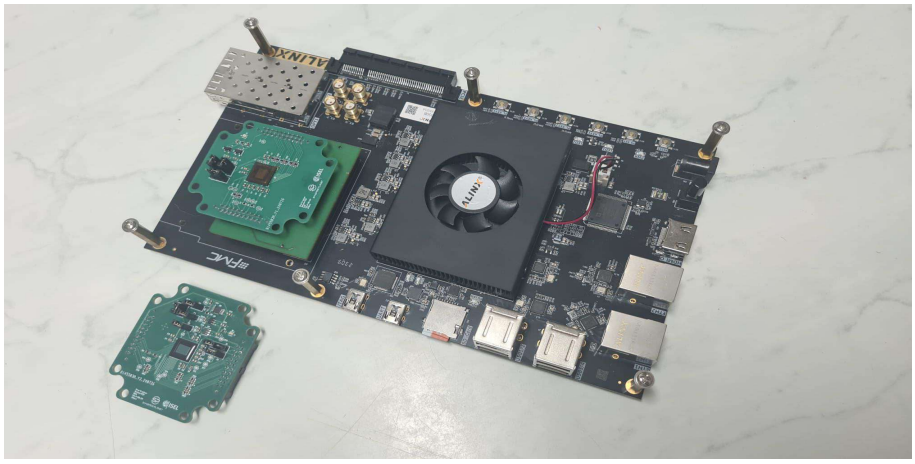
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia Electrónica e Telecomunicações e de Computadores



Star Tracker Platform

João Carlos Garcia Maia

(Licenciado)

Trabalho de Projeto para obtenção do grau de Mestre
em Engenharia de Electrónica e Telecomunicações, no Perfil de Electrónica

Orientadores : Doutor João Casaleiro
Doutor Paulo Gordo
Doutor Rui Melicio

Júri:

Presidente: Doutor Rui Duarte

Vogais: Doutor Vitor Costa
Doutor João Casaleiro

Novembro de 2024

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Departamento de Engenharia Electrónica e Telecomunicações e de Computadores

Star Tracker Platform

João Carlos Garcia Maia

(Licenciado)

Trabalho de Projeto para obtenção do grau de Mestre
em Engenharia de Electrónica e Telecomunicações, no Perfil de Electrónica

Orientadores : Doutor João Casaleiro (ISEL)
Doutor Paulo Gordo (FCUL)
Doutor Rui Melicio (IDMEC)

Júri:

Presidente: Doutor Rui Duarte (ISEL)

Vogais: Doutor Vitor Costa (ISEL)
Doutor João Casaleiro (ISEL)

Novembro de 2024

Acknowledgments

Em primeiro lugar, gostaria de estender a minha mais profunda gratidão ao Eng. Paulo Gordo e ao Eng. João Casaleiro. O seu apoio e incentivo inabaláveis foram fundamentais enquanto eu estava nesta viagem.

Um sincero agradecimento à Synopsis Planet por não só acreditar na minha visão, mas também por a apoiar financeiramente e com equipamentos, permitindo-me realizar os meus ensaios diretamente a partir do conforto da minha casa.

Sou grato ao CEDET – Centro de Estudos e Desenvolvimento da Electrónica e das Telecomunicações. A sua generosidade no fornecimento de instalações, equipamentos e componentes foi crucial para o desenvolvimento, montagem e testes dos meus protótipos.

Refletindo sobre os meus últimos cinco anos no ISEL, sinto-me cheio de gratidão pela educação que recebi, pelos professores e pelos amigos de longa data que fiz ao longo do caminho.

Numa nota mais pessoal, à minha família tem sido o meu apoio inabalável. À minha mãe, Teresa, e ao meu pai, Carlos, o seu apoio e inspiração transformaram-me na pessoa que sou hoje. O meu pai, em particular, tem sido um guia com conselhos e insights inestimáveis, tornando a minha jornada académica mais tranquila. Minha irmã Ana, obrigada por me suportares e seres o meu pilar de calma em dias de tempestade. Ainda à minha tia Cristina, a sua paciência e capacidade de me trazer um sorriso ao rosto foram uma bênção ao longo destes anos.

Um agradecimento especial aos meus colegas de faculdade do ISEL - André, Francisco, Marco, Rafael e Rodrigo. Vocês têm sido a minha segunda família, oferecendo apoio e partilhando inúmeras risadas, tornando estes anos inesquecíveis.

À minha terra natal, Torres Novas, e aos amigos de infância que aí fiz - Bárbara,

Cerdeira, Isaac e Lídia - que me acompanharam nos altos e baixos destes tantos anos. E para a Carolina, o Joel e o Roberto, que fazem parte do meu percurso desde que começámos na ginástica artística, a amizade deles significa muito para mim.

Por último, quero agradecer a todos os que fizeram parte da minha jornada de 24 anos de vida, mas que não foram mencionados pelo nome: os meus professores, colegas e aqueles amigos que, apesar da distância, deram uma contributo significativo impacto na minha vida. Do fundo do coração, obrigado a todos por tudo.

Abstract

This master's thesis aims to advance in the area of star tracking platform through the development and testing of hardware for a star tracker prototype. A comprehensive study of the components that make up the star tracker sensor was undertaken, analyzing typologies and materials available on the market is carried out, including sensors, microprocessors, FPGAs and memories. This analysis provided a deeper understanding of the advantages and limitations associated with each component, contributing to a better understanding of their applicability in the development of the proposed hardware. The project focused on the design and layout of the CMOS sensor AR0134CS PCB for both its BGA and LCC packages, enabling the use of the HiSPi interface and a 12-bit parallel interface. The HiSPi interface underwent electromagnetic simulation to assess signal integrity in differential pairs, passing with a significant margin. The sensor was produced, assembled, and integrated into an FPGA-based platform for testing. Configuration and operation were validated through clock signal prints and two-wire interface communication. Using specific resolution and pixel clock settings, the sensor achieved a frame rate of 10 frames per second, confirmed by theoretical calculations and experimental verification. Additionally, the FPGA board schematic was developed, ensuring the feasibility of component placement and interconnections on the PCB stack. While the layout design was not optimized for signal integrity, the schematic verified the functional integration of the hardware. This work successfully produced and validated a hardware prototype of the star tracker sensor, measuring 51 x 57 mm, laying the foundation for future improvements in star tracking technology.

Keywords: Star Tracker, CMOS image sensor, FPGA, Parallel Interface, HiSPi, HDMI.

Resumo

A presente dissertação de mestrado pretende avançar na área dos sensores de atitude através do desenvolvimento e teste de *hardware* para um protótipo de *star tracker*. Foi realizado um estudo detalhado dos componentes que constituem o sensor *star tracker*, analisando topologias e materiais disponíveis no mercado, incluindo sensores, microprocessadores, FPGA e memórias. O estudo realizado examina cuidadosamente as vantagens e limitações de cada componente, contribuindo para uma melhor compreensão da sua aplicabilidade no desenvolvimento do *hardware* proposto. O projeto centrou-se no desenvolvimento dos esquemáticos e no *layout* da PCB para o sensor CMOS AR0134CS, tanto nos formatos BGA como LCC, permitindo a utilização da interface HiSPi e da interface paralela de 12 bits. A interface HiSPi foi submetida a simulações de compatibilidade eletromagnéticas para avaliar a integridade do sinal nos pares diferenciais, tendo passado com uma margem significativa. O sensor foi produzido, montado e integrado numa plataforma baseada em FPGA para teste. A configuração e o funcionamento foram validados através da visualização dos sinais de *clock* e da comunicação por *two-wired interface*. Com base em definições específicas de resolução e frequência de *pixel clock*, o sensor alcançou uma taxa de 10 *frames per second*, conforme verificado por cálculos teóricos e validado experimentalmente. Adicionalmente, foi desenvolvido o esquemático para a placa da FPGA, assegurando a viabilidade do posicionamento e interconexão dos componentes da PCB. Embora o *layout* não tenha sido otimizado para integridade de sinal, o esquema verificou a integração funcional do *hardware*. Este trabalho resultou na produção e validação de um protótipo funcional de *hardware* do sensor *star tracker*, com dimensões de 51 x 57 mm, estabelecendo uma base sólida para futuros avanços na tecnologia de deteção estelar.

Palavras-chave: Star Tracker, sensor de imagem CMOS, FPGA, Parallel Interface, HiSPi, HDMI.

Contents

List of Figures

List of Tables

Acronyms

ADC	Analogic-Digital Converter.
ADCS	Attittude Determination and Control System.
ARM	Advanced Risc Machine.
ASIC	Application-Specific Integrated Circuit.
AXI	Advanced eXtencible Interface.
BGA	Ball Grid Array.
CAN	Controller Area Network.
CCD	Charge Couple Device.
CMOS	Complementary metal-oxide-semiconductor.
CPU	Central Processing Unit.
CSI-2	Camera Serial Interface-2.
DC	Direct Current.
DDR3	Double Data Rate version 3.
DMA	Direct Memory Access.
DSP	Digital Signal Processor.
ECI	Earth-Centered Inertial.
EGSE	Electrical Ground Support Equipment.
eMMC	Embedded Multimedia Card.
ESD	Electrostatic Discharge.
FMC	FPGA Mezzanine Card.
FoV	Field of View.
FOV	Field of View.
FPGA	Field-Programmable Gate Array.
FRAM	Ferroelectric Random Access Memory.
GPS	Global Positioning System.

HDMI High-Definition Multimedia Interface.
HiSPi HighSpeed Serial Pixel Interface.
I2C Inter-Integrated Circuit.
IO Input Output.
LCC Lead-less chip carrier.
LEO Low Earth Orbit.
LIDAR Light Detection and Ranging.
LVDS Low Voltage Differential Signaling.
MST Miniature Star Tracker.
OBCS On-Board Computer System.
OS Operationg System.
PCB Printed Circuit Board.
PL Programmable Logic.
PMU Power Management Unit.
PRR Plano de Recuperação e Resiliência.
PS Processing System.
QUEST Quaternion Estimator Algorithm.
SDR Software Defined Radio.
SDRAM Synchronous Dynamic Random-Access Memory.
SEE Single Event Effect.
SLVS Scalable Low Voltage Serial.
SoC System-on-Chip.
SP Synopsis Planet.
SPI Serial Peripheral Interface.
SRAM Static Random Access Memory.
UART Universal Asynchronous Receiver-Transmitter.
VGA Video Graphics Array.

1

Introduction

In this chapter, the motivations for the development of the hardware platform are reported, and it is revealed not only the structure, as well as the methodology considered in the preparation of this dissertation.

1.1 Motivation

In the ever-expanding realm of space missions, satellite attitude determination stands as a pivotal element, guiding navigation, maneuvering thrusters, and aligning antennas and scientific payloads. Diverse sensors, such as sun sensors, horizon sensors, magnetometers, and star sensors, contribute to a comprehensive Attittude Determination and Control System (ADCS).

Among these sensors, star trackers emerge as the paragon of precision, particularly crucial for high-precision scientific objectives that demand meticulous pointing of payloads. The task of determining a satellite's attitude reaches its zenith with star trackers instrument, which excel in accuracy [1]. These instruments play a crucial role in ensuring the success of agile satellites capable of swift maneuvers. The agility not only enhances specialized tasks but also significantly boosts the collection of earth observation and space science mission data, elevating overall efficiency [2].

The evolution of CubeSat standards has revolutionized nano satellites, opening doors to experiments and applications in Low Earth Orbit (LEO). Figure 1.1 depicting

the quantity of nano satellites launched since the emergence of the CubeSat standard in 1998 showcases a burgeoning trend, fueled by the low cost, increased launch opportunities, and standardized modules [3] [4]. As these satellites delve into applications like communication repeaters, terrestrial monitoring, astronomy, and more, the importance of efficient attitude determination and control systems is underscored. Here, star trackers, or star sensors, emerge as key components, facilitating robust communication, image acquisition systems for astronomical and terrestrial purposes, and efficient energy capture through precise solar panel positioning.

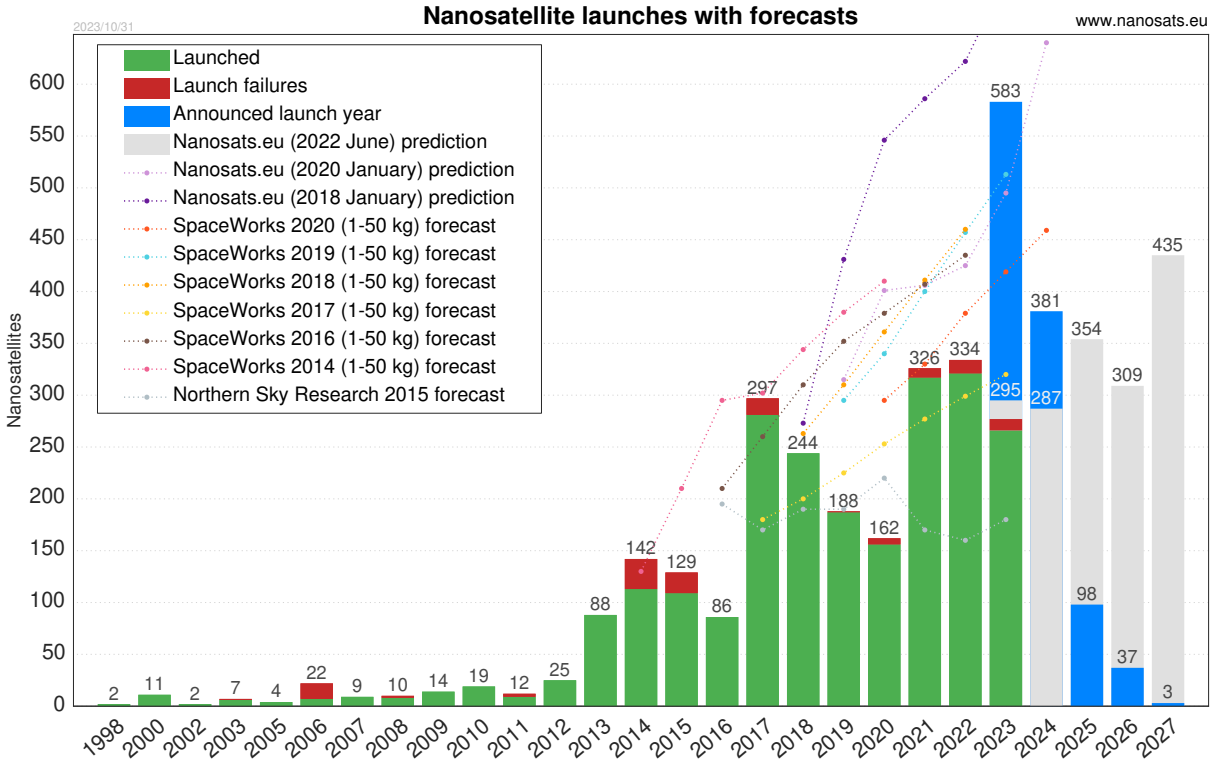


Figure 1.1: Nano satellites launched until 2023, extracted from [4]

The commercial demand for lightweight, low power and smaller star sensors aligns with the rapid development of micro satellite technology, where the reduction in optical system aperture is compensated by high-sensitivity imaging devices. This trend is essential for achieving precise attitude measurements and control in compact satellite designs.

The determination of the attitude of satellites by the star tracker is done by identifying stars in camera Field of View (FoV). Thus, an algorithm is used to correlate the observed stars with an internal star catalog and, with this information, the attitude is calculated. The star detection algorithm is therefore essential for the navigation and operation of the star tracker. Without a accurate algorithm, the star tracker will not be able to provide reliable guidance, compromising the satellite and the mission.

The star tracker architecture is composed of an optical system, image sensor, electronics and a signal processing system. The image sensor, which can be a Charge Couple Device (CCD) or a Complementary metal-oxide-semiconductor (CMOS), is responsible for recording images of objects observed by the optical system. Drivers and controllers, i.e. electronic devices and signal processing systems, are composed of processors and software that are responsible for controlling and operating the hardware. The electronic unit has reasonable processing power, but sufficient to run the star tracking algorithm.

This thesis will be carried out in partnership with the company Synopsis Planet (Start-up incubated at TECLABS, at FCUL, University of Lisbon). Synopsis Planet carries out its activity in instrumentation for space applications.

Synopsis Planet (SP) is currently involved in the HERA mission, developing Light Detection and Ranging (LIDAR), and is also involved in Plano de Recuperação e Resiliência (PRR) New SPACE, whose main theme is the development of a constellation of satellites for Earth observation. SP is responsible for developing the orientation sensor (star tracker) for the constellation of small satellites.

1.2 Objectives

This dissertation aims to develop the electronic part of a star tracker sensor. So the development of the Printed Circuit Board (PCB)s for the star tracker is crucial. This is stage as the hardware has small dimensions and is based on re-configurable hardware. Proper layout is essential to prevent signal interference from other signals or components.

A secondary objectives is to minimize the payload mass of the instrument, i.e, more compact and lighter.

Last step if for the prototype have a validation phase, where the circuits and PCBs are thoroughly tested. In a controlled laboratory environment, secure electronics development kits are used to facilitate preliminary testing of the sensor. The PCB integrating signal processing components undergoes evaluation to verify the efficacy of the implemented algorithm.

1.3 Document Structure

The report is structured into seven chapters with bibliographical references and annexes.

Chapter 2 comprehensively analyses the current related work and relevant literature on star-tracking devices. It explores critical elements associated with these devices and explains their principal components.

Chapter 3 extends the discussion by elucidating elements crucial to the project's implementation, including a detailed exploration of the chosen architecture, the sensor technology used, and the signal processing equipment.

Chapter 4 focuses on developing hardware and PCB prototypes, meticulously crafting circuit design and layout for the PCBs, followed by rigorous testing to ensure signal integrity and system reliability.

Chapter 5 is dedicated to Vivado-based design and software development for the FPGA and the implementation of the I2C protocol, with a specific focus on its application to the AR0134CS sensor.

Chapter 6 presents the project results, encompassing various tests conducted to verify the performance and functionality of the AR0134CS sensor and ensure that it operates as expected.

The final chapter, Chapter 7, synthesises the dissertation to provide a comprehensive overview of the results.

2

Related Work

In this chapter, a survey of several scientific articles is carried out with the aim of exploring, in detail, the evolution of star trackers and the main characteristics, architectures and components used to carry out their sensors.

2.1 History and Landmarks of Star Tracker devices

The use of star trackers can be observed since 1946 where they were designed to control telescopes in large observation centers [5]. At this time, the sensors did not have the capabilities to be placed in a space environment but they contained the base system that led them to evolve to that point.

Between 1950 and 1960, due to military pressure from intercontinental wars, star tracking sensors evolved to be implemented in aircraft and missiles in order to use the directions of the stars to obtain the altitude [5], position and orientation to determine the direction.

In 1959, the first experiment was carried out to obtain orientation and altitude from a star tracker sensor carried by a balloon placed above the Earth's atmosphere by the Johns Hopkins Applied Physics Laboratory [5]. This study continued for the next 8 years in order to improve the guidance system.

It was only in 1985 that star trackers began to be manufactured with the aim of being placed in a space environment [6]. At this time, the ASTROS sensor was

developed, with a mass value of 31 kg and a power consuming value of 43W

Another star tracker, HDOS'HD-1003 created by Goodrich in 1996 was a huge leap in progress because it only had approximately 3.2 kg of mass[7].

Currently, star trackers have smaller sizes, mass and reduced energy consumption. The NST-3 Nano Star commercial star tracker had less than 0.5 kg of mass and have a power consumption lower than 1 W [8].

Another sensor developed at the KTH Royal Institute of Technology uses an Field-Programmable Gate Array (FPGA) to read the acquired data and subsequently sends the location and intensities of each star to the main processor [8].

2.2 Star Trackers

2.2.1 Commercial Star Tracker

In order to find the appropriate specifications for a star tracker, a survey was carried out on existing star trackers. Comtech AeroAstro Miniature Star Tracker (MST) is a joint invention of MIT and AeroAstro, which bridges the gap between low-cost yet rough attitude sensors and expensive commercial star trackers. Miniature Star Tracker (MST) has a 1-megapixel CMOS Active Pixel Sensor and provides a 30-degree FoV with an accuracy of 70 arc seconds. It is suitable for CubeSats, despite mass value of 425g and having dimensions of 5.1 x 7.6 x 7.6 cm, and requires more than 2 W of power consumption. MST was launched with NASA's FASTSAT microsatellite in 2010 [9].

The ST-16 star tracker (Figure 2.1) is comparable to larger star trackers in terms of specifications. It has a 5-megapixel CMOS active-pixel sensor, a 20 x 15 degree Field of View (FOV), an update rate of 2 Hz, and claims an accuracy of 7 arc seconds [10]. Notable features include a full lost-in-space calculation in every frame and a reboot after each frame to counteract radiation-induced memory upsets. It uses a rolling shutter for its image sensor.

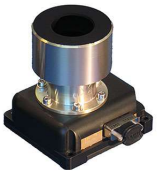


Figure 2.1: ST-16 Star Tracker [10]

The Blue Canyon Technologies (BCT) Nano Star Tracker (Figure 2.2) claims an accuracy of 7 arc seconds and an update rate of 5 Hz. The sensor body includes a baffle, but technical details remain limited. It weighs <0.5 kg, has dimensions of 5 x 5 x 10 cm, and power consumption value of <0.5 W [11].

Figure 2.2: BCT Nano Star Tracker [11]

The ST-200 star tracker from Berlin Space Technologies (Figure 2.3) is based on the fifth generation developed at Technische Universität Berlin. It has a compact size of 3 x 3 x 3.8 cm and 50 g (without a baffle or enclosure). It contains an internal gyro that reports slew rates of up to 200 degrees per second, even when it is unable to lock onto any stars. The ST-200 is priced at around 30,000$ [12].

Figure 2.3: ST-200 Star Tracker [12]

The SHEFEX 2 star tracker integrates a suitable (CCD) camera and a computer for image processing [13]. The ASTRO star tracker family, which includes models like ASTRO5, ASTRO10 (Figure 2.4a), and ASTRO15, uses CCD detectors. ASTRO15, the

most widely used globally, is deployed in geotelecom and Earth observation satellites. The autonomous ASTRO APS star sensor (Figure 2.4b), qualified in 2008/09 [14], uses CMOS active pixel detector technology.

(a) ASTRO10 [14] (b) ASTRO APS [14]

Figure 2.4: ASTRO Star Trackers

In [15], the authors present a digital imaging system that replaces the previous master controller FPGA with Advanced Risc Machine (ARM) + Digital Signal Processor (DSP) (digital signal processor) for enhanced processing. The CubeSat star tracker includes a Melexis image sensor (MLX75412) and an FPGA with an Static Random Access Memory (SRAM) module functioning as a frame buffer [16]. In another instance, Zunzong Yea et al. developed a Star Sensor Imaging Driver Module based on IMX224, utilizing an FPGA with a Cortex-M3 ARM soft core as the imaging driver chip [17].

2.2.2 Star Tracker Characteristics

In this section, we delve into the essential characteristics that define star trackers. The attributes selected were as follows:

1. **Precision**

   The precision of a star tracker system is strongly defined by the algorithm used. To measure its accuracy, the angular distance error from the boresight is calculated, and it's common to express this as either 1σ or 3σ values [29].

   Commercial products provide two accuracy measurements - cross-boresight (pitch and yaw) and boresight (roll). Cross-boresight errors are generally lower than roll errors, and modern trackers can achieve accuracies ranging from 0.1 to 20 arc seconds (cross boresight) [29].

   To achieve such precision, its essential sub-pixel centroid detection is crucial.

2. **Sensor Update Rate and Processing Speed**

   The frequency at which sensor data updates primarily depends on the sensor's exposure time and the speed of the processing algorithms. The optical and hardware components, such as the lenses, sensor sensitivity, and field of view mainly determine exposure time. Longer exposure times can capture more photons, which improves the signal-to-noise ratio (SNR) but reduces the refresh rate. However, there are several methods to enhance processing time based on optical and hardware limitations [26].

3. **Physical Volume and Mass**

   When comparing commercial star trackers, it is important to consider their physical size and mass. High-performance star trackers like the HAST by Ball Aerospace are designed for large satellites and are usually massive instruments [30]. It is safe to assume that they outperform smaller star trackers in most areas. However, the growing market for microsatellites requires smaller and lighter instruments. Hence, this thesis focuses on creating a star tracker with the lowest possible dimensions using the chosen hardware [26].

4. **Power Consumption**

   Like volume and mass, all satellites have a limited power budget that restricts the hardware resources such as processors and memory that can be used. CubeSat-compatible star trackers usually consume less than 1 W on average. Larger star trackers can consume up to 10 W of power [26].

5. **Lens**

   The optics of star tracker devices play a crucial role, encompassing systems of lenses that focus starlight onto a focal plane where the sensing device resides. The lens in this system is designed with a fixed focus, optimized for the operational requirements of star tracking in space. A fixed-focus design eliminates the complexity of moving parts, enhancing robustness and reliability in the challenging space environment.

   The lens must meet dual requisites. Firstly, it must ensure a sufficiently large field of view to capture the requisite number of stars for accurate tracking measurements. Secondly, the lens must endure the rigours of launch and space environments while retaining its imaging capability.

   The field of view, a critical aspect, correlates with sensitivity, wherein sensitivity is gauged against the influx of starlight photons. The number of photons

captured (N_php) is determined by a formula accounting for factors such as lens aperture (A_l), permeance rate of optics(T_l), bandwidth(ΔB), luminous flux(ϕ_m), and exposure time (T) [31].

$$N_{php} = A_l \times T_l \times \Delta B \times \phi_m \times T \tag{2.1}$$

Here 2.1 is derived from Li et al. [2007], calculates the number of photons per second hitting the image plane, thereby influencing the efficacy of star tracking measurements [31].

To ensure the functionality and reliability of the camera system in the harsh conditions of space, the lens system undergoes rigorous testing. Mechanical strength is a particularly vital characteristic, as the lens must endure high acceleration and vibration forces during launch. Precision optics, integral to the lens's construction, are scrutinized for resilience under these stresses. The lens system go through a vacuum chamber assessment to quantify outgassing, the release of chemical vapours induced by vacuum conditions. This process is essential to prevent residues from settling on the lens, which could impair star tracker performance [31].

Collectively, these stringent tests serve as imperatives, including vibration and vacuum assessments, the lens system is fortified against mechanical and environmental challenges, affirming its operational integrity in the demanding circumstances of launch and space [31].

6. **Radiation tolerance and thermic resistance of electronic components**

The effects of radiation and thermal resistance are crucial for the functioning of star tracker devices. These devices are usually placed in small satellites that orbit at LEO altitude, typically between 400-1000 km [10]. The high-energy protons that are prevalent in this orbit are a primary concern. Proton flux is influenced by the fluctuating solar cycle, with increased activity during solar minimum. When protons interact with semiconductor devices, they cause displacement damage in junctions, which alters the electrical properties of the circuit. This effect is particularly pronounced in silicon-based detectors, leading to elevated dark current and changes in operational characteristics [10]. Shielding is effective against electrons and low-energy protons [10], but less so against high-energy particles such as cosmic rays. In addition to shielding, other methods can be employed to increase the resilience of star tracker electronics. For instance, radiation-hardened CMOS sensors can be used, as they are designed to tolerate high levels of ionizing radiation. Techniques such as error correction coding (ECC), redundancy

in electronic circuits, and adaptive filtering can mitigate the effects of radiation-induced errors. Additionally, specific design considerations, such as the use of guard rings, enclosed layout transistors, and circuit-level hardening techniques, are often applied to enhance the radiation tolerance of MOS transistors. This can exacerbate Single Event Effect (SEE)s due to secondary particle generation [10].

Radiation damage not only contributes to increased dark levels and dark noise but also amplifies the occurrence of hot pixels, which remain saturated even under zero-illumination conditions [10]. The impact of radiation exposure shows a direct correlation between elevated temperature, dosage, and the proliferation of hot pixels. It is crucial to have a comprehensive understanding of radiation-induced effects on electronic components to optimize the performance and time span of star tracker devices in demanding space environments.

Some studies show that by minimizing unit mass in star tracker devices, a trade-off emerges with increased susceptibility to radiation issues. This is evidenced by the reduced design life of the ST-200 in comparison to its predecessor, the ST-100 [10]. This compromise is further manifested in the downsizing of the baffle, which is a critical component for optical systems.

The thermal resilience of the camera systems is also essential, given the wide temperature differences experienced by satellites in space as they transition between sunlight and shadow [10].

7. **Algorithms**

The algorithms used in star tracker devices almost always follow the same philosophy and are divided into three steps as shown in Figure 2.5.

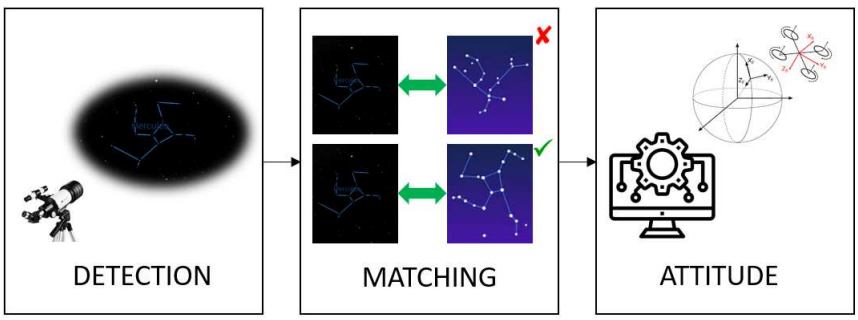


Figure 2.5: Star Trackers Algorithm Diagram

Once an image is captured by the sensor, it undergoes a filtering process to correct any distortion caused by the lens. The image is then converted into Cartesian coordinates, which are used as a vector by algorithms.

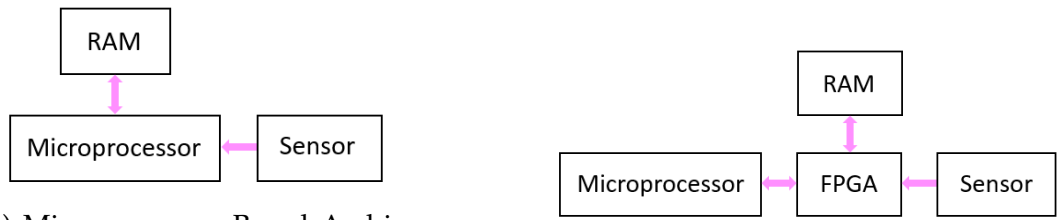
Next, the star constellation catalogue is used to identify which star corresponds to the acquired image. The most commonly used algorithm for this is the Geometric Voting System, which involves a voting scheme based on pairs of stars. The distance between each pair of stars in the image is measured, and the catalogue is searched for pairs with similar inter-star distances. Each imaged star in the pair receives a vote from each star in the matching catalogue pair, which provides possible identities for the imaged stars.

After considering all the star pairs in the image, each imaged star has multiple votes from different catalogue stars. The correct identity for the imaged star is usually the one with the most votes. This voting stage is followed by another verification stage before the matching process is considered complete.

Finally, the altitude of the space object can be determined by validating the detected stars and converting them to Earth-Centered Inertial (ECI) coordinates. The most common algorithm used to calculate altitude is the Quaternion Estimator Algorithm (QUEST). By incorporating measurements from various stars, QUEST creates an optimisation problem to minimise the discrepancy between the observed and modelled star vector measurements. This optimisation process yields the quaternion parameters that best align with the observed star positions. The use of quaternion parameterization enables a concise representation of three-dimensional orientation.

2.3 Typical Architectures

Many models developed for research projects and commercial star trackers use a conventional camera model approach to capture and process images. The process involves capturing an image with a standard image sensor, storing it in a memory buffer and then extracting the necessary data by accessing it through this buffer [18]. Figure 2.6 illustrates two of the most used schemes.



(a) Microprocessor Based Architecture

(b) FPGA Based Architecture

Figure 2.6: Star Tracker Typical Architectures

As shown in Figure 2.6b, some models use FPGAs to implement a receiver for the image sensor's Camera Serial Interface-2 (CSI-2) interface and a memory buffer writing and reading controller. These models generally use SRAM or Synchronous Dynamic Random-Access Memory (SDRAM) chips with a few megabytes. In this approach, the processor or microcontroller that performs software processing accesses the image through the FPGA as a bridge [18].

Another scheme uses more robust microcontrollers that have a CSI-2 protocol receiver already implemented in hardware. In this approach, the image sensor is directly connected to the microcontroller, and the images are redirected to an external volatile memory (usually through Direct Memory Access (DMA)). Due to the greater complexity of the microcontroller, this approach generally consumes more energy and resources, which may require sharing with other satellite modules or tasks[13]. This type of model is illustrated in Figure 2.6a.

There is also another variation that uses both an FPGA and a microcontroller, each with an exclusive memory module, without sharing memory space between them [18].

2.3.1 Microprocessor Based Architecture

Figure 2.6a showed that a processor is a key component responsible for reducing noise and implementing various algorithms, including image processing and tracking algorithms. A thorough examination of microprocessors has revealed several noteworthy candidates, each contributing to the development of star-tracking technology.

