BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIOENGINEERING
ÚSTAV RADIOELEKTRONIKY

THERMAL CAMERA FOR BIOLOGICAL APPLICATIONS
TERMÁLNÍ KAMERA PRO BIOLOGICKÉ APLIKACE

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR — **Bc. Jáchym Macura**
AUTOR PRÁCE

SUPERVISOR — **Ing. Michal Kubíček, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2020

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Master's Thesis

Master's study field **Electronics and Communication**

Department of Radioengineering

Student: Bc. Jáchym Macura ***ID:*** 164327

Year of study: 2 ***Academic year:*** 2019/20

TITLE OF THESIS:

Thermal camera for biological applications

INSTRUCTION:

Study properties and get acquainted with analog thermal detector BIRD17-XGA. Design a camera based on this sensor. The primary purpose of the camera is monitoring of activity of plat's stomata. The detector should be controlled using an FPGA and will utilize standard camera subsystems used within PSI company. To control the camera create suitable HDL IP cores for target FPGA. To complete the design write also basic software modules for Linux operating system, including TCP server for communication with a PC. Integrate the software into PSI company internal generic camera platform and verify its functionality.

RECOMMENDED LITERATURE:

[1] VOLLMER, Michael a Klaus-Peter MO̎LLMANN. Infrared thermal imaging: fundamentals, research and applications. Weinheim: Wiley-VCH, c2010. ISBN 3527407170.

[2] UNSALAN, Cem a Bora TAR. Digital system design with FPGA: implementation using Verilog and VHDL / Cem Ünsaln, Yeditepe University, Bora Tar, The Ohio State University. New York, NY: McGraw-Hill Education, [2017]. ISBN 9781259837906.

Date of project specification: 3.2.2020 ***Deadline for submission:*** 28.5.2020

Supervisor: Ing. Michal Kubíček, Ph.D.

prof. Ing. Tomáš Kratochvíl, Ph.D.
Subject Council chairman

ABSTRACT

This master thesis describes the process of designing a thermal camera for biological applications. The first part describes infrared radiation, also known as thermal radiation. It goes on to describe an emissivity of objects, which plays an important role in determining the temperature of the object. The end of first chapter overviews the types of thermal detectors and their basic properties. The second chapter briefly introduces the basic hardware components of the developed camera. The last chapter describes the process of software development. This includes the development of IP cores, kernel modules and C++ libraries.

KEYWORDS

Thermal camera, VHDL, Kernel module, C++, Qt

ABSTRAKT

Diplomová práce se zabývá návrhem termální kamery pro biologické aplikace. V první části je popsáné infračervené záření, také známé jako thermální záření. Pokračuje popisem emisivity objektů, která hraje důležitou roli při stanovení teploty snímaného objektu. Dále je v první kapitole uveden přehled typů termálních detektorů a jejich základní vlastnosti. Druhá kapitola stručně seznamuje se základními hardwarovými komponenty vyvíjené kamery. V poslední kapitole je popsán celý proces vývoje softwaru. Ten zahrnuje vývoj IP jader, kernel modulů a C++ knihoven.

KLÍČOVÁ SLOVA

Termální kamera, VHDL, Kernel modul, C++, Qt

MACURA, Jáchym. *Thermal camera for biological applications*. Brno, 2019/2020, 80 p. Master's Thesis. Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Advised by Ing. Michal Kubíček, Ph.D.

DECLARATION

I declare that I have written the Master's Thesis titled "Thermal camera for biological applications" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author I furthermore declare that, with respect to the creation of this Master's Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno ....

author's signature

Contents

List of Figures

List of Tables

Listings

INTRODUCTION

Infrared thermal imaging, also called thermography is applied in research and development such as nondestructive testing, predictive maintenance and in many more areas.

This master thesis goal is a design of a thermal camera for non-invasive measurement in biological application. First chapter of this thesis is focused on describing of the infrared radiation and detailed back-ground information about important parameters like emissivity. Next part of this chapter describes thermovision and its use in the real world. Thermovision topic also includes division and description of thermal detectors and their characteristic parameters.

The second chapter is devoted to get reader familiar with the individual electronic components of developed thermal camera. The hardware of the camera was supplied by the contracting company Photon Systems Instruments (PSI). Most attention is devoted to the heart of the camera which is XGA-Bird17 - an infrared bolometric detector and the brain, which is SoC Xilinx Zynq.

The last chapter called Software describes step by step the development of a camera control system. The chapter is divided into following parts:

- Ensuring high data throughput for video stream between Zynq programmable logic and processing system.
- Interfacing with high speed external components.
- Temperature stabilization of XGA-Bird17 detector.
- Image quality improving by Correlated Double-Sampling correction, Non-Uniformity Correction and Bad Pixels Replacement.
- Graphical User Interface for TCP communication with the camera.

1 INFRARED RADIATION AND THERMAL DETECTORS

This chapter describes the basics of InfraRed (IR) and thermal radiation. This chapter continues acquainting the reader with cooled and uncooled thermal detectors and their parameters.

1.1 Infrared radiation

IR radiation or IR light is type of an energy that is invisible to the human eyes but that can be felt as heat. IR is type of the electromagnetic (EM) radiation same as (lowest to highest wavelength) gamma-rays, X-rays, ultraviolet radiation, visible light, infrared radiation, microwaves and radio waves. Together, these types of radiation make up the electromagnetic spectrum which can be described as particles, or more common, as EM waves. The wavelength (λ) of IR spectrum is in range from 0.78 µm to 1 mm. With known speed of propagation (c[1] = 299792458 m/s) the frequency (f) in Hertz (Hz) can be calculated using Eq.1.1.

$$f = \frac{\lambda}{\mathrm{c}} \tag{1.1}$$

The frequency domain, boundary of IR spectrum is approximately from 300 GHz to 400 THz. From this huge bandwidth only a small range is used for thermal imaging. Typically, three spectral ranges (Tab.1.1) are defined for thermal imaging (thermography).

Tab. 1.1: Infrared spectral ranges

Spectral range	**Short**	**Wavelength [µm]**
Short-wave	SWIR	0.9 - 1.7
Mid-wave	MWIR	3 - 5
Long-wave	LWIR	8 - 14

[1]Speed of propagation depends on index of refraction but in this text it is assumed as a constant.

1.1.1 Thermal Radiation

Every object with temperature higher than absolute zero emits thermal radiation. Radiation power that can be emitted depends on the temperature of the object and on the wavelength (Planck's law - more description in Black-Body section). The element with radiation surface dA generates total energy flux dϕ, also called radiant power M [Wm^{-2}] (Eq.1.2).

$$M = \frac{d\phi}{dA} \tag{1.2}$$

.

Eq.1.2 is simplified by using total derivatives of all emitted wavelengths and radiant angles. Resulting sum of all wavelength energies for Eq.1.2 is shown in Figure 1.1.

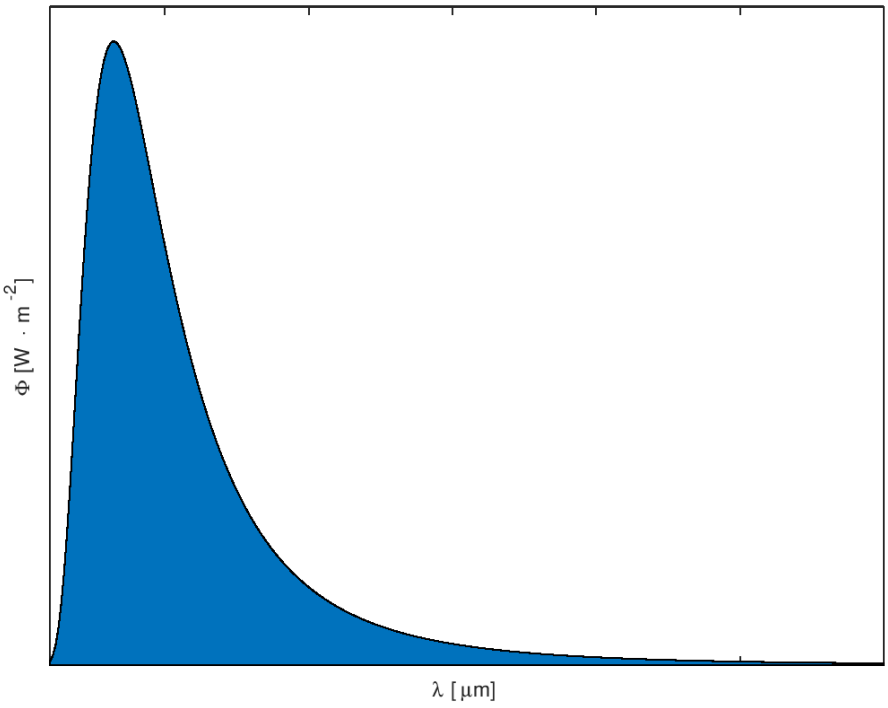

Fig. 1.1: Total radiant power

1.1.2 Black-Body

Black-Body is a theoretical perfect emitter of thermal radiation, that is able to emit the maximum radiant power which depends only on the body temperature. For the object to be called Black-Body, it must meet these conditions:

- Black-body has to absorb all incident radiation, regardless on wavelength or direction.
- No object can emit more radiation than Black-Body with the same temperature and wavelength.
- Emitted radiation of Black-Body depends only on wavelength but not on the direction.

Planck's law, Wien's law and Stefan–Boltzmann law

Planck's law describes the spectral density of EM radiation emitted by the Black-Body, when there is no net flow of energy between the body and its environment. Planck's law Eq.1.3 defines the intensity of radiation for the Black-Body as a function of temperature and wavelength.

$$M_\lambda(T)d\lambda = \frac{2\pi hc^2}{\lambda^5} \cdot \frac{1}{e^{hc/(\lambda kT)} - 1} d\lambda \tag{1.3}$$

.

where $h = 6.626 \cdot 10^{-34}$ [J · s] is the Planck's constant, c [m/s] is the speed of light in vacuum, λ [m] is the wavelength of the radiation, T [K] is the absolute temperature of the Black-Body and $k = 1.38 \cdot 10^{-23}$ [J · K^{-1}] is the Boltzmann constant.

Figure 1.2 shows plot of several curves depending on the wavelength. Each curve represents the energy distribution at different temperature. Based on Fig.1.2, 2 rules can be derived:

- If temperature increases, the intensity of radiation increases.
- If the temperature of an object increases, the peak intensity moves to shorter wavelengths.

With usage of these rules, we can observe this phenomenon by slowly heating a piece of metal. At room temperature, the metal does not emit any light visible to the human eye. However, if it's heated to approximately 500°C, it begins to glow red. At 500°C, the metal emits visible light primarily in the red portion of the visible spectrum. As the temperature of the metal increases to approximately 1500°C, it begins to glow with white color. At 1500°C, the metal emits energy at all visible wavelengths. White light is the combination of all visible colors.[3]

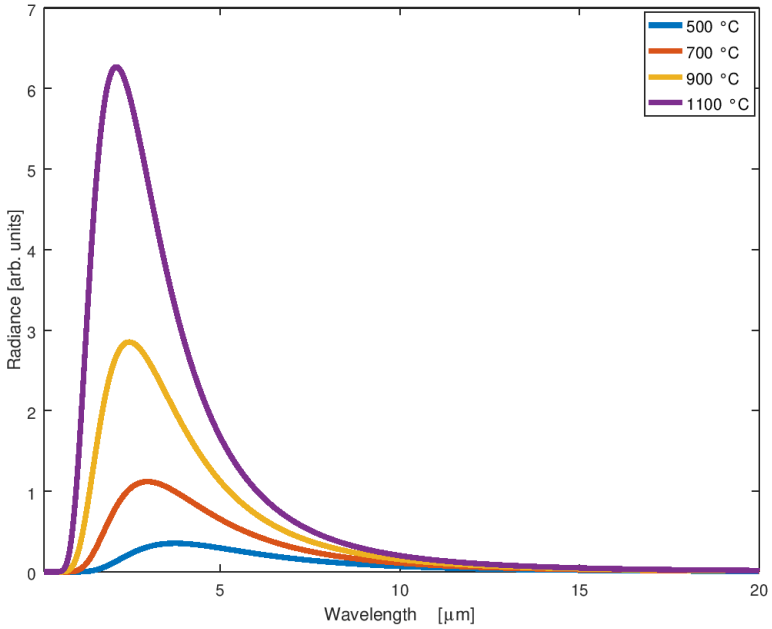

Fig. 1.2: Radiance of the Black-Body

With condition $\mathrm{d}M_\lambda(T)/\mathrm{d}\lambda = 0$, it's possible to locate wavelength with maximal intensity. This leads to Wien's displacement law Eq.1.4:

$$\lambda_{peak} = \frac{b}{T} \tag{1.4}$$

Where T [K] is absolute temperature, b is Wien's displacement constant, equal to 2898 [µm · K].

The excitance of the Black-Body source can be calculated from Eq.1.5, that is so called Stefan–Boltzmann law.

$$M(T) = \int_0^\infty M_\lambda(T) d\lambda = \sigma T^4 \tag{1.5}$$

Where $\sigma = 5.69 \cdot 10^{-8}$ $[Wm^{-2}K^{-4}]$ is the Stefan-Boltzmann constant and T $[K]$ is temperature. Final emissive power is equal to the area in Fig.1.1.

In the real world, IR imagining never detects the whole spectrum, but only predefined spectral range, which is determined by detector. The integral of Eq.1.5 doesn't have analytic solution for arbitrary values of lower and upper limits. To simplify the results, band radiation can be obtained as the fraction of the Black-Body radiation in the interval from 0 to λ, compared to the total emission from 0 to ∞. The mathematical analysis shows that the integral only depends on the parameter $\lambda \cdot T$ and therefore integrals can be evaluated numerically for this parameter and

tabulated as a function of λT. The Black-Body radiation in wavelength interval 0 to λ as function of λT is in Figure 1.3a.

For quick power estimation, simple graphical method can be used: For example, if the used detector works in LWIR band (8 - 14 µm) and we are interested in temperature 500 Kelvin. Get values for 8 µ$m \cdot 500\ K$ and 14 µ$m \cdot 500\ K$ from the graph (dashed lines in Figure 1.3b) and then put them into Eq.1.6. The resulting value (Fig.1.3b blue area) is a very good estimation of the band energy. Table 1.2 shows result for two temperatures and all IR spectral ranges.

$$F_{(\lambda_1 \to \lambda_2)} = F_{(\lambda_0 \to \lambda_2)} - F_{(\lambda_0 \to \lambda_1)} \tag{1.6}$$

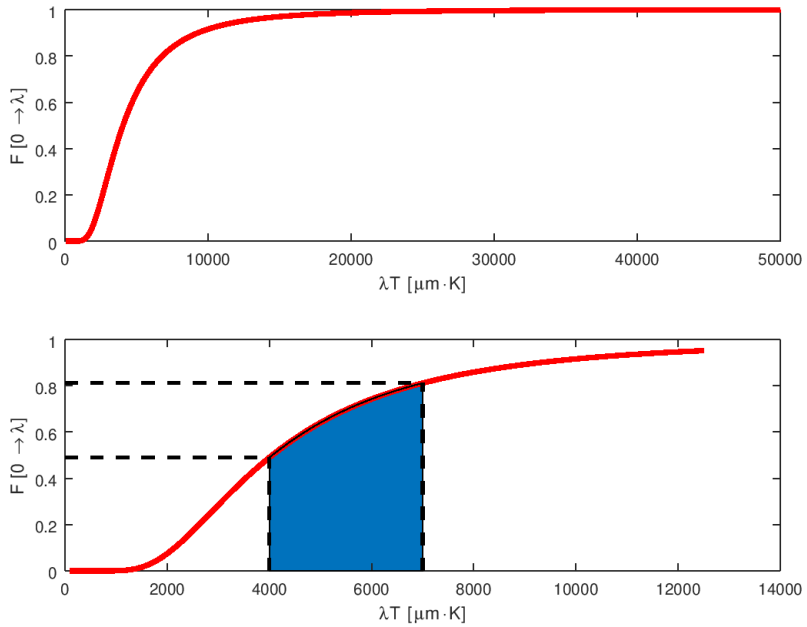

Fig. 1.3: Black-Body's radiation as function of λT

Tab. 1.2: Example of band emission

Temperature [K]	**SWIR [-]**	**MWIR [-]**	**LWIR [-]**
500	$4.2 \cdot 10^{-5}$	$15.6 \cdot 10^{-2}$	$32.1 \cdot 10^{-2}$
2000	$37.1 \cdot 10^{-2}$	$10.1 \cdot 10^{-2}$	$0.92 \cdot 10^{-2}$

1.1.3 Emissivity

Real objects can't emit the maximal thermal radiation at given temperature as the Black-Body. Thermal radiation of real objects can be easily computed by multiplying the Black-Body radiation by the emissivity ε. Emissivity is a ratio of emited radiation from the surface of real objects and Black-Body radiation at the same temperature. Emissivity is influenced by wavelength, temperature and solid angle. But IR cameras work in predefined wavelength ranges with limited dynamic range temperature of the scene. In practice, averaged emissivity is mostly used over the desire wavelength and temperature range. The object with constant emissivity over all wavelengths is called Gray-Body. Emissivity can have a value between $0 \leq \varepsilon \leq 1$.

Kirchhoff's Law

Kirchhoff's Law expresses that the amount of radiation absorbed by any object is equal to the amount of radiation that is emitted by this object Eq.1.7.

$$\varepsilon = \alpha \tag{1.7}$$

Where ε [-] is the emissivity and α [-] denotes the fraction of radiation that is either emitted or absorbed. This leads to Eq.1.8 which says: Radiation (ϕ_0) incident on any object must be sum of reflected (ϕ_R), transmitted through the object (ϕ_T) and absorbed (ϕ_A) radiation.

$$\phi_0 = \phi_R + \phi_T + \phi_A \tag{1.8}$$

Eq.1.8 may be rewritten as Eq.1.9.

$$1 = R + T + \alpha \tag{1.9}$$

Where R is reflected radiation and T transmitted radiation. For opaque solids is T = 0 and in this case combining Eqs. 1.7 and 1.9 allows to express the emissivity as Eq.1.10.

$$\varepsilon = 1 - R \tag{1.10}$$

It is important to note that R includes not only directed reflectivity as is usual for polished surface, but also diffusive reflectivity, which additionally occurs for rough surfaces. Highly polished surfaces can reduce emissivity, and on the other hand, objects with rough surface can increase emissivity. Table 1.3 lists materials and their emissivity (whole table can be found in [2]).

Tab. 1.3: Material emissivity [2]

Material	**Emissivity value [-]**
Aluminum: Polished	0.05 - 0.10
Aluminum: Roughly polished	0.18
Copper: Polished	0.02 - 0.10
Copper: Rough	0.74
Foods	0.85 - 1.00
Glass: Plate	0.90 - 0.95

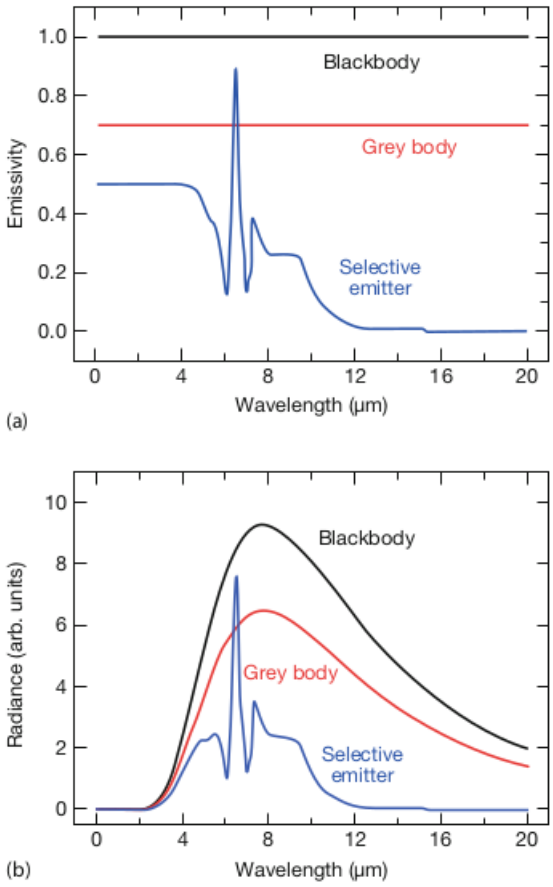

Fig. 1.4: (a) Spectral emissivity and (b) spectral radiation of object with different emissivity[1]

1.2 Thermovision and IR Detectors

1.2.1 Thermovision

Thermovision is a method of non-contact detection of IR radiation. An IR camera can therefore register the thermal radiation emitted by physical bodies at temperatures of about -40 °C to about +2,000 °C. Typically, thermal imaging cameras record a range from -20 °C to +250 °C. The great advantage of the thermovision method is its lack of invasiveness. In addition, it is possible to conduct temperature measurements in hard-to-reach places. With the use of thermovision methods, one can be quickly diagnose the correct functioning of electrical and electronic components, including industrial automation systems, the uniformity of heating of objects such as machines, furnaces, etc.[9]

There are two methods for IR thermovision:

Active - needed source of the IR light to produce a thermal contrast between the feature of interest and the background. The active approach is necessary in many cases, given that the inspected parts are usually in equilibrium with the surroundings.

Passive - temperature of the object is higher or lower than temperature of the background, which naturally makes the contrast.

