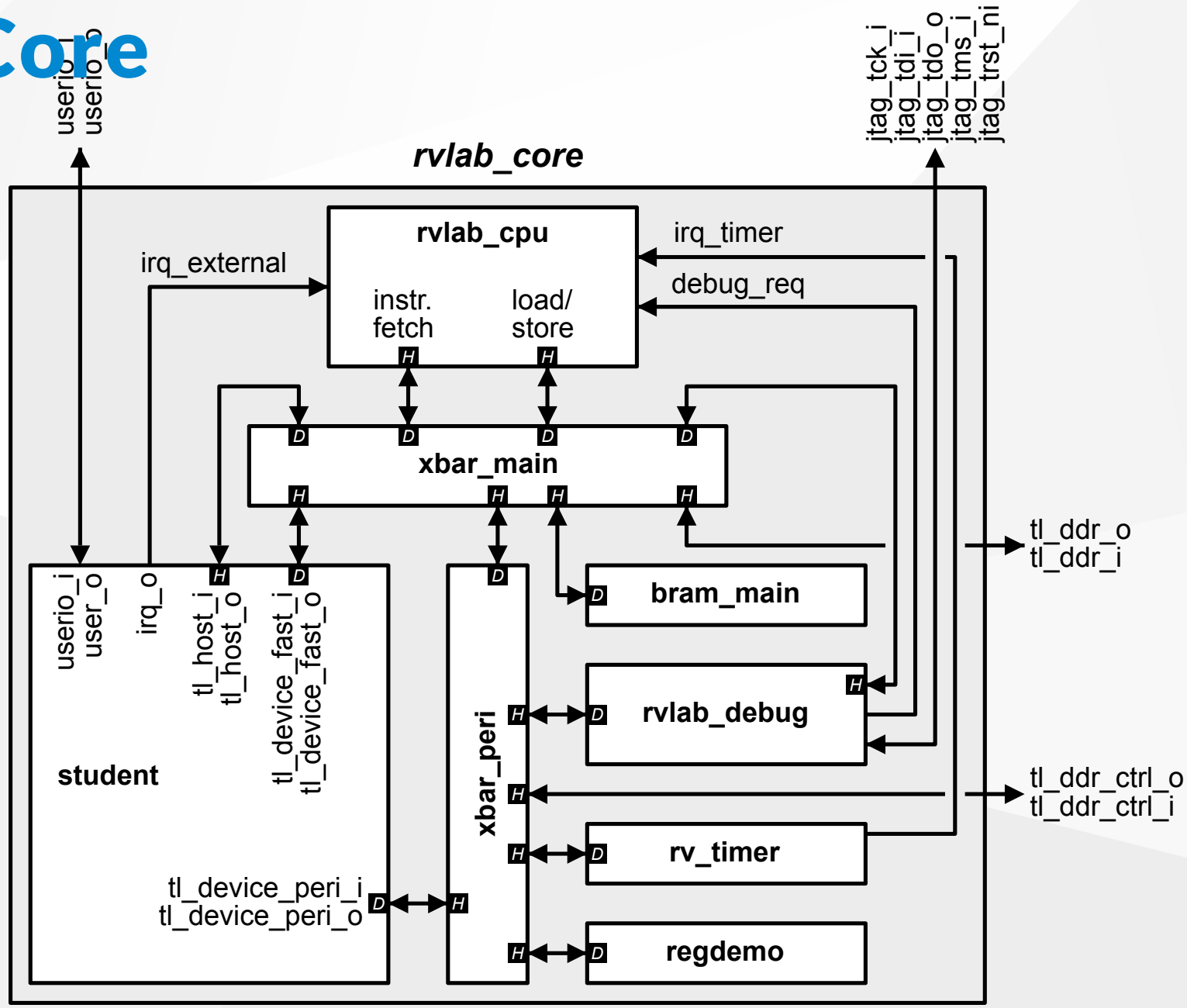
Simulation
software

Monitor

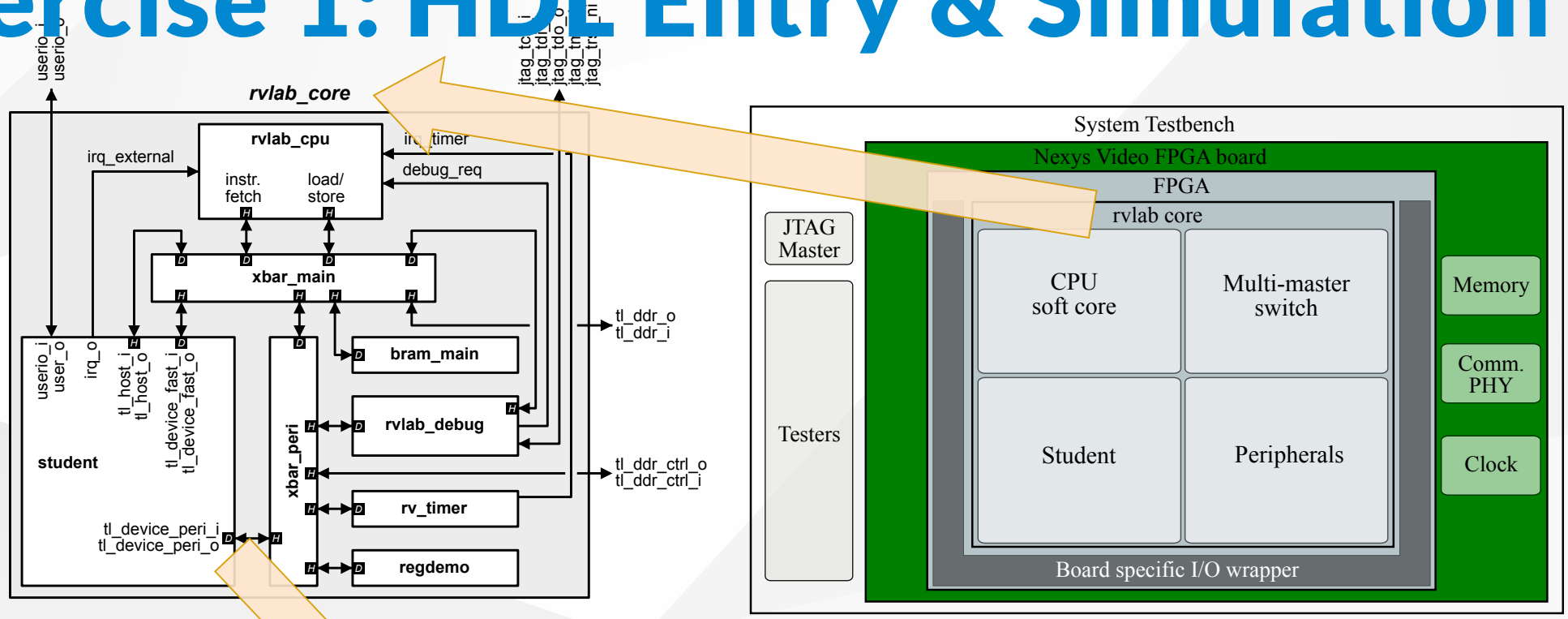
baselibc

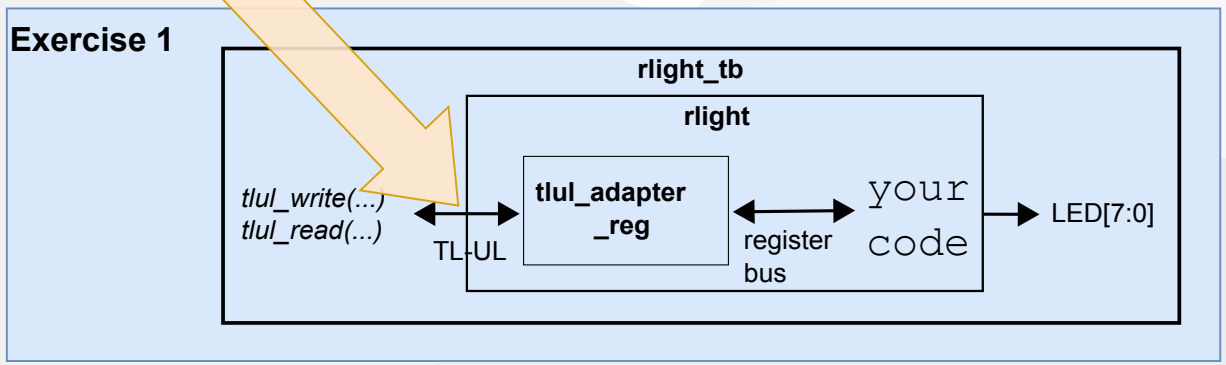


Rvlab Core


rvlab_core
userio_i
userio_o
jtag_tck_i
jtag_tdi_i
jtag_tdo_o
jtag_tms_i
jtag_trst_ni
rvlab_cpu
instr. fetch
load/ store
irq_external
irq_timer
debug_req
xbar_main
tl_ddr_o
tl_ddr_i
bram_main
rvlab_debug
student
userio_i
user_o
irq_o
tl_host_i
tl_host_o
tl_device_fast_i
tl_device_fast_o
xbar_peri
tl_ddr_ctrl_o
tl_ddr_ctrl_i
rv_timer
tl_device_peri_i
tl_device_peri_o
regdemo