The STM32F401RET6 microcontroller (Figure 2.7) is a top performer in the ESTCube-2, supporting a 32-bit ARM Cortex Central Processing Unit (CPU) [19]. It expertly manages power sequencing, sensor and FPGA configurations, temperature sensing, and communication with the On-Board Computer System (OBCS). With 512 kbytes of flash memory, 96 kbytes of SRAM, and 16 Analogic-Digital Converter (ADC) channels, each with 12-bit resolution, this microcontroller operates seamlessly within a voltage range of 1.7 to 3.6 volts. It offers a variety of connectivity options ranging from Universal Asynchronous Receiver-Transmitter (UART) to OBCS, Serial Peripheral Interface (SPI) to the FPGA, SPI to Ferroelectric Random Access Memory (FRAM), and Inter-Integrated Circuit (I2C) to the image sensor. Temperature sensors also directly interface with the analog inputs for added convenience.

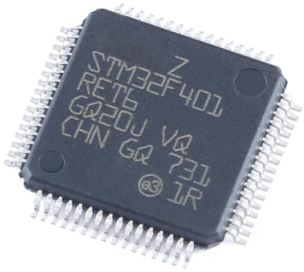


Figure 2.7: STM32F0401

Another remarkable processor is the Texas Instrument OMAP3530 (Figure 2.8), which is highly recommended by industry experts [20]. It is renowned for its powerful graphics interfaces, which contribute to the reduction of star tracker size and power consumption. With UART, SPI, and I2C ports, OMAP3530 offers a rich set of functionalities, from real-time clock and Secure Digital to graphics acceleration and Image Processing Unit. It operates seamlessly on high-level systems such as Linux, Windows CE, Symbian Operationg System (OS), and Palm OS, and features a power management system that efficiently curtails power usage. In [21], it has proven its worth in the star tracker of st-16 and the command and data handling system in Trailblazer-2 and CubeSat satellites, and it has endured radiation testing, demonstrating resilience up to 35 krad (Si).

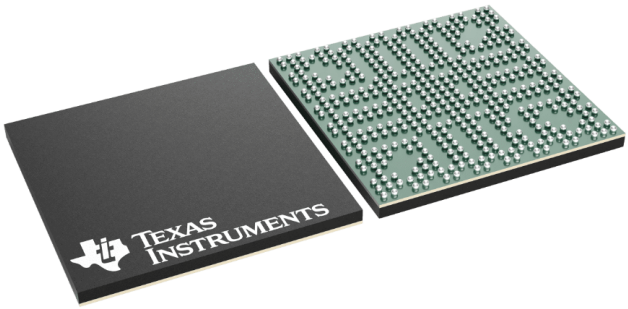


Figure 2.8: OMAP3530

Additional insights from the OreSAT author [22] shed light on microprocessors that are driving star-tracking capabilities forward. The STM32F091CC (Figure 2.9a), a nimble 32-bit 48 MHz Cortex-M0 microcontroller, excels in systems like solar modules and battery cards.

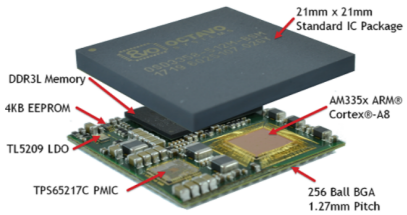
Meanwhile, the STM32F439 (Figure 2.9b) is a powerful 32-bit 168 MHz Cortex-M4F microcontroller that can handle the demands of more complex systems with ease,

thanks to its multiple memory banks, Embedded Multimedia Card (eMMC) flash interfaces, and multiple Controller Area Network (CAN) peripherals.

Lastly, the Octavo Systems OSD3358 (Figure 2.9c) is a System-in-Package (SiP) device that integrates multiple components into a single package. It includes the Texas Instruments Sitara™ AM335x ARM® Cortex®-A8 processor, DDR3 memory, a power management IC (PMIC), and necessary passive components. This SiP takes center stage in boards that require data processing capabilities as in star trackers, science cameras, or Software Defined Radio (SDR) Global Positioning System (GPS).

(a) STM32F091CCe

(b) STM32F439



(c) OSD3358

Figure 2.9: OreSat processor choices

Together, these microcontrollers exemplify the diverse technological landscape that is driving innovation in star-tracking capabilities. Each processor brings unique strengths to the forefront of this exciting field.

2.3.2 FPGA Based Architecture

In the world of star tracker sensor projects, there have been exciting advancements in utilizing FPGAs for image processing. Researchers [8] [23] have showcased the potential of customized hardware solutions through innovative approaches, such as implementing the entire centroid extraction process within an FPGA. One particularly impressive project in this category bypasses the need to store complete frames by processing the pixel stream from the image sensor, resulting in a significant boost in efficiency [10]. While the other two projects also use similar techniques, they differ in

their approach to storing images, either completely or partially, in memory buffers.

When it comes to identifying centroids efficiently, a particular project [24] catches the eye with its use of Application-Specific Integrated Circuit (ASIC) technology. However, the high manufacturing cost of ASICs is a significant hurdle, particularly in CubeSat contexts where cost-effectiveness is crucial.

For those searching for alternatives, some projects turn to Digital Signal Processors (DSPs) to handle centroid processing in software with minimal time overhead [7]. While this approach does not directly utilize FPGA hardware, it highlights various strategies aimed at achieving optimal star tracker sensor performance.

The work in [25] makes excellent example of how to use this technology in star tracking applications. Using the ZedBoard (Figure 2.10) development kit by AVNET, which utilizes a Xilinx Zynq-7000 System-on-Chip (System-on-Chip (SoC)) to seamlessly connect a dual-core ARM Cortex-A9 processor with an FPGA boasting 85,000 logic units. This versatile development kit has been utilized in a range of applications, from experimental to commercial modules, including satellites. Its utility is further enhanced by the integration of Video Graphics Array (VGA) and High-Definition Multimedia Interface (HDMI) outputs, along with Pmod connectors for connecting image sensors. The software execution is managed through the Linux operating system, running on the ARM processor, utilizing the 2017.4 version of the Linux kernel adapted by Xilinx, and the Fedora Server 27 distribution for ARM processors.

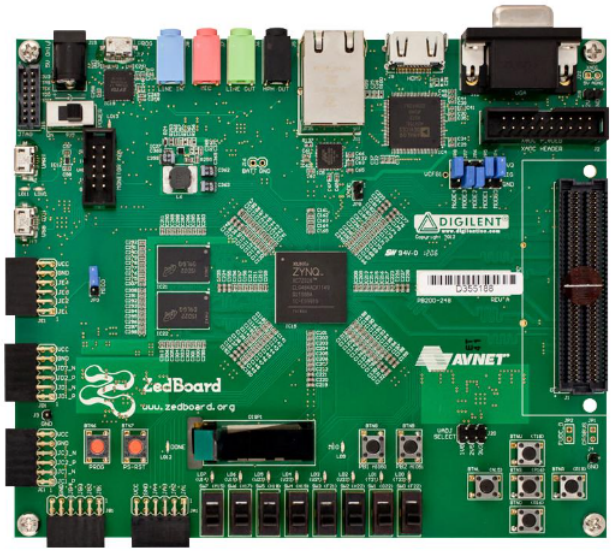


Figure 2.10: Zedboard

In a specific instance cited in the author's thesis [26], the MicroZed development board that hosts a Zynq 7020 SoC was utilized (Figure 2.11). This SoC features

two ARM Cortex-A9 CPUs, extensive I/O options, memory, clock generating circuits, memory controllers, and Advanced eXtencible Interface (AXI) interface ports facilitating data transfer between the configurable FPGA circuit (Programmable Logic) and the processor (Processing System). The close integration and cooperation between the two components is emphasized by the clock that drives the Programmable Logic (PL) logic circuit being generated by the Processing System (PS).

Figure 2.11: MicroZed development board

These various projects demonstrate the efficiency gains and adaptability possible through FPGA utilization in star tracker sensor systems. They offer insights into the evolving landscape of spaceborne image processing technologies.

2.4 Image Sensors

The Optoelectronic imaging module is a critical component of the star sensor hardware system, affecting key performance parameters. The quality of imaging and star detection plays a significant role in determining the overall effectiveness of the sensor. Common image devices for star sensors include CCD and CMOS, with the latter being the preferred choice due to its advantages. CMOS sensors have low weight, high reliability, extensive integration, cost-effectiveness, a wide dynamic range, and radio resistance, while CCD sensors have poor space radiation resistance, substantial volume, high power consumption, and considerable weight.

The Aptina MT9P031 image sensor(Figure 2.12) has become a leading choice among star trackers, with widespread usage across multiple countries [20]. Its outstanding performance is showcased in the ST-16 star tracker, which boasts fully automatic tracking capabilities, a weight of only 90 g (excluding baffle), consumes less than

1 W of power, and an arc accuracy of 7 arcseconds. Since 2013, 17 models of this star tracker have been successfully launched into orbit, making it a reliable option. The sensor's capabilities are also utilized in the ST-200 star tracker, which is equipped with an ARM9 processor. In 2010, NASA and Canada conducted rigorous radiation testing at the TRIUMF facility, proving the MT9P031 sensor's effectiveness for Low Earth Orbit (LEO) missions.

Figure 2.12: Aptina MT9P031 image sensor

Onsemi, a prominent figure in the field of star tracker devices, is proud to present the MT9P031 CMOS sensor (Figure 2.13[25]). This monochrome sensor boasts impressive specifications such as a 2592x1944 pixel resolution at 14 frames per second or 640x480 pixel resolution at up to 53 frames per second, and a maximum data rate of 96 Mp/s at a clock rate of 96 MHz. Its reliability is further cemented by its use in various star trackers, making it a trusted choice with a proven track record in space applications.

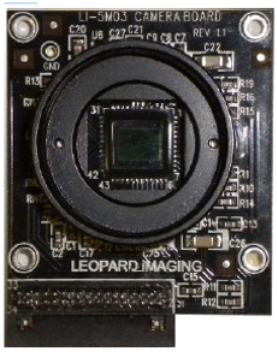


Figure 2.13: MT9P031

Another exceptional mode is MT9D111 from Micron (Figure 2.14[25]).This model has demonstrated its versatility in nanosatellite camera projects, making it a valuable asset across a range of missions.

Figure 2.14: MT9D111

Finally, by utilizing the On Semi AR0134 monochrome CMOS camera (Figure 2.15), the OreSat Star Tracker is able to capture images of star fields [22]. This cutting-edge technology, inspired by the openstartracker.org project, calculates right ascension, declination, and roll parameters, highlighting the ongoing progress in star tracking capabilities.

Figure 2.15: OreSat

3

Proposed Work

This chapter provides a more detailed exposition of the foundation for the proposed work, with a primary focus on the development of hardware for the star tracker device. It identifies the topology to be employed, outlines the mechanical constraints imposed, and emphasizes certain requisite components.

Consequently, the chapter is structured into two parts: The first part describes the selected topology and the proposed dimensions for the intended Printed Circuit Boards (PCBs). Subsequently, in the second part, two components are highlighted as foundational elements for the fabrication of the star tracker PCBs.

3.1 Topology and Size Limitation

Each star tracker needs an imaging sensor which is a CMOS sensor that takes images of the sky. Its output is read by an FPGA that processes the image data stream and extracts the star information from the image. The extracted star position and size information is then saved to an SDRAM (Synchronous Dynamic Random Access Memory).

The FPGA is utilized for its ability to perform parallel database lookups at high speeds, enabling efficient star matching. The star database, used for identifying matches with detected stars, is stored in FLASH memory. After the FPGA has finished processing, it sends the computation result (the determined attitude of the spacecraft) to the microcontroller that in turn can forward the data to the OBCS (On Board Computer System).

It was also considered using an HDMI interface, so it is possible to transmit the images that the sensor is collecting, making it possible to view the sensor data.

To manage the power required to turn on the FPGA board and the sensor properly it will be used a Power Management Unit (PMU) because these components have a power-up and power-down sequence and this process must be well managed.

This topology is depicted in the following Figure 3.1:

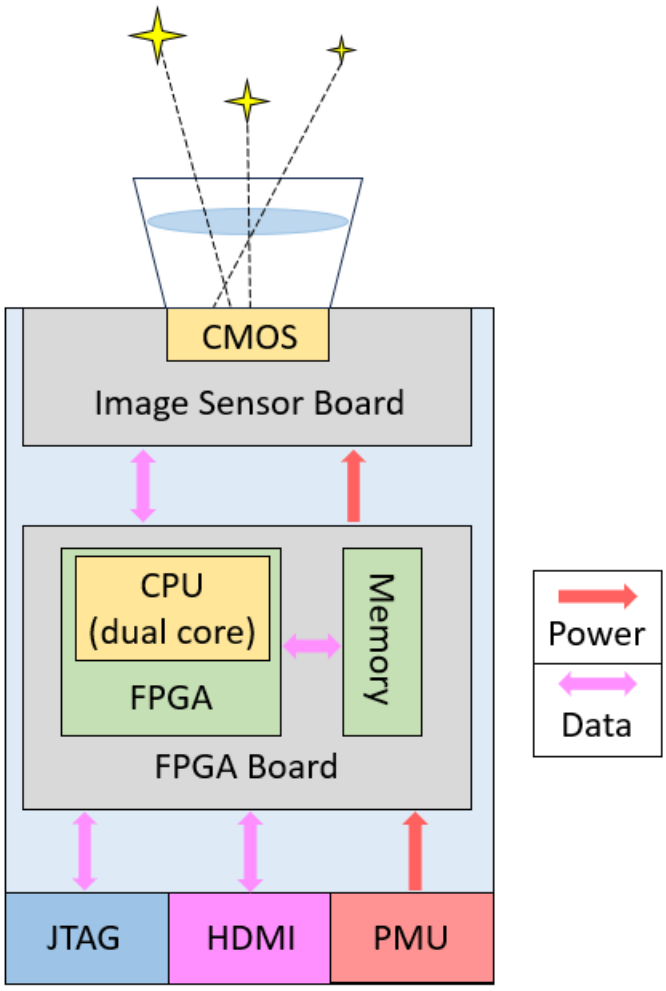


Figure 3.1: Block diagram of the Star Tracker device

Even though the diagram makes it seem that the sensor is big, it was required that it fit in a 51mm by 57mm area with the shape of the Figure 3.2 with space for screws so the module could withstand the vibrations of the launch to space.

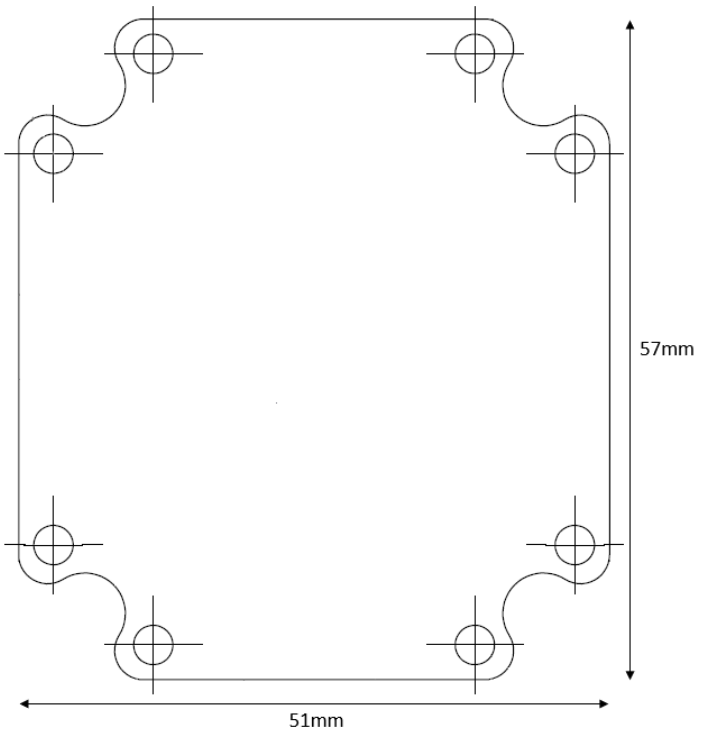


Figure 3.2: Mechanical size limitation

This happens because Synopsys Planet is already working on a similar project without the FPGA where those are the maximum dimensions. In their project, they use only a Microprocessor that manages the image sensor and does the algorithm to determine the attitude.

3.2 Required Components

Two main components were selected before designing schematics and PCBs: the image sensor AR0134CS and the FPGA Zynq-7035. The following two sections will describe these components.

3.2.1 Image Sensor (AR0134CS)

The lens team at Synopsis Planet determined that the optical format of the lens would be 1/3 inches (6 mm) making it a restriction to when choosing a sensor.

The CMOS sensor which will be used in the star tracker is AR0134CS manufactured by ON Semiconductor. It is an RGB sensor that has a resolution of 1280(H)×960(V) pixels (Figure 3.3) 54 frames per second. It has a maximum data rate of 74.25 Mp/s [27].

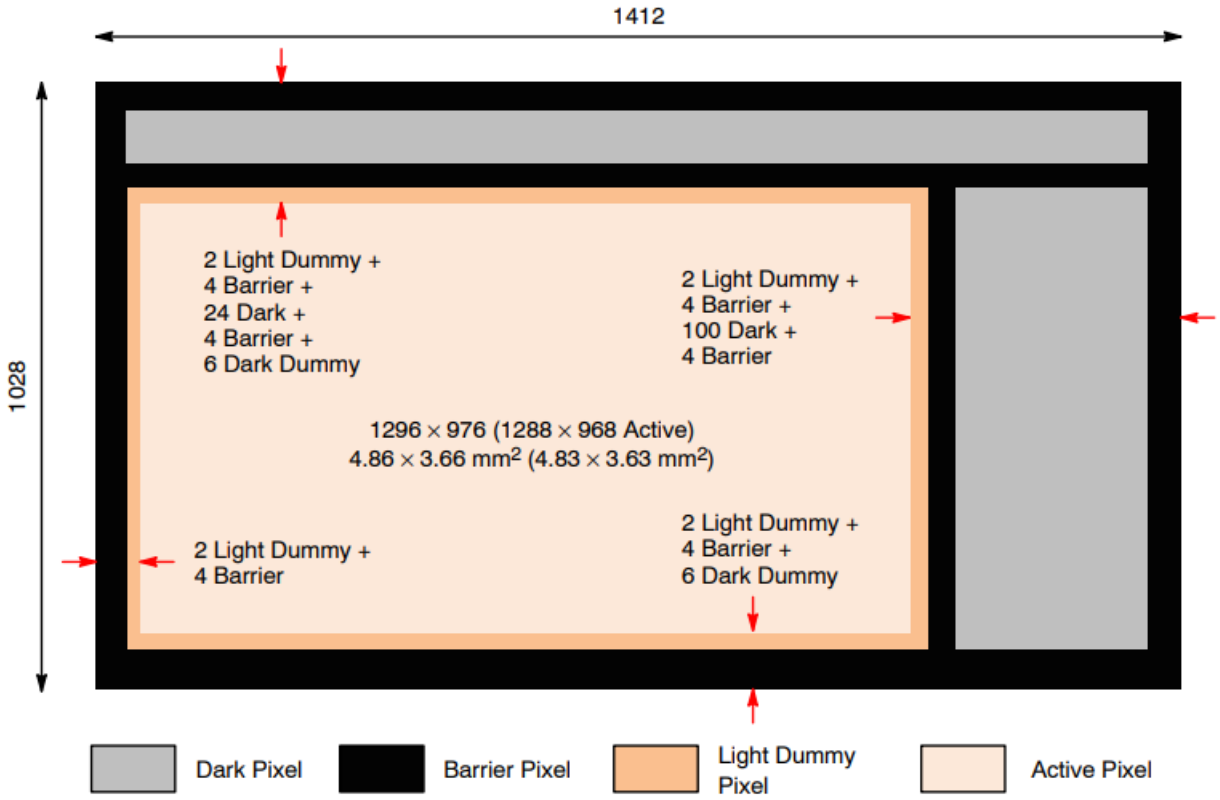


Figure 3.3: Pixel Array Description extracted from [27]

The CMOS sensor used for the star tracker also features a true global shutter. This means that since all pixels are exposed together, there is no distortion or skewing of fast-moving objects in the image, known as the rolling shutter effect.

The clock rate recommended for the sensor is 23 MHz, but it can range from 6 MHz to 50 MHz [27]. At the recommended clock speed, the sensor can read out a 1280x960 image in approximately 53.4 milliseconds. However, higher performance settings can achieve readout times as low as 18.6 milliseconds.

Exposure time is determined based on the number of row times (horizontal line periods), where each row time depends on the line length (in pixel clock cycles) and the pixel clock frequency. The exposure time can be calculated as:

$$\text{Exposure\_Time} = \text{Integration\_Time} \times \text{Row\_Time} \tag{3.1}$$

The integration time is not a direct time value but rather a multiplier that specifies how many row times are used to integrate light. For example, with a row time of 22.22 μs, an exposure time of 2.22 ms would require:

$$\frac{\text{Exposure\_Time}}{\text{Row\_Time}} = \frac{2.22\,ms}{22.22\,\mu s} = 100 \tag{3.2}$$

This means a value of 100 should be entered into register R0x3012 to achieve the desired exposure time. In this mode, only whole number row-time increments are allowed, so fractional exposure times (e.g., 2.5 ms) cannot be achieved. Adjusting the number of horizontal active or blanking pixels can modify the line length, effectively changing the row time and allowing finer control over exposure.

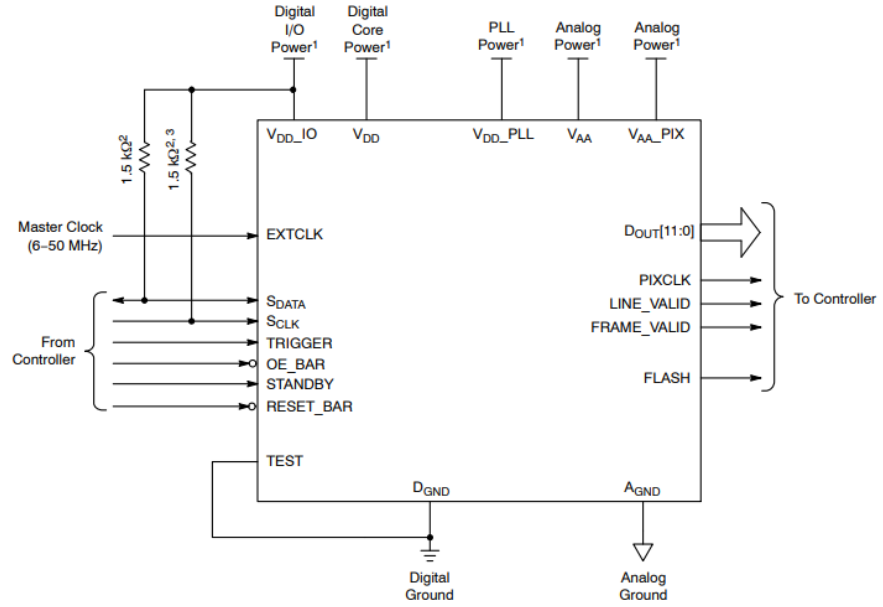
In high-performance modes, reducing the row time (e.g., by increasing the pixel clock or minimizing blanking intervals) enables shorter exposure times and faster frame rates. For instance, if the row time is reduced to 10 μs, an exposure time of 2.22 ms would require 222 rows instead of 100, allowing greater flexibility in achieving specific timing requirements.

By carefully configuring the row time and number of rows, the sensor can be optimized for both default and high-performance modes, ensuring precise exposure control for different applications.

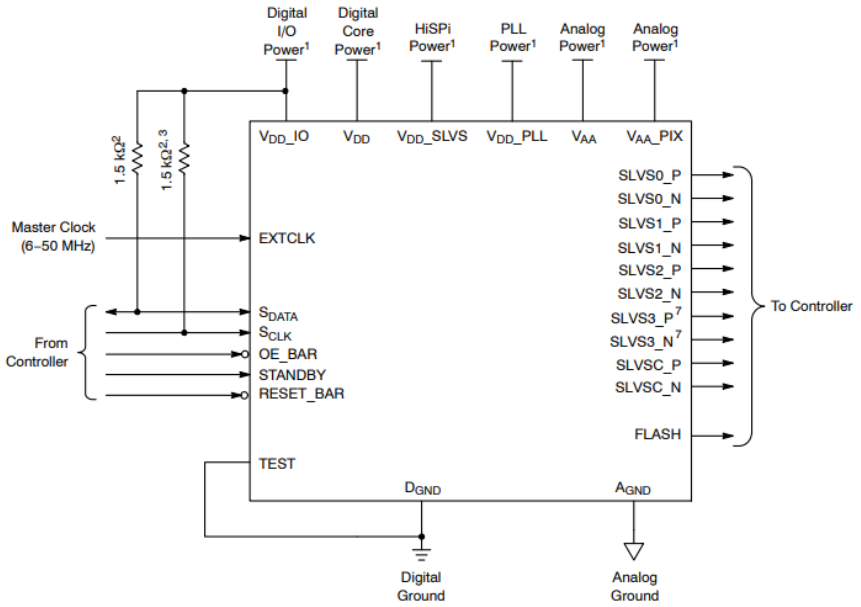
The AR0134CS includes additional features to allow application-specific tuning: windowing, adjustable auto-exposure control, auto black level correction, onboard temperature sensor, and row skip and digital binning modes. The sensor is designed to operate in a wide temperature range (–30°C to +70°C) [27].

It's also important to note that this sensor has a power-up and power-down sequence with different voltages required to work properly. This means that it will need a sequencer and voltage regulators in the if the voltages are generated from different power supplies.

The image sensor is the block responsible for taking pictures of stars in the sky and clocking the image data out to the FPGA over 12-bit raw data, using either the parallel (Figure 3.4a) or serial (HiSPi, Figure 3.4b) output ports allowing it to operate in video (master) mode or frame trigger mode [27]. It is also connected to the FPGA over an I2C bus for configuration.



(a) Parallel 12-bit data Interface



(b) Serial 4-lane HiSPi Interface

Figure 3.4: Schematic with connection for (a) parallel data interface, and (b) serial interface, extracted from [27]

The AR0134CS was chosen mainly because the team has used this image sensor for their version of this star tracker.

Since this sensor has been used in other star trackers as well, it was a safer choice than a camera with no space heritage as it is proven to work in space applications [22].

3.2.2 Zynq-7030/Zynq-7035 SoC

The star tracker device will be based on a Zynq-7030/Zynq-7035 FPGA manufactured by Xilinx. Its main role is to receive the image from the CMOS sensor, process the image data to identify the stars and match the detected stars against a database of known stars.

The FPGA was chosen based on specific criteria, including its Dual-core ARM Cortex-A9 processor, which allows it to manage the algorithm and send control signals for memory access or communication with the sensor via I2C [28]. Other factors that were considered include the ability to upgrade or downgrade with different FPGAs of the same family, the capacity to work at fast frequency rates, having a Double Data Rate version 3 (DDR3) interface for external memories, and 256 kbytes of flash memory to program the FPGA [28].

Additionally, Xilinx provides a wide range of Intellectual Property (IP) Cores that can be used in the design process. These pre-designed and pre-verified pieces of logic or functionality can be integrated into an FPGA design, such as Communication Interfaces (e.g., UART, SPI, and I2C). For example, we can use an HDMI IP Core to transmit the sensor data to a screen.

By using IP cores, FPGA designers can focus on the specific functionality unique to their application, rather than starting from scratch for every project. This approach accelerates development cycles and allows for the creation of more sophisticated and complex FPGA designs.

4

Hardware Design

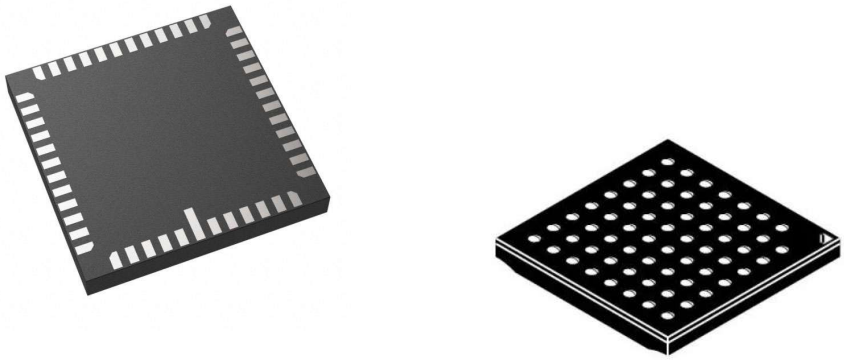
In this chapter, the development related to the CMOS Sensor board and FPGA is described, including the components needed, the layers, and tools used to verify the proper functionality of the electronics.

The first section describes the image sensor PCB with the schematic, layout, and power consumption. The second section is FPGA-related, showing the peripherals needed, the components for its power, and how it connects to the image sensor PCB. Finally, the results and some features for debugging the PCBs are presented.

4.1 Image Sensor Board

Drawing from our understanding of the AR0134CS sensor's performance in space, we recognize its strong resistance to space radiation. This sensor is widely utilized in various star trackers, including those mentioned in [39]. Given these attributes, it was selected for the development of the prototype.

For this project, two variations of the CMOS Sensor AR0134 manufactured by ON Semiconductor were used: one with an Lead-less chip carrier (LCC) package, Figure 4.1a, that only has a 12-bit parallel bus interface to send the image over to the FPGA and one with a Ball Grid Array (BGA) package Figure 4.1b that both includes a 12-bit parallel bus interface and HighSpeed Serial Pixel Interface (HiSPi) interface with three serial lanes.



(a) AR0134 LCC package (b) AR0134 BGA package

Figure 4.1: AR0134 different Pin out packages

One advantage of CMOS sensors and modules is that they reduce the number of components you need to include on your board. These sensors generally require at least two voltage sources to power the digital and analogue circuits in the chip, in our case, 2.8 V and 1.8 V for both packages and 0.4 V for the BGA package.

These sensors are typically surface-mounted. Some CMOS sensors are available as bare dies mounted in BGA packages. However, compared to FPGA BGA packages, CMOS sensor BGAs typically have fewer pins and a lower pad density (fewer connections per square millimeter). This reflects the simpler connectivity requirements of CMOS sensors compared to FPGAs. Other image sensors are integrated with their signal processing block as a single SoC. A stack with at least four layers is normally recommended for these sensors or SoCs.

For the BGA sensor, which has differential signalling standards like LVDS, the primary routing rules with impedance matching for the signalling need to be followed when routing the converted signal away from the sensor ([37]).

This allows for a continuous ground layer to be spread underneath the CMOS sensor's surface layer. However, the analogue and digital sections should use separate ground plane portions, carefully designed to prevent any unintentional gaps or splits in the ground plane. Splits or interruptions can lead to noise issues and reduce the shielding effectiveness. Additionally, high-speed or high-frequency circuits should be placed at a safe distance from the imaging sensor to minimize interference.

To verify that the differential signals were rightly adapted for the signal integrity, it was used the CST Studio Suite.

4.1.1 Image Sensor Board Schematic

The schematic for the sensor board was inspired by the AR0134CSSC00SUEAH3-GEVB Evaluation Board developed by OnSemi. This board implicitly requires pull-up resistors in the I2C bus and a Reset pin for proper functionality.

The use of 33Ω resistors on the 12 parallel DOUT interface it is mandatory. The resistors provide some protection by limiting the current that flows between the sensor's DOUT pins and the receiving device in case of unexpected voltage differences, like Electrostatic Discharge (ESD) events or slitly diferente supply voltages.

4.1.2 Image Sensor AR0134CS circuit

Two designs were done to consider the two models/packages of the image sensor as shown in Figure 4.2 and Figure 4.3.

