COMP.CE.400

System Design

Final Report

Group 13

Santeri Laurila (274301)

santeri.laurila@tuni.fi

Nuutti Mikkonen (252844)

nuutti.mikkonen@tuni.fi

# Overview of the System

*Describe here what kind of system you implemented in the exercise project. What does it do and what are its main parts and features? Focus only on the implementation on the FPGA board, not the design process, like SystemC simulations. You may illustrate the description with an image, if you wish.*

The target system had following sub-parts: camera and video encoding (Kvazaar), video streaming and playback setup. Video encoder used was an open source HEVC encoder called Kvazaar, developed at Tampere University. We used design tools such as Kactus2. Camera was connected to FPGA board, video encoding was done on SoC-FPGA board, video decoding and playback was done on PC. Compressed video was streamed over Ethernet to PC.

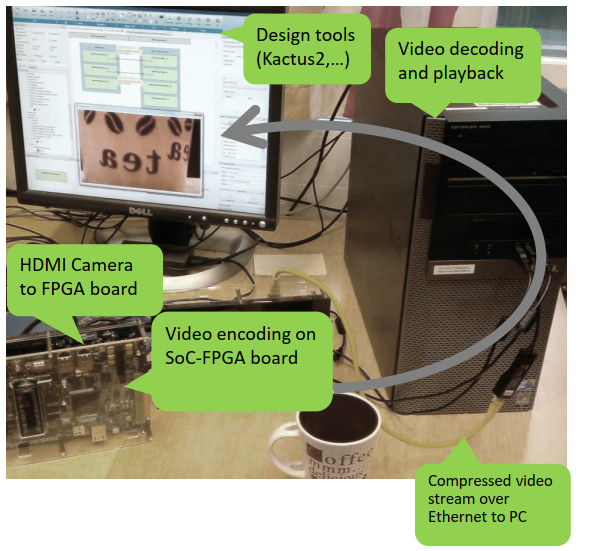


Fig. 1 Overview of the system [1]

During platform modeling part we benchmarked the relevant IO operations, found out the relevant specifications and built ARM build of Kvazaar.

## Hardware

*Describe here the hardware components of the system. Obviously, this includes the FPGA board, its add-ons and the PC, but more importantly you should describe what resources (processors, FPGAs, memories, HW accelerators, buses etc. if any) are used on the FPGA board and how they communicate with each other.*

The project was implemented on a CycloneV SoC development kit with a hard ARM core and FPGA fabric. The FPGA was used to implement hardware acceleration of intra prediction. The hard processor communicated with memories with its DDR controller and the FPGA used a DMA through an AXI bridge to use the same controller.

Camera was connected to the FPGA with an HSMC connector. The development board was connected to a PC with an Ethernet connection, through which the encoded video was streamed.

## Software

*What software modules are used by the system and what is the purpose of each one? This includes programs and processes that are running when the system is on. How are they mapped on the hardware resources and how do they communicate with each other?*

The hard ARM processor runs Kvazaar that is hardware accelerated. Kvazaar gets its input from the camera. To run the camera the operating system has to be loaded with a camera driver kernel module, and to use the hardware accelerator a kernel module for it must be loaded. These modules allow the software to access hardware resources.

The encoded video from Kvazaar is transferred to PC with ffmpeg and played back on the PC with ffplay.

# Summary of the Exercises

*Write here a short summary of what you did in each of the individual exercises. Emphasize the big picture: what was the purpose of the exercise, what tools you used and what you did with them, and what was the end result. A couple of paragraphs per exercise is sufficient. The purpose of this section is to ensure that you have a good general picture of the exercise project as a whole.*

## Exercise 1

Exercise 1 was introduction to Kvazaar video encoder and Kactus2 design tool. During the exercise we also got familiar with the course infrastructure. Kactus2 is a specific ESL (Electronic system level) tool that packs the design content in a tool and design language independent way. Kactus2 is based on IP-XACT XML metadata standard.

During the exercise we profiled Kvazaar performance with gprof profiler and visualized the results with picture that showed Kvazaars functions. After that we discussed if we chose to use HW acceleration, which function we would choose for that. **kvz\_angular\_pred\_generic** used most time so we chose that**.**

## Exercise 2

In exercise 2 we were introduced to SystemC. SystemC is used for modelling systems with an event driven interface. We did model and simulation using SystemC. Then we visualized the results using GTKWave.