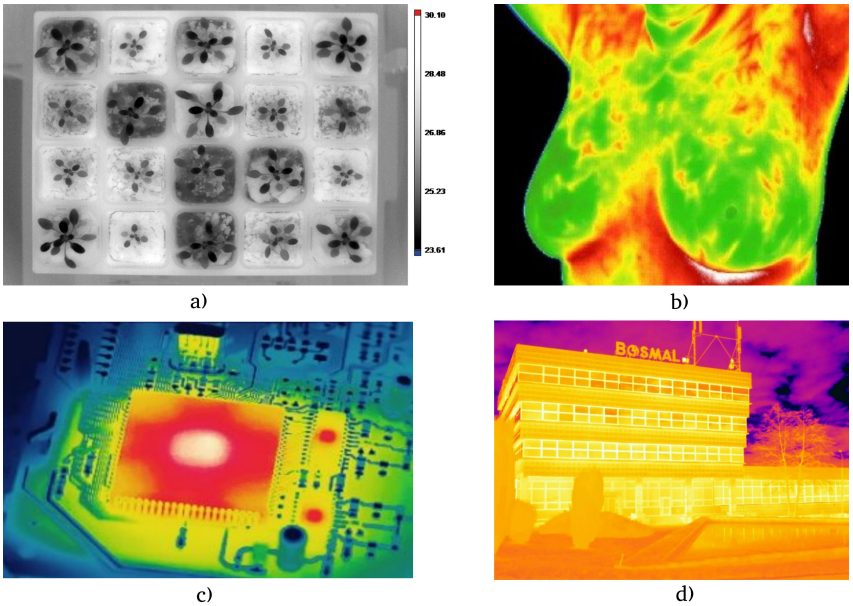

Fig. 1.5: Examples of Thermovision: a) leaf and plant temperature as indicator of water-use efficiency[10], b) breast cancer detection[11], c) PCB inspection[12], d) thermal insulation of the house[9]

1.2.2 IR Detectors

Every IR detector is a transducer that converts thermal radiation into electric signals. IR detectors can be divided into two main groups. First IR detector type is called photon detector, also known as cooled detector, and second type is thermal detector, known as uncooled detector.

1.2.3 Photon Detectors

Photon detectors work on the basis of an intrinsic photoelectric effect. Photon detectors are made of material which responds to IR by photon absorbing that moves material electrons to a higher energy state, which leads to voltage, conductivity or current changes. At room temperature, electrons at different energy levels and electrons in conduction band which are free to move can conduct an electrical current. Most of the electrons are found in the valence band, where they do not carry any current because they cannot move freely. If the material is cooled to low-enough temperature (operating temperature $\approx$ 77K), the energy of the electrons may be so low that there isn't any left to move the electrons. In this state, the material is not able to carry current. Exposing this cooled material to incident photons having sufficient energy $E = h \cdot v$ (E is photon energy, h is the constant of proportionality, v is frequency), exceeding the energy threshold (Δ E), stimulates the electrons in the valence band, causing them to move up thru energy gap into the conduction band. Thus, the cooled IR detector can carry a photocurrent, which is proportional to the intensity of the incident radiation. There is a critical value of energy of the incident photons that will allow an electron to jump from the valence band into the conduction band. This energy is related to a certain wavelength, the cut-off wavelength. Since photon energy is inversely proportional to its wavelength, the energies are higher in the SWIR/MWIR band than in the LWIR band. Therefore, the operating temperatures for LWIR detectors are lower than for SWIR/MWIR detectors.

Intrinsic detectors

Photovoltaic intrinsic detectors structure is based on PN junction devices. The reflective coating on the bottom of the detector provides double chances (injection and reflection) of the photon absorption. Under IR radiation, the potential barrier of the PN junction leads to the photovoltaic effect. An incident photon with the energy greater than the energy band gap of the junction generates electron-hole pairs and the photocurrent is excited.[16]

Extrinsic detectors

Extrinsic detectors are similar to intrinsic detectors. However, in extrinsic detectors carriers are excited from the impurity levels and not over the bandgap of the basic materials. They have the advantage of being able to operate at much lower wavelengths than intrinsic. To overcome problems with positive holes left behind in the valence band, higher doping density is required.[16]

Quantum well infrared photodetector

Quantum Well Infrared Photodetector (QWIP) uses electronic intersubband transitions in quantum wells to absorb photons. In order to be used for infrared detection, the parameters of the quantum wells in the quantum well infrared photodetector are adjusted so that the energy difference between its first and second quantized states match the incoming infrared photon energy.

1.2.4 Thermal Detectors

Thermal detectors convert incident EM radiation into thermal energy, which leads to changes in physical properties, that are converted into electrical signal with thermal-electric convertor. The utilized type of thermal-electric defines the uncooled detector type. Since the operation of thermal detectors involves a change in temperature, they have an inherently slow response and a relatively low sensitivity compared to photon detectors. Big advantage over photon detectors is the working temperature which is usually equal to room temperature (25 °C). But even if they are called uncooled detectors, they need some type of cooling mechanism for noise and non-uniformity reduction.

Thermocouples

Thermocouples are formed by connecting two dissimilar metals that generate voltage at the junction point. Generated voltage is proportional to the temperature of the junction.[16]

Pyroelectric detectors

Pyroelectric detectors consist of a polarized material which, when subjected to changes in temperature, changes the polarization. The fluctuation in the exposure to the scene generates a corresponding fluctuation in polarization and thus an alternating current that can be monitored with an external amplifier.[16]

Bolometers

Bolometers are based on the same principle as thermistors. The resistance of the elements varies with temperature changes. Unlike thermocouples or pyroelectric detectors, bolometers are passive elements and for proper function they need a source of power.[16]

1.3 Parameters That Characterize Detector Performance

1.3.1 Responsivity

The responsivity of an IR detector is defined as a ratio of the change of the received signal to to the change of the temperature scenery (Eq.1.11).

$$Responsivity_{ij} = \frac{S_{1,ij} - S_{2,ij}}{T_{1,ij} - T_{2,ij}} \tag{1.11}$$

where $S_{x,ij}$ is the received signal facing the target at Tx.

1.3.2 NETD

Noise Equivalent Temperature Difference (NETD) is the smallest detectable temperature change in the target. All temperature differences smaller than the NETD, would be smaller than the noise of the detector, thus impossible to be detected.

$$NETD = \frac{noise(T_{bb}) \cdot \Delta T_{bb}}{S_2(T_{bb} + \Delta T_{bb}/2) - S_1(T_{bb} - \Delta T_{bb}/2)} \tag{1.12}$$

where S_x is the received signal and T_{bb} is the Black-Body temperature.

1.3.3 Defective Pixels

Defective pixels or dead pixels are pixels that fail to operate properly when excitation is applied during the video sampling. Dead pixels must be repaired and that increases computational complexity. Big clusters of dead pixels are impossible to repair using the neighbour samples.

2 Hardware

This chapter introduces the main hardware parts of the evolves thermal camera (Fig.2.1). Schematics and PCB designs are not included because it was already done with engineers of Photon Systems Instruments and it is not part of this thesis. The main part of the thermal camera is the BIRD17-XGA IR detector. This detector has 4 analog outputs for video signal, which are converted to digital signals with 4 parallel Analog-Digital-Converter (ADC) LTC2387-16 [17] (Low noise, high speed ADC with 16 bit resolution and 15 MHz sample frequency). Another ADC is AD7680 [18] which is used to digitize the analog output of the thermal diode for BIRD17-XGA temperature measurement. Digital interface and all signal and data processing is done by Zynq7000 on MicroZed board [8].

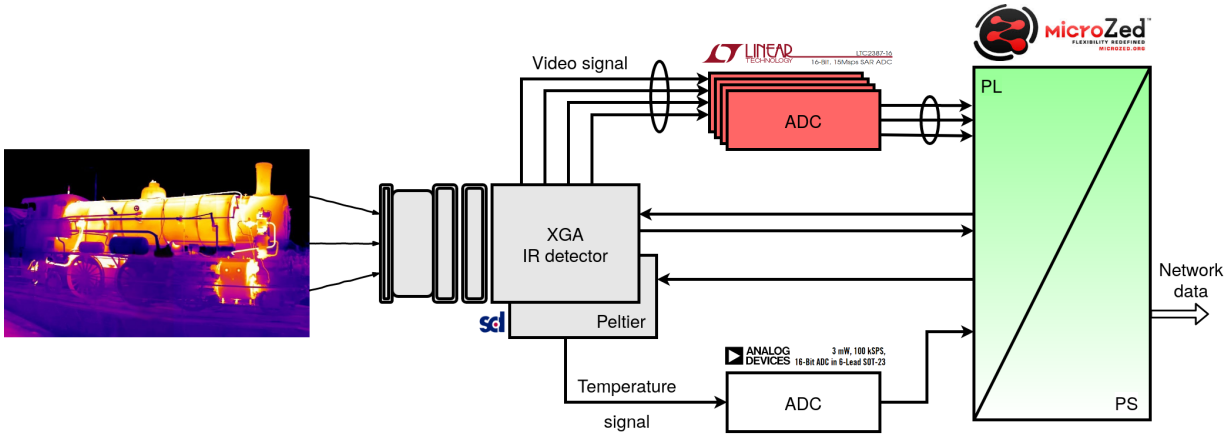

Fig. 2.1: HW block schematic of IR camera

2.1 Uncooled BIRD17-XGA Detector

SCD BIRD17-XGA (Figure 2.2) is a high performance (60 frame per second) and high resolution (1024 x 768 pixels) IR long wavelength detector operating in the 8 - 14 micrometer (LWIR) spectral range. The 17 in the name of the detector stands for 17 µm pixel pitch of the focal plane array (FPA). The detector is composed of:

- A micro-bolometer detector integrated with a silicon Read-Out Integrated Circuit (ROIC)
- Ceramic substrate
- Germanium filter with anti-reflection coating
- Thermoelectric cooler (TEC)

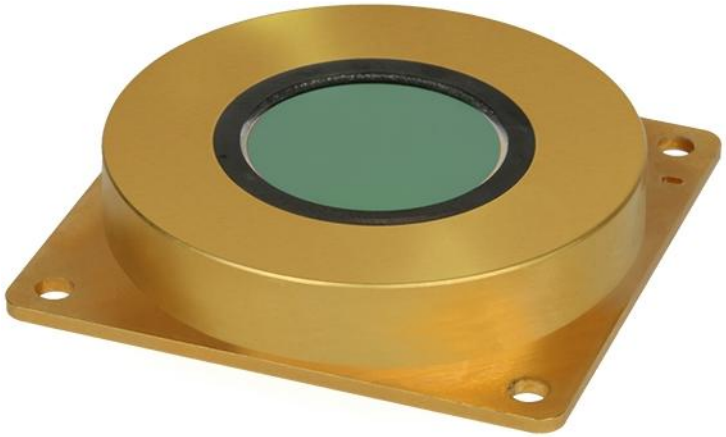

Fig. 2.2: SCD BIRD17-XGA IR Detector

SCD bolometer detector implements a full Wheatstone bridge (Figure 2.3) which gives ability of handling small signal changes (high sensitivity). The micro-bolometer matrix readout is performed using the vertical scanning. During the row integration, each of the row bolometer is connected to the Wheatstone bridge for a predefined period (integration time). After that period, the signal is gained and read out. Output voltage span is between 0.5V and 2.5V and supported span temperature is from -40°C to +80°C. NETD for BIRD17-XGA is 35 mK.

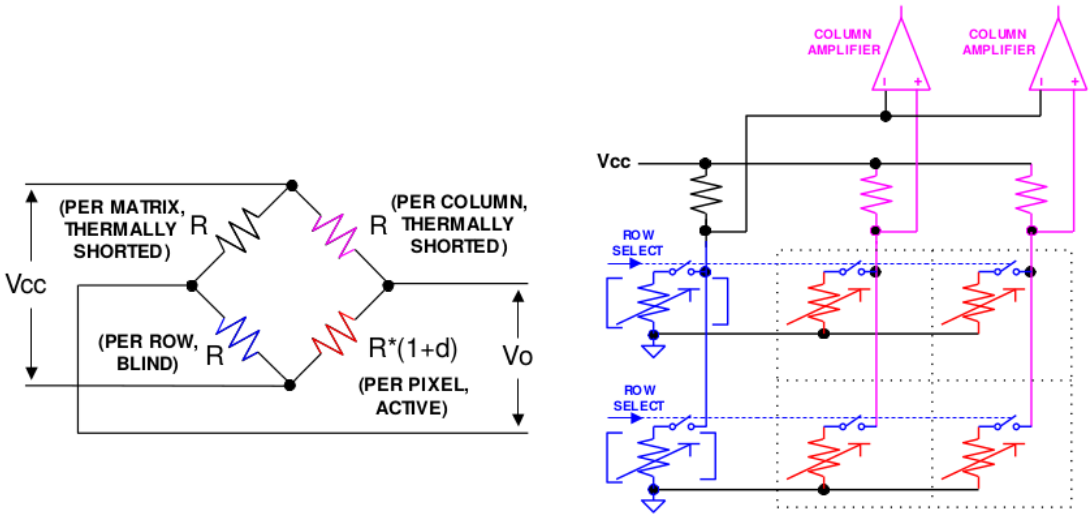

Fig. 2.3: Wheatstone bridge (left) and Read out circuit (right)

BIRD17-XGA offers two possibilities for configuration interfacing, it is Universal Asynchronous Receiver-Transmitter (UART) or Serial Peripheral Interface (SPI).

2.2 Zynq-7000 and MicroZed board

For digital data processing, interfacing with peripherals and communication, Xilinx Zynq is used. Zynq is a System-on-Chip (SoC) which combines ARM Cortex-A9 processor with traditional Field Programmable Gate Array (FPGA) logic fabric. The part formed around ARM Cortex-A9 is called Processing System (PS) and the FPGA part is called Programmable Logic (PL). The PL section is ideal for arithmetic, high-speed interface and data flow. PS serves for application that may not be as precise but rather complex. The benefit is that Zynq supports the operating system (Linux) with easy-to-use SDK called Petalinux. Communication between the PL and PS is made using industry standard called Advanced eXtensible Interface (AXI).

2.2.1 Processing System

ARM Cortex-A9 processor is a hard IP core, which means it is a dedicated and optimised silicon element on the device. PS block diagram is shown in Fig.2.4. The main part is the Application Processor Unit, which comprises of one or two ARM processing cores with Floating Point Unit (FPU), Memory Management Unit (MMU), cache memory and others. The other main parts of PS are IO peripherals (USB, UART, SPI, GPIO, etc.), memory interface for DDR memory and AXI interface. More information can be found in [5].

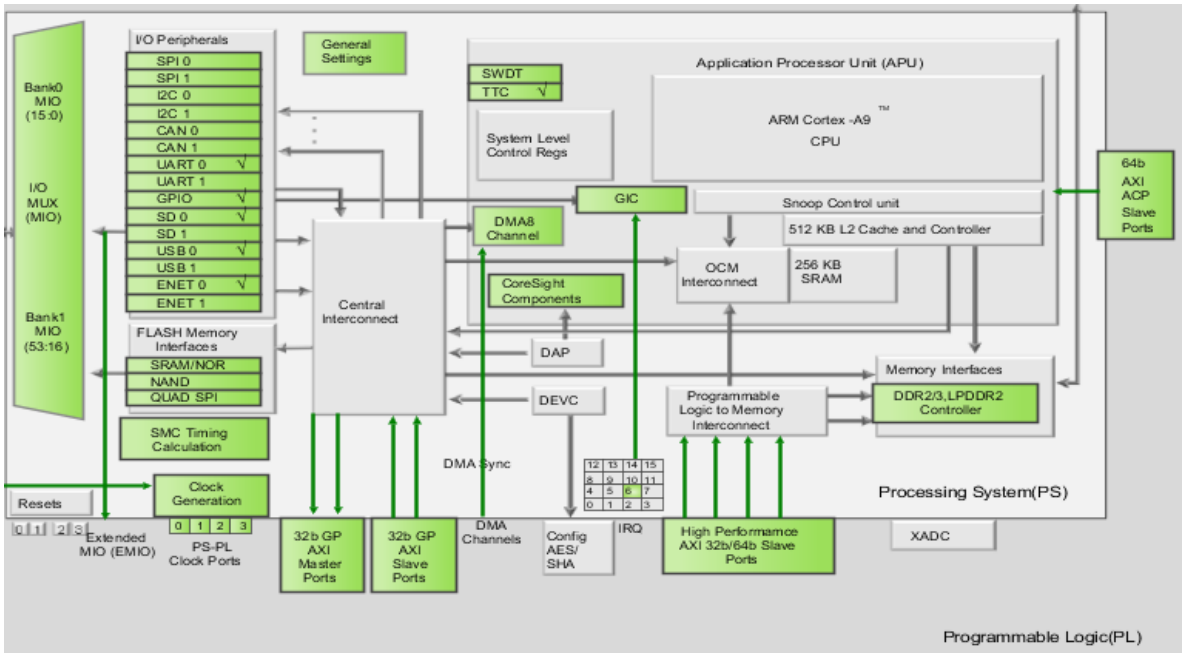

Fig. 2.4: Zynq Processing System from Vivado

2.2.2 Programmable Logic

The PL part of the Zynq device is composed of general purpose FPGA logic fabric, which contains Configurable Logic Blocks (CLBs), Input/Output Blocks (IOBs) and special resources like high-speed arithmetic block DSP48E1.

Configurable Logic Block

Configurable Logic Block (CLB) is a small group of logic elements. Regular CLB consists of slices. Slice is small sub-unit which contains resources (Lookup tables, Flip-flops) for implementing combinatorial and sequential logic circuits.

Input/Output Block

IOB provides interfacing between the PL and physical device pad used to connect external devices. IOB contains some special blocks like serial to parallel converter (SERDES), I/O programmable delay primitive and BITSLIP [7].

2.2.3 Advanced eXtensible Interface

AXI is described as a standard for on-chip communication. Zynq uses version AXI4, which is part of the ARM AMBA 3.0 open standard. AXI buses are used to connect the processor to the IP blocks or multiple IP blocks between each other. There are three types of AXI4 buses where each represents different protocol with its own special property. More information can be found in [6].

- AXI4
  The AXI4 is used for memory-mapped links. This protocol reaches the highest performance. AXI4 uses address for data burst transfer of up to 256 data words.
- AXI4-Lite
  The AXI4-Lite is a simplified AXI4 protocol. It is also memory-mapped but only single data word (no burst) is transferred on each address.
- AXI4-Stream
  The AXI4-Stream protocol is used for applications that typically focus on data-flows, where the concept of address is not present or not required. Each stream acts as a signle unidirectional channel for a handshake data flow.

For communication between PS and PL, Zynq offers 3 types of interfaces:

- General Purpose AXI
  There are four general purpose AXIs. It is a 32 bit data bus, which is suitable for low rate communication between PS and PL with absence of buffering. An

example is communication with AXI General Purpose Input/Output (GPIO) IP core.

- High Performance AXI
  There are also four high performance AXI interfaces. This interface supports high rate communication between PS and PL with buffering. Data width can be 32 or 64 bits. This interface is mainly used for Direct Memory Access.
- Accelerator Coherency Port
  Provides connection between PL and APU with unchangeable 64-bits width. The port is used to achieve coherency between the APU caches and elements within the PL.

2.2.4 MicroZed board

MicroZed (Figure 2.5) is an Avnet development board based on the Xilinx Zynq SoC. The MicroZed Evaluation Kit includes standalone MicroZed that contains fully functional Zynq Processing System (PS) with peripherals as well as enabling the Zynq Programmable Logic (PL) fabric. This PS system includes DDR3 memory, Flash memory, gigabit Ethernet, USB 2.0 Host, and UART. The capabilities of the MicroZed can be enhanced by plugging it into a carrier card, which then enables up to 108 user defined I/Os. [8]

MicroZed Key Features:

- Processor
  Zynq™-7000 AP SoC XC7Z010-CLG400-2
- Memory
  1 GB DDR3
  128 Mb Quad-SPI Flash
  microSD card
- Communication
  10/100/1000 Ethernet
  USB Host 2.0 and USB-UART
- Expansion connectors
  2 MicroHeader connectors
  108 single-ended, 48 differential pairs, Agile Mixed Signalling
  Digilent Pmod™ Compatible header (8 MIO)
- Clocking
  33.33333 MHz clock source for PS
- Configuration and Debug
  Xilinx Platform Cable JTAG connector

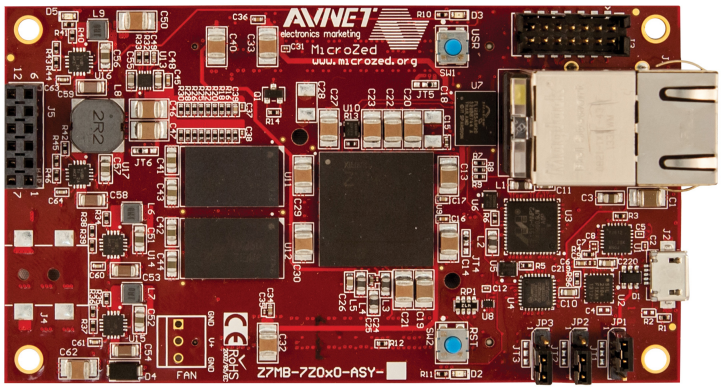

Fig. 2.5: MicroZed board [8]

2.2.5 LTC2387-16

LTC2387-16 is a high speed successive approximation register (SAR) ADC with 16-bit resolution and 15 MSps (MegaSample per second) throughput rate. The input signal must be driven differentially with common mode voltage 2.048 V and differential range ±4.096 V. In order to support high speed operation and minimizing the number of data lines for communication, serial Low Voltage Differential Signal (LVDS) digital interface is used.