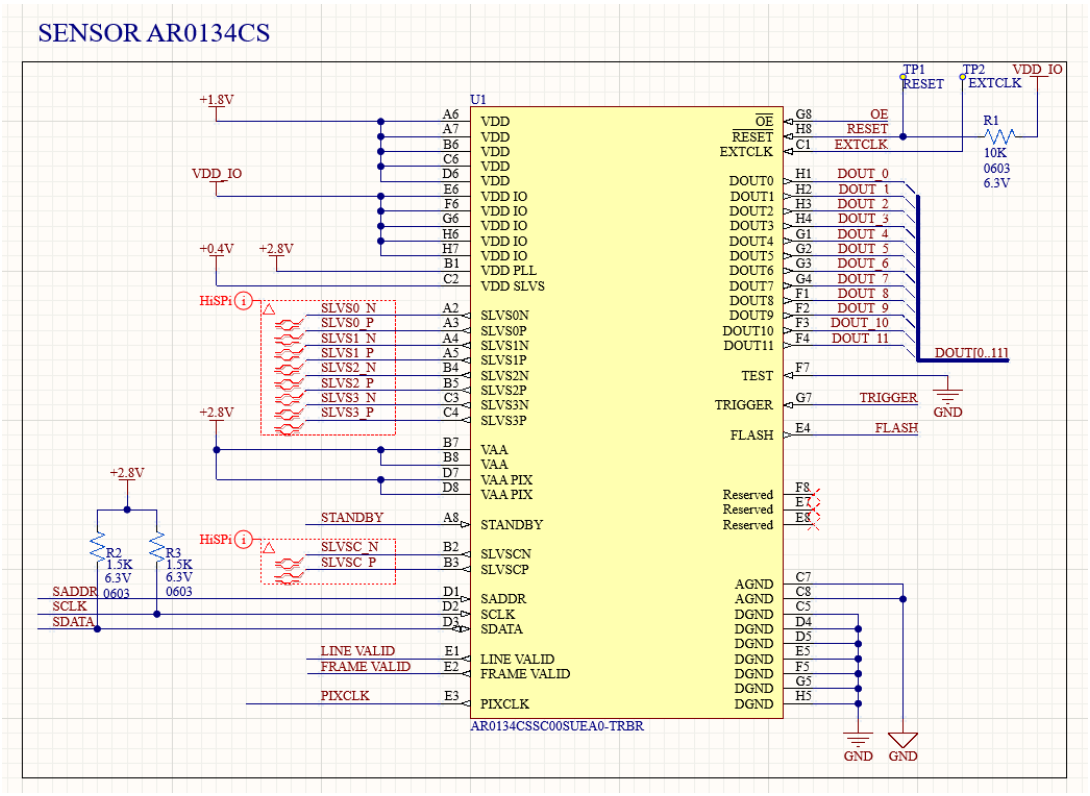


Figure 4.2: Sensor Schematic for BGA package variant with SLVS and parallel interfaces

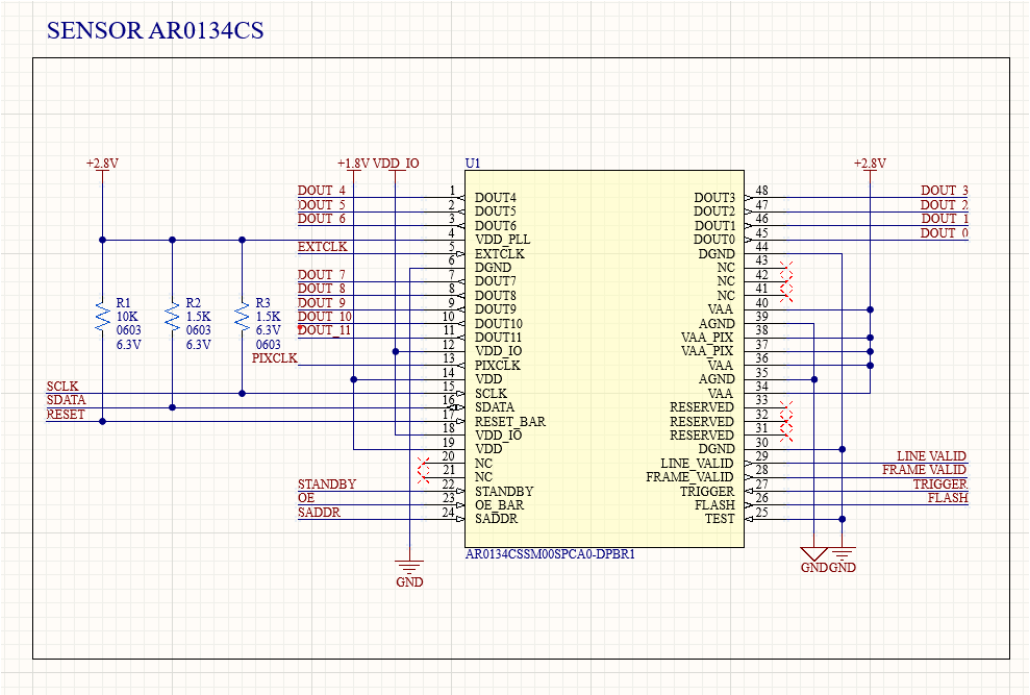


Figure 4.3: Sensor Schematic for LCC package variant only with parallel interface.

For maximum flexibility the Trigger and OE signals were made available to the FGPA to fully control both data interfaces. For example, if we use the HiSPi interface, the OE pin must be connected to the ground for proper operation. In the other case, the OE pin determines whether the data from the parallel interface is available.

The two-wire serial interface bus enables read/write access to control and status registers within the AR0134CS. The interface protocol uses a master/slave model in which a master controls one or more devices. The sensor acts as a slave device. The master generates a clock (SCLK) that is an input to the sensor and synchronises transfers. Data is transferred between the master and the slave device on a bidirectional signal (SDATA). SDATA is pulled up to VDDIO off-chip by a 1.5 kΩ resistor. Either the slave device or master device can drive SDATA LOW; the interface protocol determines which device is allowed to drive SDATA at any given time.

The protocols described in the two-wire serial interface specification allow the slave device to drive SCLK LOW; the AR0134CS uses SCLK as an input only and, therefore, never drives it LOW.

For the HiSPi interface, an LVDS driver is also needed so we can receive the data on the FPGA, as will be explained in the following section.

4.1.3 LVDS Driver selection

The image sensor with BGA package, there is a serial interface with differential signals called HiSPi.

The HiSPi, an interface and protocol devised for the rapid transmission of data, finds its primary application within the domain of image sensors, notably the ON Semiconductor AR0134. It leverages a Scalable Low Voltage Serial (SLVS) physical layer engineered to minimise power consumption while maximising data transmission speeds. This operational paradigm facilitates the efficient conveyance of pixel data by the AR0134 image sensor, rendering it particularly for camera technologies that demand elevated frame rates alongside substantial data throughput.

The HiSPi physical layer is organized into blocks of four data lanes with an associated clock lane, and its PHY (physical layer) uses a low-voltage serial differential output. The HiSPi PHY employs a simple current steering driver with two complementary outputs (V_{0a} and V_{0b}). These drivers (represented on Figure 4.4) are designed to connect to a short 100Ω differential interconnect, terminated by a 100Ω resistor at the receiver, with C_L representing the combined parasitic capacitance of the receiver and interconnect.

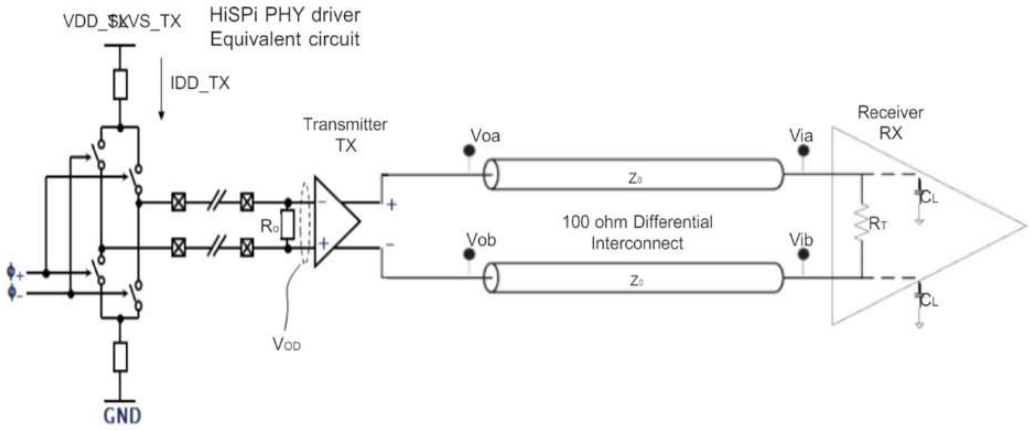


Figure 4.4: SLVS Interface, extracted from [40]

Distinct from conventional Low Voltage Differential Signaling (LVDS), SLVS operates at reduced voltage levels, achieving lower power consumption. Whereas standard LVDS signalling typically involves voltage swings in the range of 350mV with a common-mode voltage range of 1.2V to 1.4V, SLVS, in contrast, is optimised for applications where power efficiency is paramount. It functions within a voltage swing range of approximately 200mV and maintains a common-mode voltage close to 0.2V.

This reduction in operational voltage plays a crucial role in decreasing power utilisation, an aspect of critical significance in mobile imaging sensors and similar energy-sensitive applications.

Figure 4.5 and Table 4.1 present a summary of the last chapter when comparing the voltage levels for LVDS and Sub-LVDS.

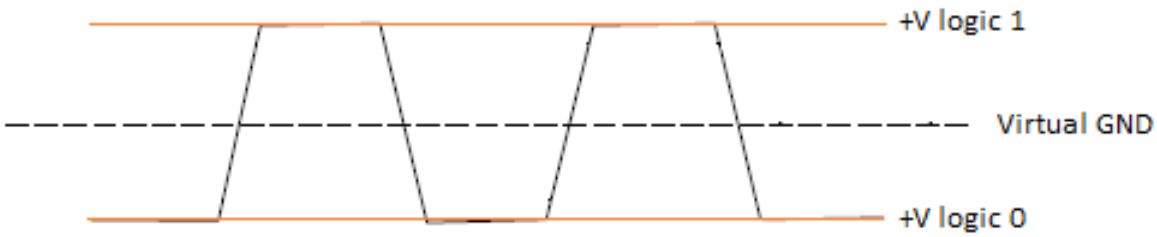


Figure 4.5: LVDS-type signal

Table 4.1: LVDS and SLVS (HiSPi) Voltage Levels

Type	Designator	Typical Voltage Value
LVDS	Virtual GND	1.25V
	+V (logic 1)	1.375V (+175mV)
	+V (logic 0)	1.05V (-175mV)
SLVDS (HiSPi)	Virtual GND	200mV
	+V (logic 1)	0.3V (+100mV)
	+V (logic 0)	0.1V(-100mV)

Nonetheless, the inherent disparities between SLVS and traditional LVDS signalling protocols may precipitate compatibility issues with FPGAs designed for LVDS signalling. Given that FPGAs are generally calibrated to accommodate higher voltage swings, they might not accurately interpret the diminished differential signals characteristic of a SLVS interface. Such discrepancies in voltage levels and signal attributes could compromise signal detection and data integrity.

To address these compatibility challenges, the use of LVDS drivers, such as the FIN1101 or DS90LV804 (Figure 4.6), is essential. These drivers are specifically designed to handle low-voltage differential signals, amplifying or converting them to meet the input requirements of an LVDS receiver. Notably, these drivers must have a common mode voltage lower than 200 mV to correctly discriminate SLVS signals. Furthermore,

the inclusion of these drivers does not result in a significant increase in power consumption, as both chips are designed to consume minimal current. This aspect is further detailed in Section 4.1.4.

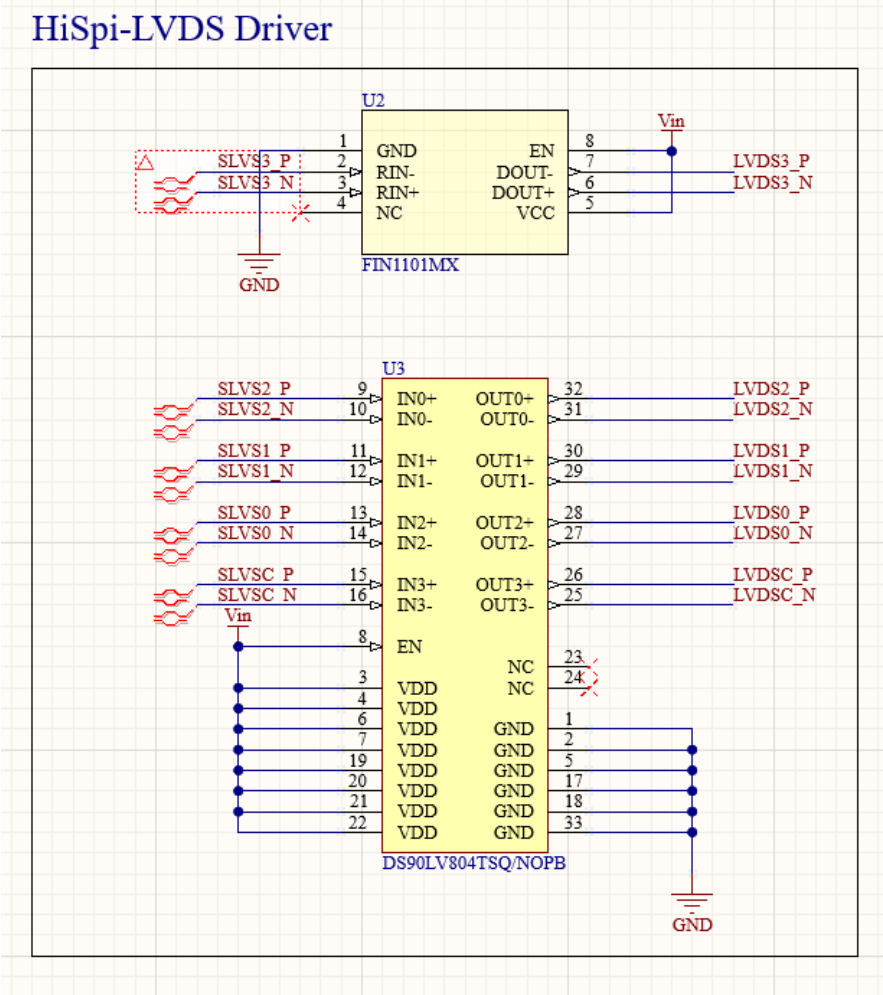


Figure 4.6: LVDS Driver Schematic

This process is contingent upon the signal integrity—most notably, the voltage swing and common-mode voltage—adhering to the prescribed input thresholds of the driver. Assuming the SLVS signal falls within the permissible range, these drivers can effectively bridge the signalling gap, enabling the FPGAs to interpret the data relayed from the AR0134CS sensor accurately. In instances where the signal exhibits excessive weakness or degradation, additional measures for signal conditioning or level adjustment might be necessary to guarantee data transmission fidelity.

The diagram on Figure 4.7 shows the typical connection between the image sensor and the driver that is acting as a repeater converting the SLVS signal to LVDS.

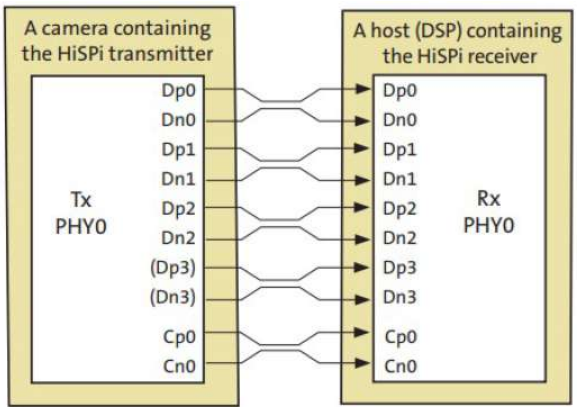


Figure 4.7: HiSPi interface, extracted from [32]

The CST Studio is an advanced electromagnetic simulation software suite used to analyse and optimise high-frequency systems, such as PCBs, connectors, and transmission lines, which are crucial for maintaining signal integrity. The CST Studio was used for the signal integrity of all image sensor lanes

Lane adaptation in signal integrity involves optimising transmission paths to ensure that electrical characteristics, such as impedance, match between the transmitter and receiver. This reduces reflections and signal degradation. Crosstalk analysis measures interference from adjacent signal lines, which can lead to transmission errors. Simulating crosstalk helps engineers optimise the design and minimise noise. Eye diagram time analysis visually represents signal quality, where a transparent, open eye indicates good signal integrity and a closed or distorted eye points to issues like noise or jitter. Together, these analyses ensure accurate high-speed data transmission.

For this, two simulations were made: one (Fig. 4.8 - Sim 1) for the connector to the LVDS driver and another (Fig. 4.8 - Sim 2) from the driver to the image sensor.

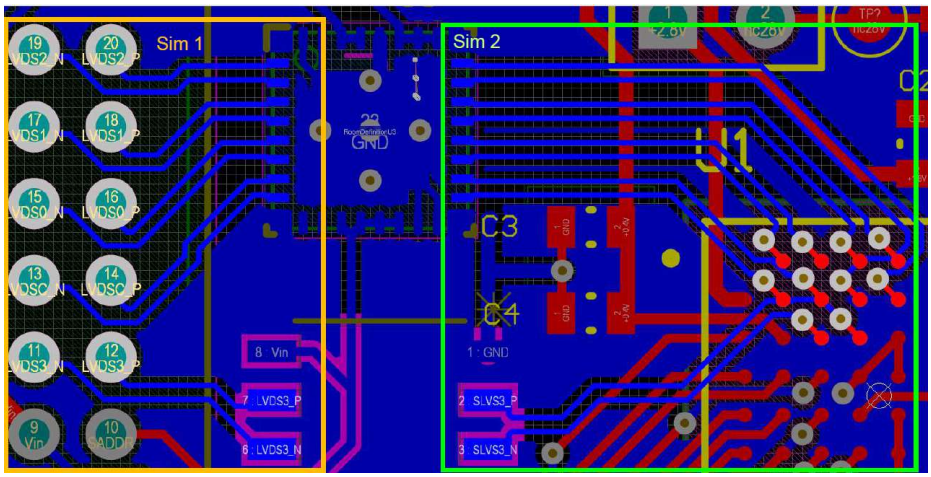


Figure 4.8: Signals for Integrity signal analysis

1. **Simulation 1 - Connector to LVDS Driver**

   For the LVDS simulation, only the worst results will be shown to avoid overwhelming the reader. Understanding the designators for the plot results is important, so Table 4.2 presents the connection between the designators and the lanes and terminals.

   Table 4.2: Connection between Designators, Pins, and Terminals - LVDS

Designator	Pin	Terminal
S1	LVDS0	Vertical Connector
S2	LVDS1	Vertical Connector
S3	LVDS2	Vertical Connector
S4	LVDS3	Vertical Connector
S5	LVDSC	Vertical Connector
S6	LVDS0	LVDS Driver
S7	LVDS1	LVDS Driver
S8	LVDS2	LVDS Driver
S9	LVDS3	LVDS Driver
S10	LVDSC	LVDS Driver

   The first simulation (Figure 4.9) shows an eye diagram that visually represents signal pulse overlap, indicating the quality of digital signal transmission. A more extensive, open "eye" generally means better signal integrity with less noise, jitter, and interference. The red lines represent the signal's voltage over time, and the purple lines define threshold voltages for a valid digital signal.

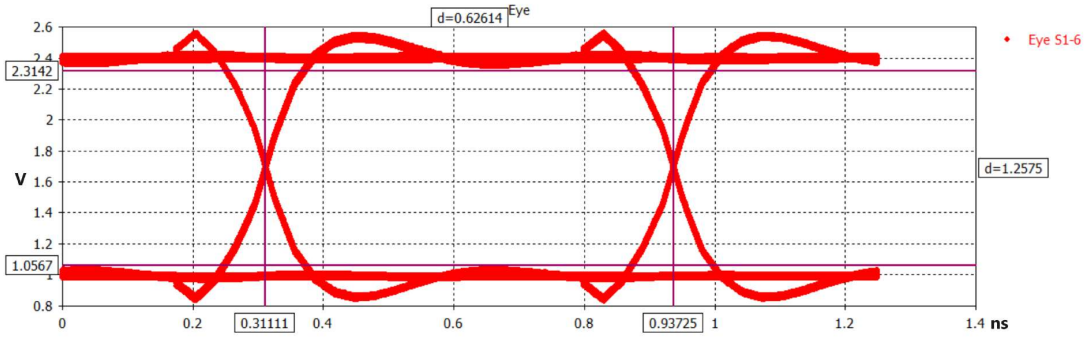


   Figure 4.9: Eye Diagram LVDS

   The timing and voltage margins help assess if the signal can be correctly interpreted at the receiver. The eye appears fairly open in this case, suggesting the

signal integrity is relatively good. Still, the width and height of the eye can be further analysed to quantify the exact performance. The values "d = 0.62614" and "d = 1.2575" represent critical timing parameters for reliable data transmission.

The S-parameters across a frequency range of 0 to 3000 MHz, is shown in Figure 4.10 and Figure 4.11. The lines represent different signal paths (S1,6; S2,7; S3,8; S4,9; and S5,10). The graph of Figure 4.10 shows a consistent, gradual increase in attenuation as the frequency increases. The attenuation is relatively small, measuring less than 0.7 dB across the entire frequency range, indicating minimal signal loss for these paths. This is typical of a well-designed transmission line, where signal degradation is primarily due to conductor and dielectric losses, which increase with frequency.

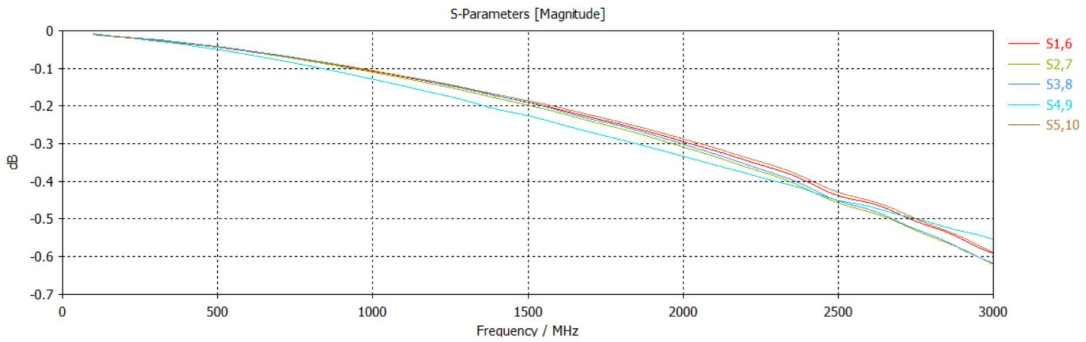


Figure 4.10: Attenuation plot LVDS lanes

The following step on Figure 4.11 involved simulating cross-talk, focusing on the S-parameters (S1,7 to S1,10) to observe transmission loss between ports 7 and 10 across the frequency range. The coupling levels are minimal, always below -50 dB. Consistent loss at higher frequencies suggests significant degradation in signal integrity, possibly due to dielectric loss or conductor roughness.

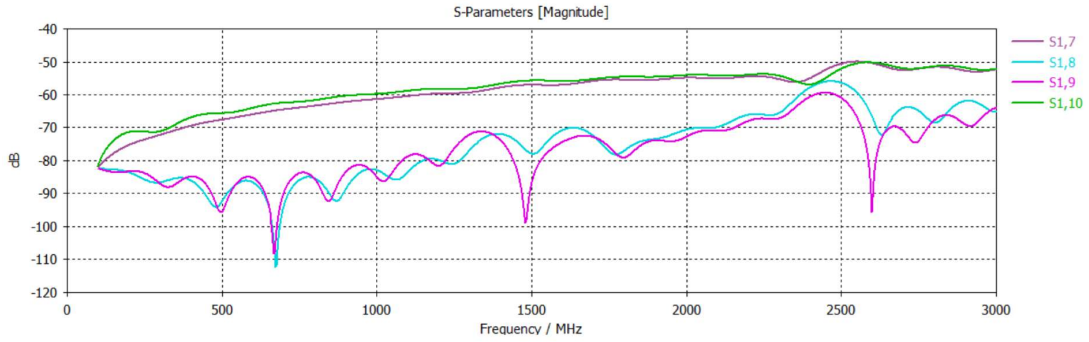


Figure 4.11: Cross-talk plot LVDS between the lanes

The following graph shows the S-parameter magnitude plotted against frequency

in MHz. The y-axis is in decibels (dB), indicating the amount of signal loss. The graph illustrates how the cross-talk increases with frequency, which is typical in high-frequency circuits. The curves suggest less couple between the lanes at lower frequencies, becoming more significant as the frequency rises to 3000 MHz. Each curve corresponds to a different transmission line on the PCB, showing that other lines may exhibit slightly different cross-talk characteristics.

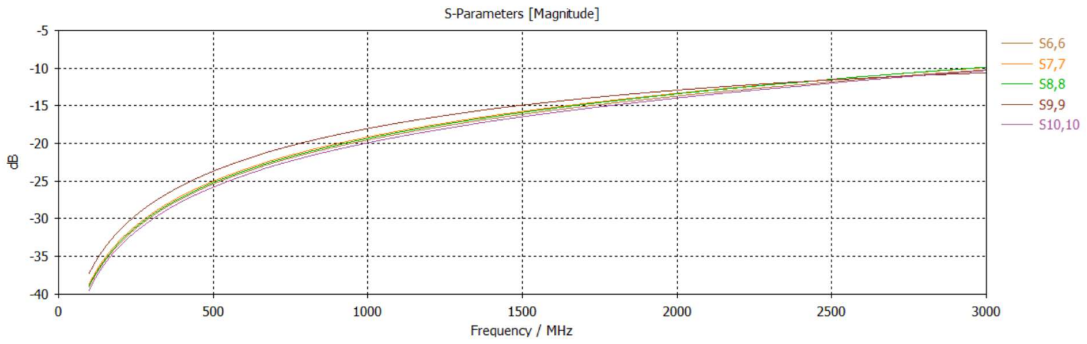


Figure 4.12: The LVDS lanes matching for 100 ohm load

From Figure 4.12 one can conclude that all lane are well matched for 100 ohm load up to 3 GHz, because S-parameters are below -10dB.

2. **Simulation 2 - LVDS Driver to Sensor**

Similar to LVDS analysis, the SLVDS signals will only show the worst results. Table 4.3 illustrates the connection between the designators, lanes, and terminals.

Table 4.3: Connection between Designators, Pins, and Directions - SLVDS

Designator	Pin	Direction
S1	SLVDS0	AR0134CS
S2	SLVDS1	AR0134CS
S3	SLVDS2	AR0134CS
S4	SLVDS3	AR0134CS
S5	SLVDSC	AR0134CS
S6	SLVDS0	LVDS Driver
S7	SLVDS1	LVDS Driver
S8	SLVDS2	LVDS Driver
S9	SLVDS3	LVDS Driver
S10	SLVDSC	LVDS Driver

The graph in Figure 4.13 shows S-parameters over a frequency range of up to

3000 MHz. The mesh pass establish that this simulation was done with a specific mesh resolution (pass = 1). The increasing magnitude of the S-parameters indicates how signals are attenuated with frequency. These curves with varied shapes and levels imply different transmission lines or paths have distinct attenuation profiles. Higher values near 0 dB suggest stronger signal reflections or losses, while the more negative values indicate better signal transmission with less reflection. Since all signals are below -25 dB, this case shows that the signal transmission is suitable for the HiSPi transmission.

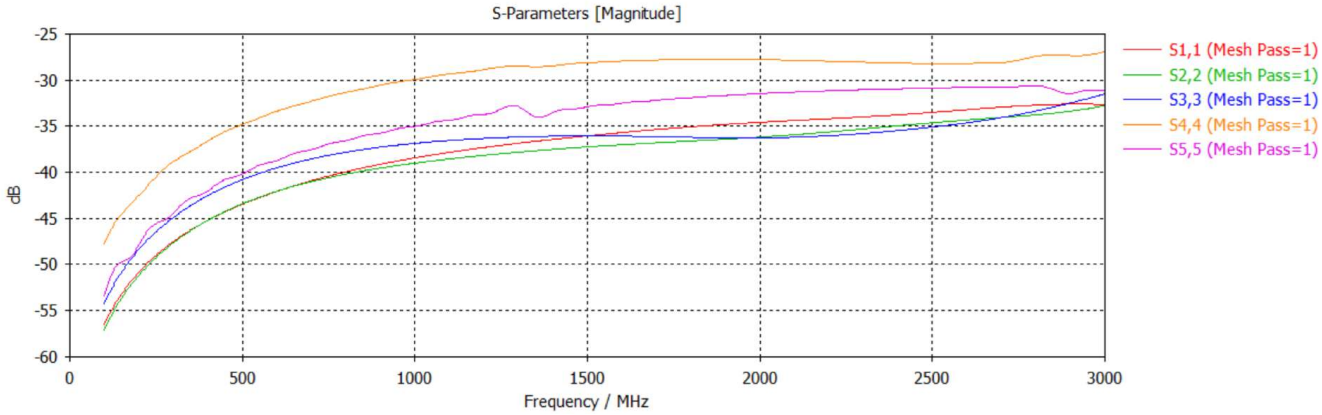


Figure 4.13: SLVS lanes matching for 100 ohms load

The image on Figure 4.14. shows the S-parameters (S7,1 to S7,5) under the same frequency range but for a different mesh pass simulation. The curves indicate that higher-frequency signals experience higher cross-talk, however, it is well below the -30dB, especially in S7,1 (red curve), which starts with very low coupling at lower frequencies. With that said, the coupling is bellow the -40dB which represents that our signals do not suffer that much signal degradation.

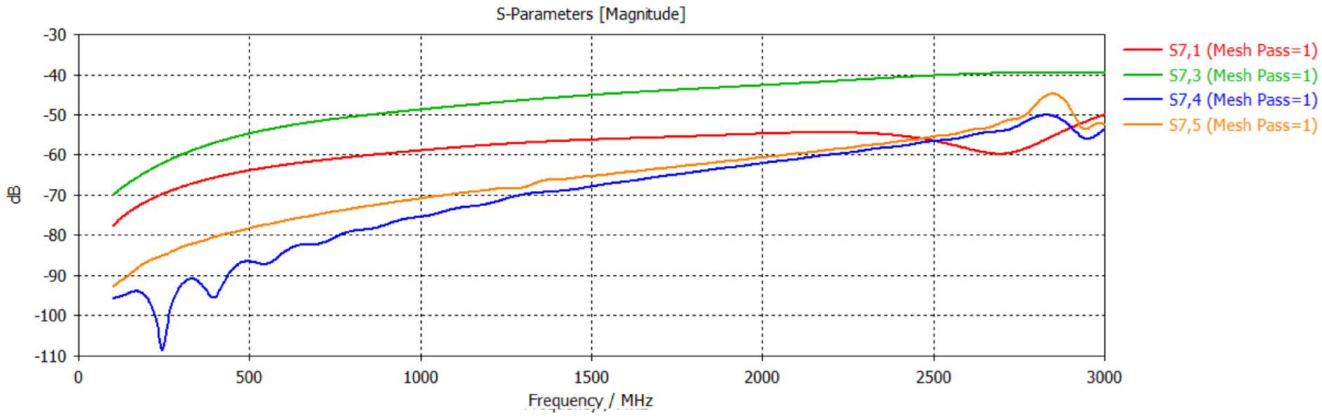


Figure 4.14: Cross-talk plot SLVDS

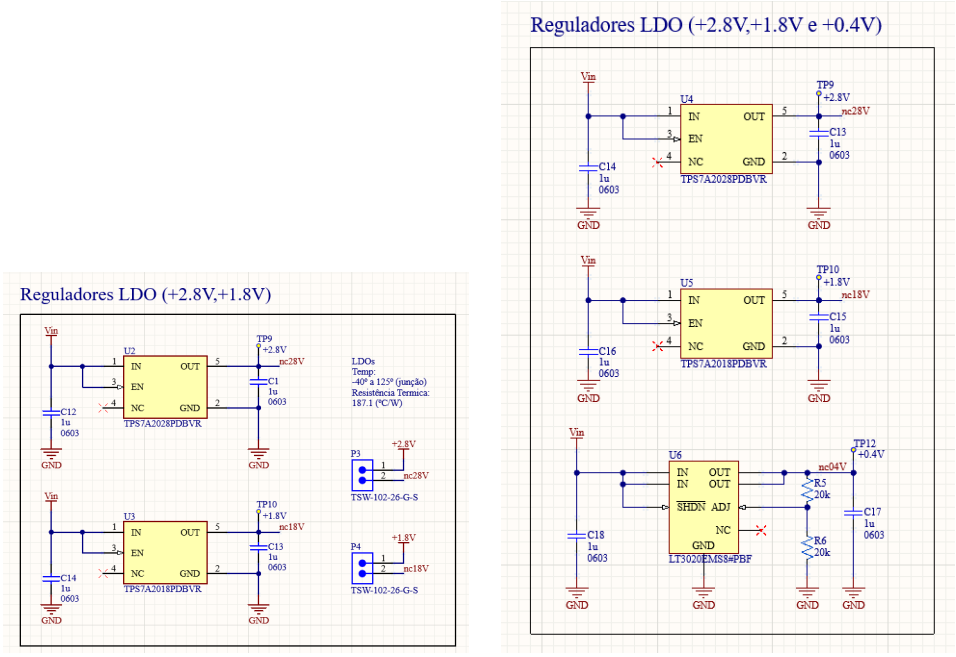
This simulations help us conclude that the HiSPi interface is well matched and experiences minimal coupling between the lanes, providing a high-quality signal integrity.

4.1.4 Power Supply

The power input voltage of both Sensor PCB Boards is recommended to be Direct Current (DC) +3.3V, but it is prepared to go from 3V to 5.5V. The input power generates +2.8V through the voltage regulator TPS7A2028PDBVR for analogue, pixel and Input Output (IO) power. The input power also generates +1.8V through the voltage regulator TPS7A2018PDVR for digital power reference. These regulators circuits can be seen in Figure 4.15b and Figure 4.15a.

The maximum current rating for this sensor when streaming is approximated 200mA, 127mA for 2.8V and 60mA for 1.8V, giving, for a parallel output, a 0.46W power consumption.

For the BGA sensor with HiSPi interface to function, the input power is used to supply both the LVDS drivers (FIN1101 and DS90LV804TSQ) and the voltage regulator chip LT3020EMS8, which generates a 0.4V output. This increases the total consumption power very little as these chips and interfaces use very little current, around 5-15uA.



(a) LCC Power Schematic (b) BGA Power Schematic

Figure 4.15: Power supply schematic

These are low dropout voltage regulators with low power dissipation, so the heat and the low noise output for the sensitive components, such as the image sensor and its outputs, are minimized.