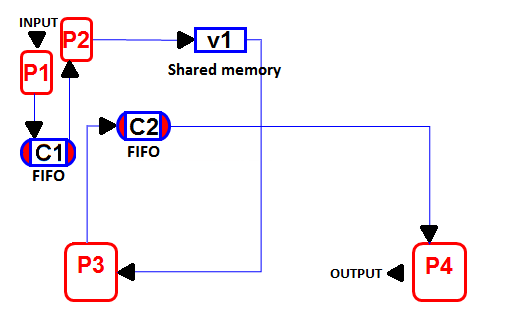


Fig. 2 System implemented [2]

Lastly, we optimized the system’s performance by adjusting the mapping of the hardware and with changing memory access.

## Exercise 3

The purpose of this exercise was to simulate transfers between the hardware accelerator and the processor. SystemC was used for this, as its TLM socket is a suitable tool for simulating the transfers. In this version of Kvazaar, intra prediction was separated from the rest of the functionality and it used interrupts and the SystemC sockets to send and receive data from the rest of the encoder program.

The results concluded that Kvazaar could be hardware accelerated while providing same encoding results. The accelerator was also split into 4 parts to prove that even greater parallelism was possible.

## Exercise 4

In this exercise the hardware platform was first used. The purpose was to create hardware configurations and access them with software on the HPS. A simple LED blinker was the target. For this, the FPGA controlled the LEDs and was connected to the HPS with an AXI bridge. The HPS could read and write to the FPGA pins by mapping them to the LWH2F with the use of BlinkerHAL.

The result was a working system where pressing buttons would light up LEDs. This confirmed that the FPGA and HPS could communicate between each other and this connection could be used for splitting Kvazaar to include hardware acceleration.

## Exercise 5

In exercise 5 the project platform was benchmarked and early estimates for final results were made. Memory operations from user and kernel space were measured, however these numbers were not yet used for estimations. The MIPS of the FPGA and HPS were calculated. For the HPS, it was also measured as much lower than the hypothetical value.

A theoretical frame rate was calculated for a hardware accelerated and software-only versions of Kvazaar for the ARM. The software version was also executed on the development board, however the FPS was much lower than what was expected. The calculations hadn’t considered the memory transfers and assumed the memory would not be the bottleneck in the system.

## Exercise 6

Topic of Exercise 6 was TLM (Transaction level modeling) and Design Space Exploration. In this exercise we matched the performance of Kvazaar on Terasic Cyclone V Veek board with SystemC simulation. We also explored Kvazaar design space exploration with SystemC simulation.

We also evaluated how many ARM cores would be needed to achieve real time encoding (25 fps) with FullHD video. We came to conclusion that 95 ARM cores would be needed.

## Exercise 7

In exercise 7 we created a HW design with intra prediction HW accelerator and camera control. Then we used the HW accelerator driver with Kvazaar and used the camera driver with an userspace application.

The purpose was to design and verify that the HW design of the intra prediction accelerator works without losing quality compared to software only Kvazaar, and to ensure the kernel modules for the drivers were functional. Kvazaar produced similar results to previous ones and the video saved from the camera worked.

## Exercise 8

In exercise 8 we did the complete streaming HEVC encoder setup and demonstrated it to course staff. In the exercise we profiled the hardware accelerated encoder and streamed the video from Cyclone V board to PC.

First, we did profiling build for HW accelerated Kvazaar, then we streamed files from FPGA board to PC using ffmpeg. Then we did the same using Kvazaar and measured the system performance by inspecting delay.

# Suggestions and Feedback

*Give us some suggestions on how we could improve the course and the exercise project. Even this section will affect the grade for this exercise so be thoughtful and constructive.*

## What Did We Learn

*What did you learn on the course as a whole and on the exercise project in particular? Did you learn everything you expected from the course? Were some topics missing?*

We learned everything from this course that we thought we would. No topics were missed on this course.

## What was Easy/Hard

*Which parts of the exercise project did you find the easiest? On the other hand, what was hard and time-consuming?*

Exercises 2 and 7 were a lot more time consuming than others. Other exercises were easy or pleasant to work on.

## What Needs Improvement

*How would you improve the course and the exercise project?*

Sometimes instructions were somewhat hard to follow. Files to be returned should be at bottom of exercise description, since we missed it at some exercises where returns were told at the beginning.

# References

**[1]** COMP.CE.400 Course material, Lecture 0: Getting Started With the Course, <https://plus.tuni.fi/comp.ce.400/spring-2021/lec/lec0/>

**[2]** COMP.CE.400 Course material, Exercise 2 material, <https://plus.tuni.fi/comp.ce.400/spring-2021/ex2/ex2_1/>