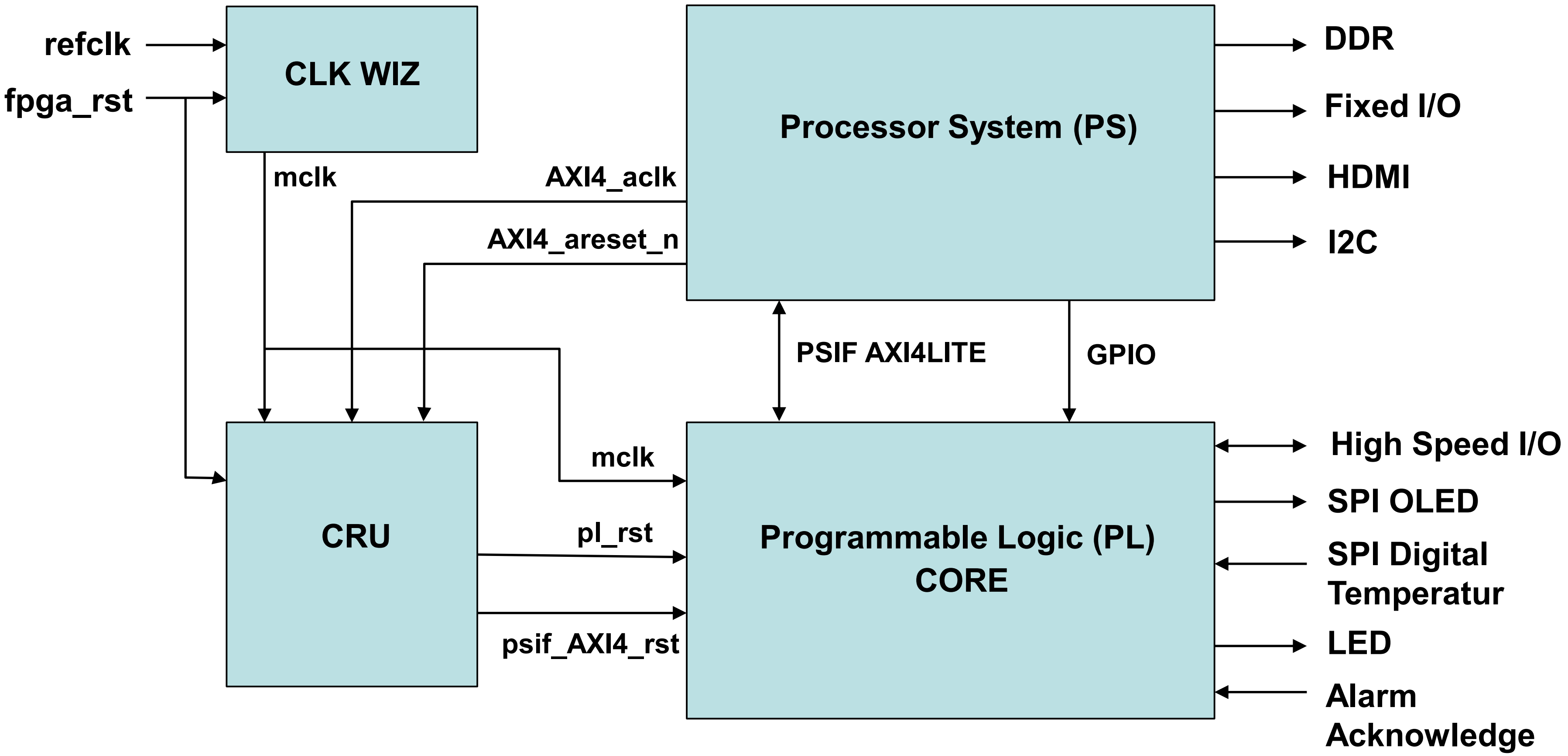
UiO : Department of Informatics
University of Oslo

