Description of Tasks for Hands-on Lab Assignments

The hands-on lab assignments, explained in the introductory lecture documents "Assignment_Intro_ES_LIACS.pdf" and "Assignment_Intro_ES_LIACS.mp4", are split and executed in four consecutive tasks.

TASK 1: Write sequential static affine nested-loop program (SANLP), using the C programming language, for the Sobel application. The written C program for Sobel must be compliant with the pnGen tool, i.e., it must obey the restrictions and programming rules, explained in "Assignment_Intro_ES_LIACS.pdf". **Generate Polyhedral Process Network (PPN) from the Sobel C program by using the pnGen tool of Daedalus!**

TASK 2: This task consists of several smaller sub-tasks:

1. Map and Simulate the generated PPN in timed SystemC for different PPN-to-MPSoC mappings. Create three MPSoCs, one with *p-2* processors, one with *p-1* processors, and one with *p* processors, where $p = n$ if $n < 7$, otherwise $p = 6$. NOTE: *n* is the number of functions in your Sobel *main()* C program.

   For each MPSoC create three different PPN-to-MPSoC mappings and simulate every mapping by generating a timed SystemC model and running a SystemC timed simulation!

2. Compare all simulated PPN-to-MPSoC mappings and choose the best mapping in terms of performance!

3. Write a report on **Task 1** and **Task 2** and send it to the teaching assistants (TAs) in PDF format by email. The report should contain:

   a. A header with the **ES labs group number** assigned to you and your group partner as well as **yours and your group partner names** and student numbers.

   b. A chapter for Task 1. Shortly describe your C program, your custom C functions, your techniques to exploit parallelism in the Sobel application, the generated PPN graph, and other details you consider important. You do not have to include your source code.

   c. A chapter for Task 2. Show and describe all the PPN-to-MPSoC mappings, you have created and simulated, and highlight the best mapping you have chosen. **Explain why the chosen mapping is the best, i.e., why it gives the best performance.** Please, try to write an elaborate explanation of the best mapping choice.

      In addition, please **describe your method to estimate the latency of every function you wrote and used in the Sobel *main()* program**. Show the estimated latency numbers that you have used during the simulations.

TASK 3: Synthesize the best PPN-to-MPSoC mapping obtained in Task 2. Generate a bitstream file for configuration of the Xilinx FPGA prototyping board and send it to the TAs. Since the bitstream file is very large, upload it on GoogleDrive and send an email to the TAs with a link to the file such that the TAs can download it.

TASK 4: Configure the Xilinx FPGA prototyping board with the bitstream file and check if the synthesized PPN-to-MPSoC mapping (multi-processor system) in Task 3 works properly. If it does not work then debug the system and fix the problem by correcting and repeating some of the steps in Task 2 and Task 3 until it works!

Tasks Schedule and Deliverables

The table below, indicates the start date of a Task, describes the deliverable for a Task, indicates the deliverable deadline, and provides references to supporting material/instructions needed to perform a Task.

NOTE: Every Task will start with a hands-on lab session moderated/supervised by the TAs!

Tasks	**Task Starts on**	**Deliverable Expected by**	**Deliverable**	**Material/Instructions (how to access them see "Assignment_Instr.pdf")**
Task 1	07.04.2025	14.04.2025	* Correctly working C program for Sobel compliant with pnGen * Correctly generated PPN graph *NOTE: Nothing to be sent to TAs, deliverable will be demonstrated by students and checked by TAs at the start session of Task 2!*	Intro to Hands-on lecture: File: "Assignment_Intro_ES_LIACS.pdf", see Sobel Edge Detection and how to write C programs compliant with pnGen Daedalus GUI manual: File "Daedalus_GUI_manual.pdf" see Sections 0.1 and 0.2
Task 2	14.04.2025	28.04.2025	* Report on Task 1 and Task 2, sent to TAs: s.r.siyadatzadeh@liacs.leidenuniv.nl	Daedalus GUI manual: File "Daedalus_GUI_manual.pdf" see Sections 0.3 and 0.4
Task 3	28.04.2025	12.05.2025	* Bitstream file, generated by the Xilinx tools and sent to TAs: s.r.siyadatzadeh@liacs.leidenuniv.nl *NOTE: Bitstream file will be tested on Xilinx FPGA board by TAs at the start session of Task 4!*	Daedalus GUI manual: File "Daedalus_GUI_manual.pdf" see Sections 0.5 Xilinx tools (XPS and XSDK) instructions: File "instructions_XPS_XSDK_command_line.pdf" (see all pages)
Task 4	12.05.2025	19.05.2025	* Correctly working bitstream file, generated by the Xilinx tools and sent to TAs: s.r.siyadatzadeh@liacs.leidenuniv.nl	Daedalus GUI manual: File "Daedalus_GUI_manual.pdf" see Sections 0.3, 0.4, and 0.5 Xilinx tools (XPS and XSDK) instructions: File "instructions_XPS_XSDK_command_line.pdf" (see all pages)