AARHUS
UNIVERSITY
SCHOOL OF ENGINEERING

Final Project in Hardware-Software codesign

Scope:
In this project you need to model, design, implement and analyze the performance of an alarm system using the HW/SW Co-design methodology. The assignment includes:

- Modeling of the system functionality using state machines
- Design of software and hardware, design space exploration and high-level synthesis of HW from SW.
- Integration of the software-based IP into the hardware model and analyzing the underlying performance.

Description:
We will use the alarm example seen in Weeks3-6. The alarm system consists of two sensors (S_1, S_2) and one camera (C) to observe the environment. The data being collected by sensors/camera is processed in real-time by the alarm system to decide whether there is an intrusion or not.

- The alarm system can be either **Inactive**, **Active** or **Alarmed**.
- Initially the system is in Inactive. It switches to Active state upon receiving a **valid pin code**.
- A pin code **P** can be an integer generated as a random number <10
- The pin code checking function **valid(P)** is a predicate that returns either *true* if **P** is even or *false* if **P** is odd.
- When system active, sensors sample data every 60 ms.
- When system active, camera samples data every 120 ms.
- The data returned by a sensor **v(S**$_i$**)** is a random integer between 0 and 5.
- The data produced by each camera reading **v(C)** is an MATRIX of 9*9 elements. Each element of the array **v(C[I,j])** is a random integer between 0 and 9.
- Intrusion checking is performed by computing **I_**$_m$(I,j)= (v(S_1)+v(S_2))* v(C[I,j]). If the sum of the 81 elements of **I_**$_m$(i,j) is greater or equal to **6290**, then there is an intrusion.
- Once an intrusion is detected, the sampling stops and an alarm is triggered (ON) upon which the system transits to Alarmed state.
- The system remains in Alarmed state until either a **valid pin code** is entered or the **duration of 10s** expires.
- After an alarm detection is handled, the system restarts (by just returning to Active state).

Assignment:

- Design individual state machines for sensor, camera and the overall alarm system.
- Make a sequence diagram to describe the system dynamics.
- Design a system architecture: objects, modules, components, threads and how they interact with each other. Argue your choices
- Implement the alarm system in SystemC or C++ or combine both SystemC and C++
- For time, you can use real clock time or just simulated time that you advance by yourself.
- Using Vivado HLS, generate an IP core for the SW implementation where the max response time (latency) is 32.

- Explore different design alternatives (by playing with shared variables and functions placing- W5-6, and also using pipelining and unrolling directives W9-10) and generate at least two different IP cores.
- Design a simple HW model using Vivado Lab. You can get inspiration from the Lab1 example available in W7-8 but without the ILA module.
- Integrate the SW-based IP cores into the HW model, one at a time and connect the port to read the pin code (SW) to AXI of peripheral. This results in 2 hardware platforms, having the same hardware model but only SW-IP core differs.
- Run solutions simulation and compare the two obtained solutions in terms of response time, used HW, energy, size and latency of IP cores.
- How such solutions can be improved? (ideas)

- ➢ Make your report as concise as possible.
- ➢ Reports should not exceed 15 pages
- ➢ In your report, you can put code snippets and screenshots from Vivado tools.
- ➢ The submission includes Report, SW code and Vivado models.
- ➢ The entire material should be submitted as a single Zip file.

Hints:

- C++ rand function cannot be synthesized directly by Vivado so you either implement yours or use a randseed function.
- Not all C++ time functions are synthesizable with Vivado, feel free to use simulated time if you cannot utilize real (machine) time or read clock cycles and convert to actual time.