A block diagram of the power architecture can be seen in Figure 4.16.

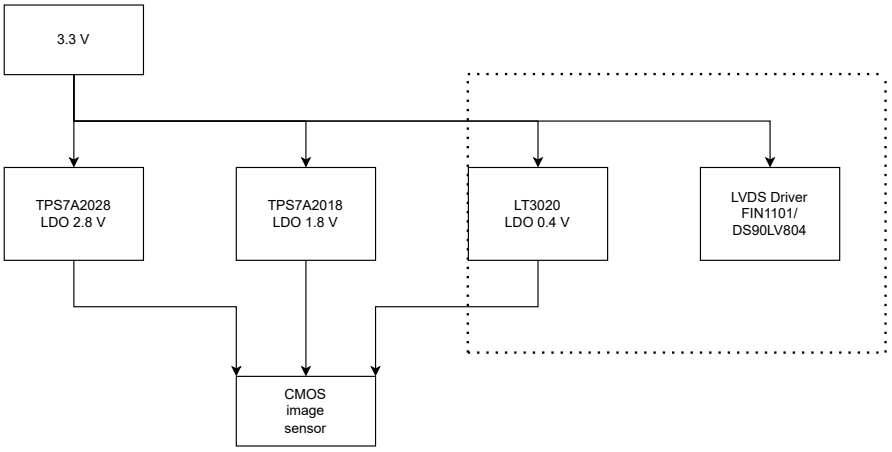


Figure 4.16: Sensor Power Supply Diagram

The datasheet specifies the power-up(Figure4.17) and power-down(Figure4.18) sequence, indicating that the interval between each voltage should be 0 seconds(t0, t1, t2 and t3 for both power on and off). This implies that a dedicated power sequencer is unnecessary to ensure the correct order of voltage application. All voltage regulators can be turned on or off simultaneously, provided the 2.8V supply is present when the 1.8V and 0.4V supplies are active. The absence of 2.8V while 1.8V or 0.4V is on can lead to operational issues, making the simultaneous presence of 2.8V essential for stable operation.

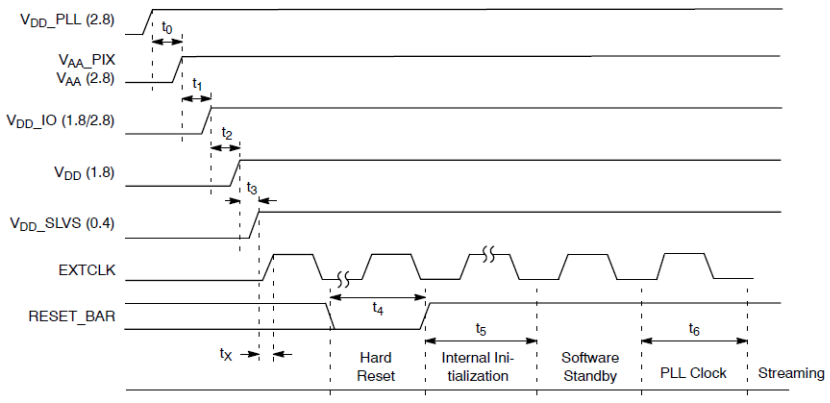


Figure 4.17: Power-Up Sequence, extracted from [27]

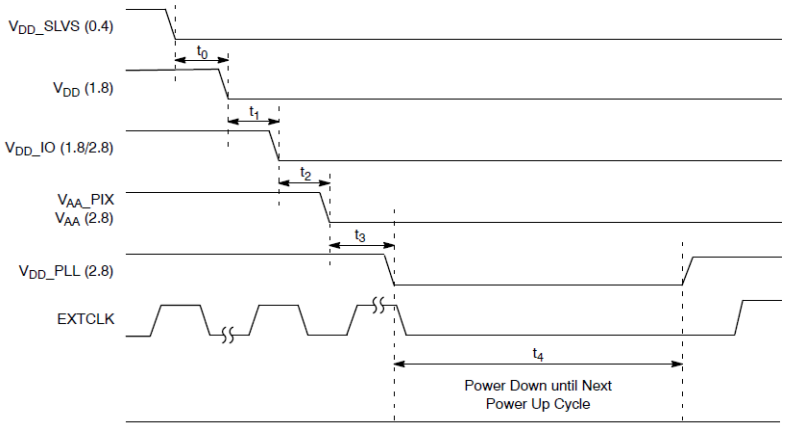


Figure 4.18: Power-Down Sequence, extracted from [27]

4.1.5 Image Sensor PCB layout

The sensor prototype was designed in Altium Designer on a 2-layer PCB for an LCC package(Figure 4.19) and a 4-layer PCB for a BGA package(Figure 4.21), which Sinuta later manufactured.

In the case of the LCC sensor, the setup was straightforward. The components were positioned on the top layer, with most of the traces being routed here. The bottom layer was used solely for certain tracks that overlapped when attempting to reach their destination. Additionally, the bottom layer featured a GND polygon covering the entire board area.

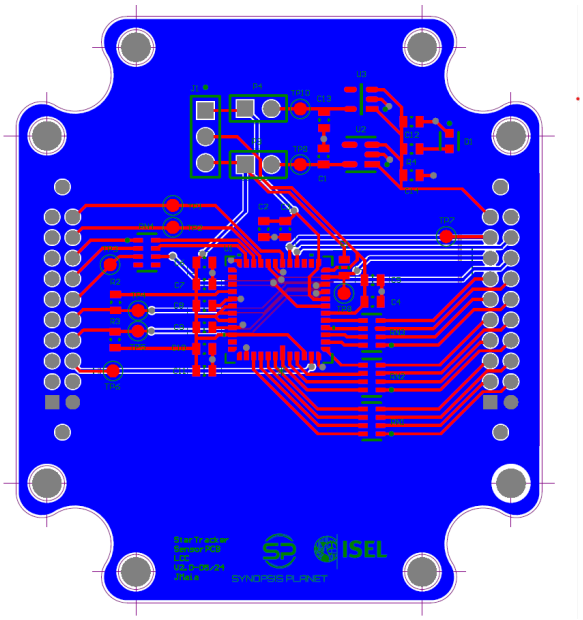


Figure 4.19: PCB Sensor LCC

The decision to use a 4-layer PCB for the BGA package was influenced by the sensor's pinout. It was challenging to create the sensor's fanout (Figure 4.20) on just one layer, so using four layers allowed for better isolation of signals from the HiSPi interface and reduced interference from other components. The layers can be seen in Appendix 1—Copper layers of the PCB.

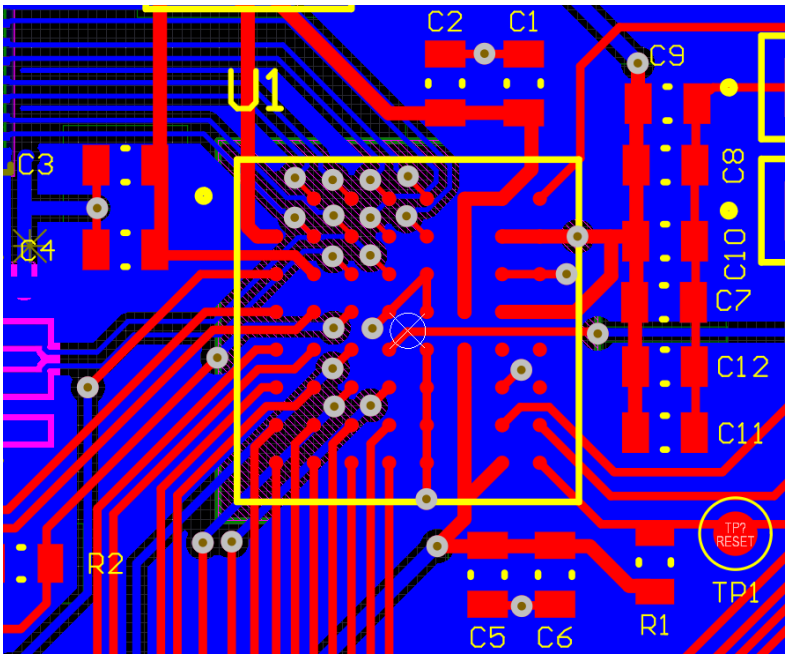


Figure 4.20: BGA Sensor Fanout

The top layer (layer 1) and bottom layer (layer 4) were employed to interconnect surface-mount components positioned close to each other, thereby reducing the need for vias and simplifying the routing of power supply traces. The power traces were intentionally designed with a wider width to reduce resistance and inductance. Additionally, a star-shaped layout was taken into consideration to avoid making 90° angles on the tracks.

The top layer handled the parallel interface and the communication protocol I2C, while the bottom layer was dedicated to the serial interface.

Layers 2, 3, and 4 featured a GND polygon pouring over the entire board area, with layer 3 specifically used for low-impedance ground connections and to maintain the impedance between the differential lanes of the serial interface between 50 and 75 Ω and layer 4 designed to shield the differential lanes, minimizing crosstalk interference and ensuring signal integrity.

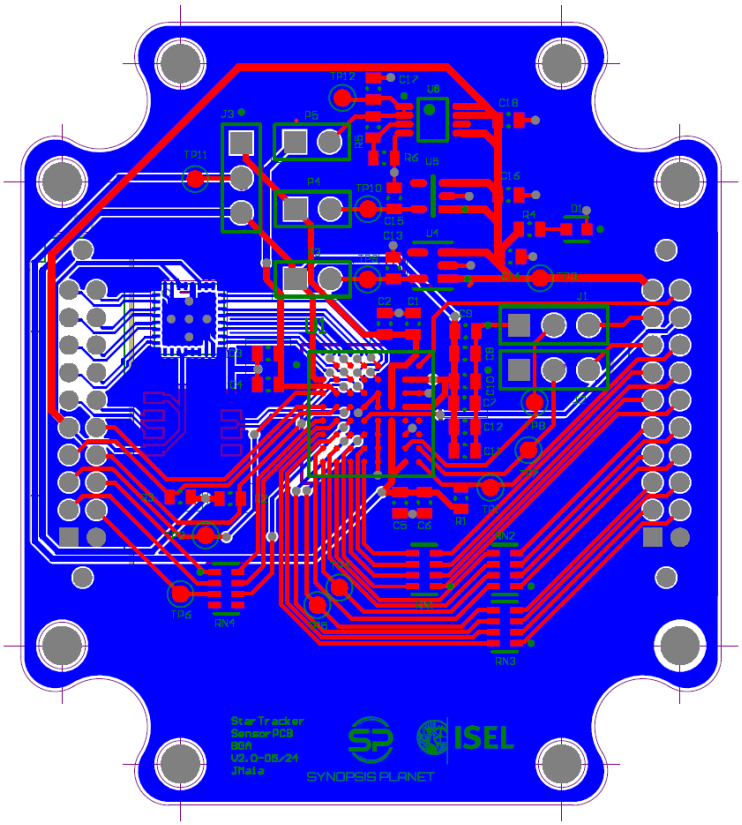


Figure 4.21: PCB Sensor BGA

4.2 FPGA Board

The star tracker's schematic was designed based on the Alynx AX7350 [38]. This kit incorporates XILINX's Zynq7000 SOC chip XC7Z035 solution. The ZYNQ7Z035 chip features a Processor System (PS) and Programmable Logic (PL). On both the PS and PL sides of the ZYNQ7Z035 chip, there are two DDR3s, each with a capacity of up to 512 Mbytes.

This board offers a variety of peripheral circuit options, including an HDMI output interface for displaying captured images, a UART serial interface for software debugging, and an FMC expansion interface. The diagram below illustrates the relevant components of this kit and their connections to the image sensor board.

This section provides a comprehensive overview of each block utilized in the FPGA board schematic, along with references to the hardware components illustrated in the accompanying block diagram represented on Figure 4.22.

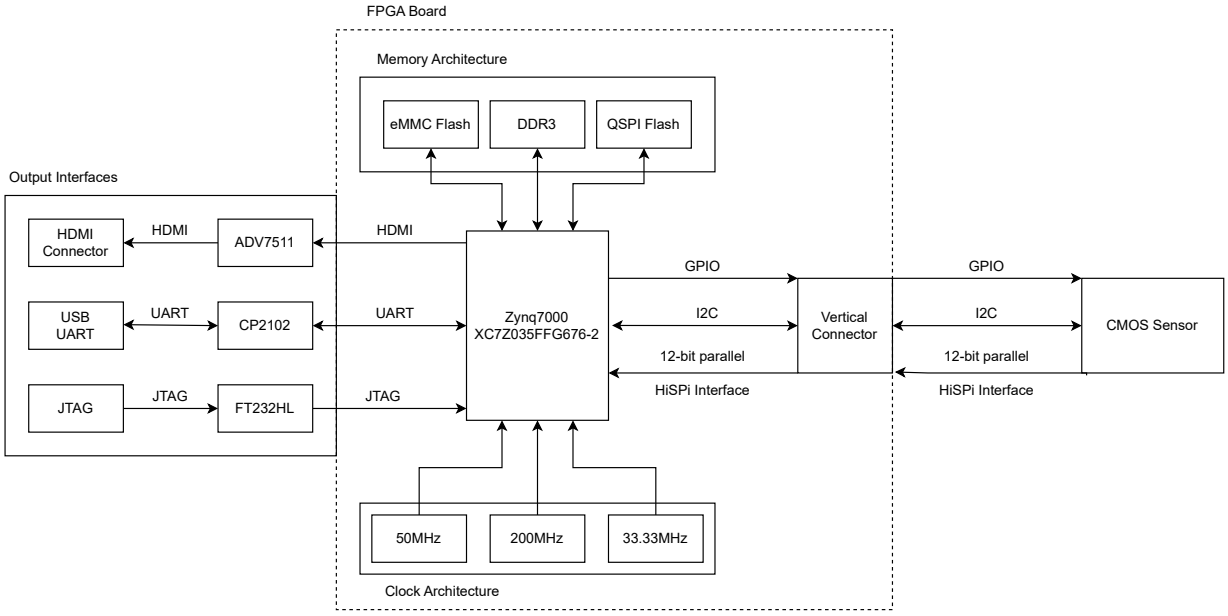


Figure 4.22: FPGA Diagram

To enhance clarity, the components, nets, and connections from the electronic schematic are represented as simplified block diagrams. This approach effectively addresses the complexity and size of the actual schematic, which is too intricate to fit on a single page and challenging to interpret fully. For the complete schematics, please consult Appendix D and Appendix E.

4.2.1 Memory Architecture

In FPGA-based systems, the architecture of memory is vital for ensuring that the programmable logic (PL) and processing system (PS) have the appropriate storage resources for their specific roles. Systems that run operating systems like Linux and perform tasks such as real-time data acquisition or image processing depend on various types of memory, each optimized for particular functions within the FPGA system.

The memory types typically employed in these projects include eMMC Flash, QSPI Flash, and DDR3 DRAM, with each fulfilling a distinct role:

- **DDR3 DRAM** provides high-speed, volatile memory, facilitating the execution of applications, buffering data, and ensuring rapid data access for both the PS and PL.
- **eMMC Flash** primarily serves as mass storage, containing the Linux operating system, applications, and user data.

- **QSPI Flash** acts as non-volatile memory, storing the FPGA configuration bitstream and essential bootloaders, enabling the FPGA to independently load its configuration during startup.

By integrating all three memory types into FPGA designs, systems can operate efficiently across various tasks, from booting up to real-time data processing. Section 4.2.1.1, Section 4.2.1.2, and Section 4.2.1.3 will delve into each memory type, analyzing its characteristics and contribution to the overall functionality of FPGA systems that handle complex applications.

4.2.1.1 DDR3 Memory

The system includes four large-capacity 512MB (2GB total) high-speed DDR3 SDRAM modules. Two of these modules are installed on the PS side, where they can serve as a buffer for ZYNQ chip data or as memory for the operating system.

The other two modules are located on the PL end and can be used for data storage, image analysis cache, and data processing within the FPGA.

The DDR3 memory modules utilize the Micron 512MB model MT41J256M16HA-125. The PS-side DDR3 SDRAM operates at a maximum speed of 533MHz (1066Mbps data rate) and is directly connected to the memory interface of BANK 502 (Figure 4.23) in the ZYNQ Processing System (PS). On the other hand, the PL-side DDR3 SDRAM operates at a maximum speed of 800MHz (1600Mbps data rate) and is connected to the BANK33 and BANK34 interfaces of the FPGA (Figure 4.24).

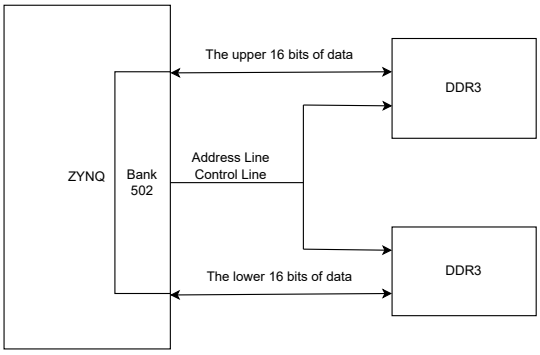


Figure 4.23: DDR PS Diagram

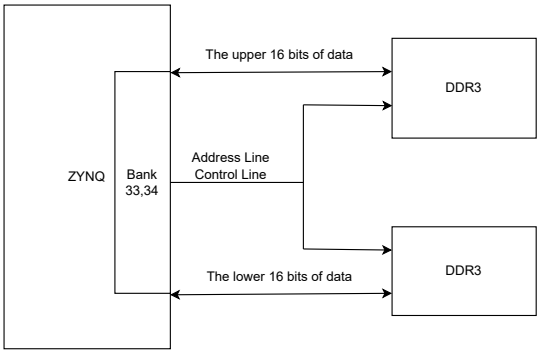


Figure 4.24: DDR PL Diagram

In designing the DDR3 hardware, it is crucial to consider signal integrity. High-speed and stable operation require careful attention to matching resistor/terminal resistance, trace impedance control, and trace length control in the circuit and PCB design.

4.2.1.2 eMMC Flash

The star tracker is equipped with a large-capacity 32GB eMMC FLASH chip, model THGBMFG6C1LBAIL, which supports the JEDEC e-MMC V5.0 standard HS-MMC interface with level support of 1.8V or 3.3V. The data width of the eMMC FLASH and ZYNQ connections is 4 bits. Due to its large capacity and non-volatile nature, eMMC FLASH can be used as a large-capacity storage device for the ZYNQ system, such as ARM applications, system files, and other user data files. Figure 4.25 shows the eMMC Flash in the schematic.

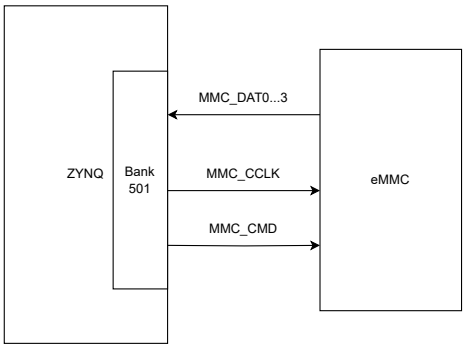


Figure 4.25: eMMC Flash Diagram

4.2.1.3 QSPI Flash

The current project has a 256Mbit Quad-SPI FLASH chip(Figure 4.26), model W25Q256FVEI, which uses the 3.3V CMOS voltage standard. Due to its non-volatile nature, QSPI FLASH is mainly used as a boot device to store the system's boot image. These images include FPGA bit files, ARM application code, and other user data files.

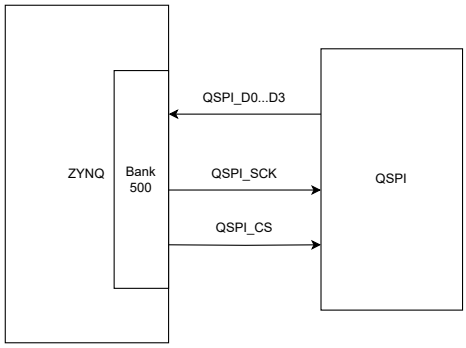


Figure 4.26: QSPI Flash Diagram

4.2.2 Output Interfaces

The FPGA PCB for the star tracker platform was designed with several output interfaces to support system functionality, testing, and debugging. These interfaces include an HDMI video output, UART for serial communication, and a JTAG interface for configuration and debugging. Due to space limitations on the star tracker's FPGA board, it was not feasible to include the necessary driver circuitry for HDMI, UART, and JTAG directly on the PCB. Therefore, an EGSE (Electrical Ground Support Equipment) system will be developed to provide these interfaces externally, supporting flexible testing and debugging of the platform during development.

To interface with the image sensor, there were two solutions provided. One where was developed a daugtherboard, and another with two smaller vertical 2x10 through-hole pin connectors for connection between the image sensor PCB and FPGA PCB.

4.2.2.1 Expansion port

The initial design for the AX7350B FPGA kit employed a standard FMC LPC connector for linking to a daughterboard to interface with the sensor to debug it. This standard FMC LPC expansion port provided access to 34 differential pairs and one high-speed

GTX transceiver that could connect the development board to various FMC modules, such as the image sensor.

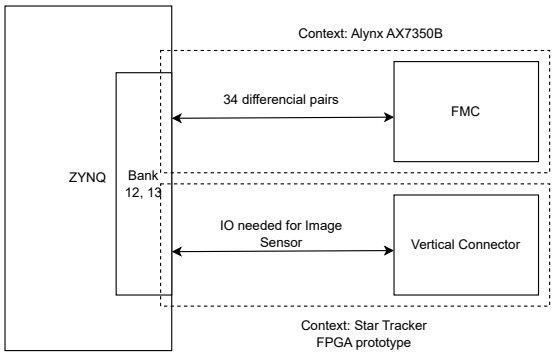


Figure 4.27: Expansion Port Diagram

However, as this level of connectivity proved to be more than necessary and to conserve space on the PCB, the final design replaced the FMC with two smaller vertical 2x10 through-hole pin connectors as shown in Figure 4.27. This streamlined solution continues to facilitate reliable communication between the FPGA PCB and the sensor PCB in the star tracker prototype.

4.2.2.2 HDMI video output

The HDMI output interface is implemented using ADALOG DEVICE's HDMI (DVI) encoding chip, which supports 1080P@60Hz output. The video digital interface, audio digital interface, and I2C configuration interface of the chip are connected to the BANK35 IO of the PL part. The system initializes and controls the chip through the I2C pin. The hardware connection diagram of the chip is shown in Figure 4.28.

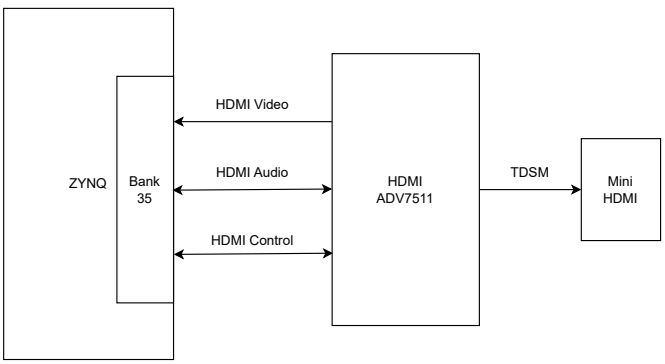


Figure 4.28: HDMI Diagram

4.2.2.3 USB UART bridge

The device is equipped with 2-way UART to USB interfaces, enabling communication with the computer for user debugging purposes. The conversion chip utilizes the USB-UART chip from Silicon Labs CP2102GM (4.29). The USB interface features a MINI USB interface and can be connected to the PC's USB port using a USB cable for separate power supply and serial data communication with the core board.

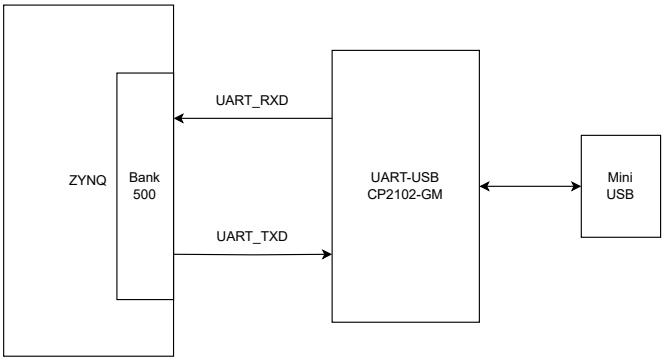


Figure 4.29: UART Diagram

4.2.2.4 USB JTAG bridge

The FPGA development board AX7350B includes integrated JTAG download and debug circuitry, which eliminates the need for an external Xilinx downloader. This feature should be implemented on the star tracker board to facilitate debugging and configuration during the initial prototype phase. To connect the PC's USB to the JTAG debug signals, an FTDI USB bridge chip FT232HL was selected to ensure seamless data communication with the ZYNQ FPGA.

4.2.3 Clocks design

The FPGA board features an active clock for both the PS system and the PL logic, enabling them to operate independently. The ZYNQ chip supplies a 33.333MHz clock input to the PS section via the X4 crystal on the development board. This clock input is connected to the PS_CLK_500 pin of the BANK500 of the ZYNQ chip. The schematic diagram is illustrated in Figure 4.30.

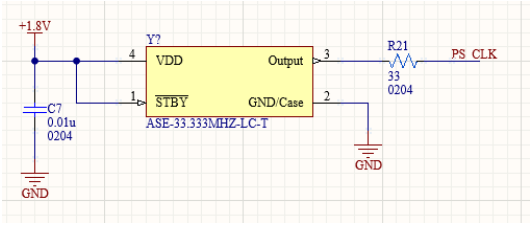


Figure 4.30: Program System Clock

Furthermore, the AX7350B development board offers a single-ended 50MHz PL system clock source with a 1.8V supply. The crystal output is linked to the global clock (MRCC) of the FPGA BANK35, which can be utilized to drive the user logic circuit within the FPGA. The schematic diagram of the clock source is depicted in Figure 4.31.

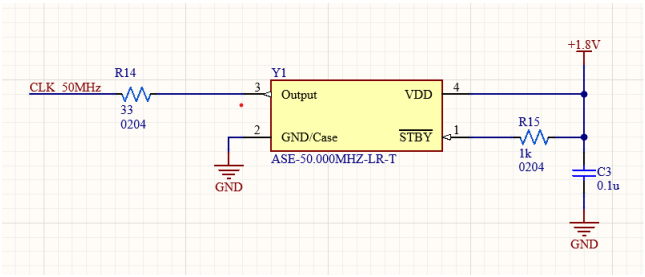


Figure 4.31: Program Logic Clock

Additionally, a 200MHz differential crystal is provided to BANK34 as a reference clock for the PL's DDR controller(Figure 4.32).

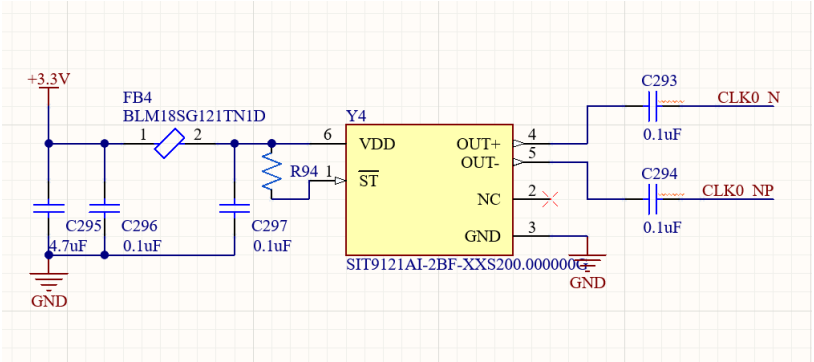


Figure 4.32: System Clock

4.2.4 Power Supply

This board will consume a high amount of power due to the high demand of ZYNQ Core Voltage. The Table 4.4 shows the voltages which the board operates:

Table 4.4: Power Supply Functions

Power Supply	Function
+1.0V	ZYNQ PS and PL section Core Voltage
+1.8V	ZYNQ PS and PL partial auxiliary voltage, BANK501 IO Voltage, HDMI
+3.3V	ZYNQ Bank0, Bank500, Clock Crystal
+1.5V	DDR3, ZYNQ Bank501, Bank33, Bank34
+2.5V	ZYNQ Bank12, Bank13, FMC
+0.75V	PS DDR3, PL DDR3
+1.0V MGTAVCC	ZYNQ Bank111, Bank112
+1.2V MGTAVTT	ZYNQ Bank111, Bank112

The ZYNQ FPGA's power supply necessitates a specific power-on sequence, so a circuit was devised to adhere to the chip's power requirements. The power-on sequence entails +1.0V->+1.8V->(+1.5 V, +3.3V, VCCIO) to ensure the chip operates normally.

Regarding power consumption, it is contingent on the hardware platform and sensor usage. The kit's manual indicated that the core could consume up to 20W, but this was in an unrealistic scenario where all peripherals, including those not utilized in the star tracker implementation, were operational.

It is also worth mentioning that both hardware platforms outlined in Chapter 5.1, indicate that the kit, for the configuration and data acquisition, only consumes 2W of power.

It is also important to note that this FPGA requires at least 10 layers due to the 23x23 pad configuration beneath the package, which features very tight spacing between the pads.

4.2.5 FPGA PCB

To complete the FPGA schematics, a custom PCB was designed, as illustrated in Figure 4.22. Due to space limitations, the output interfaces and power supply components have been placed on a separate board. Ideally, these interfaces should only be integrated into the Electrical Ground Support Equipment Electrical Ground Support Equipment (EGSE) interface for our system or on the OBCS board.

For the FPGA fanout, several guidelines were established based on references [34], [35] and [36]:

- Use a minimum of 10 signal layers to accommodate the high number of BGA (Ball Grid Array) pads on the FPGA.
- Vias should have an outer diameter of 19 mils and a hole diameter of 10 mils.
- Track width should be 3 mils to allow routing between pads.
- Clearance between tracks should be 3 mils, permitting up to two tracks to pass between pads.

Following these basic design guidelines for the FPGA, the DDR3 DRAM was placed on the top layer next to the FPGA, while the eMMC Flash and QSPI Flash were positioned on the bottom layer to make optimal use of space. Due to the high component density and the routing complexity within these spatial constraints, blind vias were used, and two additional signal layers were added to support the design.

It should be noted that the routing in this PCB design is preliminary and not fully optimised, particularly for the memory components, which require careful routing to maintain signal integrity. This PCB layout is intended to verify that all components can physically fit on a single board, excluding the power supply and output interfaces.

Figure 4.35 shows the resulting 3D views of the top and bottom layers, while Figure 4.33 and Figure 4.34 depicts the 2D layer stack-up of top and bottom layer, respectively.

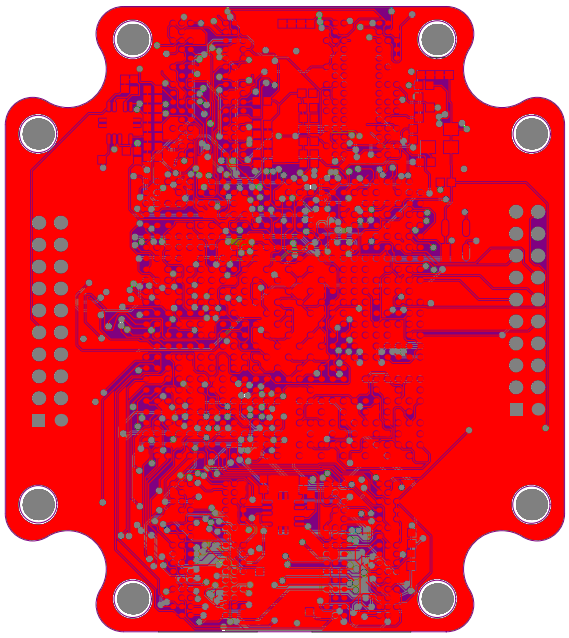


Figure 4.33: FPGA PCB Top layer

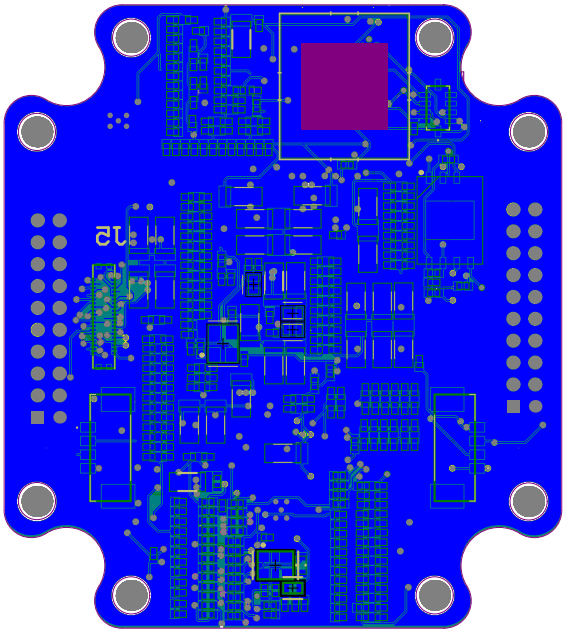


Figure 4.34: FPGA PCB Bottom layer

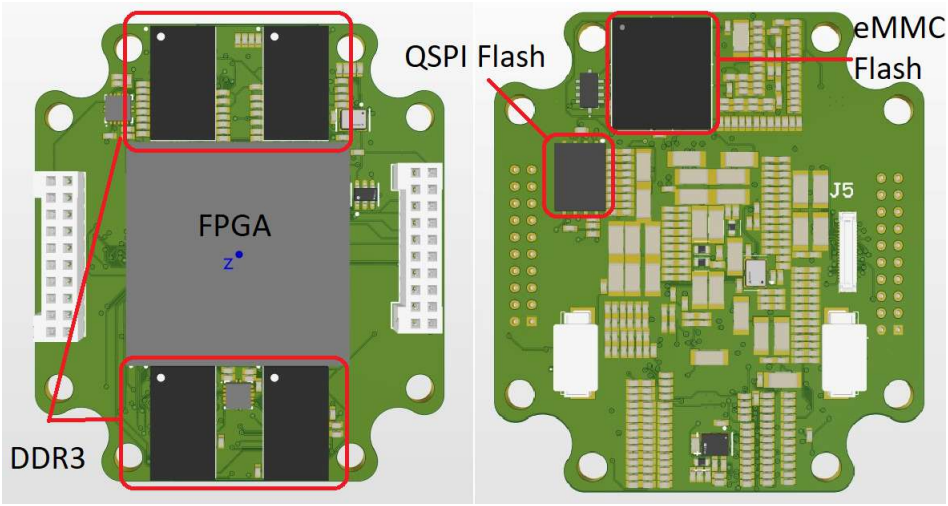


Figure 4.35: 3D image of FPGA PCB

These images demonstrate the component density on a compact circuit board, emphasizing the challenges associated with routing. This complexity underscores the necessity for a multilayer design, specifically 14 layers, to accommodate the spatial constraints and ensure proper functionality.