2.2.6 AD7680

AD7680 is a 16-bit successive approximation register ADC with 100 kSps throughput rate. Single ended input signal can reach values from 0 V to the VDD (designed to 3.3 V). AD7680 offers high speed serial interface compatible with SPI/QSPI/µWire/DSP. SPI was chosen for the purpose of this project.

3 Software

This chapter describes the design of the implementation in Xilinx SoC Zynq-7000 in PL and PS. Intellectual Property (IP) cores (written in VHDL) were designed in the Xilinx Vivado 2019.1. The embedded Linux solutions on the PS side was designed in Xilinx Petalinux Tools 2019.1 [13] based on Yocto project. Kernel modules (written in C) and the main control program (written in C++) were designed in Xilinx SDK 2019.1. Graphical User Interface (GUI) for data visualisation was designed in Qt with Qt Creator 4.11.1

The PL part and IP cores are used for high-speed interfacing, signal processing and communication with peripherals. Linux with kernel version 4.19.0-xilinx-v2019.1 is running on the PS side. Kernel modules are pieces of code running in the kernel space. They can be imagined like communication gates with hardware, or in our case, with the PL part (IP cores). The main control program runs in the user space and it has control over communication and setting of the BIRD17-XGA, temperature stabilization, TCP server for data transmissions to PC and etc. SW block diagram is in Fig.3.1.

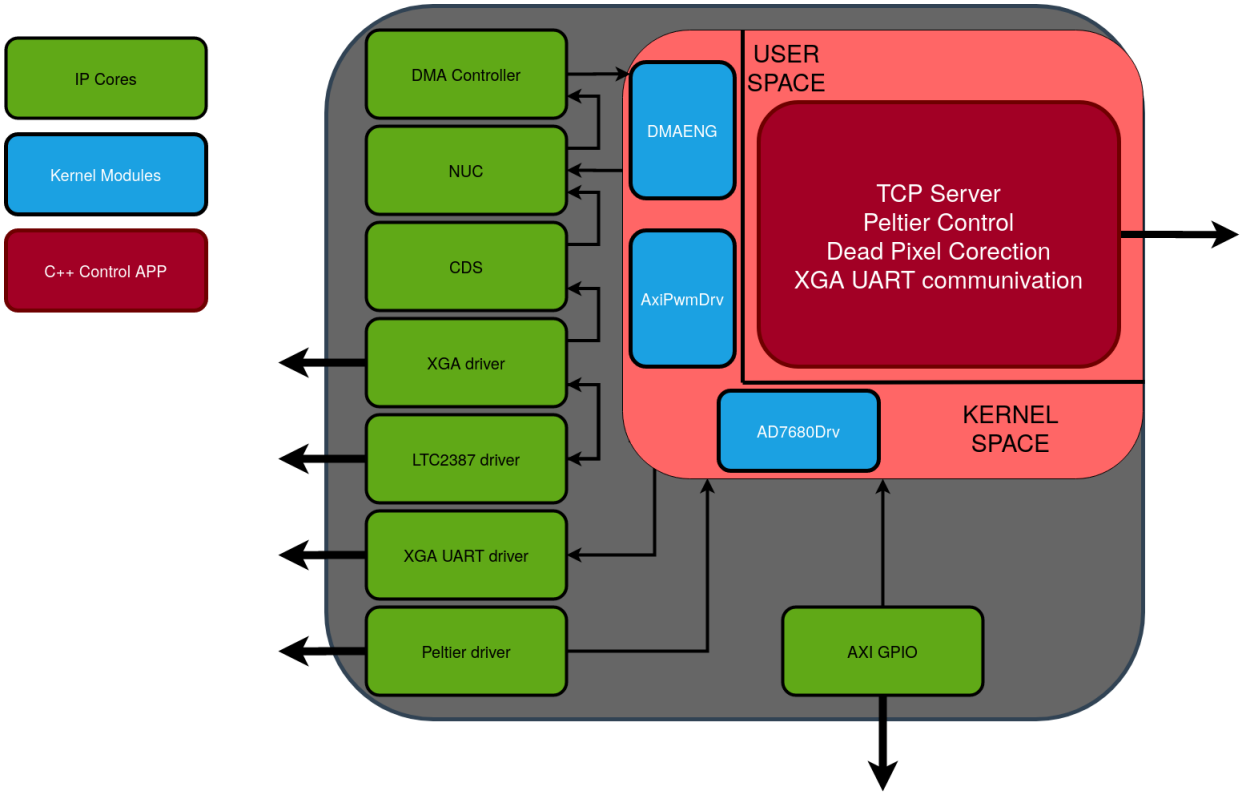

Fig. 3.1: SW block diagram

Designed IP cores, kernel modules and C++ libraries in this chapter were designed in the same order as listed. During development, focus was placed on the reusability of the developed IP cores, kernel modules and libraries.

3.1 AXI4-Stream

Before presentation of the designed IP cores, here are a few words about AXI4-Stream, which is used for data forwarding. AXI4-Stream is master (M_AXIS) -> slave (S_AXIS) interface which uses 5 communication signals: TREADY, TDATA, TVALID, TLAST and TKEEP. The TREADY signal is driven by slave and is informs the master, when data are ready to receive. After that master starts sending data through the TDATA bus. For valid data, the TVALID signal must be set (if the TVALID = '0' -> DATA are discared). The TLAST is used to inform the slave that the last data from packet are being sent. The TKEEP bus is used to indicate the amount of valid bytes in the last data transfer. AXI4-Stream diagram can be seen in Fig.3.2.

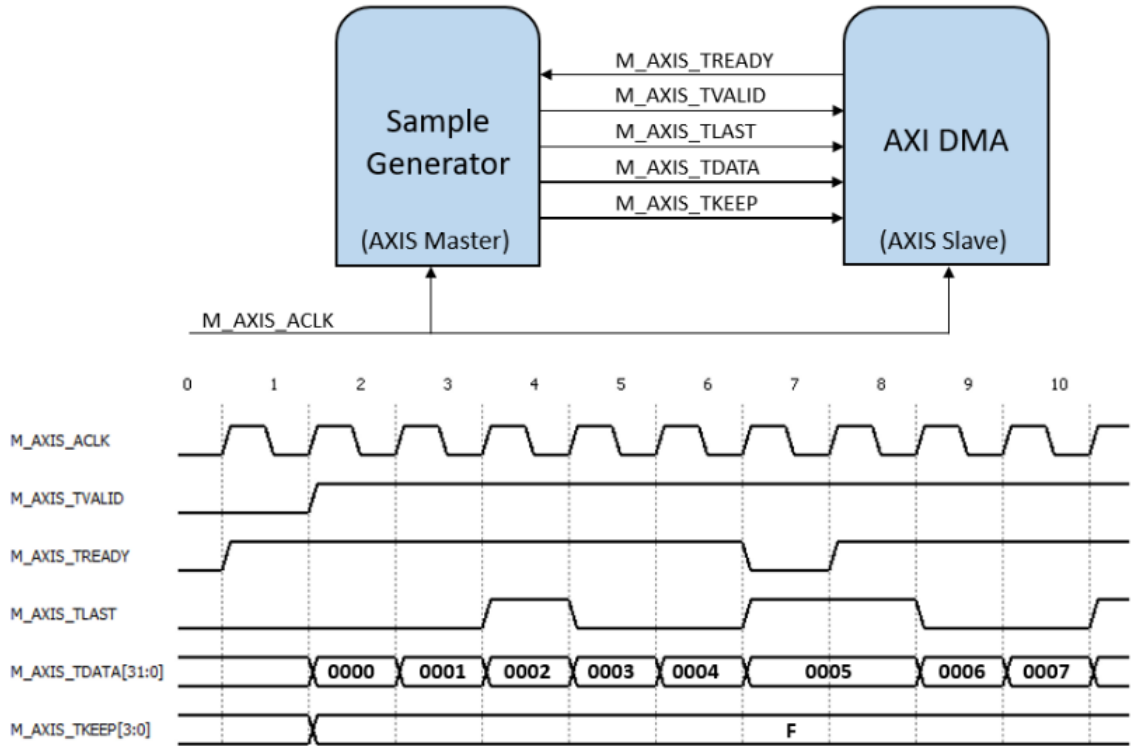

Fig. 3.2: AXI4-Stream diagram [15]

3.2 DMA Controller and DMAENG

DMA Controller is a name of the hierarchy block in Vivado design, which is used for transferring high amounts of data between PL and PS. For high throughput of data Xilinx offers an IP core called AXI Direct Memory Access (DMA). This IP core is able to transfer theoretically 400 MB/s with 100 MHz clock. On the PS side, there was created a kernel module called dmaeng, which receives data from the PL and saves them into the shared (user/kernel space shared buffer) ring buffer.

As it was written in chapter 2.1, resolution of the BIRD17 is 1024 x 768 pixel with 60 fps. Every pixel is represented by 16 bits. These numbers allow to calculate necessary throughput from PL to PS. From Eq.3.1 calculated data throughput is equal to 94.3 MB/s.

$$Throughput_{data} = \frac{Pixels \cdot fps \cdot BitsPerPixel}{8} \tag{3.1}$$

3.2.1 DMA Controller

DMA controller is made from connection of three IP cores (Fig.3.3). AXI4-Stream Data FIFO is used as buffer for the case when DMA is busy. The FIFO size is set to 64 bits x 4096 words. Data from the FIFO flows into the AXI DMA block. The DMA supports two data-flow modes - stream-to-memory-mapped (S2MM) and memory-mapped-to-stream (MM2S). In this design I only used S2MM channel for transferring data to PS memory. DMA AXIS interfaces support 8/16/32/64/128/265/512/1024 bit wide TDATA busses. This IP core also supports scatter/gather functionality (necessary for cyclic transferring). AXI SmartConnect core connects one or more AXI memory-mapped master devices into one or more memory-mapped slave devices (Vivado automatically handles this connection for us). AXI SmartConnect is directly connected to HP port of the PS. The s2mm_introut signal from the AXI DMA interfacing together with IRQ_F2P is capable of producing interrupts into the processing system.

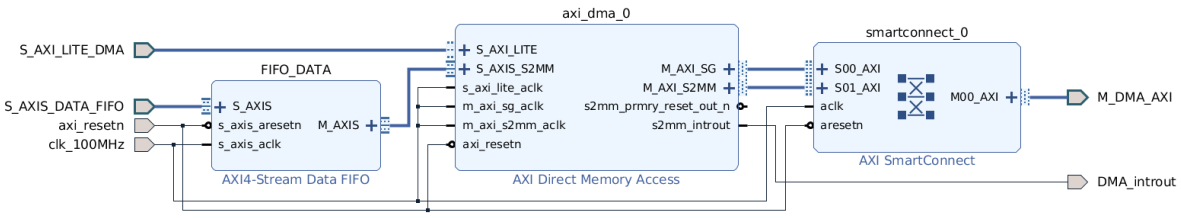

Fig. 3.3: Block diagram of the DMA Controller

3.2.2 DMAENG

The AXI DMA core is controlled by the Linux device driver dmaengine offered by Xilinx. This driver was wrapped into custom character device driver called dmaeng. During the boot sequence, the kernel module is probed according to the configuration file. The configuration file (config.h) activates tx, rx or both DMA channels, defines board memory size and starting address of the rx/tx ring-buffer. The ring buffer is allocated in free memory. Reservation of a memory section in the DDR memory can be done by changing of the device tree. Reserved memory should be rounded to the Linux page size (in version 4.19.0 kernel it is PAGESIZE = 4096 B = 0x1000 B). The following example describes a device tree which reserves last 40960 bytes and another device tree node, which is DMA node.

```
memory {
        device_type = "memory";
        reg = <0x0 0x3FFF6000>;
};

axidmaeng_1: axidmaeng@1 {
        compatible ="PSI,dmaeng";
        dmas = <&axi_dma_0 0>;
        dma-names = "rx_channel";
};
```

The most critical part of the this system is where the samples from kernel ring-buffer are copied to the user space. If copying the image data takes longer than it takes for the IR detector to generate new data, the ring-buffer will eventually fill and samples will be lost. For the zero-copy data method the POSIX standard offers mmap function (library sys/mman.h). From the manual of the function: This function creates a new mapping in the virtual address space of the calling process. In another words, the mmap function creates shared ring-buffer for the user space and kernel space and no data copying is necessary.

IOCTL (input/output control) and sysfs are two types for communication with the kernel module. IOCTL is a system call to express the unique functions of the device which are not included in standard system call. IOCTL is not usable from bash and needs sudo privileges for access. The kernel driver parameters are hidden behind ioctl numbers and accessed from application through ioctl system function. Sysfs is a newer method for communication with the kernel module through generated virtual files (called attributes). These attributes can be controlled from bash

without sudo privileges. Another method of controlling the driver from application is using standard read/write system functions. For all designed modules I chose the sysfs model, because of easier debugging from console and cleaner access. Table 3.1 shows attributes for communication with the driver.

Tab. 3.1: DMA Kernel driver attributes

startTX/RX	DMA starts transmitting/receiving
stopTX/RX	DMA stops transmitting/receiving after valid data packet
configTX/RX	Set [(INT) number of descriptors for ring buffer, (INT) ring buffer size]

C++ class was created in order to control the DMA driver from the application.

3.2.3 Performance test

For verifying Vivado design, DMA driver, C++ API and finding the maximum throughput a test design was created (Fig.3.4). An IP core called Data_Generator was built specially for testing purposes. The Data Generator (Fig.3.5) simulates the source of video data with adjustable output data width, frame height, frame width and bits per pixel. Generated data create pattern that can be used for fast visual check. Average throughput of 375 MB/s (theoretical throughput is 400 MB/s) was measured with configured XGA17-BIRD resolution. Measured throughput is sufficient and usable for the desired XGA_throughput = 94.3 MB/s.

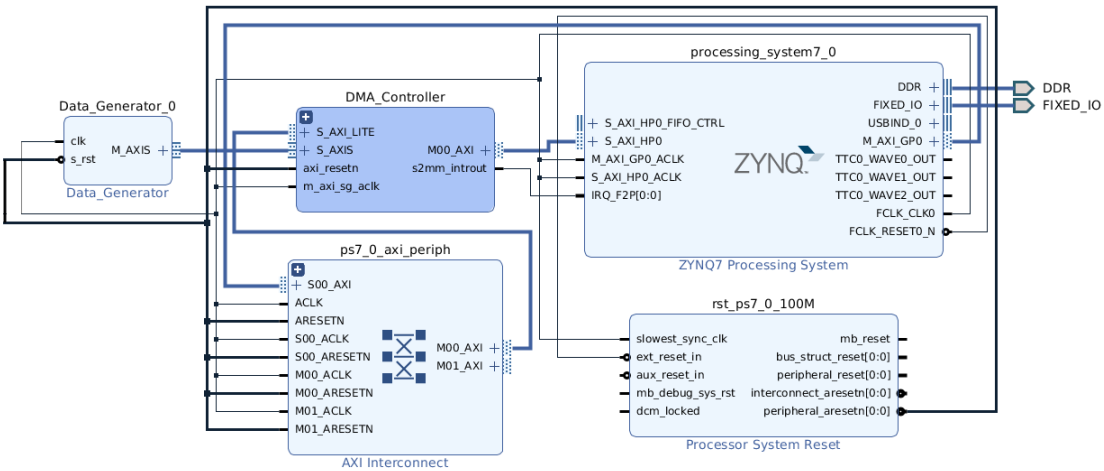

Fig. 3.4: DMA test design

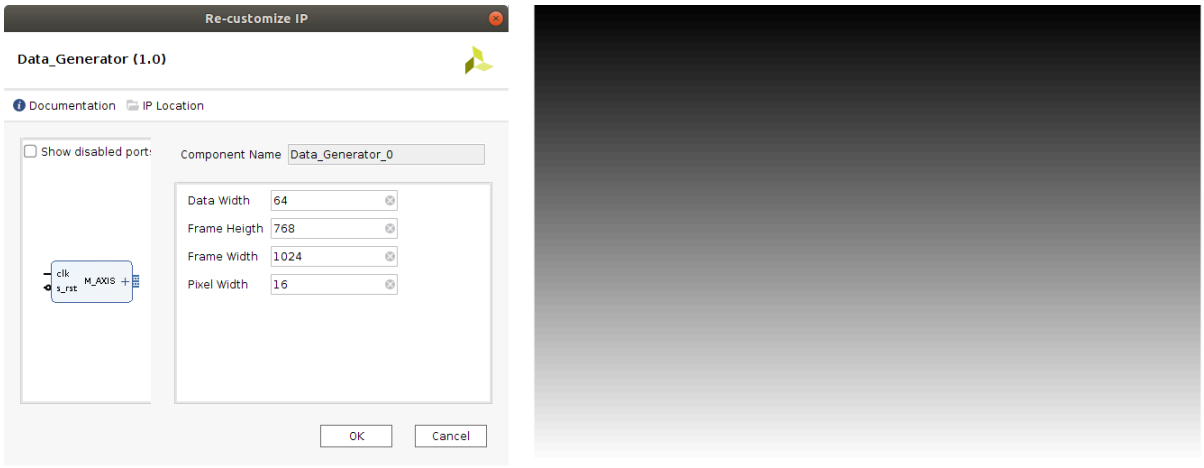

Fig. 3.5: Data generator (a) and generated pattern (b)

3.3 ADC Interface

ADC Interface hierarchy block is used for interfacing with the LTC2387-16. LTC2387-16 uses for communication interface these LVDS signals:

- Input clock (CLK)
- Data Clock Output (DCO), which is echoed version of CLK
- Conversion (CNV), which puts the internal sample-and-hold circuit into the hold mode and starts the conversion cycle - all synchronized with the CNV rising edge
- Data outputs (DA/DB)

Due to lack of free pins on Microzed board, DCO and DB signals are not connected. This leads to certain limitation in interfacing. Because of unconnected DCO, the system synchronous clocking has to be used instead of source synchronous clocking. Due to unconnected DB, the two-line mode can not be used for data receiving. The power-down mode can be involved if low state is connected to control pin $\overline{PD}$. For setting up the correct timing and data verifying, the ADC is able to generate a test pattern after connecting the logic-high on the TESTPAT pin.

A conversion is controlled with a rising edge on CNV, which must be at least 5 ns long. When the conversion is complete (t_{CONV} - max 63 ns), the most-significant data bit is available on DA. In this moment, the data is ready to be shifted out by applying a burst of 8 clock pulses to the CLK. Data is transfer in Double Date Rate (DDR), which means data is updated with both CLK edges. The edges of DA and DCO are aligned, but DCO is not connected and can not be used to latch data. The data has to be synchronized with system clock CLK. The data must be shifted

out after the current conversion is complete and before the next conversion finishes. The timing diagram is shown in Fig.3.6.

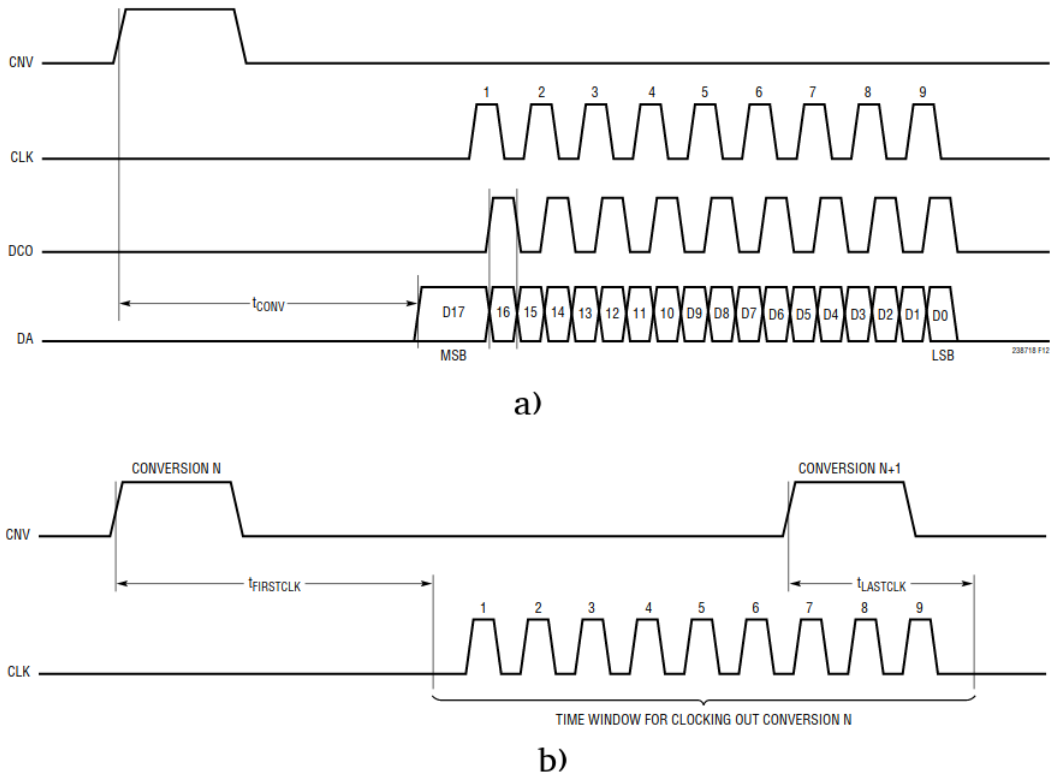

Fig. 3.6: LTC2386-16 Timing diagram a) Single conversion and b) Valid time window for clocking out data [17]

XGA-BIRD17 generates video data with 15 MHz (66.66 ns) frequency, identical to the sampling frequency of the ADC. In the ADC datasheet [17] are listed the ADC timing characteristic. From these we can calculate Eq.3.2 the smallest working clock frequency = 158 MHz. The upper limit of the clock frequency is 400 MHz.

$$CLOCK_{PERIOD} < \frac{ADC_{SAMPLE\_PERIOD} + t_{LASTCLK} - t_{FIRSTCLK}}{8} \tag{3.2}$$

In the design the sampling frequency of the ADC driver is set on 180 MHz. This frequency was picked with respect to XGA-BIRD17 clock which is 60 MHz. The main benefit of this value is that Phase Lock Loop (PLL) can be used for its generation (multiplying by 3) and no clock domain crossing is necessary.