Exercise 1: HDL Entry & Simulation

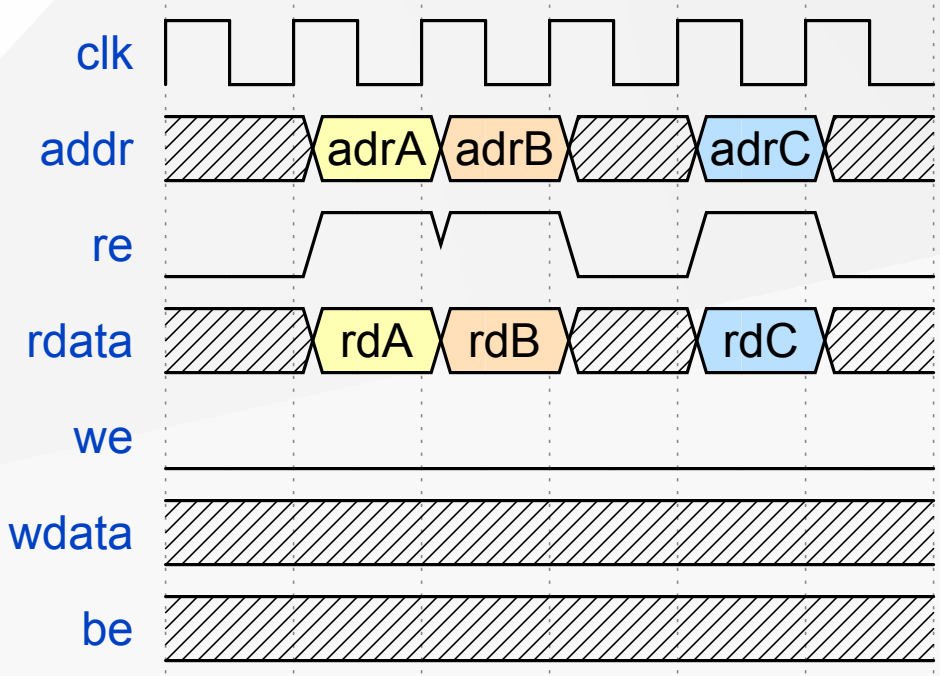




Register bus read

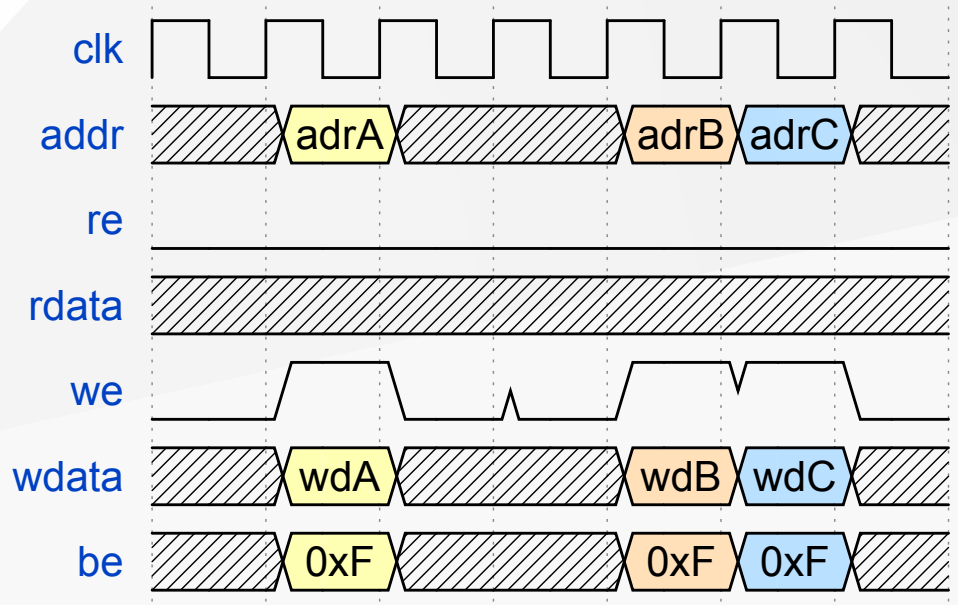
```
logic [ 7:0] regA;
logic [31:0] regB;
localparam logic [3:0] ADDR_REGA = 4'h0;
localparam logic [3:0] ADDR_REGB = 4'h4;

always_comb begin
  rdata = '0;  // !!!
  case (addr)
    ADDR_REGA:  rdata[7:0]  = regA;
    ADDR_REGB:  rdata[31:0] = regB;
    rdata = 32'hDEADAFFE;
  endcase
end
```



Register bus write

```
always_ff @(posedge clk_i, negedge rst_ni) begin
  if (~rst_ni) begin
    regA   <= 8'b10101010;
    regB   <= 32'hDEADAFFE;
  end else begin
    if (we) begin
      case (addr)
        ADDR_REGA:   regA <= wdata[7:0];
        ADDR_REGB:   regB <= wdata[31:0];
        default: begin
        end
      endcase
    end // if(we)
  end // if (~rst_ni) else
end
```



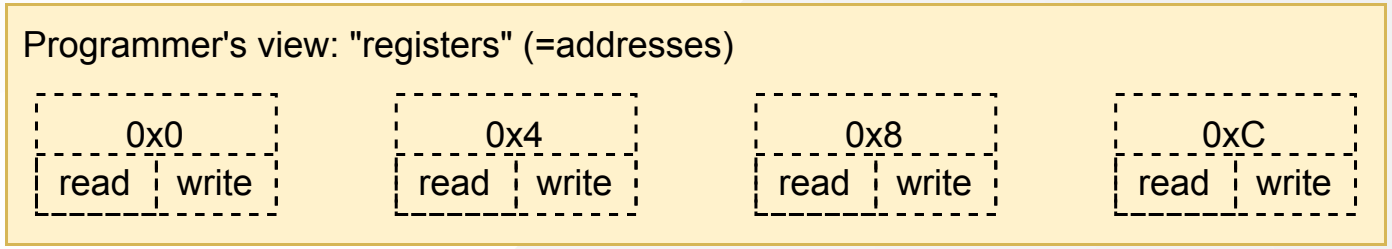
RTL Simulation with Questa Sim

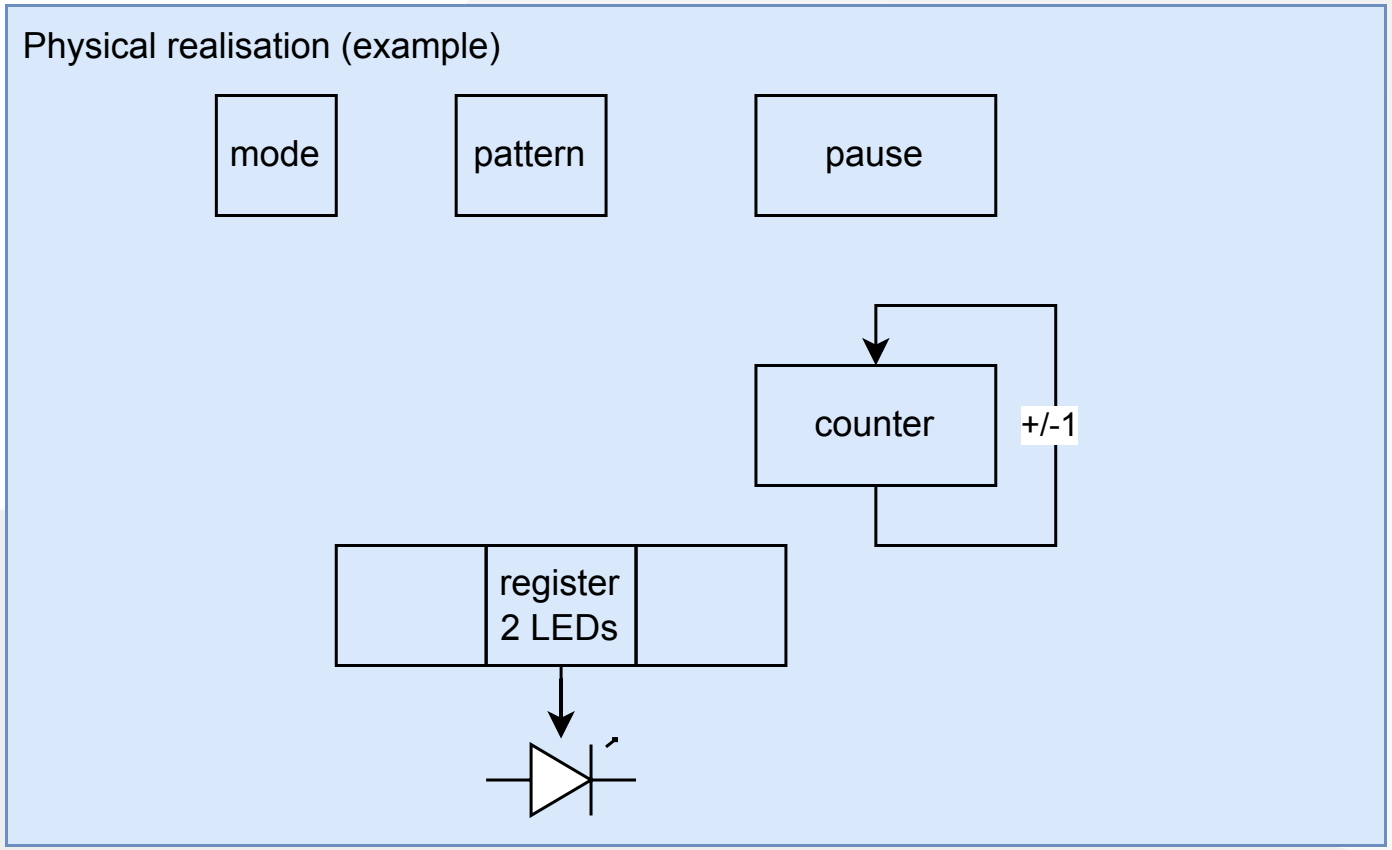
- Inputs
  - RTL design in *synthesisable* HDL
  - test bench in HDL
  - tcl scripts: wave view formatting (.do), ...
- Outputs
  - value trace of all signals
  - messages from display$()

```
flow rlight_tb.sim_rtl_questa
```

Hints for Ex.1

- address space ("registers") != physical registers
- handle conflicting writes (HW vs. SW)! Suggestions !





Group registration

1. **Self organize** groups of 4 students each
2. Each group comes forward & registers
3. Each group receives one account for the msc network
   user: rvlab[1-9] // password: idea2soc2023
4. Log in using wiki.x2go.org
   i) connect to MSC VPN via TUBIT VPN
   ii) X2GO server: msclab.msc.tu-berlin.de, desktop: XFCE)
5. Change password immediately: `passwd`

Detailed instructions for VPN & X2GO linked on ISIS page.

Appendix

Kill a dead session on the server

1. ssh on the server
2. `x2golistsessions` to list your running sessions and get their ID
   It will return a string, e.g. 494|slehmann-62-1536514901_stDXFCE_dp24|62|mscx2go|...
   The second field is the session ID e.g. "slehmann-62-1536514901_stDXFCE_dp24"
3. `x2goterminate-session slehmann-62-1536514901_stDXFCE_dp24`