4.2.6 Daughter Board

To test the sensor and the designed sensor PCBs, it was used the AX7350B FPGA development board. It was aimed to connect the sensor board to an FPGA kit using the

available FPGA Mezzanine Card (FMC) connector. However, as they are significant, the sensor PCB did not possess an FMC connector. Instead, it featured two vertical sockets, each with 20 pins to connect to another board with an identical shape.

Upon reviewing the schematic of the FPGA kit, it was found that the FMC expansion port includes 34 pairs of differential I/O signals and a high-speed GTX transceiver signal. These 34 differential signals are routed to the I/O pins of BANK12 and BANK13 on the ZYNQ chip. The I/O voltage standard for these banks is set by the voltage VADJ, which defaults to 2.5V. This configuration supports LVDS data communication, similar to the HiSPi interface.

It was crucial to carefully connect the positive differential signals from the sensor to the FPGA to ensure proper communication between the sensor and the FPGA. Additionally, it was needed to identify the correct 3.3V pin to power the sensor board while avoiding using any clock pins, as this could interfere with the FPGA's operation.

To address these challenges, it was designed a custom PCB with two vertical sockets to match the sensor's connections, and an FMC connector to interface with the FPGA kit (Figure 4.36). This PCB bridges the sensor's I/Os with the appropriate I/Os on the FPGA, ensuring proper alignment with the differential signal pairs and the necessary power supply.

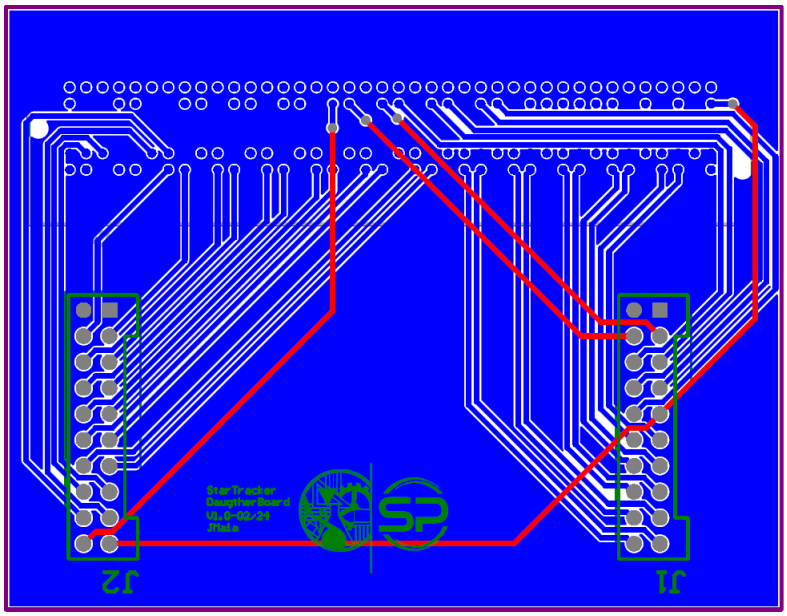


Figure 4.36: Daughter board PCB

For this board it was also made a CST analysis to verify the signal integrity. Table 4.5 displays the connection between the designators and the lanes and terminals.

Table 4.5: Connection between Designators, Pins, and Directions - Daugther Board

Designator	Pin	Direction
S1	SLVS0	Vertical Connector
S2	SLVS1	Vertical Connector
S3	SLVS2	Vertical Connector
S4	SLVS3	Vertical Connector
S5	SLVSC	Vertical Connector
S6	SLVS0	FMC
S7	SLVS1	FMC
S8	SLVS2	FMC
S9	SLVS3	FMC
S10	SLVSC	FMC

The first test, shown in Figure 4.13, examines the cross-talk between the lines in the frequency range of 0 to 3000 MHz. The S-parameters indicate the transmission characteristics between port SLVSC (FMC connector) and other ports SLVS0 – SLVS3 (Vertical Connector). The curves exhibit various dips and peaks, reflecting frequency-dependent losses and interference. A sharp drop below -40 dB implies strong attenuation or isolation. For instance, S10,1 (blue curve) shows substantial attenuation starting around 500 MHz. However, between 1000MHz and 2000MHz, some peaks imply there might be interference between the signals.

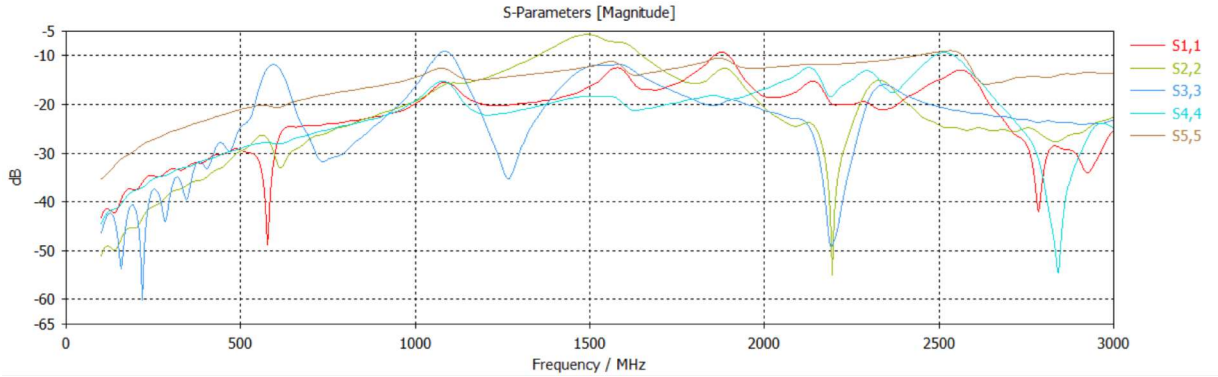


Figure 4.37: Daughter board cross-talk

Figure 4.14 shows the results of the return loss measurements, which were conducted to assess the reflections from various ports on the PCB. The plot revealed significant variations in return loss across frequencies, with specific frequencies, such as around 1000 and 2000 MHz, experiencing higher reflection. These peaks and dips are likely due to impedance mismatches in the PCB design or at connector interfaces, resulting in signal reflections. A lower dB value signifies better matching and reduced

reflection, which are essential for maintaining signal quality, especially at higher frequencies. Based on these results, it was observed that the serial clock frequency should be below 400 MHz (1200MHz/3, as the frequency on the x-axis represents the third harmonic wave) so that all the signals could be around -20 dB.

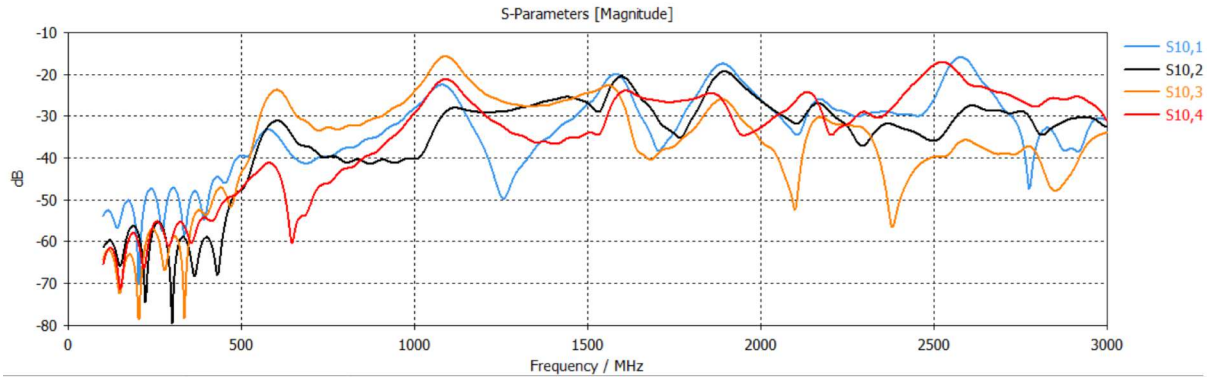


Figure 4.38: Daughter board attenuation plot

5

Software Development and Synthesis

The software design and synthesys is an important step so that the capture of the image on the FPGA is possible. This process was marked by the creation of the Vivado platform, which facilitated the configuration of the FPGA and the integration of all requisite peripherals. This phase was succeeded by the sensor configuration, employing necessary Input/Output configurations and the development of a two-wire protocol to enable communication with the sensor. Subsequently, the configuration of the camera's resolution and streaming frequency was executed through I2C communication, commanding the camera to initiate photo and video capture operations.

The following stages encompassed the reception of the data stream, its storage within DDR3 memory, and its retrieval from the DDR3 storage medium. The final step of this process involved outputting the retrieved data stream through HDMI, thus completing the cycle from data capture to display.

The following sections will describe this process, starting with the Vivado platform, followed by the i2c protocol definition, and ending with the software design developed to capture an image.

5.1 Vivado Design

The first thing it is needed to do is create the Vivado platform, this will receive the images from the AR0134CS.

To create the block diagram it will be used mostly IP cores from the Vivado library. Within this design it was included the following IP:

- Zynq MPSoC Processing System (1) - Processing system configured to have a system clk of 50MHz;
- Clock Wizard (2) – to convert the 50MHz Clock into the 27MHz Clock for the image sensor;
- AXI GPIO (3) - Three AXI GPIO IPs were added to address different issues, the clock enable, the custom I2C made in the software design and the control and data signals from the sensor.

This approach allowed to overpass the visualization of the image on the screen but it made it possible to configure the sensor and verify if it was working properly making it possible to see on the osciloscope the parallel data signals such as PIXCLK, FRAME_VALID, LINE_VALID and the 12-bit Data.

The diagram on Figure 5.1 show the block diagram that was described and generates the hardware platform.

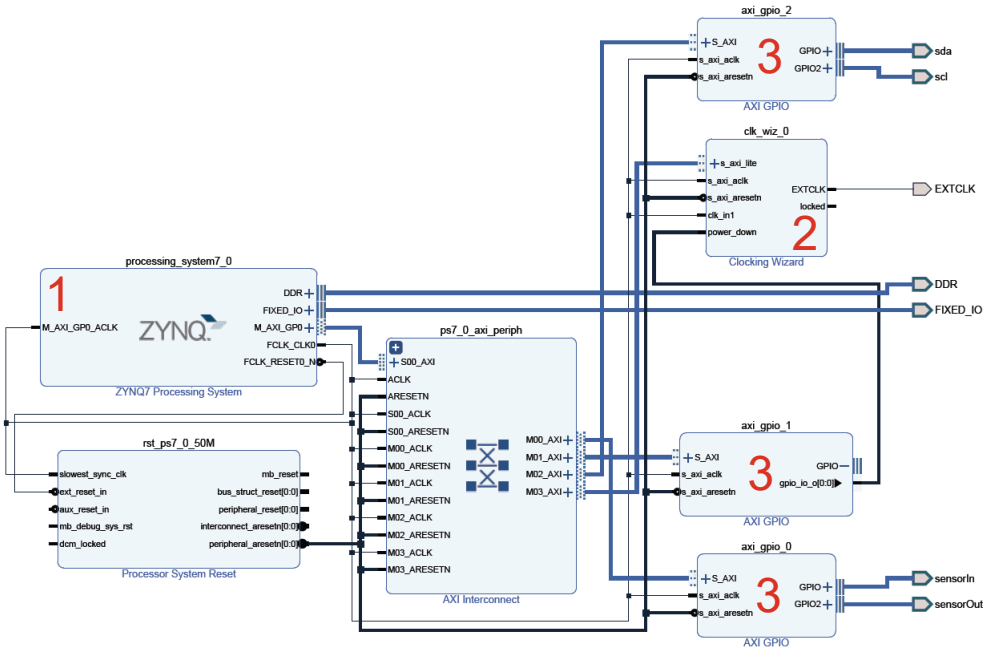


Figure 5.1: Hardware Platform for simple testing

The development of a more comprehensive version of the Vivado platform is currently in progress, but it has not been implemented yet. This updated version aims

to include the HDMI peripheral, along with the necessary peripherals for its configuration and proper synchronization. Additionally, it plans to integrate support for saving images in DDR3 memory using Video Direct Memory Access (VDMA). As part of this ongoing work, the GPIO configuration will be revised to use the EMIO capability instead of AXI GPIO, with the intention of improving system efficiency and achieving better synchronization between the pixel clock and image data. However, these enhancements are still under development and remain to be implemented.

The following IP Cores will be needed in this implementation:

- Zynq MPSoC Processing System - Processing system configured to enable the I2C(for HDMI configuration) and GPIO, EMIO;
- Video In to Axis – converts the raw 12-bit data from the sensor to AXI Stream;
- Sensor Demosaic - Convert the RAW pixel format into a RGB pixel format;
- VDMA - Writes the image to DDR memory in the processor system;
- Video Timing Generator - Generates the Output video timing;
- AXIS to Video Out - Converts from AXI Stream to parallel video;
- Clock Wizards - used to generate the video pixel clock and AR0134 EXTCLK;

Along with the IP it also needs to consider the clocking architecture, using a Dynamic Clock IP, for this solution it will be needed to implement at least the following clocks:

- 142 MHz - This clocks is for the AXI Stream and the AXI lite interface and to read data from VDMA;
- Pixel Clock - 74.25 Mhz - used for the image sensor - generated by clock wizard but it may vary based on the resolution and EXTCLK.

For the final stage, it will also need to be added the necessary peripherals to enable the HiSPi Interface to work in the same way as the parallel interface. This involves saving the image in the DDR3 and transmitting it to a screen over HDMI.

5.1.1 Two-wire serial interface

The two-wire serial interface is based on the I2C communication protocol, with a few differences. Data transfers on the two-wire serial interface bus involve a sequence of low-level protocol elements:

1. A (repeated) start condition
2. An enslaved person's address/data direction byte
3. An (a no) acknowledge bit
4. A message byte
5. A stop condition

The bus is considered idle when both SCLK and SDATA are HIGH. The bus's control begins with a start condition and ends with a stop condition, which only the controller device can generate.

Figure 5.2 shows how the bus behaves in a Read situation.

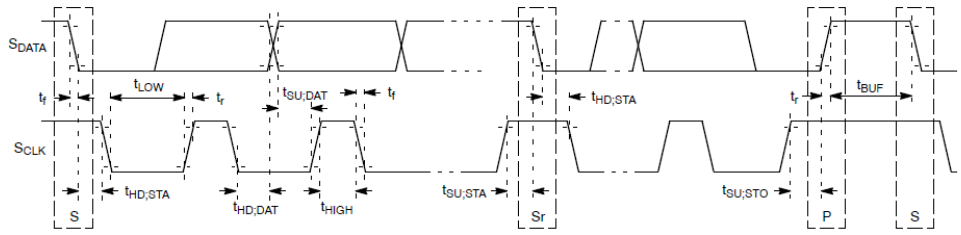


Figure 5.2: Two-Wire Serial Bus, extracted from [32]

Start Condition (S): A start condition is transitioning from HIGH to LOW on SDATA while SCLK is HIGH. At the end of a data transfer, the master can generate a start condition without previously generating a stop condition, known as a "repeated start" or "restart" condition.

Stop Condition (P): A stop condition is a transition from LOW to HIGH on SDATA while SCLK is HIGH.

Data Transfer: Data is transferred serially, 8 bits at a time, with the Most Significant Bit (MSB) transmitted first. Each data byte is followed by an acknowledge bit or a no-acknowledge bit. This data transfer mechanism is used for the slave address/-data direction byte and message bytes. One data bit is transferred during each SCLK

clock period. SDATA can change when SCLK is LOW and must be stable when SCLK is HIGH.

Slave Address/Data Direction Byte: Bits [7:1] of this byte represents the device slave address, and bit [0] indicates the data transfer direction. A "0" in bit [0] indicates a WRITE, and a "1" indicates a READ.

Acknowledge Bit: Each 8-bit data transfer is followed by an acknowledge or no-acknowledge bit in the SCLK clock period following the data transfer. The transmitter (the master when writing or the enslaved person when reading) releases SDATA. The receiver indicates an acknowledge bit by driving SDATA LOW. As with data transfers, SDATA can change when SCLK is LOW and must be stable when SCLK is HIGH.

No-Acknowledge Bit: The no-acknowledge bit is generated when the receiver does not drive SDATA LOW during the SCLK clock following a data transfer. It is used to terminate a read sequence.

This protocol was already available for kernel implementation by Oresat [28], and was adapted for the FPGA. We used two of the GPIOs and managed how and when they were at high or low levels. This protocol should be written in VHDL for the FPGA hardware platform for a better and more efficient implementation.

The Fig 5.3 shows the implemented protocol with yellow signal being SDA and blue being SCLK from a read instruction on register 0x3000.

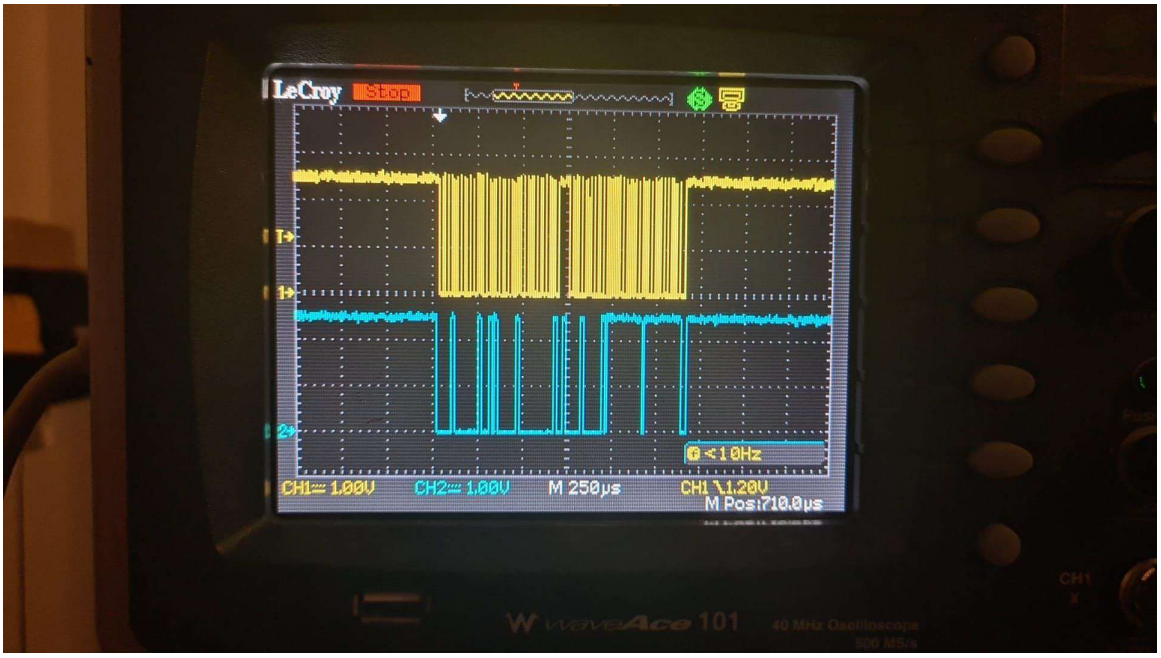


Figure 5.3: I2C on oscilloscope

5.1.2 Software Design

To test the image sensor it was also developed a bit of the software design to make the capture of a frame. The software design algorith implemented provides the following:

- Initialise the peripherals
- Start the IP cores
- Enables EXTCLK and initializes GPIOs needed for the sensor
- Configure the image sensor over I2C
- Capture a Frame

The pending software to be implemented in a later phase of the project involves configuring the HDMI device, setting up and accessing the DDR3 memory, enabling the ability to switch between different capture modes (such as Trigger mode, Stream mode, and HiSPi mode), and making the algorithm available in the kernel system for other services to analyze the captured images.

6

Testing and Results

After designing the prototypes and configuring the FPGA, the next step was to conduct a comprehensive system test.

The assembled PCBs are depicted in Figure 6.1, Figures 6.2, and Figure 6.3.

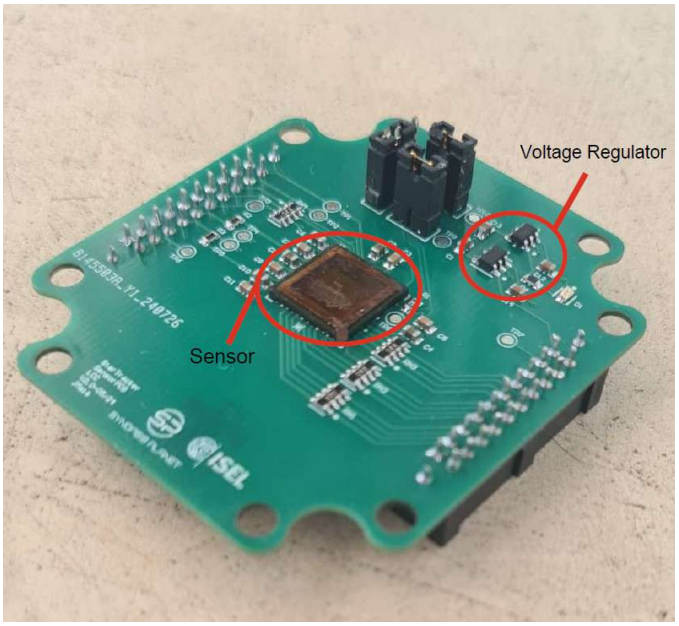


Figure 6.1: Photograph of the Image Sensor(LCC) Assembled with parallel interface

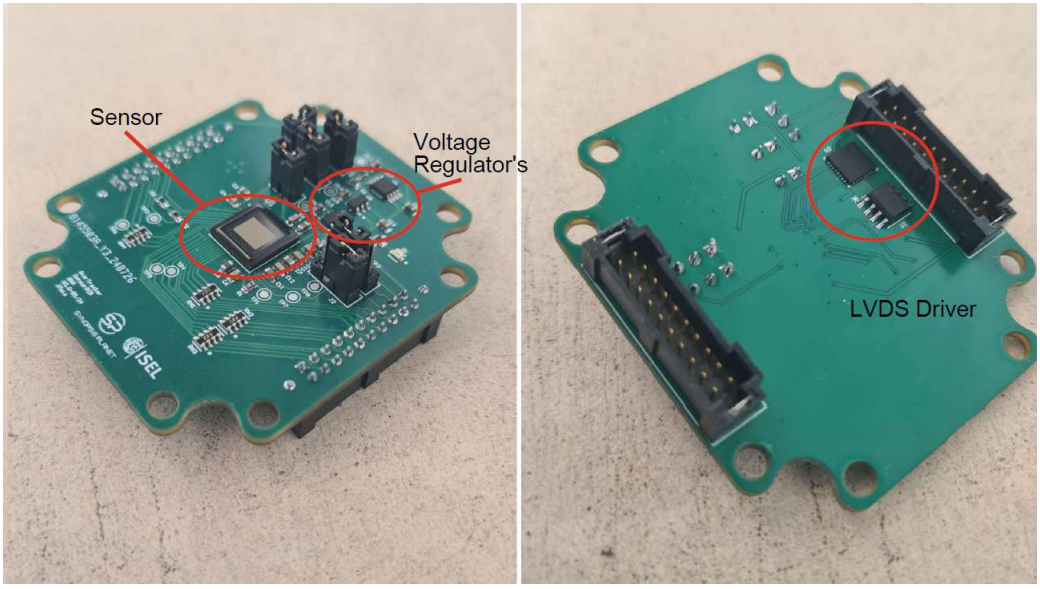


Figure 6.2: Photograph of the Image Sensor(BGA) Assembled, image sensor and supply regulators o top layer (left) and LVDS drivers on bottom layer (right)

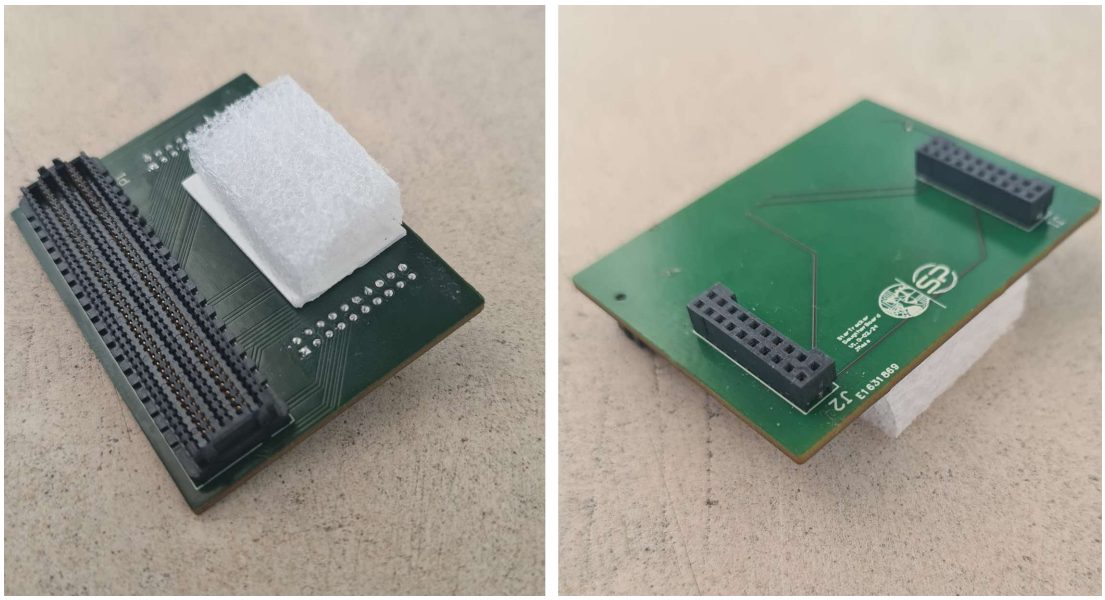


Figure 6.3: Photograph of the Daughter Board Assembled, (left) bottom layer with FFC connector and (right) top layer with image sensor connectors

Notably, the PCB with the BGA package accommodates components on both sides. A foam cube was affixed to the FMC side of the Daughter Board to facilitate pressure application on top of the prototype without disconnecting the sensor from the FPGA.

Both sensor prototypes were equipped with test points to check power levels and vertical connectors between the different regulators and the sensor to facilitated

current and voltage measurements, minimizing the risk of sensor damage during testing.

To debug all the assembled PCBs, the following tests were performed:

- **Short Circuit Testing:** Ensured no short circuits existed between traces, especially in power traces. (Multimeter)
- **Voltage Verification:** Powered on the board and confirmed that the regulators generated the correct voltages. Additionally, verified that pull-ups were at the correct levels. (Multimeter at test points
- **Current Measurement:** Checked that the current consumption stayed within expected limits. (Multimeter at jumpers)
- **Clock Generation:** Validated the external clock and pixel clock generation (Oscilloscope)
- **Sensor Configuration:** Set the frame rate to 10 frames per second using the Two-Wire Interface to configure the sensor. (Oscilloscope)

Troubleshooting involved using a multimeter and oscilloscope to verify correct voltages and currents before connecting the jumpers between the regulators and the sensor. Figure 6.4 illustrates the most relevant testing points.

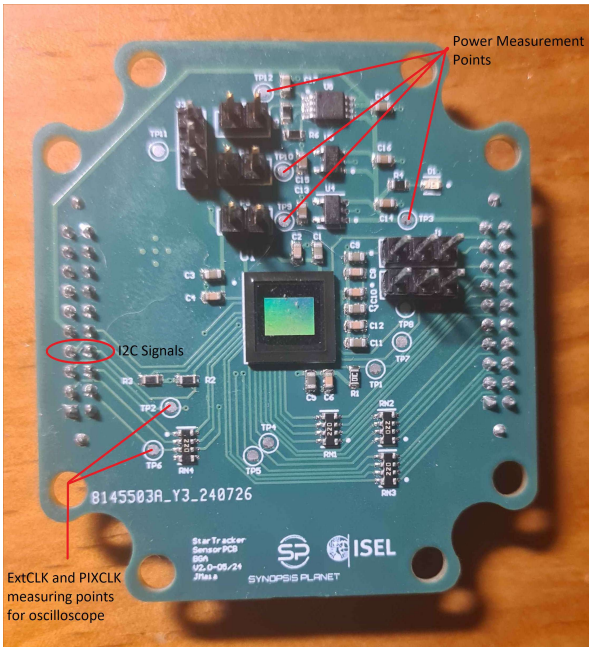


Figure 6.4: Image Sensor Testing Points

6.1 Testing the Parallel Output

It was chosen to test the parallel output as more information on how this one worked was available. The diagram in Figure 6.5 illustrates the necessary components for the parallel interface:

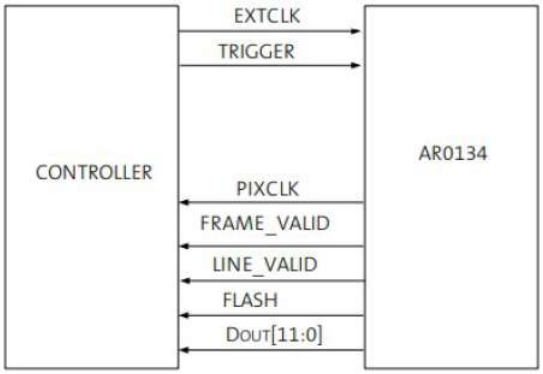


Figure 6.5: Block diagram for parallel interface, extracted from [32]

A pulse on the Trigger signal indicates when the sensor should capture the image. The FRAME_VALID signal corresponds to the time when the capture occurs, while LINE_VALID represents the capture completion for one of the frame rows. The flash turns on during the capture exposure. Figure 6.6 provides the timing for the parallel output capture.

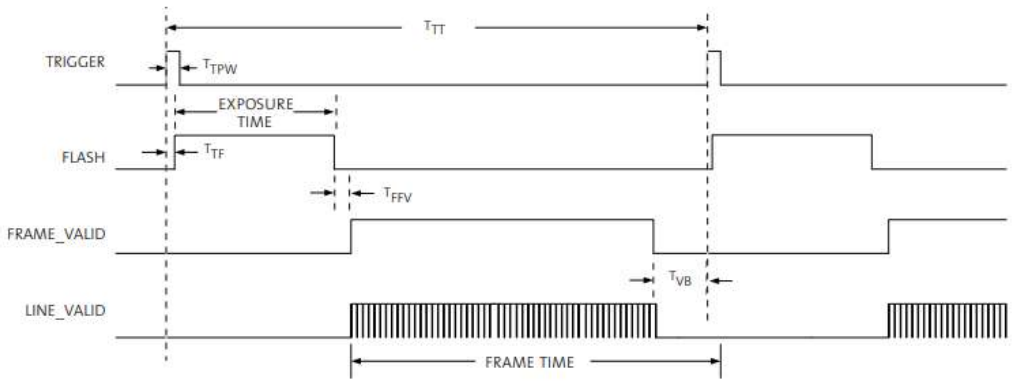


Figure 6.6: Trigger Mode, extracted from [32]

In order to test the image sensor parallel data, it was utilized the available configuration for the image sensor on the Oresat Github repository ([33]) for their prototype. This configuration is crucial for setting up the camera mode and the Pixel clock to ensure that everything is functioning correctly.

Understanding how to calculate the Pixel clock is essential. The PLL consists of a prescaler to divide the input clock applied on EXTCLK, a VCO to multiply the prescaler output, and two divider stages to generate the output clock. The clocking structure is illustrated in Figure 6.7. The PLL control registers can be programmed to generate frequencies other than the default power-up state.