The output is in two's complement format. The ADC reference is set to 4.096 V and this results in the LSB voltage of 62.5 µV.

3.3.1 LTC2387_v1 IP core

The LTC2387_v1 IP core is controlled with a Finite State Machine (FSM) (Fig.3.7) which has three states:

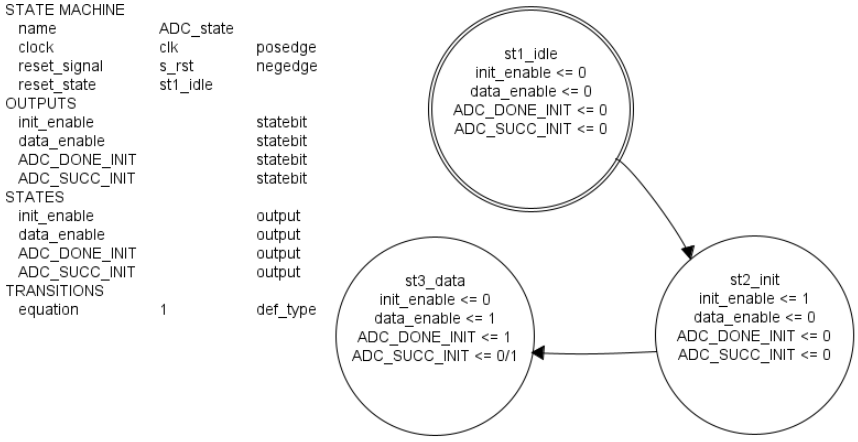

Fig. 3.7: LTC2387_v1 finite state machine

Idle - st1_idle

Default state after power up or reset signal is the Idle state. This state serves only for waiting untill delay_locked signal become ready. The delay_locked signal comes from IDELAYCTRL [19] block primitive, which must be present in conjunction with the IDELAY primitive. The RDY output (connected to delay_locked signal) of the IDELAYCTRL primitive indicates when the IDELAY module in the specific region is calibrated.

Init - st2_init

The init state is used for timing synchronization with the internal clock. Because of absence of the DCO signal, variabled CLK to DA delay (0.7 ns to 2.3 ns) and static delay (PCB route) synchronization process must be done. Without the synchronization, metastability might occurs when the sampling frequency is aligned with the data change or skew and the received data might corrupt.

For finding ideal sampling windows (center of a data eye) is used a programmable delay primitive called IDELAYE2 [20]. Every I/O block contains the IDELAY and ODELAY block. IDELAYE2 has 31 taps and every tap is 58 ps long (IDELAY ranges from 0 to $31 \cdot 58$ ps).

The synchronization starts with generation of the known ADC data pattern. Two counters are prepared. The first one is used as a tap counter named in the test bench

as sig_cnt_32 and the second one as a pattern counter named sig_cnt_100. The process starts with tap delay set to 0 ps. The pattern counter increments up to 100 and in every step it checks if the pattern was received correctly. If all received data patterns were correct, the value of the tap counter would set as the lowest working delay value in the sig_taps_min signal, the tap counter would be incremented, the pattern counter reset and and the process would start over again. With this method it is possible to find the lowest and highest working tap delay. The average tap of these two values is set to IDELAY2 as the optimal working delay.

Figure 3.8 shows the test bench for the synchronization process. For this test bench the internal delay of the ADC is set to 1.3 ns and the PCB path delay to 1.5 ns. The final average tap value set to IODELAY2 is 11. Figure 3.9 shows real captured data by the Integrated Logic Analyzer (ILA). From the ILA's data we get almost the same values as from the test bench. The final tap value is equal to the test bench tap value = 11.

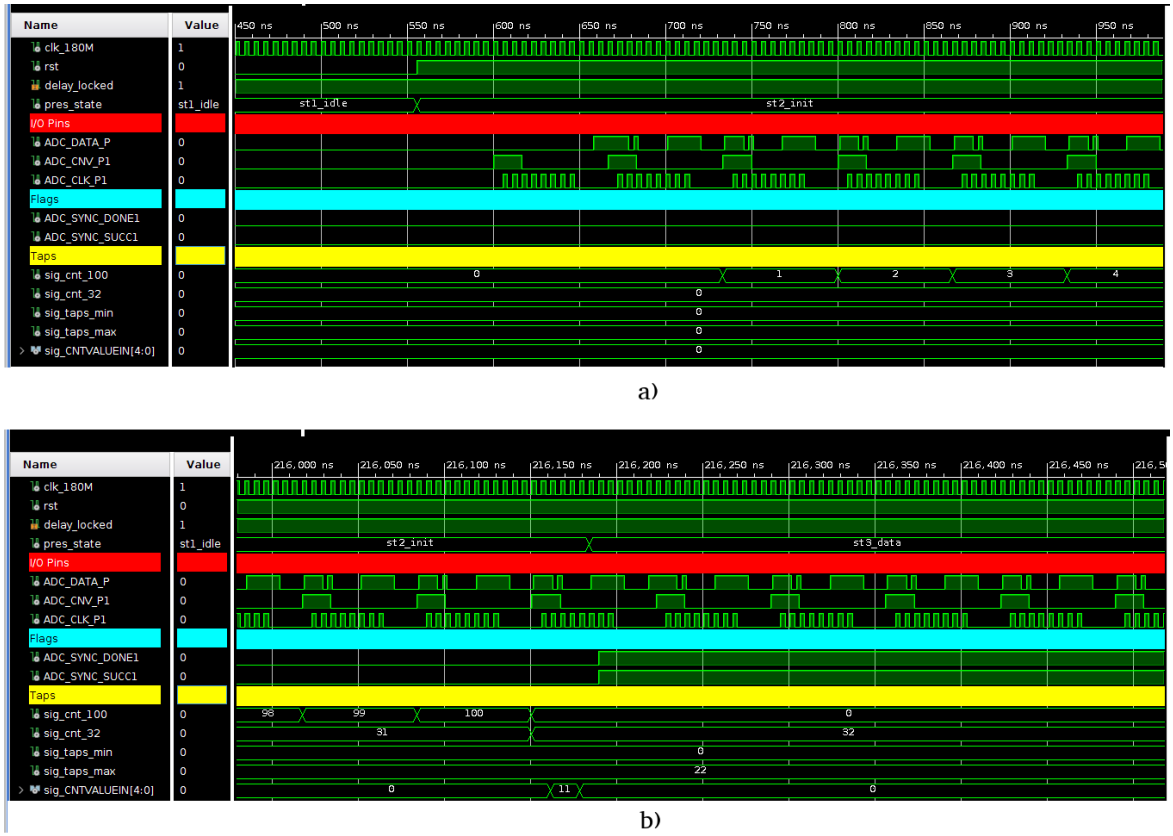

Fig. 3.8: Test bench of timing synchronization a) start of the st2_init state and b) end of the st2_init state

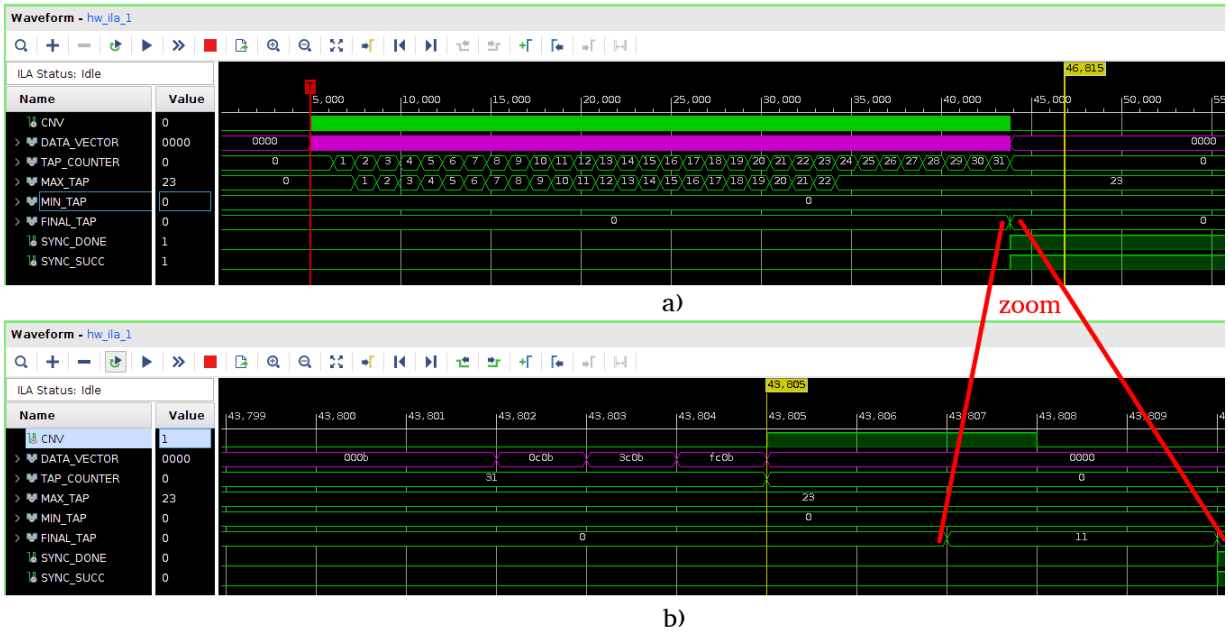

Fig. 3.9: a) Captured data of the timing synchronization by ILA and b) detailed view

Data - st3_data

After the synchronization process, the output bit ADC_DONE_INIT is set to high level. ADC_SUCC_INIT is set high if the synchronization was successful. In this moment, the ADC is ready to use. High level on the input port enable_CNV starts the ADC conversion. The delay of the IP core is 4 clock cycles, untill the CNV signal is physicaly shifted out. New conversions are started when enable_CNV stays in high level. Valid data are shifted to the DATA output port with DATA_VALID signal set.

Primitives

For interfacing with LVDS-DDR signal coming from LTC2387-16, Zynq primitives must be used. Three primitives, connected in series, are used for the incoming data. The first one is IBUFDS for converting differential signal into single ended. The signal from IBUFDS is fed into IODELAY2 for timing synchronization. The last primitive is called IDDR and it converts signal from double date rate into single date rate. After this process, data can be latched and shifted for next processing.

For forward clock generation, ODDR primitive is used. The ODDR is able to enable or disable the output clock signal, which is very useful for ADC clock interface constraints. For single ended to differential signal conversion, OBUFDS is used. [20]

ADC Interface Hierarchy

Final version of the IP core LTC2387_v1 has two properties, which are "Number of ADC", which configurates the number of used external LTC23787-16 chips and "Ref Clk", which is the value of the reference clock for IDELAYCTRL. This IP core is inserted into the hierarchy block (Fig.3.10) called the ADC_Interface. Beside LTC2387_v1, AXI_GPIO IP core and Simple_Clock_Synchronisation are inserted. The AXI_GPIO is used as a reset and LTC2387_v1 status reading from the PS side. Because LTC2387_v1 runs on 180 MHz and AXI_GPIO on 100 MHz, Simple_Clock_Synchronisation is used for Clock Domain Crossing (CDC).

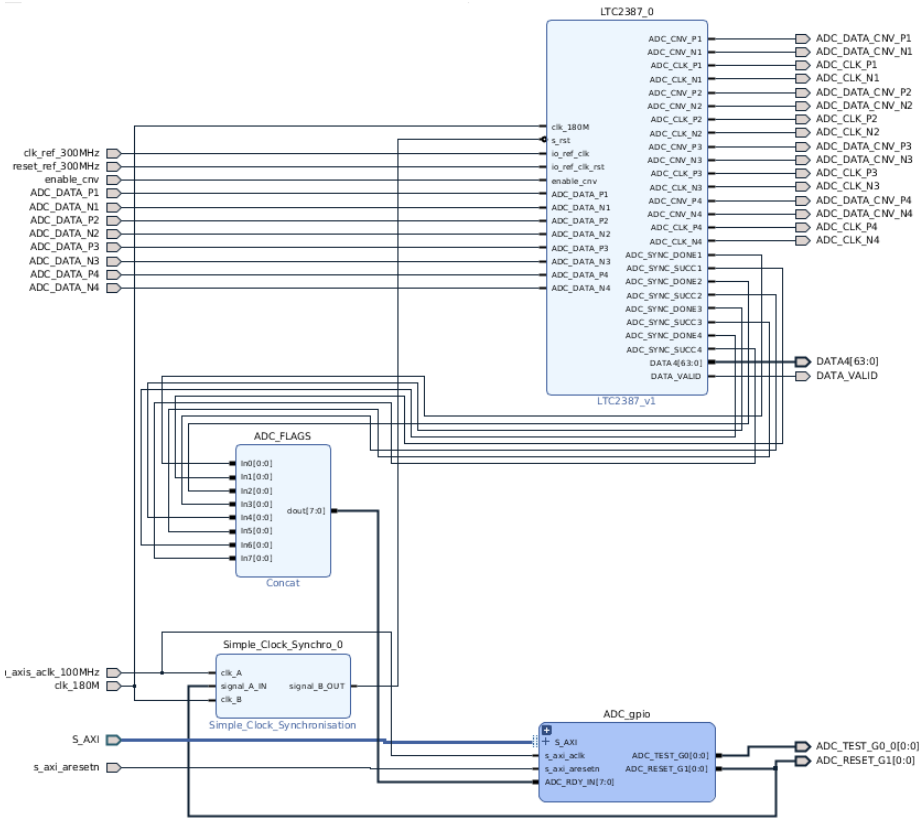

Fig. 3.10: ADC Interface Hierarchy

3.3.2 ltc2387drv library

The ltc2387drv is a C++ library for controlling the LTC2387_v1 which contains only one overloaded function. It is an initialization function which should be called at the beginning of the program. The function starts with resetting the IP core and waiting, until ADC_SYNC_DONE signal becomes ready. When ready, the ADC_SYNC_SUCC flag is read.

3.4 XGA Interface

XGA Interface is a hierarchy block, which integrates communication and video interface with XGA-BIRD17. For this detector it generates the master clock with frequency of 60 MHz (clk_out_60MHz).

3.4.1 XGA Video Interface

XGA-BIRD17 has three digital video signals which are used to synchronize events between the system (FPGA) and detector (XGA-BIRD17): Frame start synchronization (F_Sync), Video frame indication (V_Sync) and Video line indication (H_Sync). All these synchronization signals are synchronous to the system output clock clk_out_60MHz.

The requested frame starts with assertion of the F_Sync signal, driven by system. The detector responds with assertion of signal V_Sync. The V_Sync stays set during whole frame. After assertion of the V_Sync, H_Sync pulses are generated by the detector. Every H_Sync pulse covers the duration of the output of the video of one row. General frame timing and the activity of digital video signals is shown in the following Fig.3.11.

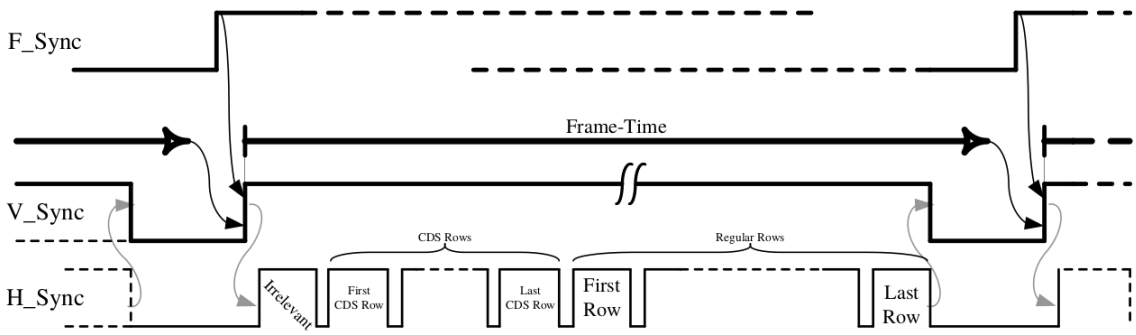

Fig. 3.11: General frame timing [21]

According to F_Sync signal, there are two video modes: System-Controlled and Free-Running.

System-Controlled Operating-Mode

In System-Controlled mode system issues F_Sync pulse with a duration of at least one clock cycle, whenever a new frame should start. The frame duration should be constant and even number of clocks, otherwise image quality is degraded.

Free-Running Operating-Mode

In Free-Running mode is continuously F_Sync = '1' and when frame time is elapsed, XGA-BIRD17 automatically starts a new frame. When F_Sync is deasserted, the detector completes the frame in progress and stops.

For developing the camera in this master thesis, Free-Running mode was chosen, which is more suitable for video data stream and better image quality should be reached.

IP core XGA_driver_v2

IP core XGA_driver_v2 is controlled by FSM with 4 states (Fig.3.12).

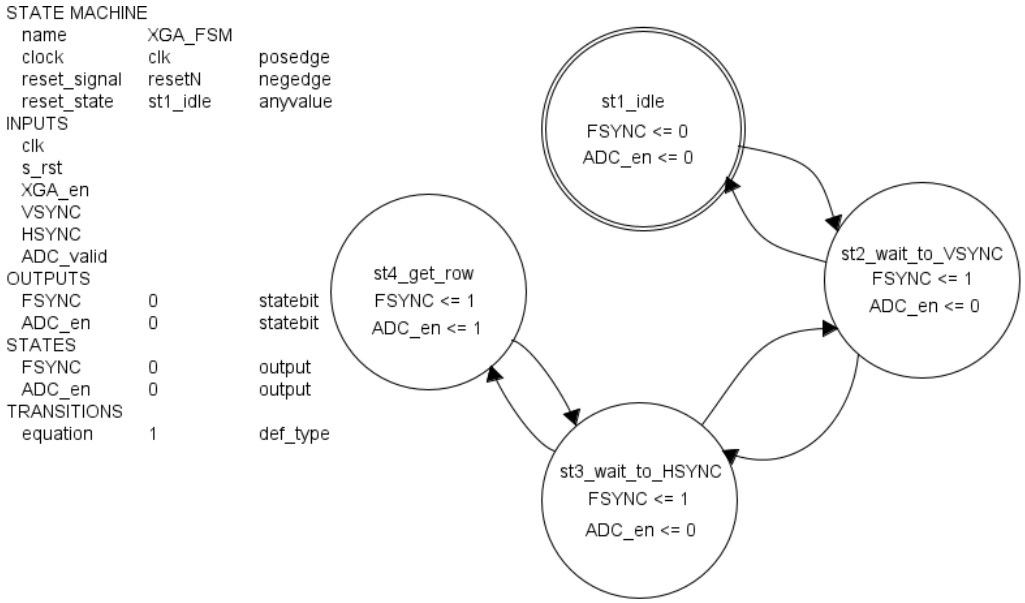

Fig. 3.12: XGA_driver_v2 finite state machine

The default state is st1_idle and for starting the video data streaming the XGA_en signal must be asserted. When FSM comes to the second state (st2_wait_to_VSYNC) F_Sync is set (XGA-BIRD17 runs in Free-Running mode) and FSM waits for rising edge of the V_Sync signal from XGA-BIRD17. After rising edge of the V_Sync signal, FSM is updated to st3_wait_to_HSYNC, where it waits for a rising edge of the H_Sync signal. In the moment when H_Sync and V_Sync are set, FSM goes into st4_get_row state. In this state, output signal ADC_en is asserted. This signal is connected to enable_cnv of LTC2386_v1 IP core which starts the conversion. For controlling how many pixels were already read, signal DATA_VALID from LTC2386_v1 is routed back to XGA_driver_v2.

After receiving the DATA_VALID = Row Pixels (Row Pixels is an input variable of the XGA_driver_v2), FSM goes back to st3_wait_to_HSYNC and waits for the next rising edge of the H_Sync signal. When the counter of H_Sync is equal to the frame height, FSM moves to st2_wait_to_VSYNC. Depending on the XGA_en signal, FSM stays in this state and waits for new rising edge of the V_Sync signal or goes to st1_idle.

The final version of XGA_driver_v2 (Fig.3.13) has three customized parameters for setting the frame size. The test bench for this IP core is in Fig.3.14. Test bench in conjunction with IP core LTC2387_v1 can be found in Appendix B.2.