IN5200

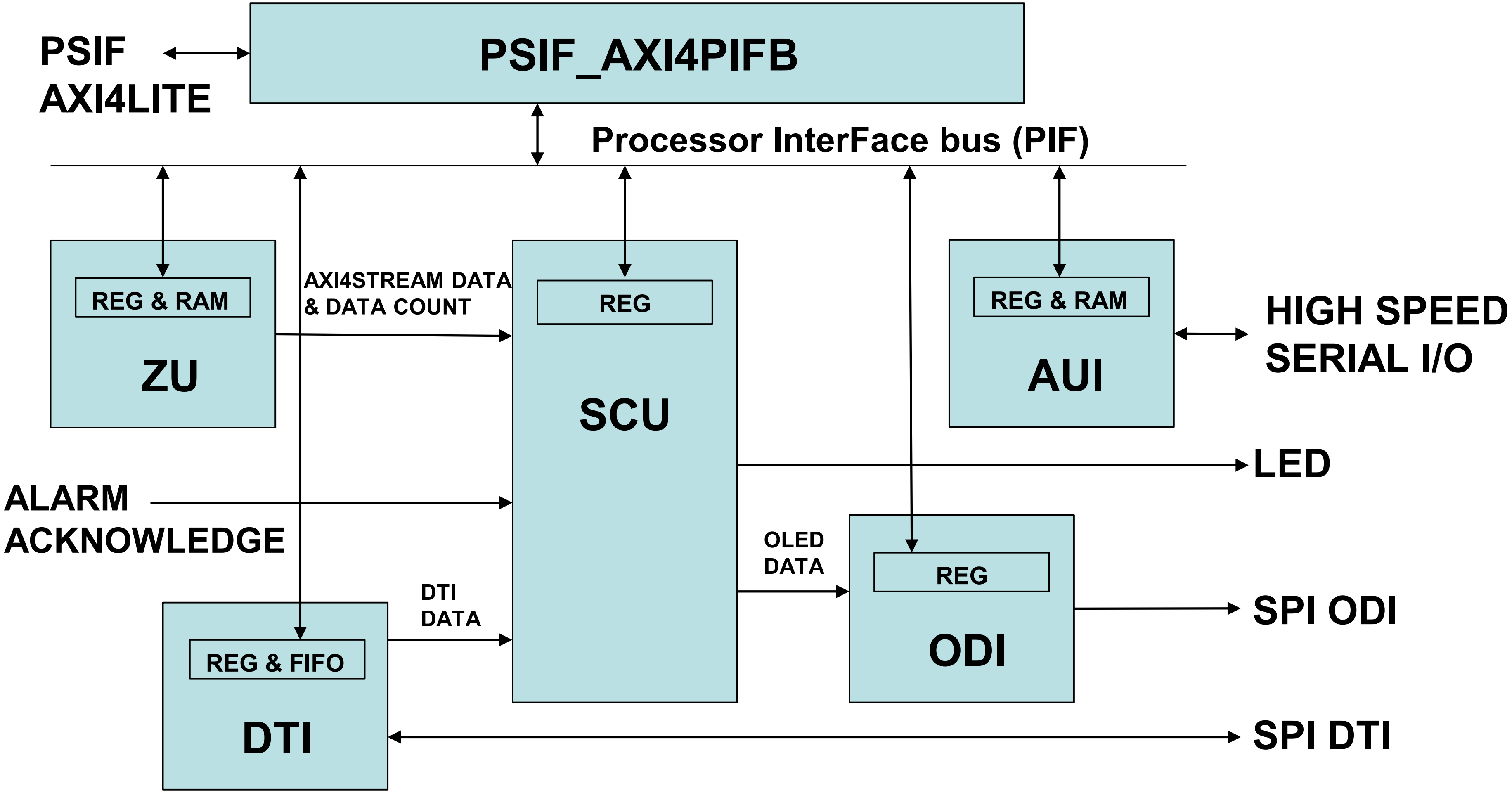
Introduction to lab 3; System on Chip Design

Introduction to lab 3: System on Chip Design;

Top level on ZEDBOARD Zynq7020



PL Core Modules



PL Core Modules cont.

- **PSIF_AXI4PIFB**: Processor interface bridge module for AXI4Lite interconnect to Processor shared Interface Bus (PIF).
- **ZU**: Decryption module that decrypts the AES128 encrypted messages from the processor system.
- **ODI**: **O**LED **D**isplay **I**nterface module transmits characters to OLED display on the ZEDBOARD.
- **DTI**: **D**igital **T**ermometer **I**nterface to MAX31723 PMOD module.
- **SCU**: **S**ystem **C**ontrol **U**nit receives AXI4Stream messages from ZU module, reads temperatures through DTI module SPI bus, and transmits the messages and temperatures to the ODI module.
- **AUI: Au**rora High Speed Serial **I**nterface module. This module transmits and receives data from the processor. It will only be used in simulation due to missing Gigabit Tranceivers in the Xilinx Zynq 7020 SoC used on the ZEDBOARD.

Lab3 tasks

- Integrate a given simple processor system and implement a dummy top level VHDL architecture for testbench.
- Learn about SystemVerilog/UVM SoC testbench environment and sequences.
- Implement UVM sequence to simulate data to ZEDBOARD OLED display.
- Implement design in Vivado project mode and non-project batch mode.
- Test design on the ZEDBOARD with the processor system.
- Use Xilinx SystemILA and ILA modules to observe internal FPGA signals.
- Integration of the HLS AES128 decryption IP module from lab1 in the ZU module.
- Implementation of the zu_fsm module VHDL RTL architecture of the given zu_fsm entity that controls the execution of the AES128 decryption module.
- Simulating the case case_psif_zu and verifying that the AXI4Stream data output of the ZU module are as expected.

Lab3 MLA design project directory structure

Classic directory structure

- Top level: top, core, packages, ip, vip, scripts and doc.
- In **top** directory: hdl, tb, tb_base, svim, libs (i.e. Questa libraries) and pr.
- In **core** directory: **hdl** for core level design files, module1, … , moduleN.
- For all module directories: **hdl** for design files, submodule1, .., submoduleN.
- Directories tb and svsim may be included in module directories for module simulation.
- svsim directory has a directory for each simulation case (i.e. case_xxx).
- top/pr directory has a directory for **project_pr** and for **non_project_pr**.
- Scripts directory has subdirectories for questa and vivado scripts
- Simulation and PR (Place&Route) may be done outside version control

Questa compilation

- The Questa compilation scripts are in directory scripts/questa.
- See text file *READ_ME_SIMULATION* for instructions.

- The command "./comp_all.tcl -help" returns the usage with options.

- To compile the complete design with testbench and test cases run the command "./comp_all.tcl ZEDBOARD SETUP_KBAXI4LITE“.

- To only compile the testbench and test cases run the command "./comp_all_tb.tcl ZEDBOARD SETUP_KBAXI4LITE“.

Questa simulation

- The Questa simulation scripts are in directory scripts/questa.
- See text file ***READ_ME_SIMULATION*** for instructions.

- User must select target equal ZEDBOARD (i.e. currently ZEDBOARD only available) and a test case with sequence.

  Usage: ./simulate.tcl <target> <test case> <sequence> [-gui] [-help]

- Example: "**$MLA_DESIGN/scripts/questa/simulate.tcl ZEDBOARD case_psif_reg psif_reg_test -gui«**

Questa simulation continue

- Simulation results including coverage data is written to the selected test case simulation directory:

  $MLA_DESIGN/top/svsim/case_<test>/<test>.sim.

- Example: See the log file "top/svsim/case_psif_reg/psif_reg_test.sim/**psif_reg_test.log**" after running the simulation above.

  Errors are written to the file

  "top/svsim/case_psif_reg/psif_reg_test.sim/**psif_reg_test_error.log**",

  and is empty when no errors have been found in the simulation.

- Perform all simulations in batch mode with ./simulate_all.tcl <target> [-help]

- Example "**$MLA_DESIGN/scripts/questa/simulate_all.tcl ZEDBOARD**"

Register access task using UVM Register Layer and Questa RUVM; Write

// Value of task **exitstatus**:

//   0: Register write done without error

//   1: Interface not found

//   2: Module in interface not found

//   3: Register in module not found

task automatic **regw**(output int **exitstatus**,

    input string moduleif, regmodule, register,

        int value);

Examples:

```
regw(es,"PSIF","ODI","ODI_PS_ACCESS_ENA",'h1);
regw(es,"PSIF","ODI","ODI_OLEDBYTE3_0",'h12345678);
```

Register access task using UVM Register Layer and Questa RUVM; Read

```
// Value of task exitstatus:
//   0: Register read done without error
//   1: Interface not found
//   2: Module in interface not found
//   3: Register in module not found
task automatic regr(output int exitstatus,
                    input string moduleif, regmodule, register,
                    output int value,
                    input string mirrorcheck = "ON");
```

Example:

```
int rxfifo_cnt;
regr(es,"PSIF",”AUI",”AUI_AURORA_PS_RXFIFO_COUNT",rxfifo_cnt,”OFF”);
while (rxfifo_cnt < 2) begin
  regr(es,"PSIF",”AUI",”AUI_AURORA_PS_RXFIFO_COUNT",rxfifo_cnt,”OFF”);
end;
```

Register access task using UVM Register Layer and Questa RUVM; Compare

```
// Value of task exitstatus:
//   0: Register compare done without error
//   1: Interface not found
//   2: Module in interface not found
//   3: Register in module not found
//   4: Register data read error
//   5: Register data read error, but with UVM mirror error suppressed
task automatic regc(output int exitstatus,
                    input string moduleif, regmodule, register,
                          logic [31:0] value,
                          string mirrorcheck = "ON",
                          string suppress_error = "OFF");
```

Examples:

```
        regc(es,"PSIF","ODI","ODI_PS_ACCESS_ENA",'h1);
        regc(es,"PSIF","ODI","ODI_OLEDBYTE3_0",'h12345678);
```

Register access task using UVM Register Layer and Questa RUVM; Poll for value

```
// Value of task exitstatus:
//   0: Register poll done without error
//   1: Interface not found
//   2: Module in interface not found
//   3: Register in module not found
//   4: Register data poll read error
//   5: Register data poll read error, but with UVM mirror error suppressed
task automatic regp(output int exitstatus,
                    input string moduleif, regmodule, register,
                        int value,
                        string mirrorcheck = "ON",
                        time POLLTIME= 1us,
                        time POLLTIMEOUT= 1000ms,
                        string suppress_error = "OFF");
```

Example:

```
       regp(es, "PSIF","ZU", "ZU_DONE_DECRYPTION", 'h1, "OFF");
```

RAM and FIFO access task using UVM Register Layer and Questa RUVM; Write

```
// Value of task exitstatus:
//   0: Memory write done without error
//   1: Interface not found
//   2: Memory in interface not found
task automatic memw(output int exitstatus,
                    input string moduleif, memory,
                         int value,
                         int offset);
```

Examples:

```
        memw(es, "PSIF", "ZUKEY", ’h12345678, ’h5);
        memw(es, "PSIF", "ZUKEY", keydata, address_offset);
```

RAM and FIFO access task using UVM Register Layer and Questa RUVM; Read

// Value of task **exitstatus**:

// 0: Memory read done without error

// 1: Interface not found

// 2: Memory in interface not found

task automatic **memr**(output int **exitstatus**,
    input string moduleif, memory,
    output int value,
    input int offset);

Examples:

int data, address_offset;

memr(es, "PSIF", "ZUKEY", **data**, ’h5);

memr(es, "PSIF", "ZUKEY", **data**, **address_offset**);

RAM and FIFO access task using UVM Register Layer and Questa RUVM; Compare

```
// Value of task exitstatus:
//   0: Memory check done without error
//   1: Interface not found
//   2: Memory in interface not found
//   3: Memory data read error
//   4: Memory data read error, but with UVM mirror error suppressed   task
automatic memc(output int exitstatus,
                              input string moduleif, memory,
                              int value, offset,
                              string suppress_error = "OFF");
```

Examples:

```
          memc(es, "PSIF", "ZUKEY", ’h12345678, ’h5);
          memc(es, "PSIF", "ZUKEY", keydata, address_offset);
```

Vivado batch mode Synthesis and P&R

- The Vivado non-project build scripts are in directory scripts/vivado and is executed from the build script build.tcl found in $MLA_DESIGN (i.e. design top level).

- For help: «**$MLA_DESIGN/build.tcl –help**»

- Example: «**$MLA_DESIGN/build.tcl ZEDBOARD SIMPLE**» build the design for ZEDBOARD with the SIMPLE processor block design.

- **NOTE:** Run the non-project build script on a copy of all the design files as described in lab3 task 4 to ensure that the Xilinx IP generated design files are not partially modified.