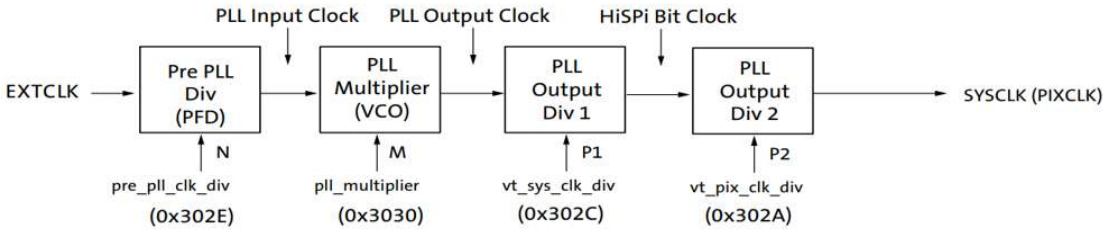


Figure 6.7: PLL Block Diagram, extracted from [32]

Using the formula $fPIXCLK = (fEXTCLKM)/(NP1xP2)$ the pixel clock can be calculated. The M represents the PLL multiplier, N is the pre_pll_clk_div, $P1$ stands for vt_sys_clk_div, and $P2$ corresponds to vt_pix_clk_div. In the current test, with an EXTCLK of 27MHz (Figure 6.8), the PLL values used were $M = 150$, $N = 10$, $P1 = 7$, and $P2 = 3$, resulting in a PIXCLK of 19.28MHz (Figure 6.9).

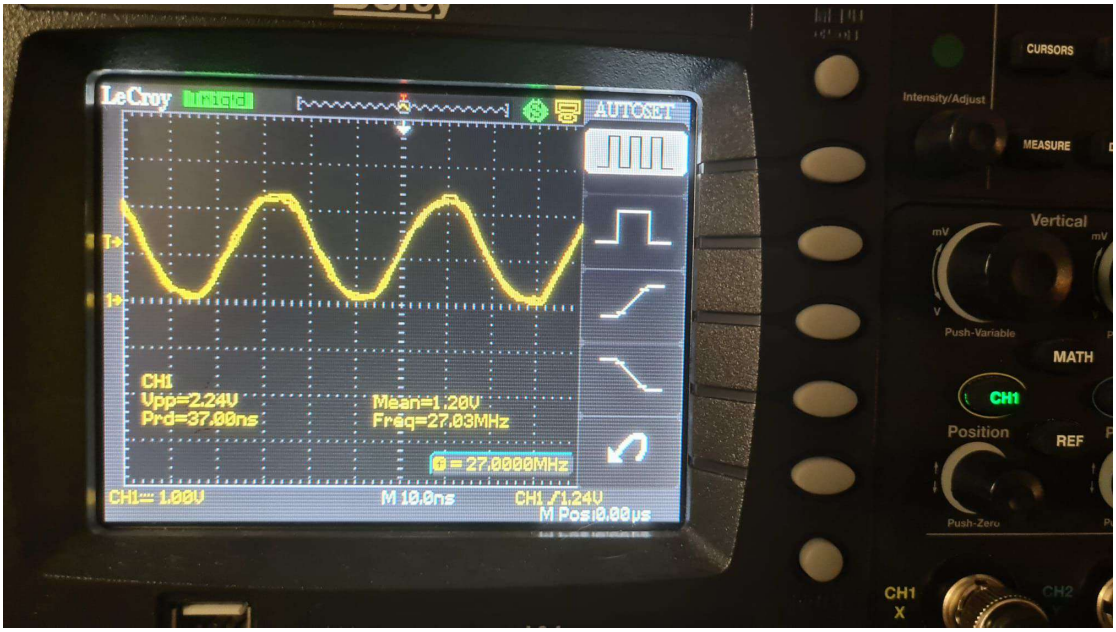


Figure 6.8: External Clock Oscilloscope Snapshot

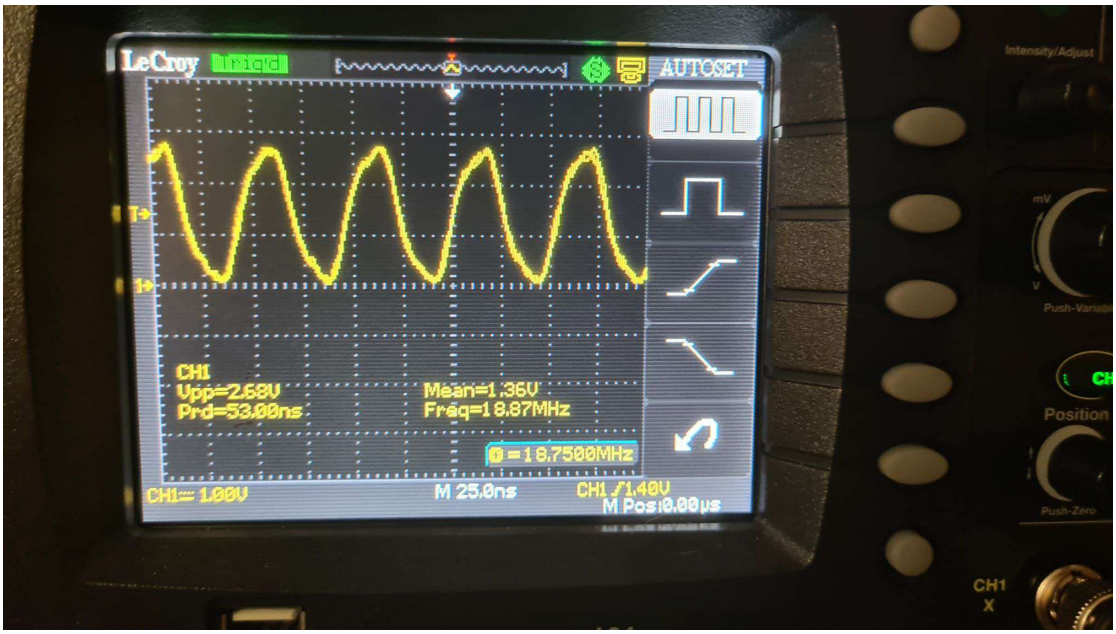


Figure 6.9: Pixel Clock Oscilloscope Snapshot

To verify the capture's functionality, we calculated the frame time and compared it with the oscilloscope results. To conduct this test, we had to configure additional registers for the full-resolution test. The Table 6.1 outlines the image condition we aimed to configure, the specific setting we adjusted, and the corresponding values we input:

Table 6.1: Image Sensor Condition Settings

Image Sensor Condition	Setting Description	Register [Bits] = Value
Exposure mode	Trigger	R0x301A[2] = 0 R0x301A[8] = 1 R0x301A[11] = 1
PIXCLK frequency	74.25 MHz	N/A
Window height	960	R0x3008 = 959 R0x3004 = 0
Window width	1280	R0x3008 = 1279 R0x3004 = 0
Line length (pck)	1650	R0x300C = 1650
Frame length (lines)	990	R0x300A = 990
Integration time	100	R0x3012 = 100

The following formulas were available in the AR0134 Developer Guide [28], and the result shows the frame time and frequency for this test:

$$\text{row\_time} = \frac{\text{line\_length\_pck}}{\text{PIXCLK\_freq}} = \frac{1650}{19.28\,\text{MHz}} = 85.58\,\mu s \tag{6.1}$$

$$\text{rows\_per\_frame} = \text{frame\_length\_lines} = 990\,\text{rows} \tag{6.2}$$

$$\text{frame\_time} = (\text{rows\_per\_frame}) \times (\text{row\_time}) = 84.7\,\text{ms} \tag{6.3}$$

$$\text{exposure\_time} = \text{integration\_time} \times \text{row\_time} = 8.558\,\text{ms} \tag{6.4}$$

$$T_{TF} + T_{FFV} = (8.21\,\text{rows} + 18.21\,\text{rows}) \times \text{row\_time} = 2.26\,\text{ms} \tag{6.5}$$

$$\text{frame\_rate} = \frac{1}{\text{frame\_time} + \text{exposure\_time} + T_{TF} + T_{FFV}} = 10.469\,\text{Hz} \approx 10\,\text{fps} \tag{6.6}$$

With these calculations and using the oscilloscope, Figure 6.10 verifies the frame rate:

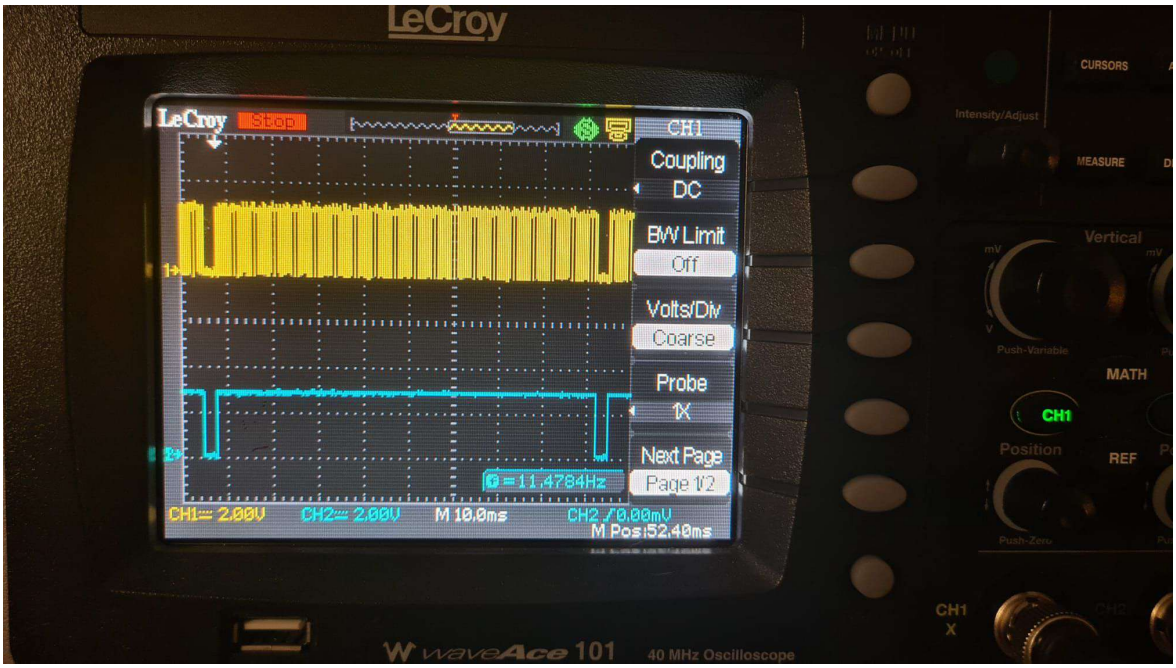


Figure 6.10: Frame_Valid(blue trace) and Line_Valid(yellow trace) Oscilloscope Snapshot

the blue trace corresponds to FRAME_VALID, while the yellow trace represents LINE_VALID. This allows us to confirm that the board and the sensor's clock circuitry are functioning as expected.

6.2 FPGA Board

Although the FPGA board was not produced, it is important to recognize the significant accomplishment of finalizing its design. The intricate routing still requires refinement and optimization to meet the signal integrity specifications for each serial signal. Nonetheless, this design effectively demonstrates that all components can be successfully integrated onto a single board.

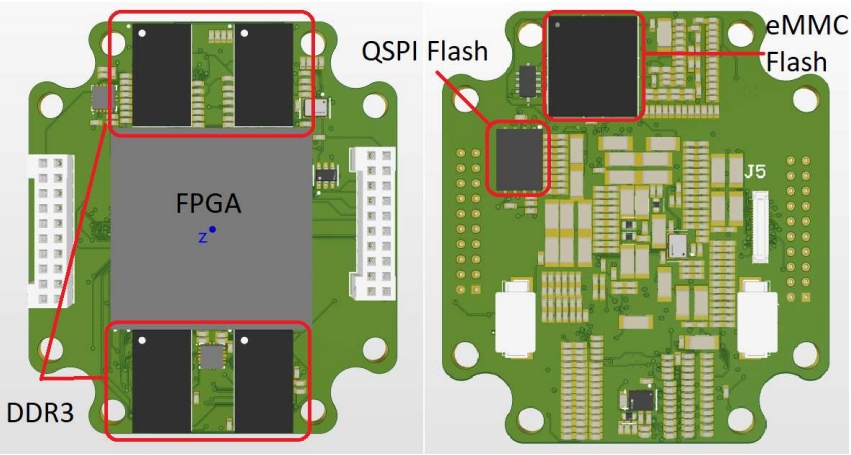


Figure 6.11: 3D image of FPGA PCB

7

Conclusions

In the final analysis, the "Star Tracker Platform" project successfully demonstrated the feasibility and initial implementation of critical hardware components within specified design constraints.

The CMOS image sensor AR0134CS PCB was developed for both the BGA and LCC packages, supporting both the High-Speed Serial Interface (HiSPi) and a 12-bit parallel interface.

For the BGA package, the HiSPi interface underwent electromagnetic compatibility simulation to evaluate signal integrity for differential pairs. This board can accommodate power supply mechanisms to convert SLVS signals into LVDS and implement differential transmission lines tailored for a 100-ohm load, with prospects for further miniaturising the board's dimensions. Although prioritising maximum-speed image transmission is not essential for the application in question, simulation results has been presented that supports the implementation of both Parallel and Serial interfaces, facilitating transmission rates of up to 700 MHz.

The sensor board was manufactured and assembled, and its functionality was verified using an FPGA development kit. Configuration of the sensor achieved theoretical calculations of 10 frames per second, which were experimentally validated with the configured system. This verification included snap-shots of clock signals and two-wire communication, confirming successful initialization and operation of the sensor.

Concerning the FPGA module, it has been established that fitting all requisite components within the same mechanical constraints is achievable. However, the

layout's complexity is significantly high, a minimum of 14 layer PCB is required, indicating that a strategy involving either the division of the design across two boards or an enlargement of the board dimensions might be more pragmatic.

Despite the hardware not being fully configured for data collection and display functionalities, preliminary power consumption estimates have been acquired. The FPGA board exhibits a consumption of approximately 2 Watts, whereas the sensor, when operating at its maximum resolution and streaming at 54 frames per second, demonstrates a consumption of around 0.46 Watts. This results in an aggregate power consumption of 2.5 watts, which, despite surpassing the energy consumption of specific star trackers, is justified by employing a high-end FPGA engineered for space radiation resilience, thus offering superior durability relative to lower-power microcontrollers.

Looking forward, several tasks remain for future work. Foremost among these is the completion of the hardware platform on Vivado, encompassing the finalisation of the FPGA board layout and the system configuration for image acquisition from the sensor. After image acquisition, the display must be facilitated via an HDMI connection. This attempt will demand not only the accomplishment of the hardware design but also the formulation of requisite software for managing image acquisition, processing, and output.

An additional critical task is realising the PCB layout for the FPGA board. Although the schematic has reached completion, the tangible PCB design and layout analysis in an eletromagnetic simulator for the signal integrity of DDR3 critical signals.

Lastly, the sensor, currently operational through the parallel interface, needs to be reconfigured to employ the High-Speed Serial Interface (HiSPI). This transition will augment system performance and data throughput, necessitating firmware updates and ensuring compatibility with the extant hardware components.

References

[1] Carl Christian Liebe Allan Read Eisenman and John Leif Joergensen, "New generation of autonomous star trackers", *SPIE*, 1997.

[2] Willem Steyn Vaios Lappas and Craig Underwood, "Practical results on the development of a control moment gyro based attitude control system for agile small satellites", 2002.

[3] "The cubesat program". (2014), [Online]. Available: http://www.cubesat.org/.

[4] "Nanosatellite and cubesat database". (2018), [Online]. Available: http://nanosats.eu/.

[5] Kollsman Instrument Corp, "Space star trackers", NASA Electronics Research Center, 1970.

[6] Puneet Singla, John Crassidis, and John Junkins, "Spacecraft angular rate estimation algorithms for star tracker-based attitude determination", *Advances in the Astronautical Sciences*, vol. 114, Jan. 2003.

[7] T. Schwar, "Prototyping of a star tracker for pico-satellites", Lulea University of Technology, 2015.

[8] M. Lind Marcus, "Development and implementation of star tracker electronics", KTH Royal Institute of Technology, 2014.

[9] "Fast, affordable, science and technology satellite (fastsat) minisatellite". (2010), [Online]. Available: http://www.nasa.gov/centers/marshall/pdf/709025main_FASTSAT_Facts_11_2012.pdf.

[10] D. Sinclair J. Enright and K. C. Fernando, "Cots detectors for nanosatellite star trackers", 2011.

[11] G. Stafford S. Palo and A. Hoskins, "Ssc13-iii-5 an agile multi-use nano star camera for constellation applications", 2013.

[12] "St-200 autonomous star tracker flyer". (2013), [Online]. Available: http://www.berlin-space-tech.com/fileadmin/media/BST_ST-200_Flyer.pdf.

[13] Stephan Theil Malak, "Development of a low cost star tracker for the shefex mission", in *Aerosp. Sci. Technol. 23*. 2012.

[14] A. Kwiatkowski Steinbach Pradarutti Michel Benzi U. Schmidt Fiksel, "Autonomous star sensor astro aps: Flight experience on alphasat", *CEAS SpaceJournal 7*,

[15] Desheng Wen Shaodong Yang Meiying Hu Wang, "A novel design of subminiature star sensor's imaging system based on tms320dm3730", *Proc. of SPIE, Vol.10256 1025635 -1*,

[16] Willem H. Steyn Alexander O. Erlank, "Arcminute attitude estimation for cubesats with a novel nano star tracker".

[17] Gangyi Wang Zunzong Ye Xinguo Wei, "Esign of star sensor imaging driver module based on imx224, advances in materials".

[18] Alexander O ERLANK, "Development of cubestar a cubesat-compatible star tracker", Stellenbosch University, 2013.

[19] Jürgen Laks, "Developing hardware for the estcube-2 star tracker", Faculty of Science and Technology, 2019.

[20] Mojtaba Abolghasemi Heshmatollah Hoseini Mehdi Nasiri Sarvi Dariush Abbasi-Moghadamb, "Design and implementation of a star-tracker for leo satellite", *Optik - International Journal for Light and Electron Optics*, 2020.

[21] K.C. Fernando, "Hot pixel detection for star trackers", Ryerson. University, 2009.

[22] Glenn Lebrasseur Emma Levy Ryan Medick Hayden Reinhold Risto Rushford Andrew Greenberg David Lay and Miles Simpson, "Oresat: A student team-based approach to an inexpensive, open, and modular (1-3u) cubesat bus",

[23] Jingxin; YE Tao; CHEN Lipeng ZHOU Fuqiang; ZHAO, "Fast star centroid extraction algorithm with sub-pixel accuracy based on fpga", *Journal of Real-Time Image Processing*,

[24] Alireza; GHAZNAVIGHOUSHCHI M. B. VLSI AZIZABADI Mohsen; BEHRAD, "Implementation of star detection and centroid calculation algorithms for star tracking applications", *Journal of Real-Time Image Processing*,

[25] Gabriel Mariano Marcelino, "Desenvolvimento e implementação de um algoritmo para extração de centroides em star trackers para cubesats utilizando fpgas", UNIVERSIDADE FEDERAL DE SANTA CATARINA, 2019.

[26] Christiaan Lodewyk von Wielligh, "Fast star tracker hardware implementation and algorithm optimisations on a system-on-a-chip device", Faculty of Engineering at Stellenbosch University, 2019.

[27] ONSEMI, "Ar0134cs",

[28] XILINX, "Zynq-7000 soc data sheet: Overview",

[29] Carl Christian Liebe John Leif Joergensen Allan Read Eisenman, "New generation of autonomous star trackers", 1997.

[30] D. Michaels and J. Speed, "Ball aerospace star tracker achieves high tracking accuracy for a moving star field".

[31] Willem H. Steyn Alexander O. Erlank, "Arcminute attitude estimation for cubesats with a novel nano star tracker".

[32] Aptina, "Ar0134 developer guide",

[33] OreSat, "Oresat open source cubesat project", [Online]. Available: https://github.com/oresat.

[34] Xilinx, "Recommended design rules and strategies for bga devices user guide",

[35] Xilinx, "Zynq-7000 soc data sheet: Overview",

[36] Xilinx, "Zynq-7000 soc pcb design guide",

[37] Alpatina, "High-speed serial pixel (hispi)™ interface protocol",

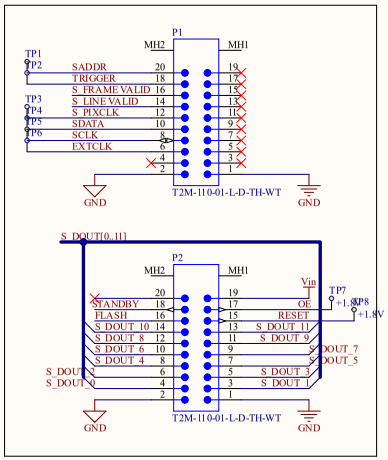
[38] Alinx, "Zynq development platform basic tutorial ax7350/ax7350b",

[39] OreSat, "Oresat satalites", [Online]. Available: https://www.oresat.org/satellites.

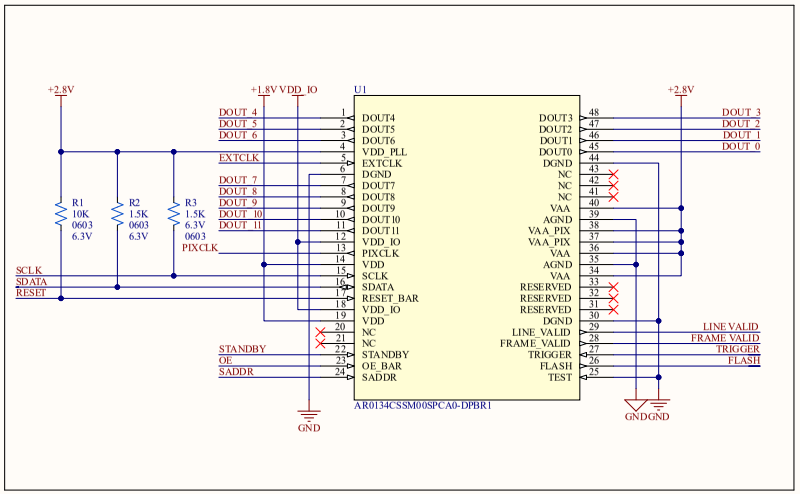
[40] Aptina, "High-speed serial pixel (hispi) interface physical layer specification v2.00.00",

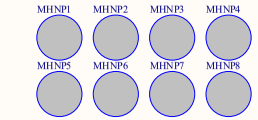
Schematic Sensor LCC

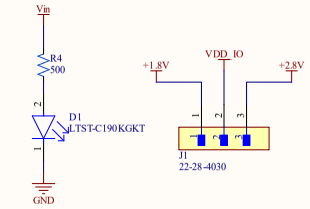
Connectors

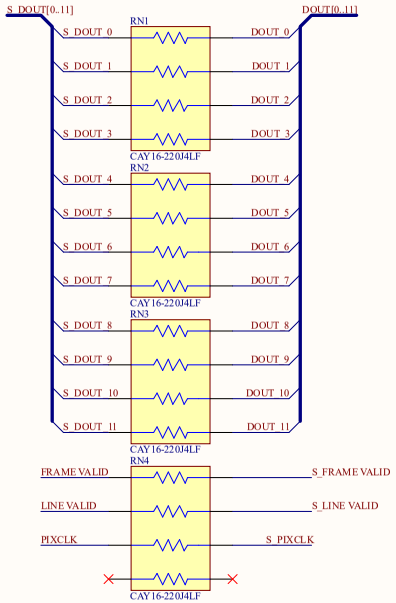


SENSOR AR0134CS

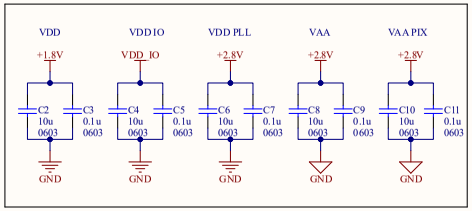




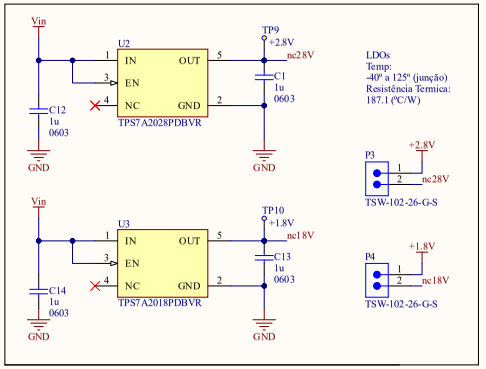




ByPass Capacitors



Reguladores LDO (+2.8V,+1.8V)

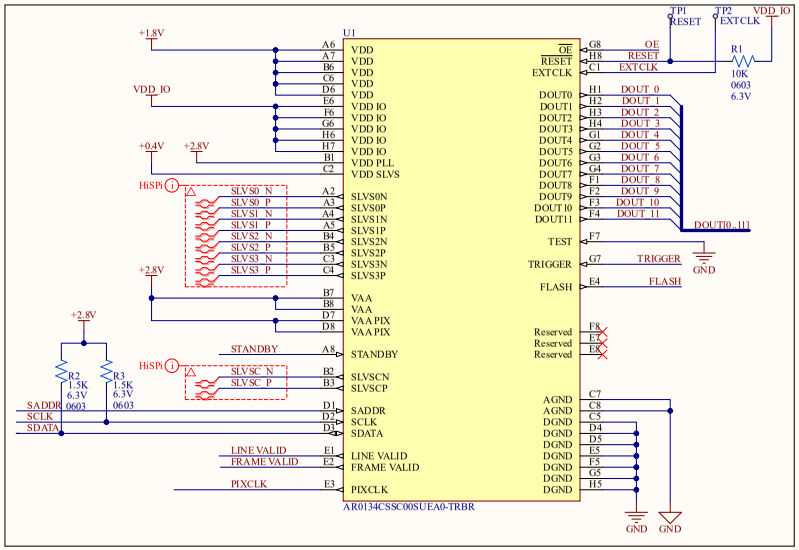


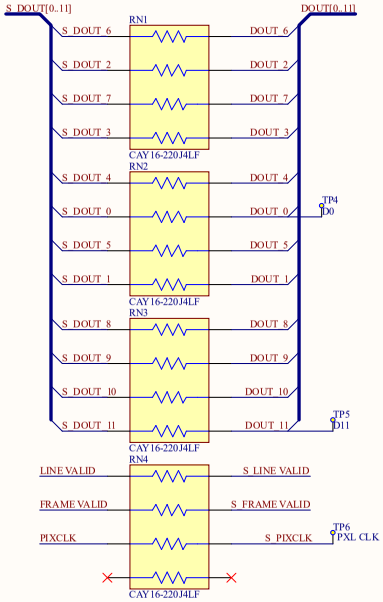
Title		
Size: A3	Number	Revision
Date: 11/05/2024		Sheet of
File: C:\Users\...\Sensor.SchDoc		Drawn By:

B

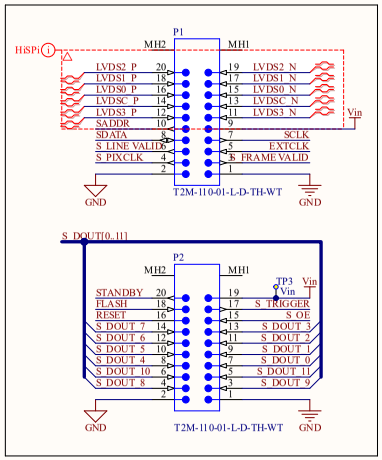
Schematic Sensor BGA

SENSOR AR0134CS

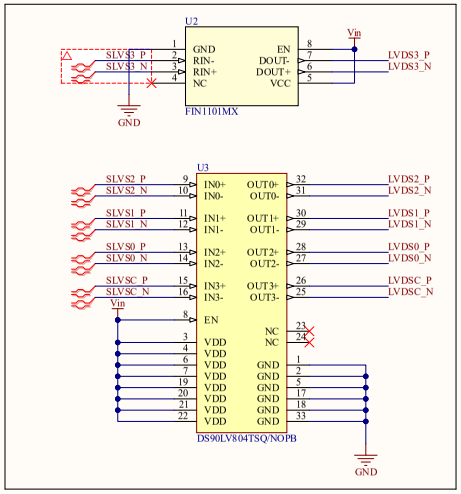




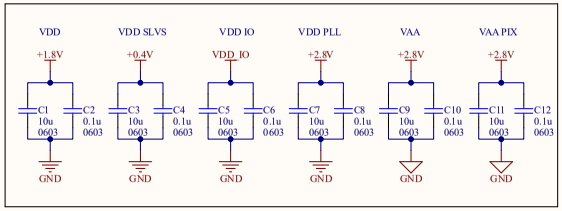
Connectors



HiSpi-LVDS Driver

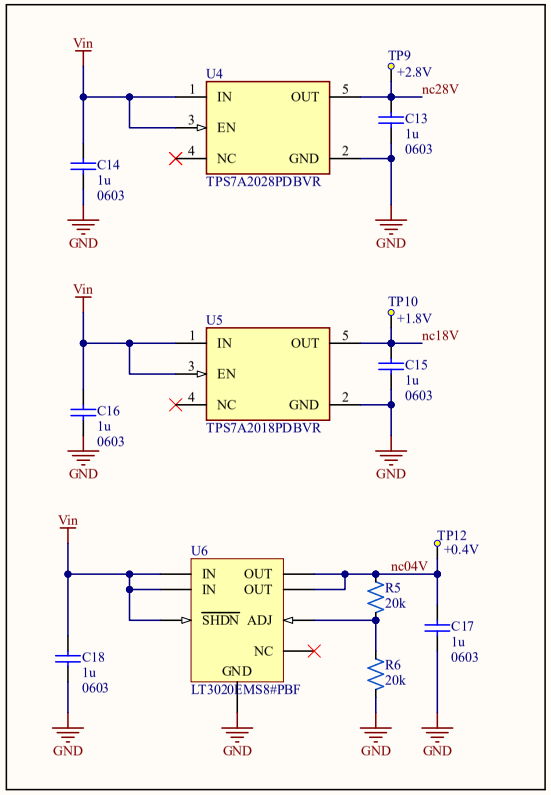


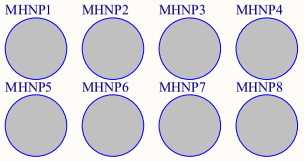
ByPass Capacitors

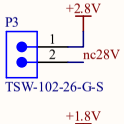


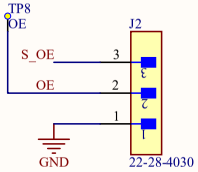
Title			
Size A3	Number		Revision
Date:	11/05/2024	Sheet of	
File:	C:\Users\...\Sensor.SchDoc	Drawn By:	

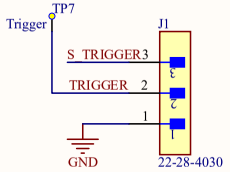
Regulators LDO (+2.8V,+1.8V e +0.4V)

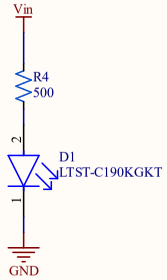








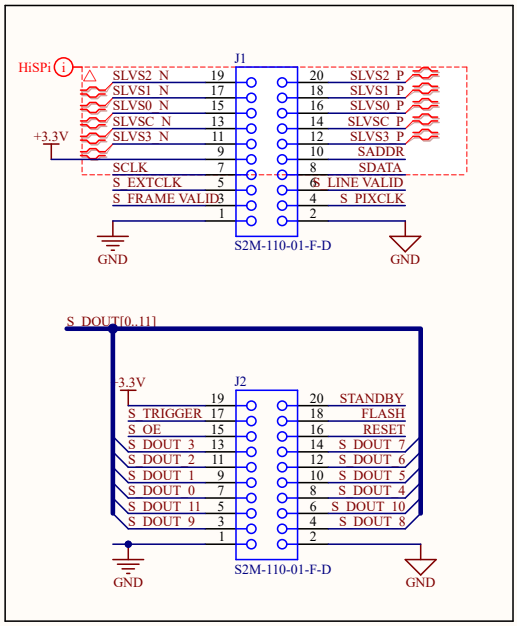


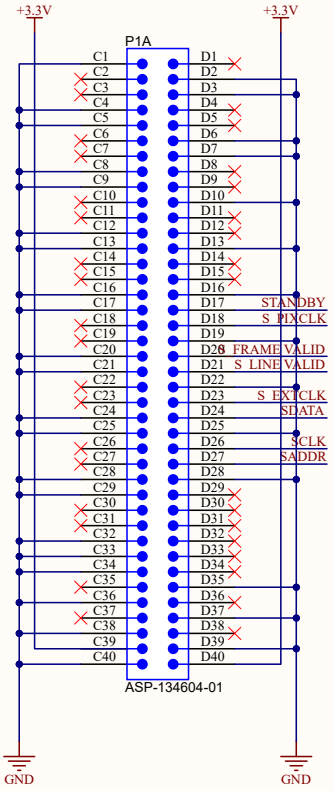


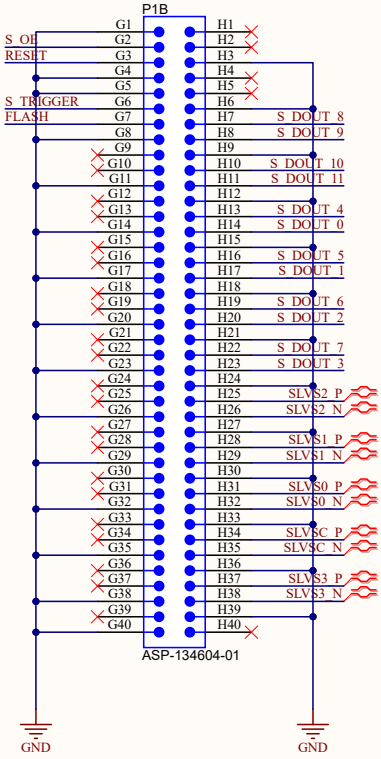
Title		
Size: A4	Number	Revision
Date: 11/05/2024		Sheet of
File: C:\Users\...\Sensor2.SchDoc		Drawn By:

Schematic Connector

Connectors





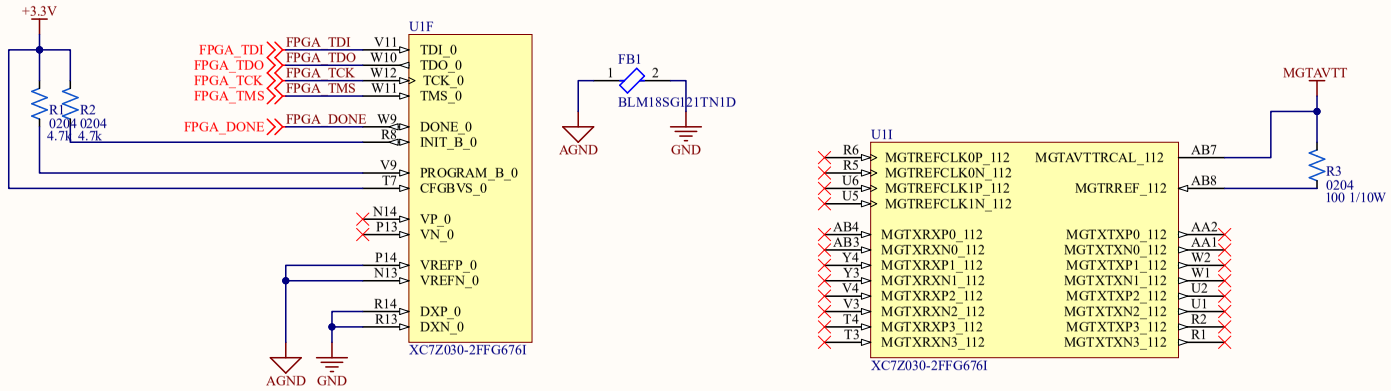


Title		
Size A4	Number	Revision
Date: 11/05/2024		Sheet of
File: E:\AltiumProjects\..\Connector.SchDoc		Drawn By:

D

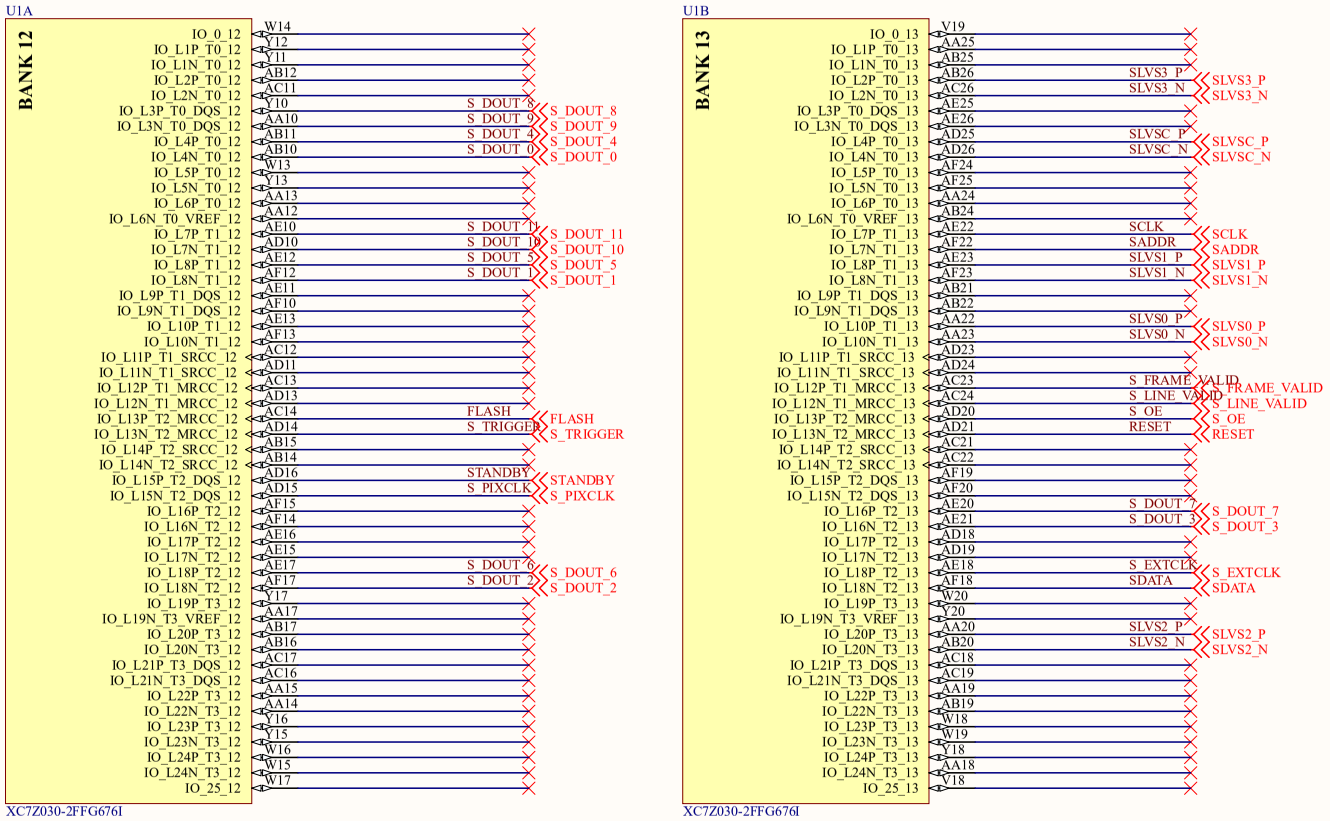
Schematic Sensor FPGA

[1] BANK 0



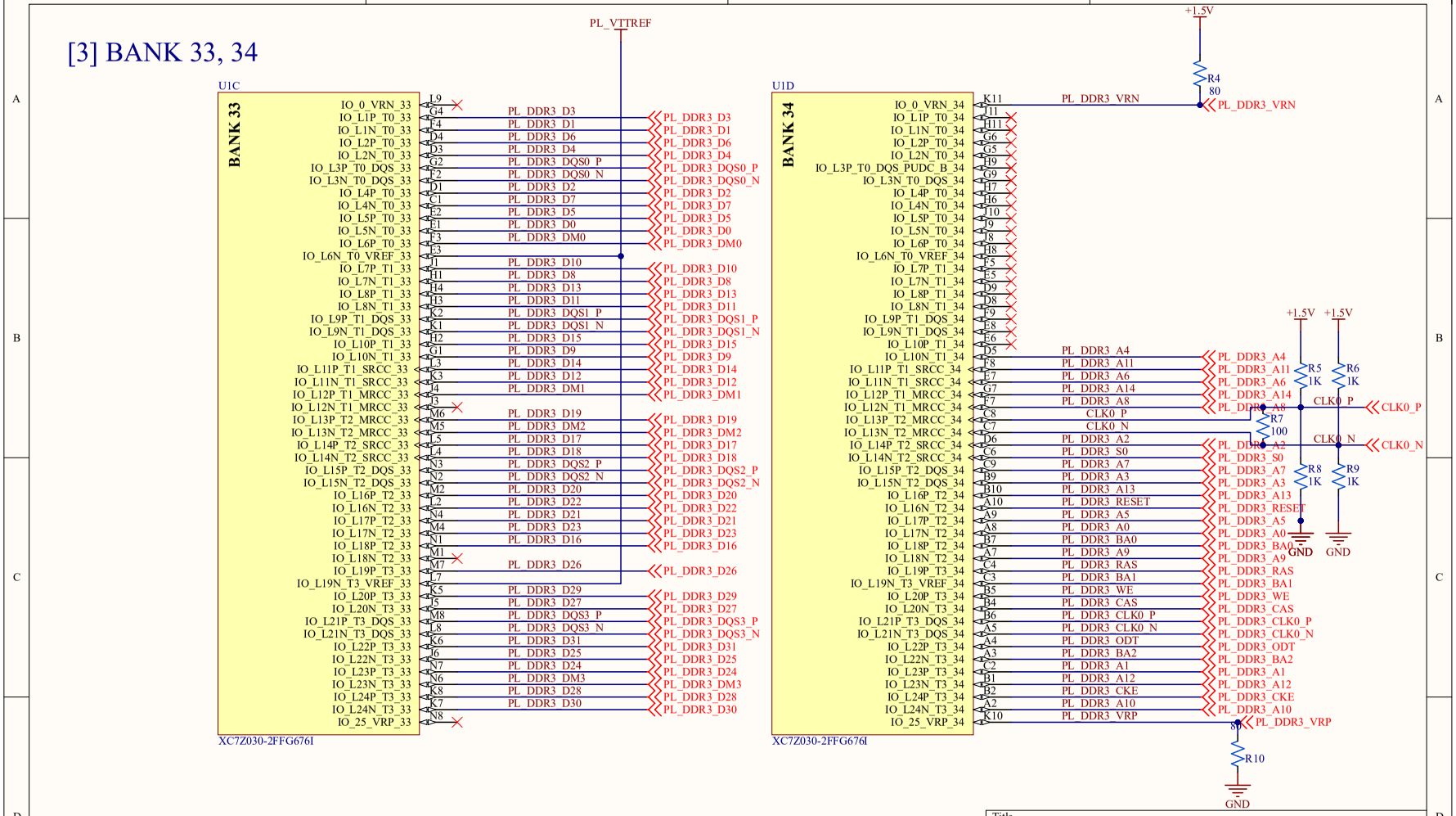
Title			
Size: A4	Number		Revision
Date: 11/07/2024		Sheet of	
File: E:\AltiumProjects\..\Bank0.SchDoc		Drawn By:	

[2] BANK 12, 13

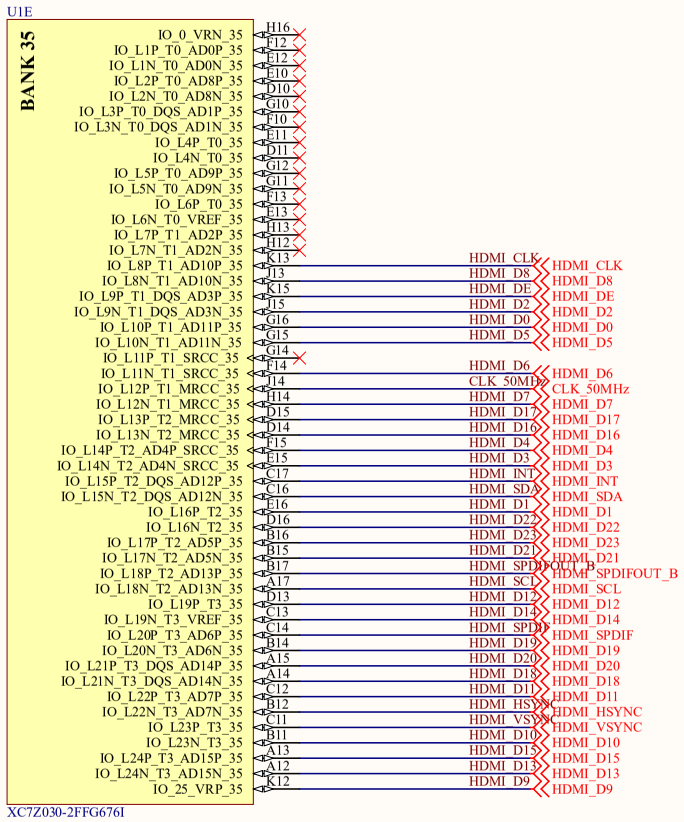


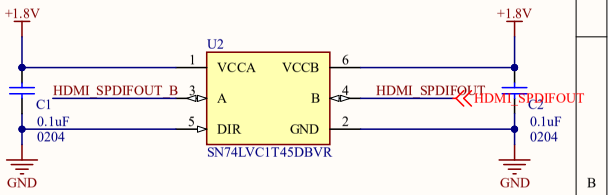
Title		
Size: A4	Number	Revision
Date: 11/07/2024		Sheet of
File: E:\AltiumProjects\..\Bank12_13.SchDoc		Drawn By:

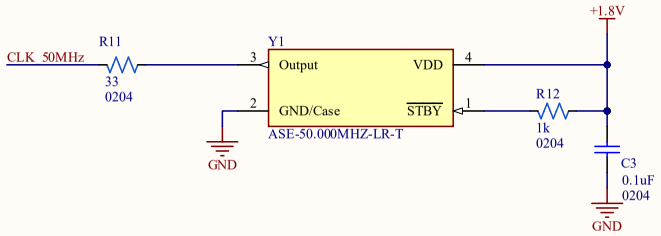
[3] BANK 33, 34


PL_VTTREF
U1C
BANK 33
IO_0_VRN_33
IO_L1P_T0_33
IO_L1N_T0_33
IO_L2P_T0_33
IO_L2N_T0_33
IO_L3P_T0_DQS_33
IO_L3N_T0_DQS_33
IO_L4P_T0_33
IO_L4N_T0_33
IO_L5P_T0_33
IO_L5N_T0_33
IO_L6P_T0_33
IO_L6N_T0_VREF_33
IO_L7P_T1_33
IO_L7N_T1_33
IO_L8P_T1_33
IO_L8N_T1_33
IO_L9P_T1_DQS_33
IO_L9N_T1_DQS_33
IO_L10P_T1_33
IO_L10N_T1_33
IO_L11P_T1_SRCC_33
IO_L11N_T1_SRCC_33
IO_L12P_T1_MRCC_33
IO_L12N_T1_MRCC_33
IO_L13P_T2_MRCC_33
IO_L13N_T2_MRCC_33
IO_L14P_T2_SRCC_33
IO_L14N_T2_SRCC_33
IO_L15P_T2_DQS_33
IO_L15N_T2_DQS_33
IO_L16P_T2_33
IO_L16N_T2_33
IO_L17P_T2_33
IO_L17N_T2_33
IO_L18P_T2_33
IO_L18N_T2_33
IO_L19P_T3_33
IO_L19N_T3_VREF_33
IO_L20P_T3_33
IO_L20N_T3_33
IO_L21P_T3_DQS_33
IO_L21N_T3_DQS_33
IO_L22P_T3_33
IO_L22N_T3_33
IO_L23P_T3_33
IO_L23N_T3_33
IO_L24P_T3_33
IO_L24N_T3_33
IO_25_VRP_33
PL_DDR3_D3
PL_DDR3_D1
PL_DDR3_D6
PL_DDR3_D4
PL_DDR3_DQS0_P
PL_DDR3_DQS0_N
PL_DDR3_D2
PL_DDR3_D7
PL_DDR3_D5
PL_DDR3_D0
PL_DDR3_DM0
PL_DDR3_D10
PL_DDR3_D8
PL_DDR3_D13
PL_DDR3_D11
PL_DDR3_DQS1_P
PL_DDR3_DQS1_N
PL_DDR3_D15
PL_DDR3_D9
PL_DDR3_D14
PL_DDR3_D12
PL_DDR3_DM1
PL_DDR3_D19
PL_DDR3_DM2
PL_DDR3_D17
PL_DDR3_D18
PL_DDR3_DQS2_P
PL_DDR3_DQS2_N
PL_DDR3_D20
PL_DDR3_D22
PL_DDR3_D21
PL_DDR3_D23
PL_DDR3_D16
PL_DDR3_D26
PL_DDR3_D29
PL_DDR3_D27
PL_DDR3_DQS3_P
PL_DDR3_DQS3_N
PL_DDR3_D31
PL_DDR3_D25
PL_DDR3_D24
PL_DDR3_DM3
PL_DDR3_D28
PL_DDR3_D30
XC7Z030-2FFG676I
U1D
BANK 34
IO_0_VRN_34
IO_L1P_T0_34
IO_L1N_T0_34
IO_L2P_T0_34
IO_L2N_T0_34
IO_L3P_T0_DQS_PUDC_B_34
IO_L3N_T0_DQS_34
IO_L4P_T0_34
IO_L4N_T0_34
IO_L5P_T0_34
IO_L5N_T0_34
IO_L6P_T0_34
IO_L6N_T0_VREF_34
IO_L7P_T1_34
IO_L7N_T1_34
IO_L8P_T1_34
IO_L8N_T1_34
IO_L9P_T1_DQS_34
IO_L9N_T1_DQS_34
IO_L10P_T1_34
IO_L10N_T1_34
IO_L11P_T1_SRCC_34
IO_L11N_T1_SRCC_34
IO_L12P_T1_MRCC_34
IO_L12N_T1_MRCC_34
IO_L13P_T2_MRCC_34
IO_L13N_T2_MRCC_34
IO_L14P_T2_SRCC_34
IO_L14N_T2_SRCC_34
IO_L15P_T2_DQS_34
IO_L15N_T2_DQS_34
IO_L16P_T2_34
IO_L16N_T2_34
IO_L17P_T2_34
IO_L17N_T2_34
IO_L18P_T2_34
IO_L18N_T2_34
IO_L19P_T3_34
IO_L19N_T3_VREF_34
IO_L20P_T3_34
IO_L20N_T3_34
IO_L21P_T3_DQS_34
IO_L21N_T3_DQS_34
IO_L22P_T3_34
IO_L22N_T3_34
IO_L23P_T3_34
IO_L23N_T3_34
IO_L24P_T3_34
IO_L24N_T3_34
IO_25_VRP_34
PL_DDR3_VRN
PL_DDR3_A4
PL_DDR3_A11
PL_DDR3_A6
PL_DDR3_A14
PL_DDR3_A8
CLK0_P
CLK0_N
PL_DDR3_A2
PL_DDR3_S0
PL_DDR3_A7
PL_DDR3_A3
PL_DDR3_A13
PL_DDR3_RESET
PL_DDR3_A5
PL_DDR3_A0
PL_DDR3_BA0
PL_DDR3_A9
PL_DDR3_RAS
PL_DDR3_BA1
PL_DDR3_WE
PL_DDR3_CAS
PL_DDR3_CLK0_P
PL_DDR3_CLK0_N
PL_DDR3_ODT
PL_DDR3_BA2
PL_DDR3_A1
PL_DDR3_A12
PL_DDR3_CKE
PL_DDR3_A10
PL_DDR3_VRP
XC7Z030-2FFG676I
+1.5V
R4
80
R5
1K
R6
1K
R7
100
R8
1K
R9
1K
R10
GND
Title
Size
A4
Number
Revision
Date:
11/07/2024
Sheet of
File:
E:\AltiumProjects\...\Bank33_34.SchDoc
Drawn By:

[4] BANK 35

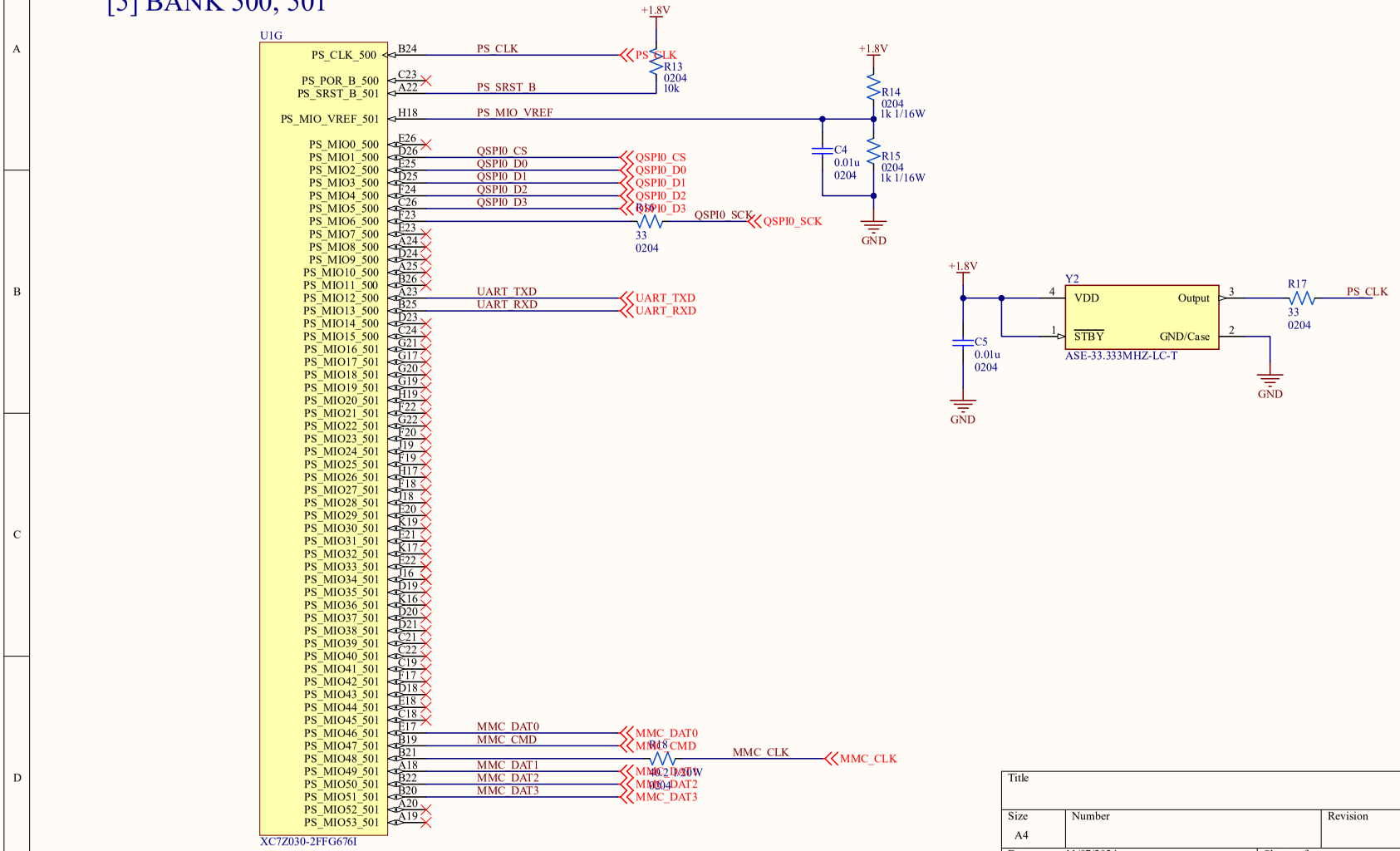




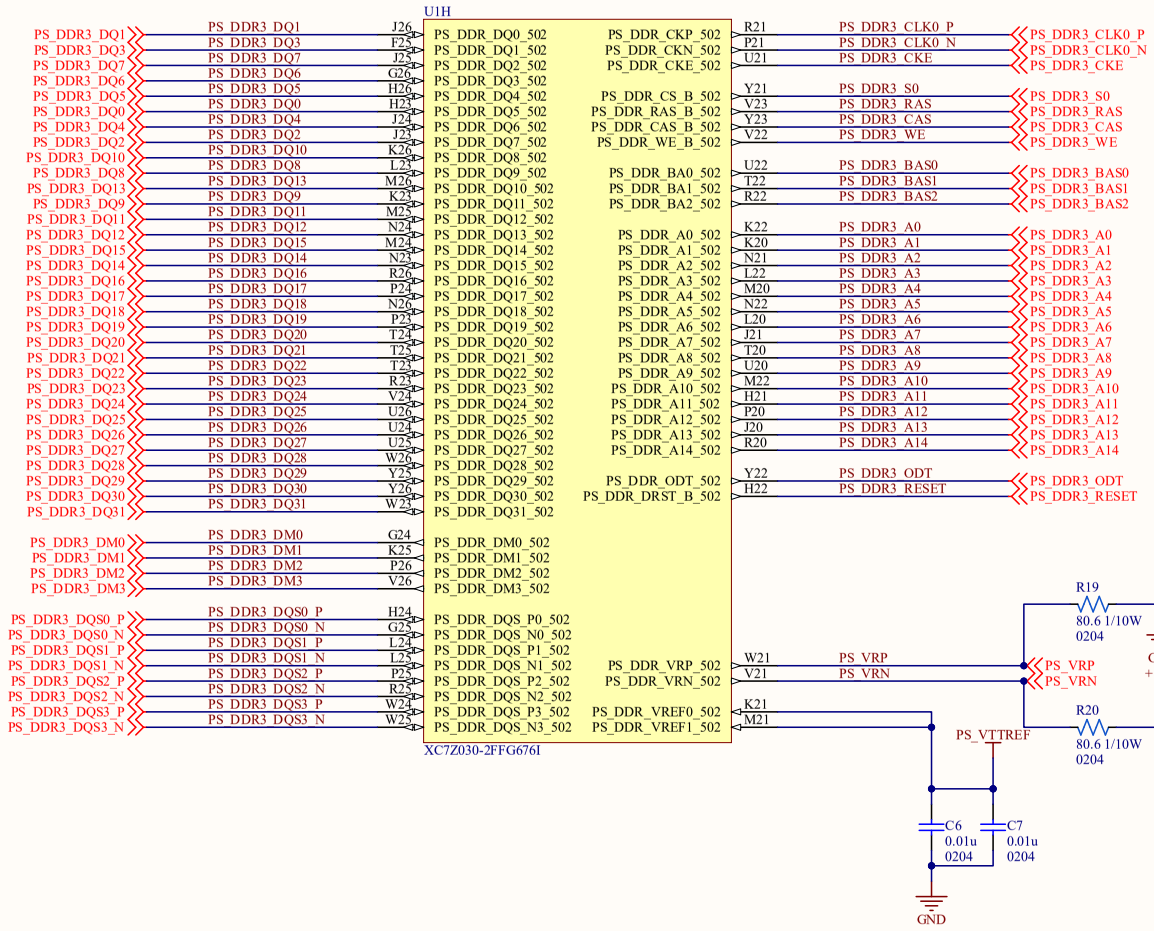


Title		
Size: A4	Number	Revision
Date: 11/07/2024		Sheet of
File: E:\AltiumProjects\..\Bank35.SchDoc		Drawn By:

[5] BANK 500, 501

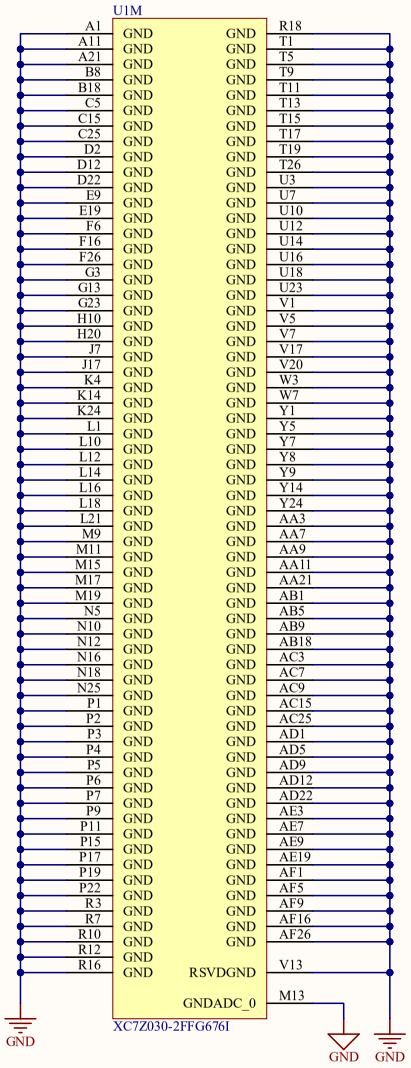


[6] BANK 502



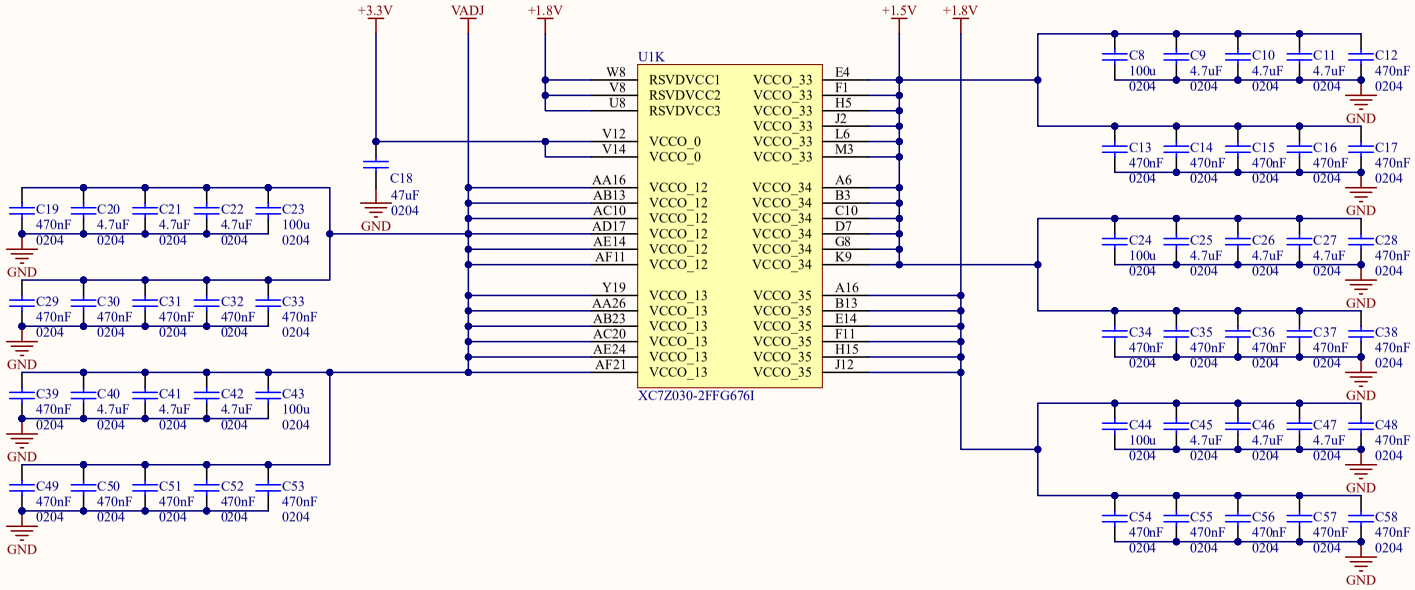
Title		
Size: A4	Number	Revision
Date: 11/07/2024		Sheet of
File: E:\AltiumProjects\..\Bank502.SchDoc		Drawn By:

[7] FPGA GND



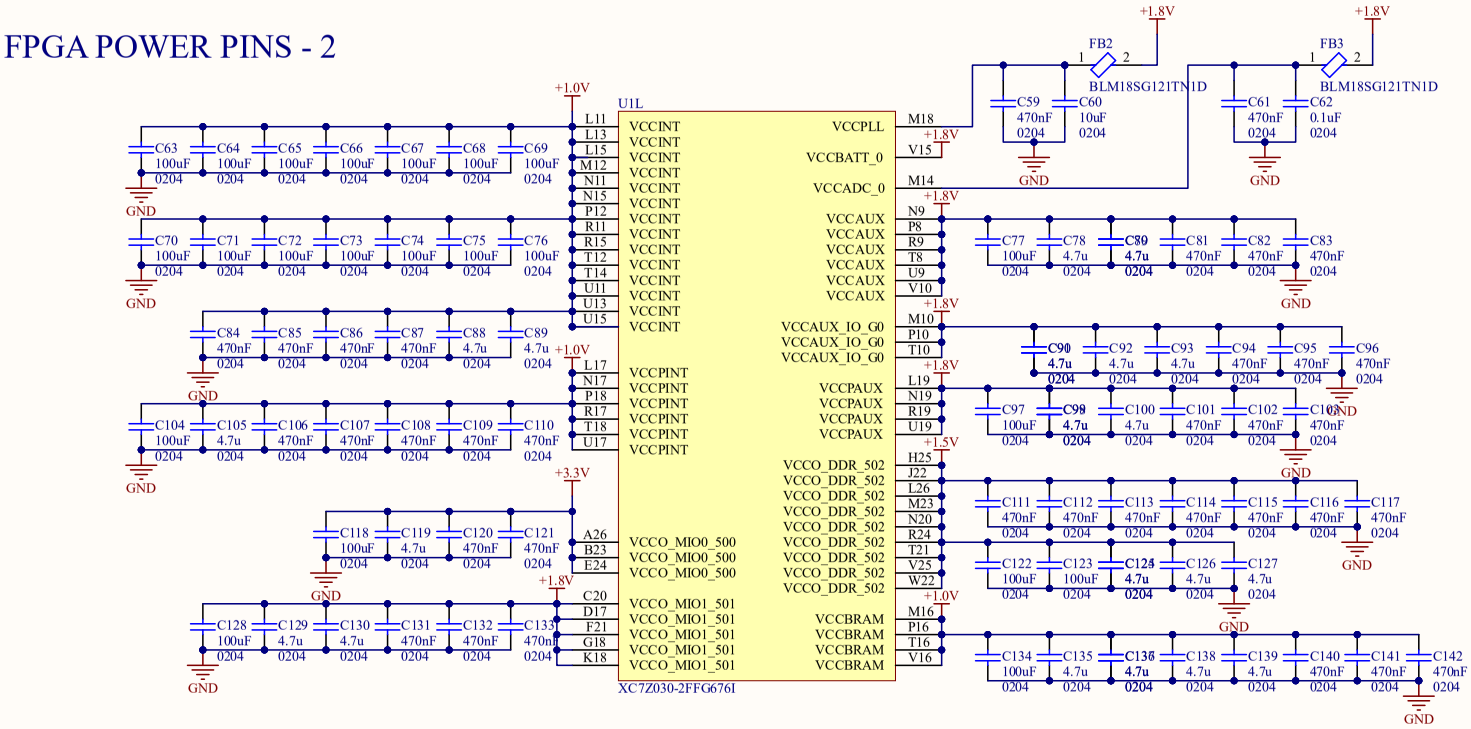
Title			
Size A4	Number		Revision
Date:	11/07/2024	Sheet of	
File:	E:\AltiumProjects\..\BankGND.SchDoc	Drawn By:	