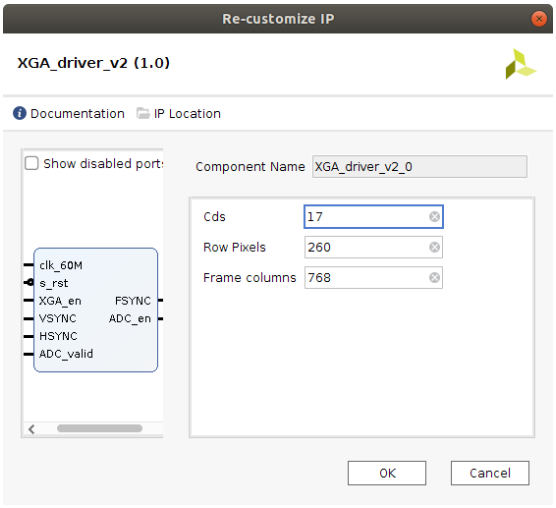

Fig. 3.13: Customized IP – XGA_driver_v2

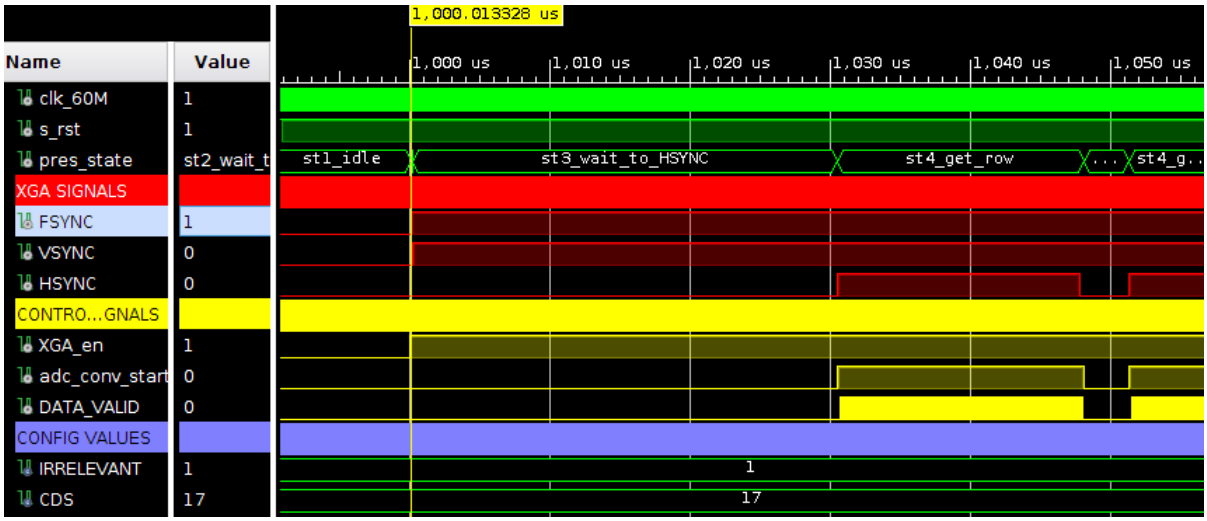

Fig. 3.14: Test bench of the XGA_driver_v2

3.4.2 XGA communication interface

XGA-BIRD17 detector supports communication interfaces, which allow access to management of various detector parameters. The communication with the detector can be implemented in two modes:

- Parallel interface - synchronous with CLK (similar to SPI)
- Serial interface - Asynchronous, using UART

Due to routed UART on the design of the PCB, only the Serial interface will be further considered. The UART parameters are described in the following Tab.3.2:

Tab. 3.2: XGA UART parameters [21]

Parameters	**Details**	**Comments**
8N1	8-bit data, no parity, 1 stop-bit	Simple Protocol: "Message" followed by optional "Response"
Half-Duplex	No simultaneous Receive and Transmit.	
Baud-rate	19.2K ÷ 230.4K	BIRD17-XGA auto-detects the System Baud-Rate

BIRD17-XGA detector supports the communication baud rate automatic detection functionality. UART mode requires the system (FPGA) to initialize baud rate auto detection sequence prior to sending any communication message.

XGA_UART Hierarchy

XGA-BIRD17 enters into baud-rate auto-detection in 2 cases. First is during power-on-reset sequence and second is after UART-reset. The flowchart of baud-rate auto-detect sequence is in Fig.3.15. In the first step, the system continuously transmits "0XF5" (hexadecimal) characters. XGA-BIRD17 monitors received characters and runs auto-detect algorithm in order to identify the baud-rate. After baud-rate is recognized, detector starts responding with "0x35" character repeatedly. The system stops the transmission and waits until detector stops the response. The system waits for the Inter-Message Interval (which is 24 characters long) and then the communication is prepared. The auto-detection fails when the detector doesn't receive response within 100 transmitted characters or detector responses with more than 100 characters.

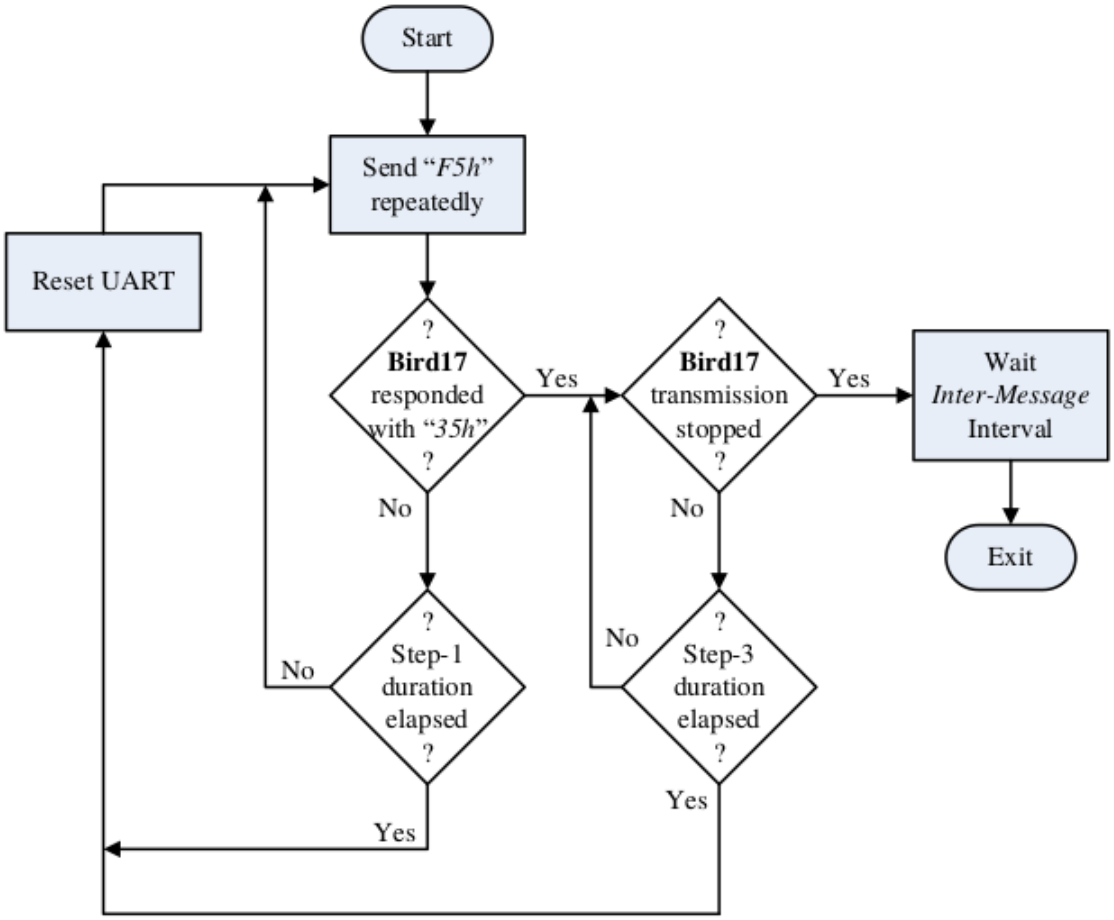

Fig. 3.15: Auto-detect flowchart [21]

Linux is able to access the serial port from the user space. Unfortunately, Linux isn't real time operation system and for strict timing parameters of the auto-detection it isn't suitable. For this reason the auto-detection is moved to the PL and after the auto-detection is control over the UART handed over to the PS user space.

The content of hierarchy block called XGA_UART is in Fig.3.16. XGA_UART_Synchronisation_v1 is a custom IP core used for the auto-detect phase. The start of the auto-detection is controlled from the PS via input port Init_start connected to axi_gpio. Signals Init_done and Init_successful inform the PS about the auto-detect result. If the auto-detection is successfully done then axi_uartlite _0 [23] takes control over the UART communication. The test bench can be found in Appendix B.3.

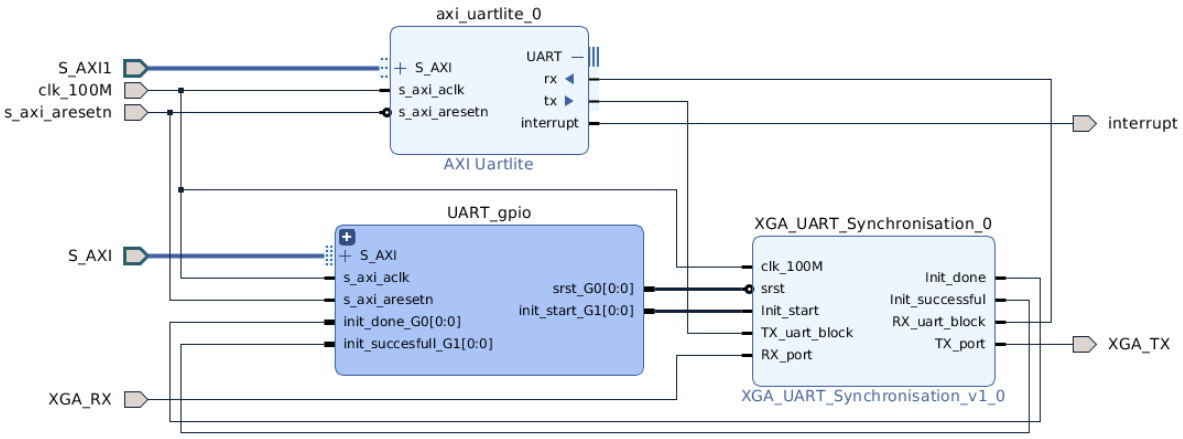

Fig. 3.16: XGA UART Hierarchy

Xgauartdrv class

From Tab.3.2 it is clear that the used UART format is of 8-bit data length, no parity and 1 stop-bit. However, the word size is 6-bit long in the communication protocol with the detector. Bit6 and bit7 of each Byte must be 0, otherwise the message is considered faulty. The system is always the initiator of communication and the detector responds with the appropriate response.

The message structure is shown in Fig.3.17. It consists of 4 parts:

- **Cmd_ID** - Command identification number
- **Cmd_ID_N** - The inverse of Cmd_ID (1's complement)
- **Message_body** - Data of command (one or more words)
- **CRC** - Complement of "0x00" to the bitwise XOR of all the previous words in the whole message
  $CRC(\mathbf{i}) = XOR\{1^{st}word(\mathbf{i}), 2^{nd}word(\mathbf{i}), ..., (last-1)word(\mathbf{i})\}$

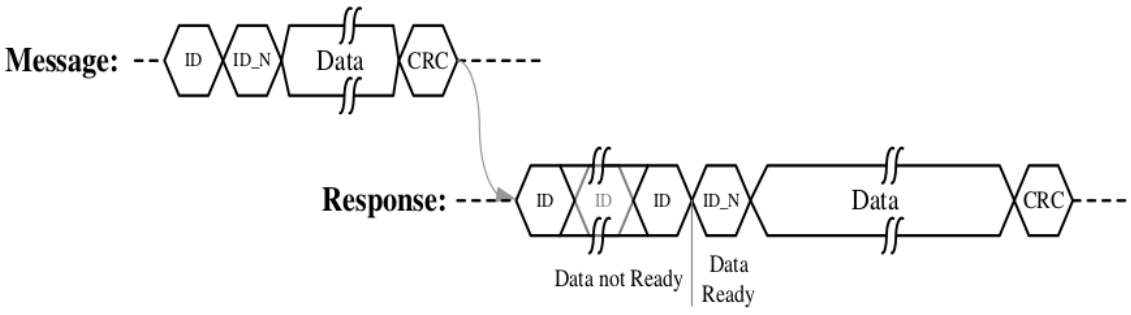

Fig. 3.17: Message and Response Protocol [21]

For communication via UART, Uartdrv class was written. This class contains standard functions for setting UART parameters, write and read function. From this class inherits the Xgauartdrv class, created for communication with the detector with all its communication constraints. The detector has a large number of configuration parameters, but only the most interesting are present (the rest can be found in [22]).

Reset-Command - This command executes the soft-reset. The soft-reset stops video processing operation, all parameters are configured to default values. [22]
Status-Command - This command is used for checking indication regarding various errors as well as the internal compensation process status. [22]
RRB-Select-Command - Each row of the detector contains 32 Row-Redundant-Bolometers (RRB). Each one of these bolometers can be either selected or deselected by this command. These bolometers on the left branch of the Wheatstone bridge (Fig.2.3), a.k.a "Blinds", are blocked from the scene and hence used together with the "Global" resistors as a reference to the right branch in order to measure minute changes in pixel resistance. The Global resistors are common for all the pixels in the matrix, and in order to keep the bridge balanced, the same number of Global and Blind resistors in each row must be chosen. The RRB map is unique for each detector. It is created during the factory detector characterization and supplied with the detector as a binary file with predefined format. [22]
Initialization-Command - This command defines detector initialization internal parameters and is rarely changed by the user. [22]
Configuration-Command - The system can configure different operation modes of the BIRD17-XGA detector. This command defines detector parameters such as timing, row and frame readout direction, overheat protection mode and other video output parameters. [22]
Setup-Command - This command allows the system to define the values of parameters, whose combination defines the detector gain. [22]
Compensation-Command - This command instructs the BIRD17-XGA detector to execute an internal process of compensation calculation. All the matrix pixels are required to line up around the working level (usually mid-voltage-range) after power-up by injecting "compensation" current. This action is referred to as compensation.

The compensation calculation algorithm should be processed with a shutter covering the FOV or in front of a uniform scene while the XGA-BIRD17 detector operates normally.

The compensation calculation algorithm should be processed every time the FOV temperature changes. [22]

For verification of the so far designed IP cores and C++ classes, first photo from the detector was taken. Figure 3.18 shows the first taken photo. In this figure, the effect of non-uniform temperature can directly be seen as fog. Vertical noise patterns are generated by the ROIC column amplifiers.

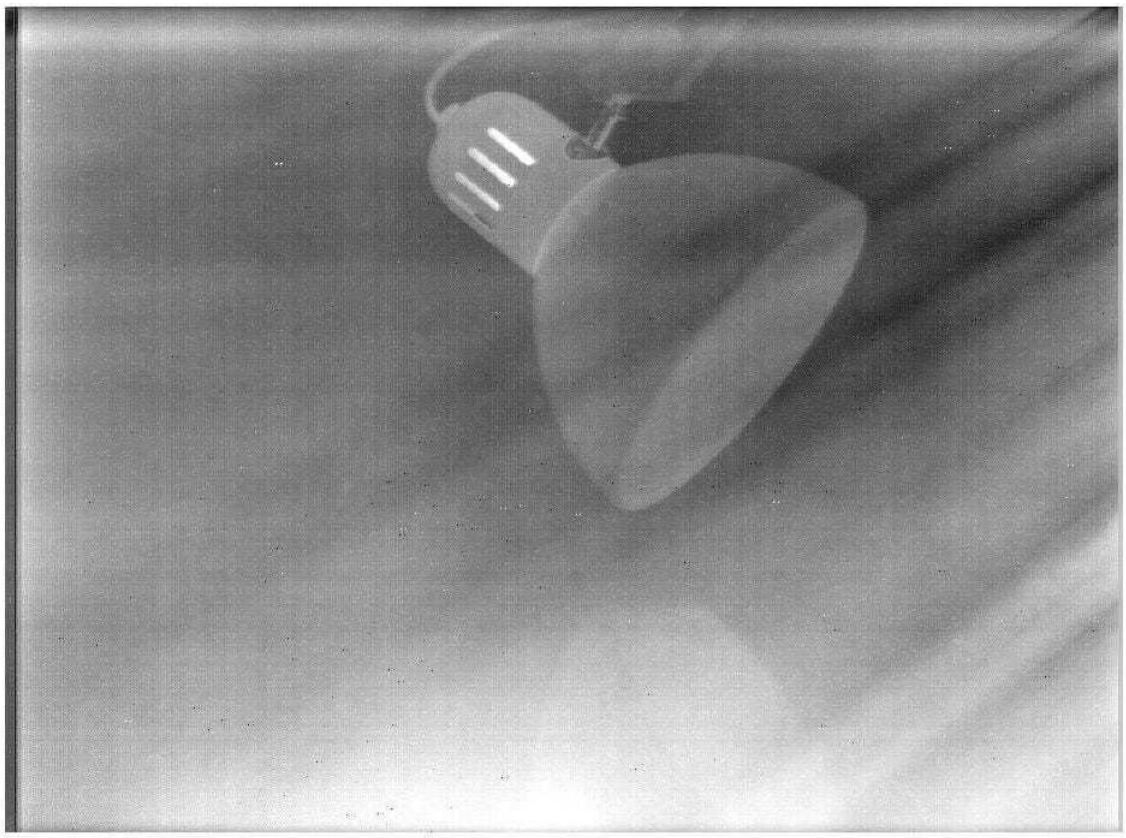

Fig. 3.18: The first taken photo

3.5 Temperature stabilization

The XGA-BIRD17 detector is equipped with a thermoelectric cooler (TEC) element which allows controlling the FPA temperature. Implementing the TEC controller allows the system to stabilize detector FPA at specific temperature which best suits the system requirements. The detector produces an image during the TEC stabilization, the image quality is deteriorated (Fig.3.18) until full stabilization is attained. [22]

Temperature of the TEC is read directly on the output pin of an internal diode for independent FPA temperature measurement. This pin is connected to ADC AD7680. The power of the TEC is driven by Pulse Width Modulation (PWM) signal. Duty cycle for the PWM is calculated by PID regulator based on the diode temperature.

Table 3.3 shows the output voltage dependent on a temperature of the temperature diode. With values from this table, equation for any temperature [°C] of FPA can be expressed as:

$$FPA_{temperature} = \frac{V_{diode} - 2.1385}{-0.01126} \tag{3.3}$$

Tab. 3.3: Temperature-Diode Specifications [21]

FPA temperature	**Min.**	**Typ.**	**Max.**
-40°C			2.4V
25°C		1.857V	
85°C	1.2V		

3.5.1 Temperature reading

IP core AD7680_v1

IP block AD7680_v1 was designed to interface with AD7680. AD7680 uses for communication only three pins: $SDATA$, $\overline{CS}$ and $SCLK$.

The conversion is initiated on the falling edge of $\overline{CS}$ and must remain low until at least 10 $SCLK$ falling edges. If $\overline{CS}$ is brought high at any time after the 10th $SCLK$ falling edge, but before the 20th $SCLK$ falling edge, the part remains powered up, but the conversion is terminated and $SDATA$ goes back into three-state. At least 20 serial clock cycles are required to complete the conversion and access the complete conversion result. Additionally, a total of 24 $SCLK$ cycles accesses the four trailing

zeros. $\overline{CS}$ may remains high until the next conversion or it may remains low until $\overline{CS}$ returns high sometime prior to the next conversion. [18]

In Figure 3.19 is the designed IP core AD7680_v1. Except for the ports used for interfacing with AD7680, it contains AXI4-Lite interface, *ext_trig*, $DATA$ and $VALID\_IRQ$. This IP core can work in two modes:

Trigger mode - The trigger mode is set if "External trigger" option is checked. In this mode *ext_trig* port is used for triggering the new conversion. Selecting the trigger edge is done by "External trigger on:" option.

Free-running mode - If "External trigger" option is unchecked, then the IP core runs in the Free-running mode. In this mode "Samples Per Second" parameter is applied. Conversion is triggered automatically and periodically with respect to "Samples Per Second".

The output $DATA$ port can be used for data reading if AXI4-Lite is not used. Port $VALID\_IRQ$ is used as a valid signal for validation if new data are available or as an IRQ signal for the PS.

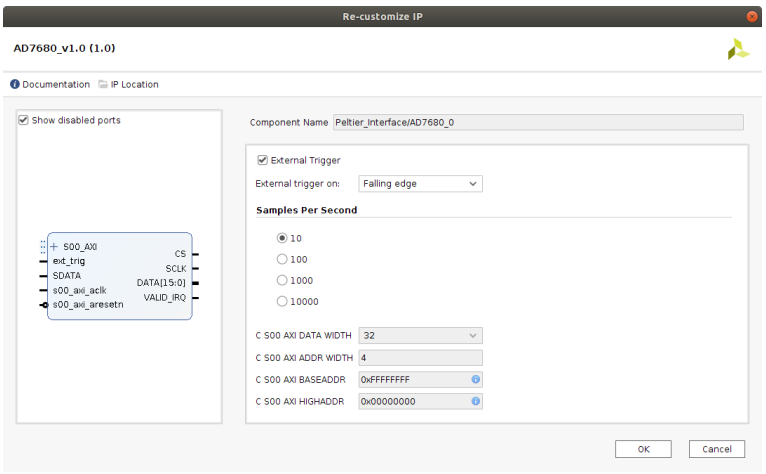

Fig. 3.19: AD7680_v1.0 IP core

ADC data are available via AXI4-Lite on the base address + offset. The IP core contains only one register for data (offset = 0x00). The register is 32-bit long, but only lower 16-bit are used (Tab.3.4).

Tab. 3.4: AD7680_v1 AXI4-Lite interface

Address Space Offset	Register Name	Access Type	Default Value
0x00000000	ADC_DATA	R_only	0x0000

Kernel module ad7680

For transmission of the ADC data from the PL to user space of the PS serves the kernel module called ad7680. The read-only attribute called read_ADC_reg0_r returns the ADC value. When $VALID\_IRQ$ pin is connected to IRQ_vector of the PS, the module can be registered with system functions "poll" or "select" and wait for IRQ.

Ad7680 class

Ad7680 class is used for communication with the kernel module. This class has two public methods for get raw ADC data or already converted data to voltage values. Reference voltage of the ADC is inserted in the constructor of the class.

3.5.2 TEC driver

TEC inside the XGA-BIRD17 is driven by H-bridge DRV592. DRV592 is a high-efficiency, high-current H-bridge ideal for driving a wide variety of thermoelectric cooler elements in systems. The DRV592 may be driven from any external PWM generator such as a microcontroller or an FPGA. [24]

IP core axi_pwm_v1.0

The PWM signal is generated by axi_pwm_v1.0 IP core. This IP core is controlled from PS via the AXI4-Lite bus. Table 3.5 shows the status/control registers.

Tab. 3.5: axi_pwm_v1.0 AXI4-Lite interface

Address Space Offset	Register Name	Access Type	Default Value
0x00000000	PWM_PERIOD_REG	R/W	0x0000
0x00000004	PWM0_REG	R/W	0x0000
0x00000008	PWM1_REG	R/W	0x0000
0x0000000C	STATUS_REG	R/W	0x0000

PWM_PERIOD_REG - This register is 32-bits wide and is used as clock divider for the reference clock of the IP core. For example, if loaded value = 100 and clock of the IP core = 100 MHz then PWM period will be 1 MHz.
PWM0/1_REG - This register is 32-bits wide and is used as a duty-cycle value. This value shouldn't be higher then the value in PWM_PERIOD_REG (duty cycle = 100 %).
STATUS_REG - This register is 32-bits wide but only the lowest 6 bits are used.

This register is used for turning on and off the PWM channels and watchdog. The status register is shown in table 3.6.

Tab. 3.6: axi_pwm_v1.0 AXI4-Lite interface

Bits	Name	Core Access	Reset Value	Description
31-6	Reserved	N/A	0	Reserved. Set zero on read.
5	Watchdog Reset	R/W	0	Rising edge resets watchdog.
4	Watchdog Enable	R/W	0	1 = Enable 0 = Disable
3	PWM1 Enable	R/W	0	1 = Enable 0 = Disable
2	PWM0 Enable	R/W	0	1 = Enable 0 = Disable
1	PWM global Enable	R/W	0	Control enable for bit 2-3.
0	Write Enable	R/W	0	1 = when register is updated with new values.

The watchdog is used as a software protection in case the PS gets stuck. Bit 5 of the STATUS_REG must be cleared at least every 5 seconds otherwise the PWM outputs are disabled.

The test bench can be found in Appendix B.4

Kernel module axipwmdrv

Axipwmdrv kernel module serves as a bridge between PL and PS user space. It contains 4 attributes for access to each axi_pwm_v1.0 IP core registers.

Axipwmdrv class

Direct configuration of the axi_pwm_v1.0 registers requires certain knowledge about this IP core. To eliminate these need, Axipwmdrv class was written as API for this IP core. For example, PWM period is set by the method "void setPeriod(const uint32_t period_us)". Input parameter is (unsigned integer) period in microseconds. From the input parameter, clock prescaler value is calculated and written into PWM_PERIOD_REG.

3.5.3 PID controller

Proportional Integral Derivative (PID) controller is a control loop mechanism employing feedback. PID controller continuously calculates an error value $e(t)$ as a difference between desired setpoint and measured process variable and it applies

correction based on proportional, integral, and derivative terms (denoted P, I, and D respectively). Mathematical form of the PID controller can be expressed as:

$$u(t) = K_p e(t) + K_i \int_0^t e(t')dt' + K_d \frac{de(t)}{dt} \tag{3.4}$$

where K_p, K_i, and K_d, all non-negative, denote the coefficients for the proportional, integral, and derivative terms. These coefficients for the XGA TEC were found by experimental method.

Pid class

Pid class was taken over from Photon System Instruments company library. This class was created by Tomáš Matějka for his Master thesis "Měřicí systém termoluminiscence"[25].

TEC (Peltier device) stabilization process is depicted in Fig.3.20. The system is successfully stabilized in 40 seconds and temperature is constant, equal to 25 °C.

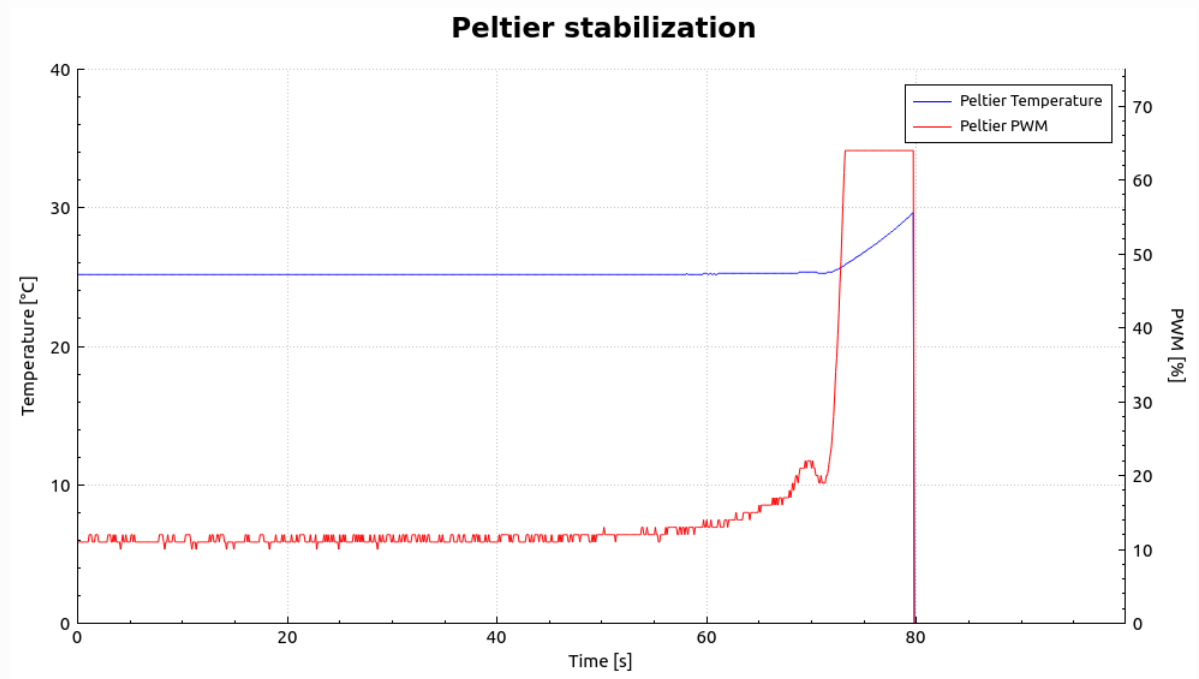

Fig. 3.20: TEC (Peltier) stabilization

3.6 CDS calibration

Correlated Double Sampling (CDS) removes the column non-uniformity caused by low-frequency column drift. This drift is caused primarily by the low frequency components of 1/f noise in the XGA-BIRD17 ROIC. An estimate of the drift is obtained by sampling the ROIC in the absence of signal. Special non-signal pixels are embedded into the BIRD output video inside the CDS window at the top of the frame (Fig.3.11). Up to 32 lines of CDS pixels can be generated by the XGA-BIRD17 detector. [22]

3.6.1 CDS_v1.0 IP core

CDS_v1.0 IP core follows the CDS algorithm described in [22] and adds some work-around.

1. Video data incoming from LTC2387_v1 IP core with sample frequency 180MHz are written into clock independent FIFO. The FIFO serves as CDC synchronization element because the reading frequency of the FIFO is 100 MHz.
2. Selecting number of generated CDS rows is defined by input parameter "CDS rows" (Fig.3.11). The same number must be set into the Configuration-command. Selecting higher number of CDS rows allows better correction. However, it might introduce the timing issues at high frequency operation, since it increases the frame time. The recommended number of CDS rows is 17 and therefore this number has also been used in this project.
3. Incoming CDS lines are read while ignoring the first and the last CDS video lines (the first is redundant and the last is ignored due to noise considerations). For example, with total 17 CDS rows, only 2÷16 rows are used.
4. The CDS vector (called $CDS_{current}$) is created by averaging the collected lines in every frame. But the vector, calculated with CDS rows of the first frame obtained after the compensation, should be stored as a reference CDS_{ref} until the next compensation.
5. The even rows (rows 0, 2, 4, ... 766) - the leftmost pixel is read through Top amplifier, the subsequent pixel is read through Bottom amplifier and so on.
   The odd rows (rows 1, 3, 5, ... 767) - the leftmost pixel is read through Bottom amplifier, the subsequent pixel is read through Top amplifier and so on.
   All CDS rows are treated as odd rows (Figure 3.21).

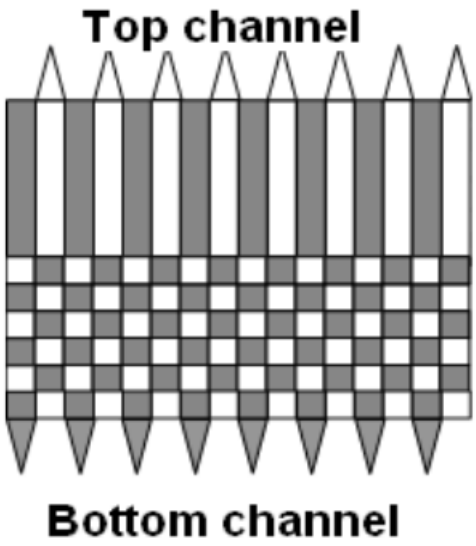

Fig. 3.21: ROIC amplifiers [22]

6. The compensation of the drift over time - the CDS correction- is done by using the column drift measurement $\Delta_{CDS} = CDS_{ref} - CDS_{current}$. The Δ_{CDS} vector is deduct from each data row. When the current data row is odd then the subtraction is done between pixels with the same indices (Fig.3.22.a). For the current data even row the subtraction is done by switching between odd and even pixels in the Δ_{CDS} vector (Fig.3.22.b).

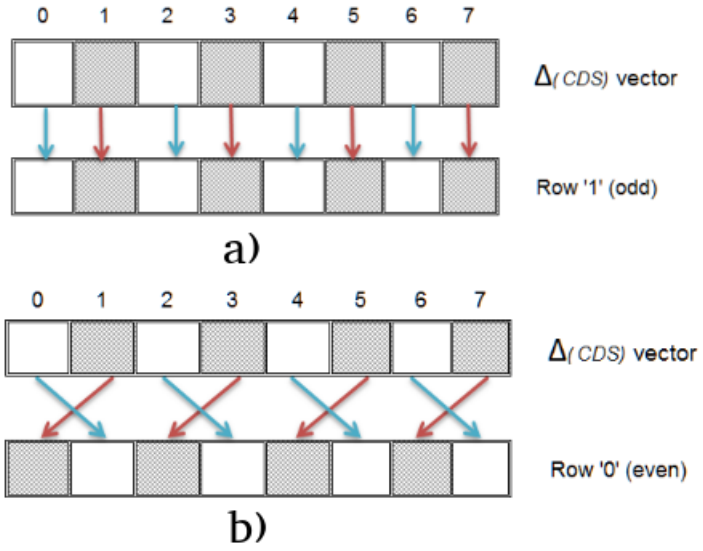

Fig. 3.22: CDS correction a)Odd row b)Even row [22]

7. For reducing the temporal noise, Infinite Impulse Response (IIR) filter was design for averaging the $CDS_{current}$ vector. The equation for the first order IIR filter is:

$$CDS_{current}[n] = \alpha \cdot CDS_{new} + (1 - \alpha) \cdot CDS_{current}[n - 1] \tag{3.5}$$

Multiplication is a very demanding operator in terms of area and timing of the FPGA resources. Eq.3.5 can be transformed in Eq.3.6 and multiplication can be replaced by bit shifting.

$$CDS_{current}[n] = CDS_{current}[n - 1] + (CDS_{new} - CDS_{current}[n - 1]) \cdot \alpha \tag{3.6}$$

The output data interface of this IP block is the AXI4-Stream. The IP core is able to enable or disable the CDS correction for easier debugging. Following Fig.3.23 shows the histograms of frames, where the pixel drift and pixel dispersion can be observed. The reference frame is blue line. Red line is the frame after 15 minutes with the CDS correction and orange line without the CDS correction. During the test, detector faced to closed shutter.

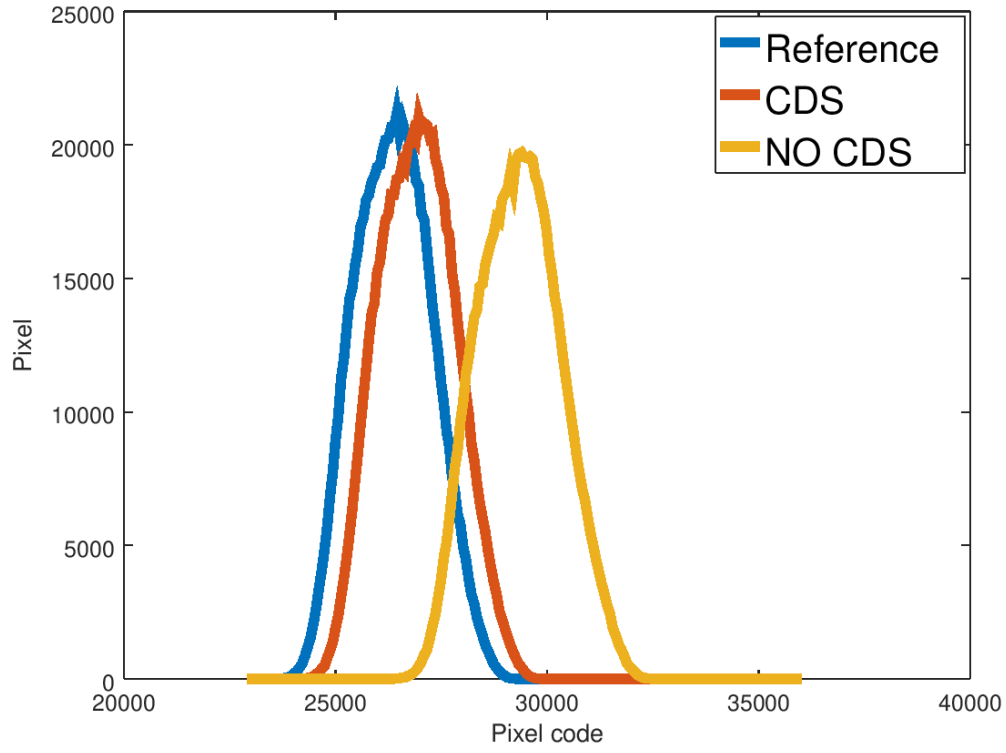

Fig. 3.23: CDS vs. NO CDS signal

3.7 Non-Uniform Correction

Due to manufacturing tolerances and geometry of the detector, the signal of each pixel differs from each other. In order to correct the build-in non-uniformity, compensation procedure must be be applied through UART Compensation-command which corrects the coarse non-uniformity. After the compensation, two-point calibration is realized.

Using the linear response of the detector pixels, it is possible to correct the raw signal non-uniformity by a linear transformation:

$$S_{ijcorrected} = G_{ij} \cdot (S_{ij} + O_{ij}) \tag{3.7}$$

Where S_{ij} is the signal of the pixel in i-th row and j-th column, G_{ij} is the gain coefficient and O_{ij} is the offset coefficient. This procedure is called the 2 point Non-Uniformity Correction (NUC). Figure 3.24 shows pixel signal values versus the scene temperature. The curve of each pixel differs by offset (β_{ij}) and its slope (α_{ij}).

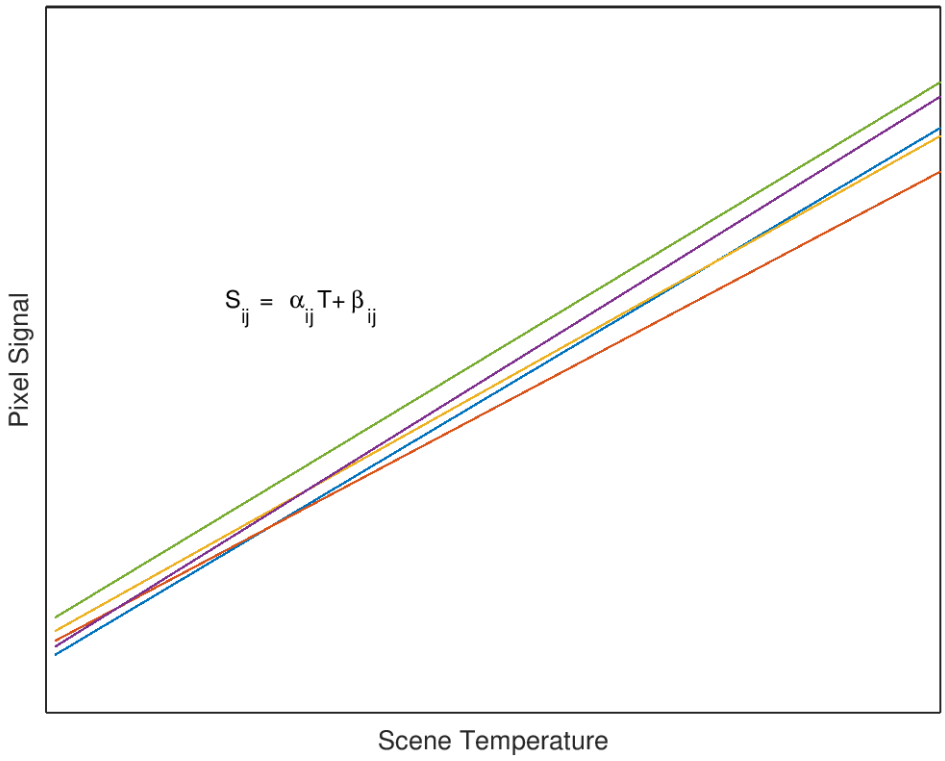

Fig. 3.24: Pixel signal vs. Scene temperature

2-point NUC should be performed within a linear span of detector for all the pixel values. The NUC coefficients (gain and offset) depend on the operating and setup condition of the detector (detector gain, RRB table, frame rate, optical path). Whenever is any of these parameters changed, NUC coefficients must be updated.

3.7.1 NUC calibration

For NUC calibration two reference temperatures are needed. Temperatures should have signal within the linear span of each non-defect pixel.

Black-Body

As the source of reference temperatures for NUC calibration so called "Black-Body" object (see Appendix D) was made. The temperature area of Black-Body is made of 10 mm thick copper which has very low temperature coefficient. Temperature of the copper is changed by thermoelectric coolers and read via resistance thermometer PT1000. As the control unit, Thermoluminescence unit [25] with few changes in software was used. The whole interior of the Black-Body is covered by special spray with defined emissivity. Table 3.7 summarizes the Black-Body parameters.

Tab. 3.7: Custom Black Body parameters

T_{min} [°C]	15
T_{max} [°C]	65
ΔT_{max} [s]	360
Emissivity [-]	0.96

Calibration procedure

The detector is placed in front of the uniform Black-Body (low or high temperature), facing it with a de-focused lens. Total 32 frames are recorded and averaged for noise reduction. The signal of each pixel is averaged over 32 frames. After data acquisition at low and high temperature, gain and offset can be calculated (Eqs. 3.8 and 3.9) as:

$$G_{ij} = \frac{\overline{S}_{2,ij} - \overline{S}_{1,ij}}{S_{2,ij} - S_{1,ij}} \tag{3.8}$$

$$O_{ij} = \frac{\overline{S}_{1,ij}}{G_{ij}} - S_{1,ij} \tag{3.9}$$

Where $\overline{S}_{ij}$ is the average signal over S_{ij} (signal of each pixel). After these calculations the NUC table is created and it can be applied.

It is recommended to perform 1 point NUC calibration to compensate any drift of the signal that might occur in time or temperature change in the optical path. In 1 point calibration is closed shutter and recorded, then average 8/16 frames. The averaged frame is used only for new offset calculation which updates the NUC table (gain is unchanged).

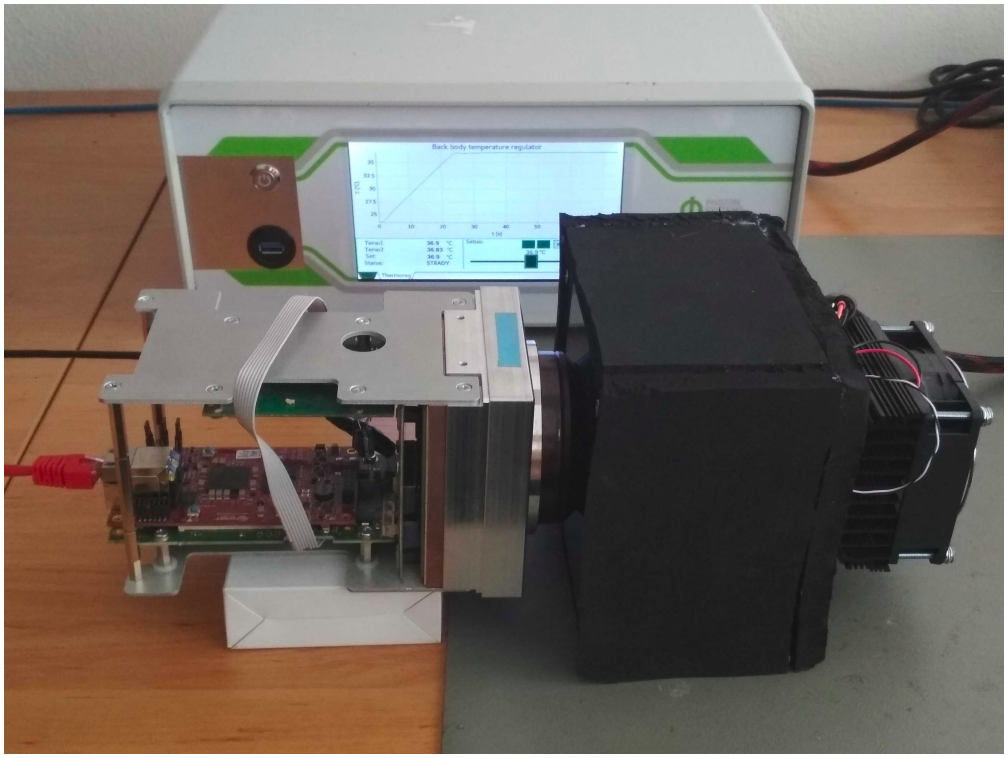

Fig. 3.25: NUC calibration process

3.7.2 NUC Hierarchy

The size of created NUC table is approximately 3.1 MB. Because of this size the table can't be stored in PL and must be put into PS. The NUC table is transferred via DMA over and over with the speed of 190 MB/s. The NUC coefficients (gain and offset) are read from the FIFO buffer into nuc_v1_0 IP core. The custom IP core nuc_v1_0 implements the NUC correction and checks the overflow or underflow of the arithmetic operations. For the arithmetic operations, format UQ1.15 is used. The NUC hierarchy is in Fig.3.26. The 1-point NUC calibration is done in PS every time the compensation is performed.

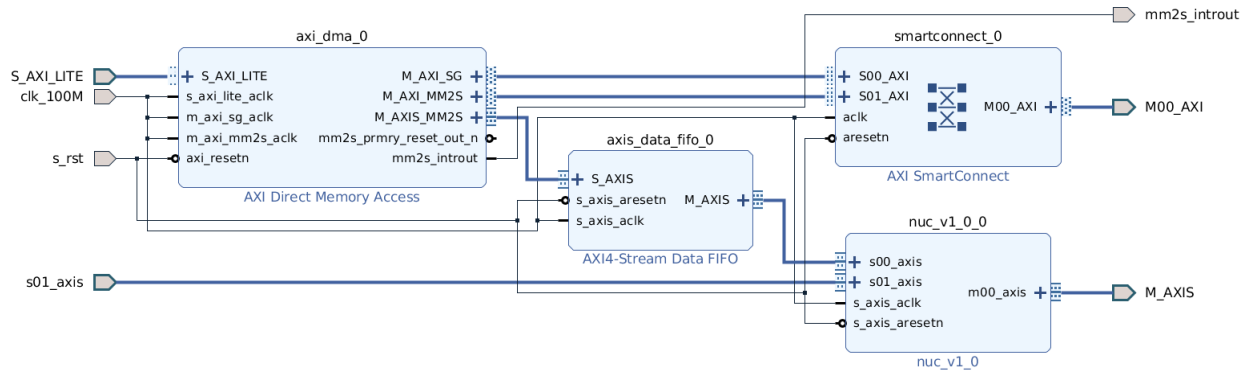

Fig. 3.26: NUC Hierarchy

After the NUC correction, image quality was significantly improved. Figure 3.27 shows the image with and without NUC correction.

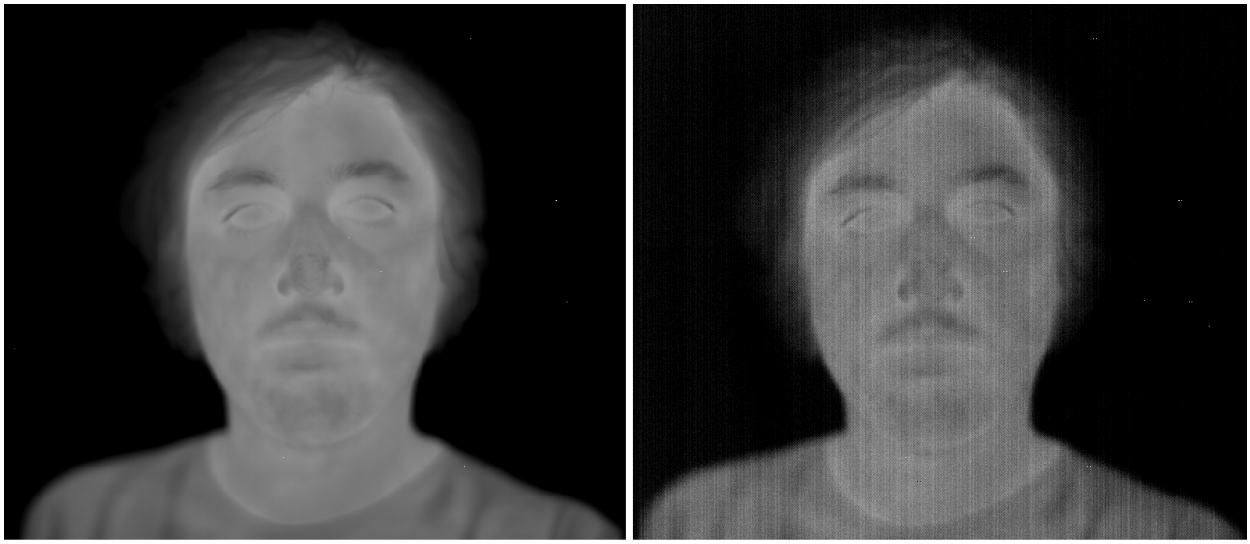

Fig. 3.27: a) Image with NUC b) Image without NUC

3.8 Bad Pixels Replacement

Every XGA-BIRD17 detector is supplied with the map of the defect pixels as a text file. The file includes the pair of coordinates of the defective pixels. The defect pixels are those, which have non-standard response on the IR signal.

Standard method for bad pixel replacement is replacing the bad pixels with average from the neighboring pixels.

3.8.1 DeadPixel class

DeadPixel class serves for bad pixel replacement. The class constructor takes the reference of the defect pixels table file and loads it in. For repairing actual frame serves method "template<typename T> void fixPixels(T data_ptr)", which takes the start-of-frame pointer as the input parameter. For correct averaging one must set set the correct frame dimensions via constructor or method.

3.9 XGA_control_app

XGA_control_app is an application running in PS as a process, which integrates all camera functions and takes care of communication with Graphical User Interface (GUI).

For communication with the camera, Transmission Control Protocol (TCP) is used. Because TCP is connection-oriented, camera is a server running on port 8888 and GUI is a client.

The camera communication interface with GUI on an application level is so-called task-driven. It means there is a master which sends tasks to slave, which after completing its task responds with status and optional data. The utilized protocol is called PSI protocol and it is Photon System Instruments internal protocol.

3.9.1 PsiProtServer class

PsiProtServer is a class based on C++ boost::asio library. It serves as TCP server class which works fully asynchronous. Data are forwarded via shared buffers and synchronized by mutexes and condition waits.

3.9.2 PsiProtParser class

PsiProtParser class is derived from PsiProtServer class and it is extended with PSI protocol. A callback function of the camera class for the PSI protocol handler is inserted into the class constructor.

3.9.3 XgaBird class

XgaBird class represents the XGA-BIRD17 detector. It integrates all described the properties in this thesis up to this point. For example, it integrates the Dmaeng class for video data receiving, Xgauartdrv class for communication with the detector, Pid controller class, DeadPixel class and etc.

For easy debugging of the program, custom exceptions are integrated in all classes, which are captured in the main function of the program.

Listing 3.1: XGA_control_app main loop

```
int main()
{
  try {
    XgaBird camera;

    PsiProtParser parser
    (
      TCP_PORT,
      std::bind(&XgaBird::xgaPsiProtocolParser,
      &camera,std::placeholders::_1, std::placeholders::_2)
    );

    parser.Start();

  }
  catch (std::exception &e) {
    std::cerr   <<"Captured␣exception␣in␣main:␣"
                << e.what() << endl;
  }
}
```

3.10 Resource utilization

Complete resource utilization of the camera design in PL is shown in Tab.3.8. The FPGA floor-planning view is depicted in Fig.3.28. Most of the resources have been used for the AXI DMA and FIFO buffers.

XGA_control_app control application uses approximately 70 percent of the processor for ensuring stable frame rate.

Tab. 3.8: FPGA resource utilization

Resource	Utilization	Available	Utilization %
LUT	15067	53200	28.32
LUTRAM	3458	17400	19.87
FF	20222	106400	19.01
BRAM	22	140	15.71
DSP	4	220	1.82
IO	41	125	32.80
BUFG	4	32	12.50
MMCM	1	4	25.00

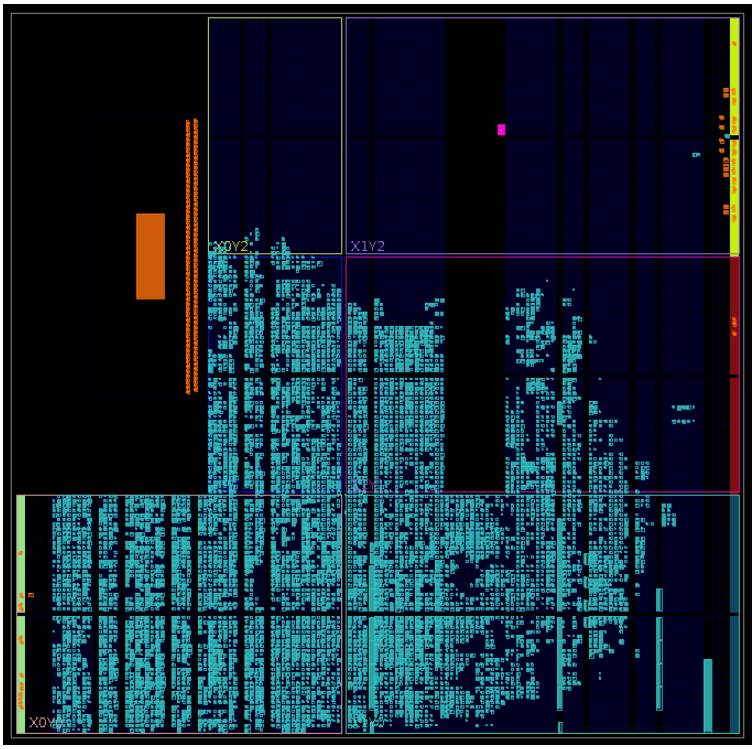

Fig. 3.28: FPGA floor-planning view after implementation

3.11 Graphical User Interface

For data visualization and detector configuration, GUI in Qt language was created. The GUI integrates TCP client with PSI protocol. The GUI contains 3 control tabs, where every tab offers different options.

1. **TCP** - Tab for TCP connect/disconnect. There must be inserted correct IP address and port of the camera.
2. **XGA** - Tab for configuration XGA parameters. For example, in this tab can be changed the Video delay parameter or number of CDS lines. Additionally, this tab is used for creating the NUC table.
3. **V/G** - Tab for interaction with the Video window and Peltier graph (Fig.3.20).

The Video window serves for the video data visualization. The Peltier window can be display temperature of the detector and PWM in the graph. The "Peltier stabilization" graph was created with QCustomPlot library.

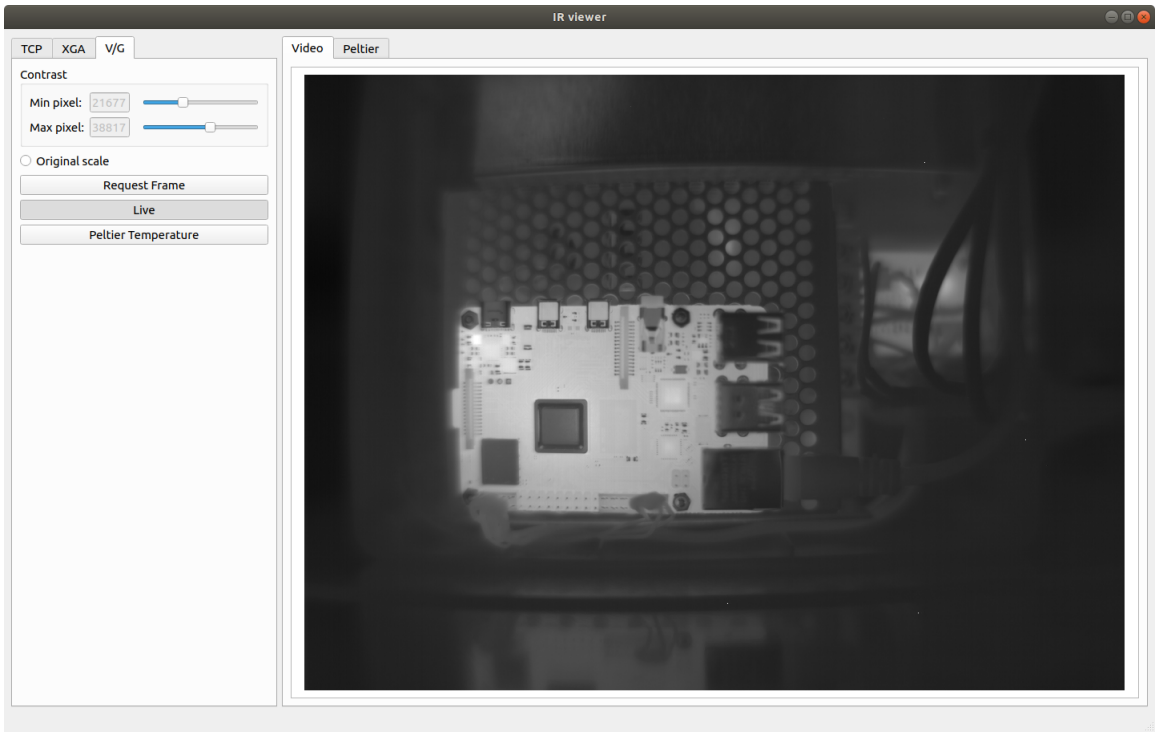

Fig. 3.29: GUI - IR viewer

Conclusion

A brief study of the IR radiation and thermovision has been made in chapter 1. The emissivity parameter has been introduced and so was its dependence on the measured temperature. In the thermovision section there have been described the most popular applications that use thermal cameras, followed by the description of basic division of thermal detectors and their main parameters.

In chapter 2 main hardware parts of the developed camera have been described. The IR detector XGA-BIRD17 and its properties have been studied in detail.

Chapter 3 describes design of the IP cores, kernel modules, C++ classes and GUI. A special care has been taken for the reusability of designed IP core/libraries/modules.

The first part of chapter 3 is focused on design of a kernel module for the DMA in order to ensure high video data throughput between PL and PS. It has been managed to design the universal DMA driver with throughput (375 MB/s) close to the theoretical value (400 MB/s).

Because XGA-BIRD17 has analog video outputs, external ADCs had to be used for digitization of the signal. An IP core for interfacing with LTC2387-16 ADC has been created. This IP core communicates with LTC2387-16 at speed 180 MHz DDR. Due to this speed and unconnected synchronization clock from LTC2387-16 (system synchronous clocking is used), the IP core had to be extended with synchronization circuit for finding ideal sampling moment by configuring the proper delay on IDELAY2 primitive.

The next part describes IP cores and classes for video interface and UART communication with XGA-BIRD17. The IP core for video interfacing takes care of the correct timing (start ADC conversion) of image reading. That is based on vertical and horizontal synchronization signal driven by the IR detector. For communication with the IR detector via UART, an IP core has been designed, which takes care of auto-detecting the baud-rate. In C++, class library for configuration of IR detector has been programed.

For the TEC integrated in XGA-BIRD17, temperature stabilization hierarchy block has been designed. For temperature reading from the IR detector via ADC AD7680, IP core, kernel module and class library have been developed. The TEC power is driven by PWM signal and for this purpose there have been also developed an IP core, kernel module and class library. The PWM value is calculated by PID controller. The maximum temperature stabilization time since the camera power-cycle has been measured to approx. 40 seconds.

At this stage of the work, video data has been visualized. Large temperature and time drift and strong non-uniformity could be observed in picture. The temperature and time drift have been suppressed by Correlated Double Sampling algorithm. This algorithm has been implemented into the CDS IP core. For the NUC calibration, custom Black-Body had to be made (see Appendix D). For video data improvement, measured NUC table corrects data in the designed IP core. After applying the NUC table, image quality has been significantly improved (see Appendix A). The last step in quality image improvement was removing the defect pixels. It has been done by a custom developed class library.

The last part of chapter 3 describes the data transfer via TCP protocol and its visualization in PC. The developed GUI is not a commercial product supplied with the camera. The GUI has been developed mainly for debugging, internal configuration and NUC calibration.

Unfortunately, the parameters of the IR camera have not been measured due to current version of the chassis (see Appendix C), which will be upgraded and its parameters changed. This upgrade should improve the thermal isolation between the FPA and the rest of the camera body, which is currently insufficient.

The developed software has been completed and tested. The IR camera has a stable frame-rate of 20 fps. The 20 fps limitation is caused by memory controller in Zynq, which works at full speed now. To increase the frame-rate, external SRAM should be added and NUC table moved in there.

Bibliography

[1] VOLLMER, Michael a Klaus-Peter MÖLLMANN. *Infrared Thermal Imaging: Fundamentals, Research and Applications.*. Second Edition. Germany: WILEY-VCH, 2018. ISBN 978-3-527-41351-5.

[2] *Emissivity Table for Infrared Thermometer Readings [online].* [cit. 2020-03-08]. Available from: <https://ennologic.com/wp-content/uploads/2018/07/Ultimate-Emissivity-Table.pdf>

[3] *Emissivity in the Infrared: Planck's Blackbody Distribution Law.* [online]. [cit. 2020-03-08]. Available from: <https://www.optotherm.com/infrared-planck.htm>

[4] *IR Detectors for Thermographic Imaging* [online]. [cit. 2020-03-15]. Available from: <http://www.techni-tool.com/site/ARTICLE_LIBRARY/FLIR%20-%20IR%20Detectors%20For%20Thermographic%20Imaging.pdf>

[5] *Zynq-7000 SoC Data Sheet: Overview* [online]. , 25 [cit. 2020-03-21]. Available from: <https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf>

[6] *AXI Reference Guide* [online]. , 82 [cit. 2020-03-21]. Available from: <https://www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf>

[7] *7 Series FPGAs SelectIO Resources* [online]. , 188 [cit. 2020-03-21]. Available from: <https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf>

[8] *MicroZed™ Getting Started Guide* [online]. , 42 [cit. 2020-03-21]. Available from: <http://zedboard.org/sites/default/files/documentations/MicroZed_GettingStarted_v1_2.pdf>

[9] *Thermovision measurements. BOSMAL* [online]. [cit. 2020-03-30]. Available from: <https://www.citacepro.com/dok/xNm6qRebZHydzOFi>

[10] *PlantScreen Imagining sensors. PSI* [online]. [cit. 2020-03-30]. Available from: <http://plantphenotyping.com/products/plantscreen-imaging-sensors/#thermalcam>

[11] *Oprah's buddy Dr. Christiane Northrup and breast thermography: The opportunistic promotion of quackery* GORSKI David [online]. [cit.

2020-03-30]. Available from: <https://sciencebasedmedicine.org/dr-christiane-northrup-and-breast-thermography-the-opportunistic-promotion-o

[12] *XGA 1024×768 Scientific/security analysis FLIR HD Thermal Infrared Imaging cameras* [online]. [cit. 2020-03-30]. Available from: <https://www.x20.org/scientific-analysis-flir-thermal-infrared-imaging-cameras/>

[13] *PetaLinux ToolsDocumentation* [online]. [cit. 2020-03-30]. Available from: <https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug1144-petalinux-tools-reference-guide.pdf>

[14] *Vivado Design Suite AXI Reference Guide* [online]. [cit. 2020-03-30]. Available from: <https://www.x20.org/scientific-analysis-flir-thermal-infrared-imaging-cameras/>

[15] Xilinx, Inc. [online]. [cit. 2020-03-30] Available from: *Vivado Design Suite AXI Reference Guide..*2017. <https://www.xilinx.com/support/documentation/ip_documentation/axi_ref_guide/latest/ug1037-vivado-axi-reference-guide.pdf>

[16] INFRA RED DETECTORS [online]. Bombay, 2004 [cit. 2020-04-21]. Available from: <https://www.ee.iitb.ac.in/~esgroup/es_mtech04_sem/es_sem04_paper_04307417.pdf.> Seminar Report. IIT Bombay. Supervisor Prof. U. B. Desai.

[17] Linear Technology Inc. [online technical-documentation]. [cit. 2020-04-23]. Available from: <https://www.analog.com/media/en/technical-documentation/data-sheets/238718fa.pdf>

[18] Analog Devices, "AD7680"[online technical-documentation]. [cit. 2020-04-23]. Available from: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD7680.pdf>

[19] Xilinx, "LogiCORE IP Utility IDELAYCTRLLogic (v1.0)" [online technical-documentation]. [2020-04-21]. Available from: <https://www.xilinx.com/support/documentation/ip_documentation/util_idelay_ctrl/v1_0/pb044-util-idelay-ctrl.pdf>

[20] Xilinx, "7 Series FPGAs SelectIO Resources" [online technical-documentation]. [cit. 2020-04-21]. Available from: <https://www.xilinx.com/support/documentation/user_guides/ug471_7Series_SelectIO.pdf>

[21] SCD, “BIRD17-XGA Electrical ICD” [online technical-documentation]. [cit. 2020-04-28]. Available from: <https://www.scd.co.il/products/bird-xga-17/>

[22] SCD, “Uncooled BIRD17-XGA Detector User Guide” [online technical-documentation]. [cit. 2020-04-28]. Available from: <https://www.scd.co.il/products/bird-xga-17/>

[23] Xilinx, “AXI UART Lite v2.0” [online technical-documentation]. [cit. 2020-04-28]. Available from: <https://www.xilinx.com/support/documentation/ip_documentation/axi_uartlite/v2_0/pg142-axi-uartlite.pdf>

[24] Texas Instruments, "DRV592"[online technical-documentation]. [cit. 2020-05-1]. Available from: <http://www.ti.com/lit/ds/slos390a/slos390a.pdf?&ts=1589119071658>

[25] MATEJKA Tomas. *Měřicí systém termoluminiscence*. [cit. 2020-05-1] Available from: <https://www.vutbr.cz/studenti/zav-prace?zp_id=110435>

List of symbols, physical constants and abbreviations

λ	Wavelength
c	Speed of propagation
f	Frequency
ADC	Analog-Digital converter
API	Application Programming Interface
CDS	Correlated Double Sampling
CDC	Clock Domain Crossing
CLB	Configurable Logic Block
DDR	Double Date Rate
DMA	Direct Memory Access
EM	Electromagnetic
FPA	Focal Plain Array
FSM	Finite State Machine
GUI	Graphical User Interface
IIR	Infinite Impulse Response
ILA	Integrated Logic Analyzer
IP	Intellectual Property
IR	InfraRed
LW	Long Wave IR
LVDS	Low Voltage Differential Signal
MW	Mid Wave IR
NETD	Noise Equivalent Temperature Difference
NUC	Non-Uniform Correction
PID	Proportional Integral Derivative
PLL	Phase Lock Loop
PSI	Photon Systems Instruments
PWM	Pulse Width Modulation
QWIP	Quantum Well Infrared Photodetector
ROIC	Read-Out Integrated Circuit
RRB	Row Redundant Bolometers
SPI	Serial Peripheral Interface
SW	Short Wave IR
TCP	Transmission Control Protocol
TEC	ThermoElectronic Cooling
UART	Universal Asynchronous Receiver-Transmitter

UV	UltraViolet
VIS	Visible Light
TEC	Thermo-Electric-Cooler

List of appendices

A Example IR images

Fig. A.1: IR image - flower with sick leafs (grayscale)

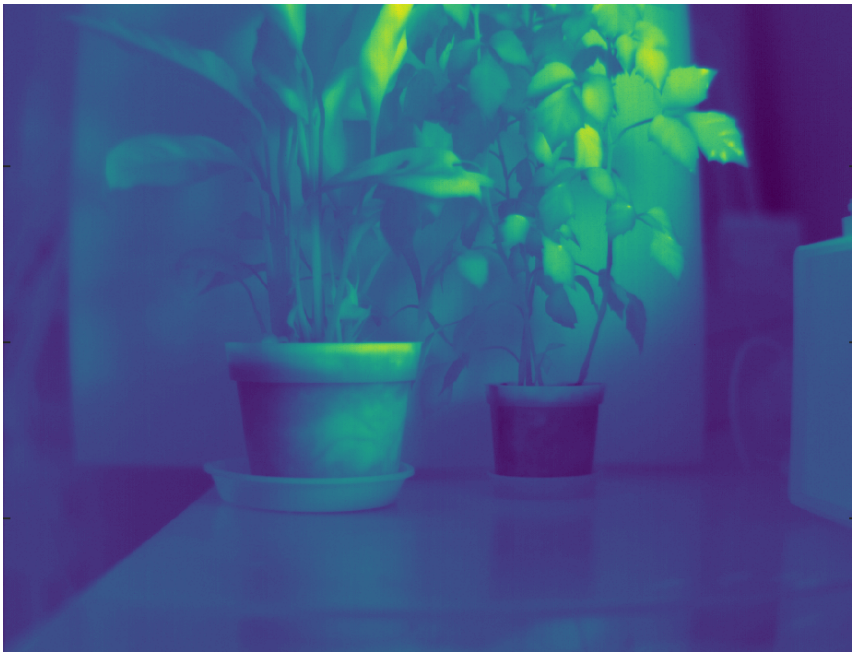

Fig. A.2: IR image - flower with sick leafs (colorscale)

Fig. A.3: IR image - veins (greyscale)

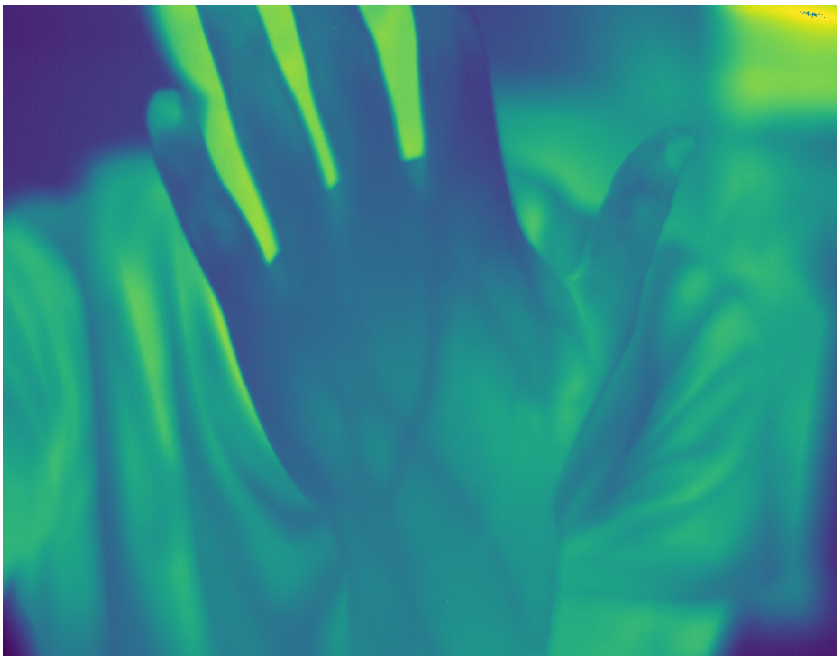

Fig. A.4: IR image - veins (colorscale)

B Vivado block desing and test benches

B.1 Vivado Block Design

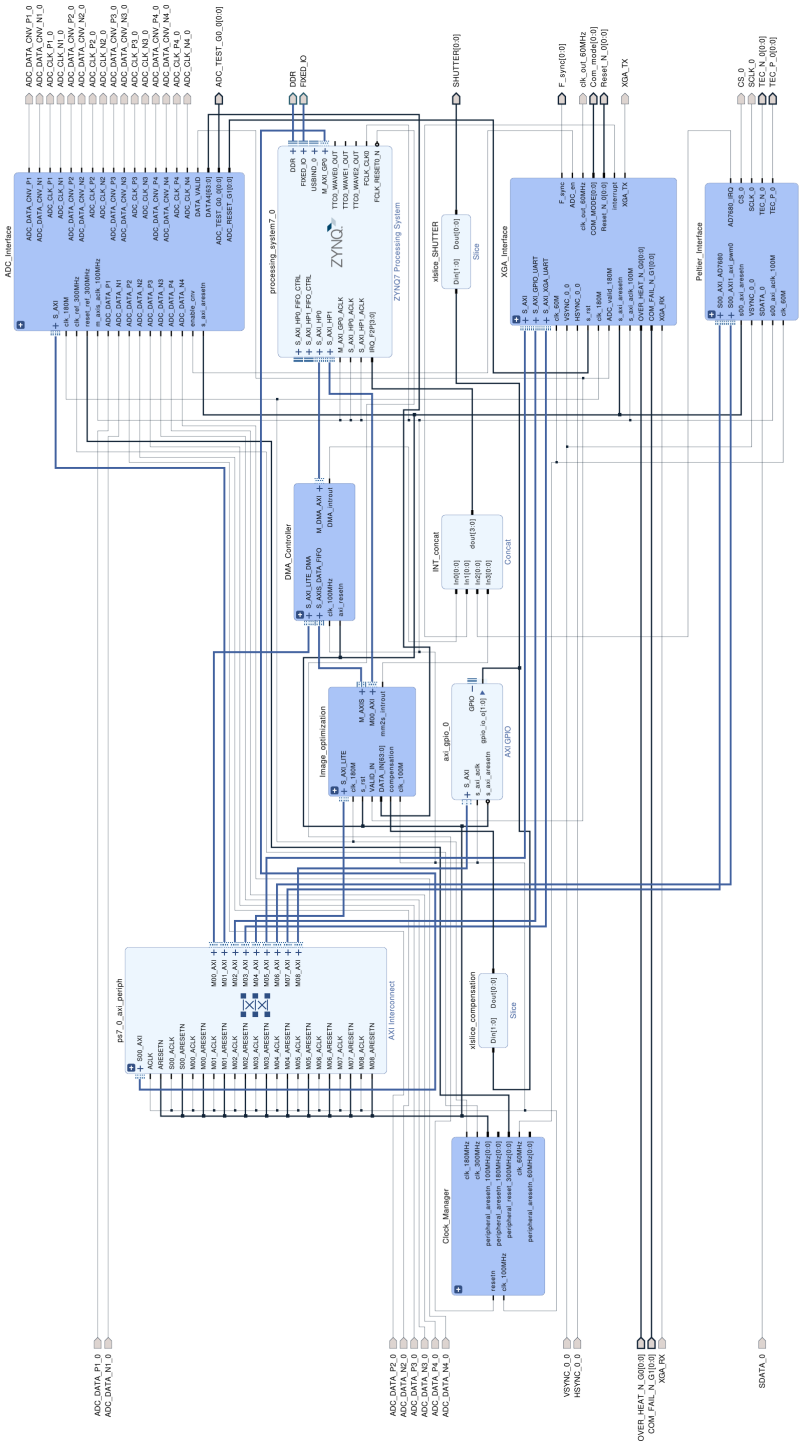

Fig. B.1: Final Vivado Block Design

B.2 XGA_driver_v2 + LTC2387_v1 Test Bench

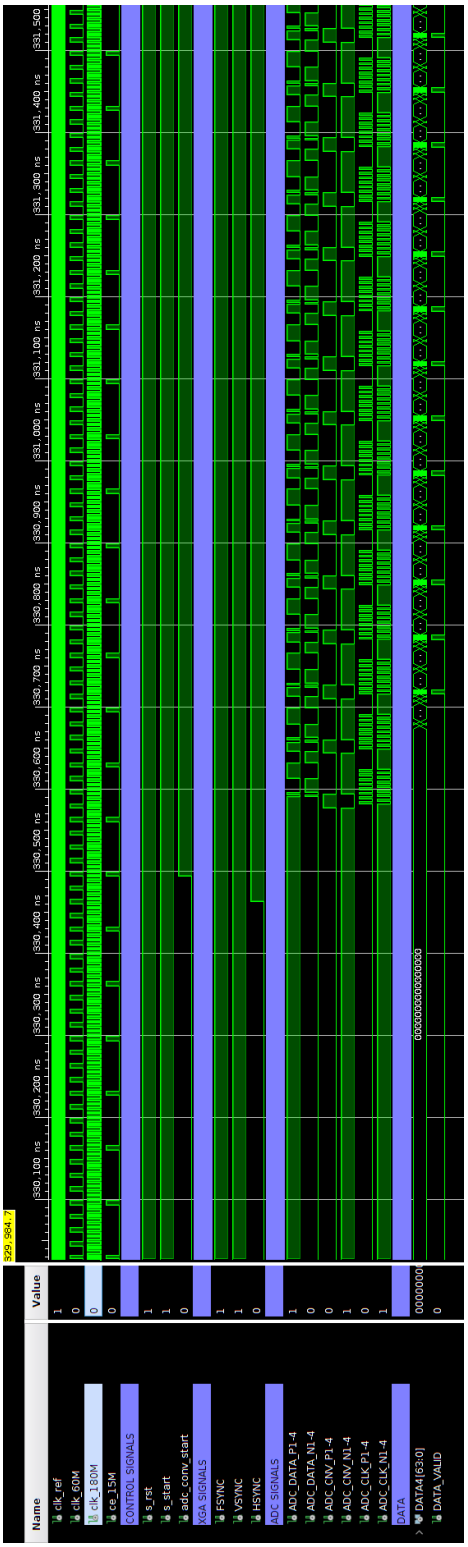

Fig. B.2: XGA_driver_v2 + LTC2387_v1 Test Bench

B.3 XGA_UART_Synchronisation_v1_0 Test Bench

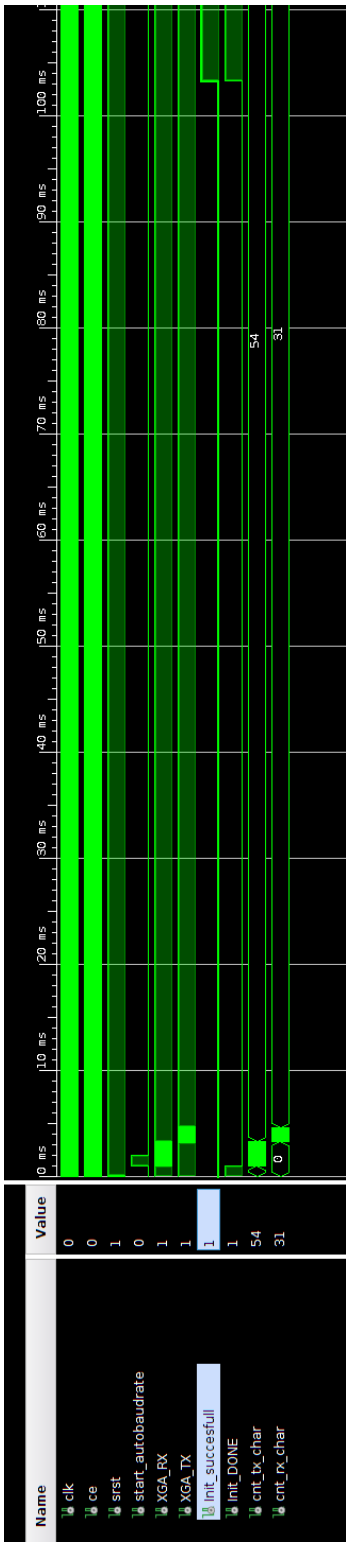

Fig. B.3: XGA_UART_Synchronisation_v1_0 Test Bench

B.4 axi_pwm_v1.0 Test Bench

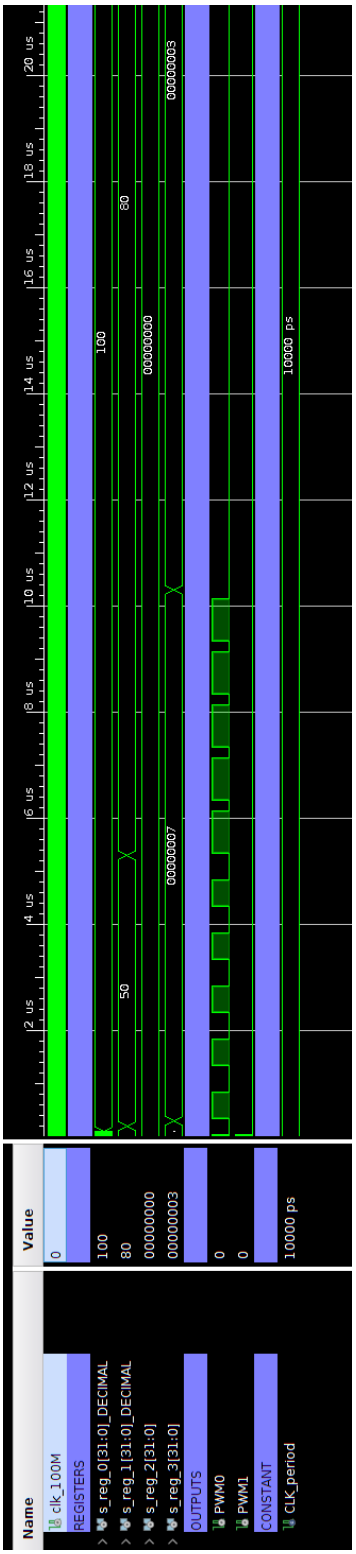

Fig. B.4: axi_pwm_v1.0 Test Bench

C IR camera chassis

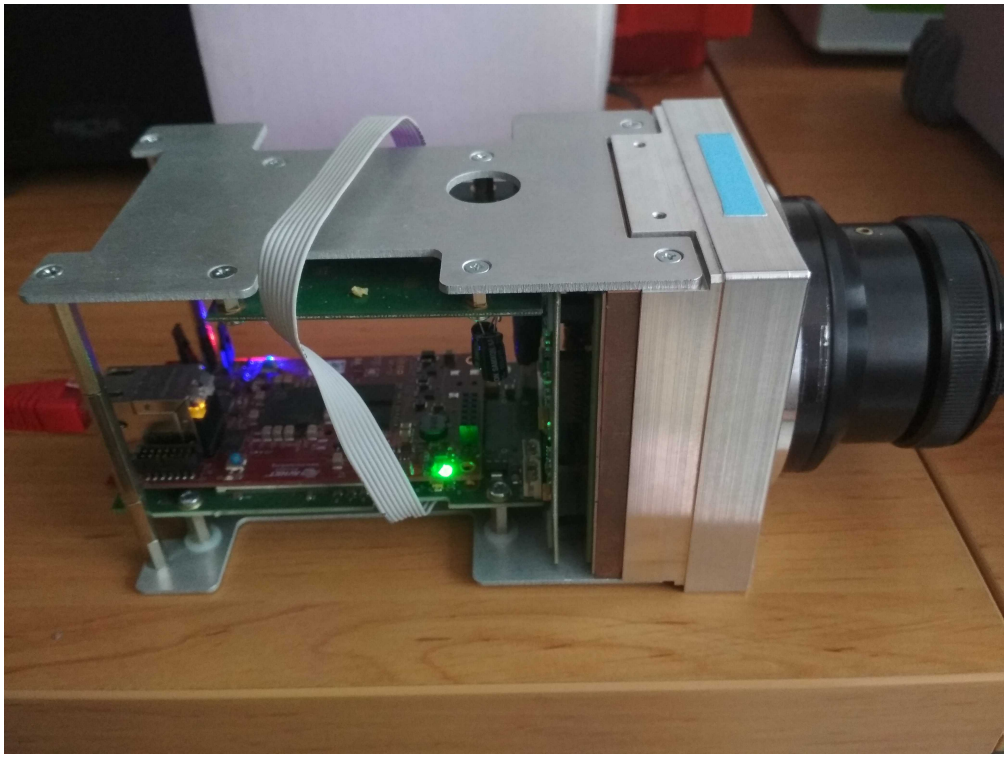

Fig. C.1: Actual chassis

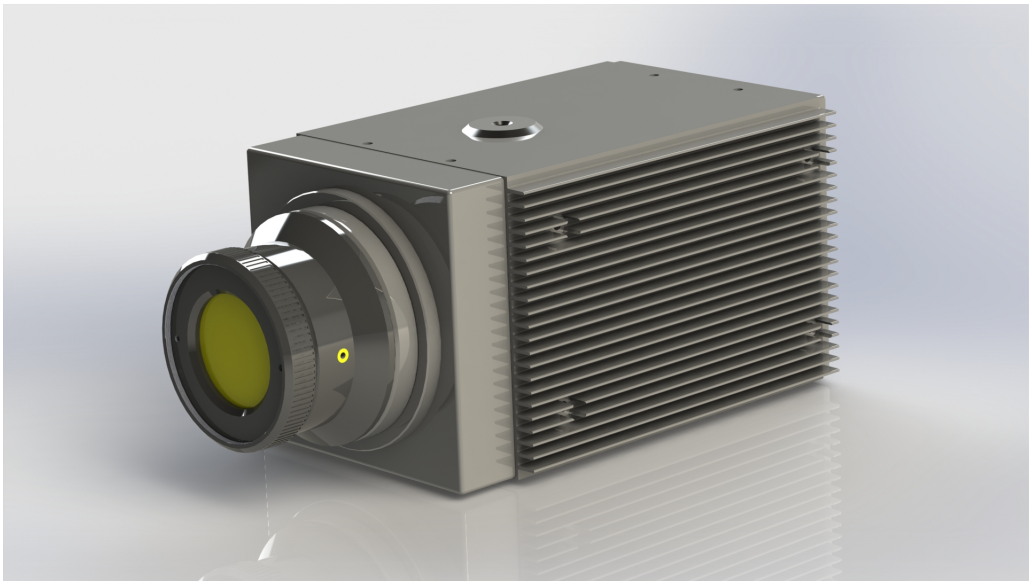

Fig. C.2: Upgraded chassis

D Custom Black-Body

Fig. D.1: Black-Body

E Contents of the attached CD

- /
  - Application project
    - readme.txt ..... git link
  - Example pictures
  - GUI
    - readme.txt ..... git link
  - Octave scripts
    - Plank_Distribution
      - Band_emission.m
      - Plank_Dis_MAIN.m
      - Planks_law_MAIN.m
    - NUC_static
      - main.m
      - nuc_low_temperature_octave_18_noNUC.txt
      - nuc_low_temperature_octave_18_NUC.txt
      - nuc_low_temperature_octave_30_noNUC.txt
      - nuc_low_temperature_octave_30_NUC.txt
      - nuc_low_temperature_octave_60_noNUC.txt
      - nuc_low_temperature_octave_60_NUC.txt
    - CDS_drift
      - main.m
      - CDS_15min.txt
      - CDS_COMP.txt
      - NOCDS_15min.txt
  - Petalinux project
    - readme.txt ..... git link
  - text
    - Latex ..... Latex source files
    - DP_Macura_ele.pdf
  - Vivado project
    - readme.txt ..... git link