[8] FPGA POWER PINS - 1



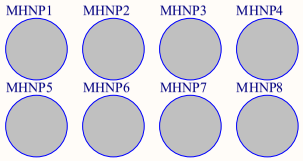
Title		
Size A4	Number	Revision
Date: 11/07/2024		Sheet of
File: E:\AltiumProjects\...\PowerPins0.SchDoc		Drawn By:

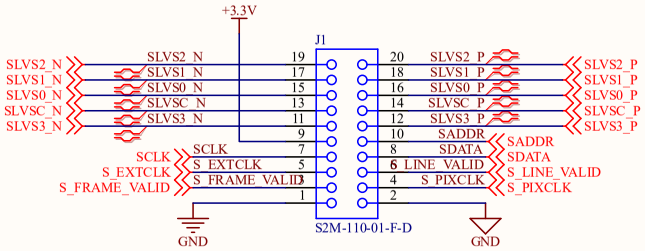
[9] FPGA POWER PINS - 2

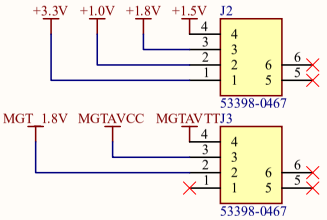


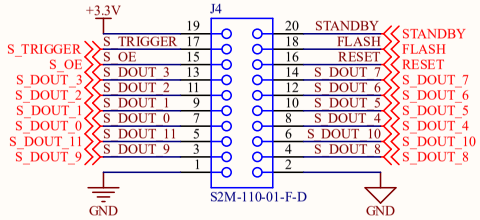
Title		
Size: A4	Number	Revision
Date: 11/07/2024		Sheet of
File: E:\AltiumProjects\...\PowerPins1.SchDoc		Drawn By:

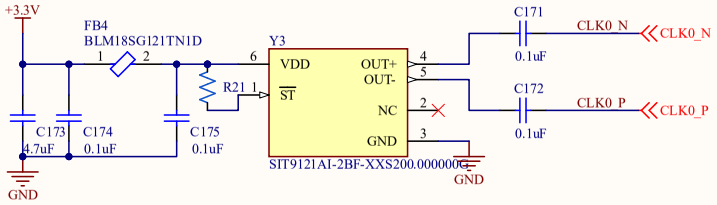
[10] Output Connectors

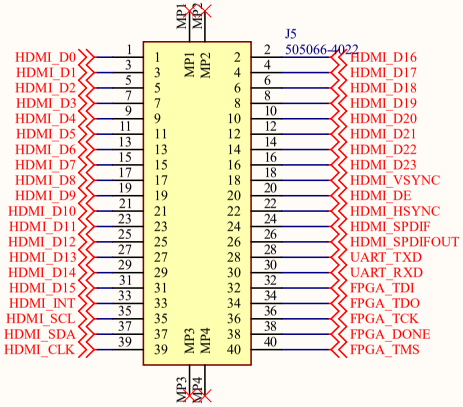






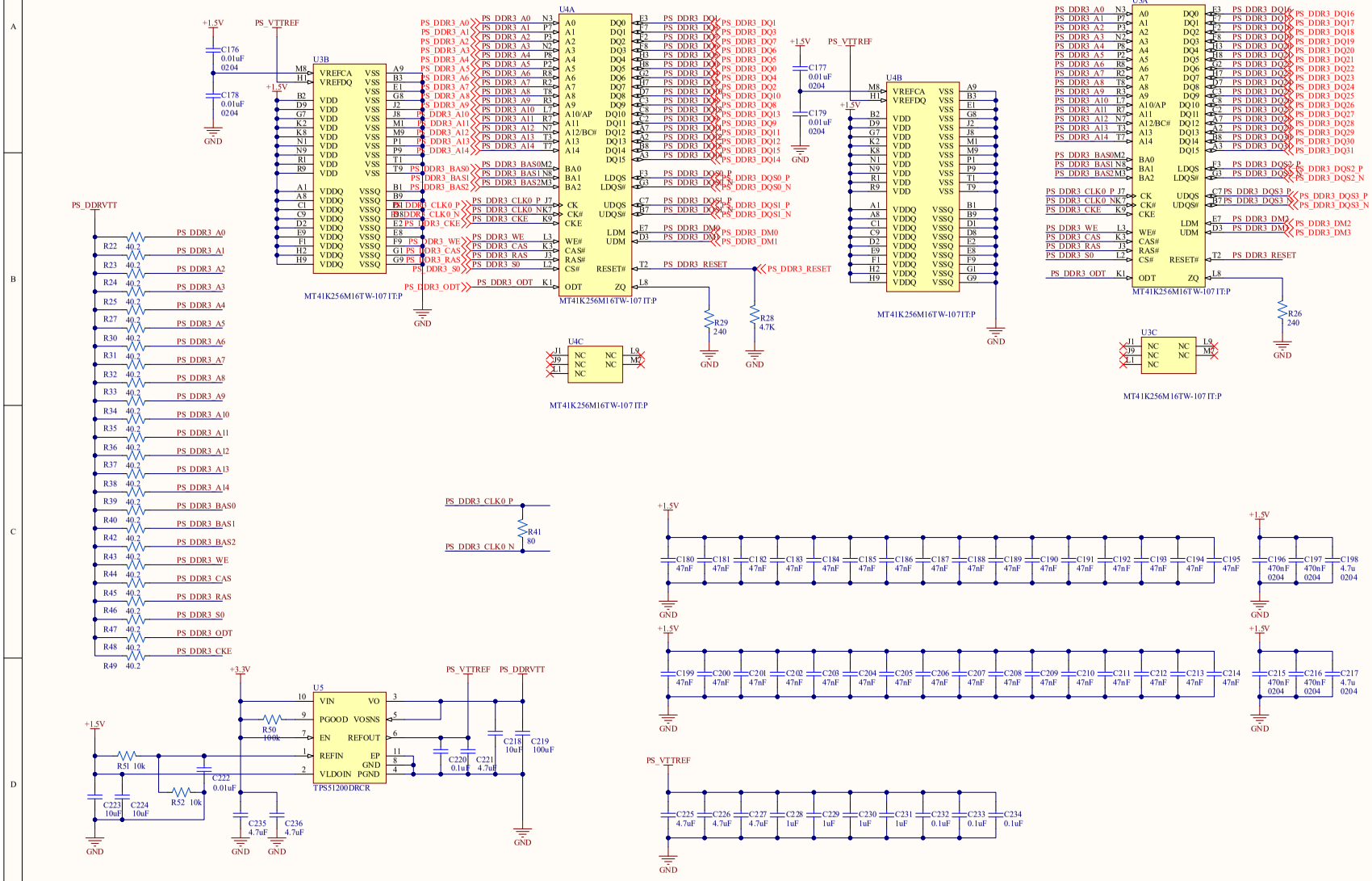




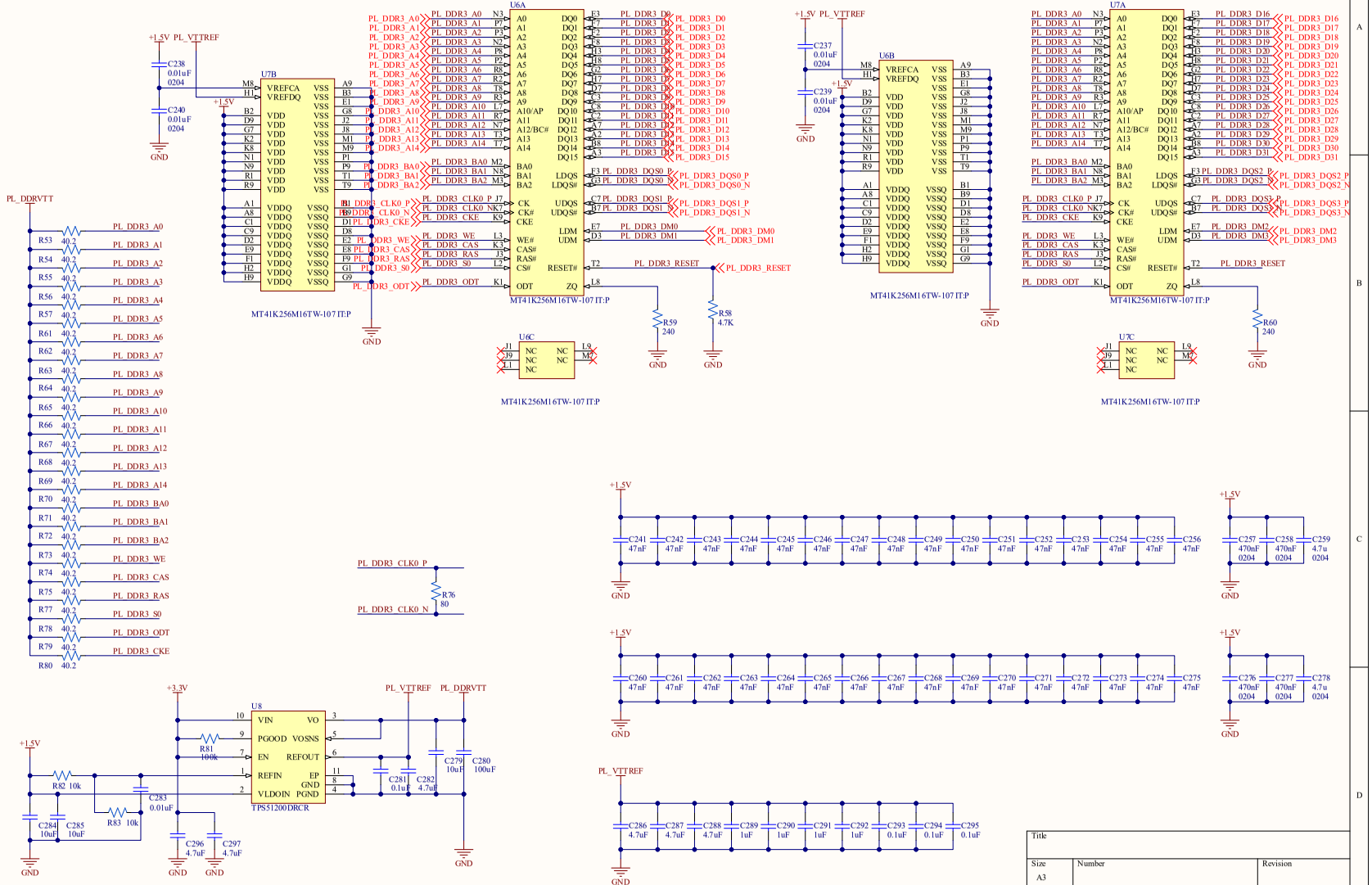


Title		
Size: A4	Number	Revision
Date: 11/07/2024		Sheet of
File: E:\AltiumProjects\..\fpgaConnections.SchDoc		Drawn By:

[11] DDR3 - Processing System (PS)

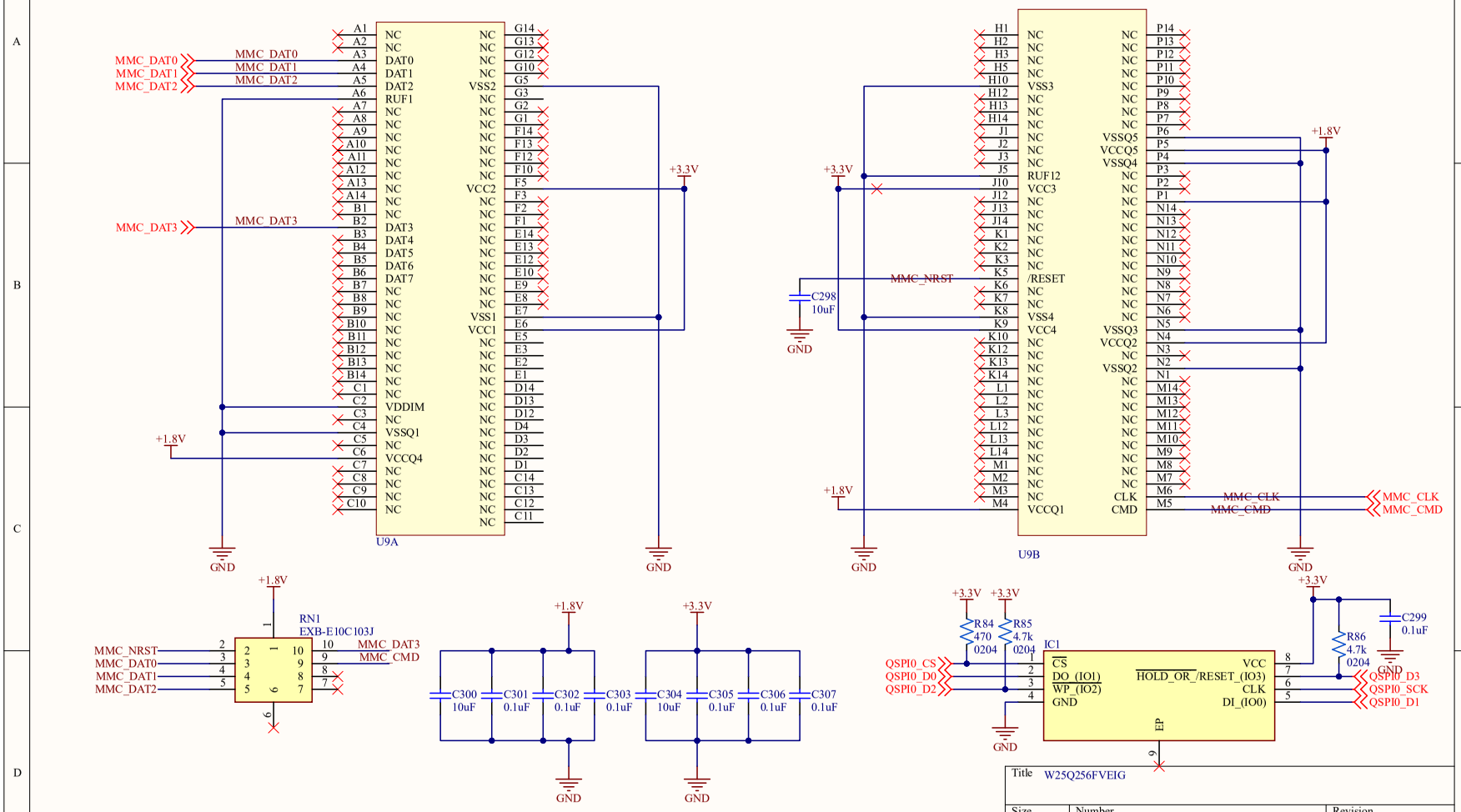


[12] DDR3 - Programable Logic (PL)



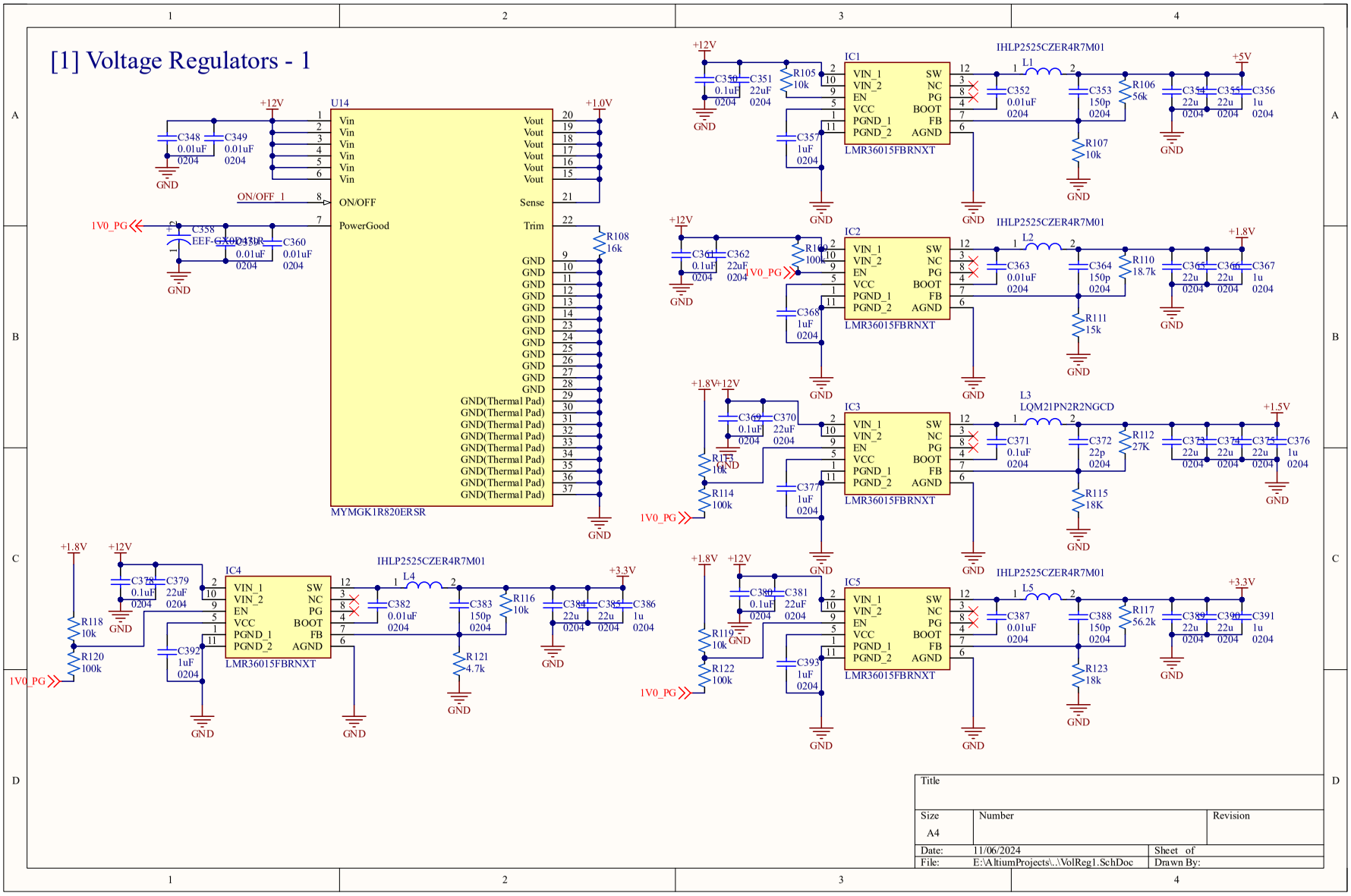
Title			
Size: A3	Number		Revision
Date: 11/07/2024		Sheet of	
File: E:\AltiumProjects\..\DDR3_PL.SchDoc		Drawn By:	

[13] eMMC Flash & QSPI Flash

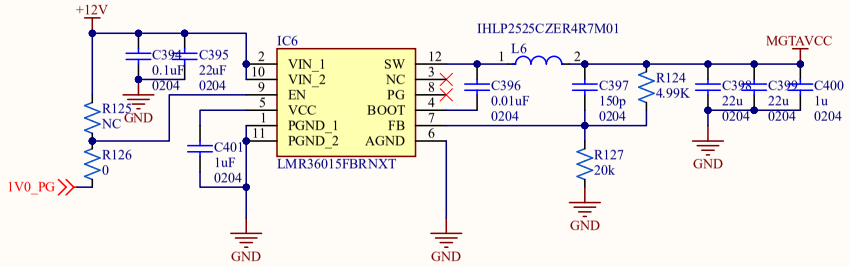


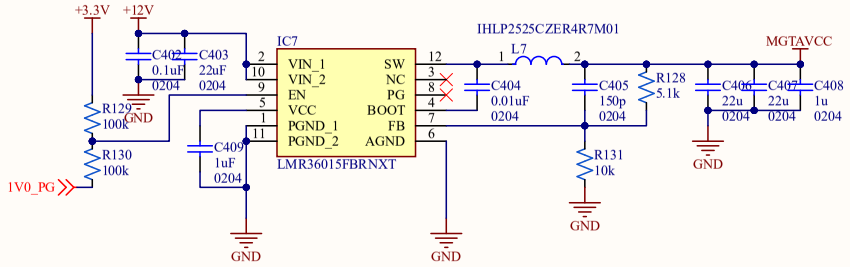
Title	W25Q256FVEIG	
Size: A4	Number	Revision
Date: 11/07/2024		Sheet of
File: E:\AltiumProjects\..\EMMc.SchDoc		Drawn By:

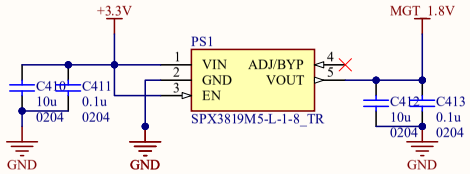
Schematic Power Supply and Output Interfaces


[1] Voltage Regulators - 1
U14
MYMGK1R820ERSR
Vin
Vout
ON/OFF
Sense
PowerGood
Trim
GND
GND(Thermal Pad)
IC1
IC2
IC3
IC4
IC5
LMR36015FBRNXT
VIN_1
VIN_2
EN
VCC
PGND_1
PGND_2
SW
NC
PG
BOOT
FB
AGND
IHLP2525CZER4R7M01
LQM21PN2R2NGCD
+12V
+1.8V
+1.0V
+5V
+3.3V
+1.5V
1V0_PG
ON/OFF_1
L1
L2
L3
L4
L5
R105 10k
R106 56k
R107 10k
R108 16k
R110 18.7k
R111 15k
R112 27K
R113 10k
R114 100k
R115 18K
R116 10k
R117 56.2k
R118 10k
R119 10k
R120 100k
R121 4.7k
R122 100k
R123 18k
C348 0.01uF 0204
C349 0.01uF 0204
C350 0.1uF 0204
C351 22uF 0204
C352 0.01uF 0204
C353 150p 0204
C354 22u 0204
C355 22u 0204
C356 1u 0204
C357 1uF 0204
C358 EEF-GX0D470R
C360 0.01uF 0204
C361 0.1uF 0204
C362 22uF 0204
C363 0.01uF 0204
C364 150p 0204
C365 22u 0204
C366 22u 0204
C367 1u 0204
C368 1uF 0204
C369 0.1uF 0204
C370 22uF 0204
C371 0.1uF 0204
C372 22p 0204
C373 22u 0204
C374 22u 0204
C375 22u 0204
C376 1u 0204
C377 1uF 0204
C378 0.1uF 0204
C379 22uF 0204
C380 0.1uF 0204
C381 22uF 0204
C382 0.01uF 0204
C383 150p 0204
C384 22u 0204
C385 22u 0204
C386 1u 0204
C387 0.01uF 0204
C388 150p 0204
C389 22u 0204
C390 22u 0204
C391 1u 0204
C392 1uF 0204
C393 1uF 0204
Title
Size A4
Number
Revision
Date: 11/06/2024
Sheet of
File: E:\AltiumProjects\..\VolReg1.SchDoc
Drawn By:

[2] Voltage Regulators - 2

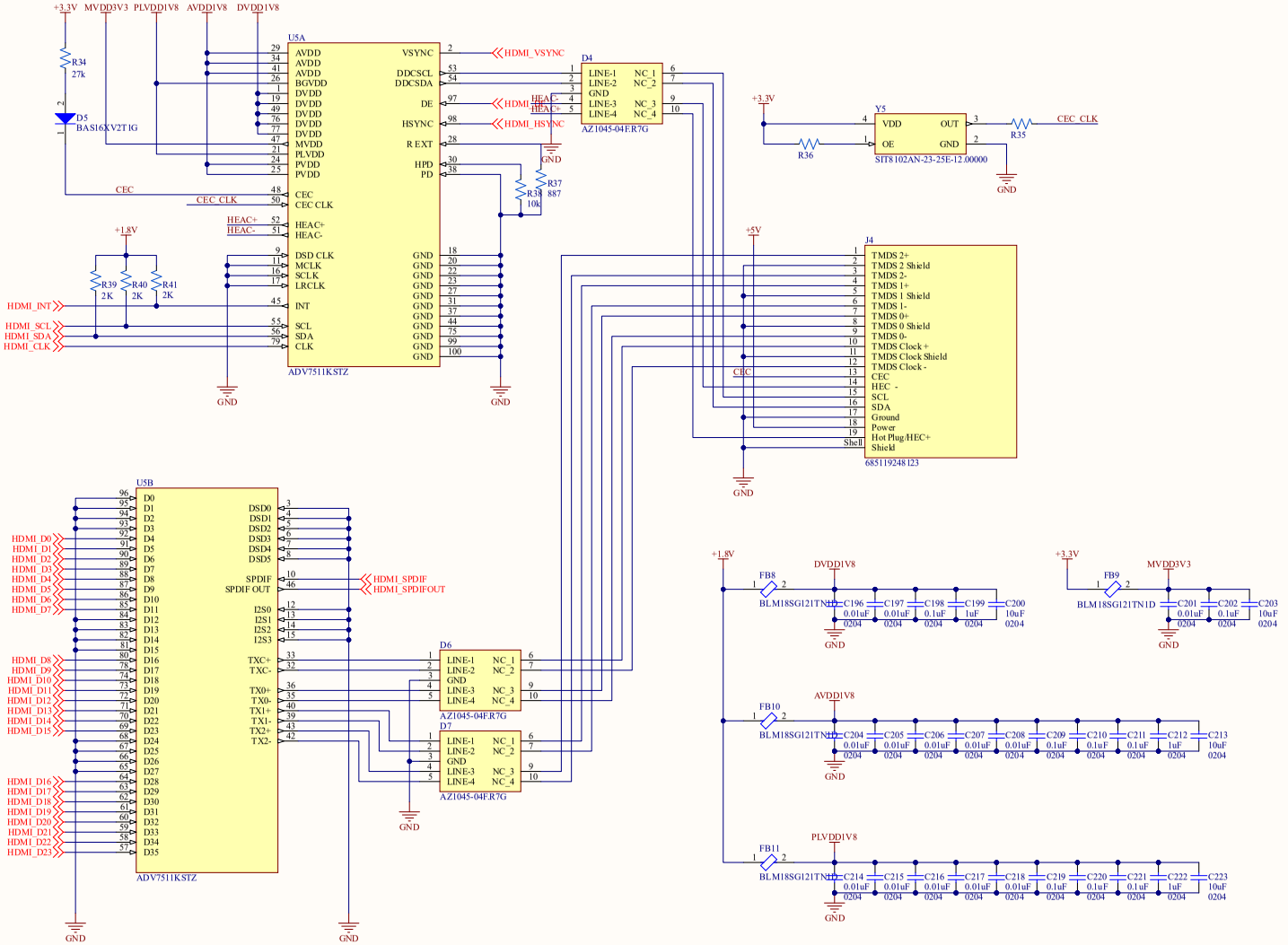




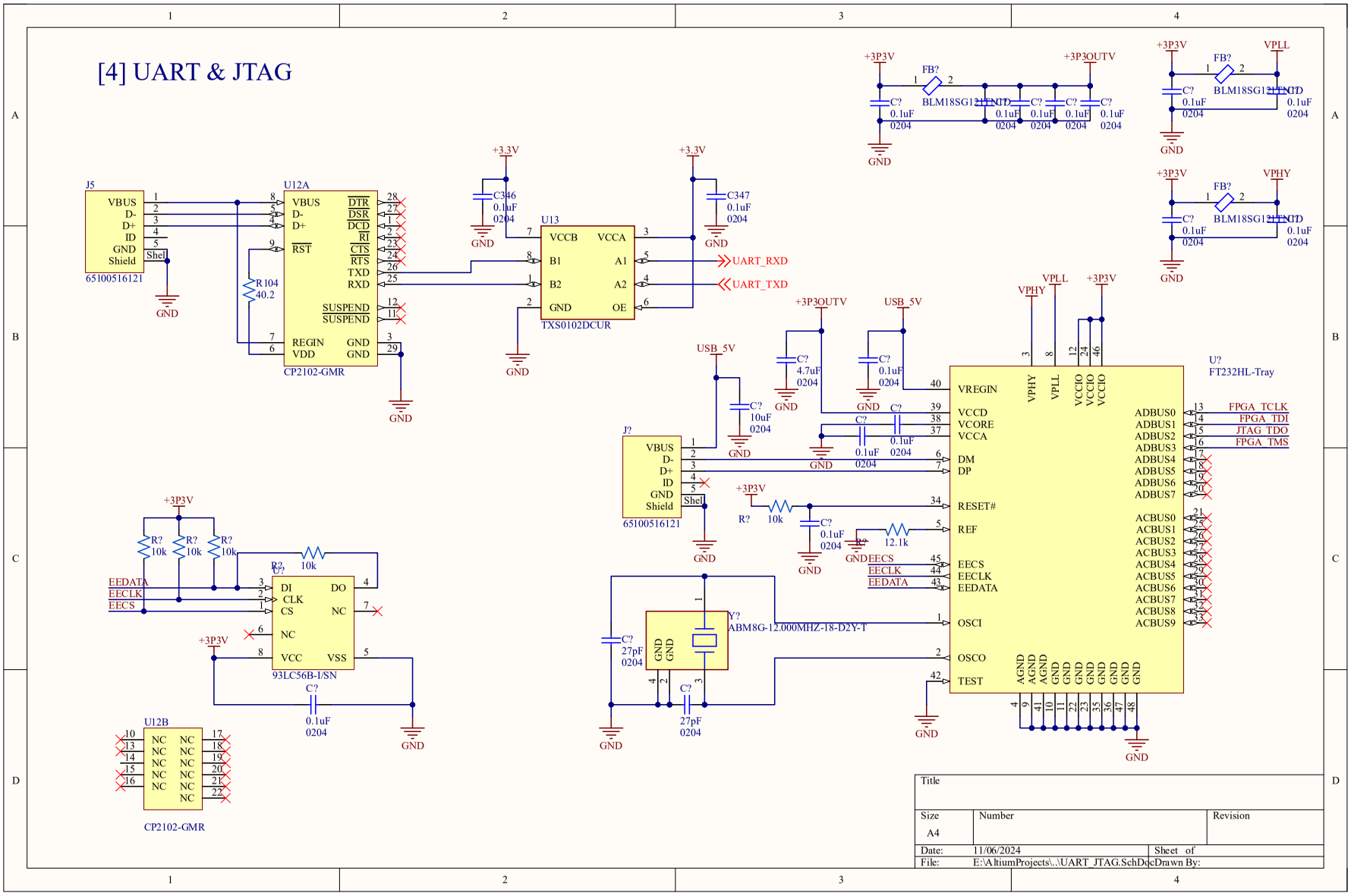


Title			
Size: A4	Number		Revision
Date:	11/06/2024	Sheet of	
File:	E:\AltiumProjects\...\VolReg2.SchDoc	Drawn By:	

[3] HDMI



Title			
Size A3	Number		Revision
Date:	11/06/2024		Sheet of
File:	E:\AltiumProjects\..\HDMI.SchDoc		Drawn By:


[4] UART & JTAG
J5
VBUS
D-
D+
ID
GND
Shield
65100516121
U12A
CP2102-GMR
R104
40.2
U13
TXS0102DCUR
UART_RXD
UART_TXD
J?
65100516121
U?
FT232HL-Tray
VREGIN
VCCD
VCORE
VCCA
DM
DP
RESET#
REF
EECS
EECLK
EEDATA
OSCI
OSCO
TEST
ADBUS0
ADBUS1
ADBUS2
ADBUS3
ADBUS4
ADBUS5
ADBUS6
ADBUS7
ACBUS0
ACBUS1
ACBUS2
ACBUS3
ACBUS4
ACBUS5
ACBUS6
ACBUS7
ACBUS8
ACBUS9
FPGA_TCLK
FPGA_TDI
JTAG_TDO
FPGA_TMS
Y?
ABM8G-12.000MHZ-18-D2Y-T
93LC56B-I/SN
EEDATA
EECLK
EECS
U12B
CP2102-GMR
+3.3V
+3P3V
+3P3OUTV
USB_5V
VPHY
VPLL
GND
Title
Size
A4
Number
Revision
Date:
11/06/2024
Sheet of
File:
E:\AltiumProjects\...\UART_JTAG.SchDoc
Drawn By: