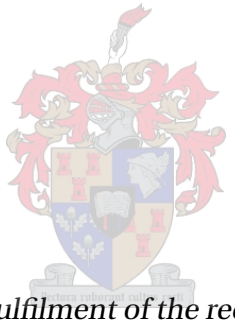
The Design and Development of a 64-Bit Linux Based Single Board Computer Specifically for Visible Light Positioning

by

Philip Strydom



Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electrical and Electronic) in the Faculty of Engineering at Stellenbosch University

Supervisor: Willem Smit

April 2022

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: April 2022 .....

Abstract

The Design and Development of a 64-Bit Linux Based Single Board Computer Specifically for Visible Light Positioning

P. Strydom

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MEng (E&E)

April 2022

The University of Stellenbosch and Katholieke Universiteit Leuven currently utilise freely available single board computers (SBC) for teaching and research purposes, but updates in future hardware iterations may render current software incompatible. A custom SBC is designed specifically for the needs of both institutions.

This SBC is based on the NXP i.MX8MQ ARM processor. The processor has 4 high performance ARM Cortex-A53 cores and 1 high efficiency ARM Cortex-M4F core.

This work successfully implements the i.MX8MQ processor alongside $2GB$ of LPDDR4 memory and SD card storage. This SBC has an analogue to digital converter (ADC), 2 46-pin expansion connectors, $100Mbps$ Ethernet, HDMI, 2 USB 3.0 and a UART-to-USB serial debug port. The power system of this SBC provides 16 voltage rails and is capable of delivering up to $50W$.

This design is implemented on a 6-layer $86.36mm$ x $55.88mm$ printed circuit board (PCB). The PCB has $4mil/4mil$ minimum width and spacing and $0.2mm$ via holes. The layer stackup of the PCB is custom designed to meet required impedance-, crosstalk- and timing constraints. The stackup has 4 signal layers, 1 power layer and 1 ground layer.

The PCB is manufactured and sub-assembled in China and completed at the University of Stellenbosch. Debugging is performed and the design is deemed to function well. A custom Linux image is compiled, loaded and found to function reliably.

Key words: Single Board Computer; Linux; 64-bit; Visible Light Positioning

Uittreksel

Die Ontwerp en Ontwikkeling van 'n 64-Bis Linux Gebaseerde Enkelbordrekenaar Spesifiek vir Sigbare Lig Liggingsbepaling

P. Strydom

Departement Elektries en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MIng (E&E)

April 2022

Die Universiteit van Stellenbosch en Katholieke Universiteit Leuven gebruik tans kommersiële enkelbordrekenaars vir onderrig- en navorsingsdoeleindes, maar toekomstige opdateering van hardeware kan bestaande sagteware onbruikbaar maak. 'n Doelgemaakte enkelbordrekenaar is spesifiek ontwerp vir die behoeftes van beide instansies.

Hierdie enkelbordrekenaar is gebaseer op die NXP i.MX8MQ ARM verwerker. Die verwerker het 4 hoë krag ARM Cortex-A53 kerne en 1 hoë effektiwiteit ARM Cortex-M4F kern.

Die navorsingsprojek implementeer die i.MX8MQ verwerker suksesvol tesame met $2GB$ LPDDR4 geheue en SD kaartberging. Hierdie enkelbordrekenaar het 'n analoog na digitaal omsetter, 2 46-pen uitbreidingsverbindings, $100Mbps$ Ethernet, HDMI, 2 USB 3.0 en 'n UART-na-USB ontfoutingspoort. Die kragstelsel van hierdie enkelbordrekenaar lewer 16 spanningsvlakke en is in staat om tot en met $50W$ te verskaf.

Die ontwerp is geïmplementeer op 'n 6-laag $86.36mm$ x $55.88mm$ etsbord. Die etsbord het $4mil/4mil$ minimum wydte en spasiëring en $0.2mm$ via gate. Die laagstapel van die etsbord is spesiaal ontwerp om aan die vereiste impedansie-, kruiskoppeling- en tydsbeperkings te voldoen. Die laagstapel het 4 seinlae, 1 kraglaag en 1 grondlaag.

Die etsbord is in Sjina vervaardig en gedeeltelik aanmekaar gesit en dan by die Universiteit van Stellenbosch voltooi. Ontfouting is gedoen en daar is gevind dat die ontwerp goed funksioneer. 'n Doelgemaakte Linux bedryfstelsel is gebou, gelaai en gevind om betroubaar werk.

Sleutelwoorde: Enkelbordrekenaar; 64-Bis; Linux; Sigbare Lig Liggingsbepaling

Acknowledgements

To my supervisor, Willem Smit for his support and guidance in the field of embedded design.

To Johan Arendse, Wessel Croukamp, Gift Lecholo and Wynand van Eeden for their technical assistance.

To my family for their enduring support.

Contents

List of Figures

List of Tables

Listings

Nomenclature

ADC	Analogue to Digital Converter
ARM	Advanced Risc Machines
AVB	Audio Video Bridge
BGA	Ball Grid Array
CEC	Consumer Electronics Control
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CSI	Camera Serial Interface
DDC	Display Data Channel
DDR	Double Data Rate
DIMM	Dual Inline Memory Module
DMA	Direct Memory Access
DSI	Display Serial Interface
DSP	Digital Signal Processing
EMI	Electromagnetic Interference
ESD	Electrostatic Discharge
ESR	Equivalent Series Resistance
FCBGA	Flip Chip Ball Grid Array
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input Output
GPU	Graphics Processing Unit
HCSL	High Speed Current Steering Logic
HDCP	High Bandwidth Digital Copy Protection
HDMI	High Definition Multimedia Interface
HPD	Hot-Plug Detect
I2C	Inter Integrated Circuit
I2S	Inter IC Sound
IEEE	Institute of Electrical and Electronic Engineers
JTAG	Joint Test Action Group
LED	Light Emitting Diode
MAC	Media Access Control
MDIO	Management Data Input Output
MIPI	Mobile Industry Processor Interface
MMC/SDIO	Multi Media Card/Secure Digital Input Output
OTP	One Time Programmable
PCB	Printed Circuit Board
PHY	Physical Layer
PLL	Phase Locked Loop
PMIC	Power Management Integrated Circuit

POR	Power On Reset
PRU-ICSS	Programmable Real-time Unit and Industrial Communications Subsystem
RAM	Random Access Memory
RF	Radio Frequency
RGMII	Reduced Gigabit Media Independent Interface
RISC	Reduced Instruction Set Computer
RMII	Reduced Media Independent Interface
RTC	Real Time Clock
RTOS	Real Time Operating System
SAR	Serial Approximation Register
SBC	Single Board Computer
SD	Secure Digital
SoC	System on Chip
SPI	Serial Peripheral Interface
TCK	Test Clock
TCM	Tightly Coupled Memory
TDI	Test Data In
TDO	Test Data Out
TI	Texas Instruments
TMDS	Transmission Minimised Differential Signaling
TMS	Test Mode Select
TRST	Test Reset
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USB-OTG	Universal Serial Bus On The Go
USD	United States Dollars
USMA	Micro Sub Miniature Version A Connector
VLP	Visible Light Positioning

Units

SI Prefixes

Symbol	Name	Quantity
T	tera-	1×10^{12}
G	giga-	1×10^{9}
M	mega-	1×10^{6}
k	kilo-	1×10^{3}
m	milli-	1×10^{-3}
μ	micro-	1×10^{-6}
n	nano-	1×10^{-9}
p	pico-	1×10^{-12}

SI Base Units

Symbol	Name	Quantity
A	ampere	Electric Current
m	metre	Distance
s	second	Time

SI Derived Units

Symbol	Name	Quantity
^oC	degree Celsius	Temperature
F	farad	Capacitance
H	henry	Inductance
Hz	hertz	Frequency
Ω	ohm	Resistance, Real Impedance
rad	radian	Angle
V	volt	Electrical Potential Difference
W	watt	Power

SI Compound Units

Symbol	Name	Quantity
s/V	Slew Rate	Signal Rise Rate

Allowed Non-SI Units

Symbol	Name	Quantity
d	day	Time
h	hour	Time
m	minute	Time

Imperial Units

Symbol	Name	Quantity
^{o}	degree	Angle
in	Inch	Distance
mil	thousands of an Inch	Distance
oz	Ounce	Weight

Industry Standard Units

Symbol	Name	Quantity
b	Bit	Binary Integer
bps	Bit per second	Binary Integer Transfers per second
B	Byte	8 Binary Integers
dB	decibel	Logarithmic Ratio
fps	Frames per second	Complete Video Frames Transfers per second
ppm	Parts per Million	Linear Ratio
S/s	Sampling Rate	Digital Samples per second
T/s	Transition Rate	Digital Threshold Transitions per second

Monetary Units

Symbol	Name	Quantity
$	United States Dollar	Monetary Value
R	South African Rand	Monetary Value

*Note that 8-bit, 16-bit, 32-bit and 64-bit are considered classification categories and therefore do not conform to the above standard.

Chapter 1

Introduction

1.1 Single Board Computers

A single board computer (SBC) is a complete computer system implemented on a single printed circuit board. Subsystems such as the power system, processor, memory, storage, I/O and graphics are integrated without the need for expansion cards. The size of a SBC typically ranges between that of a credit card to that of post card [81].

There are more than 340 SBCs on the market today [19]. The 1st SBC is widely regarded to be the Dyna-Micro that was built in the 1970s [84]. This computer utilised the Intel C8080 processor.

The 1st modern small form factor SBC is the BeagleBoard. The BeagleBoard was developed by the BeagleBoard Foundation and was released in June 2008 [15]. The popularity of SBCs exploded in the early 2010s with the release of the Raspberry Pi in February 2012 [17]. The Raspberry Pi was developed as a cooperation between Cambridge University's computer science department and engineers from Broadcom. The main advantage is their adequate performance at a very low price point. The cost for the 1st iteration of the Raspberry Pi was $35 [20].

SBCs are ubiquitous in projects ranging from hobby electronics to professional industrial designs [2] [63].

Advanced RISC Machines (ARM) based SBCs are by far the most popular [19]. The ARM architecture is a type of reduced instruction set computer (RISC). The philosophy of this architecture revolves around optimising the computer to do the more common tasks as quickly as possible and the more complex tasks as infrequently as possible. The result is a simpler CPU that is more power efficient and less expensive than more commonly used architectures such as x86 [22].

In recent years the release of 64-bit ARM architectures like the ARM Cortex-A53 and ARM Cortex-A57 has lead to very powerful products.

SBCs run a variety of operating systems depending on which architecture is employed. Most ARM SBCs run a variant of the Linux operating system while some x86 based SBCs are capable of running versions of Windows.

1.2 Visible Light Positioning

Visible light positioning (VLP) is an indoor positioning system that utilises already existing lighting infrastructure to determine location. It is meant to be an alternative to Global Positioning Systems (GPS) as the required line of sight to the satellite constellations are not typically available indoors [4]. The technology aims to be cost effective and very accurate [3].

VLP has been proposed for a number of applications such as leading visitors through a museum or as an aid in piloting autonomous vehicles.

Results can be obtained from numerous sources such as photodiodes, low resolution cameras, gyroscopes, magnetometers and LIDAR and then combined to achieve very accurate measurements. Computation of results are currently performed on a variety of SBCs and traditional computers. Traditional computers tend to have lower latency and tend to be less susceptible to thermal heating effects [1].

The higher performance of traditional computers also come at a higher cost. Some SBCs such as the BeagleBone Black have programmable real-time subsystems [35] that allow time critical applications such as waveform generation [85] to be offloaded from the main processor thereby increasing overall system performance for certain tasks.

A SBC of relatively low cost, higher performance and a higher analogue to digital sampling rate is recommended to further develop this research area.

1.3 Problem Statement

The BeagleBone Black is currently used by the University of Stellenbosch to teach programmable logic, hardware description languages, embedded systems and computer networks.

Katholieke Universiteit Leuven (KU Leuven) uses the BeagleBone Black as main processor of analogue signals to determine position within indoor environments where alternative technologies such as GPS is currently impractical [86].

The development of a custom SBC will ensure that current software will not become incompatible in future hardware iterations.

1.4 Requirement

The University of Stellenbosch requires a high-performance 64-bit computer with a rich operating system and KU Leuven requires a system with integrated analogue to digital converter (ADC) and fast interrupt return. This SBC must be relatively affordable. Design requirements are set by the 2 institutions and combined to achieve a system that would meet the needs of both. This study is funded by the University of Stellenbosch and KU Leuven.

1.5 Overview of study

This study documents the process of developing the Bokkiebord SBC. The requirements are first discussed in Chapter 2. These requirements are then translated into an ideal product in Chapter 4. The electronic design of the project is then proposed in Chapter 5. High-speed

layout theory is discussed, specifications are chosen, parameters are calculated, layout is completed and simulation is performed and discussed in Chapter 6. Manufacturing of the SBC is described in Chapter 7. The SBC is debugged and basic hardware functionality is achieved as discussed in Chapter 8. Software is configured and compiled as shown in Chapter 9. Benchmarks are performed in Chapter 10. A cost analysis is given in Chapter 11. This study is completed successfully.

1.6 Bokkiebord Design

The Bokkiebord design is shown in Figure 1.1 - 1.2. The Bokkiebord is built around the NXP i.MX8MQ processor. It has 4 ARM Cortex-A53 high-performance cores and 1 low power ARM Cortex-M4 high-efficiency core. The Bokkiebord has $2GB$ of low power double data rate version 4 (LPDDR4) memory running at up to $1580MHz$, a power system with 16 voltage rails providing up to $50W$, an ADC with a sampling rate of $3MS/s$, configurable boot options, reduced media independent interface (RMII) $100Mbps$ Ethernet, BeagleBone Black compatible general purpose input output (GPIO) expansion headers, High Definition Multimedia Interface (HDMI), Joint Test Action Group (JTAG) debug port, 1 Raspberry Pi compatible Display Serial Interface (DSI) connector, 2 Raspberry Pi Compatible Camera Serial Interface (CSI) connectors, Secure Digital (SD) Card storage, Universal Serial Bus (USB) to universal asynchronous receiver transmitter (UART) bridge and 2 USB 3.0 ports.

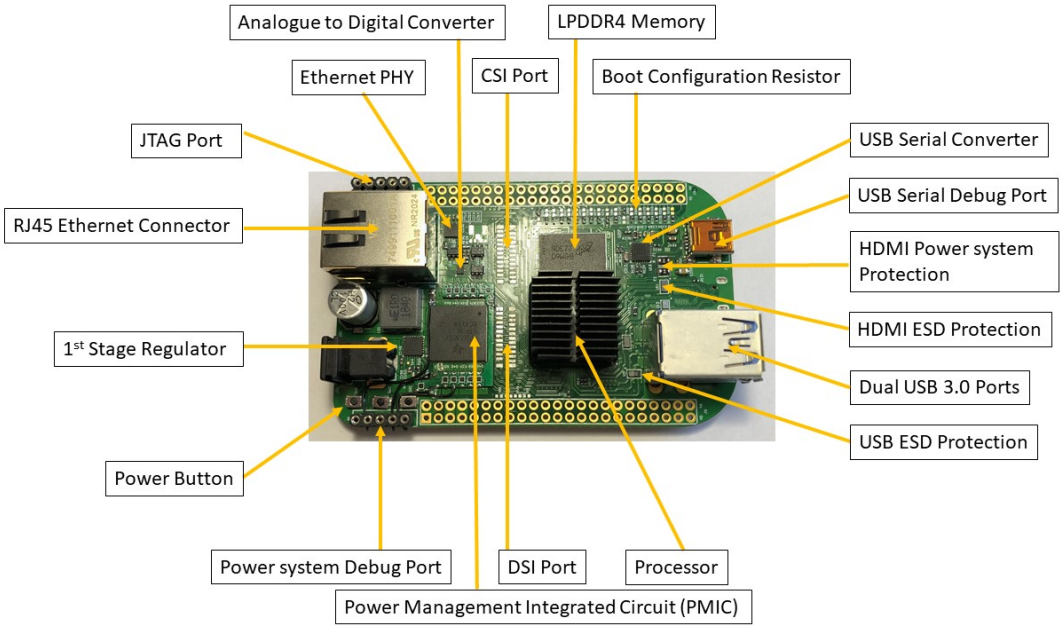


Figure 1.1: Final Bokkiebord Top View

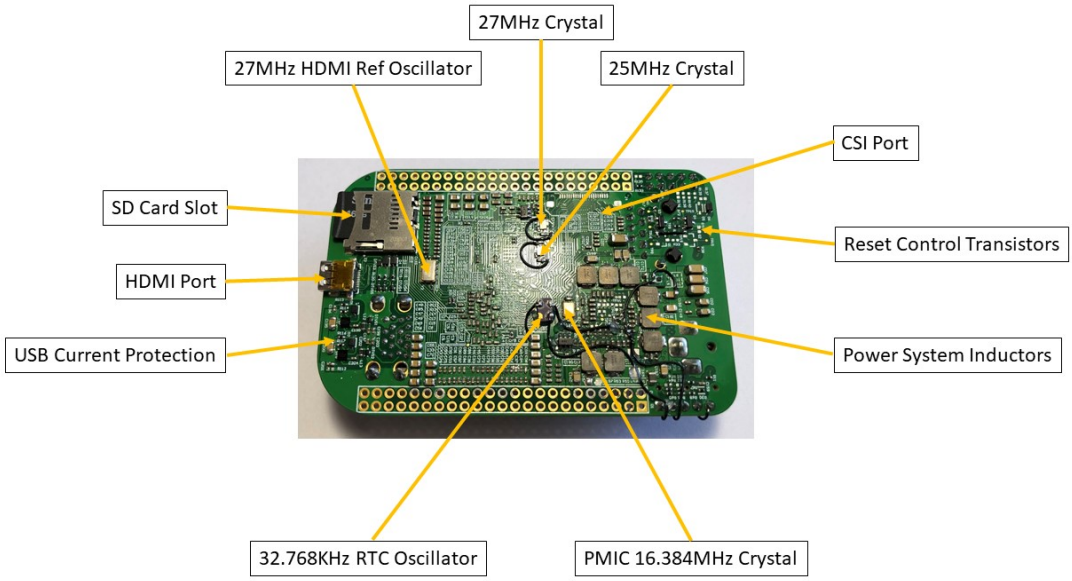


Figure 1.2: Final Bokkiebord Bottom View

Chapter 2

Requirements

This study aims to design and develop a Linux-based SBC specifically for VLP. This project is a dual effort by the University of Stellenbosch (South Africa) and KU Leuven (Belgium) to extend the development of visible light positioning algorithms outside of engineering into fields like computer science and mathematics.

KU Leuven and the University of Stellenbosch currently utilise the BeagleBone Black SBC but require their own SBC design to ensure that their software does not become incompatible with future hardware iterations.

Requirements from the 2 universities are diverse but are in line with the current SBC market offering. The requirements of the 2 universities are:

2.1 University of Stellenbosch Requirements

- At least 2 USB 2.0 ports
- A HDMI display interface
- At least 100$Mbps$ Ethernet
- At least 2GB of RAM
- An OpenGL and OpenCL capable graphics processing unit (GPU)
- Capable of running a rich 64-bit Linux Desktop like Ubuntu
- Use an SD card as primary storage
- Consume less than 10W
- Cost less than $100 to manufacture

2.2 KU Leuven Requirements

- At least 100$Mbps$ Ethernet
- Capable of running Linux as well as a real-time compatible operating system
- A real-time co-processor such as the Programmable Real-time Unit and Industrial Communications System (PRU-ICSS) from Texas Instruments

- Access to low power modes
- Power via micro-USB and at least one other alternative
- An ADC with a sampling rate of between $100kS/s$ and $3MS/s$
- Wireless connectivity such as WiFi and Bluetooth

The study aims to incorporate the above requirements as far as possible in the design, component selection and manufacturing process to meet most of the requirements of both institutions and within budget.

Chapter 3

Processor selection

3.1 Architectures

The current SBC market offering is evaluated and it is found that the majority of SBCs utilise 32-bit processing architectures while newer designs utilise 64-bit processing architectures. The SBC market is competitive, with the lower end being represented by products like the Raspberry Pi zero and the CHIP computer, while the higher end of the market is represented by products such as the Parallella board and the more GPU focused platforms such as the Nvidia Jetson Tegra K1, X1 and X2.

3.1.1 ARM

The processor core offering from ARM Holdings is divided into 3 main segments. The Cortex-A is optimised for the support of rich operating systems, the Cortex-R is optimised for real-time processing and fast reaction times and the Cortex-M range is developed specifically for low power consumption and micro-controller applications.

3.2 Ending support for 32-bit Linux

Support for 32-bit Linux operating systems is ending with software engineers from projects like Ubuntu calling it "increasingly unnecessary" as more devices are now 64-bit based. Building Linux for 32-bit systems increase build loads on development systems [47]. Popular programs such as Google Chrome have stopped support for 32-bit Linux as of March 2016 [21]. This is considered during processor selection.

3.3 Manufacturer Comparison

ARM does not manufacture processors but licenses the intellectual property to 3rd party manufacturers. A search is conducted to identify a manufacturer with a suitable 64-bit processor. The search process is outlined below and results are summarized in table 3.1:

- Licensees: Identify ARM architecture licensees by type.
- Feasibility: Determine whether manufacturers sell suitable processors to the public.
- Obtainability: Determine product obtainability by the University of Stellenbosch.
- Suitability: Evaluate processor suitability for project requirements.

Company	Core		Architecture		GPU	Feasible	Distributors					Obtainable	Cortex M	Cortex A	64 bit Application	Suitability
	32-Bit	64-Bit	32-Bit	64-Bit			RS	Mouser	Digikey	Avnet	Direct					
Allwinner	-	-	-	-	Yes	No										
AMD	Yes	Yes	-	Yes	-	No										
Apple	-	-	Yes	Yes	-	No										
Applied Micro	-	Yes	-	Yes	-	No										
Broadcom	Yes	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes	No	Network
Calvium	-	-	-	Yes	-	No										
Calexia	-	Yes	-	-	-	No										
Faraday Technology	-	-	Yes	-	-	No										ASIC
Huawei (HiSilicon)	Yes	Yes	-	Yes	Yes	No										Contact
IBM	Yes	-	-	-	-	No										
Infineon	Yes	-	-	-	-	Yes	Yes	Yes	Yes	-	Yes	Yes	Yes	No	No	
Intel	Yes	-	Yes	-	-	No										
LG	Yes	-	-	-	-	No										
Marvell Technology	-	-	Yes	Yes	-	Yes	-	-	-	-	Yes	Yes	-	Yes	Yes	
MediaTek	-	-	-	-	Yes	Yes	-	-	-	-	-					
Microchip	Yes	-	-	-	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	32-Bit
Microsemi	Yes	-	-	-	-	Yes	-	Yes	Yes	-	Yes	Yes	Yes	No		
Microsoft	-	-	Yes	-	-	No										
Nvidia	-	-	-	Yes	-	No										
NXP	Yes	Yes*	-	-	-	Yes	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes	Yes	64-Bit
Qualcomm	-	-	Yes	-	-	Yes	-	Yes	Yes	-	-	Yes	-	Yes	Yes	Contact
Renesas	Yes	-	-	-	-	Yes	Yes	Yes	Yes	-	-	Yes	Yes	Yes	Yes	Automotive
Rockchip	Yes	Yes	-	-	Yes	Yes	-	-	-	-	-					
Samsung	Yes	Yes		Yes	-	Yes	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes	Yes	Contact
Spreadtrum	-	-	-	-	Yes	Yes	-	-	-	-	Yes	Yes	-	Yes	Yes	Contact
STMicro	Yes	Yes	-	-	-	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	-		
TI	Yes	Yes*	-	-	-	Yes	Yes	Yes	Yes	-	Yes	Yes	Yes	Yes	No	32-Bit

Table 3.1: Manufacturer Comparison Summary

Table 3.1 shows the manufacturers that produce suitable components for this SBC with only 4 supplying products through local distributors. The 4 manufacturers are Microchip, NXP, Renesas and Texas Instruments.

3.3.1 Microchip

Microchip offers 2 lines of 32-bit processors, the SAMA5 (ARM Cortex-A5 based) and the SAM9 (ARM 926EJ-S based). The processors offer a rich set of peripherals and user interfaces. They deliver high performance, market-leading low power and advanced security for cost-sensitive applications [48]. Linux for these devices is supplied by Microchip as a starting point. These processors are located at the bottom to the middle of the market, offering good value for money.

3.3.2 Renesas

Renesas has an extensive range of low power devices that is primarily aimed at cost-sensitive applications. Renesas offers a 64-bit range of processors that is aimed primarily at the automotive market.

3.3.3 Texas Instruments

Texas Instruments offers a range of 32-bit application Cortex-A processors. This Sitara range is offered in the AM335x, AM4x and AM57x series. Of the 3, the AM335x offers a cost-optimized solution. The AM4x offers real-time control and security, while the AM57x line of processors offers advanced ARM architectures with digital signal processing (DSP) capabilities optimised for performance and multimedia. The Sitara range of processors includes the PRU-ICSS real-time subsystem, which is given as a requirement in the specification of KU Leuven [26]. Texas Instruments support Linux and real-time operating systems (RTOS) for the AM335x, AM4x and AM57x.

3.3.4 NXP

NXP offers a range of ARM Cortex-A processors that is sold under the i.MX name. The i.MX range is one of the most versatile platforms for multimedia and display applications, delivering a good balance of power, performance and integration. The i.MX8 range contains the high-performance i.MX8MQ and i.MX8 processor families. These processors offer very high performance in 64-bit systems. This 64-bit range is also one of the few that can be easily obtained through distributors. This range offers support for Linux, Android and FreeRTOS.

3.4 Processor Selection

The above processor search has delivered 3 possible options each offering specific advantages and disadvantages. These processors are:

Texas Instruments:

- AM572x (2x Cortex-A15 32-bit) [$22.33-$40.47]

NXP (Freescale Semiconductor):

- i.MX8MQ (4 Cortex-A53 1 Cortex-M4F 1 GPU 64-bit) [$50.49]
- i.MX8 (2 Cortex-A72 4 Cortex-A53 2 Cortex-M4F 64-bit) [$211.84]

(Prices as on selection date)

The AM572x offers the best compatibility to the BeagleBone Black but has a limited service life due to limited future support for 32-bit Linux [74]. The i.MX8MQ offers a long service life with high versatility but reduced compatibility with the current system. Finally, the i.MX8 provides long service life with very high performance but relatively high cost.

3.4.1 Most Compatible

The most compatible option is the Sitara AM572x family of processors. This option offers an increased amount of processing capacity while requiring the least amount of software rework. Software compatibility is increased as both the currently used Cortex-A8 processor of the BeagleBone Black and the Cortex-A15 of the AM572x are based on the the ARMv7 microarchitecture.

The AM572x also contains the PRU-ICSS real-time processing subsystem, which is propriety to Texas Instruments.

The major disadvantage of this option is that the processor is 32-bit that will result in limited service life due to ending support for 32-bit Linux [74].

3.4.2 Most Versatile

The most versatile option is the NXP i.MX8MQ processor. The i.MX8MQ contains 4 ARM Cortex-A53 high power cores, 1 ARM Cortex-M4F low power core and 1 GPU. It supports multiple GPU libraries, including OpenGL 3.1, OpenCL 1.2 and Vulkan and can display both $1080p$- and $4K$ video. It supports gigabit Ethernet and DDR4, LPDDR4 and DDR3L memory types.

Standard features include advanced security, Ethernet with Audio Video Bridge (AVB), USB 3.0 with physical layer (PHY), Multi Media Card / Secure Digital Input Output (MMC/SDIO), UART, Serial Peripheral Interface (SPI), Inter Integrated Circuit (I2C), Inter IC Sound (I2S), Timers, secure real time clock (RTC), Neon Media Processor Engine and Integrated Power management [51].

This processor is ideally suited to offer the best 64-bit performance at the most affordable price of all the processors considered.

The ARM Cortex-A53 core implements the ARMv8-A microarchitecture, which might require some software rework as it differs from the ARMv7 microarchitecture that is implemented on the Cortex-A8 used by the BeagleBone Black [14].

The i.MX8MQ contains a Cortex-M4F low power processor which can be be used as a separate core, but does not contain a dedicated real-time unit.

The main advantage of this design is the continuing support for 64-bit Linux, which extends the service life of the SBC. This option also best fits the specifications of the University of Stellenbosch.

3.4.3 Most Powerful

The most powerful option is the NXP i.MX8 processor line. The i.MX8 contains 2 Cortex-A72 and 4 Cortex-A53 high power cores with 2 Cortex-M4F low power cores, 1 DSP and 2 GPUs. It supports multiple GPU Libraries, including OpenVX(Vision), OpenGL ES, OpenCL and Vulkan, and can display in $1080p$ and $4K$ video. It supports gigabit Ethernet and DDR4 and LPDDR4 memory types.

Standard features include advanced security, Ethernet with AVB, USB 3.0 with PHY, MMC/SDIO, UART, SPI, I2C, I2S, Timers, Secure RTC, Neon Media Processor Engine and Integrated Power management [51].

The i.MX8 processor requires the same amount of software rework as the i.MX8MQ. The i.MX8 processor has the ability to expand the application of this SBC to fields that require intense processing, such as image recognition and machine learning.

This processor is in the pre-production phase at the time of processor selection and a non-disclosure agreement is required to obtain documentation before the official release.

3.5 Conclusion

A choice is made between the highly compatible AM572x processor, the long service life of the 64-bit NXP i.MX8MQ and the high performance of the NXP i.MX8 processor. The **NXP i.MX8MQ** is chosen as the most affordable option that can be engineered to meet the specifications of both the University of Stellenbosch and KU Leuven.

The i.MX8MQ is one of the few processor options that offer a 64-bit architecture that can be acquired and implemented by the University of Stellenbosch. The i.MX8MQ is capable of extended service life due to its 64-bit architecture.

Chapter 4

Conceptual Design

This study aims to create a SBC that meets the consolidated requirements as discussed in the Chapter 2. The SBC is designed to be as similar as possible to the BeagleBone Black by ensuring pin to pin compatibility and a similar board outline. The function of the two SBCs are the same, but different components are utilised to implement this functionality. The SBC is developed directly without the use of a development board to contain project cost. Figure 4.1 shows the Bokkiebord Block Diagram.

4.1 Processor

Figure 4.1 [55] shows that the i.MX8MQ has a main CPU and a low power/security CPU. The main CPU consists of 4 ARM Cortex-A53 cores with integrated NEON and floating-point unit. The NEON unit could be used for implementing an advanced instruction set that helps accelerate multimedia, graphics and signal processing, while the floating-point unit can be used for floating-point number calculation. The processor has integrated $32KB$ L1 data cache, $32KB$ L1 instruction cache and $1MB$ L2 cache. The low power core has a $16KB$ instruction cache, $16KB$ data cache and a Tightly Coupled Memory Module (TCM). The TCM can be considered similar to on-die RAM and is primarily used for security purposes.

4.2 Peripherals

The **system on chip (SoC)** is implemented on a printed circuit board (PCB) with a similar size and shape to the BeagleBone Black. The SoC is placed in the middle of the PCB.

The **LPDDR4 Memory** is placed as close as practically possible to the SoC to maintain timing requirements. The LPDDR4 is $16Gb$ ($2GB$) in capacity and is connected directly to the central processing unit (CPU) through 2 16-bit wide data channels running at up to $1600MHz$ double data rate (DDR) ($3200MT/s$).

The onboard clock is produced using a **phase-locked loop (PLL)** utilising 2 crystals operating at $25MHz$ and $27MHz$. **RTC** functionality is provided by an external oscillator and is required for board bring-up. Reset and control circuitry is implemented using discrete CMOS circuitry.

Primary storage is provided through a **SD card**. The SD card is connected directly to the SoC and is accessible via a connector placed on the periphery of the PCB.

The primary debugging ports are the **JTAG** port placed on the periphery of the board and 2

UART ports. **UART ports** are connected to an onboard USB-to-UART converter. Communication from the onboard USB-to-UART interface is made possible via a miniUSB connector on the board's periphery.

USB connectors are connected directly to the SoC via electrostatic discharge (ESD) protection and filtering components. Both USB ports are of **USB 3.0** type and can be used in both host and OTG modes. Power for USB devices is supplied via integrated over-current protection and de-glitching circuits.

A micro **HDMI 2.0** port is connected directly to the CPU via ESD protection and filtering circuits. This port is capable of Consumer Electronics Control (CEC) bi-directional communication and can transmit video of up to $4K$ at $60fps$. The reference clock for the HDMI interface is provided by an external HCSL oscillator.

The SoC has 1 onboard **DSI** port and 2 onboard **CSI** ports. These Mobile Industry Processor Interface (MIPI) ports are compatible with Raspberry Pi connectors.

$100Mbps$ **Ethernet** network communication is implemented using a standard RJ45 Ethernet connector. The SoC is connected to an Ethernet PHY via RMII.

Expansion header pin types and voltage levels are kept the same as the BeagleBone Black except for the pins used for analogue to digital conversion.

Analogue pins are removed from the expansion header and remain unconnected. ADC input is moved to a dedicated uSMA connector elsewhere on the board. The signal from the uSMA connector is first buffered and then low pass filtered. The input impedance of the ADC can be set by populating 0402 resistors reserved for this purpose. The **ADC** is connected to the SoC via a SPI port running at $48MHz$. The analogue to digital converter is of the serial approximation register (SAR) type and can run at up to $3MS/s$ with a 12-bit resolution. The ADC requires a voltage reference that is placed just adjacent.

The SoC is powered via a **power management integrated circuit (PMIC)**. Power for the PMIC is supplied via a $5V$ input phase. The power system can deliver up to $50W$.

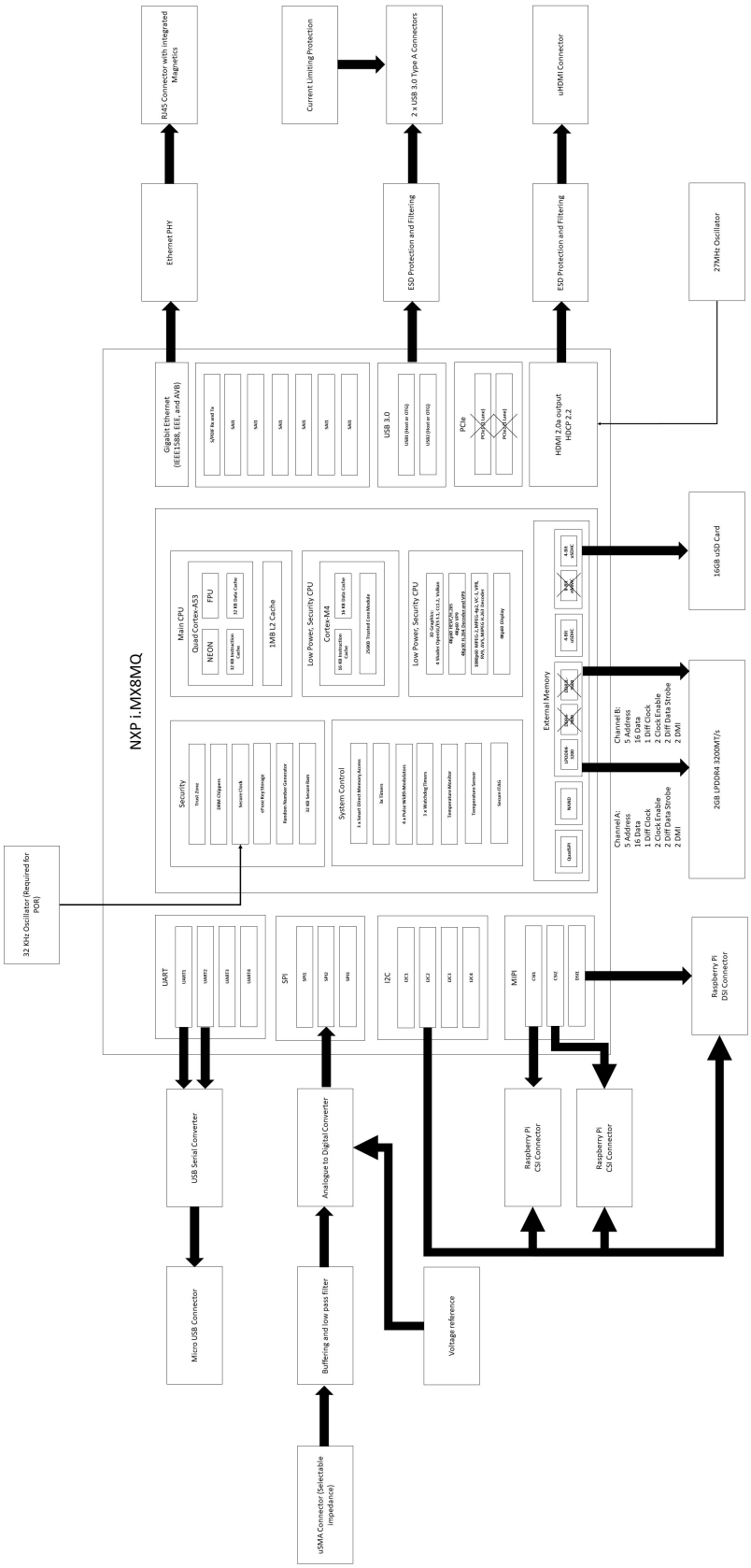


Figure 4.1: Bokkiebord Block Diagram

Chapter 5

Component Selection and Electronic Design

This chapter documents specific aspects of the Bokkiebord design. Refer to appendix A.1 for the final schematic and refer to appendix A.3 for bill of materials.

5.1 Power System

Most components on the PCB are powered directly from a TPS659037 (PMIC). This TPS659037 PMIC can only accept a very specific input voltage range of between $-0.3V$ and $6V$. A TPS54A24 first power system input stage is added to protect the PMIC from an over voltage condition. Upon debugging the 1st power system input stage is found to function perfectly while the PMIC requires extensive debugging to function acceptably. PMIC problems are attributed mostly to subtle documentation errors seldom seen from Texas Instruments.

5.1.1 1st Power Input Stage

The PMIC only accepts a very specific voltage range. An input switching power supply stage ensures that PMIC input voltage does not exceed recommended ranges. It consists of a TPS54A24 buck regulator with a WQFN footprint type that is primarily chosen for its small size, large input range and high current capacity. The 1st power input stage is shown in Figure 5.1.

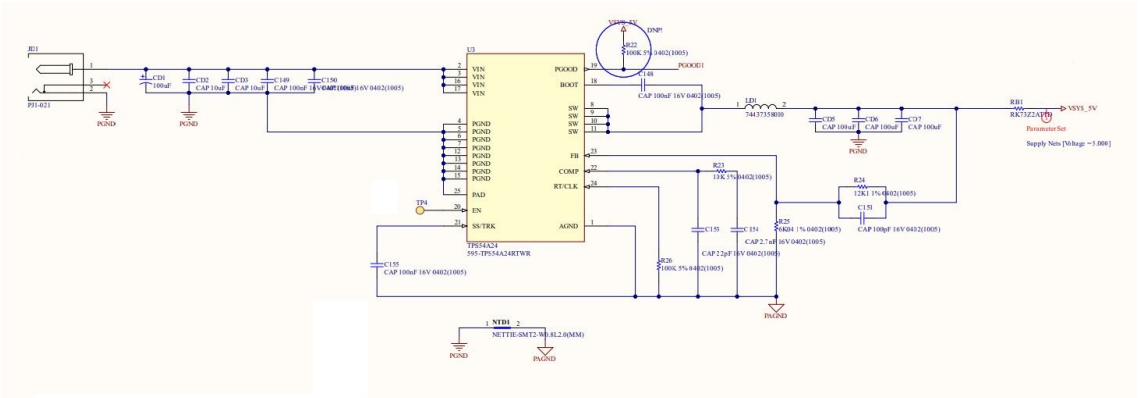


Figure 5.1: TPS54A24 1st Power Input Stage.

5.1.1.1 Switching Frequency Selection

The 1st power input stage supports switching frequencies between $200kHz$ and $1.6MHz$. Smaller components can be used if a high switching frequency is chosen. The main trade off is extra switching power loss. Maximum switching frequency is limited if a specific turn on time is required. Ideally the switching frequency of the 1st power input stage would be the same as the 2nd stage so that the range of frequencies being emitted from the board, however small, can be contained to a known specific band.

The 2nd power stage, the PMIC, supports switching frequencies between $1.7MHz$ and $2.7MHz$. The maximum switching frequency of $1.6MHz$ is chosen for the 1st power input stage as it will allow for the smallest components to be used, cause the least ripple, and contain the range of frequencies being emitted. A $5V$ output at up to $10A$ is required as input by the PMIC and the maximum 1st stage input voltage is chosen to be below $12V$.

The switching frequency is given as a function of **turn on time**, maximum input voltage and required output voltage by [38, Eq13, P24]:

$$f_{sw_{(max)}} = \frac{1}{t_{on_{(min)}}} \times \frac{V_{out}}{V_{in_{(max)}}} \tag{5.1}$$

Rewritten as:

$$t_{on_{(min)}} = \frac{V_{out}}{V_{in_{(max)}} \times f_{sw_{(max)}}} \tag{5.2}$$

$$t_{on_{(min)}} = \frac{5}{12 \times 1.6 \times 10^6} \tag{5.3}$$

$$\Rightarrow t_{on_{(min)}} = 260.416666810^{-9}[s] \tag{5.4}$$

Where:
$f_{sw_{(max)}}$ is the regulator switching frequency in hertz.
$t_{on_{(min)}}$ is the minimum regulator startup time in seconds.
$V_{in_{(max)}}$ is the maximum regulator input voltage in volt.
V_{out} is the regulator output voltage in volt.

A startup time of $260.41ns$ is required which is deemed acceptable.

The R_T resistance controls the **switching frequency** of the regulator. A smaller resistor value results in a higher switching frequency. The required R_T resistor is a function of desired switching frequency and is given by [38, Eq14, page 24]:

$$R_T = 58650 \times (f_{sw})^{-1.028} \tag{5.5}$$

$$R_T = 58650 \times (1600)^{-1.028} \tag{5.6}$$

$$\Rightarrow R_T = 29.81485313 \approx 30[k\Omega] \tag{5.7}$$

Where:
R_T is the switching frequency configuration resistor value in kilo-ohm.
f_{sw} is the switching frequency in kilohertz.

The R_T resistance is calculated to be $30k\Omega$ which is deemed to be acceptable. R_T is given by R26 in Figure 5.1.

5.1.1.2 Output Inductor Selection

The required **inductance for inductor** selection is a function of required transient voltage, output current, switching frequency and current ripple and is given by [38, Eq15, page 25]:

$$L_1 = \frac{V_{in_{(max)}} - V_{out}}{I_O \times K_{ind}} \times \frac{V_{out}}{V_{in_{max}} \times f_{sw}} \tag{5.8}$$

$$L_1 = \frac{12-5}{10 \times K_{ind}} \times \frac{5}{12 \times 1.6 \times 10^6} \tag{5.9}$$

$$L_1 = \frac{0.182291666 \times 10^{-6}}{K_{ind}} [H] \tag{5.10}$$

Where:
L_1 is the required regulator inductor inductance in henry.
$V_{in_{(max)}}$ is the maximum regulator input voltage in volt.
V_{out} is the regulator output voltage in volt.
I_O is the required output current in ampere.
f_{sw} is the regulator switching frequency in hertz.
K_{ind} is the ratio of inductor current ripple relative to maximum output current and is dimensionless.

K_{ind} is the ratio of inductor current ripple relative to maximum output current. Smaller inductor ripple results in slower transient response. Smaller current ripple means that the inductor is capable of supplying a smaller gulp of current to the capacitor bank. It can be seen from equation 5.10 that the inductor size increases with smaller current ripple. Choose K_{ind} as 0.2 for smaller ripple and slower transient response.

Calculated required inductor inductance is:

$$\Rightarrow L_1 = 0.911458333 \times 10^{-6} \approx 1[\mu H] \tag{5.11}$$

The datasheet recommends inductor part number 74437358010. It is a $1\mu H$ inductor with a saturation current of $32.5A$, peak RMS current of $14A$ and a typical DC resistance of $3.65m\Omega$. The inductor is well shielded.

The **ripple current** of this configuration is given by [38, Eq16, page 25]:

$$I_{ripple} = \frac{V_{in_{(max)}} - V_{out}}{I_O \times K_{ind}} \times \frac{V_{out}}{V_{in_{(max)}} \times f_{sw}} \tag{5.12}$$

$$\Rightarrow I_{ripple} = 1.822916667[A] \tag{5.13}$$

Where:
I_{ripple} is the regulator output ripple current in ampere.
$V_{in_{(max)}}$ is the maximum regulator input voltage in volt.
V_{out} is the regulator output voltage in volt.
I_O is the required output current in ampere.
f_{sw} is the regulator switching frequency in hertz.
K_{ind} is the ratio of inductor current ripple relative to maximum output current and is dimensionless.

The constant **RMS current** of this configuration is given by [38, Eq17, page 25]:

$$I_{L_{RMS}} = \sqrt{{I_O}^2 + \frac{1}{12} \times \left(\frac{V_{out} \times (V_{in_{(max)}} - V_{out})}{V_{in_{(max)}} \times L_1 \times f_{sw}} \right)^2} \tag{5.14}$$

$$\Rightarrow I_{L_{RMS}} = 10.01383637[A] \tag{5.15}$$

Where:
$I_{L_{RMS}}$ is the RMS current through the inductor in ampere.
I_O is the required output current in ampere.
V_{out} is the regulator output voltage in volt.
$V_{in_{(max)}}$ is the maximum regulator input voltage in volt.
f_{sw} is the regulator switching frequency in hertz.
L_1 is the regulator inductor inductance in henry.

The **peak current** of this configuration is given by [38, Eq18, page 25]:

$$I_{L_{peak}} = I_{out} + \frac{I_{ripple}}{2} \tag{5.16}$$

$$\Rightarrow I_{L_{Peak}} = 11.83675303[A] \tag{5.17}$$

Where:
I_{ripple} is the regulator output ripple current in ampere.
$I_{L_{Peak}}$ is the peak current though the inductor in ampere.

The minimum necessary **output capacitance** is a function of the maximum output voltage- and current change and switching frequency during periods of high current consumption and is given by [38, Eq19, page 26]:

Choose voltage change as $35mV$ for a 100% change in load current.

$$C_{out} > \frac{I_{out_{change}}}{V_{out_{change}}} \times \frac{1}{2 \times \pi \times (\frac{f_{sw}}{10})} \tag{5.18}$$

$$C_{out} > \frac{10}{35 \times 10^{-3}} \times \frac{1}{2 \times \pi \times \frac{1.6 \times 10^6}{10}} \tag{5.19}$$

$$C_{out} > 284.205 \approx 300[\mu F] \tag{5.20}$$

Where:
C_{out} is the minimum required output capacitance in farad.
$I_{out_{change}}$ is the maximum change in current draw in ampere.
$V_{out_{change}}$ is the maximum allowable voltage dip or spike in volt.
f_{sw} is the regulator switching frequency in hertz.

Output Capacitor will consist of 3 $100\mu F$ capacitors in parallel.

5.1.1.3 Output Voltage Resistor Selection

The **output voltage** is controlled by a voltage divider biasing the feedback port FB. The internal voltage reference is $0.6V$. [38, Eq25, P27]:

$$R_{FBT} = R_{FBB} \times \left(\frac{V_{out}}{V_{ref}} - 1 \right) \tag{5.21}$$

Choosing $R_{FBB} = 30k\Omega$:

$$R_{FBT} = 30 \times 10^3 \times \left(\frac{5}{0.6} - 1 \right) \tag{5.22}$$

$$\Rightarrow R_{FBT} = 220[k\Omega] \tag{5.23}$$

Where:
R_{FBT} is the value of the voltage divider resistor connected between the output voltage and FB measure voltage in ohm.
R_{FBB} is the value of the voltage divider resistor between the FB measure voltage and ground in ohm.
V_{out} is the regulator output voltage in volt.
V_{ref} is the regulator internal reference voltage in volt and is always $0.6V$.

R_{FBT} is given by R24 which is chosen to be $220k\Omega$ and R_{FBB} is given by R25 and is chosen to be $30k\Omega$ in Figure 5.1.

5.1.2 2nd Power Stage: PMIC

The TPS659037 is the main power supply for all components except USB on the Bokkiebord. It is chosen because of its 14 regulated outputs, high current capacity and the assumed capability to be reprogrammed. At selection time it is thought a more elegant solution than the PF4210 recommended by NXP because it requires far less external components. It is controlled via I2C. It is supplied by $5V$ at up to $10A$ from the TPS54A24 1st power input stage.

The PMIC powers the entire board. Many different voltages and sequencing orders are required. The TPS659037 is adapted for use with the Bokkiebord. Table 5.1 shows voltage rail-voltages, currents, sequencing orders and power margins. Note that this table shows the required parameters that are not standard TPS659037 One Time Programmable (OTP) programming.

Seq	Symbol	Grouping	Voltage (V)			Cur. (mA)	Total	Source	Capacity	Margin
			Min	Typ	Max	Max				
0	RTC_RESET_B	-	-	-	-	-	-	LDO9		
0	POR_B	-	-	-	-	-	-	LDOLN		
1	NVCC_SNVS	NVCC_SNVS	3	3.3	3.6	5	5	LDO3	200	195
2	VDD_SNVS	VDD_SNVS	0.81	0.9	0.99	5	5	LDO4	200	195
3	RTC_RESET_B	-	-	-	-	-	-	LDO9		
4	VDD_SOC	VDD_SOC	0.9	0.95	0.99	2500	2500	SMPS45	4000	1500
5	VDD_ARM	VDD_0V9	0.81	0.9	1.05	3100	6620	SMPS123	9000	2380
	VDD_GPU		0.81	0.9	1.05	2040				
	VDD_VPU		0.81	0.9	1.05	610				
	VDD_DRAM		0.81	0.9	1.05	870				
6	VDDA_DRAM	VDDA_1P8	1.71	1.8	1.89	30	108.3	LDO1	300	191.7
	VDDA_1P8		1.71	1.8	1.89	20				
	VDD_1P8_XTAL_25M		1.71	1.8	1.89					
	VDD_1P8_XTAL_27M		1.71	1.8	1.89					
	VDD_1P8_TSENSOR		1.71	1.8	1.89					
	EXT_MEM_VDD1		-0.4	1.8	2.1	58.3				
7	DRAM_VREF	NVCC_DRAM	0.519	0.55	0.597	10	10	EXT	35	25
7	NVCC_DRAM	NVCC_DRAM	1.06	1.1	1.17	750	2044.13	SMPS6	3000	955.87
	EXT_MEM_VDD2		-0.4	1.1	1.5	1162				
	EXT_MEM_VDDQ		-0.4	1.1	1.5	86.55				
	EXT_MEM_BUFF		0.9	1.1	3.1	45.58				
7	NVCC_JTAG	NVCC_XXX	3	3.3	3.6	1000	1428	SMPS7	2000	572
	NVCC_SD1		3	3.3	3.6					
	NVCC_SD2		3	3.3	3.6					
	NVCC_NAND		3	3.3	3.6					
	NVCC_SAI1		3	3.3	3.6					
	NVCC_SAI2		3	3.3	3.6					
	NVCC_SAI3		3	3.3	3.6					
	NVCC_SAI5		3	3.3	3.6					
	NVCC_ECSPI		3	3.3	3.6					
	NVCC_I2C		3	3.3	3.6					
	NVCC_UART		3	3.3	3.6					
	NVCC_ENET		3	3.3	3.6					
	NVCC_GPIO1		3	3.3	3.6					
	EXT_ETH_VDDIO3V3		3	3.3	3.6	14				
	EXT_SD_VDD3V3		2.7	3.3	3.6	400				
	EXT_DAC		2.97	3.3	3.63	14				
8	USB1_DVDD	VDD_PHY_0V9	0.85	0.9	0.99	9.2	132.2	SMPS8	1000	867.8
	USB2_DVDD		0.85	0.9	0.99	9.2				
	USB1_VP		0.85	0.9	0.99	35.7				
	USB2_VP		0.85	0.9	0.99	35.7				
	USB1_VPTX		0.85	0.9	0.99	21.2				
	USB2_VPTX		0.85	0.9	0.99	21.2				
	PCIE_VP		0.85	0.9	0.99	76.2				
	PCIE_VPTX		0.85	0.9	0.99	28.6				
	MIPI_VDDA		0.85	0.9	0.99	35.89				
	MIPI_VDD		0.85	0.9	0.99	14.4				
	MIPI_VDDPLL		0.85	0.9	0.99	3.8				
	HDMI_AVDDCLK		0.85	0.9	0.99	98.89				
	HDMI_AVDDCORE		0.85	0.9	0.99	95.89				
8	PCIE_VPH	VDD_PHY_1V8	1.7	1.8	1.9	86	99.721	LDO2	300	200.279
	MIPI_VDDHA		1.7	1.8	1.9	7.17				
	HDMI_AVDDIO		1.7	1.8	1.9	6.551				
8	EXT_ADC_CONV	VDD_ANALOGUE_3V3	3.069	3.3	3.63	1.1	41.56	LDOUSB	100	58.44
	EXT_ADC_REF		3.069	3.3	3.63	0.46				
	EXT_ADC_OPAMP		3.069	3.3	3.63	40				
8	USB1_VDD33	VDD_PHY_3V3	3.069	3.3	3.63	24.5	139.6	SMPS9	1000	860.4
	USB1_VPH		3.069	3.3	3.63	20.3				
	USB2_VDD33		3.069	3.3	3.63	24.5				
	USB2_VPH		3.069	3.3	3.63	20.3				
	EXT_ETH_AVDD3V3		3	3.3	3.6	50				
8	USB1_VBUS	VDD_PHY_5V	0.8	1.4	5.25	1000	2000	LDSW		
	USB2_VBUS		0.8	1.4	5.25	1000				
9	POR_B	-	-	-	-	-	-	LDOLN		

Table 5.1: PMIC Voltage Rail Voltages, Currents, Sequencing orders and Power Margins

5.1.2.1 PMIC Internal Power

Figure 5.2 shows power input and sensing lines for the PMIC. VCC1, LDO_SUPPLY, VCC_SENSE, VBUS are fed from the VSYS_5V output supplied by the 1st power input stage. VIO_IN is initially designed to be supplied by a PMIC output, but is finally supplied from a regulated $3.3V$ delayed load switch as discussed in Section 8.3.2. LDORTC_OUT is an always on rail that is used for internal logic and power-on detection. LDORTC_OUT is recommended as debugging voltage. VPROG is a programming voltage that can theoretically be used to alter the OTP. This possibility is discussed in Section 9.1.1.

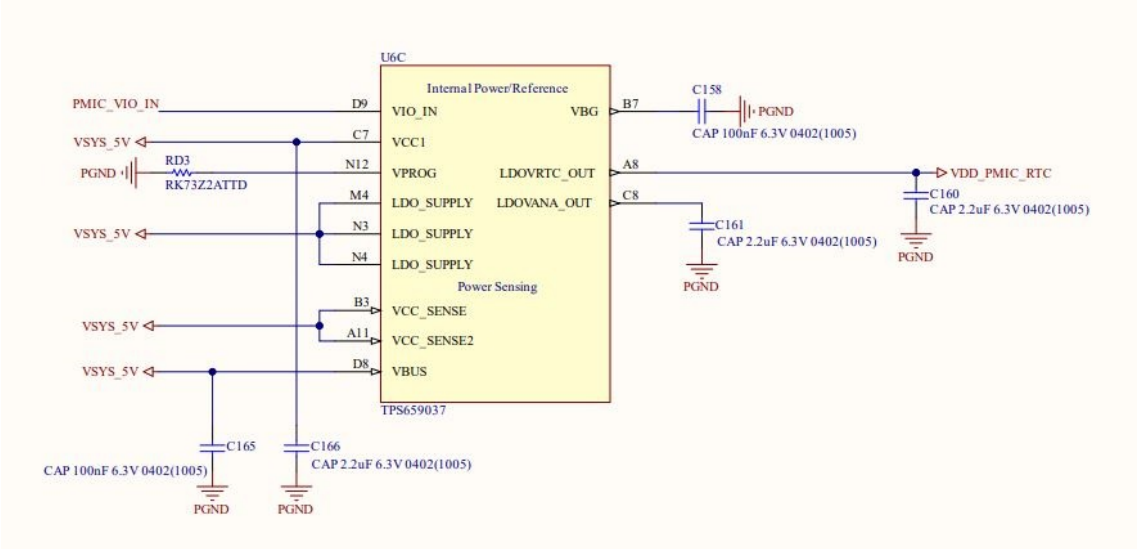


Figure 5.2: PMIC Internal

Figure 5.3 shows the System Control, GPIO and Communication of the TPS659037 PMIC.

5.1.2.2 PMIC Control Pins

Boot configuration pins **BOOT0 and BOOT1** control the power-up behavior of the PMIC. These pins set the initial voltage of the SMPS3 and LDO2 outputs [41, page 8-9]. Both pins are in the VRTC 1.8V voltage domain. BOOT0 is a floating input while BOOT1 has an internal 14$k\Omega$ pull-down resistor by default [39, page 11].

The **PWRDOWN** Pin is a power-down signal [39, page 9]. The pin is in the VRTC voltage domain and has a 5.25V absolute maximum rating. The pin is active-high and has an internal 400$k\Omega$ pull-down resistance by default [39, page 11]. The pin is connected via a 10$k\Omega$ pull-down resistor to ground in this design. The pin can be left floating if unused [39, page 9].

The **RESET_IN** Pin is a reset input signal[39, page 9]. The pin is in the VRTC voltage domain and has a 5.25V absolute maximum voltage rating. The pin is active-low and has an internal 400$k\Omega$ pull-down resistance [39, page 12]. The pin is connected to the VRTC voltage rail via a 10$k\Omega$ pull-up resistor. The pin is also connected to ground via push-button switch SW2. Figure 5.4 shows this implementation. The pin must be pulled high to exit reset state.

The **ENABLE1** Pin is an enable signal [39, page 9]. The pin is in the VIO_IN voltage domain. The pin is active-high and has an internal 400$k\Omega$ pull-down resistance [39, page 12]. The pin is left floating in this design. Note that the function of this pin is controlled by the OTP and is masked by default which means that the pin cannot be used to disable outputs until the pin is set not-masked by the I2C [40, page 97]. I2C communication is not possible until after initial power-up sequence has been completed.

The **PWRON** Pin is an external power-on signal [39, page 8]. The pin is in the VSYS voltage domain. This pin is active-low and has a fixed internal pull-up resistance of 120$k\Omega$ [39, page 11]. The pin is connected to ground via a 200Ω resistor and the SW1 push-button switch. Figure 5.4 shows this implementation. The pin can be left floating if unused [39, page 8].

The **RPWRON** Pin is an external power-on signal [39, page 7]. The pin is in the VSYS voltage domain. The pin is active-low and has a fixed internal pull-up resistance of $120k\Omega$ [39, page 11]. The pin is connected to to a non-inverting level shifter consisting of a 2N7002 N-Channel Enhancement Mode transistor that allows the processor PMIC_ON_REQ pin in the $3.3V$ NVCC_SNVS voltage domain to control the RPWRON pin on the PMIC in the $5V$ VSYS voltage domain. The pin can be left floating if unused [39, page 7].

The **NSLEEP** Pin is an external not-sleep signal [39, page 8]. The pin is in the VRTC voltage domain. The pin is active-low and has a fixed internal pull-up resistance of $400k\Omega$ [39, page 12]. The pin is connected to an external pull-up resistor with resistance of $10k\Omega$. This pin is also connected to an inverting level shifter consisting of a 2N7002 N-Channel Enhancement Mode transistor. The NSLEEP pin is connected to the drain port of the transistor and the gate port is connected to the PMIC_STBY_REQ pin of the processor. The source port of the transistor is connected to ground. This allows the processor PMIC_STBY_REQ pin in the $3.3V$ NVCC_SNVS voltage domain to control the NSLEEP pin on the PMIC in the $1.8V$ VRTC voltage domain. The pin can be left floating if unused [39, page 7].

The **NRESWARM** Pin is an external warm reset input request pin [39, page 8]. The pin is in the VRTC voltage domain. The pin is active-low and has an internal pull-up resistance of $120k\Omega$ [39, page 12]. When triggered the PIMC reloads default OTP values while running [39, page 72]. This pin is left floating in this design. Note that although the pin is connected to a 0Ω resistor to ground in the schematic, this resistor is not populated and the footprint is reserved for debugging purposes only. This pin can be left floating if unused[39, page 8].

The combined PMIC control pin configuration is shown in Table 5.2.

Name	Active State	Default Internal Impedance	Configuration	Comment
BOOT0	Active-High	Floating	Connected to Ground via 10$k\Omega$ Resistor	Populate RD5, Do Not Populate R35
BOOT1	Active-High	14$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate RD4, Do Not Populate R34
PWRDOWN	Active-High	400$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate R74
RESET_IN	Active-Low	400$k\Omega$ Pull-Down	Connected to VRTC via 10$k\Omega$ Resistor	Populate R31 and SW2
ENABLE1	Active-High	400$k\Omega$ Pull-Down	Floating	Do Not Populate R40
PWRON	Active-Low	120$k\Omega$ Pull-Up	Connected to Ground via 200Ω Resistor and Push-Button Switch	Populate R30, C159, SW1
RPWRON	Active-Low	120$k\Omega$ Pull-Up	Connected to Non-Inverting Level Shifter, Not Currently Implemented	Do Not Populate R33, Q1
NSLEEP	Active-Low	400$k\Omega$ Pull-Up	Connected to Inverting Level Shifter, Not Currently Implemented	Populate R36, Do Not Populate R38, R41, Q2
NRESWARM	Active-Low	120$k\Omega$ Pull-Up	Floating	Do Not Populate R42

Table 5.2: PMIC Control Pin Configuration

5.1.2.3 PMIC GPIO Pins

The **GPIO_0** Pin is a GPIO pin [39, page 7]. The pin is configured as input by default [39, page 11]. The pin is in the VRTC voltage domain. The pin is active-high or low and has an internal pull-down resistance of 400$k\Omega$ [39, page 11]. The pin is connected to ground via a 10$k\Omega$ resistor in this design. This pin can be connected to VSYS or Ground if unused [39, page 7].

The **GPIO_1** Pin is a GPIO pin [39, page 7]. The pin is configured as input by default [39, page 11]. The pin is in the VSYS voltage domain. The pin is active-high or low and has an internal pull-down resistance of 400$k\Omega$ [39, page 11]. The pin is connected to ground via a 10$k\Omega$ resistor in this design. This pin can be left floating if unused [39, page 7].

The **GPIO_2** Pin is a GPIO pin [39, page 7]. The pin is configured as input by default [39, page 11]. The pin is in the VSYS voltage domain. The pin is active-high or low and has an internal pull-down resistance of 400$k\Omega$ [39, page 11]. The pin is connected to ground via a 10$k\Omega$ resistor in this design. This pin can be left floating if unused [39, page 7].

The **GPIO_3** Pin is a GPIO pin [39, page 9]. The pin is configured as input by default [39, page 11]. The pin is in the VRTC voltage domain. The pin is active-high or low and has an internal

pull-down resistance of $400k\Omega$ [39, page 11]. The pin is connected to ground via a $10k\Omega$ resistor in this design. This pin can be connected to ground if unused [39, page 9].

The **GPIO_4** Pin is a GPIO pin [39, page 9]. The pin is configured as input by default [39, page 11]. The pin is in the VIO_IN voltage domain. The pin is active-high or low and has an internal pull-down resistance of $400k\Omega$ [39, page 11]. The pin is connected to ground via a $10k\Omega$ resistor and also serves as the load switch mod-board input source as discussed in Section 8.3.3.

The **GPIO_5** Pin is a GPIO pin [39, page 7]. The pin is configured as input by default [39, page 11]. The pin is in the VRTC voltage domain. The pin is active-high or low and has an internal pull-down resistance of $400k\Omega$ [39, page 11]. The pin is connected to ground via a $10k\Omega$ resistor in this design. This pin can be connected to ground if unused [39, page 7].

The **GPIO_6** Pin is a GPIO pin [39, page 10]. The pin is configured as input by default [39, page 12]. The pin is in the VIO_IN voltage domain. The pin is active-high or low and has an internal pull-down resistance of $400k\Omega$ [39, page 12]. The pin is connected to ground via a $10k\Omega$ resistor in this design. This pin can be connected to ground if unused [39, page 10].

The **GPIO_7** Pin is a GPIO pin that can be configured as POWERHOLD [39, page 8]. The pin is in the VRTC voltage domain. In GPIO Mode the pin is configured as input by default [39, page 12]. The pin is active-high or low and has an internal pull-down resistance of $400k\Omega$ [39, page 12]. In POWERHOLD Mode the pin is an input [39, page 12]. The pin is active-high and has an internal pull-down resistance of $400k\Omega$ [39, page 12]. The POWERHOLD Pin overrides the DEV_ON event to keep the PMIC active even if no other on-events are forcing the PMIC to be active [39, page 68]. The pin is connected to VRTC via a $10k\Omega$ pull-up resistor in this design. This pin can be connected to ground or VRTC if unused [39, page 8].

The combined PMIC GPIO pin configuration is shown in Table 5.3.

Name	**Active State**	**Default Internal Impedance**	**Configuration**	**Comment**
GPIO_0	Active-High or Active-Low	400$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate R43
GPIO_1	Active-High or Active-Low	400$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate R45
GPIO_2	Active-High or Active-Low	400$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate R47
GPIO_3	Active-High or Active-Low	400$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate R50
GPIO_4	Active-High or Active Low	400$k\Omega$ Pull-down	Connect to Ground via 10$k\Omega$ Resistor and Connect to Load switch Mod-Board	Populate R52 and Mod-Wire to Mod-Board
GPIO_5	Active-High or Active-Low	400$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate R53
GPIO_6	Active-High or Active-Low	400$k\Omega$ Pull-Down	Connected to Ground via 10$k\Omega$ Resistor	Populate R57
GPIO_7 (POWER-HOLD)	Active-High	400$k\Omega$ Pull-Down	Connected to VRTC via 10$k\Omega$ Resistor	Populate R58

Table 5.3: PMIC GPIO Pin Configuration

5.1.2.4 PMIC I2C Pins

The **I2C2_SCL_SCE** Pin is an I2C clock input pin for the 2nd I2C interface bus [39, page 10]. The pin is in the VIO_IN voltage domain. The pin is I2C active-high and requires an external pull-up resistor [39, page 12]. The pin is connected via a 0Ω bridge resistor to the clock input pin of the 1st I2C interface and is pulled-up to the VIO_IN voltage rail supplied by the external regulator module discussed in Section 8.3.2. This pin is connected directly to the power system debug header. Note that although the pin is connected to the processor bus via a 0Ω bridge resistor in the schematic, this this resistor is not populated in this design. This pin can be left floating if unused [39, page 10].

The **I2C1_SDA_SDI** Pin is a bi-directional I2C data pin for the 1st I2C interface bus [39, page 9]. The pin is I2C active-high and requires an external pull-up resistor [39, page 12]. The pin is connected via a 0Ω bridge resistor to the bi-directional data input pin of the 2nd I2C interface and is pulled-up to the VIO_IN voltage rail supplied by the external regulator module discussed in Section 8.3.2. This pin is connected directly to the power system debug header. Note that although the pin is connected to the processor bus via a 0Ω bridge resistor in the schematic, this resistor is not populated in this design. This pin can be left floating if unused [39, page 9].

The **I2C1_SCL_SCK** Pin is an I2C clock input pin for the 1st I2C interface bus [39, page 9]. The pin is in the VIO_IN voltage domain. The pin is I2C active-high and requires an external pull-up resistor [39, page 12]. The pin is connected via a 0Ω bridge resistor to the clock input pin of the 2nd I2C interface and is pulled-up to the VIO_IN voltage rail supplied by the external regulator module discussed in Section 8.3.2. This pin is connected directly to the power system debug header. Note that although the pin is connected to the processor bus via a 0Ω bridge resistor in the schematic, this this resistor is not populated in this design. This pin can be left floating if unused [39, page 9].

The **I2C2_SDA_SDO** Pin is a bi-directional I2C data pin for the 2nd I2C interface bus [39, page 9]. The pin is I2C active-high and requires an external pull-up resistor [39, page 12]. The pin is connected via a 0Ω bridge resistor to the bi-directional data input pin of the 1st I2C interface and is pulled-up to the VIO_IN voltage rail supplied by the external regulator module discussed in Section 8.3.2. This pin is connected directly to the power system debug header. Note that although the pin is connected to the processor bus via a 0Ω bridge resistor in the schematic, this this resistor is not populated in this design. This pin can be left floating if unused [39, page 9].

The combined PMIC I2C pin configuration is shown in Table 5.4.

Name	**Active State**	**Default Internal Impedance**	**Configuration**	**Comment**
I2C_SCL1	I2C Active-High	External Pull-Up Required	Connected to External I2C bus, Disconnected from Processor	Populate R64, R67. Do Not Populate R69
I2C_SDA1	I2C Active-High	External Pull-Up Required	Connected to External I2C bus, Disconnected from Processor	Populate R62, R68. Do Not Populate R69
I2C_SCL2	I2C Active-High	External Pull-Up Required	Connected to External I2C bus, Disconnected from Processor	Populate R60, R67. Do Not Populate R69
I2C_SDA2	I2C Active-High	External Pull-Up Required	Connected to External I2C bus, Disconnected from Processor	Populate R65, R68. Do Not Populate 69

Table 5.4: PMIC I2C Pin Configuration

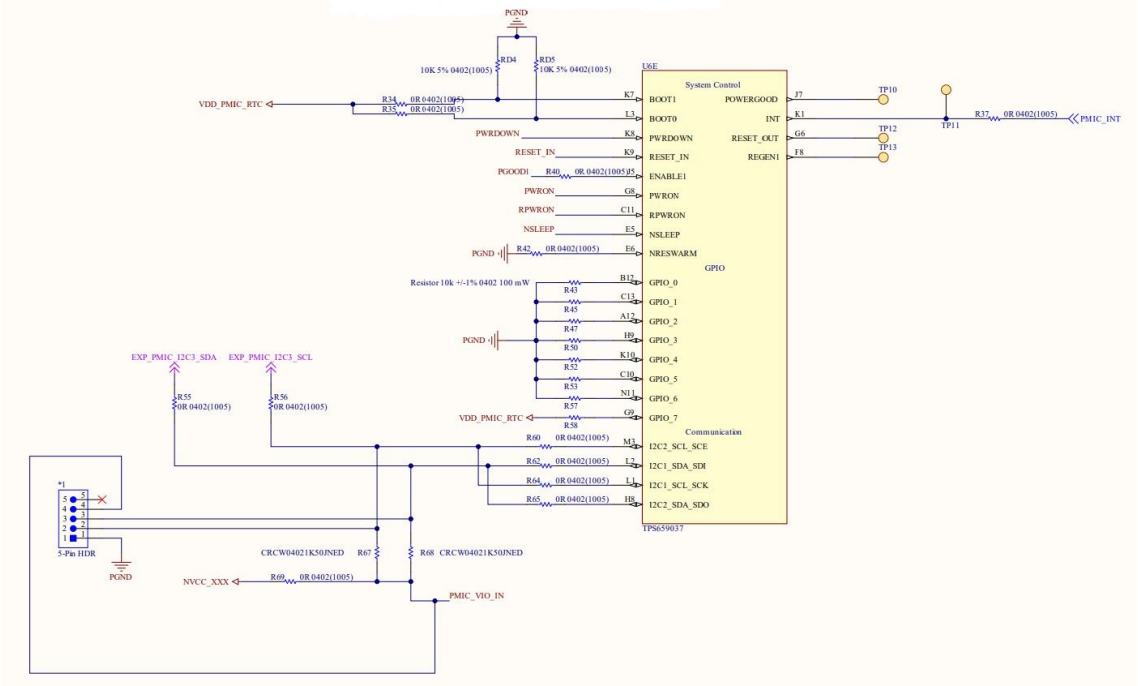


Figure 5.3: PMIC CONTROL

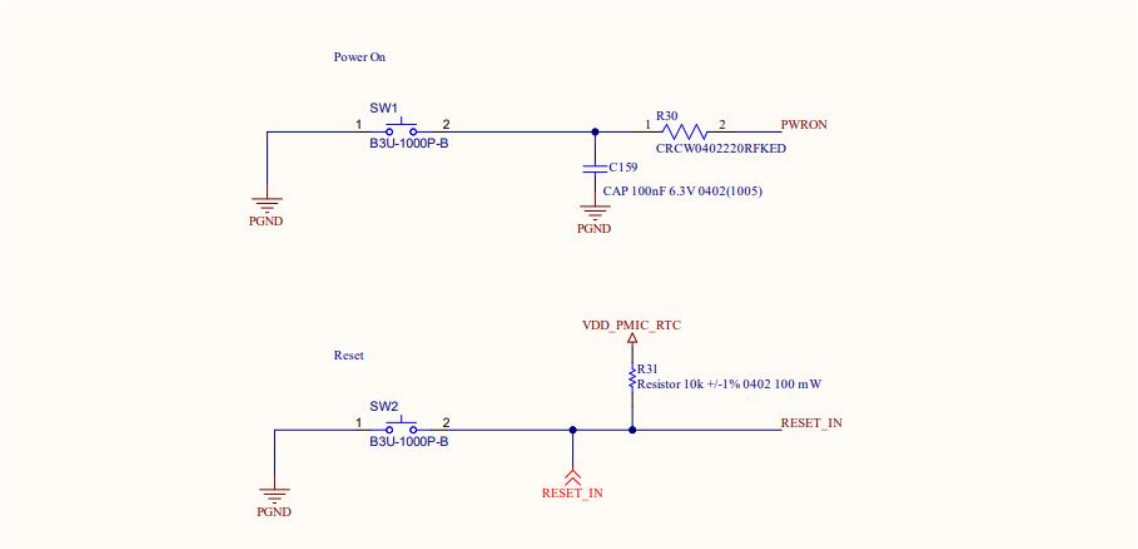


Figure 5.4: PMIC CONTROL Push Buttons

5.1.2.5 PMIC Control of Processor Reset

Reset Control is implemented through 2 DMP1045 P-Channel Enhancement Mode transistors to isolate voltage domains and is shown in Figure 5.5. PMIC_POR_B on the processor is in the NVCC_SNVS 3.3V voltage domain and has a 27$k\Omega$ internal pull-up resistor [56, page 90]. PMIC_RESET_B on the processor is in the NVCC_SNVS 3.3V voltage domain and has a 27$k\Omega$ pull-up resistor [56, page 90]. These transistors have internal resistances which must be considerably smaller than the internal resistances of the ports to function successfully. The DMP1045

has an on-resistance of about $29m\Omega$ [25, page 2]. Spice simulation predicts that the DMP1045 is capable of pulling the line as low as $734.418mV$ which is well within the 30% $990mV$ logical 0 of a $3.3V$ voltage level.

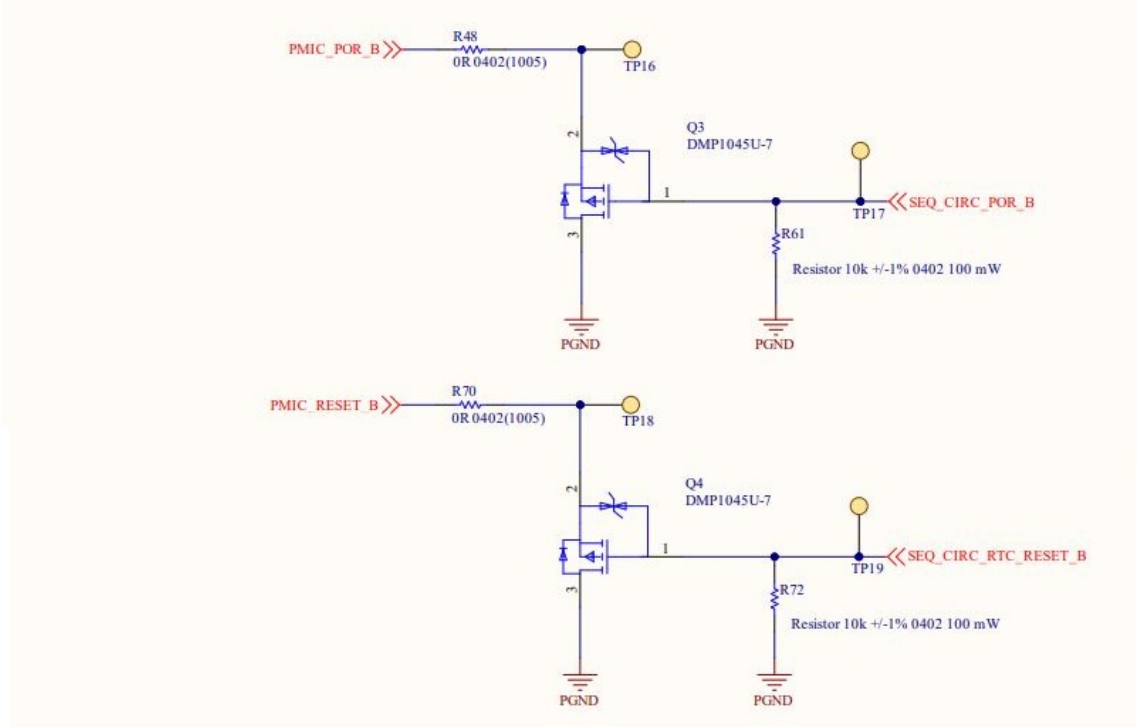


Figure 5.5: PMIC Reset Outputs, Processor Reset Inputs.

5.1.2.6 PMIC Switching Power Supplies

The **SMPS123** switching power supply supplies the VDD_0V9 voltage rail with $0.9V$ at $9A$. This rail powers inputs on the processor and external memory. The sequence position is 5. A current margin of $2380mA$ is planned.

A 3-phase configuration is extensively documented [39, page 34, 41], but Texas Instruments does not manufacture any components that have a compatible OTP that can be used in a 3-phase configuration [41, page 14, 15]. This problem is discussed in Section 8.3.1.

The resulting current margin is $620mA$ less than required for maximum performance. This sustained level of current delivery is seldom required and may cause thermal problems. The SMPS123 switching power supply implementation is shown in Figure 5.6.

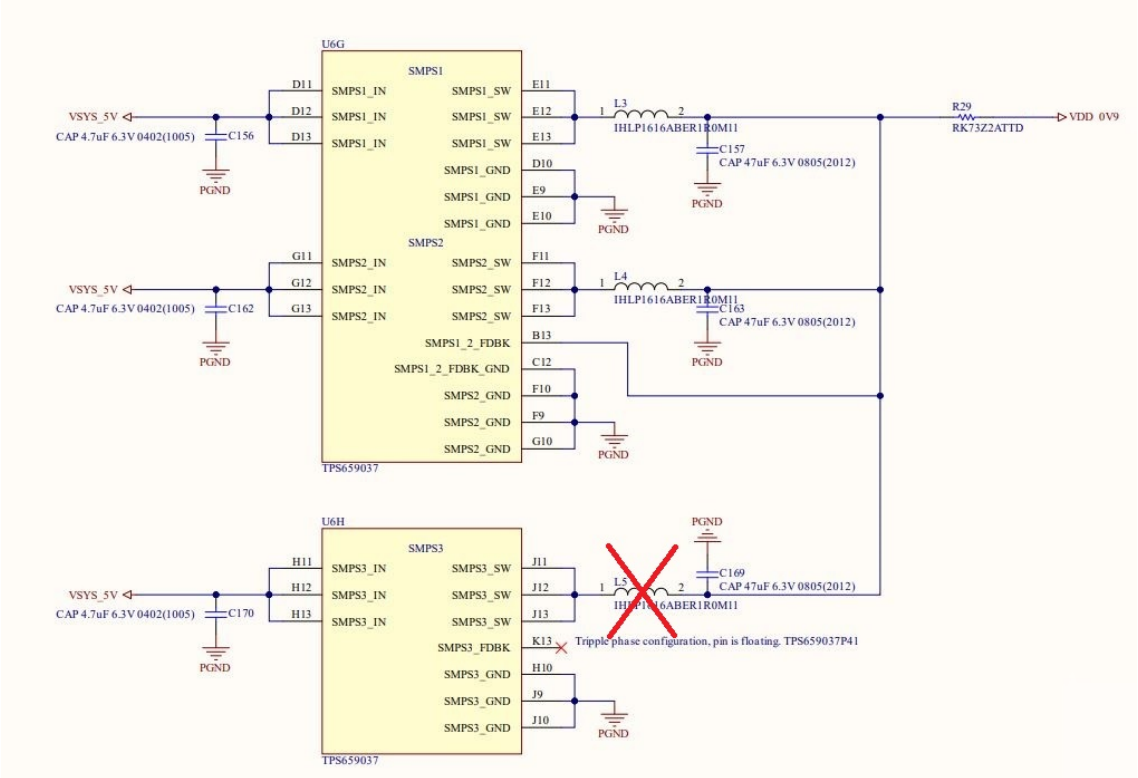


Figure 5.6: PMIC SMPS123 Switching Power Supply

The **SMPS45** switching power supply supplies the VDD_SOC voltage rail with $0.95V$ at $4A$. This rail powers inputs on the processor. The sequence position is 4. A current margin of $2380mA$ exists. The SMPS45 switching power supply implementation is shown in Figure 5.7.

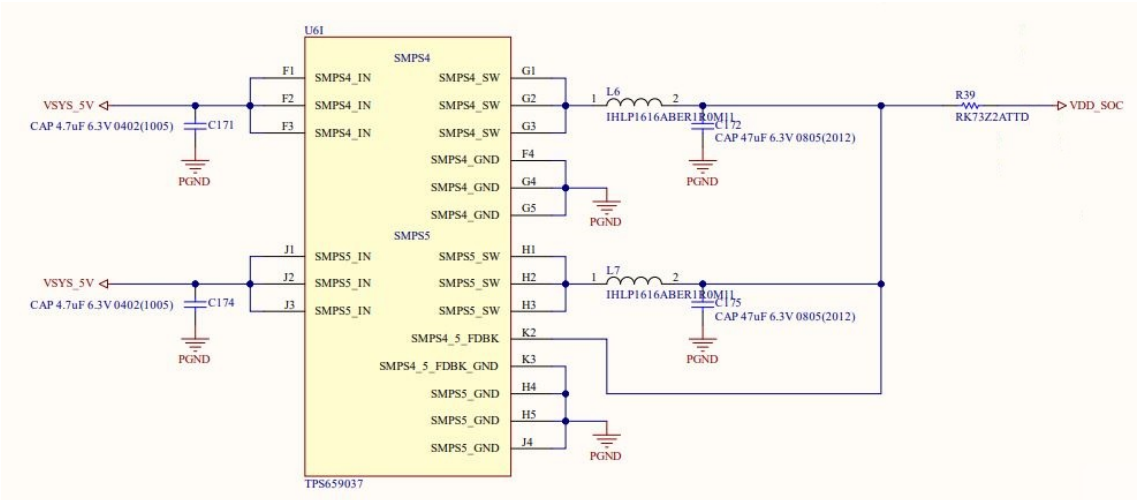


Figure 5.7: PMIC SMPS45 Switching Power Supply

The **SMPS6** switching power supply supplies the NVCC_DRAM voltage rail with $1.1V$ at $3A$. The sequence position is 7. This rail powers inputs on the processor and external memory.

A current margin of 955.87mA exists. The SMPS6 switching power supply implementation is shown in Figure 5.8.

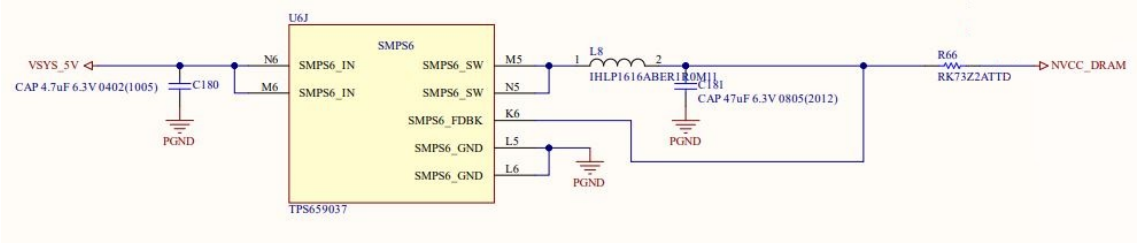


Figure 5.8: PMIC SMPS6 Switching Power Supply

The **SMPS7** switching power supply supplies the NVCC_XXX voltage rail with 3.3V at 2A. The sequence position is 7. This rail powers inputs on the processor and JTAG, SD CARD, UART, ETHERNET and GPIO pins. A current margin of 572mA exists. The SMPS7 switching power supply implementation is shown in Figure 5.9.

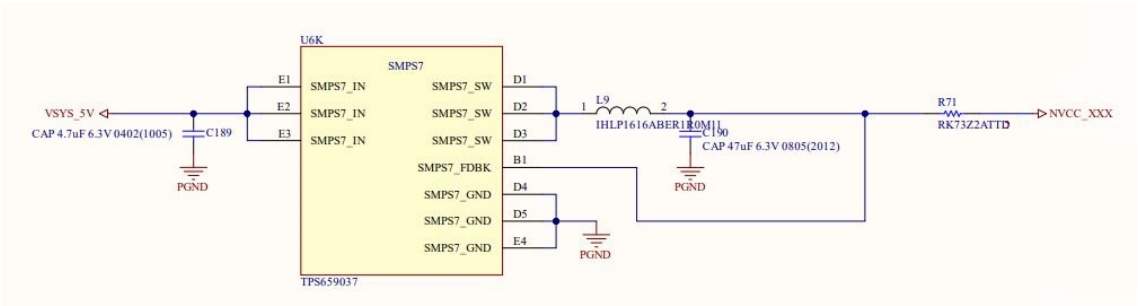


Figure 5.9: PMIC SMPS7 Switching Power Supply

The **SMPS8** switching power supply supplies the VDD_PHY_0V9 voltage rail with 0.9V at 1A. The sequence position is 8. This rail powers the processor's internal USB, PCIE and MIPI subsystems. A current margin of 867.8mA exists. The SMPS8 switching power supply implementation is shown in Figure 5.10.

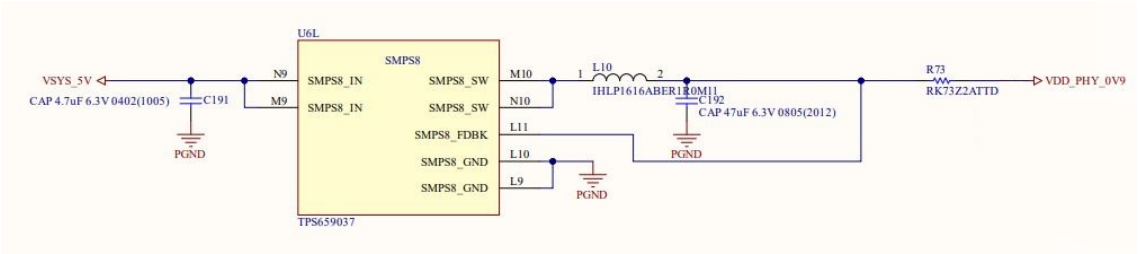


Figure 5.10: PMIC SMPS8 Switching Power Supply

The **SMPS9** switching power supply supplies the VDD_PHY_3V3 voltage rail with 3.3V at 1A. The sequence order is 8. This rail powers the processor's internal USB blocks. A current margin of 860.4mA exists. The SMPS9 switching power supply implementation is shown in Figure 5.11.

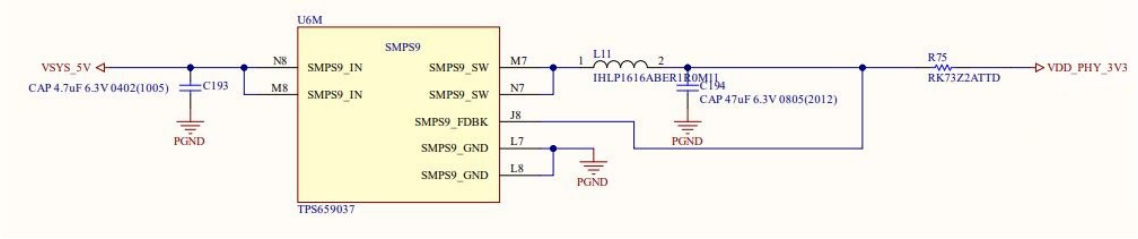


Figure 5.11: PMIC SMPS9 Switching Power Supply

5.1.2.7 PMIC Linear Drop Out Regulators

The **LDO1** low dropout regulator power supply supplies the VDDA_1P8 voltage rail with $1.8V$ at $300mA$. The sequence order is 6. This rail powers the external memory block, phase locked loop oscillator frequency generator and internal temperature sensor of the processor and the external memory.

The **LDO2** low dropout regulator power supply supplies the VDD_PHY_1V8 voltage rail with $1.8V$ at $300mA$. The sequence order is 8. This rail powers the PCIE, MIPI and HDMI internal blocks of the processor.

The **LDO3** low drop out regulator power supply supplies the NVCC_SNVS voltage rail with $3.3V$ at $200mA$. The Sequence order is 1. This supply powers the IO for the secure non-volatile storage block within the processor. It is required for successful reset.

The **LDO4** low drop out regulator power supply supplies the VDD_SNVS voltage rail with $0.9V$ at $200mA$. The sequence order is 2. This supply powers the secure non-volatile storage block within the processor. This block contains elements such as the real time clock and security components. Amongst others, it supplies reset logic and is required for successful reset of the i.MX8MQ.

The **LDO9** low drop out regulator power supply supplies the SEQ_CIRC_RTC_RESET_B transistor with $3.3V$ at $50mA$. The sequence order is 3. This supply is connected to the gate of the transistor responsible for RTC reset of the i.MX8MQ. The reset port on the processor has a $27k\Omega$ pull-up resistor. The processor is placed in the reset state when the voltage on this processor pin is pulled down to ground potential. DMP1045 transistor is of PMOS type and will attempt to pull the reset line down to ground potential if a voltage is not applied to its gate. The processor is placed in the reset state when LDO9 is turned off and placed in the active state when LDO9 is turned on. Refer to Table 5.1 for information on exactly when reset occurs in the power up-sequence.

The **LDOUSB** low drop out regulator power supply supplies the VDD_ANALOGUE_3V3 voltage rail with $3.3V$ at $100mA$. The sequence order is 8. This rail powers the analogue supply for the ADC subsystem.

The **LDOLN** low drop out power supply supplies the SEQ_CIRC_POR_B transistor with $3.3V$ at $50mA$. The sequence order is 9. This supply is connected to the gate of the transistor responsible for system reset on the i.MX8MQ. The reset port on the processor has a $27k\Omega$ pull up resistor. The processor is placed in the reset state when the voltage on this processor pin is pulled down to ground potential. DMP1045 transistor is of PMOS type and will attempt to

pull the reset line down to ground potential if a voltage is not applied to its gate. The processor is placed in the reset state when LDOLN is turned off and placed in the active state LDOLN is turned on. Refer to Table 5.1 for information on exactly when reset occurs in the power-up sequence.

Figure 5.12 shows the PMIC LDO implementation.

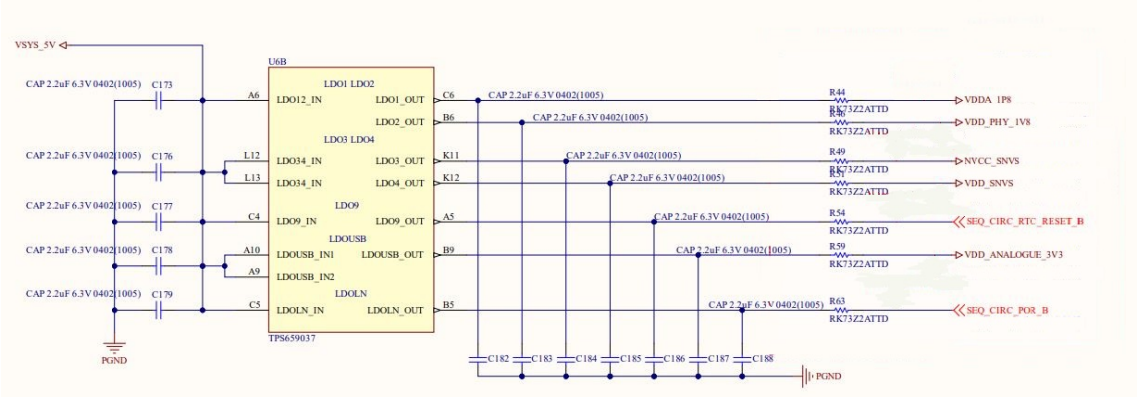


Figure 5.12: PMIC LDO Output

5.1.2.8 PMIC Analogue to Digital Converter

The 12-bit sigma delta ADC has 6 internal input channels and 3 external input channels. The 3 external input channels are unused and connected to ground as recommended in the datasheet. The ADC voltage reference line is unused and left floating as recommended in the datasheet [39, page 7]. The ADC is shown in Figure 5.13.

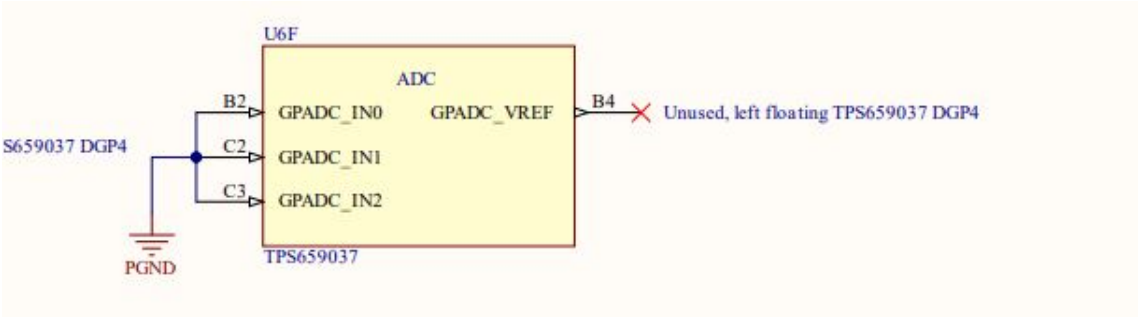


Figure 5.13: PMIC ADC

5.1.2.9 PMIC Crystal Oscillator

A FA-238 16.384MHz crystal is used as clock source for the TPS659037. This crystal is recommended by Texas Instruments. It has a load capacitance of 10pF, an equivalent series resistance (ESR) of 80Ω and an accuracy of 50ppm. 10pF Capacitors are recommended by Texas Instruments. The purpose of the crystal is primarily to generate the RTC for the TPS659037.

This RTC is used for the TPS659037 internal time keeping operations and to generate the 32.768kHz

CLK32KGO output. The CLK32KGO output is connected directly to the RTC input on the i.MX8MQ and is critical for successful reset of the i.MX8MQ processor. The $16.384MHz$ is divided down by 500 internally to produce a $32.768kHz$.

Note that the TPS659037 always requires a crystal to function although some members of this PMIC family do not [39, page 85] [40, page 192].

The TPS659037 datasheet indicates that the CLK32KGO is in the $3.3V$ V_IO voltage domain and that it is always present [39, page 10]. It is later discovered that this functionality is disabled in the OTP of all TPS659037 variants and that it cannot be re-enabled. A $1.8V$ CLK32KGO1V8 mux option is available on GPIO5, but the voltage level is too low for the i.MX8MQ. An external AKER $32.768kHz$ HCMOS oscillator is added to the Bokkiebord during the debugging phase as discussed Section 8.3.4. Texas Instruments has committed to fixing this error [12]. The crystal oscillator implementation is shown in Figure 5.14.

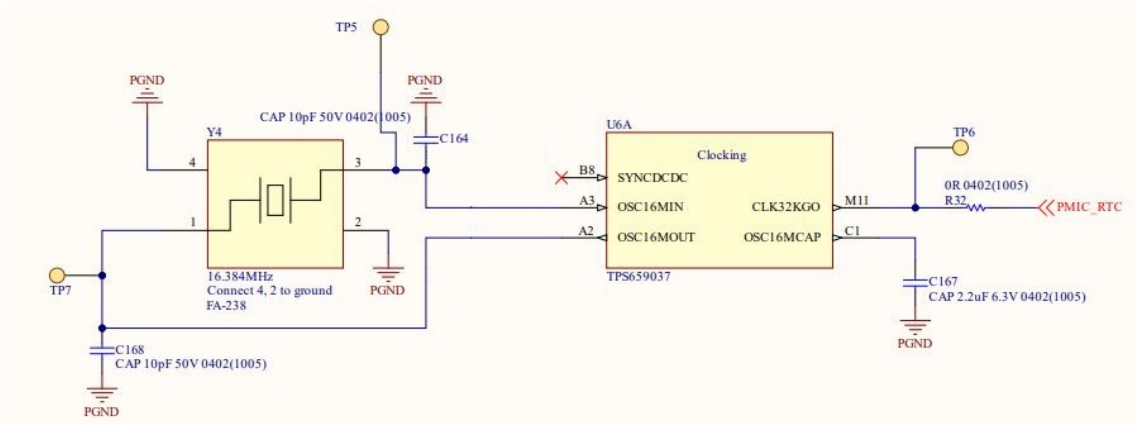


Figure 5.14: PMIC Crystal Oscillator

5.1.3 Supplementary Power Stage: 5V Load Switch

A TPS22918 load switch is used to distribute a $5V$ rail from the 1st power input stage to devices on the board requiring $5V$ power. The load switch can supply $5.5V$ at up to $2A$ [37, page 4].

These devices include the USB voltage detection pin on the processor, the USB power subsystem and VDD_PHY_5V power pins on the expansion headers. Power sequence order is controlled entirely by the PMIC and the load switch is activated when a positive voltage is applied to the enable pin of the load switch by the VDD_PHY_3V3 voltage rail produced by the PMIC.

The TPS22918 load switch has 2 pins that can be used to configure the switching characteristics of the load switch.

The CT pin is connected to a capacitor that controls the rise time of the load switch. The charging time of the capacitor is directly proportional to the rise time of the load switch output and is given by [37, Eq3, page 14]:

$$S_R \times 10^6 = 0.55 \times C_T \times 10^{12} + 30 \tag{5.24}$$

With:

$$\frac{V_{max}}{t_{on_{time}}} = \frac{1}{S_R} \tag{5.25}$$

Rewriting yields:

$$t_{on_{time}} = \frac{V_{max} \times (0.55 \times C_T \times 10^{12} + 30)}{10^6} \tag{5.26}$$

Choosing C_T as 100nF

$$t_{on_{time}} = \frac{5 \times (0.55 \times (100 \times 10^{-9}) \times 10^{12} + 30)}{10^6} \tag{5.27}$$

$$\Rightarrow t_{on_{time}} = 0.27515[s] \tag{5.28}$$

Where:
S_R is the slew rate of the output voltage in seconds per volt.
C_T is the capacitance on the C_T pin in farad.
$t_{on_{time}}$ is the 100% turn on time of the load switch in seconds.
V_{max} is the 100% on output voltage in volt.

Which is deemed acceptable as no other power rail is dependent upon VDD_PHY_5V. R_QOD pin is connected to an internal and external bleeder resistor that controls the fall time of the load switch. The discharge time of the capacitor is directly proportional to the value of the bleeder resistor and is given by [37, Eq1, page 13]:

$$R_{QOD} = R_{PD} + R_{EXT} \tag{5.29}$$

And [37, Eq2, page 13]:

$$V_{CAP} = V_{IN} \times e^{\frac{-t}{\tau}} \tag{5.30}$$

With:

$$\tau = R_{QOD} \times C_L \tag{5.31}$$

Rewriting yields:

$$V_{CAP} = V_{IN} \times e^{\frac{-t}{(R_{PD}+R_{EXT}) \times C_L}} \tag{5.32}$$

Choosing C_L as $100\mu F$, R_{EXT} as 0Ω and V_{CAP} as 10% of maximum input voltage:

$$\ln 0.1 = \frac{-t}{(24+0) \times 100 \times 10^{-6}} \tag{5.33}$$

$$\Rightarrow t = 5.52620[ns] \tag{5.34}$$

Which is deemed to be acceptable since the USB subsystem is only fed by $5V$.

Where:
R_{QOD} is the total output discharge resistance in ohm.
R_{PD} is the internal pull-down resistance in ohm and is always 24Ω.
R_{EXT} is the external resistance between the V_{OUT} and QOD pin in ohm.
V_{CAP} is the voltage across the C_T capacitor in volt.

t is the time since removal of power to the load switch enable pin in seconds.

The $5V$ load switch implementation is shown in Figure 5.15.

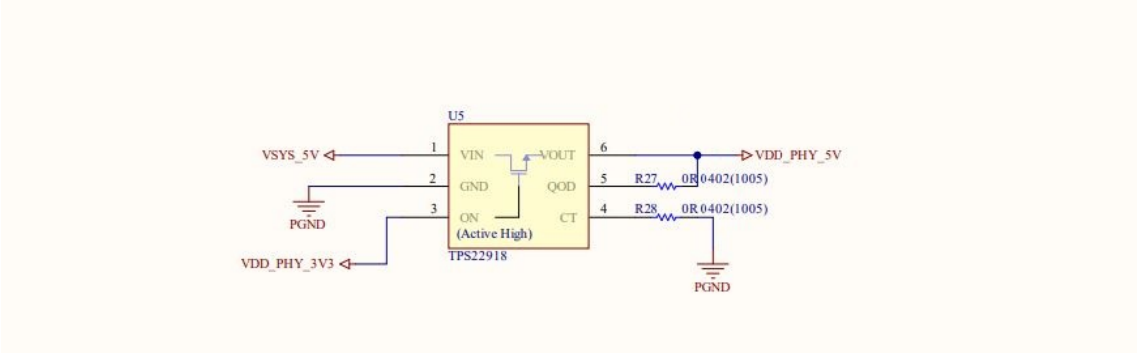


Figure 5.15: TPS22918 Load switch enabled by VDD_PHY_3V3 from SMPS9 .

5.1.4 Supplementary Power Stage: LPDDR4 Voltage Reference

DRAM_VREF is the LPDDR4 reference voltage. This reference voltage is connected directly to the processor and is exactly half of the $1.1V$ NVCC_DRAM LPDDR4 supply voltage which is $0.55V$. This voltage is obtained by means of a voltage divider consisting of 2 1% $10k\Omega$ resistors that are kept stable relative to the ground plane and the NVCC_DRAM rail by means of 2 $100nF$ capacitors placed on the input. This voltage is then buffered by means of a LMV951 op-amp that is capable of supplying up to $45mA$ of current [34, page 1]. The output of the op-amp is then further filtered by means of a $100nF$ and $10uF$ and 1 $100nF$ capacitor placed directly under the processor on the bottom side of the board. The maximum current draw of the processor on this input is expected to be $10mA$. A margin of $35mA$ therefore exists. The sequence order is the same as the DVCC_DRAM rail which is 7. The LPDDR4 Voltage Reference is shown in Figure 5.16.

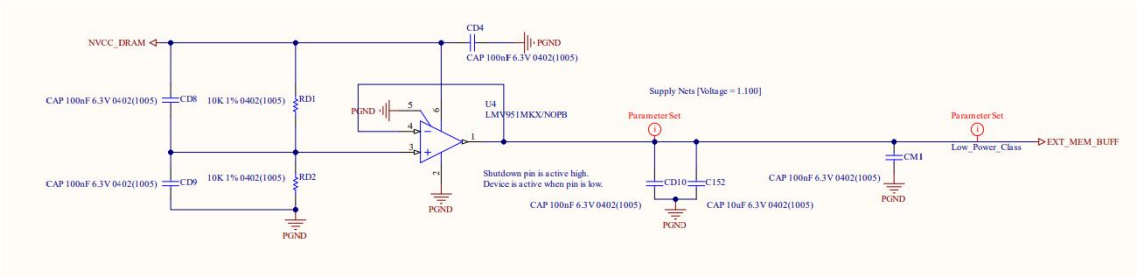


Figure 5.16: Buffered and decoupled voltage divider circuit .

5.2 Analogue to Digital Converter

The University of KU Leuven sets the requirement for a 12-bit ADC with a sampling rate of between $100kS/s$ and $3MS/s$. A 12-bit SAR ADC with a sampling rate of $3MS/s$ is implemented by means of a ADS7046 from Texas Instruments. It is selected for its fast sampling frequency, small X2QFN footprint and SPI Bus communication.

The ADS7046 requires 1 analogue and 1 digital power source. It is powered from $3.3V$ analogue and digital sources. It consumes $1.1mA$ on the analogue rail from the REF1933 voltage reference and $0.85mA$ on the digital NVCC_XXX rail. It has split analogue and digital grounds that are connected at a single point with 1 net tie to reduce noise.

The voltage reference is the REF1933 from Texas Instruments. Voltage is accurate to within $3ppm$. The voltage reference consumes a maximum of $460\mu A$ [36, page 5].

One-way communication with the processor is facilitated by means of an SPI bus. The ADC is connected on the 2nd SPI interface of the i.MX8MQ processor. The ECSPI2 interface of the i.MX8MQ is in the same $3.3V$ NVCC_XXX voltage domain as the ADC.

Maximum i.MX8MQ interface speed is $52Mbps$ [56, page 9]. The minimum clock period of the ADC is $16.66ns$ ($60.0240096MHz$). The conversion time of the ADC is 15 clock cycles [28] and new data is available on every 16th clock cycle.

For a sampling rate of 3MS/s, an interface speed of 48Mbps is required. As the maximum interface speed is 52Mbps, only 4Mbps remains available. A 2nd ADC can therefore not be added on the 2nd SPI interface bus and it is unlikely that ADC functionality could be expanded further without implementing a PCI interface which would add considerably more complexity to the design.

The processor is responsible for providing the clock.

Analogue input is buffered, filtered and if required clamped by means of 2 LMV710 op-amps. The LMV710 op-amp has a bandwidth of $5MHz$ at unity and can supply up to $40mA$ if required. The filter stage consists of a 2nd order Butterworth filter with $-3dB$ cutoff at $1.5MHz$ Nyquist frequency.

ADC input impedance can be controlled by means of resistors R131 and R132. If a 50Ω input impedance is required, both these footprints can be populated with 100Ω resistors.

External ADC input is facilitated by means of a micro sub miniature version A connector (uSMA) connector. A uSMA connector is a small co-axial connector that is common in the radio frequency (RF) world. This connector offers excellent signal integrity and is usually only used for very high input frequencies.

It is chosen for this design as it is a surface mount connector that can be placed anywhere on the board and because it is thought that the end user might have compatible cables due to this connector's common usage in RF applications.

This surface mount connector is required as the area just adjacent to the pins reserved for analogue input on the BeagleBone Black are highly constrained on the Bokkiebord which means that the analogue to digital conversion circuitry will not fit.

The usage of this connector also makes the analogue input of the Bokkiebord distinct from the BeagleBone Black as input voltages and impedances may differ.

The ADC subsystem implementation is shown in Figure 5.17. Figure 5.18 shows the expected

frequency response of the system.

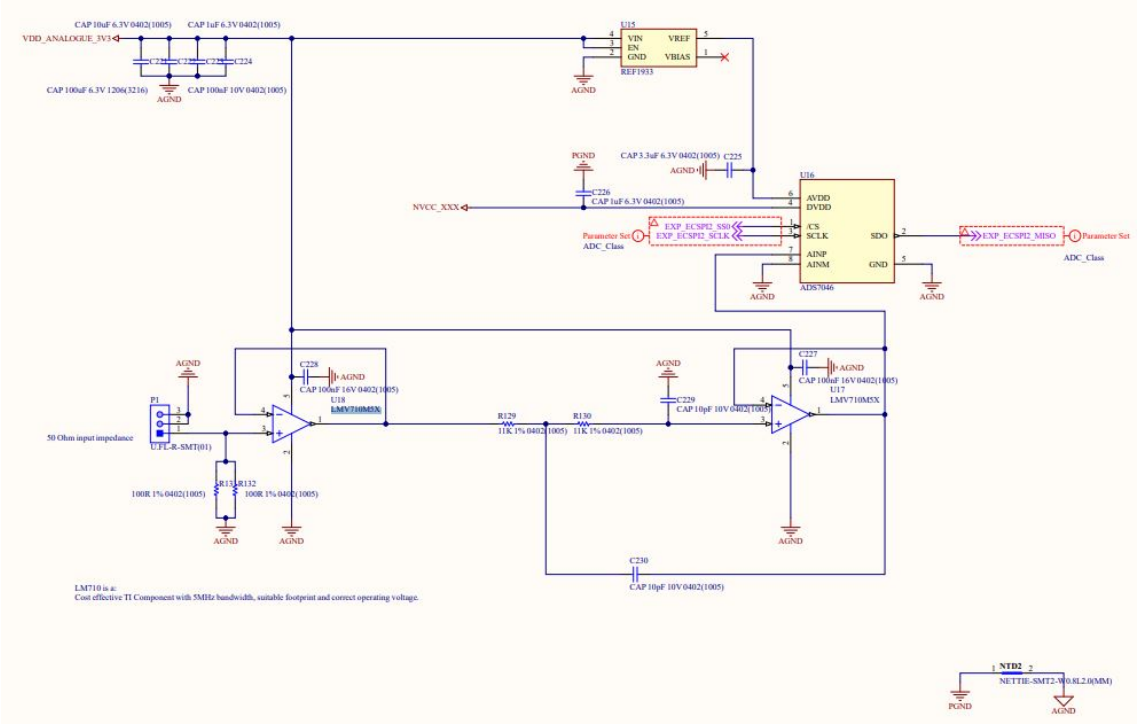


Figure 5.17: Analogue to digital converter circuit.

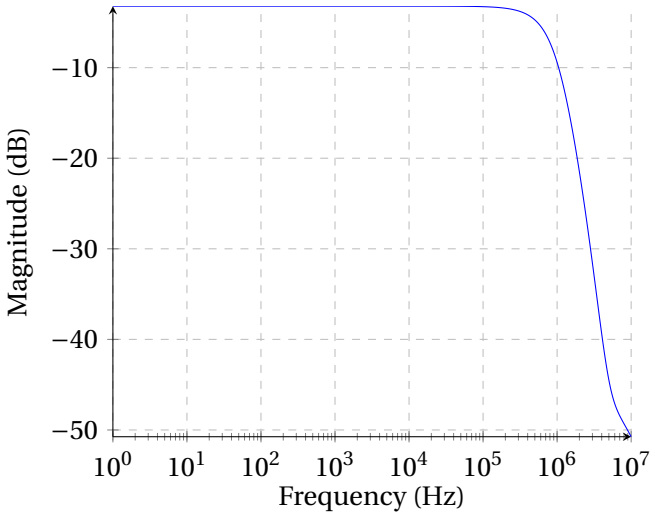


Figure 5.18: Expected Frequency Response of the ADC filtering Circuit.

5.3 Boot resistors

The i.MX8MQ processor can be configured to boot from different storage devices if the "BT_FUSE_SEL" fuse is not programmed. The "BT_FUSE_SEL" fuse is 0 when shipped from the factory. The boot configuration can be overriden when in this state by sampling the boot configuration resistors [57, Table 2-6, page 837]. Boot fuse override pins consist of 14 GPIO pins that can be pulled-up or down to set the configuration in addition to the 2 dedicated boot configuration resistors. The boot resistor configuration for the Bokkiebord is shown in Table 5.5.

Configuration Bit	Pin Name	Value	PU/PD	PCB Configuration	Explanation
BOOT_MODE_0	BOOT_MODE_0	0	PD	Populate R78, Do Not Populate R77	Boot From Fuses
BOOT_MODE_1	BOOT_MODE_1	0	PD	Populate R79, Do Not Populate R76	Boot From Fuses
BOOT_CFG_0	SAI1_RXD0	0	PD	Populate R95, Do Not Populate R80	Reserved
BOOT_CFG_1	SAI1_RXD1	1	PU	Populate R81, Do Not Populate R94	High Speed SD
BOOT_CFG_2	SAI1_RXD2	0	PD	Populate R93, Do Not Populate R82	High Speed SD
BOOT_CFG_3	SAI1_RXD3	0	PD	Populate R92, Do Not Populate R83	High Speed SD
BOOT_CFG_4	SAI1_RXD4	1	PU	Populate R84, Do Not Populate R91	4-Bit Bus Width
BOOT_CFG_5	SAI1_RXD5	0	PD	Populate R90, Do Not Populate R85	Reserved
BOOT_CFG_6	SAI1_RXD6	0	PD	Populate R89, Do Not Populate R86	Reserved
BOOT_CFG_7	SAI1_RXD7	0	PD	Populate R88, Do Not Populate R87	Regular Boot
BOOT_CFG_8	SAI1_TXD0	1	PU	Populate R96, Do Not Populate R111	SD Clock Loopback from Pad
BOOT_CFG_9	SAI1_TXD1	1	PU	Populate R97, Do Not Populate R110	No Power Cycle
BOOT_CFG_10	SAI1_TXD2	0	PD	Populate R109, Do Not Populate R98	Boot From SD Port 1
BOOT_CFG_11	SAI1_TXD3	0	PD	Populate R108, Do Not Populate R99	Boot From SD Port 1
BOOT_CFG_12	SAI1_TXD4	1	PU	Populate R100, Do Not Populate R107	Boot From SD Card
BOOT_CFG_13	SAI1_TXD5	0	PD	Populate R106, Do Not Populate R101	Boot From SD Card
BOOT_CFG_14	SAI1_TXD6	0	PD	Populate R105, Do Not Populate R102	Boot From SD Card
BOOT_CFG_15	SAI1_TXD7	1	PU	Populate R103, Do Not Populate R104	Disable Infinite Loop

Table 5.5: Bokkiebord Boot Resistor Configuration

5.4 Reduced Media Independent Interface Ethernet

The University of Stellenbosch and KU Leuven sets the requirement for an Ethernet interface without specifying the data rate.

The i.MX8MQ supports reduced media independent interface (RMII) and reduced gigabit media independent interface (RGMII) network interfaces [56, page 48]. 2 Ethernet PHYs are considered for this design. The 1st is the $100Mbps$ DP83825 from Texas Instruments and the 2nd is the $1000Mbps$ DP83867 from Texas Instruments.

The $100Mbps$ DP83825 Ethernet PHY is preferred over the $1000Mbps$ DP83867 [27] Ethernet PHY for its small size, because it only has a single $3.3V$ power input requirement and the fact that its driver is implemented in the Linux kernel [31].

The $1000Mbps$ DP83867 requires 3 additional power inputs of $2.5V$, $1.0V$ and $1.8V$. Of the additional input voltages 2 are uncommon and are not reused on different subsystems of the Bokkiebord. As power plane routing area is also already highly constrained, routing the additional power supplies from the PMIC to the DP83867 are challenging.

The $1000Mbps$ DP83867 is also a much larger package due to the higher pin count requirement of a $1000Mbps$ Ethernet PHY. This further constrains routing area.

The $1000Mbps$ DP83867 utilises a RGMII interface which requires 4 additional lines to be routed between the processor and the Ethernet PHY. These 4 additional lines require additional routing area for spacing to manage coupling and additional routing area to facilitate length matching. This additional routing area also has to be duplicated on the power plane to ensure a solid continuous reference plane below the lines.

The additional routing area requirements make it unlikely that the design could be implemented on only 6-layers. The much higher data rate of RGMII is also regarded to add too much design risk to a 1st iteration. The $100Mbps$ DP83825 Ethernet PHY is chosen over the $1000Mbps$ DP83867 Ethernet PHY for the 1st iteration of the Bokkiebord.

The $100Mbps$ DP83825 Ethernet PHY is supplied by 2 external power sources. The VDDIO input on the Ethernet PHY is supplied by the NVCC_XXX voltage rail. The sequence order of this rail is 7. The VDDA3V3 input on the Ethernet PHY is supplied by the VDD_PHY_3V3 voltage rail. The sequencing order of this voltage rail is 8. The power supplies of the Ethernet PHY are decoupled as indicated by the datasheet [30, page 94]. The Ethernet PHY requires that the VDDIO input must be ramped up to $3.3V$ before the VDDA3V3 voltage rail is ramped up to $3.3V$. If this is not feasible, the Ethernet PHY must be reset before operation.

Communication is facilitated by means of RMII with management data input output (MDIO) as the main administration interface. Clock is provided by the processor and additional reset and PWRDN lines exist. Both reset and powerdown lines are active low. A reset pulse must be at least $25\mu s$ in duration.

The RMII and MDIO lines of the Ethernet PHY are connected directly to the Ethernet interface on the processor. The voltage domain of the Ethernet interface on the processor is also NVCC_XXX. The reset and powerdown lines on the Ethernet PHY are connected to the GPIO

interface on the processor. The power domain of the GPIO interface is NVCC_GPIO1 which is also connected to the NVCC_XXX voltage rail. As the digital parts of both chips are powered up concurrently the risk of latch-up damage on the channel is mitigated.

The lines connected to the Ethernet Transformer require a differential impedance of 100Ω and all other lines require a 50Ω routing impedance.

A RJ45 Jack with integrated magnetics is chosen to comply with the Ethernet specification.

The Ethernet PHY implementation is shown in Figure 5.19. Refer to Section 9.14 for software implementation.

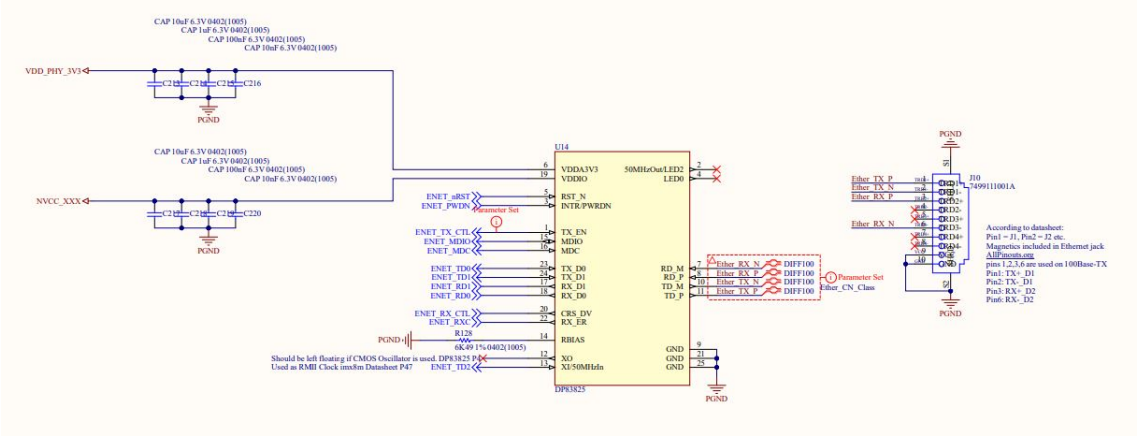


Figure 5.19: Ethernet PHY Circuit

5.5 General Purpose Input Output

The expansion header interface consists of 2 sets of 46 pins organised in 2 rows each. The pin pitch of these interfaces is $100mil$or $2.54mm$ apart. The expansion header interface is based primarily on that of the BeagleBone Black and although the interfaces are mostly identical, but some differences do exist.

Most pins on the Bokkiebord are connected directly to the processor and their function is determined by the configuration of the device tree. Refer to Chapter 9 for more information on this topic. Some pins are connected to the power system and may be connected to always-on sources. Special care must be taken to avoid latch-up damage.

Analogue input pins used on the BeagleBone Black are unused on the Bokkiebord and the ADC functionality is moved to the uSMA connector at the ADC.

Some pins are shared with boot configuration resistors.

Figure 5.20 shows the schematic implementation of the GPIO header interface.

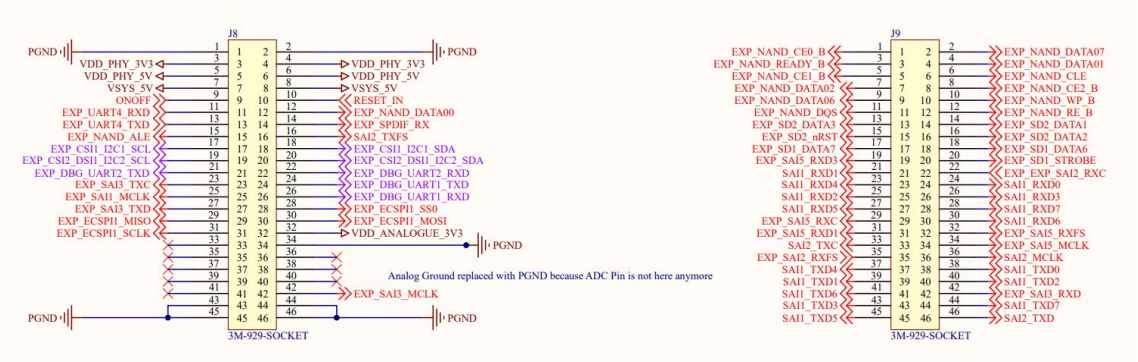


Figure 5.20: GPIO Pins

Table 5.6 shows the standard BeagleBone Black pinout and Table 5.7 shows the standard Bokkiebord pinout.

Left Connector					Right Connector			
DGND	1	2	DGND		DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3		MMC1_DAT6	3	4	MMC1_DAT7
VDD_5V	5	6	VDD_5V		MMC1_DAT2	5	6	MMC1_DAT3
VSYS_5V	7	8	SYS_5V		GPIO_66	7	8	GPIO_67
PWR_BUT	9	10	SYS_RESETN		GPIO_69	9	10	GPIO68
UART4_RXD	11	12	GPIO_60		GPIO45	11	12	GPIO_44
UART4_TXD	13	14	EHRPWM1A		EHRPWM2B	13	14	GPIO_26
GPIO_48	15	16	EHRPWM1B		GPIO_47	15	16	GPIO_46
SPI0_CS0	17	18	SPI0_D1		GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA		EHRPWM2A	19	20	MMC1_CMD
SPI0_D0	21	22	SPI0_CLK		MMC1_CLK	21	22	MMC1_DAT5
GPIO_49	23	24	UART1_TXD		MMC1_DAT4	23	24	MMC1_DAT1
GPIO_117	25	26	UART1_RXD		MMC1_DAT0	25	26	GPIO_61
GPIO_115	27	28	SPI1_CS0		LCD_VSYNC	27	28	LCD_PCLK
SPI1_D0	29	30	GPIO_112		LCD_HSYNC	29	30	LCD_AD_BIAS
SPI1_SCLK	31	32	VDD_ADC		LCD_DATA14	31	32	LCD_DATA15
AIN4	33	34	GNDA_ADC		LCD_DATA13	33	34	LCD_DATA11
AIN6	35	36	AIN5		LCD_DATA12	35	36	LCD_DATA10
AIN2	37	38	AIN3		LCD_DATA8	37	38	LCD_DATA9
AIN0	39	40	AIN1		LCD_DATA6	39	40	LCD_DATA7
GPIO_20	41	42	EXAPPWM0		LCD_DATA4	41	42	LCD_DATA5
DGND	43	44	DGND		LCD_DATA2	43	44	LCD_DATA3
DGND	45	46	DGND		LCD_DATA0	45	46	LCD_DATA1

Table 5.6: BeagleBone Black Expansion Header Interface Pinout

Left Connector					Right Connector			
PGND	1	2	PGND		NAND_CE0_B	1	2	NAND_DATA7
VDD _PHY _3V3	3	4	VDD _PHY _3V3		NAND_ READY_ B	3	4	NAND_DATA1
VDD_PHY_5V	5	6	VDD_PHY_5V		NAND_CE1_B	5	6	NAND_CLE
VSYS_5V	7	8	VSYS_5V		NAND_DATA2	7	8	NAND_CE2_B
ONOFF	9	10	RESET_IN		NAND_DATA6	9	10	NAND_WP_B
UART_RXD	11	12	NAND_DATA0		NAND_DQS	11	12	NAND_RE_B
UART_TXD	13	14	SPDIF_RX		SD2_DATA3	13	14	SD2_DATA1
NAND_ALE	15	16	SAI2_TXFS		SD2_nRST	15	16	SD2_DATA2
I2C1_SCL	17	18	I2C1_SDA		SD1_DATA7	17	18	SD1_DATA6
I2C2_SCL	19	20	I2C2_SDA		SAI5_RXD3	19	20	SD1_STROBE
UART2_TXD	21	22	UART2_RXD		SAI1_RXD1	21	22	SAI2_RXC
SAI3_TXC	23	24	UART1_TXD		SAI1_RXD4	23	24	SAI1_RXD0
SAI_MCLK	25	26	UART1_RXD		SAI1_RXD2	25	26	SAI1_RXD3
SAI3_TXD	27	28	ECSPI1_SS0		SAI1_RXD5	27	28	SAI1_RXD7
ECSPI1_MISO	29	30	ECSPI1_MOSI		SAI5_RXC	29	30	SAI1_RXD6
ECSPI1_SCLK	31	32	VDD_ ANA-LOGUE_ 3V3		SAI5_RXD1	31	32	SAI5_RXFS
NC (Safety)	33	34	PGND		SAI2_TXC	33	34	SAI5_MCLK
NC (Safety)	35	36	NC (Safety)		SAI2_RXFS	35	36	SAI2_MCLK
NC (Safety)	37	38	NC (Safety)		SAI1_TXD4	37	38	SAI1_TXD0
NC (Safety)	39	40	NC (Safety)		SAI1_TXD1	39	40	SAI1_TXD2
NC (Safety)	41	42	SAI3_MCLK		SAI1_TXD6	41	42	SAI3_RXD
PGND	43	44	PGND		SAI1_TXD3	43	44	SAI1_TXD7
PGND	45	46	PGND		SAI1_TXD5	45	46	SAI2_TXD

Table 5.7: Bokkiebord Expansion Header Interface Pinout

5.6 High Definition Multimedia Interface

The i.MX8MQ processor supports the High Definition Multimedia Interface (HDMI) 2.0, Displayport 1.3 and Displayport 1.4 standards.

The HDMI Interface consists of 4 pairs Transmission Minimised Differential Signaling (TMDS) Lines [93]. The Bokkiebord is the sole driving source on these lines, and is protected from ESD and electromagnetic interference (EMI) by means of the PCMF3HDMI2S and PCMF2HDMI2S.

The Display Data Channel (DDC) link requires pull-up resistors to $5V$ with a pull-up resistance of between $1.5k\Omega$ and $2k\Omega$ [56, page 61]. The DDC link is a protocol [91] [89] based on I2C that is primarily used to negotiate display types and available resolution [90]. It can also be used for High Bandwidth Digital Content Protection (HDCP) [92]. This implementation of I2C does not use standard I2C addressing and the host device is master while the display device is slave. The data rate is usually $100kbps$ but can be increased to $400kbps$.

The Consumer Electronics Control (CEC) line requires a $27kOhm$ pull-up resistor to $3.3V$. The

CEC is a 1 wire bi-directional bus that is used to control multiple devices from a single remote control device. This line is kept separate from other signals and although the wiring is required, the implementation of the function is optional. The bus is an open collector line similar to I2C that is pulled low to transmit a bit. The data rate is $417bps$ [88].

The HDMI 2.1 specification requires that output voltage of the power line must be between $4.8V$ and $5.3V$ and that short circuit current must be no more than $0.5A$. Shorting any wire pair on the connector must not result in any damage [33]. F2 is added to the $5V$ supply of this design to meet this requirement. It is a PICOSMDC035S $100mA$ resetable fuse from LittleFuse, trip time is $200ms$ [18]. Current draw may be $55mA$. Note that D1 and D2 prevent current flowing into the Bokkiebord power system from the display device.

HDMI_AUX_P is used as Utility/HEAC+ line while HDMI_AUX_N is used as HPD/HEAC- line. Note that the HPD/HEAC- line is also connected to the hot plug detect (HPD) port on the processor in addition to being connected to the HDMI_AUX_N line. A pull down resistor of $1M\Omega$ is recommended on the HPD line for pull-down resistance. This value is large because a display device pulls this line up to $5V$ through a $1k\Omega$ resistor and the majority of the voltage of this voltage divider should fall over the HPD pin.

The i.MX8MQ datasheet [56, Table 26, page 31] shows the operating ranges for the external $27MHz$ Reference Clock with maximum allowable voltage of HDMI_AVDDIO. The voltage domain of the HDMI clock input port is HDMI_AVDDIO and is connected to the VDD_PHY_1V8 voltage rail with sequence order 8 while the oscillator is connected to the NVCC_XXX $3.3V$ voltage rail.

A XUN0536027 $27MHz$ HCSL oscillator from Renesas is used as reference clock for the HDMI. A NX5427001Z from Diodes Incorporated can be used if this component is unavailable. When fed from $3.3V$ the HCSL XUN0536027 Oscillator has a high voltage level of between $0.6V$ and $1.1V$ and a low voltage level of between $0V$ and $0.2V$.

A 499Ω 1% resistor is required as reference impedance on the processor's HDMI_REXT port.

Figure 5.21 shows HDMI EMI and ESD protection, Figure 5.22 shows HDMI connector implementation. Figure 5.23 shows the external $27MHz$ HCSL Oscillator.

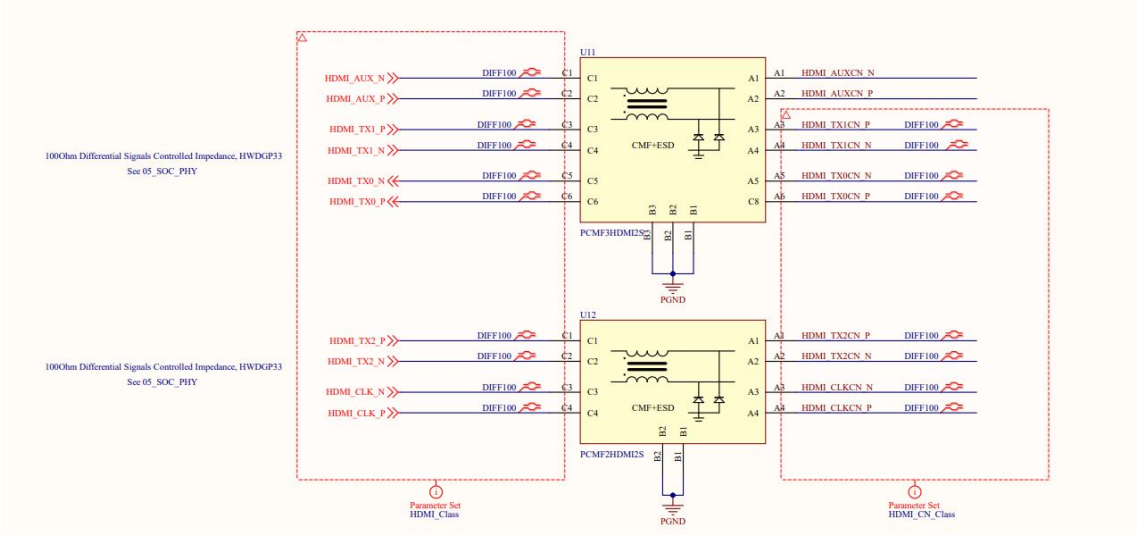


Figure 5.21: HDMI EMI Filtering and ESD Protection

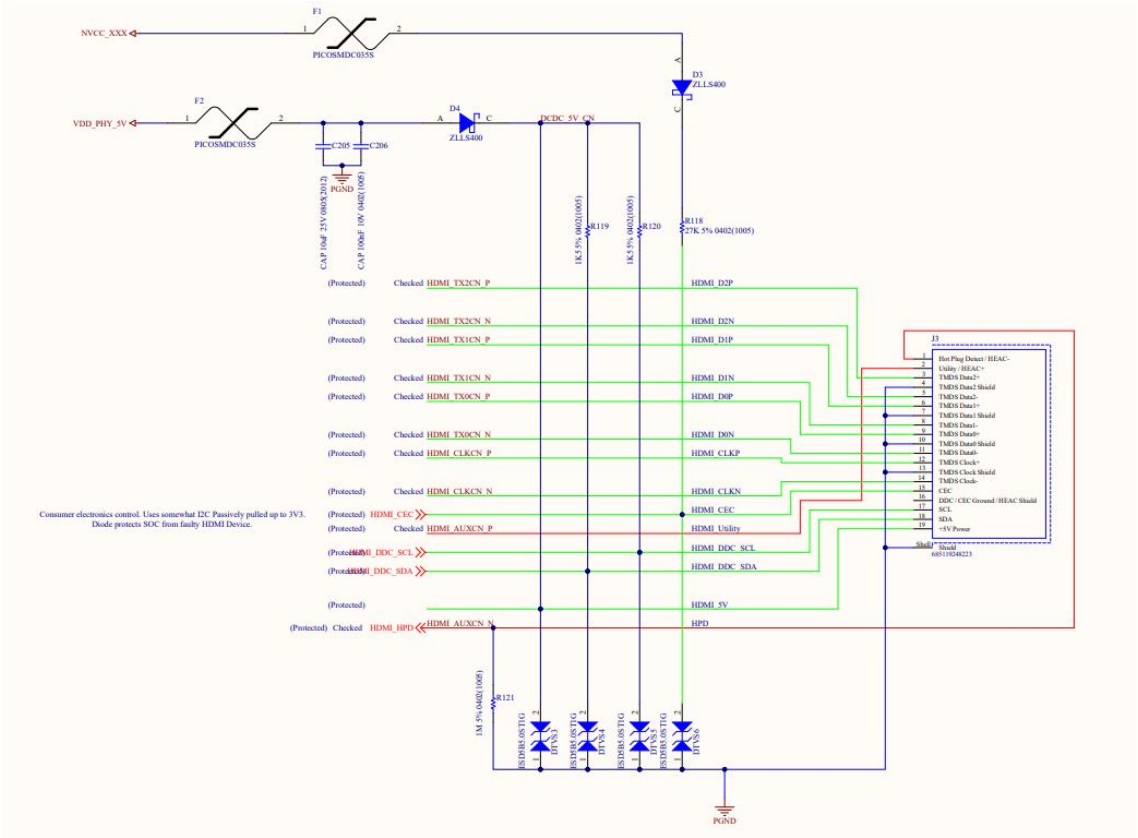


Figure 5.22: HDMI Connector Implementation

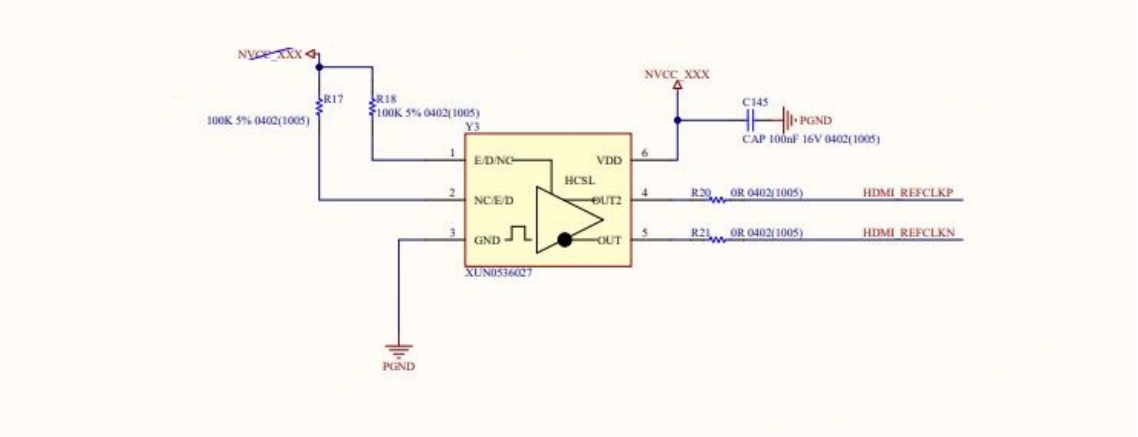


Figure 5.23: HDMI HCSL 27MHz Oscillator

5.7 Joint Test Action Group Debug Port

The Joint Test Action Group Debug Port (JTAG) Specification is standardised as the Institute of Electrical and Electronic Engineers (IEEE) 1149.1 standard. The standard pins are Test Data In (TDI), Test Data Out (TDO), Test Clock (TCK), Test Mode Select (TMS) and Test Reset (TRST). Test Reset is an optional pin and its function differs based on how it is implemented.

The hardware development guide recommends an external 10$k\Omega$ pull-down resistor for the Test Clock input (JTAG_TCK) to the processor. JTAG_TMS and JTAG_TDI are pulled up internally to the NVCC_JTAG domain via a 27$k\Omega$ resistor. The hardware development guide recommends that the optional TRST pin not be connected to the debugger directly but, recommends that the system reset be used instead. As per the standard the TRST pin is active low.

A Segger JLINK EDU Debugger is used as debugging device. It is primarily used to identify the state of the processor and to reset it. It can also be useful when supervising the memory initialisation procedure. The JTAG functions reliably at 50MHz and could run up to 100MHz, but this speed is not supported by the JLINK EDU debugger.

Note that memory will have to be initialised by means of the operating system or by loading firmware via the OTG port before external memory can be read via the JTAG port. The JTAG cannot initialise memory as it cannot write to the registers in question directly. Figure 5.24 shows the JTAG implementation.

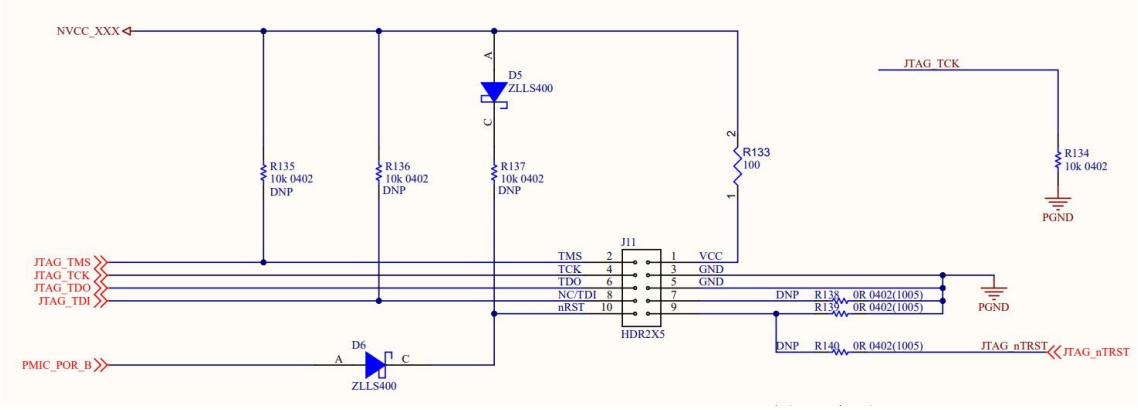


Figure 5.24: JTAG Port Implementation

5.8 Low Power Double Data Rate Version 4 Memory

$2GB$ of memory is required as per the University of Stellenbosch specification. $2GB$ of LPDDR4 memory is implemented by means of a Micron MT53E512M32D2NP-046 WT:E LPDDR4 memory module. LPDDR4 is chosen as it only requires 1 component to be implemented and can be powered from only 2 voltage rails.

In total the memory consumes up to $58.3mA$ on the $1.8V$ VDD1 input that is connected to the VDDA_1P8 voltage rail. The memory consumes up to $1162mA$ on the $1.1V$ VDD2 input that is connected to the NVCC_DRAM voltage rail. The memory consumes up to $86.55mA$ on the $1.1V$ VDDQ input that is also connected to the NVCC_DRAM voltage rail.

The processor receives a reference voltage from the the LPDDR4 voltage reference source discussed in Section 5.1.4.

The memory runs at up to $2133MHz$ ($4266MT/s$) with a cycle time of $468ps$. The MT53E512M32D2NP-046 WT:E LPDDR4 consists internally of 2 $8Gb$ dies that are contained in a 200 ball WFBGA package.

Each of the 2 dies can operate independently and a combined 32-bit memory interface is implemented through 2 16-bit channels.

Each die has its own CS, CKE, CK, DMI, DQS and ODT lines. Each die has 6 CA and 16 DQ lines. The Memory Reset Line is shared between dies [83, page 19].

Each of these dies are internally addressed by 16 row address bits and 10 column address bits. Each bit of the 16-bit data bus bits is represented over $(2^{16}) \times (2^{10}) = 67108864$ memory locations per bank. The total 16-bit wide memory density is thus $67108864 \times 16 = 1073741824$ per bank. A die contains 8 banks which are addressed by 3 bank address bits which means that the total bit density of a die is $1073741824 \times 8 = 8589934592$ bits [83, page 31].

The **CS** line is a single ended Chip Select that is used to select between memory ranks (dies) on the same channel.

The **CKE** line is a single ended Clock Enable line that deactivates internal clock signals, in-

put buffers and output drivers. Power saving mode states are controlled via clock enable lines.

The **CK** line is a differential memory clock line. Address, command and control signals are sampled on CK_t rising edge and CK_c falling edge.

The **DMI** bus is a single ended Data Bus Intervention line. This bus is driven high when data on data lines is inverted and low when data on data lines is in normal state.

The **DQS** bus is a differential Data Strobe. Data strobe is edge aligned with data.

The **ODT_CA** line is a single ended On Die Termination control signal used for controlling termination resistance for CK, CS and CA lines.

The **CA** bus is a single ended command/address bus and is used to send commands and addresses from the processor to the memory. Refer to [83, Table 94, Page 65] for more information.

The **DQ** Bus is a single ended bus used to transmit data bi-directionally between the memory and processor.

Signals are routed point to point between the processor and memory. As the exact type of available memory device is uncertain at the time of board design, the decision is made to route all chip select lines between the processor and memory footprints.

The physical design therefore supports higher memory sizes, but only 2 of the 4 available chip select lines are used in the current configuration. Memory size can easily be upgraded by replacing the memory module and updating the software.

Figure 5.25 shows the schematic implementation of the memory. High-speed routing is discussed in Chapter 6. Memory initialisation is discussed in Chapter 8. Software configuration and utilisation of the memory is discussed in Chapter 9.

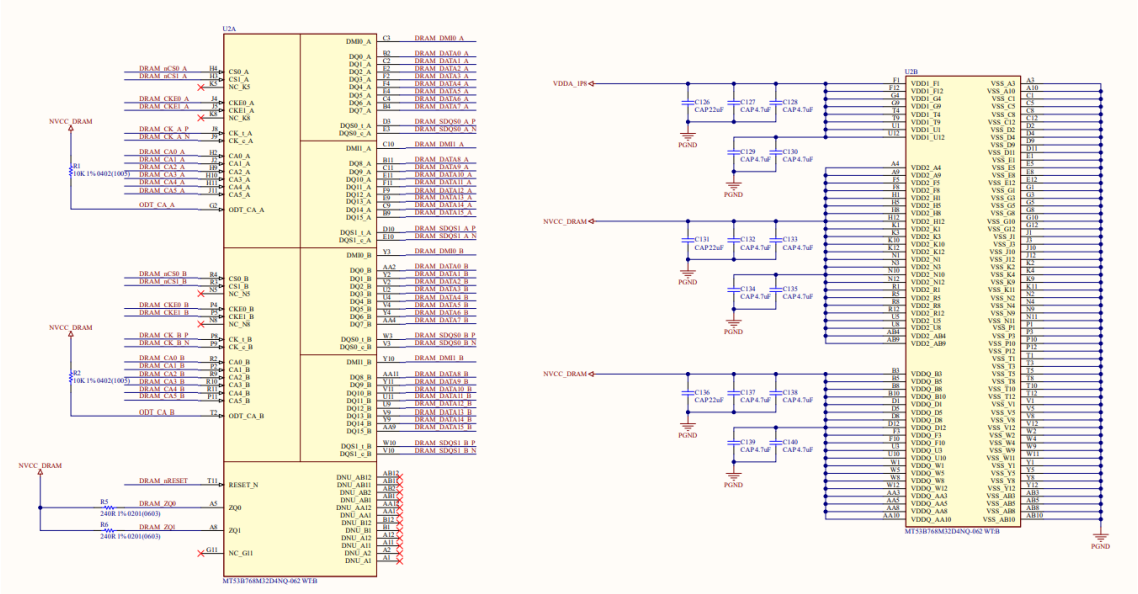


Figure 5.25: Memory Interface

5.9 Mobile Industry Processor Interface

The i.MX8MQ has 2 MIPI Camera Serial Interfaces (CSI) and 1 MIPI Display Serial Interface (DSI).

The Display interface functions independently from the HDMI interface. These interfaces are implemented to be electrically compatible with the Raspberry Pi interface [16, page 3].

Raspberry Pi compatible 22-pin connectors are placed for routing efficiency. A Camera interface is placed on the bottom of the board with a horizontal insertion angle and 1 camera interface is placed on the top of the board with a vertical insertion angle. The display interface is placed on the top of the board with a vertical insertion angle.

Both camera interfaces and the display interface have 4 differential line pairs and 1 differential clock. Power is connected to the VDD_PHY_3V3 rail. The maximum allowable current draw by MIPI devices is 860.4mA. All these MIPI devices share the I2C2 bus. Camera interface 1 has GPIO1_00 and GPIO1_01 GPIO lines while camera interface 2 has GPIO1_02 and GPIO_03 gpio lines.

In high speed mode output differential voltage is 200mV. Output impedance is typically 50Ω. Signal rise time is at least 160ps. In low speed mode output voltage is 1.1V, Output impedance is more than 110Ω and signal rise time is 25ns. Maximum load capacitance is 70pF. [56, page 62] Figure 5.26 shows the MIPI connector implementation.

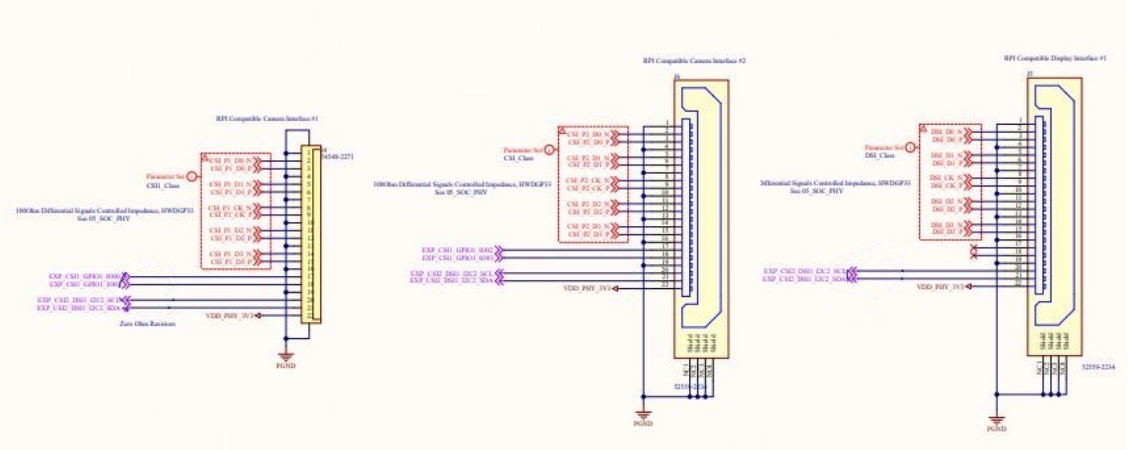


Figure 5.26: MIPI Interface Implementation

5.10 Reset and Control

A 25MHz and 27MHz crystal are used in a phase locked loop (PLL) configuration to generate the required i.MX8MQ frequencies. The hardware design guide indicates that a crystal with 100Ω ESR, a load capacitance of 8pF and a maximum drive level of 200μW is acceptable.

The 830108212509 quartz crystal from Wurth Electronic is selected as the 25MHz crystal. It is 2.0mm x 1.6mm in size. It has a typical Equivalent series resistance (ESR) of 100Ω, a load capacitance of 8pF and a maximum drive level of 100μW. It has an accuracy of 10ppm and typically ages at about 3ppm per year.

The R2016-27.000-8-F-1010-EXT-TR quartz crystal from Raltron Electronics is selected as 27MHz Crystal. It is 1.6mm x 2mm in size. It has a typical maximum ESR of 100Ω, a load capacitance of 8pF and a maximum drive level of 100μW. It has an accuracy of 10ppm and typically ages at about 2ppm per year.

The crystal oscillators are implemented in Colpitts configuration as suggested in the crystal oscillator troubleshooting guide [53]. The value of the 2 external capacitors is calculated as follows:

$$\frac{C_1 \times C_2}{C_1 + C_2} \approx C_L \tag{5.35}$$

If the value of the capacitors are the same: $C_1 = C_2 = C_C$

$$\frac{{C_C}^2}{2 \times C_C} \approx C_L \tag{5.36}$$

$$\frac{C_C}{2} \approx C_L \tag{5.37}$$

If CL is 8pF:

$$2 \times C_L = C_C = 16[pF] \tag{5.38}$$

Where:
C_1 is the value of the 1st capacitor in farad.

C_2 is the value of the 2nd capacitor in farad.
C_C is the value of the both capacitor in farad if their value is the same.
C_L is the load capacitance of the crystal in farad.

The crystals described above are connected to the i.MX8MQ processor as shown in Figure 5.27. The On/Off power switch shown in figure 5.28 is also connected to this processor block.

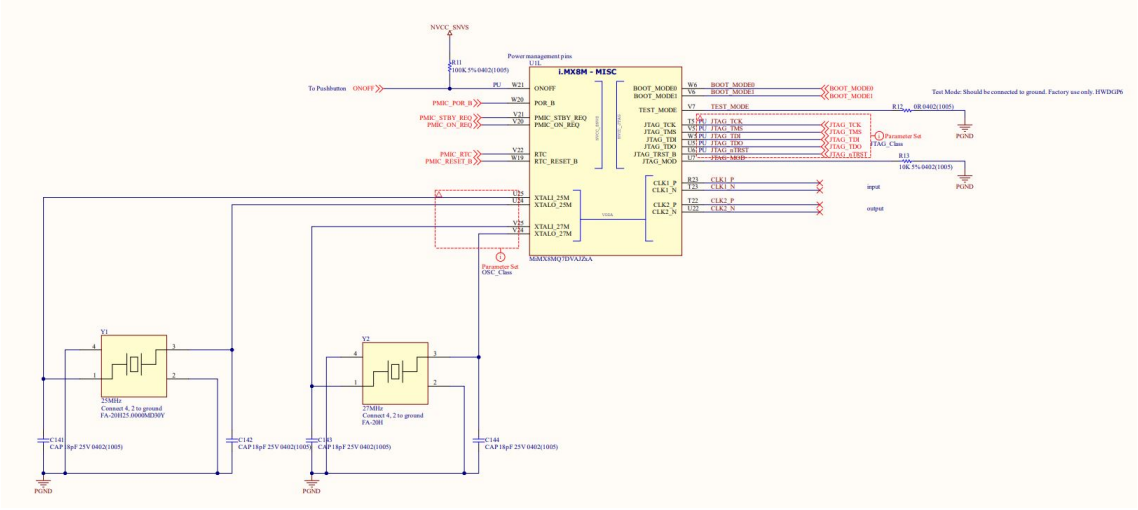


Figure 5.27: Low Level Reset and Control of the i.MX8MQ

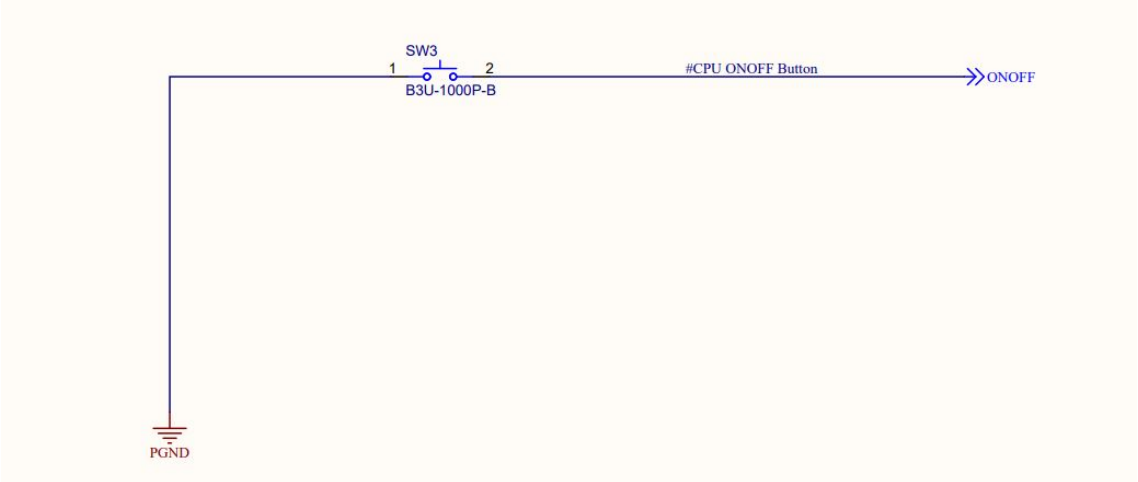


Figure 5.28: Processor On/Off power button

The voltage inverter circuits shown in figure 5.29 are also connected to the processor block shown in Figure 5.27. These transistors are used to translate standby and on requests from the processor to the PMIC as discussed in Section 5.1.2.2.

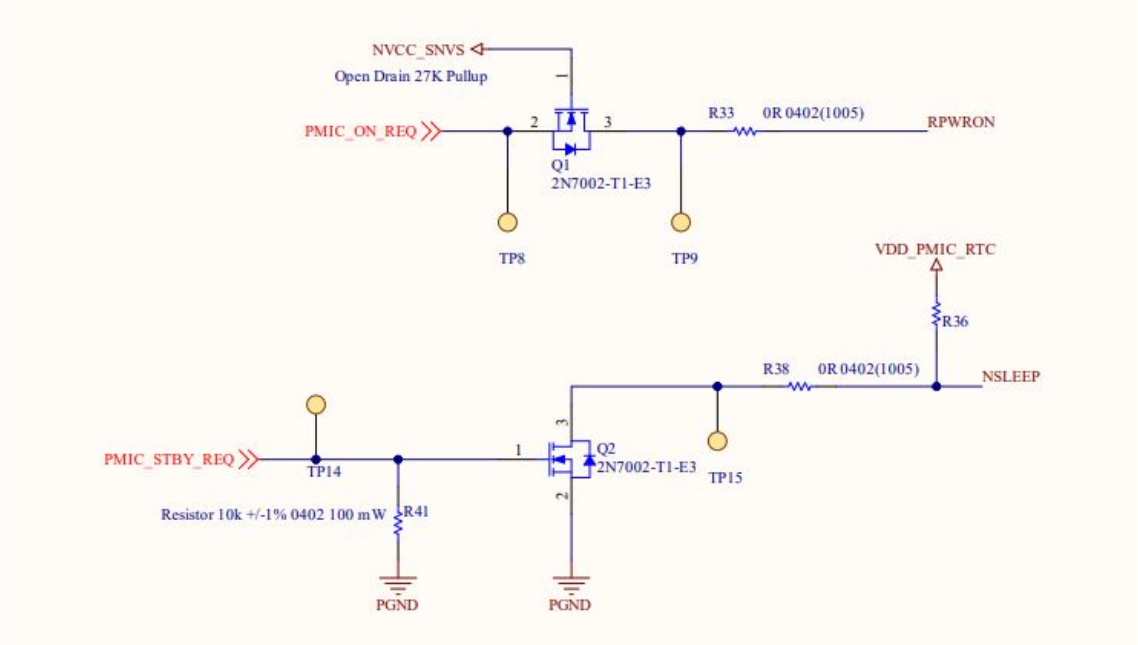


Figure 5.29: Voltage inverters for communication from the processor to the PMIC

5.11 Secure Digital Card Storage

The i.MX8MQ supports SD/eMMC4.3 single data rate), SD/eMMC4.4/4.41 (double data rate) and SDR104/50 Secure Digital (SD) card standards [56, page 43].

SD/eMMC4.3 and SD/eMMC4.4/4.41 is a $3.3V$ standard while SDR104/SDR50 is a $1.8V$ standard. The SD/eMMC internal block in the processor is powered by the NVCC_SD1 and NVVC_SD2 voltage domains which is connected to the NVCC_XXX voltage rail. It is highly recommended not to reprogram this rail to $1.8V$ as the JTAG, I2C, Ethernet and ADC are $3.3V$ and depend on it.

Only the SD/eMMC4.3 (single data rate) and SD/eMMC4.4/4.41 (double data rate) standards are therefore supported.

For the SD/eMMC4.3 standard clock frequency is $400kHz$ in Low-Speed Mode and $50MHz$ in High-Speed Mode. Clock duty cycle is 50%, setup time is $2.5ns$ and hold time is $1.5ns$. The datasheet recommends that the routed delay difference not exceed $2ns$.

SD lines are routed with a 50Ω impedance, lines have a delay of approximately $520ps$ and a maximum difference of $0.641ps$ between lines.

The lower 4 bits of the 8-bit eMMC compatible SD1 interface are routed to the SD card.

$3.3V$ is supplied to the SD card from the same NVCC_XXX rail as the internal processor block. The SD card is decoupled by means of a $22\mu F$ and $100nF$ capacitor. The peak operating current of the SD card is expected to be $300mA$. [82, page 14]

Expected transfer speeds are 4 x $50Mbps$ or 4 x $100Mbps$. SD Card interface configuration is discussed further in the software section.

Figure 5.30 shows the SD card implementation.

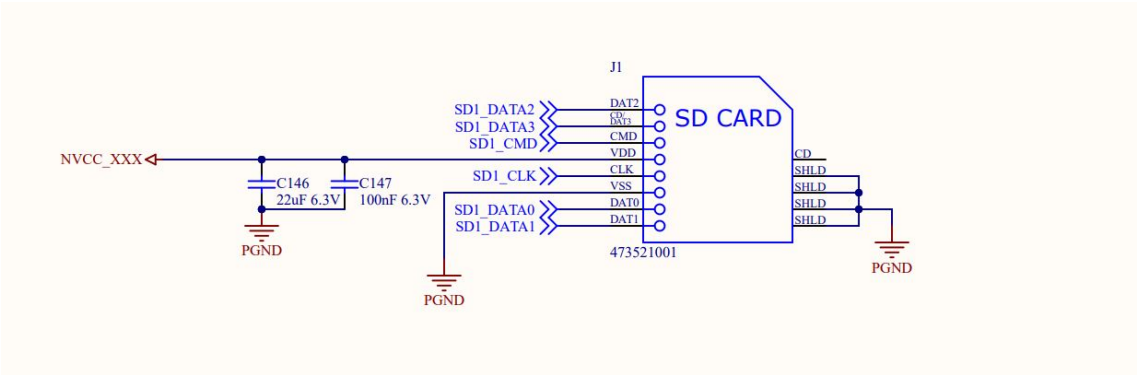


Figure 5.30: SD Card implementation

5.12 Universal Asynchronous Receiver Transmitter

The i.MX8MQ has 4 universal asynchronous receiver transmitter (UART) ports of which 3 are implemented and accessible via the left expansion connector. UART1 and UART2 are connected to a CP2105 USB-to-UART Bridge and is pivotal to the initial board bring up process. UART1 is reserved for control while UART2 is reserved for debugging information in the standard software configuration.

The CP2105 from Silicon Labs is specifically chosen as it is recommended by NXP for use with its debugging software. It supports baud rates of $2400bps$ to $921600bps$ [46, page 1].

The UART internal block of the processor is part of the NVCC_UART voltage domain and is connected to the NVCC_XXX voltage rail with sequence order 7.

The USB part of the CP2105 is powered by the USB bus while the UART part of the component is powered by the VDD_PHY_3V3 voltage rail that has sequence order 8.

As the CP1025 is powered up after the i.MX8MQ and latch-up is mitigated.

The CP2105 reset is active low and must be pulled up to the VDD_PHY_3V3 voltage rail via a $4.7k\Omega$ resistor. The component produces a $3.45V$ output that is capable of delivering up to $18.5mA$ for self-powered applications, but is entirely unused in this design.

The REGIN and VBUS ports of the CP2105 are fed via the miniUSB connector by USB host device. USB lines are routed to the connector with 90Ω differential impedance. USB ground is connected directly to system ground. Figure 5.31 shows the USB to UART bridge implementation.

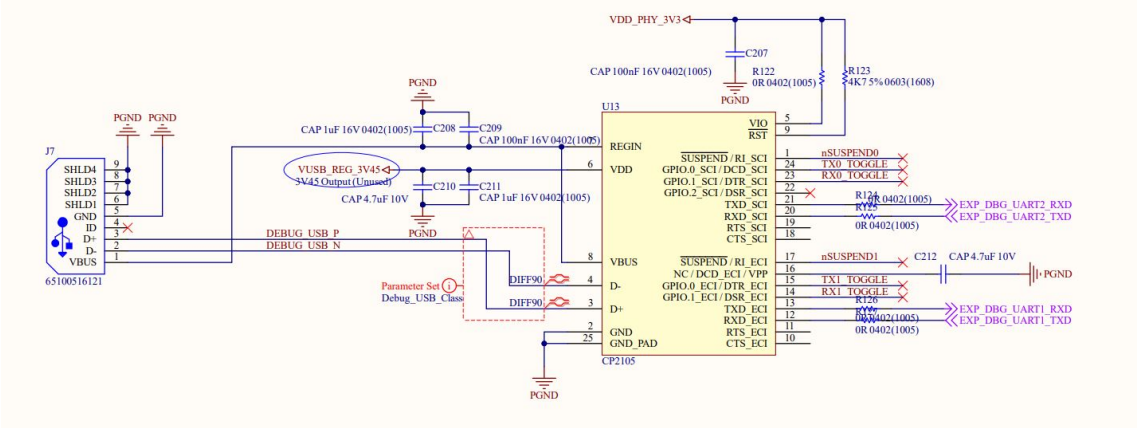


Figure 5.31: USB to UART Bridge

5.13 Universal Serial Bus

The i.MX8MQ has 2 Universal Serial Bus (USB) 3.0 that can be used in either host or on the go configurations depending on the USBID pin configuration. In host mode the Bokkiebord must supply a relatively large amount of power to the USB device. The NX5P3290 current limited power switch is used to protect the USB interface.

The sequenced VDD_PHY_5V voltage rail is connected to the VIN input of the NX5P3290. VBUS is the output that feeds the USB device. This protected voltage rail is then decoupled by means of a $4.7\mu F$ and $100nF$ capacitor and connected to a ESD5B5.0ST1G transient voltage protection diode. This voltage rail is connected directly to the VBUS voltage of the USB connector.

A $33k\Omega$ Resistor is chosen to connect the ILIM pin to ground. This will limit the maximum USB current to $1650mA$ [62, page 15]. A ZLLS400 diode is connected between the enable line and the $\overline{fault}$ port. The enable pin is usually pulled high by a $10k\Omega$ resistor to NVCC_XXX. Upon a fault occurring, $\overline{fault}$ will pull the enable line low, thereby disabling the supply to the USB device. When the fault is removed $\overline{fault}$ will return to high impedance allowing enable to be pulled high and supply to resume. The de-glitch interval is $8ms$.

A 1nF capacitor is recommended to be connected to the CAP pin by the datasheet. Figure 5.32 shows the USB over-current protection implementation.

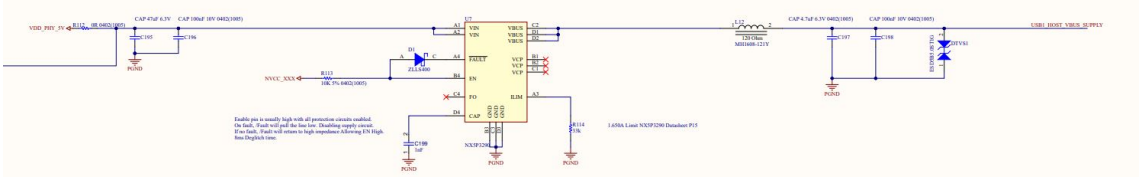


Figure 5.32: USB Over-current protection.

USB 3.0 consists of the standard USB 2.0 differential data pair as well as 1 differential super speed transmit and 1 differential super speed receive pair. All these signals are filtered and

protected from ESD and EMI by means of 2 PCMF3USB3S filters from Nexperia. Figure 5.33 shows the USB EMI filter implementation.

Figure 5.33: USB Filtering.

The dual USB 3.0 interface is primarily designed to interface with peripheral devices and the Bokkiebord will be the bus master in most of the cases. For debugging purposes 1 of the USB ports might have to be used as the USB OTG port especially for the initial memory initialisation process. In this case a cable with 2 USB type A connectors are used **with the VBUS line Broken**. The ID pins must be left floating and the VBUS voltage detection resistor must be populated. Figure 5.34 shows the dual USB3.0 connector implementation.

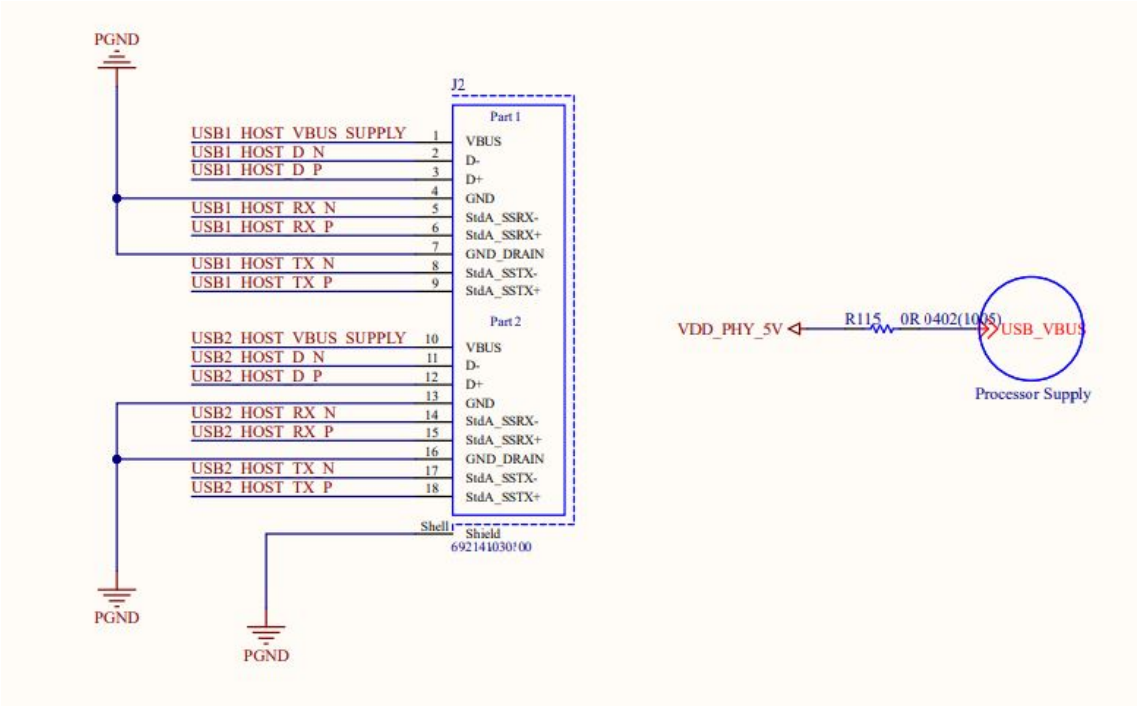


Figure 5.34: Dual USB3.0 Connector

5.14 Heat Sink and Thermal management

The i.MX8MQ is a powerful chip that consumes substantial current if under heavy load. The i.MX8MQ is expected to dissipate $12.61126W$ of energy under high load condition. This value is calculated by summing the resultant power draw from only processor voltage rails given in

Table 5.1 measurements indicate that the i.MX8MQ draws approximately $1.75W$ at idle. Thermal management can be done passively or actively.

Thermal management is integrated into the device tree of the operating system and clock speed is reduced if temperature exceeds acceptable limits. Standard temperature zones are alert at 80^oC and critical at 90^oC. The i.MX8MQ has a safe operating range of between 0^oC and 105^oC.

The package to case thermal resistance of the processor is $0.1^oC/W$.

A heat sink is attached to the processor by means of a thermally conductive adhesive strip. The per square inch thermal resistance of the strip is $0.82^oCin^2/W$. The package size is $17mm$ x $17mm$ or $0.6693in$ x $0.6693in$. The normalised thermal resistance of the strip is $0.82/(0.6693^2) = 1.83^oC/W$.

Thermal management is maintained by the use of a $19mm$ x $19mm$ x $12.7mm$ APF19-19-13CB heat sink. It is chosen primarily for its small size and fairly good thermal resistance. The heat sink has a thermal resistance of $4.4^oC/W$ for forced air condition and $15.6^oC/W$ for natural convection condition.

Forced air convection is made possible by the use of a Sunon MC17080V1-0000-A99 $17mm$ x $17mm$ x $8mm$ fan that is connected to the sequenced VDD_PHY_5V voltage rail of the left expansion connector. The approximate current draw is $184mA$.

The temperature difference between the die of the processor and the environment is described by equation 5.39 [49, page 568].

$$T_{dev} - T_{amb} = P_D \times (\theta_{dev-case} + \theta_{case-sink} + \theta_{sink-amb}) \tag{5.39}$$

$$P_D = \frac{T_{dev} - T_{amb}}{\theta_{dev-case} + \theta_{case-sink} + \theta_{sink-amb}} \tag{5.40}$$

Where:
T_{dev} is the temperature of the processor die in degree Celsius.
T_{amb} is the ambient temperature of the environment in degree Celsius.
P_D is the thermal dissipation of the processor die in watt.
$\theta_{dev-case}$ is the thermal resistance of the path from die to case in degree Celsius per watt.
$\theta_{case-sink}$ is the thermal resistance of the path from case to heat sink in degree Celsius per watt.
$\theta_{sink-amb}$ is the thermal resistance of the path from heat sink to the environment in degree Celsius per watt.

Assuming an ambient room temperature of 23^oC and a maximum die temperature of 95^oC. The maximum allowable thermal dissipation of the processor with natural convection is:

$$P_D = \frac{95 - 23}{0.1 + 1.83 + 15.6} \tag{5.41}$$

$$P_D = 4.107W \tag{5.42}$$

The maximum allowable thermal dissipation of the processor with forced convection is:

$$P_D = \frac{95 - 23}{0.1 + 1.83 + 4.4} \tag{5.43}$$

$$P_D = 11.37W \tag{5.44}$$

This is deemed to be acceptable as the probability of maximum load condition being continuous is low and can be managed by adjusting the clock speed if absolutely required.

Chapter 6

Printed Circuit Board Layout

This chapter discusses the calculation of parameters used to complete layout and routing. Refer to Appendix A.1 for the layout result.

6.1 Manufacturing Tolerance

The Bokkiebord is designed to be as similar as possible to the BeagleBone Black to ensure compatibility between the 2 boards.

The BeagleBone Black measures 86.36mm x 55.88mm (3400mil x 2200mil). It is a 6-layer board with 0.12065mil via neck width. The BeagleBone Black's processor, the Texas Instruments AM3358, has a pin spacing of 0.8mm (31.5mil). The NXP iMX8MQ has a pin spacing of 0.65mm and a 0.31mm pad size.

Preliminary prices are determined using PCBWay's online calculator tool [69].

Cost analysis is performed by isolating and evaluating the contribution of each parameter to the total cost. All parameters except for the parameter being investigated is fixed to a modest setting while the parameter under investigation is varied over different options and cost is recorded. This information is used to determine when the parameter under investigation becomes a dominant cost factor.

The technical requirements are compared to the cost information obtained. Some factors are directly determined by the needs of the conceptual design or by processor footprints.

The lowest cost specification is chosen if no technical limitation exists. Refer to Chapter 11 for exact cost breakdown of the Bokkiebord.

6.1.1 Size

The conceptual design requires that the board outline be the same as the BeagleBone Black and the Bokkiebord designed to also be 86.36mm **x** 55.88mm (3400mil **x** 2200mil).

6.1.2 Thickness

PCB Thickness is determined by copper and dielectric thicknesses of the board. If a design has many layers, the board is made as thick as needed to realise the required number of layers and impedances.

For designs with comparatively few layers the thickness is suited to mechanical integrity and designer preference.

A $1.6mm$ board is chosen for mechanical integrity reasons to reduce bending during the assembly process. Reduced bending lowers the risk of ball grid array (BGA) components becoming dislodged and reduces debugging time [68]. A $1.6mm$ board will require more heat during the soldering process.

The PCB lamination process is also less susceptible to errors caused by tight press tolerances during board manufacture. There is no cost difference between thicknesses.

6.1.3 Layers

Layer cost is determined by material quantity and manufacturing time required. Copper cores consist of prepregs and copper layers that have been pressed in advance. If a stackup can be manufactured using cores instead of layering prepregs and copper foils, the total assembly time is reduced, resulting in a lower cost PCB.

All parameters except the layer count under investigation are fixed to a modest setting while the layer count under investigation is varied over different options. This information is used to determine when the layer count becomes a dominant cost factor. The layers are increased and the resultant cost is and tabulated in Table 6.1.

The cost per layer column shows the effective cost of a copper layer as $ value. The layer-on-layer column shows the percentage cost increase from the previous per layer cost. Low-cost increase values can be considered an incentive to upgrade to a higher layer count. The fixed parameters are via hole: $0.3mm$, width/spacing: $6mil/6mil$, copper thickness: $1oz$.

Number of Layers	**Cost** ($)	**Cost Per Layer** ($)	**Layer on Layer Increase (%)**
1	5	5	0
2	5	2.5	-50%
4	49	12.25	390%
6	179	29.33	139.428%
8	273	34.125	16.348%
10	545	54.5	59.707%

Table 6.1: Printed Circuit Board Layer Cost Comparison

Table 6.1 shows that cost does increase with layer count. There is a clear distinction between simpler 1- 2-layer boards and 4- 10-layer boards. 1- 2-Layer boards can be prepared in advance and only require etching and finishing steps, while 4- 10-multi-layer boards require extensive additional lamination and finishing steps. 1-Layer boards are discouraged by offering a much better per layer price with the 2-layer board.

Note the low-cost increase value between the 6- and 8-layer boards. While total cost is slightly higher, substantial extra board area is cost-efficiently obtained.

A **6-layer** stack up is chosen as it is the minimum amount of layers that allow for successful routing of the required number and density of high-speed signals. 8-Layers can be considered if only additional power planes are required and 10-layers can be considered if more high-speed signals are required.

The prices shown above are for standard manufacturer stackups and not custom stackups but they provide a good guideline for layer cost. A 10-layer stackup is significantly more expensive than a 6-layer stackup.

6.1.4 Vias

Via cost is determined by hole size and via technology.

Vias can be drilled mechanically or by laser. Mechanically drilled vias are the most common, with laser drilling mainly being reserved for micro-vias placed within pads. Mechanically drilled vias are manufactured using tiny drill bits that have finite lifespans since their small diameter does not allow such material to be lost to wear. A higher cost is thus associated with smaller via size.

Vias can be drilled entirely- or partway through the PCB with depth controlled drilling methods [65]. If different board layers are drilled before lamination, blind vias can be manufactured that are entirely embedded inside the PCB.

All values except the via hole size being investigated are fixed to a modest setting while the via hole size is varied over different options. This information is used to determine when the via hole size becomes a dominant cost factor. The via hole size is decreased and the resultant cost is tabulated in Table 6.2. The fixed parameters are layer count: 6-layers, width/spacing: $6mil/6mil$, copper thickness: $1oz$.

Via Hole Size (mm)	**Cost** ($)	**Cost per** $0.05mm$ ($)	**Cost Increase %**
0.3	179	0	0
0.25	212	33	∞
0.2	234	27.5	-16.67%
0.15	380	67	143.636%

Table 6.2: Printed Circuit Board Via Cost Comparison

Table 6.2 shows that cost does increase with decreased via hole size. A jump in cost is observed between $0.3mm$ and $0.25mm$ via hole size. The cost for $0.25mm$ via hole size and $0.2mm$ via hole size is very similar. A big jump in cost is observed when decreasing from $0.2mm$ to $0.15mm$ via hole size.

The i.MX8MQ Processor has a BGA footprint with 621 pins and a pin spacing of $0.65mm$ between pins in both the x or y direction. This dense pin spacing does not allow traces on the inner rows of the processor to be routed out on the same layer as the footprint. Vias are placed within the BGA footprint to the signals out to different layers.

The via hole size requirement of the via is constrained by the BGA footprint and the manufacturer specifications. Via hole size is a factor of the pin spacing, pad diameter and pad to via clearance. Figure 6.1 and Equation 6.1 show how via hole size is to be determined.

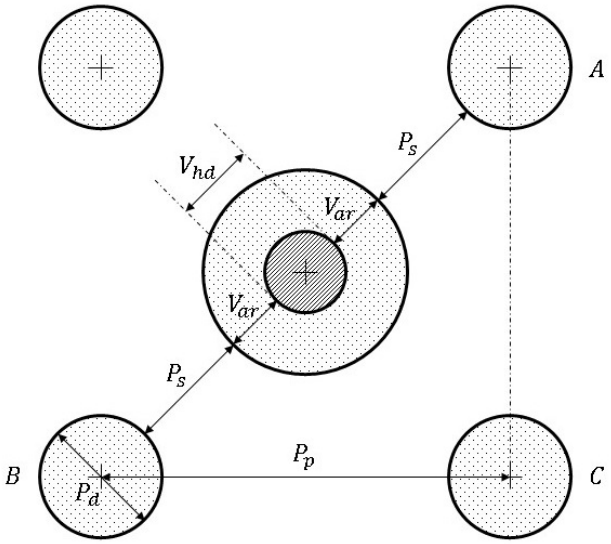


Figure 6.1: Via Hole Size as Product of BGA Pad Size, Pitch and Required Spacings

$$\frac{P_p}{\sin(45^o)} = hypotenuse_{ab} = 2 \times \frac{P_d}{2} + 2 \times P_s + 2 \times V_{ar} + V_{hd} \tag{6.1}$$

Where:
P_p is BGA pin pitch in millimetre.
P_d is BGA pad diameter in millimetre.
P_s is minimum pad to via spacing in millimetre.
V_{ar} is minimum width of the annular ring in millimetre.
V_{hd} is required hole size in millimetre

For the i.MX8MQ the pin pitch is $P_p = 0.65mm$ and pad diameter is $P_d = 0.31mm$. For PCBWay the minimum pad to via spacing is $P_s = 0.102mm$ and the minimum width of an annular ring is $V_{ar} = 0.102mm$.

$$0.65 \times \frac{2}{\sqrt{2}} = 2 \times \frac{0.31}{2} + 2 \times 0.102 + 2 \times 0.102 + V_{hd} \tag{6.2}$$

$$\Rightarrow V_{hd} = 0.2012388155[mm] \tag{6.3}$$

Equation 6.3 shows that a maximum via hole size of $0.201mm$ is acceptable. The via hole size is selected to be $0.2mm$.

6.1.5 Minimum Copper Width and Spacing

Spacing cost is determined by the care taken during the etching process. The copper layers of the PCB are first laminated between photosensitive layers and are then exposed to ultraviolet light projected through a photomask of the design.

Exposed areas cure to form a temporary protective film and he uncured photosensitive material is washed away to expose bare copper.

The PCB is then immersed in a chemical solution such as ferric chloride ($FeCl_3$). The unconnected parts of the PCB are created by dissolving the exposed copper.

A greater control over the etching process is required when trace width requirements become very thin. This means that a more diluted etching solution must be used which increases manufacturing time and cost [67].

All values except for the spacing tolerance being investigated are set to a modest setting while the spacing tolerance is varied over different options. This information is then used to determine at which point the spacing tolerance becomes a dominant cost factor. Spacing tolerance is decreased and the resultant cost is tabulated in Table 6.3. The fixed parameters are layer count: 6-layers, via size: $0.3mm$ and copper thickness: $1oz$.

Min. Width (mil) **/ Spacing** (mil)	**Cost** ($)	**Cost per** $1mil$ ($)	**Cost increase (%)**
6/6	179	0	0
5/5	212	33	∞
4/4	234	27.5	-16.667
3/3	330	50.3	82.909

Table 6.3: Printed Circuit Board Width and Spacing Cost Comparison

It is concluded that the difference in cost for $5mil/5mil$ spacing and $4mil/4mil$ spacing is negligible and $4mil/4mil$ **is selected**.

6.1.6 Combination

Results from these comparisons are combined in Table 6.4. 6-layers with $1.6mm$ thickness, $0.2mm$ via hole size and $1oz$ copper thickness is selected. Note that due to the $0.2mm$ via hole size requirement the overall complexity of the PCB is high enough that the cost difference between most width / spacing options is negligible.

Min. Width (mil) **/ Spacing** (mil)	**Cost** ($)
8	234
6	234
5	234
4	234
3	332

Table 6.4: Printed Circuit Board Spacing Cost Comparison for a 6-Layer Board With 0.2mm Via Size

The smallest track spacing that does not incur extra cost is selected. The final selected specifications are thus: 6-layers with 1.6mm thickness, 0.2mm via hole size, $4mil/4mil$ spacing, $1oz$ copper and 5 boards quantity.

6.1.7 Summary

Final manufacturing specifications are chosen upon evaluation of the compiled cost analysis results. These manufacturing specifications successfully achieve the requirements for implementing the i.MX8MQ processor without over-designing at a higher cost.

All quotations are given as a guideline only and are subject to review by PCBWay. Cost differences are due primarily to the custom stackup requirement and are expected to reduce significantly when more PCBs are manufactured on a custom PCBWay panel of standard size.

The final PCB cost is $430 for 5 boards. The final specifications selected for the Bokkiebord are shown in Table 6.5:

Parameter	**Value**
Board type	Single Pieces
Size	55.88mm x 86.36mm
Quantity	5
Layers	6
Material	FR-4
Thickness	1.6mm
Min Track/Spacing	$4mil/4mil$
Min Hole Size	0.2mm
Soldermask	Green
Silkscreen	White
Surface finish	Immersion Gold
Via Process	Tenting vias
Finished Copper	$1oz$ Cu
Inner Copper	$1oz$ Cu
Extra PCB Product Number	No

Table 6.5: Printed Circuit Board Final Specification Selection

6.2 Stackup Design

PCB manufacturing costs have decreased significantly in recent years, primarily due to economies of scale. PCB manufacturers can panelise many printed circuit boards on 1 standard panel to reduce waste and cut costs.

A critical requirement of this method is that PCB designers commit to designing PCBs using the standard manufacturer stackups [71], [77], [64]. Standard manufacturer stackups have been rigorously investigated, but it is found that very few are capable of achieving both all the impedances and required levels of integration.

A custom stackup is therefore designed to meet the exact requirements of Bokkiebord.

6.2.1 Available Materials

PCBWay manufactures a wide range of PCB types. Different materials may be used depending on the application. Commonly stocked FR4 materials [66] are shown below in Table 6.6 - 6.8.

Impedance is primarily determined by trace width and dielectric thickness and will be a deciding factor in selecting core and prepreg thickness. The relative permittivity of the materials is fixed, and selection is limited.

Other materials such as Teflon can be used for higher frequency designs but at a considerably increased cost. Due to the high data rates of this design, propagation delay is a critical design consideration. Propagation delay is a factor of relative permittivity and is kept in mind when choosing a dielectric.

The weave of the materials [10] also become important when digital frequencies approach about $6GHz$. This design's maximum fundamental digital clock frequency is $1.6GHz$ DDR ($3.2GT/s$), and the weave is therefore not that important.

Copper Foils Thickness (μm)
12
18
35
54
90

Table 6.6: PCBWay Standard Stocked Printed Circuit Board Copper Foils

Prepreg Material	**Thickness** (mm)
1080	0.08
2116	0.12
7628	0.185

Table 6.7: PCBWay Standard Stocked Printed Circuit Board Prepreg Materials

Finished Thickness (mm)	**Core Thickness** (mm)	**Copper Thickness** (μm)
0.6	0.2	18, 35
0.8	0.5	18, 35
1.0	0.8	18, 35, 70
1.2	1.0	18, 35, 70
1.6	1.2	18, 35, 70,
2.0	1.6	18, 35, 70
2.4	2.0	18, 35, 70,

Table 6.8: PCBWay Standard Stocked Printed Circuit Board Core Materials

6.2.2 Impedance Requirements

The ratio of the voltage and current at any point z in an infinitely long transmission line is known as the characteristic impedance of the line. The NXP i.MX8MQ Hardware Developers Guide recommends impedances as specified in Table 6.9.

The stackup is specifically designed to realise these impedances while maintaining routing density.

Stackups are evaluated directly in Altium Designer. Altium Designer utilises the Simbeor [80] filed solver from Simberian.

Signal Type	**Impedance**(Ω)	**Topology**
LPDDR4 DQ	42	Single Ended
LPDDR4 DMI	42	Single Ended
LPDDR4 DQS	85	Differential
LPDDR4 CLK	85	Differential
USB	90	Differential
MIPI	100	Differential
HDMI	100	Differential
All Other	50	Single Ended

Table 6.9: Printed Circuit Board Impedance Requirements for Different Signal Types.

6.2.3 Microstrip

A microstrip is a controlled impedance trace located on an outer layer of a PCB. It consists of a copper trace and a reference plane separated by a dielectric of known thickness and permittivity. Figure 6.2 shows an example of microstrip implementation.

6.2.3.1 General Case

In most cases the characteristic impedance of microstrip lines can be approximated fairly accurately using the equation 6.4 [24, page 187]. This however only holds for the "skinny" case when ($0.1 < w/h < 2.0$).

$$Z_{0_{Microstrip}} = \frac{87}{\sqrt{\epsilon_r + 1.41}} \ln\left(\frac{5.98 \times h}{0.8 \times w + t}\right) \tag{6.4}$$

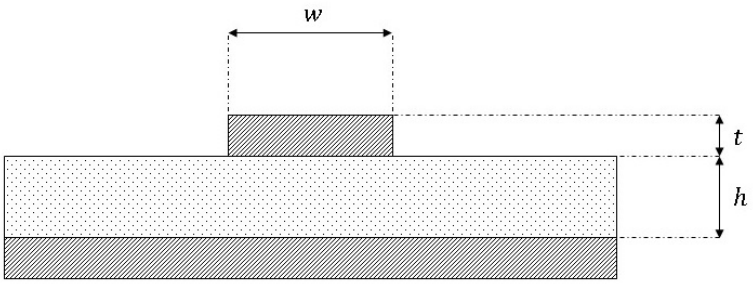


Figure 6.2: Microstrip Topology

Where:
Z_0 is trace impedance in ohm.
ϵ_r is relative permittivity which is dimensionless.
h is height above the reference plane in inches.
w microstrip trace width in inches.
t is the final trace thickness in inches.

6.2.3.2 Advanced Case

A more accurate description of the characteristic impedance of microstrip lines that hold for both the skinny and wide cases are shown in equations 6.5 - 6.12 [24, page 430–434] below. This topic is discussed in more detail from a microwave perspective [73, 141-143] and a 1st principle perspective [44, page 61-68].

Effective relative permittivity as a function of trace geometry:
For skinny traces ($w < h$):

$$\epsilon_{skinny} = \frac{\epsilon_r + 1}{2} + \left(\frac{\epsilon_r - 1}{2}\right) \times \left(\left(1 + \frac{12 \times h}{w}\right)^{-\frac{1}{2}} + \frac{1}{25} \times \left(1 - \frac{w}{h}\right)^2\right) \tag{6.5}$$

For wide traces ($w > h$):

$$\epsilon_{wide} = \frac{\epsilon_r + 1}{2} + \left(\left(\frac{\epsilon_r - 1}{2}\right) \times \left(1 + \frac{12 \times h}{w}\right)^{-\frac{1}{2}}\right) \tag{6.6}$$

Special adjustment to account for trace thickness:

$$\epsilon_{eff} = \epsilon_{chosen} - \frac{(\epsilon_r - 1) \times \frac{t}{h}}{4.6 \times \sqrt{\frac{w}{h}}} \tag{6.7}$$

Effective trace width as a function of other parameters:
For skinny traces ($2\pi w < h$):

$$WE_{skinny} = w + \frac{1.25 \times t}{\pi} \times \left(1 + \ln\left(\frac{4 \times \pi \times w}{t}\right)\right) \tag{6.8}$$

For wide traces ($2\pi w > h$):

$$WE_{wide} = w + \frac{1.25 \times t}{\pi}\left(1 + \ln\left(\frac{2 \times h}{t}\right)\right) \tag{6.9}$$

Characteristic impedance:
For skinny traces ($w < h$):

$$Z_{MS_{skinny}} = 60 \times \ln\left(\frac{8 \times h}{WE_{skinny}} + \frac{WE_{skinny}}{4 \times h}\right) \tag{6.10}$$

For wide traces ($w > h$):

$$Z_{MS_{wide}} = \frac{120\pi}{\frac{WE_{wide}}{h} + 1.393 + 0.667 \times \ln\left(\frac{WE_{wide}}{h} + 1.444\right)} \tag{6.11}$$

Normalising for effective permittivity:

$$Z_{Microstrip} = \begin{cases} \frac{Z_{MS_{wide}}}{\sqrt{\epsilon_{eff}}}, & \text{if } w \geq h \\ \frac{Z_{MS_{skinny}}}{\sqrt{\epsilon_{eff}}}, & \text{if } w < h \end{cases} \tag{6.12}$$

Where:
h is trace height above reference plane in inches.
w is trace width in inches.
t is trace thickness in inches.
ϵ_r is relative permittivity which is dimensionless.

6.2.4 Stripline

A stripline is a controlled impedance trace located within the inner layers of a PCB. It consists of a copper trace located a known distance from 2 reference planes. It is surrounded by a dielectric of known permittivity and thickness. Figure 6.3 shows an example of stripline implementation.

6.2.4.1 General Case

In most cases the characteristic impedance of the stripline can be approximated by equation 6.13 [24, page 188]. Both dielectrics are not always of the same thickness and do not always have the same relative permittivity. This topic is discussed in more detail from a microwave perspective [73, 147-149].

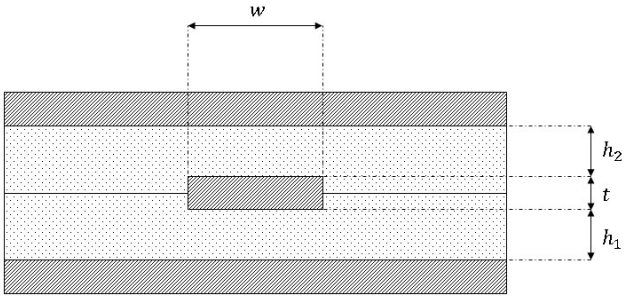


Figure 6.3: Stripline Topology

$$Z_{0_{Stripline}} = \frac{60}{\sqrt{\epsilon_r}} \ln\left(\frac{1.9 \times b}{0.8 \times w + t}\right) \tag{6.13}$$

Where:
Z_0 is trace impedance in ohm.
ϵ_r is relative permittivity which is dimensionless.
b is the separation between reference planes in inches.
w is stripline trace width in inches.
t is the final trace thickness in inches.

6.2.4.2 Advanced Case

A more accurate description of the characteristic impedance of striplines that hold for both the skinny and wide cases are shown in equations 6.14 - 6.17 [24, page 436–438] below.

For skinny traces:

$$Z_{STR_{k1}} = \left(\frac{w}{2}\right) \times \left(1 + \frac{t}{\pi \times w} \times \left(1 + ln\left(\frac{4 \times \pi \times w}{t}\right)\right) + 0.255 \times \left(\frac{t}{w}\right)^2\right) \tag{6.14}$$

$$Z_{STR_{skinny}} = \frac{6}{\sqrt{\epsilon_r}} \times ln\left(\frac{4 \times b}{\pi \times Z_{STR_{k1}}}\right) \tag{6.15}$$

For wide traces:

$$Z_{STR_{k2}} = \left(\frac{2}{1 - \frac{t}{b}} \times \ln\left(\frac{1}{1 - \frac{t}{b}} + 1\right) - \left(\frac{1}{1 - \frac{t}{b}} - 1\right) \times \ln\left(\frac{1}{\left(1 - \frac{t}{b}\right)^2} - 1\right)\right) \tag{6.16}$$

$$Z_{STR_{wide}} = \frac{94.15}{\frac{\frac{w}{b}}{1 - \frac{t}{b}} + \frac{Z_{STR_{k2}}}{\pi}} \times \frac{1}{\sqrt{\epsilon_r}} \tag{6.17}$$

Where:
$h1$ is trace height above bottom reference plane in inches.
$h2$ is trace height below upper reference plane in inches.
b is the separation between reference planes in inches.
w trace width in inches.
t trace thickness in inches.
ϵ_r is relative permittivity.

6.2.5 Co-Planar Wave Guide

A co-planar wave guide is a controlled impedance trace located on an outer layer with 2 adjacent ground planes. When microstrips or striplines are placed too close to surrounding grounds, the trace might inadvertently act as a co-planar wave guide and impedances might be substantially lower than expected. Equations 6.18 - 6.22 show impedance calculations for a co-planar wave guide [9, page 166–169] [87, Appendix G]. Figure 6.4 shows an example of co-planar wave guide implementation.

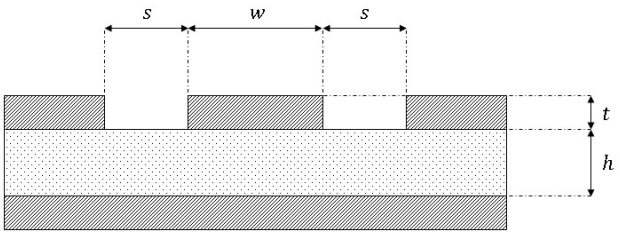


Figure 6.4: Coplanar Topology

The characteristic impedance of a can be written as Z_0:

$$Z_0 = \frac{30 \times \pi}{\sqrt{\epsilon_{eff}}} \times \frac{K^{'}(k)}{K(k)} \tag{6.18}$$

$$\epsilon_{eff} = \frac{\epsilon_r - 1}{2} \times \left(\tanh\left(1.785 \times \log\left(\frac{h}{s}\right) + 1.75\right) + \left(\frac{k \times s}{h} \times (0.04 - 0.7 \times k + 0.01 \times (1 - 0.1 \times \epsilon_r) \times (0.25 + k))\right)\right) \tag{6.19}$$

$$k = \frac{w}{w + 2s} \tag{6.20}$$

For $(0 \leq k \leq 0.707)$:

$$\frac{K(k)}{K'(k)} = \frac{1}{\pi} \times \ln\left(2 \times \frac{1+\sqrt{k}}{1-\sqrt{k}}\right) \tag{6.21}$$

For $(0.707 \geq k \geq 1)$:

$$\frac{K'(k)}{K(k)} = \frac{1}{\pi} \times \ln\left(2 \times \frac{1+\sqrt{k}}{1-\sqrt{k}}\right) \tag{6.22}$$

Where:
Z_0 is characteristic impedance in ohm.
ϵ_r is relative permittivity which is dimensionless.
w is trace width in inches.
s is separation between trace and ground planes in inches.

Figure 6.5 illustrates how the impedance of a $1.375mm$ trace on a dielectric with dielectric constant of 4.2 separated $1.53mm$ from a ground plane changes depending on the ground spacing s. Notice how the co-planar wave guide impedance becomes dominant when s becomes smaller than $0.75mm$.

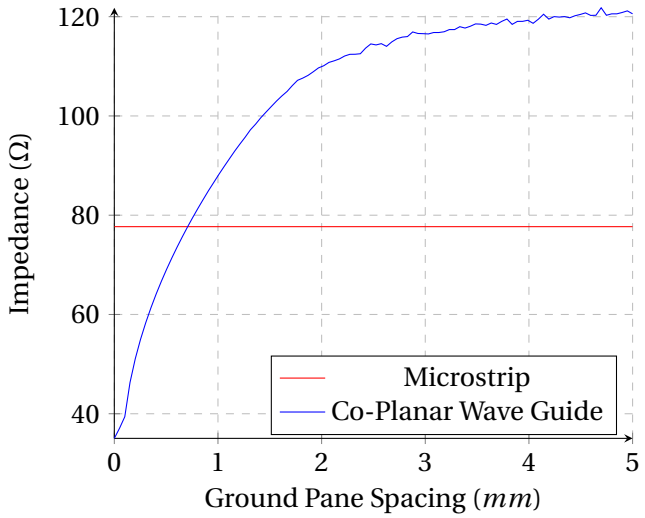


Figure 6.5: Co-Planar Wave Guide Impedance vs Ground Plane Spacing Distance

6.2.6 6-Layer Design

The 6-layer stack up used in the Bokkiebord is made from materials normally stocked by PCB-Way. Signal layers are placed very close to their reference planes to minimise trace thickness to that of the manufacturing tolerance of the manufacturer.

This also serves to maximise routing density as crosstalk is contained by electric fields being able to terminate on the reference plane sooner.

The central core of this PCB is a $0.667mm$ thick layer of FR4 material. The core provides the needed structural rigidity while separating L3 and L4 to minimise crosstalk. The chosen configuration of materials are shown in Table 6.10.

The required width and spacings for the respective impedances for the designed 6-layer stackup are shown in table 6.11 - 6.15.

Layer Name	Layer Material Type	Layer Material Name		Thickness Value	Thickness Unit	Thickness (mm)	Dielectric Constant
L1	Copper	$1oz$ Copper		1	ounce	0.035	
PP1	Prepreg	Prepreg 1080		2.874	mil	0.0729996	4.09
L2	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
C1	Core	$1oz$ - $0.2mm$ - $1oz$		5.07	mil	0.128778	4.9
L3	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
PP2	Prepreg	Prepreg 2166		4.961	mil	0.1260094	4.25
C2	Core	FR4		26.16	mil	0.664464	4.9
PP3	Prepreg	Prepreg 2166		4.961	mil	0.1260094	4.25
L4	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
C3	Core	$1oz$ - $0.2mm$ - $1oz$		5.07	mil	0.128778	4.9
L5	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
PP4	Prepreg	Prepreg 1080		2.874	mil	0.0729996	4.09
L6	Copper	$1oz$ Copper		1	ounce	0.035	

Table 6.10: Custom Designed 6-Layer Stackup

Layer Name	**Width** (mm)	**Impedance** (Ω)	**Deviation** (%)	**Propagation Delay** (ns/m)
L1	0.167	43.6	3.81	5.8329
L2	-	-	-	-
L3	0.193	42.92	2.19	7.2970
L4	0.193	42.92	2.19	7.2970
L5	-	-	-	-
L6	0.167	43.6	3.81	5.8329

Table 6.11: Trace Width for 42Ω Single Ended on Custom 6-Layer Stackup

Layer Name	**Width** (mm)	**Impedance** (Ω)	**Deviation** (%)	**Propagation Delay** (ns/m)
L1	0.1185	52.5	5	5.7229
L2	-	-	-	-
L3	0.13559	50.0	0.01	7.2813
L4	0.13559	50.0	0.01	7.2813
L5	-	-	-	-
L6	0.1185	52.5	5	5.7229

Table 6.12: Trace Width for 50Ω Single Ended on Custom 6-Layer Stackup

Layer Name	**Width** (mm)	**Gap** (mm)	**Impedance** (Ω)	**Deviation** (%)	**Propagation Delay** (ns/m)
L1	0.13252	0.1016	85	0.01	5.3843
L2	-		-	-	-
L3	0.1016	0.12987	84.97	0.04	7.2025
L4	0.1016	0.12987	84.97	0.04	7.2025
L5	-		-	-	-
L6	0.13252	0.1016	85	0.01	5.3843

Table 6.13: Trace Width for 85Ω Differential on Custom 6-Layer Stackup

Layer Name	**Width** (mm)	**Gap** (mm)	**Impedance** (Ω)	**Deviation** (%)	**Propagation Delay** (ns/m)
L1	0.10325	0.1016	89.99	0.01	5.6014
L2	-		-	-	-
L3	0.1016	0.16264	90.04	0.04	7.2077
L4	0.1016	0.16264	90.04	0.04	7.2077
L5	-		-	-	-
L6	0.10325	0.1016	89.99	0.01	5.6014

Table 6.14: Trace Width for 90Ω Differential on Custom 6-Layer Stackup

Layer Name	**Width** (mm)	**Gap** (mm)	**Impedance** (Ω)	**Deviation** (%)	**Propagation Delay** (ns/m)
L1	0.1016	0.18604	99.99	0.01	5.7019
L2	-		-	-	-
L3	0.1016	0.27491	100	0	7.2267
L4	0.1016	0.27491	100	0	7.2267
L5	-		-	-	-
L6	0.1016	0.18604	99.99	0.01	5.7019

Table 6.15: Trace Width for 100Ω Differential on Custom 6-Layer Stackup

Table shows 6.11 - 6.15 that the worst-case impedance deviation is 5%. As reflected power is directly equivalent to the impedance mismatch, a 5% loss is acceptable. The best-case scenario is a 0% impedance mismatch. This stackup is deemed to be acceptable.

6.2.7 10-Layer Design (Back-up Stackup)

A 10-layer stackup is designed as contingency if the layout is unable to fit on 6-layers. The 6-layer stackup can be changed into the 10-layer stackup by replacing the thick inner core with more layers. The impedances on the existing layers will be unaffected.

Layer Name	Layer Material Type	Layer Material Name		Thickness Value	Thickness Unit	Thickness (mm)	Dielectric Constant
L1	Copper	$1oz$ Copper		1	ounce	0.035	
PP1	Prepreg	Prepreg 1080		2.874	mil	0.0729996	4.09
L2	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
C1	Core	$1oz$ - $0.2mm$ - $1oz$		5.07	mil	0.128778	4.9
L3	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
PP2	Prepreg	Prepreg 2166		4.961	mil	0.1260094	4.25
L4	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
C1	Core	$1oz$ - $0.2mm$ - $1oz$		5.07	mil	0.128778	4.9
L5	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
PP3	Prepreg	Prepreg 2166		4.961	mil	0.1260094	4.25
L6	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
C1	Core	$1oz$ - $0.2mm$ - $1oz$		5.07	mil	0.128778	4.9
L7	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
PP4	Prepreg	Prepreg 2166		4.961	mil	0.1260094	4.25
L8	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
C1	Core	$1oz$ - $0.2mm$ - $1oz$		5.07	mil	0.128778	4.9
L9	Copper	$1oz$ - $0.2mm$ - $1oz$		1	ounce	0.035	
PP5	Prepreg	Prepreg 1080		2.874	mil	0.0729996	4.09
L10	Copper	$1oz$ Copper		1	ounce	0.035	

Table 6.16: Custom Designed 10-Layer Stackup

6.3 Via Design

The wave with all of its induced fields has to transit through the via. Propagation speed may be maximised by choosing adequate isolation distances. Via impedance is determined by these values and it is important to design for the same impedance as the feeding transmission line to minimise impedance mismatch. Stackups may contain many different dielectrics, propagation speeds and impedances in different parts of the via and thus introduce unpredictability in propagation delay.

The propagation delay of a via is a function of the via capacitance. The via capacitance is given by equation 6.23 [24, page 257]:

$$C = \frac{\epsilon_r T D_1}{18.0141844 \times [D_2 - D_1]} \tag{6.23}$$

Where:
C is capacitance in farad.
ϵ_r is the relative permittivity of the substrate which is dimensionless.
T is the thickness of the substrate in millimetre.
D_1 is the diameter of the pad surrounding the via in millimetre.
D_2 is the diameter of the clearance hole in the reference plane in millimetre.

The via rise time is given by equation 6.24 [24, page 258]:

$$T_{r_{10\%-90\%}} = 1.1 \times C Z_0 \tag{6.24}$$

Where:
$T_{r_{10\%-90\%}}$ is the 10% - 90% step response time of the via in picoseconds.
C is capacitance in farad.
Z_0 is the impedance of the transmission line leading to the via in ohm.

Combining the 2 equations yields:

$$T_{r_{10\%-90\%}} = \frac{\epsilon_r T D_1 Z_0}{16.37653127 \times [D_2 - D_1]} \tag{6.25}$$

Where:
$T_{r_{10\%-90\%}}$ is the 10% - 90% step response time of the via in picoseconds.
ϵ_r is the relative permittivity substrate which is dimensionless.
T is the thickness of the substrate in millimetre.
D_1 is the diameter of the pad surrounding the via in millimetre.
D_2 is the diameter of the clearance hole in the reference plane in millimetre.
Z_0 is the impedance of the transmission line leading to the via in ohm.

Parameters are varied to find a via that could achieve the memory timing constraint on all lines while not affecting the return path or power integrity of the PCB too much by compromising the continuity of the power plane. From the above equation, it can be seen that if the

relative permittivity of the substrate ϵ_r, the substrate thickness T and feeding transmission line impedance Z_0 are kept constant, rise time can only be decreased by either minimising D1 or maximising D2.

The pad diameter is minimised to the manufacturing tolerance, and the isolation ring is iteratively enlarged until the timing requirement is met. Since the relative permittivity of the dielectrics within the PCB differ slightly, an equation is 1st derived to describe the rise time as a function of D_2. The chosen stackup design has only 6 actual copper layers, but since the inner core comprises multiple dielectrics of different permittivities, 2 ghost copper layers are added for calculation. The board may be considered an 8-layer board with the 2 innermost copper layers completely etched away.

The via rise time can thus be evaluated over the entire length by summing the delay through each dielectric part. This procedure is described in equation 6.26.

$$T_{r_{10\%-90\%}} = \frac{Z_0 D_1}{16.37653127 \times [D_2 - D_1]} \times \sum_{L=1}^{n=\infty} \epsilon_r T \tag{6.26}$$

The diameter of the pad surrounding the via is $D_1 = 2 \times 0.1016 + 0.2 = 0.403mm$ and the impedance of the transmission line leading to the via is $Z_0 = 50\Omega$.

The layer conditions for the different parts of the via used in the Bokkiebord are given in Table 6.17. These values are used in equation 6.28 to calculate isolation ring diameter as a function of total propagation delay. Figure 6.6 shows via delay as a function of isolation ring diameter.

Dielectric name	**Thickness** (mm)	**Relative permittivity**
$L_1 \rightarrow L_2$	0.0729996	4.09
$L_2 \rightarrow L_3$	0.128778	4.9
$L_3 \rightarrow L_4$	0.1260094	4.25
$L_4 \rightarrow L_5$	0.664464	4.9
$L_5 \rightarrow L_6$	0.1260094	4.25
$L_6 \rightarrow L_7$	0.128778	4.9
$L_7 \rightarrow L_8$	0.0729996	4.09

Table 6.17: Via Delay Per Layer

$$T_{r_{10\%-90\%}} = \frac{1.230419291}{D_2 - 0.403} \times [6.186114628] \tag{6.27}$$

$$\Rightarrow = D_2 = \frac{7.611514775}{T_{r_{10\%-90\%}}} + 0.403 \tag{6.28}$$

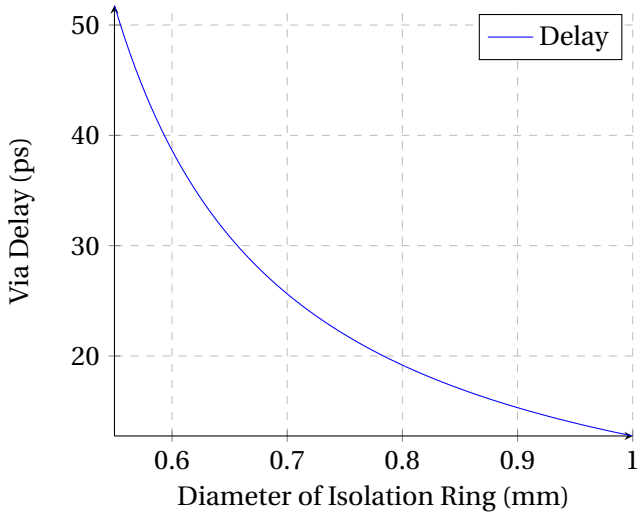


Figure 6.6: Via Delay as a Function of Reference Plane Isolation Ring

An isolation ring diameter of $0.73mm$ is selected. Layout via delay is $23.28ps$.

6.4 Propagation Delay

Timing requirement ensures that signals arrive from the driver at the source at the same time. This is achieved by ensuring that all lines are precisely the same lengths. As signals may transverse many layers with different velocities, each section of the line has to be investigated individually and delays have to be summed to find total propagation delay. Unpredictability may be minimised by reducing the number of layers on which a signal is routed.

Microstrip propagation delay is given by equation 6.29 [5, page 4]:

$$t_{pd} = 85 \times \sqrt{0.475\epsilon_r + 0.67} \tag{6.29}$$

Where:
t_{pd} is propagation delay time in pico-seconds per inch.
ϵ_r is the relative permittivity substrate which is dimensionless.

Stripline propagation delay is given by equation 6.30 [5, page 5]:

$$t_{pd} = 85 \times \sqrt{\epsilon_r} \tag{6.30}$$

Where:
t_{pd} is propagation delay time in pico-seconds per inch.
ϵ_r is the relative permittivity substrate which is dimensionless.

Table 6.18 shows mathematically calculated propagation speed per layer.

Dielectric name	Relative permittivity	Propagation speed (ns/m)
L_1	air - 4.09	5.409
L_2	4.09 - 4.9	-
L_3	4.9 - 4.25 and 4.9	7.158
L_4	4.9 and 4.25 - 4.9	7.158
L_5	4.9 - 4.09	-
L_6	4.09 - air	5.409

Table 6.18: Propagation Delay per Layer

6.5 Crosstalk Design

The movement of charge on a transmission line induces magnetic and electric fields around the line. These fields radiate out of the line following the right-hand rule adjacent to the charge moving over the line.

Fields may couple to adjacent traces which may degrade signal integrity. The trace inducing the disturbance is referred to as the aggressor trace, while the trace being coupled to is referred to as the victim trace.

Because a victim trace could be located between 2 aggressor traces in the 2D plane, it is important to consider the behaviour of both aggressor traces. The worst-case scenario would be if both aggressor traces switched simultaneously in a constructive interference manner.

The coupling coefficient of either aggressor trace should thus be confined to at least half of the digital safety margin.

This could be done by physically separating the traces, routing them orthogonally to minimise coupling, or by placing the traces close to their reference plane thereby minimising the effective coupling area by allowing fields to terminate sooner.

Figure 6.7 shows a typical 30%-70% digital threshold. Noise caused by crosstalk must be confined below the 30% threshold or above the 70% threshold but should not cause uncertainty about the digital voltage level.

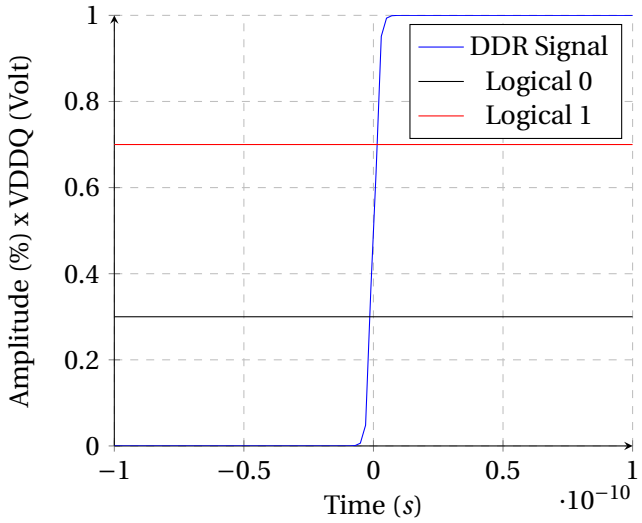


Figure 6.7: Figurative 1.1V LPDDR4 Transition with 30% and 70% digital thresholds.

6.5.1 Crosstalk between adjacent lines

A maximum cross-talk of 10% per aggressor is decided to be acceptable. Table 6.19 shows required spacing distances for 10% crosstalk on different layers for the 6-layer Bokkiebord stackup. The coupling coefficient for microstrip lines at the same height is given by equation 6.31 [8]:

$$Crosstalk = \frac{1}{1 + (\frac{D}{H})^2} \tag{6.31}$$

Where:
Crosstalk is the coefficient of aggressor voltage measured on the victim trace which is dimensionless.
D is the spacing distance in millimetre.
H is the height above the reference plane in millimetre.

This expression can be rewritten as:

$$Crosstalk = \frac{H^2}{H^2 + D^2} \tag{6.32}$$

This expression demonstrates that cross-talk can be minimised either by by minimising the distance above the reference plane or increasing the spacing distance. The outer dielectric is 0.07299mm:

$$Crosstalk = \frac{(0.07299)^2}{(0.07299)^2 + D^2} \tag{6.33}$$

Figure 6.8 shows how crosstalk is affected for different spacing distances D on outer layers.

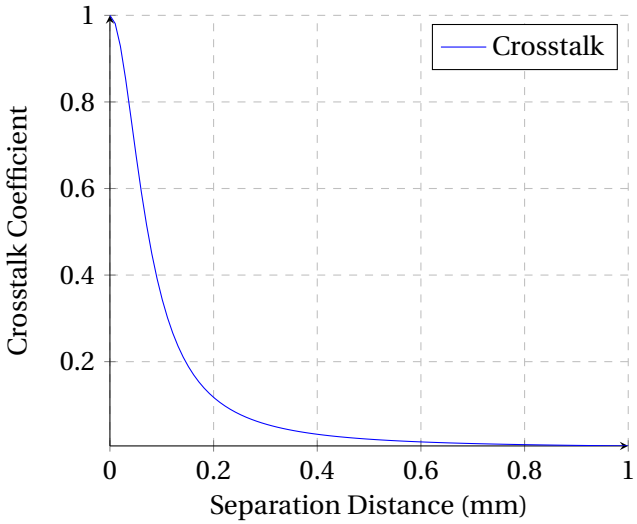


Figure 6.8: Crosstalk Coefficient vs Separation Distance on Outer Layer

For microstrips at different heights:

$$Crosstalk = \frac{1}{1 + (\frac{D^2}{H_1 \times H_2})} \tag{6.34}$$

For striplines the effective height H_n can be given as the parallel combination of heights.

$$H_n = \frac{H_{n_a} \times H_{n_b}}{H_{n_a} + H_{n_b}} \tag{6.35}$$

$H1a = 0.128778$
$H1b = 0.128778 + 0.1260094 + 0.664464 + 0.1260094 = 1.0452608$
$Hn = 0.1146525952$

Figure 6.9 shows how crosstalk is affected for different spacing distances D on inner layers.

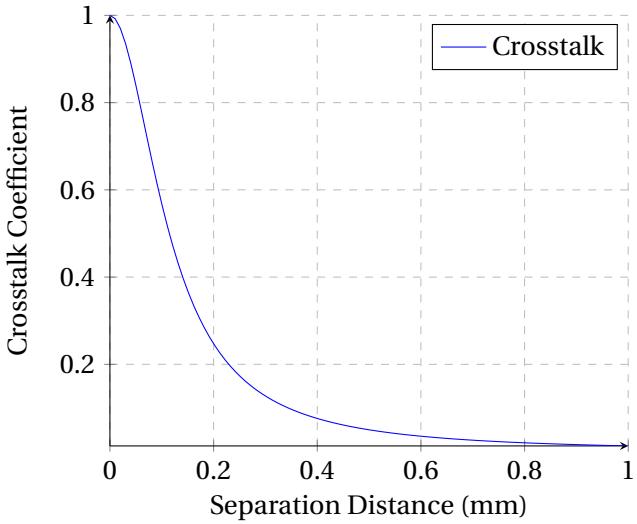


Figure 6.9: Crosstalk Coefficient vs Separation Distance on Inner Layer

Layer name	**Required Spacing for 10% Coupling** (mm)
L_1	0.201
L_2	Non-Routeable Ground
L_3	0.325
L_4	0.325
L_5	Non-Routeable Power
L_6	0.201

Table 6.19: Required 10% Crosstalk Spacing Distances per Layer

6.5.2 Self Destructive Crosstalk

The length of traces are sometimes artificially extended to achieve timing requirements. This is done by adding serpentine traces.

The spacing between adjacent parts of the serpentine may need to be minimised if the available routing area is small. This action is risky as it could cause crosstalk between different edges of the same serpentine trace.

If signal rise time on the trace is short and coupling is high, the rising edge of a signal appearing in the up-going flank of the serpentine trace might collide with the earlier part of the same rising still propagating downward in the next.

This would cause mutually destructive interference, which would degrade the signal integrity of the structure.

For this reason Dr. Howard Johnson recommends that the amplitude of the serpentine trace be no larger than 1/3 of the rise time of the signal [43].

This limit would allow at least 1 flank between the edges and thereby contain this destructive nature. It could however still cause noisy signals and might still inject some unpredictability in the actual delay time.

Distortion occurs when serpentine trace exceeds 1/3 of the rise time (2 x 1/3 round trip). The rise time of the fastest signal on the Bokkiebord is approximately $100ps$.

On an outer layer the propagation speed is $5.7229ps/mm$ which means that the rising edge of a memory signal is $17.4737mm$ long and that the amplitude of a serpentine trace may not exceed $5.8245mm$ if its flanks are within coupling distance.

On an inner layer the propagation speed is $7.2813ps/mm$ which means that the rising edge of a memory signal is $13.7338mm$ long and that the amplitude of a serpentine trace may not exceed $4.5779mm$ if its flanks are within coupling distance.

6.6 Return path

When a positive charge is induced in a trace, a corresponding negative charge is created in the reference plane.

Because line voltages are considered as waves travelling over the line, the opposing negative charge also moves directly underneath the positive wave in the reference plane.

If the reference plane is cut for any reason, the companion negative charge would have to find a longer path from driver to load.

This longer path would add inductance to the signal and cause slow rising edges and overshoot and slow falling edges and overshoot thereby damaging driver and load circuitry.

Figure 6.10 shows an example of how the Ethernet RMII lines are routed. Notice how the power plane is deliberately routed with the outline of the high-speed signals.

It is possible to compromise if a reference plane absolutely has to be cut by stitching the ends of the planes together with low value and small package capacitors but it is highly recommended to rather add more routing layers if possible.

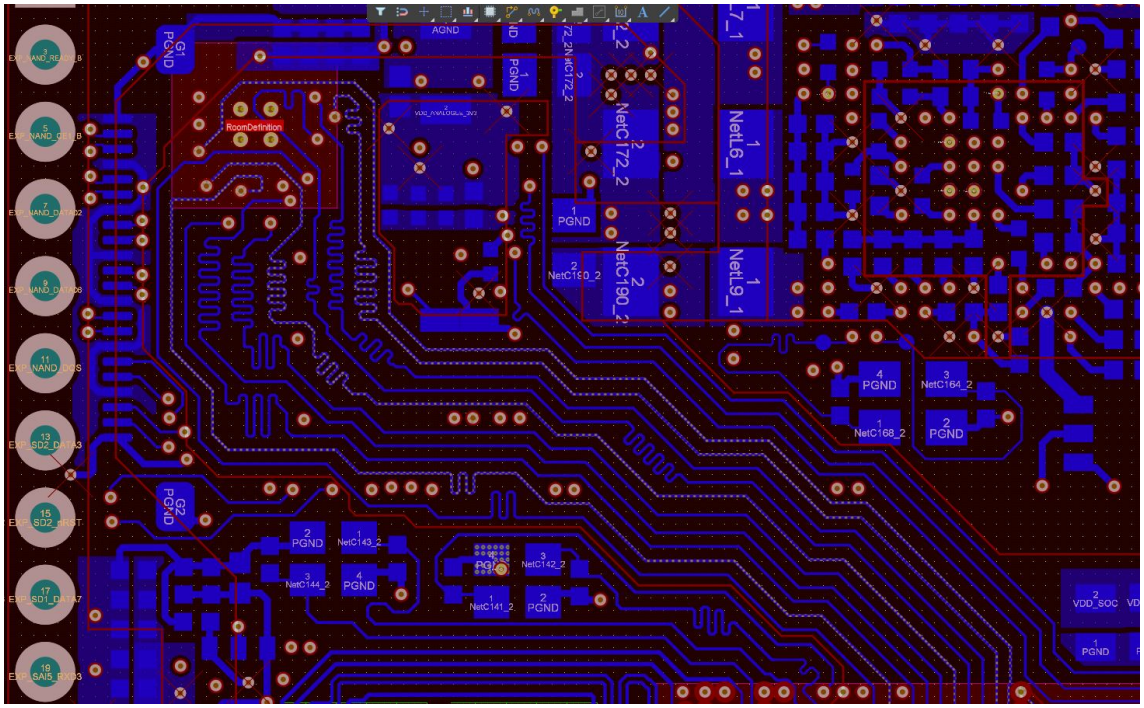


Figure 6.10: Power Planes Routed with Envelope of High-Speed Signals to Maintain Signal Integrity.

6.7 Power Distribution System Design

The Bokkiebord power supply is designed to deliver up to $50W$ of power to the various parts of the PCB. As a high instantaneous current draw is possible, the accompanying I2R losses might cause voltage rails to dip if not correctly designed.

All decoupling recommendations are for this reason strictly obeyed. Although it is not possible to place capacitors directly under the processor due to the sheer density of the design, capacitors are placed on the periphery of the chip and connected with wide power planes.

All components that share a voltage domain are connected to the same voltage rail by the same power plane or thick power trace. These connections ensure that the driver and load transistors always have the same reference voltage.

All parts of the design except the analogue part of the ADC are connected to the same uncut ground plane.

6.8 Memory Routing

The LPDDR4 interface is a high-speed interface running at up to $1.6GHz$ ($3.2GT/s$). It has 32 data lines, 10 command-and-address lines, 2 pairs of differential clock enable lines, 2 pairs of differential clock lines, 4 pairs of differential data strobe lines, and 4 chip select. It requires a range of impedances, which are realised to within 5% of the required value. All signals are matched precisely to within $1ps$ of the target timing tolerance.

6.8.1 High-Density Layout Techniques

Due to the cost limitation, 6-layers are used in this design. Of the 6-layers, only 4 are routable. To connect 62 high-speed impedance and timing controlled lines on the standard substrate between the processor and memory proves challenging.

Routing is achieved on only 4-layers by routing traces as efficiently as possible by prioritising routing completion above timing constraints. This means that lines are rather routed so that all routes can be completed without using more than 2 vias rather than routing lines with similar timing requirements on the same layer as is traditionally done.

Lines with similar timing requirements are usually routed on the same layer to account for manufacturing and calculation tolerances. Since all lines are traditionally routed on the same layer they have the same propagation delay and thus arrive at the load simultaneously without much design effort.

Timing requirements are realised in this design by the use of careful via and stackup design. A maximum of 2 vias are allowed per line partly because they contribute to timing uncertainty, but mostly because they are comparatively large compared to the traces and thus space inefficient.

Of the 4-layers, only the 2 inside layers can couple between layers. Coupling on the inside layers is controlled by routing the serpentine traces on the different layers orthogonally, thus reducing the effective coupling length. Traces are also routed so that lines are not adjacent to the same line on a different layer for very long. These techniques are demonstrated in Figure 6.11

Coupling is primarily controlled by the large separation distance between the 2 inside layers.

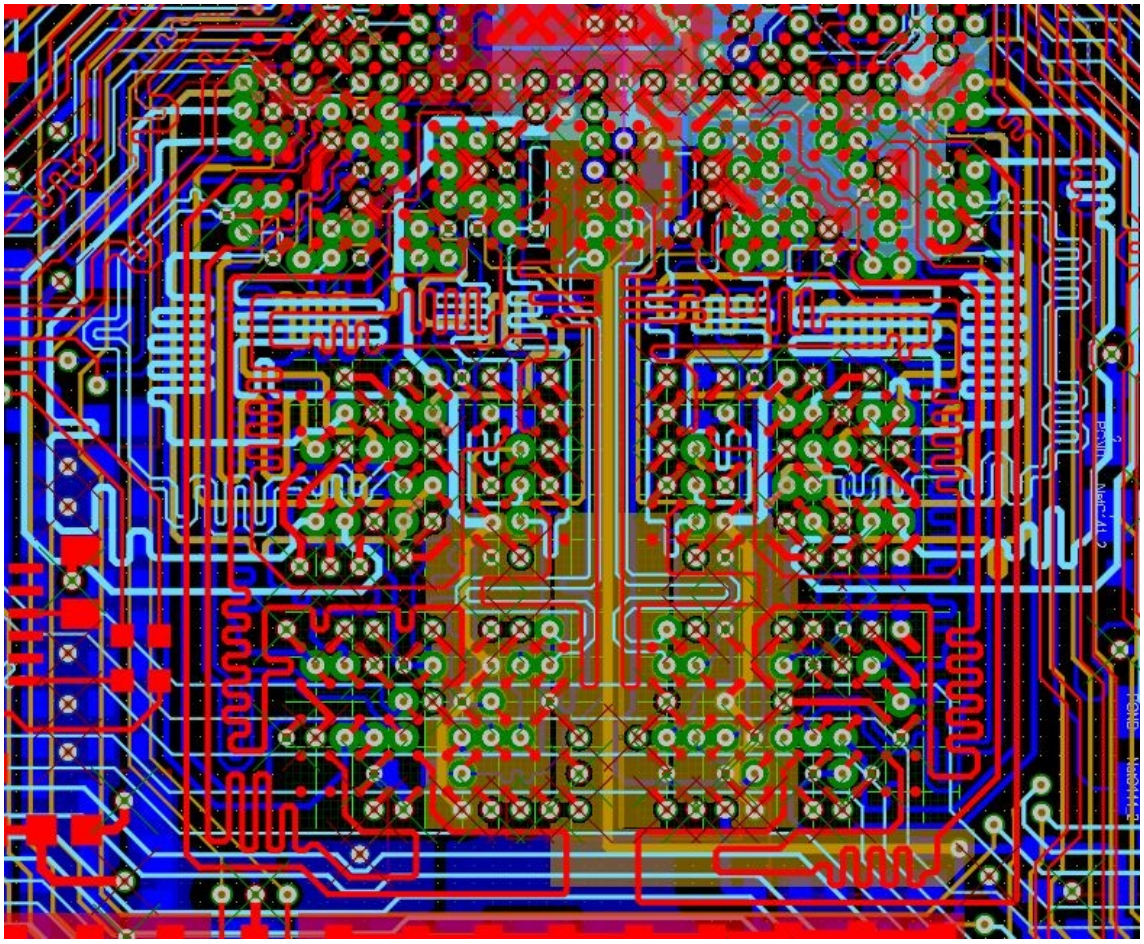


Figure 6.11: Orthogonal Routing on Inner Layers Minimise Crosstalk

6.8.2 Memory Timings

Memory timings are the acceptable margins for signals to propagate between the processor and memory devices. These margins are usually a result of the clock speed and setup- and hold times. Because the memory interface is relatively fast, much precision is required in laying out the printed circuit board. Table 6.20 shows the total allowable memory timing margins for each memory net class. These delays include the processor package delays. Processor package delays are shown in Table 6.21 - Table 6.22.

LPDDR4 Signal	**Group**	**Total Propagation Delay** (ps)		
		Min	**Max**	**Chosen**
CK_t/ CK_c	Clock	As short as possible	175	174-175
CA[5:0] CS[1:0] CKE[1:0]	Address/ Command/ Control	CK_t - 25	CK_t + 25	199-200
DQS0_t / DQS0_c	Byte 0 - DQS	CK_t - 85	CK_t + 85	250-251
DM0 DQ[7:0]	Byte 0 - Data	DQS0_t - 10	DQS0_t + 10	250-251
DQS1_t / DQS1_c	Byte 1 - DQS	CK_t - 85	CK_t + 85	250-251
DM1 DQ[15:8]	Byte 1 - Data	DQS1_t - 10	DQS1_t + 10	250-251

Table 6.20: Allowable Memory Timing Delays

Ball Name	**Memory Name**	**Ball Number**	**Delay** (ps)
DRAM_AC00	CKE0_A	AC16	36.5
DRAM_AC01	CKE1_A	AE17	47.4
DRAM_AC02	CS0_A	AE18	51.0
DRAM_AC03	CS1_A	AC18	47.8
DRAM_AC04	CK_t_a	AD14	42.7
DRAM_AC05	CK_e_A	AE14	41.5
DRAM_AC06	-	AE13	46.4
DRAM_AC07	-	AB15	40.0
DRAM_AC08	CA0_A	AD17	49.8
DRAM_AC09	CA1_A	AE16	73.5
DRAM_AC10	CA2_A	AD20	51.2
DRAM_AC11	CA3_A	AE20	52.6
DRAM_AC12	CA4_A	AD19	58.6
DRAM_AC13	CA5_A	AE19	56.2
DRAM_AC14	-	AB16	54.2
DRAM_AC15	-	AC15	47.9
DRAM_AC16	-	AE15	43.7
DRAM_AC17	-	AD15	43.0
DRAM_AC18	X	X	Pin does not exist
DRAM_AC19	MTEST	AB14	37.6
DRAM_AC20	CKE0_B	AD10	44.1
DRAM_AC21	CKE1_B	AE10	44.5
DRAM_AC22	CS1_B	AD8	55.5
DRAM_AC23	CS0_B	AC9	53.4
DRAM_AC24	CK_t_B	AD12	40.9
DRAM_AC25	CK_c_B	AE12	44.0
DRAM_AC26	-	AB12	39.6
DRAM_AC27	-	AA12	33.3
DRAM_AC28	CA2_B	AC7	51.1
DRAM_AC29	CA3_B	AE7	49.2
DRAM_AC30	CA4_B	AE6	58.1
DRAM_AC31	CA5_B	AD6	63.7
DRAM_AC32	CA0_B	AE8	47.6
DRAM_AC33	CA1_B	AE9	50.2
DRAM_AC34	-	AC10	49.2
DRAM_AC35	-	AB10	43.1
DRAM_AC36	-	AC12	38.2
DRAM_AC37	-	AE11	43.8
DRAM_AC38	-	AC11	48.2

Table 6.21: i.MX8MQ Processor package delays 1

Ball Name	Memory Name	Ball Number	Delay (ps)
DRAM_ALERT-N	MTEST1	AC13	40.4
DRAM_RESET_N	RESET_N	AB13	29.9
DRAM_DM0	DMI0_A	AD23	67.8
DRAM_DM1	DMI1_A	AB20	43.4
DRAM_DM2	DMI0_B	AD3	68.1
DRAM_DM3	DMI1_B	AB6	44.9
DRAM_DQS0_N	DQS0_C_A	AC25	70.46
DRAM_DQS0_P	DQS0_t_A	AC24	71.41
DRAM_DQS1_N	DQS1_c_A	AC21	45.29
DRAM_DQS1_P	DQS1_t_A	AB21	45.18
DRAM_DQS2_N	DQS0_c_B	AC1	68.78
DRAM_DQS2_P	DQS0_t_B	AC2	71.65
DRAM_DQS3_N	DQS1_c_B	AC5	51.56
DRAM_DQS3_P	DQS1_t_B	AB5	50.30
DRAM_DQ00	DQ0_A	AE23	65.9
DRAM_DQ01	DQ1_A	AD24	74.2
DRAM_DQ02	DQ2_A	AE22	56.3
DRAM_DQ03	DQ3_A	AD22	62.1
DRAM_DQ04	DQ4_A	AA24	74.3
DRAM_DQ05	DQ5_A	Y25	72.8
DRAM_DQ06	DQ6_A	AA25	65.7
DRAM_DQ07	DQ7_A	AB25	65.4
DRAM_DQ08	DQ08_A	AB22	44.2
DRAM_DQ09	DQ09_A	AA22	46.5
DRAM_DQ10	DQ10_A	AA23	53.8
DRAM_DQ11	DQ11_A	AA20	43.5
DRAM_DQ12	DQ12_A	AA18	44.9
DRAM_DQ13	DQ13_A	AB19	45.0
DRAM_DQ14	DQ14_A	AA19	35.3
DRAM_DQ15	DQ15_A	AA17	40.4
DRAM_DQ16	DQ0_B	AE3	60.5
DRAM_DQ17	DQ1_B	AD2	71.4
DRAM_DQ18	DQ2_B	AE4	67.0
DRAM_DQ19	DQ3_B	AD4	65.7
DRAM_DQ20	DQ4_B	AA2	75.9
DRAM_DQ21	DQ5_B	Y1	72.8
DRAM_DQ22	DQ6_B	AA1	68.0
DRAM_DQ23	DQ7_B	AB1	66.5
DRAM_DQ24	DQ08_B	AB4	50.6
DRAM_DQ25	DQ09_B	AA4	48.3
DRAM_DQ26	DQ10_B	AA3	58.9
DRAM_DQ27	DQ11_B	AA6	38.5
DRAM_DQ28	DQ12_B	AA8	37.2
DRAM_DQ29	DQ13_B	AB7	45.7
DRAM_DQ30	DQ14_B	AA7	32.2
DRAM_DQ31	DQ15_B	AA9	28.3

Table 6.22: i.MX8MQ Processor package delays 2

6.9 Simulation

The Bokkiebord is designed and manufactured without the aid of a simulator, but simulated after being completely debugged to evaluate the quality of the routing for future design. The simulated results are generated by the Advanced Design System from Keysight after a license has been obtained [45].

6.9.1 Signal Integrity

Signal integrity simulation is performed by characterising the system at $4GHz$ and setting the clock frequency to $1.6GHz$. The results show that reflections are well within limits as a result of good impedance design. Rise times are good with the eye diagram confirming adequate timing margin as a result of solid reference planes and and a very close reference plane spacing. Crosstalk is mostly within limits, but there is some room for improvement on a few lines especially at high frequencies. The higher than expected crosstalk is mostly attributed to the very high signal density per layer.

6.9.1.1 Crosstalk

Figure 6.12 shows the transmitted and received waved on the 1st address line of the 1st channel on the LPDDR4 memory. The reflected power is approximately 0.5%, which is better than expected.

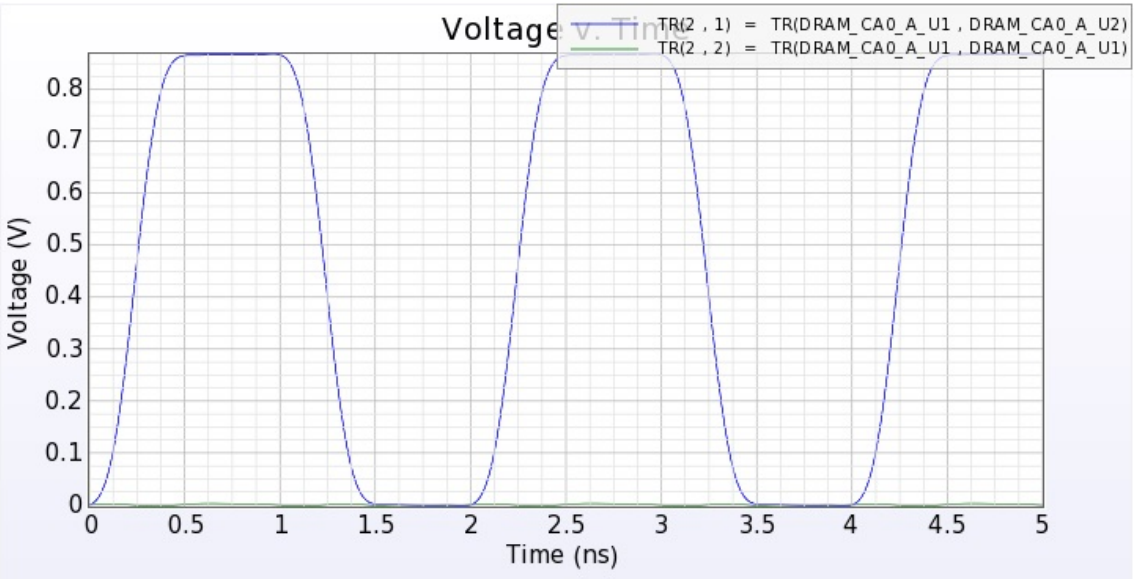


Figure 6.12: Transmitted and reflected waves of the CA0 line which is the 1st address line on the 1st channel of the LPDDR4 Memory

Figure 6.13 shows the crosstalk of different memory lines on each other for different switching frequencies. Most results are found to be good with low coupling coefficients. The worst coupling case appears around $1.6GHz$ where the coupling coefficient is about $-17.5dB$ or 13.3%.

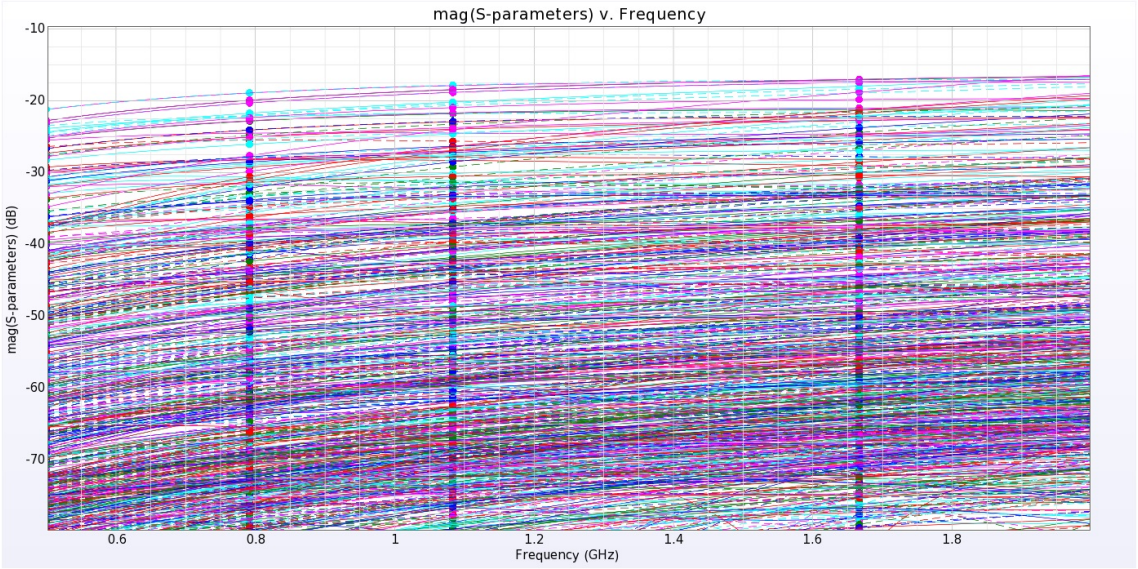


Figure 6.13: Memory crosstalk of each line on every other line.

The hardware developers guide requires an eye mask of $155mV$ high by $380ps$ wide. Figure 6.14 shows the simulated eye diagram result. The result shows that the eye height is approximately $350mV$ and the eye width is approximately $450ps$, which is well within the requirement.

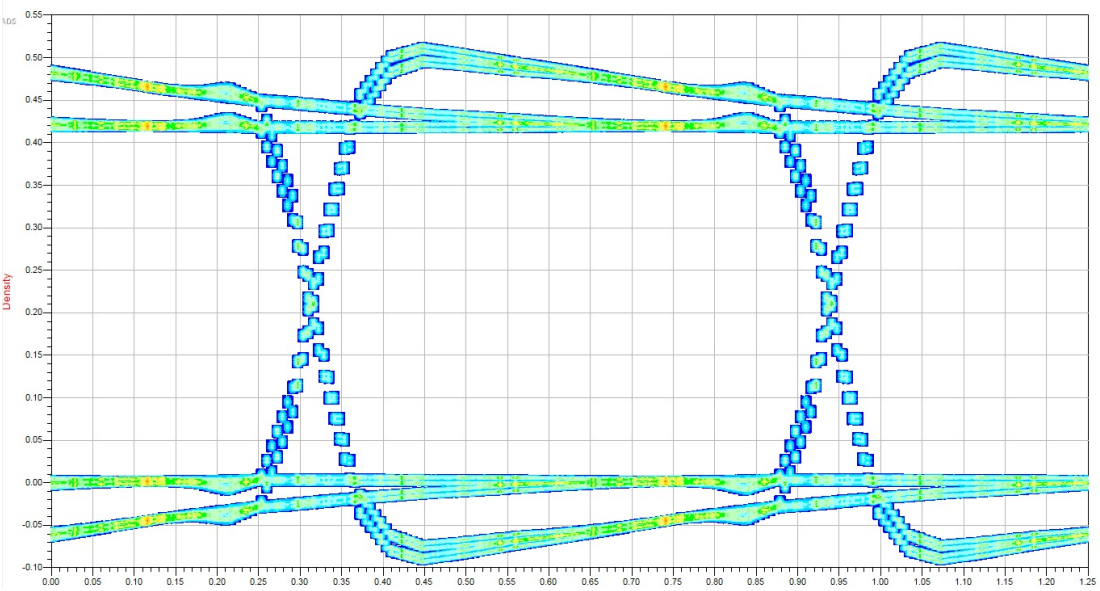


Figure 6.14: Eye diagram of the CA0 line indicating adequate timing margin.

6.9.2 Power Integrity

Power integrity simulation is performed at the maximum possible current draw and voltage drop is then determined. The result are summarised in Table 6.23. A voltage drop of up to 5% is considered to be acceptable and is achieved on all voltage rails except one.

The VDD_0V9 voltage rail has a voltage drop of approximately $87.9mV$ or 9.77% at the maximum current draw of $6.625A$. This means that the lowest voltage of this rail could be $0.8121V$ which is more than expected but still within acceptable limits as shown in Table 5.1.

Refer to Section A.4 in the appendix for figures showing simulated voltage drop. Voltage drop of this rail is due to too few vias being used when transitioning between power layers.

Voltage Rail	**Nominal Voltage** (V)	**Maximum Current** (A)	**Worst Case Voltage Drop** (mV)	**Worst Case Voltage Drop (%)**
VIN	5.000	10.000	10.700	0.214
VSYS_5V	5.000	10.000	33.500	0.670
NVCC_DRAM	1.100	2.000	48.200	4.380
VDD_0V9	0.900	6.625	87.900	9.770
VDD_SOC	0.950	2.500	21.400	2.250
NVCC_XXX	3.300	1.415	40.000	1.210
VDD_PHY_0V9	0.900	0.132	6.400	0.710
VDD_PHY_3V3	3.300	0.158	4.900	0.150
VDDA_1P8	1.800	0.167	10.100	0.560
NVCC_SNVS	3.300	0.005	0.500	0.020
VDD_SNVS	0.900	0.005	0.500	0.060
VDD_ ANALOGUE _3V3	3.300	0.042	3.800	0.120
VDD_PHY_5V	5.000	3.300	145.500	2.910

Table 6.23: Power Distribution Network Voltage Drop at Maximum Current Draw

Chapter 7

Manufacturing

7.1 Manufacturer selection

The **Electronic Engineering Department at the University of Stellenbosch** can produce simple prototype PCBs in-house that have up to 2-layers, $0.4mm$ vias and $0.2mm$ spacing. Fine soldering microscopes and equipment are available to students, and a professional soldering service is also available at the department.

More complex PCBs are sometimes manufactured by **Trax Interconnect** in Cape Town. Trax can manufacture PCB's with up to 16-layers with $0.25mm$ vias, and $3mil/3mil$ width/spacing [42]. A wide range of suitable stackup materials are available, and the quality of workmanship and support is very good. Pricing is determined by the manufacturing panel with different pricing structures available depending on the manufacturing timeline required. Trax also offers student discounts to universities.

Assembly and rework of PCBs is done by **Baracuda Holdings** in Somerset-West. The quality of workmanship and customer service of the company is found to be very good [23].

PCBWay in China is a one-stop-shop for PCB manufacturing, component sourcing and assembly services. PCBway has very advanced manufacturing capabilities [70] and, due to their global volume and economy of scale, are very price competitive.

PCBWay is ultimately chosen to manufacture the Bokkiebord due to their ability to manufacture $0.2mm$ via holes.

7.2 Manufacturing of The Power System Prototype

Because of the very high level of integration of the Bokkiebord, a prototype implementation of the Bokkiebord power system is first manufactured to facilitate debugging.

This prototype consists of a complete implementation of a TPS54A24 input stage and a TPS659037 Power management integrated circuit along with the required CMOS voltage translation circuitry.

The PCB design has 6-layers of a standard Trax stackup. The PCB has $0.25mm$ vias and the only controlled impedances are that of the crystal oscillator on the back of the board.

The PCB is manufactured by Trax and the BGA PMIC is placed by Baracuda Holdings. The rest of the circuit is populated by the student with the aid of a soldering microscope. The fully assembled power system PCB is shown in figure 7.1.

The University of Stellenbosch Forestry Department offers an x-ray service where x-rays are obtained for debugging purposes. The quality of BGA alignment can be evaluated by how well the vias appear between the dark BGA solder balls as copper pads do not distinctly appear in the x-rays. The component pictured in figure 7.2 is perfectly aligned with no short-circuiting between the solder balls.

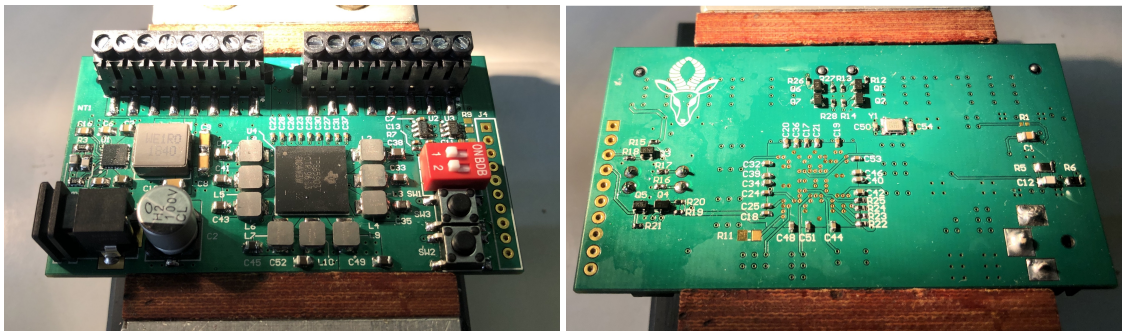


Figure 7.1: Power System Prototype

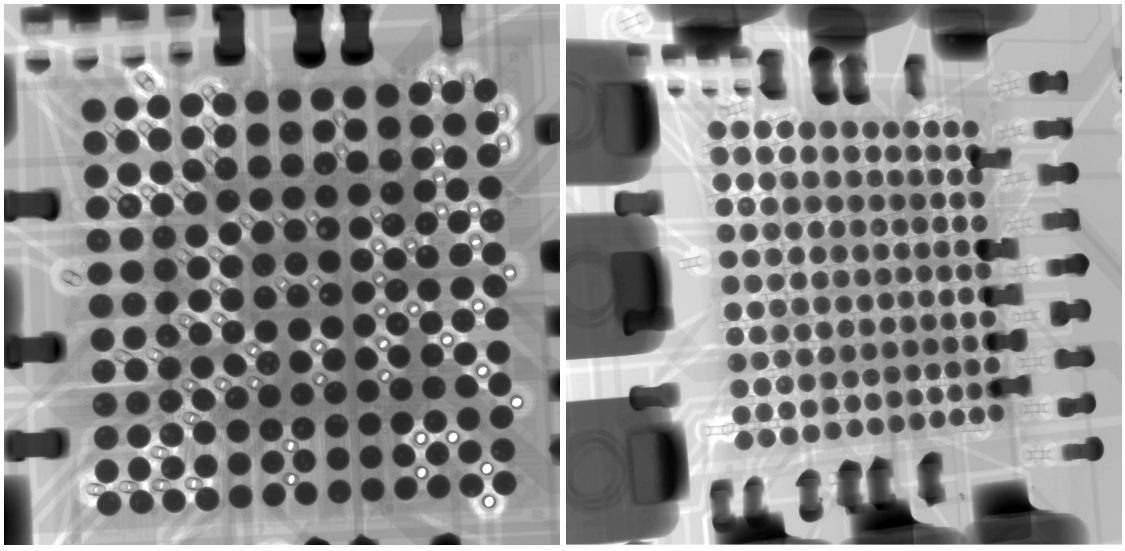


Figure 7.2: Power System Prototype BGA X-rays

7.3 Manufacturing of the Bokkiebord

Figure 7.3 shows the planned design of the Bokkiebord. The Bokkiebord is manufactured by PCBWay. The Bokkiebord measures 86.36mm x 55.88mm, consists of a 6-layer custom stackup, 0.2mm vias and 1oz copper on inner and outer layers.

The components for Bokkiebord are sourced and placed by PCBWay. Three Full Bokkiebord SBCs are sub-assembled by PCBWay.

Other components as small as 0201 resistors are placed by the student at the University of Stellenbosch' E&E fine electronics workshop.

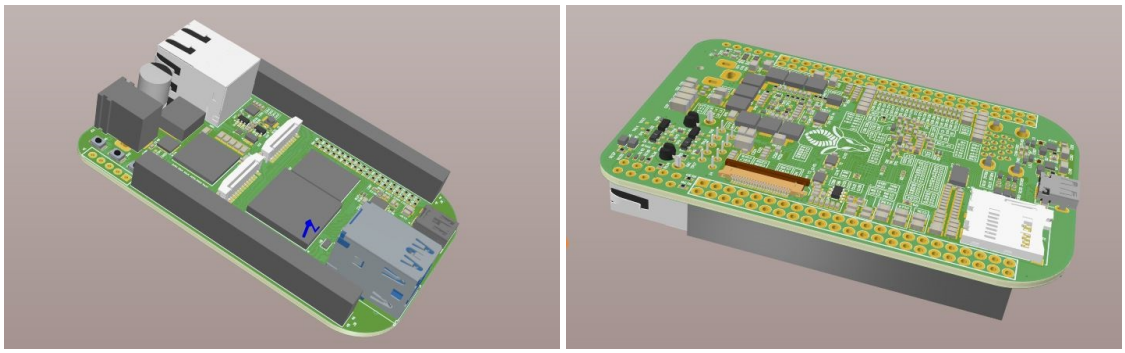


Figure 7.3: Bokkiebord Stackup Cross Section

7.3.1 Stackup Cross Section

Figure 7.4 shows a cross-section of a Bokkiebord PCB. Complete stackup information is presented in table 6.10.

The cross-section shows 6 copper layers. The outer layer is spaced $0.073mm$ from the solid copper reference plane and inner layer is spaced $0.128mm$ from the outer copper reference plane. A thick inner FR4 dielectric is visible. A $0.2mm$ through-hole via is visible on the far left of the figure.

Note how wide the $0.1185mm$ 50Ω trace appears in relation to its spacing distance from the ground plane. This spacing requirement necessitates using the wide-case equations for calculating trace width as indicated in the stackup section.

Copper inner layers would be placed between the thick dielectric if a 10-layer stackup was required.

Figure 7.4: Bokkiebord Stackup Cross Section

7.3.2 Component Placement

Figure 7.5 - Figure 7.9 show photos received from the PCBWay factory that show component orientation and placement quality. Only components deemed too difficult to place by the university are placed by PCBWay to contain manufacturing costs. PCBWay places a total of 19 components.

Mounted components are from left to right (Top Side):

- PCMF3USB3SZ USB ESD Protection and filtering
- PCMF3HDMI2SZ HDMI ESD Protection and filtering
- CP2105 Serial to USB converter
- MIMX8MQ6DVAJZAB Processor
- MT53E512M32D2NP-046 WT: E 2GB LPDDR4 Memory module
- TPS6590378ZWSR PMIC
- ADS7046IRUGR ADC
- DP83825 Ethernet PHY
- TPS54A24 Buck Converter

From left to right (Bottom Side):

- 473521001 microSD Card Holder
- 685119248223 micro HDMI Connector
- NX5P3290UKZ Current limiting power switch
- XUO536027 27MHz HCSL Oscillator
- CX2520DB27 27MHz crystal
- FA-20H 25MHz Crystal
- FA-238 16.3840MB-K 16MHz crystal

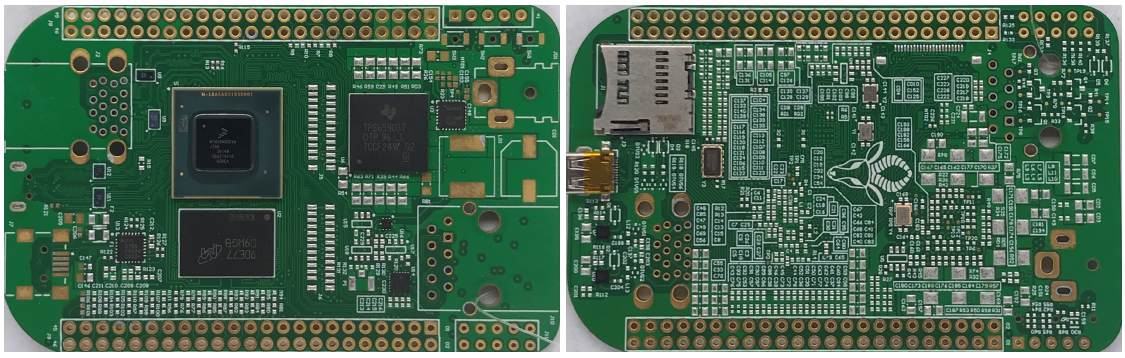


Figure 7.5: Bokkiebord After BGA Placement

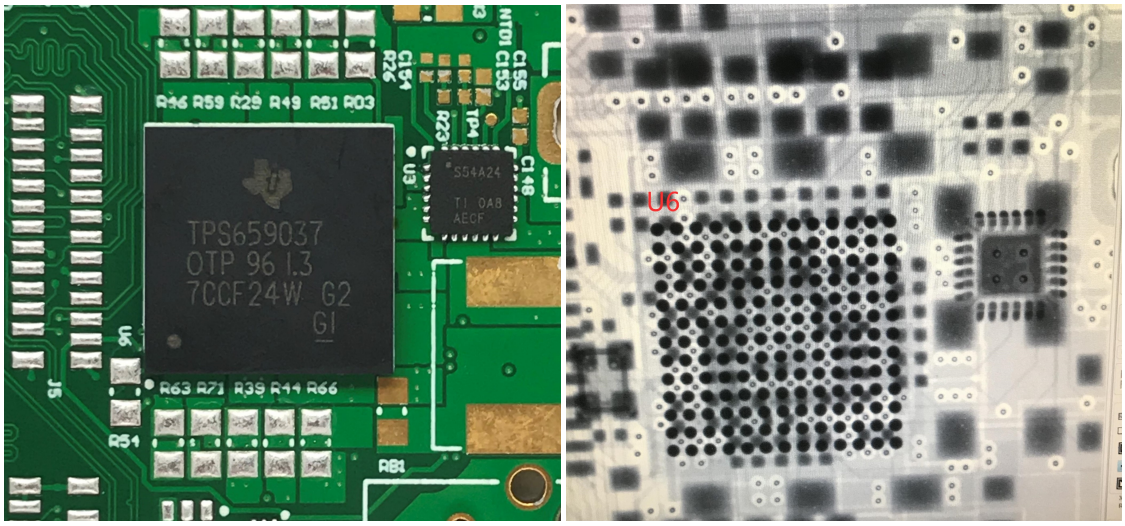


Figure 7.6: Bokkiebord PMIC Placement

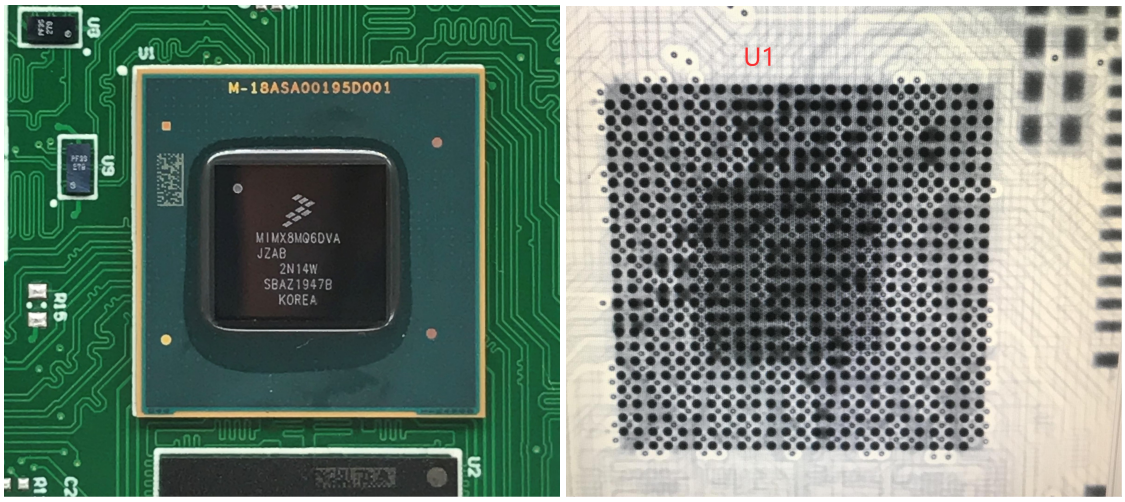


Figure 7.7: Bokkiebord Processor Placement

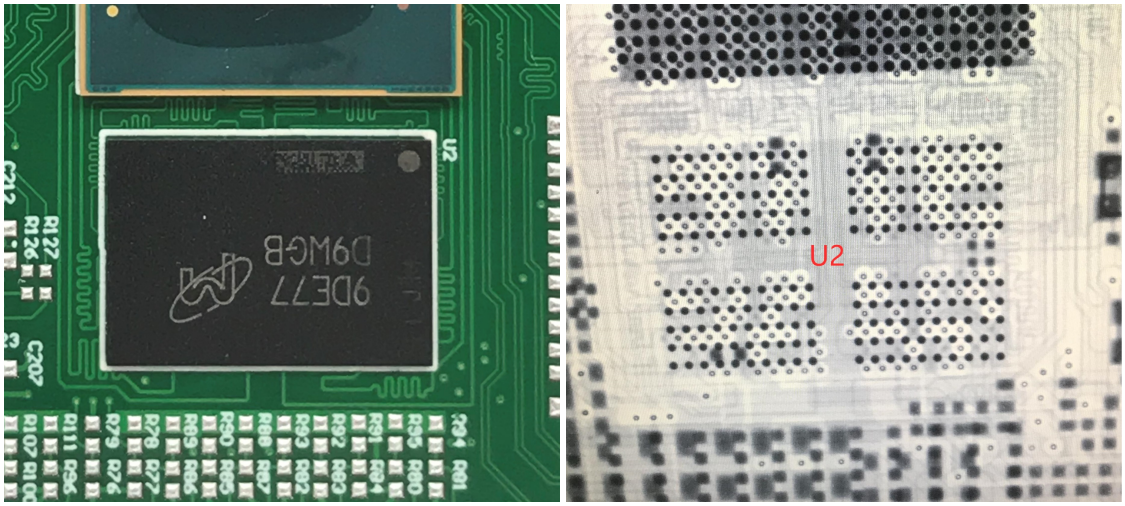


Figure 7.8: Bokkiebord Memory Placement

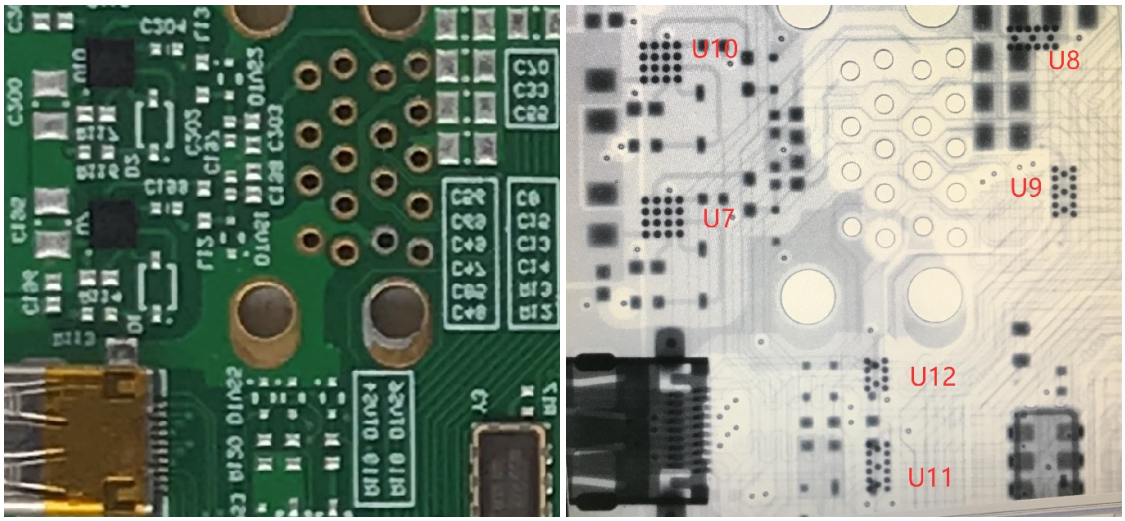


Figure 7.9: Bokkiebord After BGA Placement

7.3.3 Ultrasonic Cleaning

After surface mount component placement, excess solder flux is removed by ultrasonic cleaning.

Ethanol with a purity greater than 97% is used as a cleaning liquid. Isopropyl alcohol with a purity greater than 98% can also be used, but results are considerably better with ethanol. Water-based cleaning liquids are strongly discouraged as they might oxidize components.

Ultrasonic cleaning is done for 30m. After cleaning, the PCB is baked in an oven at $60^{o}C$ for 30 minutes to remove excess moisture as shown in Figure 7.10.

Figure 7.10: Ultrasonic Ethanol Cleaning of Bokkiebord to Remove Flux and Excess Solder

7.3.4 Completed Bokkiebord

Figure 7.11 shows a complete Bokkiebord after all component placement and debugging have been performed.

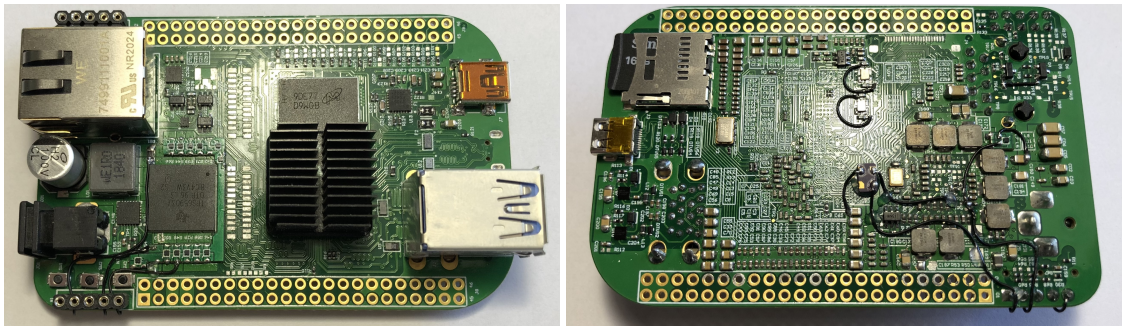


Figure 7.11: Final Bokkiebord

Chapter 8

Debugging and Testing

Each of the subsystems of the Bokkiebord is built and tested independently before working together in the larger Bokkiebord. Section 8.2 shows the final board bring-up procedure. Problems are identified and solved during the debugging process. Refer to Section 8.3 for more specific information about the circumstances that caused these issues.

8.1 Oscilloscope continuity test method

Many of the voltage rails on the Bokkiebord have absolute maximum ratings lower than $1V$, however most multimeters have continuity test voltages of about $2.3V$ that will immediately destroy the components on the Bokkiebord. It is thus necessary to have a testing method that enables the test voltage to be set as required.

This can be done with a bench power supply and an oscilloscope. The power supply is set to its minimum voltage and current limits. These voltage- and current limits are then verified independently with a multimeter. The power supply ground is connected to the ground of the oscilloscope. A multimeter test lead is inserted into the positive terminal of the power supply to act as the positive test probe and the oscilloscope probe acts as the negative test probe.

The test voltage is induced and measured relative to the same ground that only exists between the power supply and oscilloscope. When the positive test lead is touched to the oscilloscope probe, the oscilloscope measurement jumps from $0V$ to the value set by the power supply, thereby indicating continuity. In this case, the power supply can reliably be set to about $40mV$ at $30uA$.

A very small potential is indeed introduced to the circuit under test when testing, but as no ground reference exists on the circuit, the potential of the whole circuit is raised to a higher potential with no explicit potential difference being introduced on the circuit and current flows that could potentially damage the circuit.

The only return path for the current is through the high $1M\Omega$ impedance of oscilloscope probe. In this case the test voltage and current is about $40mV$ at $40nA$ which is less likely than a standard multimeter to damage the circuit under test. Figure 8.1 shows the oscilloscope continuity test method setup.

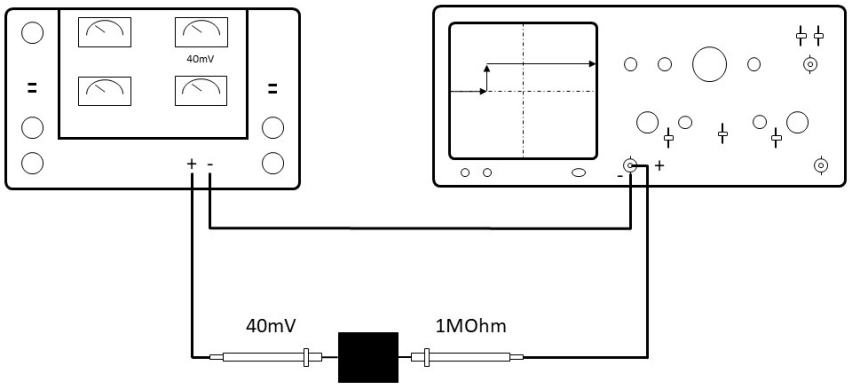


Figure 8.1: Oscilloscope Continuity Test Method

8.2 Board Bring Up procedure

The incomplete Bokkiebord circuit boards are received with BGA components already mounted as shown by pictures in Chapter 7. Additional components are placed by the student by hand. The Board bring-up procedure documents the steps involved in assembling and testing a complete Bokkiebord.

8.2.1 Building and debugging the first power system input stage

- **Except for the RB1 bridge resistor**, the 1st stage of the power system is built.
- Measurements are made for short circuits at the input and output of the 1st stage of the power system with the oscilloscope continuity testing method shown in Section 8.1.
- The 1st stage of the power system is connected to a laboratory power supply via the barrel jack. Care is taken to ensure that the positive pole of the barrel jack is located on the inner pin of the connector.
- The current limit of the laboratory power supply is set to $30mA$.
- Input voltage is ramped slowly from $0V$ to $7V$ while measuring input current consumption and open circuit output voltage on the 1st pin of the RB1 bridge resistor. The The output remains below $5V$ for inputs up to $12V$. Reference measurements for Bokkie-1A are given in Table 8.1 below. The test would be stopped immediately if current consumption exceeds $30mA$.

Input Voltage (V)	**Input Current** (mA)	**Output Voltage** (V)
0.5	0.00	0.00
1.0	0.00	0.00
1.5	0.00	0.00
2.0	0.01	0.00
2.5	0.01	0.00
3.0	0.01	0.00
3.5	0.015	0.00
4.0	2.31	4.00
4.5	2.84	4.50
5.0	9.55	5.00
5.5	12.65	5.00
6.0	14.0	5.00
6.5	19.50	5.00
7.0	27.3	5.00

Table 8.1: Typical Current Consumption of only the 1st Power Input Stage.

8.2.2 Building and debugging the PMIC

- Except for the L3 Inductor, PMIC power-, real time clock output-, I2C supply input-, I2C processor-, enable- or power good bridge resistors, the PMIC stage of the power system is built.
- Measurements are made for short circuits to ground at the input and outputs PMIC with the oscilloscope continuity testing method shown in Section 8.1.
- A temporary mod-wire is added from the VIO pin (2nd pin from top) on the power system debug header to the output of L10 as indicated in Figure 8.2. L10 is part of the SMPS8 switching regulator subsystem. For the TPS6590378ZWSR this voltage rail is 1st in the power sequence and produces $1.8V$. This $1.8V$ is used to power the VIO internal digital blocks within the TPS659037 during the early debug stages and is required to prevent latch-up. The VIO internal block can be operated in $1.8V$ or $3.3V$ modes.

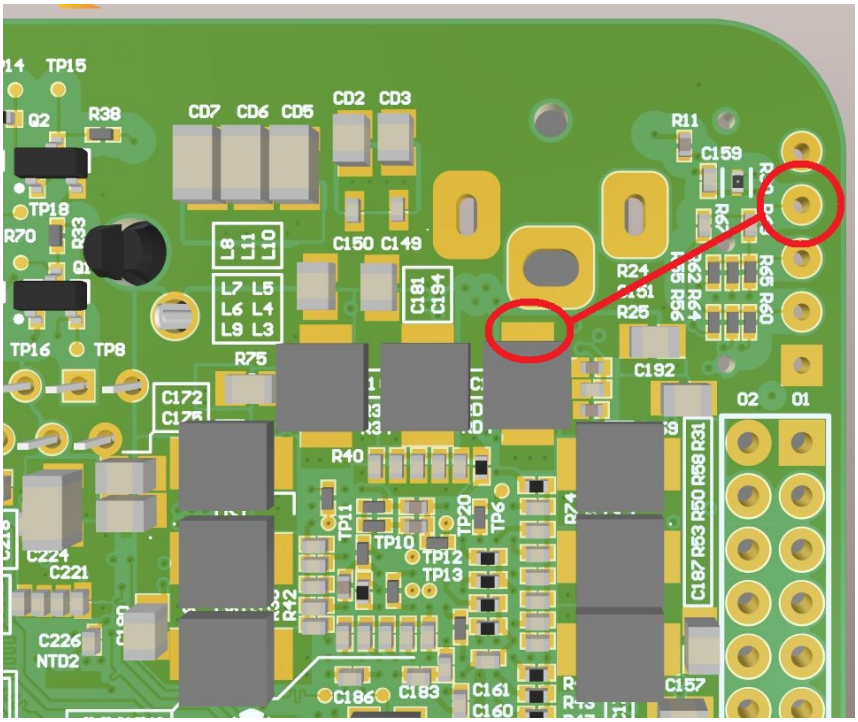


Figure 8.2: VIO Pin Shorted To L10 Output for Early Debug

- A temporary mod-wire is added to the 2nd port of the RB1 resistor.
- The positive supply of the laboratory power supply is connected to this mod-wire and the negative supply of the laboratory power supply is connected to the barrel jack.
- The current limit of the laboratory power supply is set to $60mA$.
- Input voltage is ramped slowly from $0V$ to $5V$ while measuring input current consumption and SMPS8 output voltage. It is recommended that input voltage not exceed $5.25V$. The absolute maximum voltage rating is $6V$. [39, page 13]. Reference measurements for Bokkie-1A are given in Table 8.2. The test is stopped immediately if current consumption exceeds $60mA$.

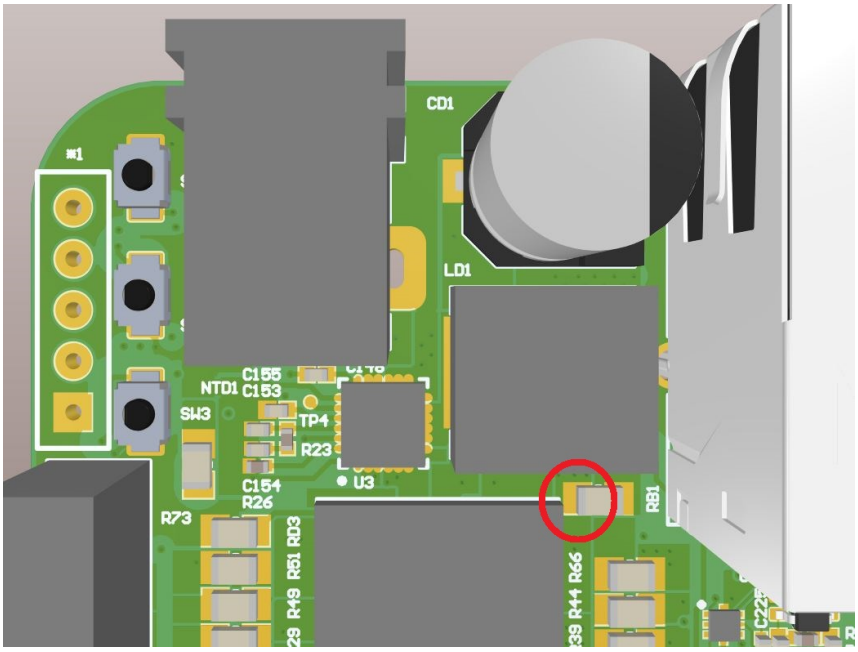


Figure 8.3: Mod-wire Added to VSYS for Early Debug

Input Voltage (V)	**Input Current** (mA)	**SMPS8 Output Voltage** (V)
0.5	0.00	0.00
1.0	0.34	0.00
1.5	1.39	0.00
2.0	3.30	0.00
2.5	3.50	0.00
3.0	3.70	0.00
3.1	39.3	1.80
3.5	37.0	1.80
4.0	39.7	1.80
4.5	42.6	1.80
4.8	44.1	1.80
4.9	44.8	1.80
5.0	45.2	1.80

Table 8.2: Typical Current Consumption of only the PMIC

- If the previous test is passed and all power output voltages are as described in the application note [41, page 14], the PMIC is functioning correctly.
- (Optional) If no output voltages are present, but input current consumption is within specification it is possible that the PMIC is simply not in active state. Measure the VRTC voltage level on the top of the reset switch to confirm that internal logic is active. Press the poweron switch and verify that current consumption increases significantly, but remains within specification.

- The PMIC is powered down if output voltages and input current consumption is as expected.
- Power is removed from the Bokkiebord.
- (Optional) If output voltages remain $0V$, but input current consumption remains within the specification in Table 8.2, it is possible that a configuration resistor has been placed incorrectly or that the OTP programming differs from the one used in this design. Refer to Section 5.1.2.2 for more information on PMIC configuration resistors. If input current consumption is excessive and chip temperature rises significantly, it is possible that the chip has been damaged. Input and output impedances are measured carefully with a low test voltage multimeter to confirm. Input and output impedances should exceed a few hundred ohm as internal diode protection is present.

8.2.3 Debugging the combined power stages

- The RB1 bridge resistor is placed.
- The laboratory power supply is connected to the Bokkiebord
- The current limit of the laboratory power supply is set to $70mA$.
- Input voltage is ramped slowly from $0V$ to $5V$ while measuring input current consumption and SMPS8 output voltage. The test is stopped immediately if input current consumption exceeds $70mA$.
- The PMIC is powered via the 1st stage regulator. The 1st stage regulator switches on when input voltage exceeds $4.0V$. The output voltage immediately becomes about $4.0V$ which means that the PMIC also switches on immediately. The input current consumption is expected to be $39.7mA + 2.31mA = 42.01mA$. at $4.0V$ input and $45.2mA + 12.65mA = 57.85mA$ at $5.0V$.
- Output voltage is verified to be as described in the application guide [41, page 14] and the system resets when the reset button is pushed.
- The Bokkiebord is powered down.
- Power is removed from the Bokkiebord.

8.2.4 Debugging with power system debugger

- A permanent mod-wire is added from the 2nd port of the RB1 bridge resistor to the 1st pin on the power system debug port as shown in Figure 8.4. This pin is unused by default, but is now used to power the power system debugger.

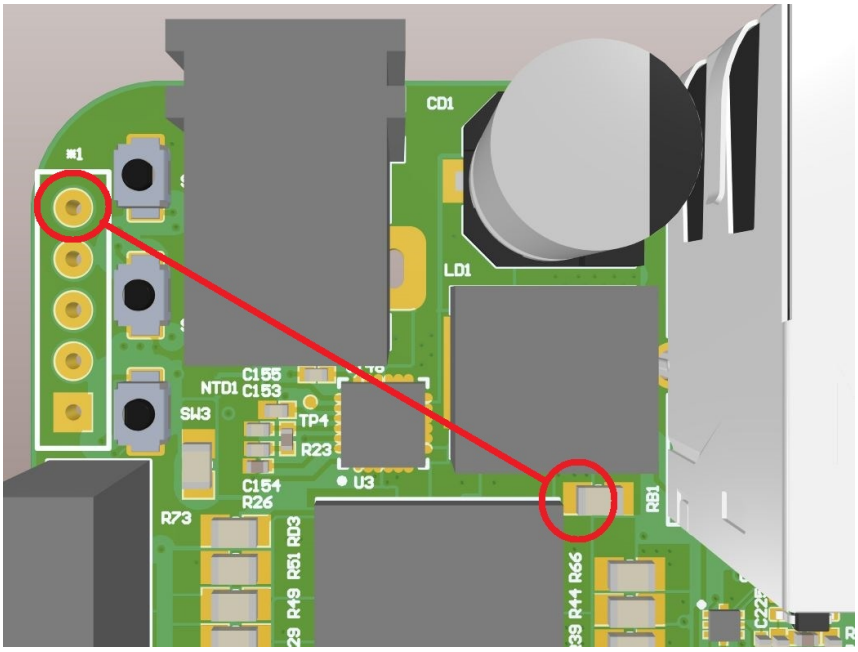


Figure 8.4: Permanent Mod Wire Between VSYS and Top Pin of Power System Debug Port

- The temporary VIO Mod-wire discussed in Section 8.2.2 is removed.
- The power system debugger is connected as shown in Figure 8.5.

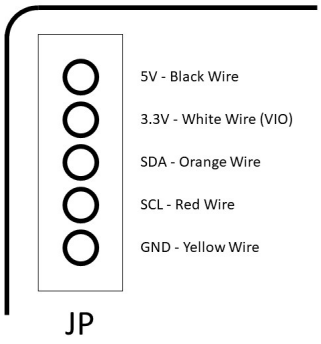


Figure 8.5: Power System Debugger Pinout

- The laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $70mA$.
- Input voltage is ramped slowly from $0V$ to $5V$ while measuring input current consumption and SMPS8 output voltage. The test is stopped immediately if input current consumption exceeds $70mA$.

- The I2C bus of the power system debug port is scanned and the PMIC answers on device address: (0x12, 0x48, 0x49, 0x4A, 0x4B) and (0x58, 0x59, 0x5A, 0x5B) [40, page 2].
- The short status register of the PMIC is read to test communication and to verify that no short to ground exist. This is done by sending the following command to the power system debugger: "0,58,49,0". A command consists of 4 values. The 1st indicates the type of operation - 0 is read, 1 is write, the 2nd is the device address in hexadecimal, the 3rd is the register address in hexadecimal and the 4th is the value to be written to the register in hexadecimal.
- The Bokkiebord power-down sequence is written to the PMIC by sending the following command to the power system debugger "4,00,00,00".
- The PMIC GPIO4 pin shown in Figure 8.6 is measured with an oscilloscope to verify that it is low.
- The Bokkiebord power-up power sequence is written to the PMIC by sending the following command to the power system debugger "5,00,00,00".
- The PMIC GPIO4 pin shown in Figure 8.6 is measured with an oscilloscope to verify that it is high.
- All power outputs are measured to comply with the values shown in Table 5.1.
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.5 Building the rest of the board

- The decoupling and peripheral sections of the Bokkiebord are built.
- Measurements are made for short circuits to ground at the outputs of the PMIC and the load side inputs of the rest of the board with the oscilloscope continuity testing method shown in Section 8.1.
- Continuity is confirmed between the load side pad of the bridge resistor footprint and the decoupling capacitor for each voltage rail under the processor on bottom side of the board with the oscilloscope continuity testing method shown in Section 8.1.
- The power system mod-board is built. The mod board is not yet mounted on the Bokkiebord at this time.
- Continuity between the input and output ports of the load switches on the mod-board is confirmed both when the enable line is high and when it is low. The load switches on the mod-board are active high and continuity only exists while the enable pin is high.
- Both parts of the mod-board are mounted on the Bokkiebord on the footprints initially intended for the bridge resistors. The enable line of the mod-board is connected to the output of GPIO4 on the PMIC as indicated in Figure 8.6.

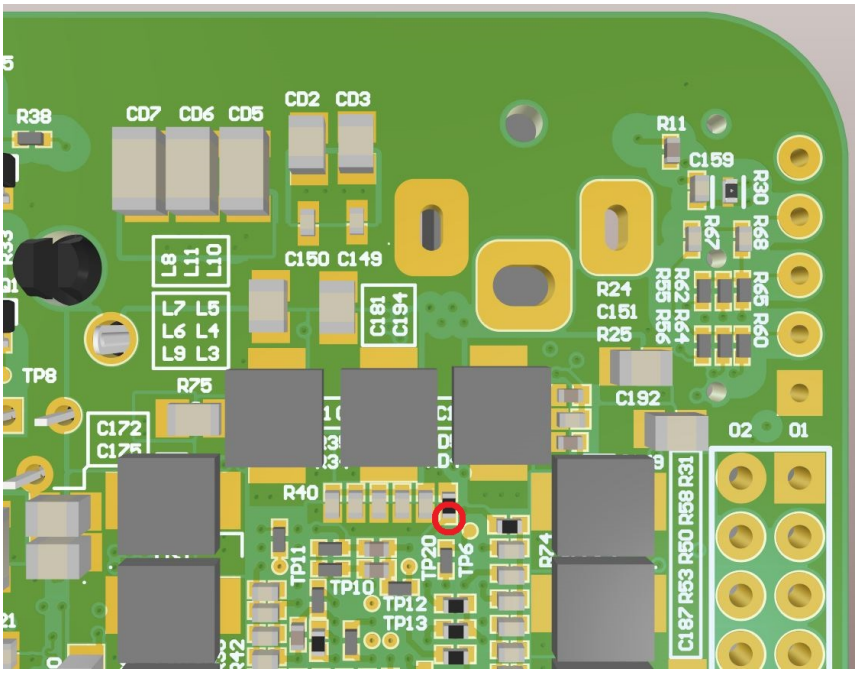


Figure 8.6: GPIO_4 output pad with a $10k\Omega$ 0402 resistor to ground

8.2.6 Powering the rest of the board

- Measurements are made for short circuits to ground at the source side of the mod-board and the load side inputs of the rest of the board with the oscilloscope continuity testing method shown in Section 8.1.
- Continuity is confirmed between the board side of the mod-board and the decoupling capacitor for each voltage rail under the processor on bottom side of the board with the oscilloscope continuity testing method shown in Section 8.1.
- The Laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $70mA$.
- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $70mA$.
- An oscilloscope is used to verify that the source side of the mod-board has voltages corresponding to the application note [41, page 14] and that the board side of the mod board has $0.00V$.
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger. The input current drops to about $20mA$.
- The current limit of the laboratory power supply is set to $280mA$.
- The Bokkiebord is started by sending the "5,00,00,00" command to the power system debugger. If continuity measurements are performed correctly the processor, memory and peripheral components start drawing current and warming up as if the Bokkiebord is in reset state. The total current draw of the board is approximately $240mA$.

- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.7 Starting the processor

- An external $32kHz$ oscillator is placed on the underside of the Bokkiebord and the output is connected to the 2nd pin of the CLK32GO pin on the underside of the PMIC as shown in 8.7. Power for the oscillator is obtained from the VDD_SNVS output of the mod-board. Reasons for this fix are provided in Section 8.3.4.

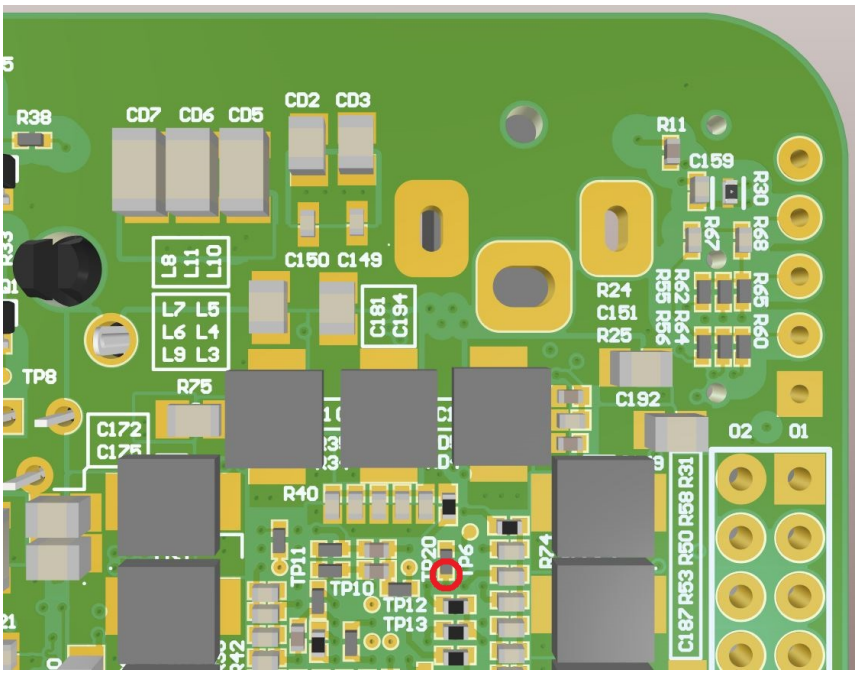


Figure 8.7: Real Time Clock Input Pin

- The Laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $380mA$.
- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $380mA$.
- The Bokkiebord is started by sending the "5,00,00,00" command to the power system debugger
- The Bokkiebord starts and begins drawing about $340mA$ which indicates that the processor is awake.
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.8 Establishing communication with a JTAG

- The Segger J-LINK EDU JTAG debugger is connected as indicated in Figure 8.8.

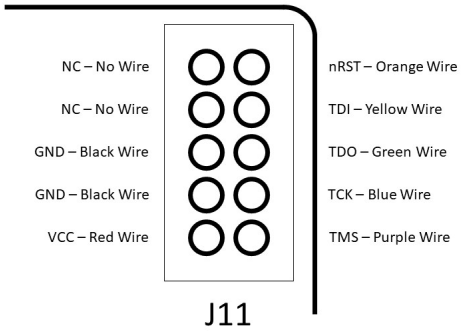


Figure 8.8: Power System Debugger Pinout

- The Laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $380mA$.
- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $380mA$.
- The Bokkiebord is started by sending the "5,00,00,00" command to the power system debugger
- The "connect" command is sent to the Segger J-LINK EDU debugger. The low-power ARM Cortex-M4 core of the i.MX8MQ processor is detected.
- (Optional) If the low power core is not visible, the "Test Data Out" pin voltage is measured with an oscilloscope. As this is a $3.3V$ digital interface a logical high or a logical low is expected. If the pin is left floating between logic levels it may indicate that the internal logic of the processor is in reset state. RTC input signal is verified to be valid as it is required to generate a valid reset signal.

8.2.9 Establishing communication with a UART-to-USB bridge

- Measurements are made for short circuits to ground on the power pin and data lines of the Micro-USB connector with the oscilloscope continuity testing method shown in Section 8.1.
- The host computer is connected to the Bokkiebord via the Micro-USB connector and the CP2105 is detected by the computer.
- The host computer is used to connect to a UART port on the CP2105 USB-to-UART bridge at 115200 baud with 8 data bits, 1 stop bit and no parity.

- A few characters are sent to the UART port on the USB-to-UART bridge while measuring the UART-TX line voltage of the CP2105 with an oscilloscope. The bits should be clearly visible and have an amplitude of $3.3V$.
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.10 Establishing communication with the USB OTG Port

- The the UART Bridge resistors are mounted.
- The Laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $380mA$.
- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $380mA$.
- The Bokkiebord is started by sending the "5,00,00,00" command to the power system debugger.
- The custom USB-OTG cable is connected to the bottom port of the dual USB 3.0 Connector. This custom USB-OTG cable consists of 2 USB type A connectors with the power line broken. The ground, D+ and D- lines of 1st connector is connected directly to the respective ground, D+ and D- of the 2nd connector while the power line is not connected.
- The Bokkiebord should appear to the computer as a HID device.
- (Optional) If the Bokkiebord is not detected by the computer, the USB 2.0 D+ and D- lines of the USB 3.0 connector are measured while the Bokkiebord is powered on to verify that the port is ready. If D- is 1 and D+ is 0 the Bokkiebord is indicating that it is set up as a low speed peripheral device to the computer. If D- is 0 and D+ is 1 the Bokkiebord is indicating that it is set up as a high-speed peripheral device to the computer. If D- and D+ is low Bokkiebord is indicating that it is not detecting a bus voltage. Verify that the VBUS Detection bridge resistor (R115) is populated. By default the bottom USB port is always set up as a USB-OTG port except when overridden by the device tree file. The top USB port is always a host port when the USB-ID bridge resistor is populated, otherwise it is an USB-OTG port.
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.
- The Bokkiebord is now ready to program.

8.2.11 Initialising Memory

- The excel memory initialisation configuration file is set up as shown in Figure 8.9.

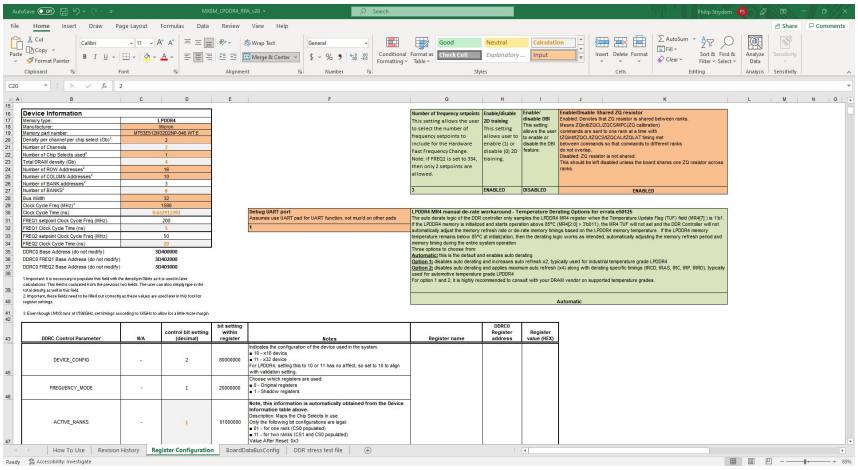


Figure 8.9: Memory Initialisation Configuration File

- The contents of the DDR STRESS TEST FILE tab is copied and saved in a text file with a .ds file extension.
- The Laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $380mA$.
- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $380mA$.
- The Bokkiebord is started by sending the "5,00,00,00" command to the power system debugger.
- The MScale DDR Tool [58] is opened and the serial to USB bridge port is selected. The above .ds file is chosen as input. Starting frequency is set as $100MHz$ and end frequency as $1580MHz$ as shown in figure 8.10.

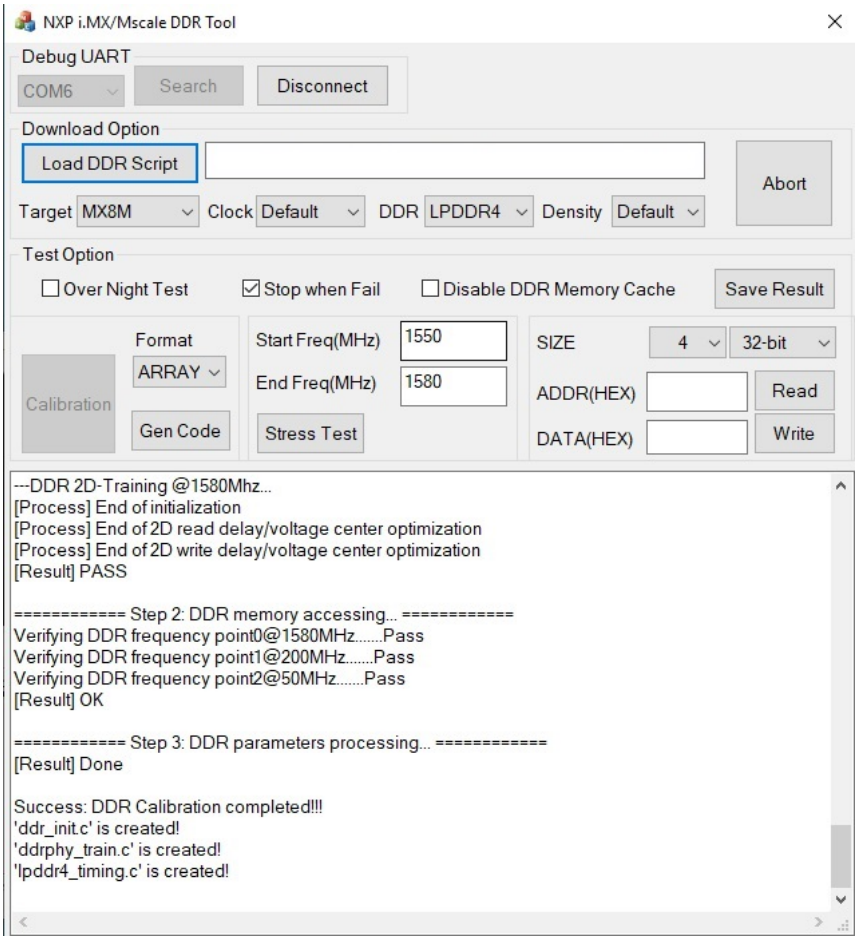


Figure 8.10: MScale DDR Tool Memory Training

- The "Calibration" button is pressed and memory training is performed.
- The "Gen Code" button is pressed to generate a lpddr4_timing.c file
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.12 Building the OS image

- The lpddr4_timing.c result from the memory initialisation step is used to generate a bootable image for the Bokkiebord as discussed in Chapter 9.

8.2.13 Loading the OS image

- A SD card is inserted into the SD card slot.
- The custom USB OTG- and the micro-USB the UART-to-USB bridge cables are connected.
- The Laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $380mA$.

- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $380mA$.
- The Bokkiebord is started by sending the "5,00,00,00" command to the power system debugger.
- Software is loaded over the USB OTG port with the use of the universal update utility by issuing the "uuu.exe uuu.auto" command to load the operating system software onto the Bokkiebord. The Bootloader flashes the new image to the SD card and attempts to start the system.

8.2.14 Negotiating with the bootloader

- The bootloader is interrupted by sending the enter command to the Bokkiebord via the serial to UART bridge.
- (Optional) The "mmc info" command is sent to verify that the sd card is visible to the Bokkiebord. Refer to Section 9.13 for more information on SD card configuration.

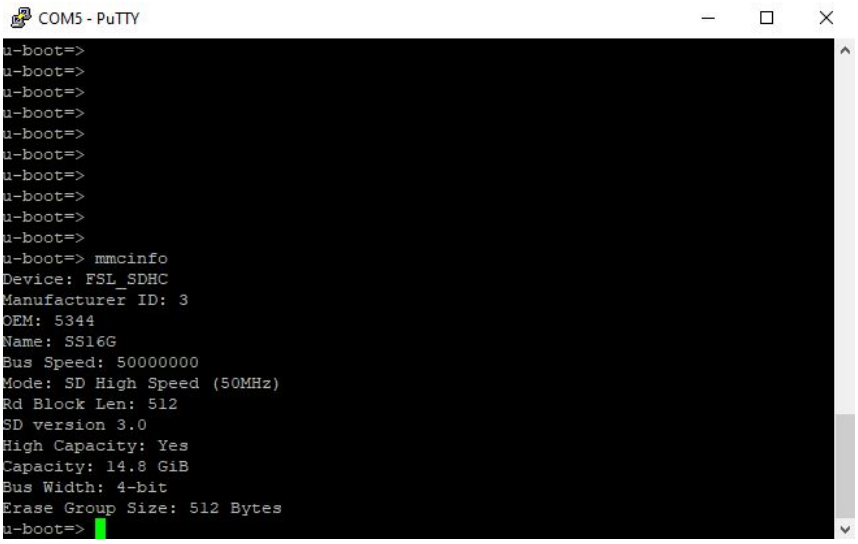


Figure 8.11: MMCINFO

- (Optional) The "mmc part" command is sent to verify that the SD card is readable and that the partition sizes are as expected.

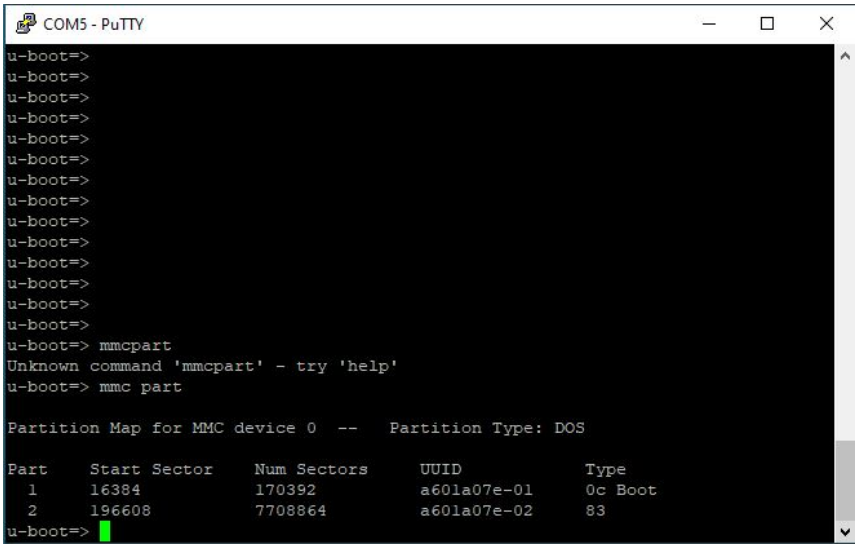


Figure 8.12: MMCPART

- (Optional) The "mii status" command is sent to verify that the Ethernet PHY is operating as expected and that it is visible to the Bokkiebord. Refer to refer to Section 9.14 for more information on setting up the Ethernet PHY in the operating system. enditemize
- (Optional) The boot device is set with the following command: setenv mmcroot "/dev/mmcblk0p2 rootwait rw" if not already set correctly, set. This command set the 2nd partition of the 1st SD card as the boot device. The "rootwait" modifier indicates that the Bokkiebord must wait for the storage device before erroring out. The "rw" modifier indicates that the boot device is readable and writable.
- The boot process is continued by issuing the "boot" command to boot the Bokkiebord. The bootloader loads the operating system into memory and starts the system.

8.2.15 Logging in to Linux

- A boot prompt appears when the boot process is complete.

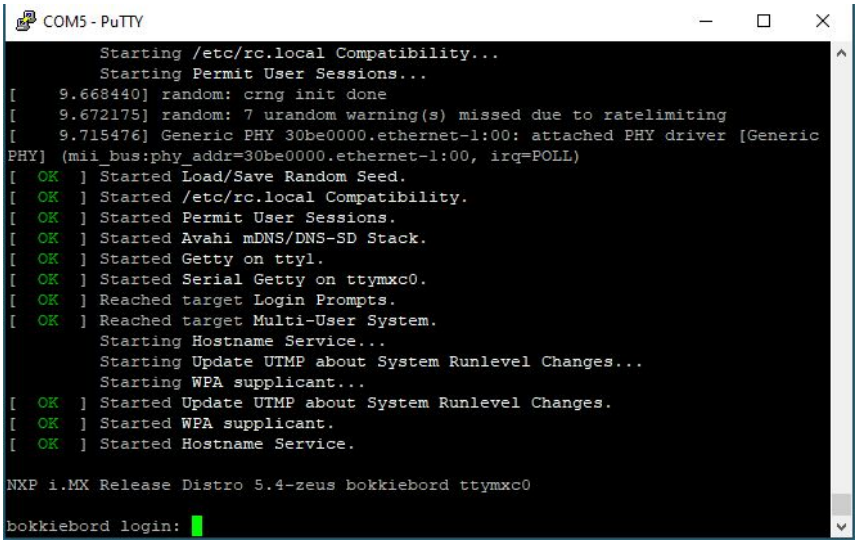


Figure 8.13: MMCPART

- The user logs in as root with no password.

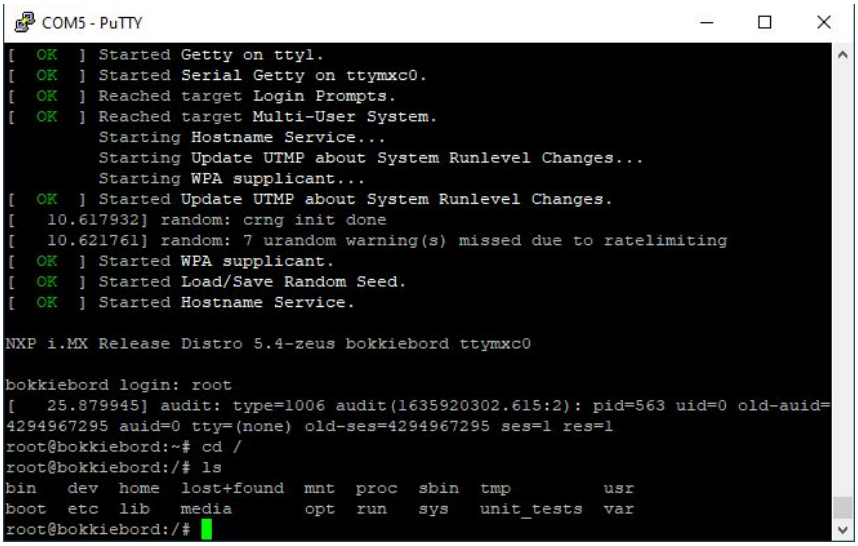


Figure 8.14: MMCPART

- The "lscpu" command is used to see cpu information.

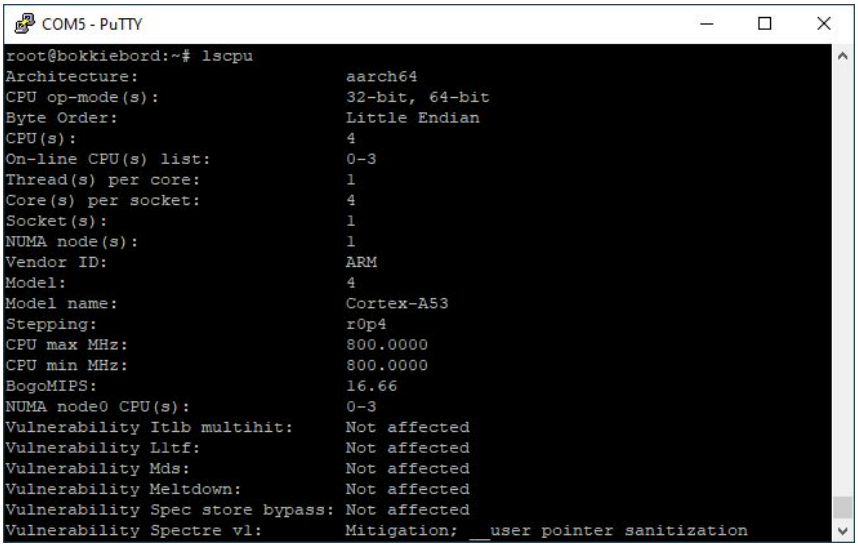


Figure 8.15: LSCPU

- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.16 Ethernet

- A network cable is connected to the RJ45 jack of the Bokkiebord.
- The Laboratory power supply is connected to the Bokkiebord.
- The current limit of the laboratory power supply is set to $380mA$.
- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $380mA$.
- The Bokkiebord is started by sending the "5,00,00,00" command to the power system debugger.
- The pins of the Ethernet Cable are measured with an oscilloscope to verify that no DC voltage is present on any of the lines. As an isolation transformer is required by the Ethernet specification and integrated into the Ethernet connector of the Bokkiebord.

8.2.16.1 Link Establishment

- The other end is connected a computer with a known good Ethernet port.
- Link establishment status is verified between the Bokkiebord and the Computer. Link establishment is the responsibility of the Ethernet PHY and can be done without intervention from the processor provided that the $50MHz$ input clock is running.
- (Optional) If a link cannot be established, reset and powerdown pins are inspected. Both pins are active low.

- (Optional) If a link cannot be established, the $50MHz$ input clock of the Ethernet PHY is probed to verify that it is running.
- (Optional) If a link cannot be established, the Ethernet cable lines are probed with an oscilloscope. A high frequency discovery pulse is sent at a regular interval by the PHY to determine if the line is terminated correctly or open. If a pulse is visible the problem is not electrical, but rather mechanical.

8.2.16.2 Processor to PHY Communication.

- PHY discovery is done at the start of the U-boot boot process. The Ethernet PHY must be ready to respond when the RMII and MDIO buses are probed. The PHY must be powered on, the powerdown pin must be set high and the reset pin must be set high and then toggled for at least a $25ms$ duration. Delays can be configured in the imx8mq-evk.c and spl.c files in the u-boot package.
- The PHY is configured over the MDIO bus and the RMII bus is initialised.
- The PHY is detected, a media access control (MAC) address is set and the device is bound to the device diver when the Linux Kernel Starts.
- (Optional) If the PHY is not detected, the "mdio list" u-boot command is used to determine if the MDIO bus is available.
- (Optional) If the MDIO bus is not available, the MDC line of the Ethernet PHY is probed with an oscilloscope. A $2.5MHz$ clock should be visible. If the clock is not visible, the device tree pin definitions are verified. Special attention is given to the pull-up resistor and drive level impedance. All Ethernet Impedances are 50Ω.
- (Optional) If the PHY is not detected, the "mii list" u-boot command is used to determine if the RMII bus is available.
- (Optional) If the RMII bus is not available, the RD0 and TD0 pins of the Ethernet PHY are probed with an oscilloscope for signs of activity.
- (Optional) If the PHY is not available the presence of the 2.4Ω resistor between the VDDIO and RD1 lines of the processor is verified. This strap resistor is required to place the PHY into RMII slave mode.
- Texas Instruments provides a good resource [32] for Ethernet debugging
- Available network interfaces are listed with the use of the "ifconfig" command.

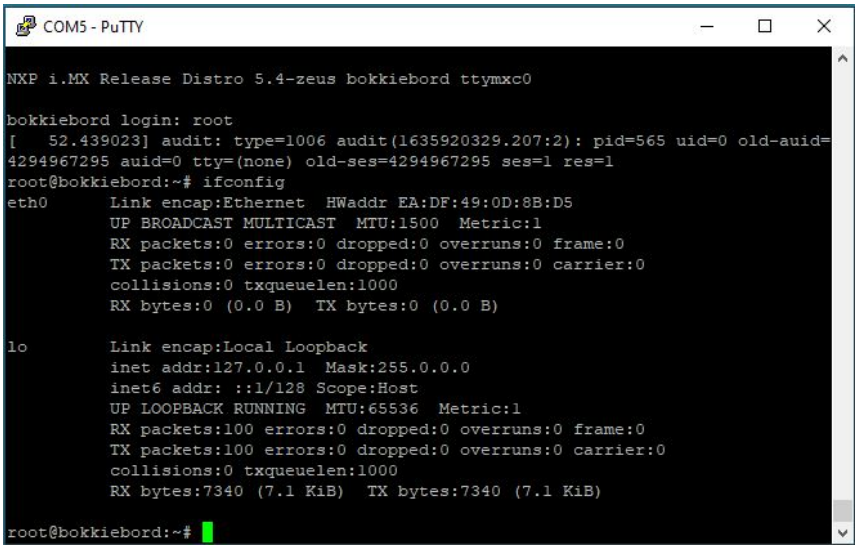


Figure 8.16: IFCONFIG

- Result: An Ethernet Link can be established and communication between the processor and the PHY is possible, but packets cannot be sent to the network. The Suspected reason is that the dp83822 Ethernet driver might not be supported.
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.17 HDMI

- A Micro-HDMI cable is connected to the Bokkiebord.
- The Laboratory power supply is connected to the Bokkiebord
- The current limit of the laboratory power supply is set to $380mA$.
- Input voltage is ramped slowly from $0V$ to $5V$. The test is stopped immediately if input current consumption exceeds $380mA$.
- Pins on the other end of the HDMI cable are measured with an oscilloscope to verify that voltage levels are as expected. Keep in mind that the pins on the cable are the mirror of the port.
- (Optional) If power pin voltages are not as expected the orientation of the HDMI power system protection diodes are verified.
- The current limit of the laboratory power supply to $600mA$.
- (Optional) If an image does not appear on the screen it is possible that the display device is set to DSI instead of HDMI. The U-Boot videolink command is used to determine display device. The "video_link" environmental variable is set to "HDMI@0x32C00000"
- (Optional) If an image still does not appear on the screen, the Hot Plug Detect Voltage is measured with an oscilloscope. Hot plug detect voltage should be between $3V-5V$. If the voltage is out of specification, the resistance of the $1M\Omega$ pull-down resistor is verified.

- Result: An HDMI connection could not be achieved. The Suspected reason is a low Hot Plug Detect Voltage generated by the monitor.
- The Bokkiebord is powered down by sending the "4,00,00,00" command to the power system debugger.
- Power is removed from the Bokkiebord.

8.2.18 ADC

The ADC is connected to the 2nd SPI interface of the i.MX8MQ processor. It is attached as a generic SPI device operating at $40MHz$ which can be increased to $52MHz$ by altering the device tree as described in Section 9.15. The device is abstracted in the Linux operating system as a file at the location "/dev/spidev1.0". This binary file is read by the simple python script shown in Figure 8.17. Python is built into the Bokkiebord operating system by default. Measurement results are shown in Figure 8.18.

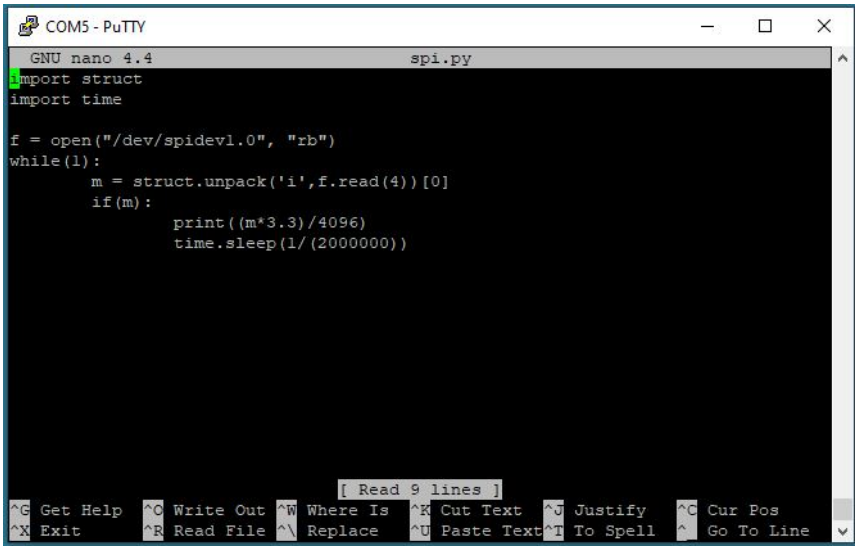


Figure 8.17: ADC Software

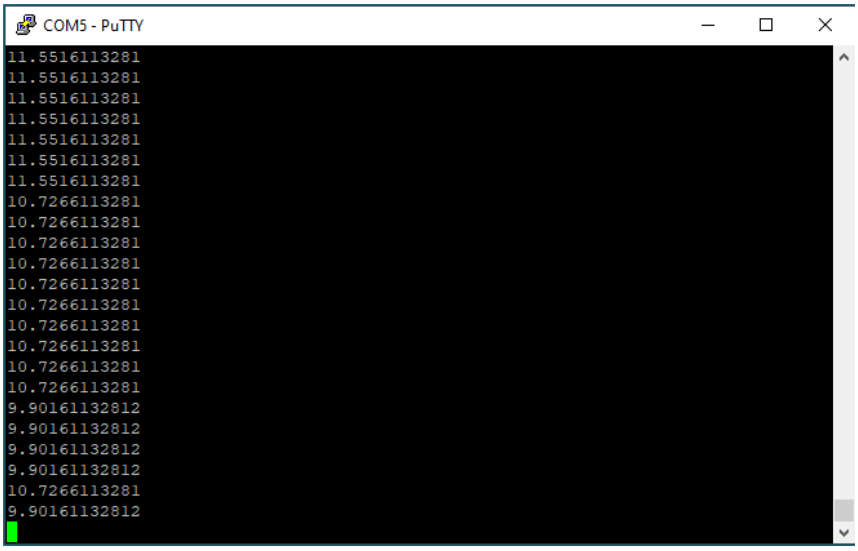


Figure 8.18: ADC Output

- Bokkiebord debug is now complete.

8.3 Bugs

8.3.1 Switching Regulator SMPS123 startup problem

8.3.1.1 Symptoms

A problem is identified when the PMIC is turned on for the 1st time. Switching regulator outputs initially measure high impedance, but measure 0.2Ω after the 1st power-up attempt. The PMIC also becomes very hot while inductors remain cool.

8.3.1.2 Diagnosis

The problem is identified by measuring the current flowing into the device. When the current flowing into the device is deemed to be excessive, output voltages are measured. The output voltage of SMPS123 is measured to be $0V$, and output impedance is measured to be as little as 0.2Ω.

The thermal images in Figure 8.19 show that current flowing into the device is dissipated entirely as heat. Heat is localised to the PMIC while the lack of temperature on the inductors indicate that the power system is not operating.

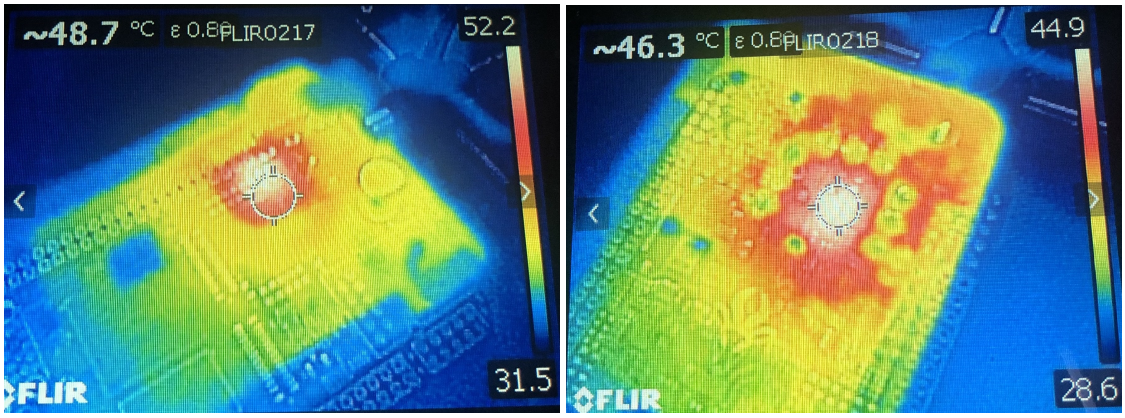


Figure 8.19: PMIC Thermal Image for Short Detection

2 Possible causes are suggested. Un-etched copper may exist under the PMIC or it is possible that the chip was destroyed by ESD or a multimeter measurement where the test voltage exceeded the absolute maximum rating of the components.

Multimeters have a test voltage that can range between $0.6V$ in resistance test mode and $2.3V$ in continuity test mode. More advanced multimeters like the Fluke 175 have a 2-stage test procedure where the final test voltage could exceed $7V$ in the final stage of test in continuity mode.

The TPS659037 is replaced by Barracuda Holdings in Somerset-West. Results of physical and x-ray inspection can be seen in Figure 8.20. No physical short is found and upon measurement of output impedance no short on the output of the replaced PMIC is present.

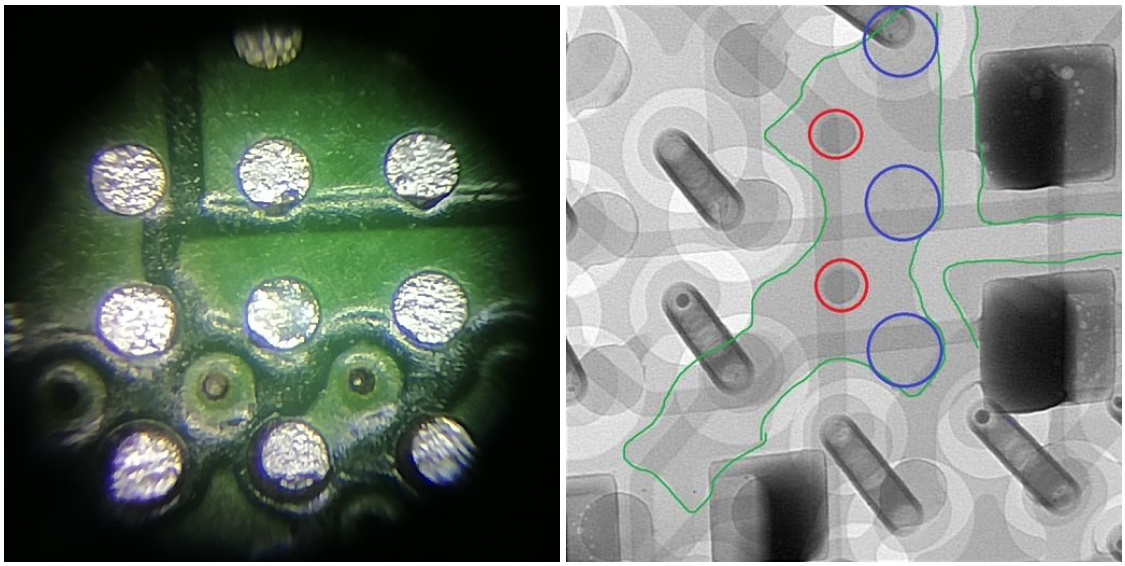


Figure 8.20: Photo and X-Ray confirms no manufacturing defect present

Upon 1st power-up of the replaced chip, the result is the same, and a low impedance on the output of the SMPS123 regulator is observed.

8.3.1.3 Cause

Upon close inspection of the datasheet, it is determined that the SMPS123 tri-phase regulator can operate either in tri-phase mode or in 1-dual and 1-single phase modes depending on the programmed OTP programming.

In tri-phase mode, each output is spaced 120^o apart, with the SMPS1 component placed at 0^o. In dual-and-single phase mode, dual phase SMPS12 fires at 0^o and 180^o, while SMPS3 also fires at 0^o. When both SMPS1 and SMPS3 fire at 0^o, an over-voltage condition is created that destroys the output MOSFET transistors of the switching regulator causing a short to ground.

This bug is not explicitly documented and detected when examining sequencing orders for different OTPs that are contained in an application guide.[41]

8.3.1.4 Solution

Correct sequencing is achieved when removing the SMPS3 inductor. Although this modification does decrease the current delivery from $9A$ at $0.9V$ to $6A$ at $0.9V$, an adequate over design margin ensures that adequate current can still be delivered for most use cases. A current budget can be found in Table 5.1.

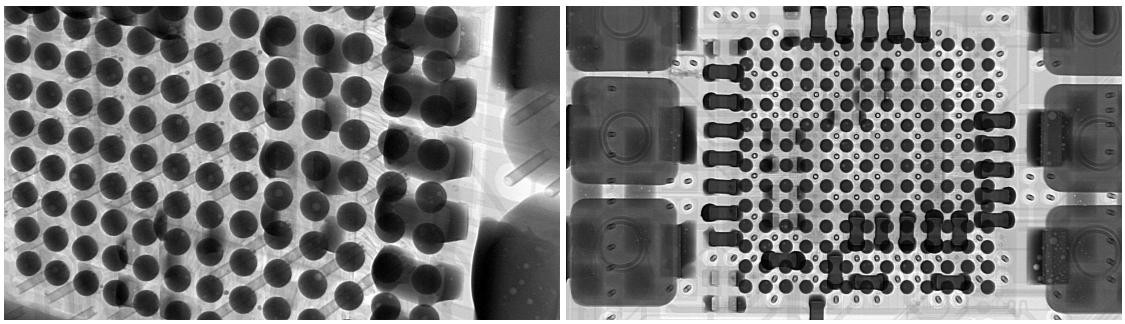


Figure 8.21: PMIC Replaced by Barracuda Holdings

8.3.2 External Regulator Module

8.3.2.1 Cause

Upon closer investigation of the circumstances that caused the previous bug, it is discovered that the TPS659037 does not offer "Configurable Power-Up and Power-Down Sequences (One-Time Programmable [OTP])" as claimed on the 1st page of the datasheet. The TPS659037 is indeed One Time Programmable as stated. However, it is programmed at the factory, and unprogrammed components are not available in small quantities (less than 10000) from Texas Instruments [11].

All outputs can still be configured to deliver the correct voltages from an external source, but it is also discovered that although the TPS659037 contains many enable pins, none can be configured to disable sources before sequencing [13].

8.3.2.2 Solution

The TPS659037 is treated as a programmable power source with no memory persistence. A fix is implemented in the form of 2 external PCBs. The 1st provides a $3.3V$ pull-up voltage for the I2C interface within $6ms$ of the $5V$ input to the TPS659037, and the 2nd gates the outputs to allow incorrect sequencing to complete before correct values are written externally.

The TPS659037 datasheet indicates that the V_IO $3.3V$ supply must be turned on at least $6ms$ after the VSYS_5V Supply [39, page 68]. This is achieved by using a Low Drop Out Linear regulator that regulates $5V$ down to $3.3V$. This regulator supplies an RC time constant that is buffered for accuracy and then level compared to produce a $3.3V$ signal that rises immediately after $6ms$. This signal is then applied to the enable line of a load switch that switches on $3.3V$ power from the linear regulator to the output. The result is a source that switches the VIO_IN supply on after a $7ms$ delay. Figure 8.22 shows a simulated RC time constant that is level compared.

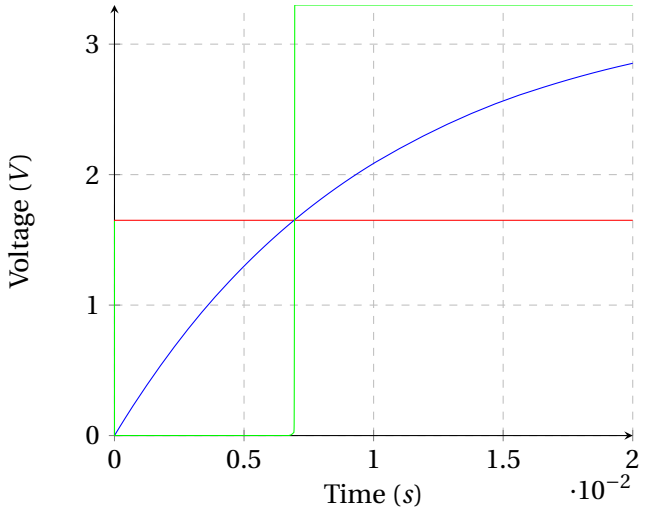


Figure 8.22: RC Time Constant Delay

This design is implemented on a small daughter board PCB that is added directly to the power system debug port on the Bokkiebord as shown in Figure 8.23.

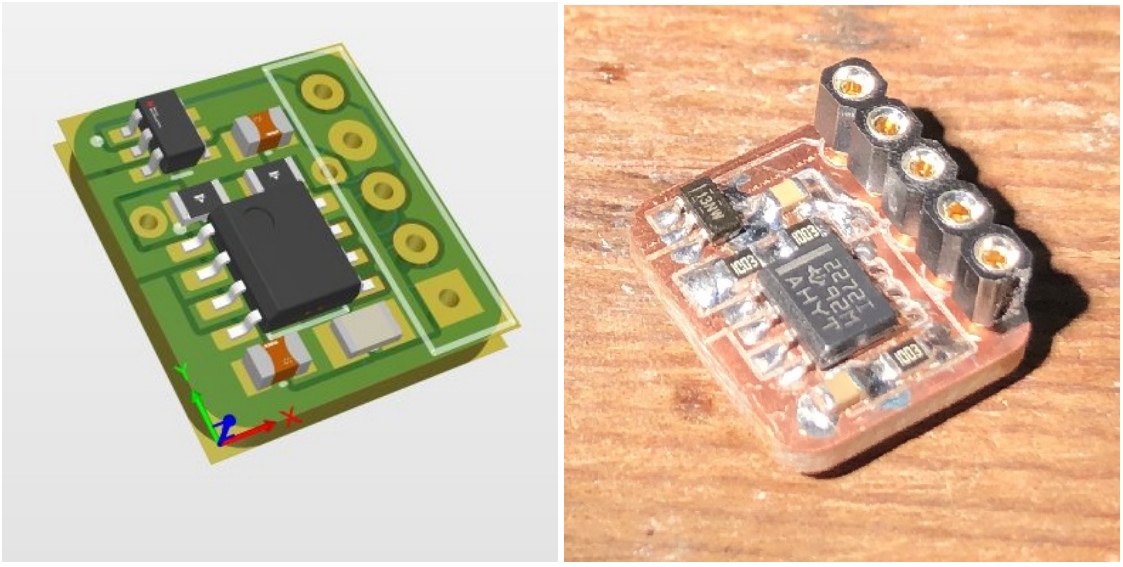


Figure 8.23: External regulator module

8.3.3 Inactive Disable Line

8.3.3.1 Symptoms

The TPS659037 cannot be reprogrammed and always sequences incorrectly upon 1st power-up while configuration cannot be altered before sequencing is complete. Since sources cannot easily be disabled by pulling an enable line low as discussed in Section 5.1.2.2, a risk exists that the power system may exceed the absolute maximum ratings of the processor and cause irreversible damage.

8.3.3.2 Solution

The power system is initially designed with 0805 0Ω bridge resistors to disconnect the power system from the processor during the programming phase. The footprints of these components are now used to implement TPS22914BYFPR load switches that gates the incorrect power sequence and allows configuration to occur before connecting to the processor.

The load switches are placed on a simple thin mod-board with castellated holes. This mod-board breaks into 2 parts that are placed on the top and bottom of the Bokkiebord. This mod-board is $0.4mm$ thick, and the load switches are of BGA footprint type. Each load switch has 4 pins and is soldered to the board with hot air and a generous amount of soldering flux. The mod-board is manufactured by PCBWay and can be seen in Figure 8.24.

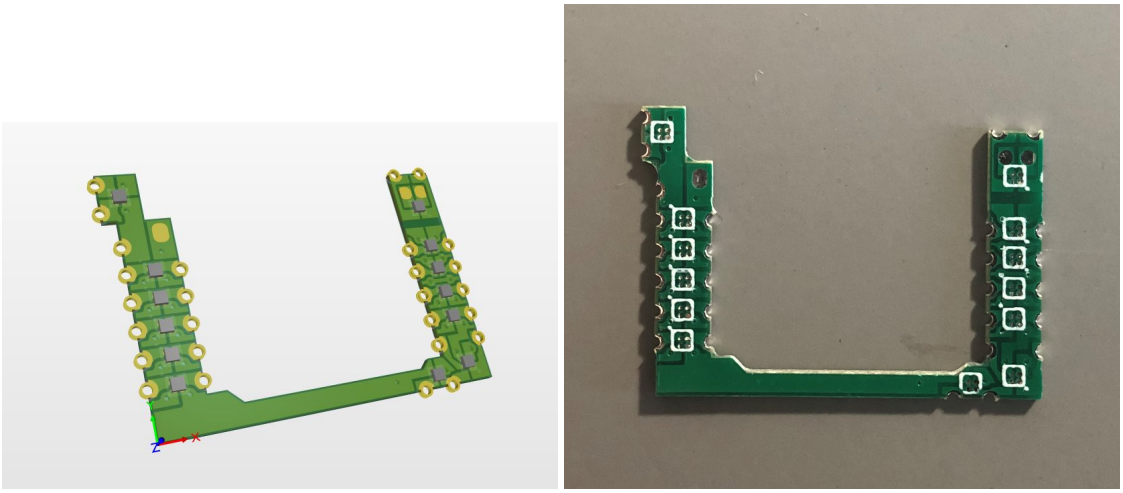


Figure 8.24: Load Switch Mod Board

The enable lines of the load switches are connected to the GPIO_4 pin on the TPS659037 that is normally held low during the standard power-sequence. After configuration is complete, the enable pin is set high in software, and the correct sequencing occurs.

This decision is made intentionally as a safety consideration. If the TPS659037 is reset for any reason, for example, if an overcurrent condition occurs and the bench power supply drops input voltage to control current, the PMIC resets, and incorrect sequencing restarts. The TPS659037 then sets the GPIO pin back to its default state, and the incorrect restarting power-up sequence is gated from the processor.

The total cost of implementing this fix is $2.80 for the mod board and 13 x *R*2.62 for the load switches. ($2.8 x 14.4 + 13 x *R*2.62) = *R*74.38. Figure 8.25 shows the mod-board mounted on the Bokkiebord.

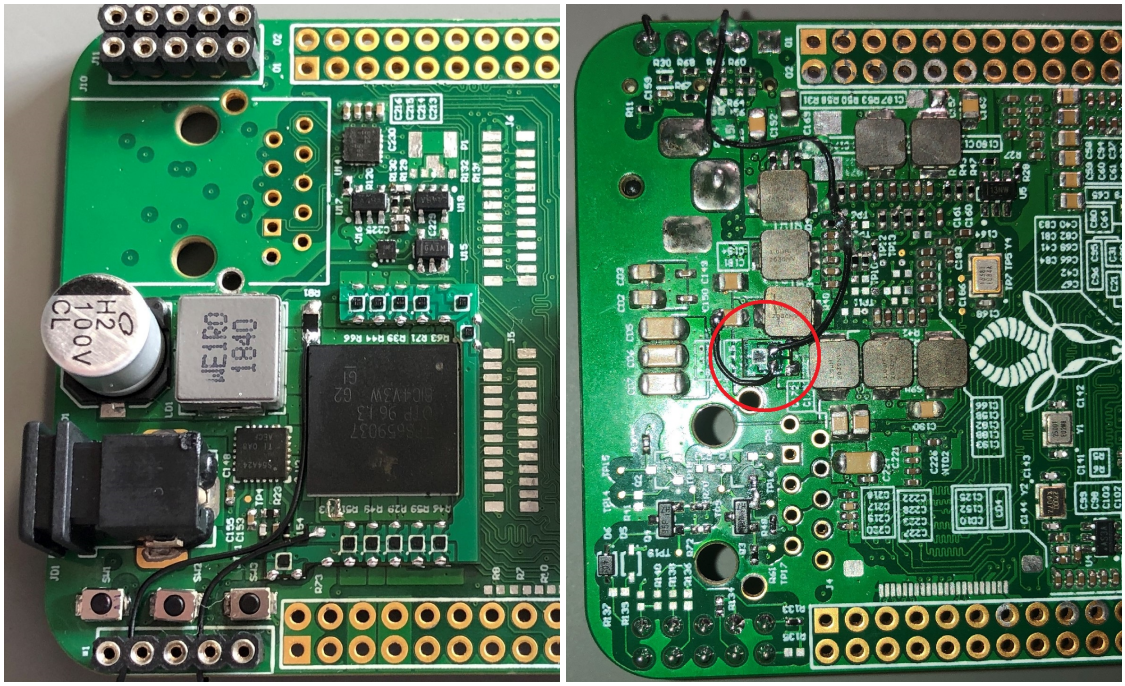


Figure 8.25: Load Switch Mod Board Mounted on The Bokkiebord

8.3.4 Real Time Clock

8.3.4.1 Symptoms

The sequencing order and voltages are verified at the decoupling capacitors underneath the processor and reset lines are both high, but the processor does not start.

The JTAG debugger is used to write data to the (Test Data Input) TDI pin, but the Bokkiebord does not respond with the (Test Data Output) TDO pin hovering about $2V$ on a $3.3V$ interface. Power consumption is $0.21A$ at $5V$.

The Processor does heat up but does not draw excessive current. This indicates a configuration problem rather than a hardware problem.

8.3.4.2 Diagnosis

When a digital signal is stuck in an uncertainty region it is an indication that internal logic is alive, but that it is not switching. The Bokkiebord has 2 external crystals that are used in a phase-locked-loop configuration to generate the required frequencies on the board. These crystals are 1 x $25MHz$ crystal and 1 x $27MHz$ crystal. Clocks are measured and found to be running.

In addition to the 2 external crystals, the Bokkiebord also has 1 x $32kHz$ RTC input. This signal is designed to be provided by the TPS659037, but due to the OTP programming this signal is unavailable [12]. An external RTC input can be supplied but is generally not required as i.MX8MQ is supposed to have an internal ring oscillator.

8.3.4.3 Cause

It is discovered from the power-up sequencing plan that the RTC clock reset is lifted before the voltage domain that powers the ring oscillator is powered.

Upon comparison of the 2018 and 2021 versions of the i.MX8MQ datasheet it is discovered that all reference to the internal ring oscillator had been removed in the latter, seeming to indicate that the internal ring oscillator functionality has either been removed from the i.MX8MQ or that it is defective in the chip design.

This RTC is used as an internal delay counter to produce a $4ms$ power on reset (POR) delay [57, page 979]. If no internally generated $32kHz$ clock is present this, POR delay time is infinite.

8.3.4.4 Solution

This problem is corrected by adding an AKER S533025-32.768K-X-15 $32.768kHz$ HCMOS Oscillator to the bottom side of the Bokkiebord. Power for the oscillator is obtained from the VDD_SNVS output of the mod-board.

Chapter 9

Software

Software is written to control the Bokkiebord on both hardware and operating system level. 2 Of the most important packages are the power system software and the Linux operating system.

9.1 Power System Software

Caution is recommended when altering the power system software as an error might cause the power system output voltages to exceed the absolute maximum allowable input ratings of the i.MX8MQ processor.

The TPS659037 PMIC [39] is advertised to be one time programmable (OTP) and to be able to supply a range of sequencing and voltage options.

It is initially planned to power up the component while its outputs are disconnected from the rest of the PCB and reprogram the OTP registers to a custom voltage and sequencing plan. The sequencing order and voltages would then be verified, and the bridge resistors would then be closed. The processor would then be powered up.

This is however not possible as the OTP of the TPS659037 is programmed at the factory and cannot be easily reprogrammed to supply the required Bokkiebord sequencing and voltage options. This problem is only discovered during the debugging stage and is discussed in Section 8.3.2. Un-programmed TPS659037 PMICs are not available, and custom OTP can only be obtained when large component orders are made directly to TI.

As this component is already integrated into the hardware, modifications are made to the current prototype rather than manufacturing new PCBs. This is done by allowing the PMIC to safely sequence incorrectly at first until the I2C registers are available. The PMIC is then reconfigured externally to provide the correct voltage and sequencing.

An enable pin is usually held low to allow a PMIC to reset successfully without applying power to the rest of the board. This is required as the I2C registers used to reconfigure the PMIC only become available after power-up sequencing has been completed.

The enable pin is unfortunately not functional as it has been disabled in the PMIC OTP memory. Enable pin functionality can be restored by one of the following methods:

9.1.1 Option 1: Reverse-Engineering the OTP

The OTP values are read from the OTP memory at reset and written to the user-accessible registers thereby switching the power resources to the correct voltage and in the correct sequencing order.

By applying a programming voltage to the VPROG pin, OTP EPROM registers can be accessed, and data can be read via I2C. The evaluation software shows that memory locations with 0 values can be written to 1 value (only once).

The documentation for the function of the memory locations is not freely available outside of Texas Instruments (TI). Clear documentation is however available for the user-accessible registers. By reading and comparing the values in both the OTP memory and user-accessible registers on power-up, the function of OTP memory locations can be determined.

By permanently rewriting the value of the POWER HOLD pin function register, the TPS659037 can be allowed to complete the power sequence without switching on power resources. Power resources can then be switched on one-by-one at the correct voltage by I2C.

This option is abandoned as it is considered too impractical to modify the OTP of a large number of boards and that the risk of incorrectly reprogramming OTP would result in the chip needing to be replaced and thereby costing time.

9.1.2 Option 2: Mod-Board

This option involves allowing the PMIC to sequence incorrectly while gating the power resource outputs with load switches. The load switches are connected to the GPIO_4 output pin on the PMIC, with a default output value of 0 in the factory OTP. When the PMIC has started, all power resources are switched off via I2C. The load switches are then enabled by writing a 1 value to the GPIO_4 output pin, thereby connecting the switched off power resources to the rest of the board. The power resources are then switched on one-by-one in the correct sequence and with the correct voltages via I2C.

The mod-board is a small and thin PCB that fits around the PMIC and connects to the pads initially reserved for bridge resistors via castellated holes. The mod board contains load switches that are connected to a communal enable line.

This option is selected as it adds relatively little cost and is easy to integrate into the final design.

9.2 Software Power-up of the PMIC

Following the above procedure, power-up is performed by writing the values contained in Table 9.1 to the registers of the PMIC via the external power system debug port. Power-down sequence is given in Table 9.2.

Power up sequence voltages are obtained from the i.MX8MQ datasheet [56, page 13] and sequencing delay values are selected from the PF4210 datasheet [50, page 17].

Sequence	Source name	Load Name	Voltage	Device Address	Register Address	Value	Reference [40]	Comment
0	LDO9	RTCRESETHIGH	3.3	0x58	0x61	0b00110001	p83 and p75	-
1	LDO3	NVCC_SNVS	3.3	0x58	0x55	0b00110001	p79 and p75	-
2	LDO4	VDD_SNVS	0.9	0x58	0x57	0b00000001	p81 and p75	-
3	LDO9	RTCRESETLOW	0.0	0x58	0x61	0b00000000	p83 and p75	
4	SMPS45	VDD_SOC	0.95	0x58	0x2B	0b00110011	p54 and p48 range is 0	-
5	SMPS12	VDD_0V9	0.9	0x58	0x23	0b00101110	p47 and p48	-
6	LDO1	VDDA_1P8	1.8	0x58	0x51	0b00010011	p75 and p75	-
7	SMPS6	NVCC_DRAM	1.1	0x58	0x2F	0b01000010	p57 and p48	-
7	SMPS7	NVCC_XXX	3.3	0x58	0x33	0b11111001	p59 and p48	-
7	-	-	-	0x58	0x30	0b00000001	p58	Enable
8	SMPS8	VDD_PHY_0V9	0.9	0x58	0x37	0b00101110	p62 and p48	-
8	LDO2	VDD_PHY_1V8	1.8	0x58	0x53	0b00010011	p77 and p75	-
8	LDOUSB	VDD_ANALOGUE_3V3	3.3	0x58	0x65	0b00110001	p87 and p75	
8	SMPS9	VDD_PHY_3V3	3.3	0x58	0x3B	0b11111001	p64 and p48	-
8	-	-	-	0x58	0x38	0b00000001	p63	Enable
9	LDOLN	PORRESET	3.3	0x58	0x63	0b00110001	p85 and p75	-

Table 9.1: PMIC Power Up Sequence

Sequence	Source name	Load Name	Voltage	Device Address	Register Address	Value	Reference [40]	Comment
0	LDOLN	PORRESET	0.0	0x58	0x63	0b00000000	p85 and p48	
1	SMPS9	VDD_PHY_3V3	0.0	0x58	0x3B	0b00000000	p64 and p48	
1	LDOUSB	VDD_ANALOGUE_3V3	0.0	0x58	0x65	0b00000000	p87 and p48	
1	LDO2	VDD_PHY_1V8	0.0	0x58	0x53	0b00000000	p77 and p48	
1	SMPS8	VDD_PHY_0V9	0.0	0x58	0x37	0b00000000	p62 and p48	
1	SMPS7	NVCC_XXX	0.0	0x58	0x33	0b00000000	p59 and p48	
1	SMPS6	NVCC_DRAM	0.0	0x58	0x2F	0b00000000	p57 and p48	
1	LDO1	VDDA_1V8	0.0	0x58	0x51	0b00000000	p75 and p75	
1	SMPS12	VDD_0V9	0.0	0x58	0x23	0b00000000	p47 and p48	
2	SMPS45	VDD_SOC	0.0	0x58	0x2B	0b00000000	p54 and p48 range is 0	
3	LDO9	RTCRESET	0.0	0x58	0x61	0b00000000	p83 and p75	
4	LDO4	VDD_SNVS	0.0	0x58	0x57	0b00000000	p81 and p75	
4	LDO3	NVCC_SNVS	0.0	0x58	0x55	0b00000000	p79 and p75	

Table 9.2: PMIC Power Down Sequence

9.3 Bokkiebord Hardware

9.3.1 Specific differences

The Bokkiebord hardware configuration is similar to the hardware configuration of the i.MX8MQ reference board but differs in the following key aspects:

- The Bokkiebord has $2GB$ LPDDR4 memory running at $1580MHz$ as opposed to the reference design's $3GB$ running at $1600MHz$.
- The Bokkiebord uses SD1 in 4-bit $50MHz$ SD card mode as opposed to 8-bit $400MHz$ eMMC mode.
- The Bokkiebord uses a TI TPS659037 PMIC of which the software is not compatible with the standard PF4210 PMIC from NXP.
- The Bokkiebord uses a different Ethernet PHY that communicates via the RMII interface as opposed to the RGMII interface and has reset and power-down pins connected to other GPIO pins than the reference design. The Ethernet PHY is reset before operation. The Bokkiebord provides a $50MHz$ clock to the Ethernet PHY.
- The Bokkiebord has a 12-bit $3MS/s$ SAR ADC running at up to $52MHz$ located on ECSPI2.

All hardware components are compatible with standard Linux device drivers but must be configured correctly. The following Section explains how a Bokkiebord Linux image is built by modifying the standard i.MX Linux image.

9.3.1.1 Memory Map

The memory subsystem consists of a $16Gb$ MT53E512M32D2NP-046 WT:E LPDDR4 module running at $1580MHz$. The memory subsystem consists of a Micron Technology LPDDR4 module operating at $1.8V$ VDD1, $1.10V$ VDD2 and $1.10V$ VDDQ voltage. The module supports 2 x 16-bit data channels with a bit density of $8Gb$ per channel. It consists internally of 2 dies that are contained in a 200 ball WFBGA package. The memory has a cycle time of $468ps$ [83, page 2].

The memory internally consists of 8 banks that are addressable by 3 bank select lines. The upper and lower part of the bank is selectable by 1 chip select line. The bit density per chip select is $1Gb$. The memory has 16 row address lines and 10 column addresses [83, page 20, 30]. In total 17,179,869,184 bits are available to be addressed in 8-bit chunks. These 17,179,869,184 bits therefore represent 2,147,483,648 8-bit memory locations. In hexadecimal, the memory is 0x80000000 bytes big.

The i.MX8MQ processor memory map [57, page 19] indicates that external memory starts at 0x40000000 an extends to 0x100000000 for a total of 3,221,225,472 possible bytes of memory. As the total installed external memory size is 2,147,483,648 (0x80000000) bytes, the Bokkiebord memory extends from 0x40000000 to 0xC0000000. A memory map of DMA mapped peripheral interfaces is also available [57, page 24].

The porting guide indicates that the bootloader is loaded into the memory from location 0x80020000 to 0x8FFFFFFF while the operating system is loaded into the memory from location 0x90000000 to 0x8C0000000 [52, page 39]. External memory allocation is shown in Table 9.3 [52, page 39]. Because the high bound of the operating system memory allocation might exceed the highest

physically available memory address, memory size has to be set correctly to avoid boot problems.

Start	End	Partition	Reservation
0x80000000	0x8001FFFF	Secure ATF	
0x80020000	0x801FFFFF	Non-secure OS	Reserved by UBoot
0x80200000	0x87FFFFFF	Non-secure OS	-
0x88000000	0x887FFFFF	M4_0	Reserved by SCFW and U-Boot for Cortex-M4
0x88800000	0x8FFFFFFF	M4_1	Reserved by SCFW and U-Boot for Cortex-M4
0x90000000	0xFDFFFFFF	Non-secure OS	-
0xFE000000	0xFFBFFFFF	Secure ATF	Reserved by ATF for OPTEE
0xFFC00000	0xFFFFFFFF	Non-secure OS	-
0x880000000	0x8C0000000	Non-secure OS	-

Table 9.3: External Memory Allocation

9.4 Introduction to the Build Environment

The Bokkiebord runs an adaptation of standard i.MX Linux [54], but can also be built from scratch [79]. The Linux operating system could change in the future and specific operating system changes are thus documented here.

9.4.1 The Yocto Project

The image is built with the use of the Yocto project [76]. The Yocto project is a set of compile- and package management tools that builds custom Linux distrobution images for specific applications by adding customisation layers. As the Yocto build environment can be daunting, Texas Instruments provides an introduction to the Yocto project [29].

9.4.2 U-boot

U-Boot [7] is the bootloader that loads the operating system from the SD card to the memory location where is executed. U-boot is responsible for configuring memory and setting up low level peripherals.

9.4.3 Linux Kernel

The Linux kernel is the central part of the Linux operating system. It is loaded at boot time into the memory by the U-boot bootloader. The bootloader hands over control to the Linux Kernel after it has been loaded into memory. The kernel then starts initialising device drivers and attempts to boot the device. The file system is mounted just before the login prompt appears. Higher-level applications are contained in the root file system partition of the SD card.

9.4.4 Device Tree

The device tree file describes to the Bootloader and Linux Kernel how the processor internals and peripheral hardware components are connected as well as hardware level setup of processor pins. The device tree specification [6] [72] describes the format in which device tree files are to be written. i.MX Linux contains different device trees for both the bootloader and the Linux Kernel. These device trees are not compatible and differ in subtle ways.

9.5 Installing the Build Environment

9.5.1 Build Host

The image is built on a x86 system running a standard freely available Linux operating system such as Ubuntu. A computer running a standard installation of Ubuntu Linux 20.04 Desktop with 2 Intel Xeon Processors, 32GB of RAM, 1TB of magnetic Storage and 240GB of solid-state storage is used. Depending on the size of the build and the speed of the source mirrors, build time can range between 5h-10h.

9.5.2 Install Dependencies

The following packages are required to build a Linux image for the Bokkiebord. These packages are installed with the apt-get command.[60, page 5]:

```
sudo apt-get install curl gawk wget git-core diffstat unzip texinfo
    GCC-multilib build-essential chrpath socat cpio python python3
   python3-pip python3-pexpect xz-utils debianutils iputils-ping
   python3-git python3-jinja2 libegl1-mesa libsdl1.2-dev pylint3
   xterm
```

Listing 9.1: Installing Build System Dependencies

9.5.3 Install the Repo Utility

The repo utility is set up. It is required to download the NXP Yocto tools. The Repo utility is a tool built on top of Git that makes repositories easier to manage.[60, page 5]:

```
mkdir ~/bin
curl https://storage.googleapis.com/git-repo-downloads/repo > ~/bin
   /repo
chmod a+x ~/bin/repo
export PATH=~/bin:$PATH
```

Listing 9.2: Obtaining the Repo Utility

9.5.4 Configure Git

The git tool is configured. This information is required when committing changes for creating patches in Yocto.

```
git config --global user.name "John Doe"
git config --global user.email "johndoe@sun.ac.za"
git config --list
```

Listing 9.3: Configuring the Git Utility

9.5.5 Create a Build Directory

A build directory with fast storage is created. This directory contains unpacked sources, configuration files, intermediate compilation results as well as final images. Storage capacity should be larger than $200GB$.

```
mkdir ~/build
```

Listing 9.4: Creating a Build Directory

9.5.6 Download the NXP Yocto tools

A Yocto directory is created within the build directory and the Yocto build system is downloaded with the repo command [60, page 6].

```
cd ~/build
mkdir imx-yocto-bsp
cd imx-yocto-bsp
repo init -u https://source.codeaurora.org/external/imx/imx-
   manifest -b imx-linux-zeus -m imx-5.4.24-2.1.0.xml
repo sync
```

Listing 9.5: Obtaining the iMX Yocto Build Tools

9.6 Configuring the Build Environment

The build environment for the i.MX8MQ is set up from within the Yocto directory.

9.6.1 Setup Build Environment

Change into the Yocto directory and setup the build environment for the i.MX8MQ processor.

```
cd ~/build/imx-yocto-bsp
DISTRO=fsl-imx-wayland MACHINE=imx8mqevk ACCEPT_FSL_EULA=1 source
   imx-setup-release.sh -b build
```

Listing 9.6: Configuring the i.MX Yocto Build Environment

9.6.2 (Optional) Optimise for processor cores

Modify Yocto build system parameters to optimise build performance. One thread per logical core is recommended when building the image but faster download of sources can be achieved by increasing the number of threads since source mirrors might have slow connections. Eight threads can be configured with:

```
echo "PARALLEL_MAKE = \"-j 8\"" >> ~/build/imx-yocto-bsp/build/conf
    /local.conf
echo "BB_NUMBER_THREADS = \"8\"" >> ~/build/imx-yocto-bsp/build/
    conf/local.conf
```

Listing 9.7: Configuring Build Threads

9.6.3 (Optional) Cache build sources

An optional local cache of sources can be created to speed up rebuilding time. It is recommended that the download cache location be placed outside the build directory. A sources directory is created in the user home directory adjacent to the build directory. Note that the sources directory could be very large and storage of about 100GB should be available.

```
echo "DL_DIR ?= \"~/sources\"" >> ~/build/imx-yocto-bsp/build/conf/
    local.conf
echo "BB_GENERATE_MIRROR_TARBALLS = \"1\"" >> ~/build/imx-yocto-bsp
    /build/conf/local.conf
```

Listing 9.8: Configuring Local Sources Cache

9.7 Configure the Build

Some peripherals of the reference board are not implemented in the Bokkiebord. These peripherals include wireless networking, cameras- and LCD interfaces. These device trees are removed from the build recipe by modifying the following file.

```
~/build/imx-yocto-BSP/sources/meta-Freescale/conf/machine/local.
    conf
```

Listing 9.9: iMX Yocto Build Recipe Location

All uboot dtb tags are removed from the "UBOOT_DTB_NAME" parameter except "fsl-imx8mq-evk.dtb".

```
# Set u-boot DTB
UBOOT_DTB_NAME = "fsl-imx8mq-evk.dtb"
```

Listing 9.10: Configuring Device Trees to Be Built

9.7.0.1 Disable Optee

The i.MX8MQ processor supports OP-TEE. OP-TEE is an open-source Trusted Execution Environment (TEE) that implements Arm TrustZone technology. It is a method of implementing hardware security through Root of Trust methods. It however prevents changing memory size in the Bokkiebord and must be disabled by editing the following file.

```
~/build/imx-yocto-BSP/sources/meta-imx/meta-BSP/conf/layer.conf
```

Listing 9.11: iMX Yocto Board Support Package Layer Location

Find the tag for the imx8mqevk and ensure that "optee" is added to the list of machine features to remove.

```
# Overrides for imx8mqevk.conf
.
.
.
MACHINE_FEATURES_remove_imx8mqevk = "optee ..."
```

Listing 9.12: Disabling Optee

9.8 Modifying source

Yocto uses version control software to manage the packages that need to be rebuilt. It is not guaranteed that Yocto will automatically detect and rebuild packages if changes are made directly in the build directory. A git version control hash needs to be updated before changes will be rebuilt. Devtool is used for this purpose. It automatically creates a software patch to facilitate changes. The following command is run from within the Yocto build directory while an environment is set up.

9.8.0.1 Invoke Devtool

Devtool is invoked by the use of the "devtool modify" command.

```
devtool modify The_Name_Of_Your_Package
```

Listing 9.13: Devtool Example Work Directory Creation

A directory is created where source code is modified.

```
~/build/imx-yocto-bsp/build/workspace/sources/
   The_Name_Of_Your_Package
```

Listing 9.14: Devtool Example Working Directory

9.8.0.2 Commit Changes

Changes are registered by commiting to git.

```
git add .
git commit -m "Your Commit Message"
```

Listing 9.15: Devtool Example Git Commit

9.8.0.3 Patch Source

A patch is created automatically to the build by updating a recipe with devtool.

```
devtool update-recipe The_Name_Of_Your_Package
```

Listing 9.16: Devtool Example Patch Creation

9.9 Preparation for patching i.MX Linux Source

Modification of standard i.MX Linux from NXP is shown in Section 9.10 - Section 9.14. These sections are divided into U-boot and Linux Kernel subsections.

Changes in the U-boot subsection is made to the "u-boot-imx" source package and changes in this subsection is seldom exactly compatible with changes made in Linux kernel packages.

Changes in the Linux kernel subsection is made in 3 different packages of which the changes are exactly identical in all 3 packages. Changes indicated in the Linux kernel subsection of the following sections are made in the "linux-imx", "linux-imx-headers" and "linux-libc-headers" packages.

In practice, changes are made in the "linux-imx" package as indicated in Section 9.10 - Section 9.14 and then copied to the other packages when changes have been finalised. This process is demonstrated in Section 9.16.

Packages are prepared for patching by invoking the devtool command.

```
devtool modify u-boot-imx
devtool modify linux-libc-headers
devtool modify linux-imx-headers
devtool modify linux-imx
```

Listing 9.17: Devtool Work Directory Creation

9.10 Unused Functionality

Some of the functionality implemented in the reference design is not available on the Bokkiebord. Two such functions are the QSPI non-volatile storage and the USB-C connector. These functions are disabled in the hardware initialisation files as it can cause the bootloader to hang.

9.10.1 U-boot

The unused function calls are commented out in the imx8mq_evk.c file. The location of the imx8mq_evk.c file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/board/
   freescale/imx8mq_evk/imx8mq_evk.c
```

Listing 9.18: iMX8MQ-EVK Low Level Initialisation File Location

Unused low level setup functions are commented out within the int board_init(void) function:

```
int board_init(void){

#ifdef CONFIG_FSL_QSPI
//board_qspi_init();
#endif
.
.
.
#ifdef CONFIG_USB_TCPC
//setup_typec();
#endif
.
.
.
}
```

Listing 9.19: Disabling Unused Functionality

9.11 Core voltage control

The processor automatically attempts to set core voltage and speed to a higher value upon boot. The bootloader first adjusts the core voltage by making a request to the PMIC to change core voltage from $0.9V$ to $1.0V$. The Linux Kernel attempts to change clock speed to a higher value. If the core voltage is not as expected spurious memory-like errors will occur which might stop the Bokkiebord from booting.

As the PF4210 has been replaced with the TPS659037, this request is unanswered. The Linux Kernel is prohibited from changing the core speed.

This is done by first disabling the function call that makes the request to the PMIC and then setting the core speed of all processor speed grades to a lower value. This problem can also be solved by adjusting the VDD_ARM rail voltage on the TPS659037.

9.11.1 U-Boot

The function call is commented out as indicated in the early initialisation spl.c file. The location of the spl.c file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/board/
   freescale/imx8mq_evk/spl.c
```

Listing 9.20: SPL Low Level Initialisation File Location

The power_init_board() function is commented out in the void board_init_f(ulong dummy) function to disable core voltage change request.

```
void board_init_f(ulong dummy){
.
.
.
/* Adjust pmic voltage to 1.0V for 800M */
//setup_i2c(0, CONFIG_SYS_I2C_SPEED, 0x7f, &i2c_pad_info1);

//power_init_board();
.
.
.
}
```

Listing 9.21: Disabling Unused Power System Control Instruction

9.11.2 Linux Kernel

Core speed is modified in the Linux Kernel device tree file. The location of the device tree file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/arm64/
   boot/dts/freescale/imx8mq.dtsi
```

Listing 9.22: iMX Linux Kernel Internal Device Tree File Location

The opp-supported-hw parameters are codes physically set on the processor. This code indicates to the software the speed grade of the processor. The Bokkiebord's i.MX8MQ processor is of speed grade 0x8 that can maintain a $800MHz$ core speed at $0.9V$ and a $1500MHz$ core speed at $1.0V$. These parameters are set for all devices by setting the "op-hz" parameter to $800MHz$ and the op-microvolt parameter to $0.9V$ as indicated below.

```
a53_opp_table: opp-table {
compatible = "operating-points-v2";
opp-shared;

opp-800000000 {
opp-hz = /bits/ 64 <800000000>;
opp-microvolt = <900000>;
/* Industrial only */
opp-supported-hw = <0xf>, <0x4>;
clock-latency-ns = <150000>;
opp-suspend;
};

opp-1000000000 {
opp-hz = /bits/ 64 <800000000>;
opp-microvolt = <900000>;
/* Consumer only */
opp-supported-hw = <0xe>, <0x3>;
clock-latency-ns = <150000>;
opp-suspend;
};
```

```
opp-1300000000 {
opp-hz = /bits/ 64 <800000000>;
opp-microvolt = <900000>;
opp-supported-hw = <0xc>, <0x4>;
clock-latency-ns = <150000>;
opp-suspend;
};

opp-1500000000 {
opp-hz = /bits/ 64 <800000000>;
opp-microvolt = <900000>;
opp-supported-hw = <0x8>, <0x3>;
clock-latency-ns = <150000>;
opp-suspend;
};
};
```

Listing 9.23: Reducing Core Frequencies For all Speed Grades

9.12 Memory

The memory configuration of the Bokkiebord differs to that of the reference design in 2 ways. The memory requirement for the Bokkiebord is 2*GB* instead of 3*GB* as in the case of the reference design. The physical transmission line characteristics of the Bokkiebord is unique. Lines are characterised in the debugging phase and the result is built into the operating system.

9.12.1 U-boot

9.12.1.1 Update memory timings

The lpddr4_timing.c file is replaced with the file obtained in Section 8.2.11. The location of the lpddr4_timing.c file is:

```
cp ~/archive/ddr_timing/lpddr4_timing.c \
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/board/
   freescale/imx8mq_evk
```

Listing 9.24: Replacing Standard LPDDR4_timing.c File For Bokkiebord Topology

9.12.1.2 Changing Memory Size in the Board Config File

Memory size is changed from 3*GB* or (0xC0000000 bytes) to 2*GB* or (0x80000000 bytes) in the imx8mq_evk.h board config file. The location of the imx8mq_evk.h file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/include/
   configs/imx8mq_evk.h
```

Listing 9.25: iMX8MQ-EVK Board Configuration File Location

The PHYS_SDRAM_SIZE parameter is changed to 0x80000000 to indicate that 2GB of RAM is available. Memory size is defined as a hexadecimal number indicating the memory size in bytes. OP-TEE has to be disabled as demonstrated earlier to reflect the memory size change at

boot time.

```
#define PHYS_SDRAM_SIZE 0x80000000 /* 2GB DDR */
```

Listing 9.26: Changing Memory Size in Board Configuration File

9.12.1.3 Change Memory Size in the U-Boot Device Tree File

Memory size is changed from 3*GB* or (0xC0000000) to 2*GB* or (0x80000000) in the memory device binding of the u-boot device tree file. The location of the u-boot device tree file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/arch/arm/
   dts/imx8mq-evk.dts
```

Listing 9.27: U-Boot Device Tree File Location

"memory@40000000" indicates that a memory device exists at processor hardware address of 0x40000000. It is segmented into 2 parts. The 1st part starts at device offset address 0x00000000 (real hardware address 0x40000000) and has a size of 0x10000000 (256*MB*). The 2nd part, which overlaps the 1st, also starts at device offset address 0x00000000 (hardware address 0x40000000) and has a size of 0x80000000 (2*GB*). The 4th value in the reg variable is set to 0x80000000 (2*GB*) instead of 0xC0000000 (3*GB*). The memory device binding should look as follows.

```
/ {
.
.
.
memory@40000000 {
device_type = "memory";
reg = <0x00000000 0x10000000 0x00000000 0x80000000>;
};
};
```

Listing 9.28: Changing Memory Size in U-Boot Device Tree

9.12.2 Linux Kernel

9.12.2.1 Change Memory Size in The Linux Kernel Device Tree

U-boot and the Linux kernel rely on 2 different sets of device tree files which are not compatible. Memory size is changed from 3*GB* or (0xC0000000) to 2*GB* or (0x80000000) in the memory device binding of the Linux Kernel device tree file. The location of the Linux kernel device tree file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/arm64/
   boot/dts/freescale/imx8mq-evk.dts
```

Listing 9.29: i.MX Linux Kernel Device Tree File Location

"memory@40000000" indicates that a memory device exists at processor hardware address of 0x40000000. It is segmented into 2 parts. The 1st part starts at device offset address 0x00000000 (real hardware address 0x40000000) and has a size of 0x10000000 (256MB). The 2nd part, which overlaps the 1st, also starts at device offset address 0x00000000 (hardware address 0x40000000) and has a size of 0x80000000 (2GB). The 4th value in the reg variable is set to 0x80000000 (2GB) instead of 0xC0000000 (3GB). The memory device binding should look as follows.

```
/ {
.
.
.
memory@40000000 {
device_type = "memory";
reg = <0x00000000 0x10000000 0x00000000 0x80000000>;
};
};
```

Listing 9.30: Changing Memory Size in Linux Kernel Device Tree File

9.12.2.2 Change DMA Memory Allocation in Linux Kernel Internal Device Tree

direct memory access (DMA) reserves a chunk of memory for peripheral devices. This parameter is modified as DMA ranges might extend over a large part of the memory and prohibit Linux from being loaded into memory if too little space is left on small memory devices. Memory size reservation is set to 0x00 size in the Linux Kernel internal device tree. The location of the internal device tree file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/arm64/
   boot/dts/freescale/imx8mq-evk.dtsi
```

Listing 9.31: i.MX Linux Kernel Internal Device Tree File Location

DMA memory reservation is set as 0x0:

```
dma-ranges = <0x40000000 0x0 0x40000000 0x80000000>;
```

Listing 9.32: Changing DMA Allocation in Linux Kernel Internal Device Tree File

9.13 SD card

The main boot device of the Bokkiebord differs from that of the reference design in that it is a SD card running in 4-bit mode at 50MHz instead of a MMC device running in 8-bit mode at 400MHz. The SD card is connected to the SD1 interface on the i.MX8MQ. The SD2 interface is unused. Interface speed is changed in pin descriptions and device descriptions are changed in device tree bindings of the device tree. The boot device is set in the configuration files.

9.13.1 U-boot

Boot device is set in the imx8mq_evk.h board config file. The location of the imx8mq_evk file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/include/
   configs/imx8mq_evk.h
```

Listing 9.33: iMX8MQ-EVK Board Configuration File Location

9.13.1.1 Change Boot Device in Board Config File

SD1 is set as the primary boot device and the root partition is set as the 3rd partition on the 1st SD device.

```
#define CONFIG_SYS_MMC_ENV_DEV 0 /* USDHC1 */
#define CONFIG_MMCROOT "/dev/mmcblk0p2" /* USDHC1 */
```

Listing 9.34: Changing Boot Device and Boot Partition

The number of SD cards is set to 1 to avoid possible hangup problems while searching for a boot device.

```
#define CONFIG_SYS_FSL_USDHC_NUM  1
```

Listing 9.35: Change Number of Boot Devices

9.13.1.2 Change SD-Card Device Binding in U-Boot Device Tree

SD card interface speed and device description is usually changed by editing the U-Boot device tree file but is left unchanged because the "int board_mmc_init(bd_t *bis)" function in spl.c correctly detects the SD-card type and because U-Boot complains when it is changed.

The "&usdhc1" definitions are left unchanged in the U-Boot device tree file and the "&usdhc2" device binding and accompanying pin definitions are removed from the u-boot device tree file.

9.13.2 Linux Kernel

9.13.2.1 Change SD-Card Device Binding in Linux Kernel Device Tree

The "bus-width" parameter in the "&usdhc1 " device binding in the Linux kernel device tree is set to 4 to allow 4-bit mode. The location of the Linux kernel device tree file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/arm64/
   boot/dts/freescale/imx8mq.dts
```

Listing 9.36: iMX Linux Kernel Internal Device Tree File Location

```
&usdhc1 {
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_usdhc1>;
bus-width = <4>;
non-removable;
```

```
no-sdio;
status = "okay";
};
```

Listing 9.37: Changing SD Card Device Binding in Linux Kernel Device Tree File

9.13.2.2 Change SD-Card Pin Definitions in Linux Kernel Device Tree

All but standard speed definitions for "&usdhc1" are removed in the Linux kernel device tree to ensure 50MHz mode. All definitions for the unused "&usdhc2" interface are removed.

```
pinctrl_usdhc1: usdhc1grp {
fsl,pins = <
MX8MQ_IOMUXC_SD1_CLK_USDHC1_CLK 0x83
MX8MQ_IOMUXC_SD1_CMD_USDHC1_CMD 0xc3
MX8MQ_IOMUXC_SD1_DATA0_USDHC1_DATA0 0xc3
MX8MQ_IOMUXC_SD1_DATA1_USDHC1_DATA1 0xc3
MX8MQ_IOMUXC_SD1_DATA2_USDHC1_DATA2 0xc3
MX8MQ_IOMUXC_SD1_DATA3_USDHC1_DATA3 0xc3
MX8MQ_IOMUXC_SD1_DATA4_USDHC1_DATA4 0xc3
MX8MQ_IOMUXC_SD1_DATA5_USDHC1_DATA5 0xc3
MX8MQ_IOMUXC_SD1_DATA6_USDHC1_DATA6 0xc3
MX8MQ_IOMUXC_SD1_DATA7_USDHC1_DATA7 0xc3
MX8MQ_IOMUXC_SD1_STROBE_USDHC1_STROBE 0x83
MX8MQ_IOMUXC_SD1_RESET_B_USDHC1_RESET_B 0xc1
>;
};
```

Listing 9.38: SD Card Pin Definitions in Linux Kernel Device Tree File

9.14 Ethernet PHY

The Bokkiebord differs from the reference design in that it uses an Ethernet PHY that communicates over RMII as opposed to RGMII. The reset pin of the Ethernet PHY is connected to the 14th pin of the 1st GPIO bank on the processor and the powerdown pin of the Ethernet PHY is connected to the 15th pin of the 1st GPIO bank on the processor. Both the reset pin and the powerdown pin on the Ethernet PHY are active low and is set to a logical high for the Ethernet PHY to be placed into active state. The i.MX8MQ supplies a $50MHz$ reference clock to the Ethernet PHY.

The Ethernet PHY is administrated by MDIO. The MDIO Port of the Ethernet PHY has a $10k\Omega$ pull-up resistor to $3.3V$.

9.14.1 U-boot

9.14.1.1 Change Ethernet PHY device binding in U-Boot device tree file

The device binding for "&fec1" is changed in the U-boot device tree file to set the Ethernet interface type to RMII. The location of the U-boot device tree is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/arch/arm/
   dts/imx8mq-evk.dts
```

Listing 9.39: U-Boot Device Tree File Location

The device binding for the Ethernet PHY is changed to that of an RMII device. The reset line is integrated into the device binding, but the powerdown pin is not as powerdown pin functionality is not supported in the Ethernet device binding as standard.

```
&fec1 {
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_fec1>;
phy-mode = "rmii";
phy-handle = <&ethphy0>;
phy-reset-gpios = <&gpio1 14 GPIO_ACTIVE_LOW>;
phy-reset-duration = <100>;
fsl,magic-packet;
status = "okay";
mdio {
#address-cells = <1>;
#size-cells = <0>;
ethphy0: ethernet-phy@0 {
compatible = "dp83822";
reg = <0>;
};
};
};
```

Listing 9.40: Ethernet PHY Device Binding in U-Boot Device Tree File

9.14.1.2 Change Ethernet Pin Definitions in U-Boot device tree

The pin definitions for the Ethernet PHY is changed to that of an RMII device. Both reset and powerdown pin definitions are included pin the Ethernet PHY pin definition. The reset pin is controlled by the device driver and the powerdown pin is set by a low level initialisation function.

```
&iomuxc {
pinctrl-names = "default";
imx8mq-val {
pinctrl_fec1: fec1grp {
fsl,pins = <
MX8MQ_IOMUXC_ENET_MDC_ENET1_MDC             0x00000006
MX8MQ_IOMUXC_ENET_MDIO_ENET1_MDIO           0x00000006
MX8MQ_IOMUXC_ENET_TD2_ENET1_TX_CLK         0x4000001f
MX8MQ_IOMUXC_ENET_TD1_ENET1_RGMII_TD1       0x00000016
MX8MQ_IOMUXC_ENET_TD0_ENET1_RGMII_TD0       0x00000016
MX8MQ_IOMUXC_ENET_RD1_ENET1_RGMII_RD1       0x00000016
MX8MQ_IOMUXC_ENET_RD0_ENET1_RGMII_RD0       0x00000016
MX8MQ_IOMUXC_ENET_RXC_ENET1_RX_ER           0x00000016
MX8MQ_IOMUXC_ENET_RX_CTL_ENET1_RGMII_RX_CTL   0x00000016
MX8MQ_IOMUXC_ENET_TX_CTL_ENET1_RGMII_TX_CTL   0x00002016
MX8MQ_IOMUXC_GPIO1_IO14_GPIO1_IO14         0x19
```

```
MX8MQ_IOMUXC_GPIO1_IO15_GPIO1_IO15        0x19
>;
};
}
```

Listing 9.41: Ethernet PHY Pin Definitions in U-Boot Device Tree File

9.14.1.3 Add Ethernet Powerdown Pin Control in Early Board Initialisation Procedure

The ethernet powerdown pin is assigned to a variable and set to a logical high in the early initialisation spl.c file. The location of the spl.c file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/board/
   freescale/imx8mq_evk/spl.c
```

Listing 9.42: SPL Low Level Initialisation File Location

Code is added to the board_init_f() function. The #define statement assigns the 15th pin on the 1st GPIO bank of the processor to a variable called ETHER_PWRDN_GPIO. The state of the pin is set by the gpio_direction_output function. The printf function is used to output debugging information to the terminal.

```
#define ENET1_PWRDN_GPIO IMX_GPIO_NR(1, 15)
void board_init_f(ulong dummy)
{
.
.
.

init_uart_clk(0);

//Bokkiebord Set ENET1 PWRDN Pin 1 to bring out of sleep mode
//Verify that the pin is not reserved by writing a safe 0 value to
   the pin.
udelay(1500000);
printf("Bokkiebord: Setting up RMII PWRDN is 0\n");
gpio_direction_output(ENET1_PWRDN_GPIO, 0);
udelay(1500000);
//Raise the pin to a logical high voltage to bring the PHY out of
   sleep
printf("Bokkiebord: Setting up RMII PWRDN is 1\n");
gpio_direction_output(ENET1_PWRDN_GPIO, 1);
udelay(1500000);
board_early_init_f();
.
.
.

}
```

Listing 9.43: Adding Code to Set Ethernet PHY Powerdown Pin High Upon U-Boot Startup.

9.14.1.4 Configure Ethernet Reference Clock in Board Initialisation Procedure

The Ethernet PHY reference clock is provided by the processor. The clock is set to active state in the imx8mq-evk.c file. The location of the imx8mq-evk.c file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/board/
   freescale/imx8mq_evk/imx8mq-evk.c
```

Listing 9.44: iMX8MQ-EVK Board Configuration File Location

Code is added to the setup_fec(void) function to start the $50MHz$ reference clock signal to the Ethernet PHY.

```
static int setup_fec(void)
{
struct iomuxc_gpr_base_regs *gpr = (struct iomuxc_gpr_base_regs *)
   IOMUXC_GPR_BASE_ADDR;

printf("Bokkiebord: Setting up RMII Ethernet Clock\n");
setbits_le32(&gpr->gpr[1],
IOMUXC_GPR_GPR1_GPR_ENET1_TX_CLK_SEL_MASK);
return set_clk_enet(ENET_50MHZ);
}
```

Listing 9.45: Modifying Code to Start $50MHz$ Reference Clock For Ethernet PHY

9.14.1.5 (Optional) Configure Ethernet PHY Administration Commands

The registers of the Ethernet PHY can be set over the MDIO interface with the use of the "phy_write()" function.

```
int board_phy_config(struct phy_device *phydev)
{
//phy_write(phydev, MDIO_DEVAD_NONE, 0x1d, 0x1f);

if (phydev->drv->config){
  phydev->drv->config(phydev);
}
return 0;
}
```

Listing 9.46: Configuring PHY internal registers over MDIO.

9.14.1.6 Change Ethernet Device Binding in Linux Kernel Device Tree

The device binding for "&fec1" is changed in the Linux Kernel device tree file to set the Ethernet interface type to RMII. The reset pin is not included in the Linux Kernel Device tree file. The location of the Linux Kernel device tree is:

```
~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/arm64/
   boot/dts/freescale/imx8mq.dts
```

Listing 9.47: iMX Linux Kernel Device Tree File Location

The Linux Kernel Device Tree Binding does not include a reset pin as in the case of the U-boot device tree binding.

```
&fec1 {
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_fec1>;
phy-mode = "rmii";
phy-handle = <&ethphy0>;
fsl,magic-packet;
status = "okay";
mdio {
#address-cells = <1>;
#size-cells = <0>;
ethphy0: ethernet-phy@0 {
compatible = "dp83822";
reg = <0>;
};
};
};
```

Listing 9.48: Ethernet PHY Device Binding in Linux Kernel Device Tree File

The pin definitions for the Ethernet PHY is changed to that of an RMII device. Both reset and powerdown pin definitions are included pin the Ethernet PHY pin definition.

```
&iomuxc {
pinctrl-names = "default";
imx8mq-val {
pinctrl_fec1: fec1grp {
fsl,pins = <
MX8MQ_IOMUXC_ENET_MDC_ENET1_MDC              0x00000006
MX8MQ_IOMUXC_ENET_MDIO_ENET1_MDIO            0x00000006
MX8MQ_IOMUXC_ENET_TD2_ENET1_TX_CLK         0x4000001f
MX8MQ_IOMUXC_ENET_TD1_ENET1_RGMII_TD1        0x00000016
MX8MQ_IOMUXC_ENET_TD0_ENET1_RGMII_TD0        0x00000016
MX8MQ_IOMUXC_ENET_RD1_ENET1_RGMII_RD1        0x00000016
MX8MQ_IOMUXC_ENET_RD0_ENET1_RGMII_RD0        0x00000016
MX8MQ_IOMUXC_ENET_RXC_ENET1_RX_ER            0x00000016
MX8MQ_IOMUXC_ENET_RX_CTL_ENET1_RGMII_RX_CTL   0x00000016
MX8MQ_IOMUXC_ENET_TX_CTL_ENET1_RGMII_TX_CTL   0x00002016
MX8MQ_IOMUXC_GPIO1_IO14_GPIO1_IO14         0x19
MX8MQ_IOMUXC_GPIO1_IO15_GPIO1_IO15         0x19
>;
};
}
```

Listing 9.49: Ethernet PHY Pin Definitions in Linux Kernel Device Tree File

9.15 Analogue to Digital Converter

The ADC is connected to the 2nd SPI interface on the i.MX8MQ. The ECSPI2 interface is added to the device tree files as a generic SPI device running at $40MHz$. Interface speed can be increased to the processor maximum of $52MHz$. The ADC is capable of operating at frequencies as high as $60.02MHz$. The ECSPI2 interface has 1 chip select line capable of representing 2 devices. The "reg" parameter defines the device "address" and setting it to 0 will associate a low chip select signal with this ADC. The ADC is abstracted as a binary file in the Linux operating system. ADC measurements can be obtained by reading the contents of this file as shown in Section 8.2.18.

9.15.1 U-Boot

The device tree binding is added to the u-boot device tree file. The Location of the U-boot device tree file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/arch/arm/
   dts/imx8mq-evk.dts
```

Listing 9.50: U-Boot Device Tree File Location

A generic SPI device running at 40MHz is added to the u-boot device tree file.

```
&ecspi2 {
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_ecspi2>;
status = "okay";
spidev@0{
compatible = "spidev";
reg = <0>;
spi-max-frequency = <40000000>;
};
};
```

Listing 9.51: ADC Device Binding in U-Boot Device Tree File

Pin definitions for the generic SPI device is added to the U-boot device tree file.

```
pinctrl_ecspi2: ecspi2grp {
fsl,pins = <
MX8MQ_IOMUXC_ECSPI2_MISO_ECSPI2_MISO        0x00000016
MX8MQ_IOMUXC_ECSPI2_SCLK_ECSPI2_SCLK        0x00001816
MX8MQ_IOMUXC_ECSPI2_SS0_ECSPI2_SS0          0x00000016
>;
};
```

Listing 9.52: ADC Pin Definitions in U-Boot Device Tree File

9.15.2 Linux Kernel

The device tree binding is added to the Linux kernel device tree file. The location of the Linux kernel device tree file is:

```
~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/arm64/
   boot/dts/freescale/imx8mq.dts
```

Listing 9.53: iMX Linux Kernel Device Tree File Location

A generic SPI device running at $40MHz$ is added to the Linux kernel device tree file.

```
&ecspi2 {
pinctrl-names = "default";
pinctrl-0 = <&pinctrl_ecspi2>;
status = "okay";
spidev@0{
compatible = "linux,spidev";
reg = <0>;
spi-max-frequency = <40000000>;
};
};
```

Listing 9.54: ADC Device Binding in Linux Kernel Device Tree File

Pin definitions for the generic SPI device is added to the u-boot device tree file.

```
pinctrl_ecspi2: ecspi2grp {
fsl,pins = <
MX8MQ_IOMUXC_ECSPI2_MISO_ECSPI2_MISO        0x00000016
MX8MQ_IOMUXC_ECSPI2_SCLK_ECSPI2_SCLK        0x00001816
MX8MQ_IOMUXC_ECSPI2_SS0_ECSPI2_SS0          0x00000016
>;
};
```

Listing 9.55: ADC Pin Definitions in Linux Kernel Device Tree File

9.16 Patching i.MX Linux Source

9.16.1 U-boot

Changes made in the "u-boot" package are committed and a patch is created with devtool:

```
cd ~/build/imx-yocto-bsp/build/workspace/sources/u-boot-imx/
git add .
git commit -m "Bokkiebord Source"
devtool update-recipe u-boot-imx
```

Listing 9.56: Changes in U-Boot are Committed and a Patch is Created

9.16.2 Linux Kernel

Changes made in the "linux-imx" package are committed and a patch is created with devtool:

```
cd ~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/
git add .
git commit -m "Bokkiebord Source"
devtool update-recipe linux-imx
```

Listing 9.57: Changes in Linux-imx are Committed and a Patch is Created

Files changed in the "linux-imx" package are copied to "linux-imx-headers" and "linux-libc-headers".

```
cp ~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/
   arm64/boot/dts/freescale/imx8mq-evk.dts ~/build/imx-yocto-bsp/
   build/workspace/sources/linux-imx-headers/arch/arm64/boot/dts/
   freescale/

cp ~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/
   arm64/boot/dts/freescale/imx8mq-evk.dtsi ~/build/imx-yocto-bsp/
   build/workspace/sources/linux-imx-headers/arch/arm64/boot/dts/
   freescale/

cd ~/build/imx-yocto-bsp/build/workspace/sources/linux-imx-headers
git add .
git commit -m "Bokkiebord Source"
devtool update-recipe linux-imx-headers
```

Listing 9.58: Copying Linux Kernel Changes to "linux-imx-headers" Package, Committing and Patching

```
cp ~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/
   arm64/boot/dts/freescale/imx8mq-evk.dts ~/build/imx-yocto-bsp/
   build/workspace/sources/linux-libc-headers/arch/arm64/boot/dts/
   freescale/

cp ~/build/imx-yocto-bsp/build/workspace/sources/linux-imx/arch/
   arm64/boot/dts/freescale/imx8mq-evk.dtsi ~/build/imx-yocto-bsp/
   build/workspace/sources/linux-libc-headers/arch/arm64/boot/dts/
   freescale/

~/build/imx-yocto-bsp/build/workspace/sources/linux-libc-headers
git add .
git commit -m "Bokkiebord Source"
devtool update-recipe linux-libc-headers
```

Listing 9.59: Copying Linux Kernel Changes to "linux-libc-headers" Package, Committing and Patching

9.17 Building The Image

Multiple image configurations can be built depending on the current stage of the debug process. A minimal image is suitable when testing if a device boots, but a full image is recommended for production environments. Bitbake is invoked with the desired build configuration to start the build process.

```
#bitbake core-image-minimal
#bitbake core-image-base
```

```
#bitbake core-image-sato
#bitbake imx-image-core
bitbake fsl-image-machine-test
#bitbake imx-image-multimedia
#bitbake imx-image-full
```

Listing 9.60: Possible Image Build Configurations

Completed image files can be found can be found in directory:

```
~/build/imx-yocto-bsp/build/tmp/deploy/images/imx8mqevk/
```

Listing 9.61: Image Build Process Output Location

The relevant output files are depending on the image type name:

```
fsl-image-machine-test-imx8mqevk.wic.bz
Image.bin
imx8mq-evk.dtb
imx-boot-imx8mqevk-sd.bin-flash_evk
```

Listing 9.62: Relevant Output Files

The image file is decompressed before flashing onto the Bokkiebord. Decompress the *.wic.bz file with the following command:

```
bunzip2 -dk -f *.wic.bz2
```

Listing 9.63: Decompressing the .wic.bz2 file

9.18 Loading The Image on to Bokkiebord

The image can either be burned directly onto an SD-Card or be loaded via the Universal Update Utility from NXP.

9.18.1 Burning SD Card

A SD Card image can be burned with the use of the "dd" command. Find the name of the sd card with the following command. It will list all storage partitions on the host computer.

```
sudo lshw
```

Listing 9.64: Show Attached Hardware

Burn image to SD card:

```
sudo dd if=<image name>.wic of=/dev/<sd-card-device> bs=1M conv=
    fsync
```

Listing 9.65: Flashing SD Card Image

The boot resistors are configured correctly for boot from the SD card. This method will be helpful when running Bokkiebord in a production environment.

9.18.2 Universal Update Utility

NXP has 2 programming tools. The 1st is the older GUI based mfgtools, and the 2nd is the universal update utility (UUU) [61]. Mfgtools has primarily been superseded by UUU.

The UUU loads software via the USB-OTG port onto Bokkiebord. The status of the upload can be monitored via the UART-USB ports.

The UUU is an executable file that automatically detects the Bokkiebord as a HID Device. The UUU requires an input file. The input file is shown below. The input file is saved as uuu.auto. The following script is provided by NXP.

```
uuu_version 1.2.39

# This command will be run when i.MX6/7 i.MX8MM, i.MX8MQ
SDP: boot -f imx-boot-imx8mqevk-sd.bin-flash_evk_no_hdmi

# This command will be run when ROM support stream mode
# i.MX8QXP, i.MX8QM
SDPS: boot -f imx-boot-imx8mqevk-sd.bin-flash_evk_no_hdmi

# These commands will be run when using SPL and will be skipped if
    no spl
# SDPU will be deprecated. please use SDPV instead of SDPU
# {
SDPU: delay 1000
SDPU: write -f imx-boot-imx8mqevk-sd.bin-flash_evk_no_hdmi -offset
    0x57c00
SDPU: jump
# }

# These commands will be run when using SPL and will be skipped if
    no spl
# if (SPL support SDPV)
# {
SDPV: delay 1000
SDPV: write -f imx-boot-imx8mqevk-sd.bin-flash_evk_no_hdmi -skipspl
SDPV: jump
# }

FB: ucmd setenv fastboot_dev mmc
FB: ucmd setenv mmcdev ${emmc_dev}
FB: ucmd mmc dev ${emmc_dev}
#FB: flash -raw2sparse all fsl-image-machine-test-imx8mqevk.wic
#FB: flash bootloader imx-boot-imx8mqevk-sd.bin-flash_evk_no_hdmi
FB: ucmd if env exists emmc_ack; then ; else setenv emmc_ack 0; fi;
#FB: ucmd mmc partconf ${emmc_dev} ${emmc_ack} 0 0

# Boot Linux Kernel
FB: ucmd setenv fastboot_buffer ${loadaddr}
FB: download -f Image.bin
FB: ucmd setenv fastboot_buffer ${fdt_addr}
FB: download -f imx8mq-evk.dtb
```

```
FB: acmd booti ${loadaddr} - ${fdt_addr}
FB: done
```

The UUU is invoked with the following command:

```
uuu.exe uuu.auto
```

This method is useful when running Bokkiebord in a development environment. The host computer searches for the Bokkiebord as a HID device.

9.19 Booting the Bokkiebord

Upon successful loading of the operating system the Bokkiebord starts to boot automatically. Progress is monitored with terminal window connected to the UART-USB Bridge debugging interface. The bootloader runs first, and the Linux kernel then takes over. The Linux kernel first loads drivers, displays service statuses, and then presents the login prompt. The bootloader can be interrupted for debugging purposes by pressing Enter.

A few very useful commands [59] are "mmc info" for checking if the sd-card is visible to the Bokkiebord and "mmc part" for checking the integrity of the file system. The "gpio" command can be used to read and set pin logical voltage levels. The "mii status" command can be used to verify the operation of the Ethernet PHY. Environmental variables can be read using the "printenv" command and changed using the "setenv" command. System Boot may be continued using the "boot" command. A boot log is provided in appendix D comparison.

Chapter 10

Performance

SBC performance is evaluated by executing a program with a predictable workload on multiple platforms and comparing the execution times.

The programming language chosen for this purpose is the Python3 [78] scripting language as it is readily available on all machines of interest and because it highlights the advantages of the Linux operating system over that of conventional microcontrollers.

The Python3 scripting language offers a vast array of native and external libraries that can be used to solve problems ranging from signal processing to machine learning.

Even though many libraries are available, external libraries are not used for this implementation as platform specific optimisations could contaminate the result.

Python3 is much slower than the conventionally utilised Assembler and C, but the same language inefficiencies are present on all platforms investigated.

The algorithm chosen to benchmark the Bokkiebord is the discrete time Fourier transform (DTFT) [75]. The DTFT is a discrete method for decomposing digitally sampled signals into its component frequencies. The result is a continuous expression that can again be digitally represented by plotting discrete values of various lengths.

The advantage of this approach over that of the discrete Fourier transform (DFT) is that the length of the inner and outer loops can be changed to suit the capabilities of the set of hardware under test.

A simulated signal with 1000 points is first generated, a 10000 point DTFT is performed and and the largest component frequency is identified. The test is repeated 10 times and an average of the program execution times is presented here.

Note that the DTFT is less efficient than the closely related FFT, but preferred as the implementation is slightly simpler and because the low level processor operations align better with the benchmarking goal. The purpose of this test is to place the CPU under stress, but if the purpose of the program is practical signal processing the FFT is recommended.

The mathematical expression for the DTFT is shown in Equation 10.1. Because the result is only dependent on the input signal $h[n]$ and not a previous $X(f_\omega)$, the $X(f_\omega)$ result can be

calculated faster in parallel by distributing ranges of f_ω amongst processing cores and recombining to produce an answer that spans the entire range of f_ω. The function is parallelised with the native "multiprocessing" module in python.

$$X(f_\omega) = \sum_{n=0}^{N-1} h[n] \times e^{\frac{-j2\pi f_\omega n}{N}} \tag{10.1}$$

Where:
n is the index of the sampled signal.
N is the total number of samples taken of the sampled signal.
$h[n]$ is an array containing the sampled signal.
f_ω is the frequency as a fraction of the sampling frequency being evaluated for.
$X(f_\omega)$ is the continuous periodic frequency spectrum of the sampled signal $h[n]$.

The implementation of the single threaded DTFT function is shown in in Listing 10.1 below.

```
#function that calculates dtft
def dft(xn, points):
  #define variables
  xfw = [0] * points
  fw = [0] * points

  #only interested in 0 - 0.5 times fw for same result with more
   spectral resolution
  step = (1/2 - 0) / points
  length = len(xn)

  #define frequency axis
  i = 0
  while (i < points):
    fw[i] = i * step
    i = i + 1

  #DTFT
  #for each output value
  i = 0
  while (i < points):
    sum = 0
    k = 0
    #for each input value
    while (k < length):
      sum = sum + (xn[k] * pow(math.e, (-1) * complex(0, 1) * 2 *
   math.pi * fw[i] * k))
      k = k + 1
    xfw[i] = abs(sum)
    i = i + 1

  return xfw
```

Listing 10.1: Implementation of the DTFT

The platforms other than the Bokkiebord chosen for this comparison include the desktop PC with an Intel Xeon processor running at 2933MHz on which the Bokkiebord is designed, the server with 2 Intel Xeon processors running at 2200MHz on which the software for the Bokkiebord is built, a BeagleBone Black running at 1000MHz and a first generation Raspberry Pi B running at 900MHz. The results are shown in figure 10.1.

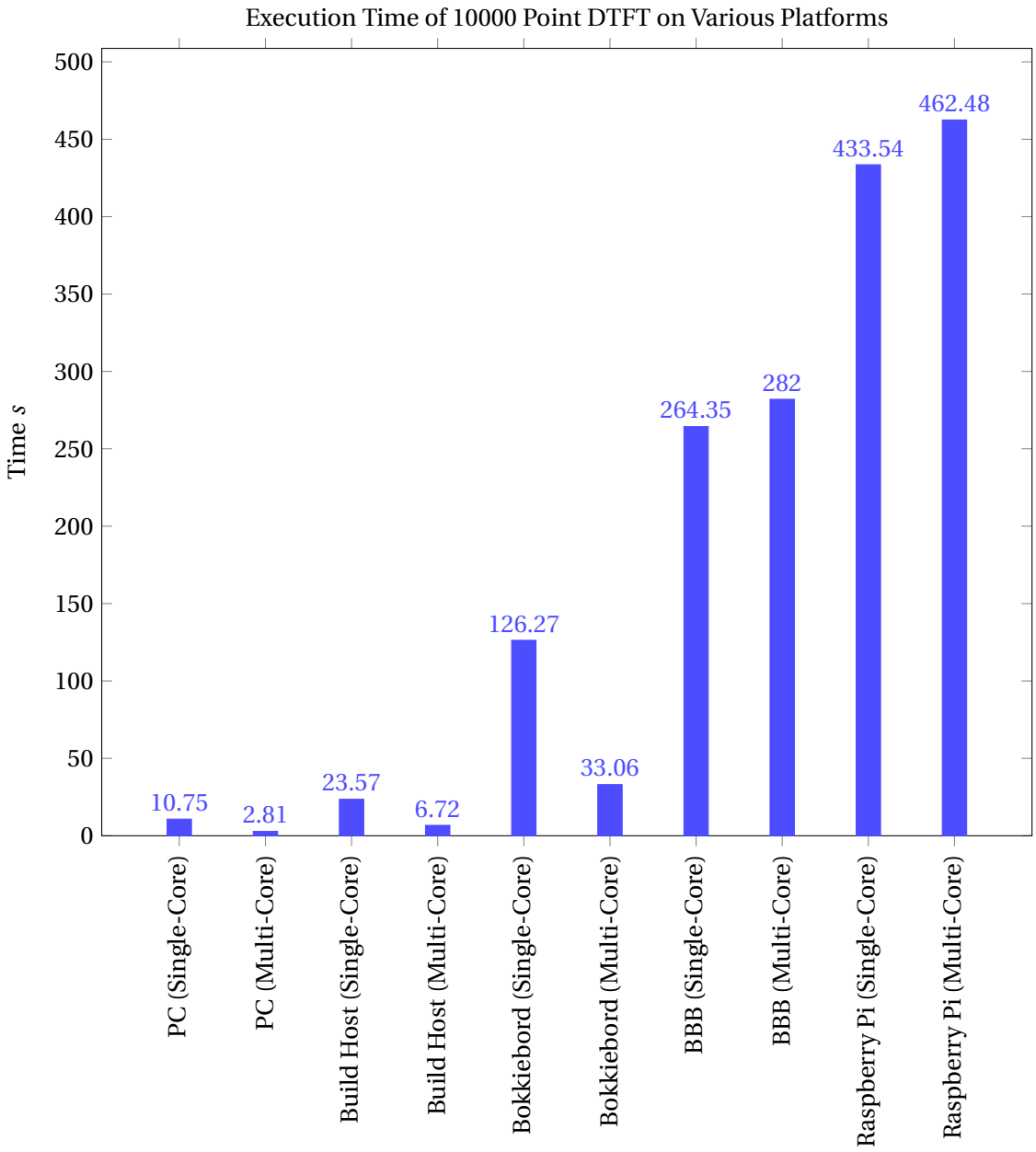


Figure 10.1: Bokkiebord Performance Measured Against Other Computers

It can be seen from the results in Figure 10.1 that the Bokkiebord is 2.1 times as fast as the BeagleBone Black when only 1 of 4 of the Bokkiebord's processing cores is used and that the Bokkiebord is 8 times as fast as the BeagleBone Black when all 4 cores are used.

It is interesting to note that multi-core software runs slower on single-core machines. This is because the platform pays the parallel processing overhead penalty without being able to capitalise on the greater speed offered by more cores.

The Bokkiebord is 3.4 times as fast as the Raspberry Pi if only 1 of 4 of the Bokkiebord's processing cores are used and 13.1 times as fast as the Raspberry Pi if all 4 cores are used.

Bokkiebord performance can be further increased by increasing the core speed from the current $800MHz$ to the maximum $1500MHz$. This can be done if core voltage is increased from $0.9V$ to $1.0V$ as discussed in the section 9.11.

The Bokkiebord is proven to offer significant performance advantages over the other commercially available single board computers based on the above results.

Chapter 11

Cost Analysis

The Bokkiebord PCB is manufactured by PCBWay in China. BGA components are sourced and placed by PCBWay and included in labour costs. Additional components are purchased from Mouser in the United States. Additional components are placed at the University of Stellenbosch.

Table 11.1 shows manufacturing specification that influence PCB cost. The main driving factors are number of layers, $4mil/4mil$ spacing and $0.2mm$ via holes.

Table 11.2 and Figure 11.1 shows total spend per service. Main cost factor is component cost which can be reduced by optimising supply lines to purchase from the manufacturer rather than from an intermediary. 57.970% of total cost is attributed to components, 23.827% is attributed to PCBs and 18.203% is attributed to labour for component placement. The total final cost per Bokkiebord is \$360.93 (R5197.39).

Table 11.3 and Figure 11.2 shows cost breakdown of component category. Integrated circuits are the largest cost contributor at \$85.76 (R1234.94), followed by connectors \$37.44 (R539.14), followed by capacitors at \$29.09 (R418.90).

This study is completed during the 2020/2021 global SARS-CoV-2 pandemic which causes major disruption in electronic component supply lines. This project is largely unaffected as components are purchased strategically early in the pandemic and in larger quantities. Components such as the i.MX8MQ processor and LPDDR4 Memory are currently completely unobtainable. Lead time on critical components is 53 weeks at the time of writing.

Factor Affecting Cost	**Bokkiebord Specification**
Choice of material	FR-4
Size and shape of PCB	55.88mm x 86.36mm, rectangular
Number of Layers	6-layers
Board finish	Immersion Gold
Size and number of holes	0.2mm
Minimum Trace and Space	$4mil/4mil$
Thickness of the PCB	1.6mm
Custom or unique specification	480 components, 445 interconnects, 2355 pads, 985 vias, 1136 holes including vias. Processor: 4 x ARM cortex A53 cores + 1 ARM cortex M4 core. External memory: 2GB Ram
Copper foil weight	1oz Cu
Drill to copper	0.25mm
Component sourcing and procurement	Catalogue supplier (Mouser) PCBWay

Table 11.1: Bokkiebord Specifications

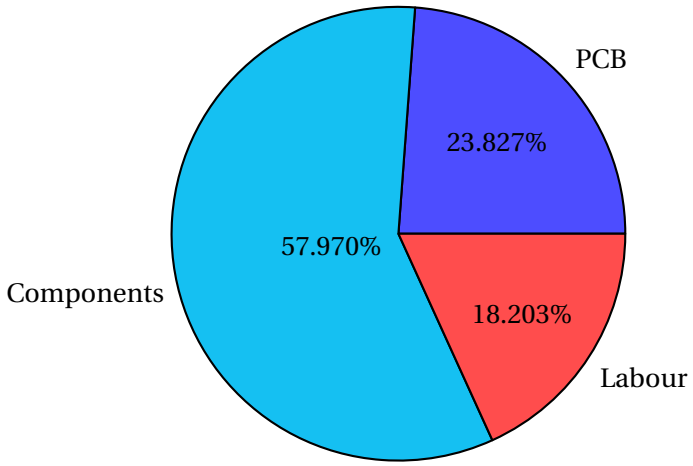


Figure 11.1: Bokkiebord Manufacturing Cost per Category

Component	**Price per Board** ($)	**Price Per Board** (*R*)
PCB (PCBWay)	86.00	1238.40
Components (PCBWay)	114.98	1655.71
Assembly (PCBWay)	65.70	746.08
Components (Mouser)	94.25	1357.20
Total	360.93	5197.39

Table 11.2: Bokkiebord Supplier Cost breakdown

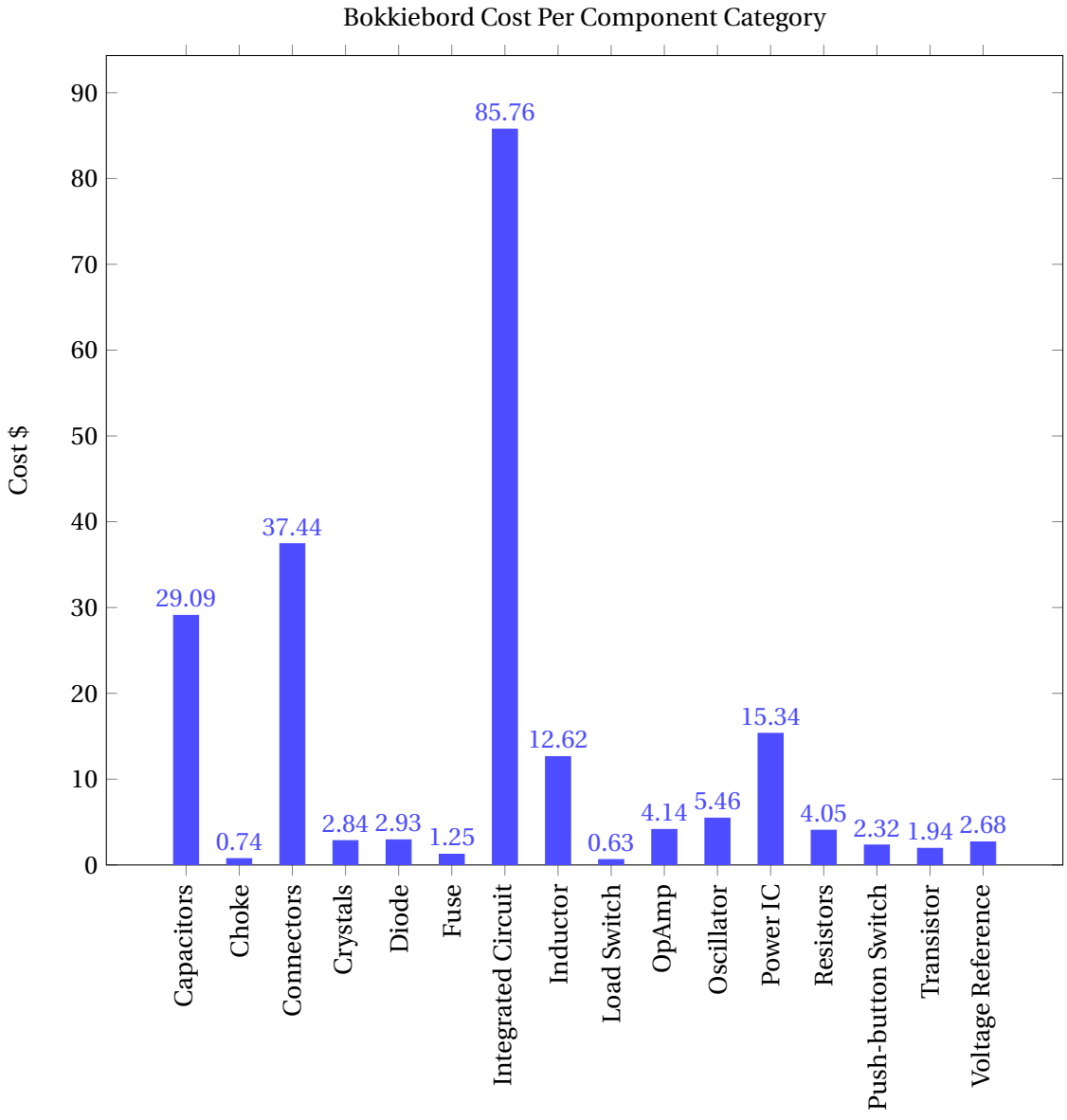


Figure 11.2: Bokkiebord Cost Per Component Type

Component	**Cost per Board** ($)	**Cost Per Board** (R)
Capacitor	29.09	418.90
Choke	0.74	10.66
Connector	37.44	539.14
Crystal	2.84	40.90
Diode	2.93	42.19
Fuse	1.25	18.00
Integrated Circuit	85.76	1234.94
Inductor	12.62	181.73
Load Switch	0.63	9.07
Op-Amp	4.14	59.62
Oscillator	5.46	78.62
Power IC	15.34	220.90
Resistor	4.05	58.32
Push-Button Switch	2.32	33.41
Transistor	1.94	27.94
Voltage Reference	2.68	38.59
Total	209.23	3012.91

Table 11.3: Component Cost Breakdown

Chapter 12

Conclusion

12.1 Objective

This project aims to build a 64-bit Linux Based single board computer specifically for visible light positioning. This single board computer becomes known as the Bokkiebord.

12.2 Requirements

The requirements are set by the 2 funding institutions, the University of Stellenbosch in South Africa and KU Leuven in Belgium. These requirements include that the Bokkiebord has a 64-bit processor, at least $2GB$ RAM, at least 2 USB 2.0 ports, a HDMI interface, utilise an SD card as primary storage, an Ethernet interface and an ADC capable of converting a signal with a sampling rate of $3MS/s$ (Chapter 2).

12.3 Processor Selection

A search is conducted to determine the most suitable processor. The i.MX8MQ is selected as it is a very powerful processor that provides 4 high performance ARM Cortex-A53 cores and 1 low power ARM Cortex-M4F core as coprocessor.

This processor offers a powerful GPU, dual 4K display interfaces, dual 4-lane MIPI camera interfaces, dual USB 3.0 interfaces, Ethernet and PCI express. The processor supports Android, Linux, Windows and FreeRTOS operating systems. The processor requires external 32-bit or 16-bit Double Data Rate memory that is expandable up to $8GB$ and runs at up to $3200MT/s$.

12.4 Power Subsystem

The i.MX8MQ requires 12 voltage rails with 7 power sequences.

The power system consists of a TPS54A24 input stage and a TPS659037 PMIC. The TPS54A24 is selected for its small size and high current capacity of up to $10A$. The TPS659037 is selected for its 14 voltage rails, high current capacity and because it requires fewer external components than the PF4210 recommended by NXP.

The TPS659037 is the main power supply for all components on the Bokkiebord except the USB.

USB power is delivered from the first stage regulator via a load switch. DDR voltage reference is provided by a buffered voltage divider circuit.

12.5 Memory Subsystem

$2GB$ of LPDDR4 memory is implemented by means of a MT53E512M32D2NP-046 WT:E LPDDR4 memory module. The memory runs at up to $2133MHz$, the cycle time is $468ps$. LPDDR4 is chosen as it only requires 1 component to be implemented and can be powered from only two voltage rails.

12.6 Peripherals

The ADC is a 12-bit $3MS/s$ SAR ADS7046. It is selected for its $3MS/s$ sampling frequency, small size and SPI Bus communication.

The i.MX8MQ supports RMII and RGMII network communication standards. $100Mbps$ Ethernet is implemented by means of the DP83825 Ethernet PHY. It is primarily chosen for its small size, single $3.3V$ voltage rail requirement and the fact that its driver is implemented in the Linux kernel.

The expansion header interface is based on that of the BeagleBone Black and although the interfaces are mostly identical, some differences do exist. Analogue input pins used on the BeagleBone Black are unused on the Bokkiebord and the ADC functionality is moved to a uSMA connector at the ADC.

HDMI 2.0 is implemented via a Micro-HDMI connector and is protected from ESD and EMI by means of PCMF3HDMI2S and PCMF2HDMI2S filters.

A JTAG debugging interface is provided on the periphery of the board.

The i.MX8MQ has 2 MIPI CSI ports and 1 MIPI DSI port. The display interface functions independently from the HDMI interface. These interfaces are implemented to be electrically compatible with the Raspberry Pi interface.

An SD Card is used as primary storage and SD/eMMC4.3 (single data rate) and SD/eMMC4.4/4.41 (double data rate) standards are supported.

A CP2105 USB-to-UART Bridge is implemented for debugging purposes. The CP2105 from Silicon Labs is specifically chosen as it is recommended by NXP for use with its debugging software.

2 USB 3.0 ports are implemented and protected from ESD and EMI by means of 2 PCMF3USB3S filters. The NX5P3290 current limited power switch is used to protect the USB interface from over-current condition.

12.7 Printed Circuit Board Layout

The i.MX8MQ processor is available in a $17mm$ x $17mm$ FCBGA package with 621 pins and $0.65mm$ pin pitch spacing. A $86.36mm$ x $55.88mm$ PCB with 6-layers, $1.6mm$ thickness, $0.2mm$ via hole size, $4mil/4mil$ spacing and $1oz$ copper is selected to implement the i.MX8MQ processor.

The i.MX8MQ processor requires 42Ω and 50Ω single ended and 85Ω, 90Ω and 100Ω differential impedances. A custom 6-layer stackup is designed to realise these impedances while minimising trace thickness and cross talk. A 10-layer stackup is also designed in case more layers are required. 6-layers are ultimately chosen to contain costs. All the required impedances are realised to within 5% with standard materials normally stocked by the PCB manufacturer.

Timing constraints of high speed interfaces are achieved by calculating the trace and via propagation delay. Custom vias are designed to achieve the required timing delay while maintaining impedances. The custom designed via has a delay of $23.28ps$.

The LPDDR4 memory regards signals with voltages lower than $0.3 \times VDD$ to be a logical low and signals voltages higher than $0.70 \times VDD$ to be a logical high. A 10% coupling coefficient is maintained so that the worst possible case of 2 constructive interfering aggressor traces leaves a 10% safety margin. This is achieved by a combination of trace separation and orthogonal serpentine traces.

Return path integrity is maintained throughout the design by routing all high-speed signal traces with the contour of their respective power planes.

Power integrity is maintained by means of adequate decoupling, wide power planes and unimpeded return path.

12.8 Manufacturing

PCBs are manufactured by PCBWay in China as the $0.2mm$ via requirement cannot be manufactured locally. Complex BGA components are mounted by PCBWay while smaller components are placed by the student with the aid of a soldering microscope.

Solder flux is removed by ultrasonic ethanol bath and excess moisture is removed by baking in an oven at 60^o for $30m$. Figure 1.1 - 1.2 show the completed Bokkiebord.

12.9 Debugging

Each of the subsystems of the Bokkiebord are built and tested independently before combining together in the larger Bokkiebord. The Bokkiebord is designed, manufactured and debugged completely without the aid of a development board.

The 1st power input stage is built and found to function correctly. The TPS659037 PMIC is built and found to not be fully programmable. A mod board is designed to address this. The power stages are then combined and found to function correctly.

The rest of the board is then built and checked for short circuits. Power is applied to the board.

It is found that the internal $32kHz$ RTC oscillator has insufficient loop gain and an external RTC oscillator is added to correct this.

The processor is started and connected with a JTAG and found to be working correctly.

The USB-OTG and UART-to-USB-Bridge is built and connected to a computer.

Memory initialisation is then performed and the memory is found to function reliably up to $1580MHz$. The result from the memory initialisation step is then used to build a custom Linux image which is loaded onto the Bokkiebord and booted. Linux is found to function reliably.

The Ethernet PHY obtains a network link, and is detected by the Linux operating system, but is currently not able to send packets to the network. A possible reason is that the Ethernet PHY's driver is currently unsupported in the operating system.

The HDMI interface is proven to be electrically sound, but does not detect a HPD signal. Possible reasons are that the monitor provides a low HPD voltage or that the pull-down resistance of the HDMI port on the Bokkiebord is too low.

The ADC is built and tested to function correctly.

12.10 Software

Linux is ported to the Bokkiebord. The image is compiled with the Yocto project and consists of security firmware, bootloader, Linux kernel and application image. Memory timing calibration data is compiled directly into the bootloader binary. Device tree modifications are done to describe custom hardware peripherals.

12.11 Performance

The performance of the Bokkiebord is compared against other platforms including the Beagle Bone Black and Raspberry Pi. This is done by calculating a DTFT of 10000 points and comparing execution time.

It is found that the Bokkiebord is 2.1 times as fast as the BeagleBone Black when only 1 of 4 of the Bokkiebord's processing cores are used and that the Bokkiebord is 8 times as fast as the BeagleBone Black when all 4 cores are used.

The Bokkiebord is 3.4 times as fast as the Raspberry Pi if only 1 of 4 of the Bokkiebord's processing cores are used and 13.1 times as fast as the Raspberry Pi if all 4 cores are used. Bokkiebord performance can be further increased by increasing the core speed from the current $800MHz$ to the maximum $1500MHz$.

12.12 Cost Analysis

The final manufacturing cost of the 1st Bokkiebord is \$360.93 ($R5197.39). Components are 57.97% of total cost, PCBs are 23.827% of total cost and labour is 18.203% of total cost. A substantial reduction in this cost is possible if more PCBs are manufactured per standard manu-

facturing panel and components are ordered in higher quantities.

The Bokkiebord successfully achieves its primary requirements as set out at the start of the project within cost expectations.

This thesis provides sufficient design and software detail for continuation of the Bokkiebord project or as a knowledge base for other embedded designs.

Chapter 13

Recommendations

Future improvement to the Bokkiebord include:

- **Power System**

  The TPS659037 PMIC is not as well suited for this project as initially thought. This can be optimised in 1 of 2 ways.

  - TPS659037 is retained: The load switches and RTC oscillator as discussed in Chapter 8 could be integrated into the PCB. This option also requires the addition of a secondary circuit that writes I2C startup procedure upon power-up. This option introduces extra complexity into the design.
  - PF4210 PMIC is used instead: A new chip with unknown complexity is integrated into the design. This option requires the introduction of extra voltage regulators.

- **PCB Layout:**

  An 8- or 10-layer board is recommended in a less cost constrained project as the routing are is highly constrained in this 6-layer design.

Bibliography

[1] Robin Amsters, Dimiter Holm, Joren Joly, Eric Demeester, Nobby Stevens, and Peter Slaets. Visible light positioning using bayesian filters. *Journal of Lightwave Technology*, 38(21):5925–5936, 2020.

[2] C. Barnatt. Single board computers in 2021. state of play. `https://www.youtube.com/watch?v=RcvMxC81r_g`, June 2021.

[3] Sander Bastiaens, Willem Raes, Nobby Stevens, Luc Martens, Wout Joseph, and David Plets. Impact of a photodiode's angular characteristics on rss-based vlp accuracy. *IEEE Access*, 8:83116–83130, 2020.

[4] Steven De Lausnay, Lieven De Strycker, Jean-Pierre Goemaere, Bart Nauwelaers, and Nobby Stevens. A survey on multiple access visible light positioning. In *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, pages 38–42, 2016.

[5] Analog Devices. Microstrip and stripline design. `https://www.analog.com/media/en/training-seminars/tutorials/MT-094.pdf`, September 2021.

[6] Devicetree.org. Devicetree specification release v0.3. `https://github.com/devicetree-org/devicetree-specification/releases/tag/v0.3`, October 2021.

[7] Read The Docs. The u-boot documentation. `https://u-boot.readthedocs.io/en/latest/`, October 2021.

[8] Ultracad Douglas Brook. How loud is your crosstalk. `https://www.ultracad.com/articles/crosstlk.pdf`, September 2021.

[9] T. C. (Terence Charles) Edwards. *Foundations of interconnect and microstrip design*. John Wiley, Chichester, 3rd ed. edition, 2000.

[10] Wurth Elektronik. Webinar 2013: Improved signal integrity through impedance matched circuit boards. `https://www.we-online.com/web/en/index.php/show/media/04_leiterplatte/2013_1/webinare_1/signalintegritaet/Webinar_Signal_final_engl.pdf`, September 2021.

[11] Texas Instruments E2E Help Forum. Tps659037: Changing osc16m_cfg otp bit. `https://e2e.ti.com/support/power-management-group/power-management/f/power-management-forum/772642/tps659037-changing-osc16m_cfg-otp-bit`, October 2021.

[12] Texas Instruments E2E Help Forum. Tps659037: Clk32kgo output. `https://e2e.ti.com/support/power-management-group/power-management/f/power-management-forum/572181/tps659037-clk32kgo-output`, October 2021.

[13] Texas Instruments E2E Help Forum. Tps659037: Operating without otp power sequence and embedded power controller? https://e2e.ti.com/support/power-management-group/power-management/f/power-management-forum/969436/tps659037-operating-without-otp-power-sequence-and-embedded-power-controller, October 2021.

[14] BeagleBoard Foundation. Beagleboard.org - black. https://beagleboard.org/black, May 2018.

[15] Beaglebord Foundation. About beagleboard.org and the beagleboard.org foundation. https://beagleboard.org/about, October 2021.

[16] Raspberry Pi Foundation. Raspberry pi compute module io board. https://datasheets.raspberrypi.com/cmio/cmio-schematics.pdf, October 2016.

[17] Raspberry Pi Foundation. Teach, learn and make with raspberry pi. https://www.raspberrypi.org/, October 2021.

[18] Little Fuse. picosmdc series surface mount datasheet. https://www.littelfuse.com/~/media/electronics/datasheets/resettable_ptcs/littelfuse_ptc_picosmdc_datasheet.pdf.pdf, October 2021.

[19] HackerBoards.com. The single board computer database. http://www.hackerboards.com, March 2021.

[20] Nick Heath. Inside the raspberry pi: The story of the usd35 computer that changed the world. https://www.techrepublic.com/article/inside-the-raspberry-pi-the-story-of-the-35-computer-that-changed-the-world/, December 2018.

[21] C. Hoffman. Google's killing chrome support for 32-bit linux, ubuntu 12.04 and debian 7. https://www.pcworld.com/article/3010404/browsers/googles-killing-chrome-support-for-32-bit-linux-ubuntu-1204-and-debian-7.html, May 2018.

[22] ARM Holdings. What is risc? https://www.arm.com/glossary/risc, October 2021.

[23] Barracuda Holdings. Barracuda capabilities. https://www.barracudaholdings.co.za/factory-capability-services, October 2021.

[24] Martin Graham Howard Johnson. *High-Speed Digital Design - A Handbook of Black Magic*.

[25] Diodes Incorporated. Dmp1045u p-channel enhancement mode mosfet. https://www.diodes.com/assets/Datasheets/DMP1045U.pdf, October 2021.

[26] Texas Instruments. Sitara processor overview. http://www.ti.com/processors/sitara/overview.html, May 2018.

[27] Texas Instruments. Dp83867ir/cr robust, high immunity 10/100/1000 ethernet physical layer transceiver. https://www.ti.com/lit/ds/symlink/dp83867ir.pdf, December 2019.

[28] Texas Instruments. Ads7046 12-bit, 3-msps, single-ended input, small-size, low-power sar adc. https://www.ti.com/lit/ds/symlink/ads7046.pdf, October 2021.

[29] Texas Instruments. Customizing a yocto-based linux distribution for production. https://training.ti.com/customizing-yocto-based-linux-distribution-production, October 2021.

[30] Texas Instruments. Dp83825i low power 10/100 mbps ethernet physical layer transceiver. https://www.ti.com/lit/ds/symlink/dp83825i.pdf, October 2021.

[31] Texas Instruments. Driver for the texas instruments dp83822, dp83825 and dp83826 phys. https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/tree/drivers/net/phy/dp83822.c, October 2021.

[32] Texas Instruments. Ethernet system software on sitara am-class processors. https://training.ti.com/sites/default/files/docs/ETH_System_SW_on_AMClass_Devices.pdf, October 2021.

[33] Texas Instruments. How to meet the hdmi 5 v source requirement. https://www.ti.com/lit/an/snva412c/snva412c.pdf, October 2021.

[34] Texas Instruments. Lmv951 1-v, 2.7-mhz, zero crossover rail-to-rail input and output amplifier with shutdown. https://www.ti.com/lit/ds/symlink/lmv951.pdf, October 2021.

[35] Texas Instruments. Pru-icss industrial software for sitara processors. https://www.ti.com/tool/PRU-ICSS-INDUSTRIAL-SW, October 2021.

[36] Texas Instruments. Ref19xx low-drift, low-power, dual-output, vref and vref / 2 voltage references. https://www.ti.com/lit/ds/symlink/ref1933.pdf, October 2021.

[37] Texas Instruments. Tps22918 5.5-v, 2-a, 52-mohm on-resistance load switch. https://www.ti.com/lit/ds/symlink/tps22918.pdf, October 2021.

[38] Texas Instruments. Tps54a24 4.5-v to 17-v input, 10-a synchronous swift step-down converter. https://www.ti.com/lit/ds/symlink/tps54a24.pdf, October 2021.

[39] Texas Instruments. Tps659037 power management unit (pmu) for processor. https://www.ti.com/lit/ds/symlink/tps659037.pdf, October 2021.

[40] Texas Instruments. Tps659037 register map - technical reference manual. https://www.ti.com/lit/ug/sliu015b/sliu015b.pdf, October 2021.

[41] Texas Instruments. Tps659037 user's guide to power am574x, am572x, and am571x. https://www.ti.com/lit/ug/sliu011f/sliu011f.pdf, October 2021.

[42] Trax Interconnect. Trax capabilities. https://www.trax.co.za/capabilities, October 2021.

[43] Howard Johnson. Serpentine delays. http://www.sigcon.com/Pubs/edn/serpentine.htm, September 2021.

[44] Ramesh Garg Inder Bahl Prakash Bhartia KC Gupta. *Microstrip lines and slotlines*. Artech House, Boston Mass, 2nd ed. edition.

[45] Keysight. Pathwave advanced design system (ads). https://www.keysight.com/zz/en/products/software/pathwave-design-software/pathwave-advanced-design-system.html, October 2021.

[46] Silicon Labs. Cp2105 single-chip usb to dual uart bridge. https://www.silabs.com/documents/public/data-sheets/CP2105.pdf, September 2021.

[47] C. Merriman. Ubuntu backs calls to wind down 32-bit linux support. https://www.theinquirer.net/inquirer/news/2463739/ubuntu-backs-calls-to-wind-down-32-bit-linux-support, May 2018.

[48] Microchip. 32-bit microprocessors. https://www.microchip.com/design-centers/32-bit-mpus, May 2018.

[49] Donald A. Neamen. *Microelectronics, circuit analysis and design*. McGraw-Hill International, 4th ed. edition, 2010.

[50] NXP. 14-channel power management integrated circuit (pmic) for audio/video applications. https://www.nxp.com/docs/en/data-sheet/PF4210.pdf, November 2018.

[51] NXP. i.mx 8 series application processors. https://www.nxp.com/products/processors-and-microcontrollers/applications-processors/i.mx-applications-processors/i.mx-8-processors/i.mx-8m-family-armcortex-a53-cortex-m4-audio-voice-video:i.MX8M, May 2018.

[52] NXP. i.mx porting guide. Secrecy Agreement Required, June 2020.

[53] NXP. Crystal oscillator troubleshooting guide. https://www.nxp.com/docs/en/application-note/AN3208.pdf, October 2021.

[54] NXP. Embedded linux for i.mx applications processors. https://www.nxp.com/design/software/embedded-software/i-mx-software/embedded-linux-for-i-mx-applications-processors:IMXLINUX, October 2021.

[55] NXP. i.mx 8m dual / 8m quadlite / 8m quad applications processors data sheet for industrial products. https://www.nxp.com/docs/en/data-sheet/IMX8MDQLQIEC.pdf, September 2021.

[56] NXP. i.mx 8m dual / 8m quadlite / 8m quad applications processors data sheet for industrial products. https://www.nxp.com/docs/en/data-sheet/IMX8MDQLQIEC.pdf, October 2021.

[57] NXP. i.mx 8m dual/8m quadlite/8m quad applications processors reference manual. Secrecy Agreement Required, October 2021.

[58] NXP. i.mx 8m family ddr tool release. https://community.nxp.com/t5/i-MX-Processors-Knowledge-Base/i-MX-8M-Family-DDR-Tool-Release/ta-p/1104467, October 2021.

[59] NXP. i.mx linux user's guide. https://www.nxp.com/docs/en/user-guide/IMX_LINUX_USERS_GUIDE.pdf, September 2021.

[60] NXP. i.mx yocto project user's guide. https://www.nxp.com/docs/en/user-guide/IMX_YOCTO_PROJECT_USERS_GUIDE.pdf, 30 September 2021.

[61] NXP. Universal update utility release 1.4.165. https://github.com/NXPmicro/mfgtools/releases/tag/uuu_1.4.165, October 2021.

[62] NXP. Usb pd and type-c current-limited power switch. https://www.nxp.com/docs/en/data-sheet/NX5P3290.pdf, October 2021.

[63] CLIFF ORTMEYER. Then and now: A brief history of single board computers. https://community.element14.com/learn/publications/element14-tech-journal/m/files/17221, 2014.

[64] Oshpark. 4 layer prototype service. https://docs.oshpark.com/services/four-layer/, September 2021.

[65] PCBWay. Blind vias and buried vias. https://www.pcbway.com/blog/Engineering_Technical/Blind_vias_and_Buried_Vias.html, September 2021.

[66] PCBWay. Cost cores foil & prepreg for pcbs. https://www.pcbway.com/pcb_prototype/Cost_cores_Foil___Prepreg_for_PCBs.html, September 2021.

[67] PCBWay. Etching outer layer of circuit board. https://www.pcbway.com/blog/Engineering_Technical/The_Printed_Circuit_Board_Fabrication.html, September 2021.

[68] PCBWay. How to prevent the pcb from being bended and warped when get through the reflow furnace. https://www.pcbway.com/blog/Engineering_Technical/How_to_prevent_the_PCB_from_being_bended_and_warped_when_get_through_the_reflow_furnace.html, September 2021.

[69] PCBWay. Pcb way online calculator tool. https://www.pcbway.com/orderonline.aspx, September 2021.

[70] PCBWay. Pcbway capabilities. https://www.pcbway.com/capabilities.html, October 2021.

[71] PCBWay. Stackup for 4,6,8 layers multi-layer laminated structure. https://www.pcbway.com/multi-layer-laminated-structure.html, September 2021.

[72] Thomas Petazzoni. Device tree for dummies. https://elinux.org/images/f/f9/Petazzoni-device-tree-dummies_0.pdf, October 2021.

[73] David M Pozar. *Microwave Engineering*. John Wiley, Amherst, 4rd ed. edition.

[74] Abhishek Prakash. Are we looking at the end of 32 bit linux? https://itsfoss.com/end-of-32-bit-linux/, May 2018.

[75] John G. Proakis. *Digital signal processing*. Prentice Hall, New Delhi, India, 4th ed. edition, 2011 - 2007.

[76] Yocto Project. Getting started: The yocto project overview. https://www.yoctoproject.org/software-overview/, October 2021.

[77] Trax Interconnect (pty) ltd. Multilayer stackups. https://www.trax.co.za/copy-of-4-layer-stackups, September 2021.

[78] Python. Python. https://www.python.org/, October 2021.

[79] Linux From Scratch. Welcome to linux from scratch! https://www.linuxfromscratch.org/, October 2021.

[80] Simberian. Simbeor electromagnetic signal integrity software. https://www.simberian.com/, September 2021.

[81] BAE Systems. What are single-board computers? https://www.baesystems.com/en-us/definition/what-are-single-board-computers, October 2021.

[82] Kingston Technology. microsdhc memory card flash storage media. https://www.kingston.com/datasheets/SDCIT-specsheet-8gb-32gb_en.pdf, October 2021.

[83] Micron Technology. Lpddr4x sdram - mt53e512m32d2, mt53e1g32d4 datasheet. Secrecy Agreement Required, November 2020.

[84] J Titus. Build dyna-micro 8080 computer. May 1976.

[85] Kevin Verniers, Liesbet Van der Perre, and Nobby Stevens. Assessment of a beaglebone black high sampling rate digital waveform generator. In *2018 International Symposium on Rapid System Prototyping (RSP)*, pages 36–41, 2018.

[86] Kevin Verniers, Liesbet Van der Perre, and Nobby Stevens. Assessment of a beaglebone black high sampling rate digital waveform generator. In *2018 International Symposium on Rapid System Prototyping (RSP)*, pages 36–41, 2018.

[87] Hubregt J Visser. *Antenna Theory and Applications*. John Wiley, Holst Centre imec The Netherlands, 1st ed. edition.

[88] Wikipedia. Consumer electronics control. https://en.wikipedia.org/wiki/Consumer_Electronics_Control, October 2021.

[89] Wikipedia. Display data channel. https://en.wikipedia.org/wiki/Display_Data_Channel, October 2021.

[90] Wikipedia. Extended display identification data. https://en.wikipedia.org/wiki/Extended_Display_Identification_Data#Enhanced_EDID_(E-EDID), October 2021.

[91] Wikipedia. Hdmi - display data channel (ddc). https://en.wikipedia.org/wiki/HDMI#Display_Data_Channel_(DDC), October 2021.

[92] Wikipedia. High-bandwidth digital content protection. https://en.wikipedia.org/wiki/High-bandwidth_Digital_Content_Protection, October 2021.

[93] Wikipedia. Transition-minimized differential signaling. https://en.wikipedia.org/wiki/Transition-minimized_differential_signaling, October 2021.

Appendices

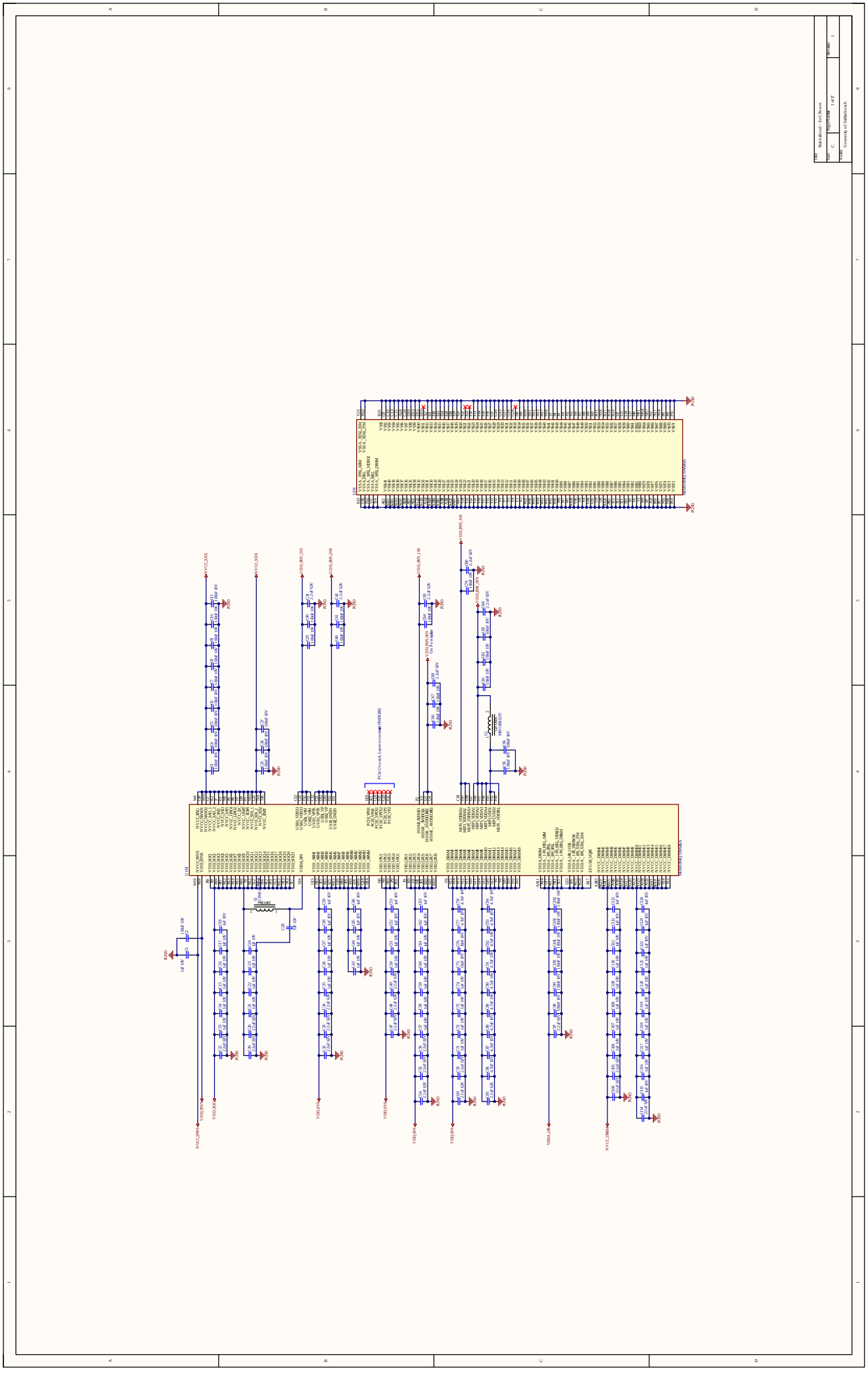
Appendix A

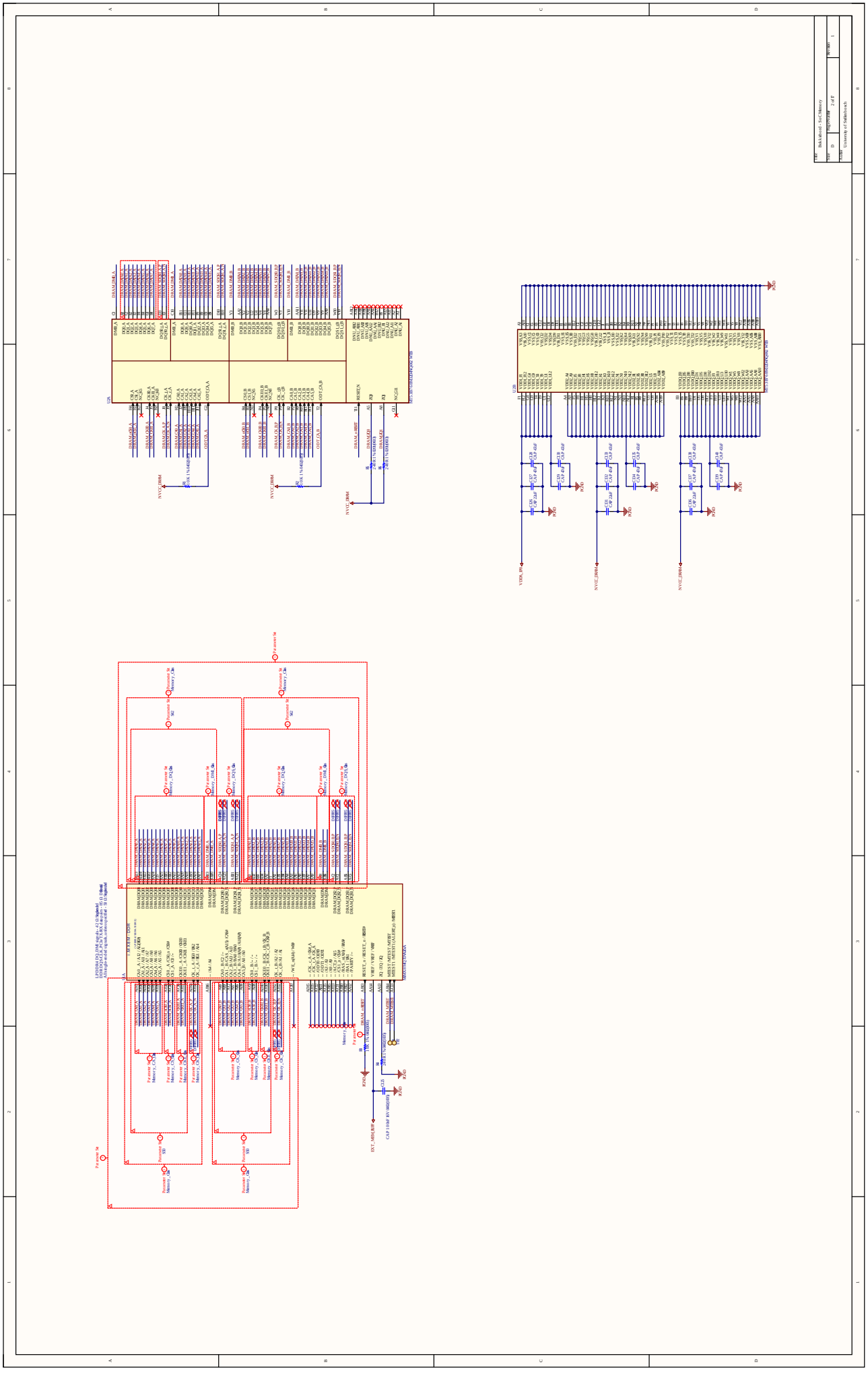
Bokkiebord

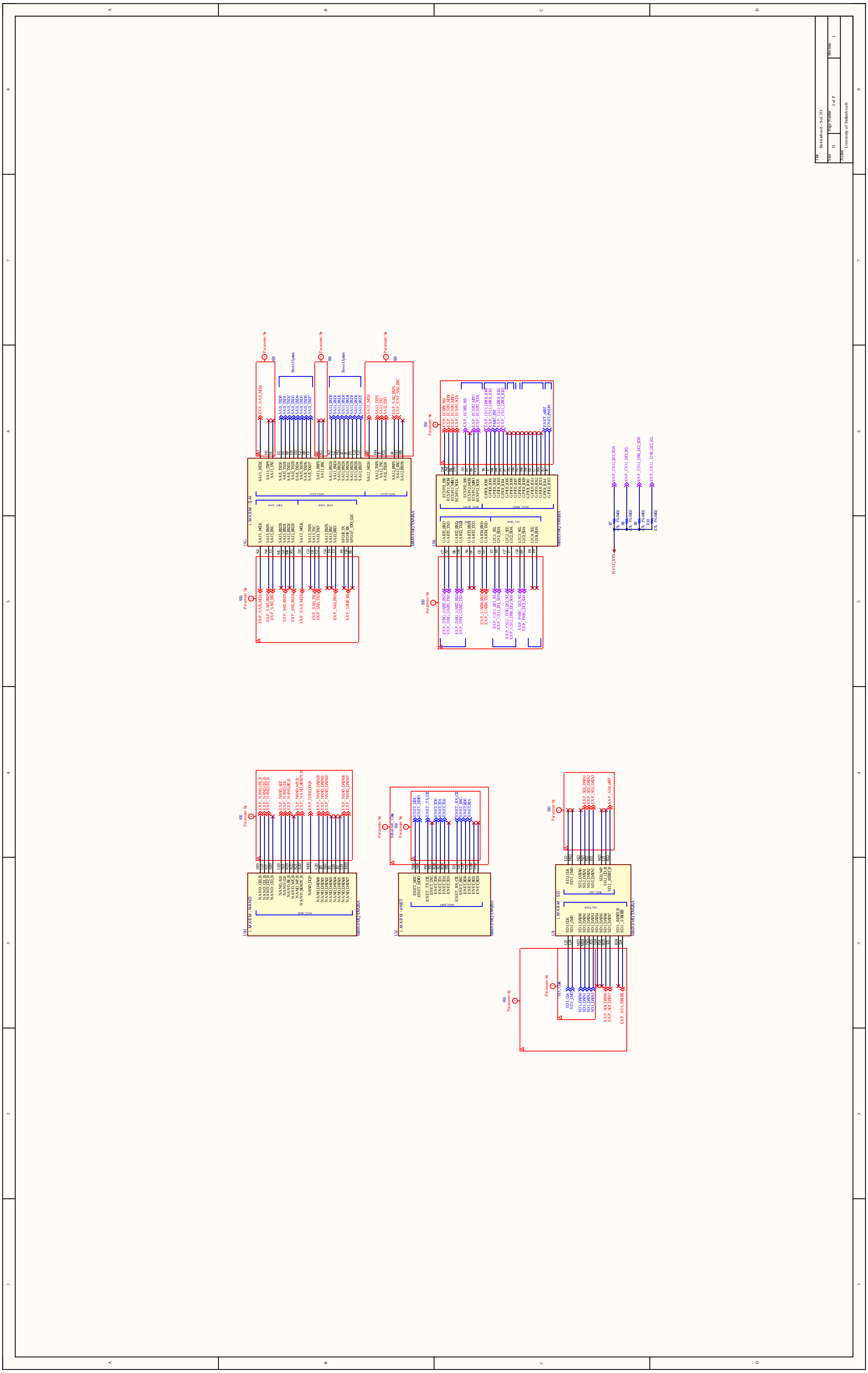
A.1 Standard Bokkiebord Design

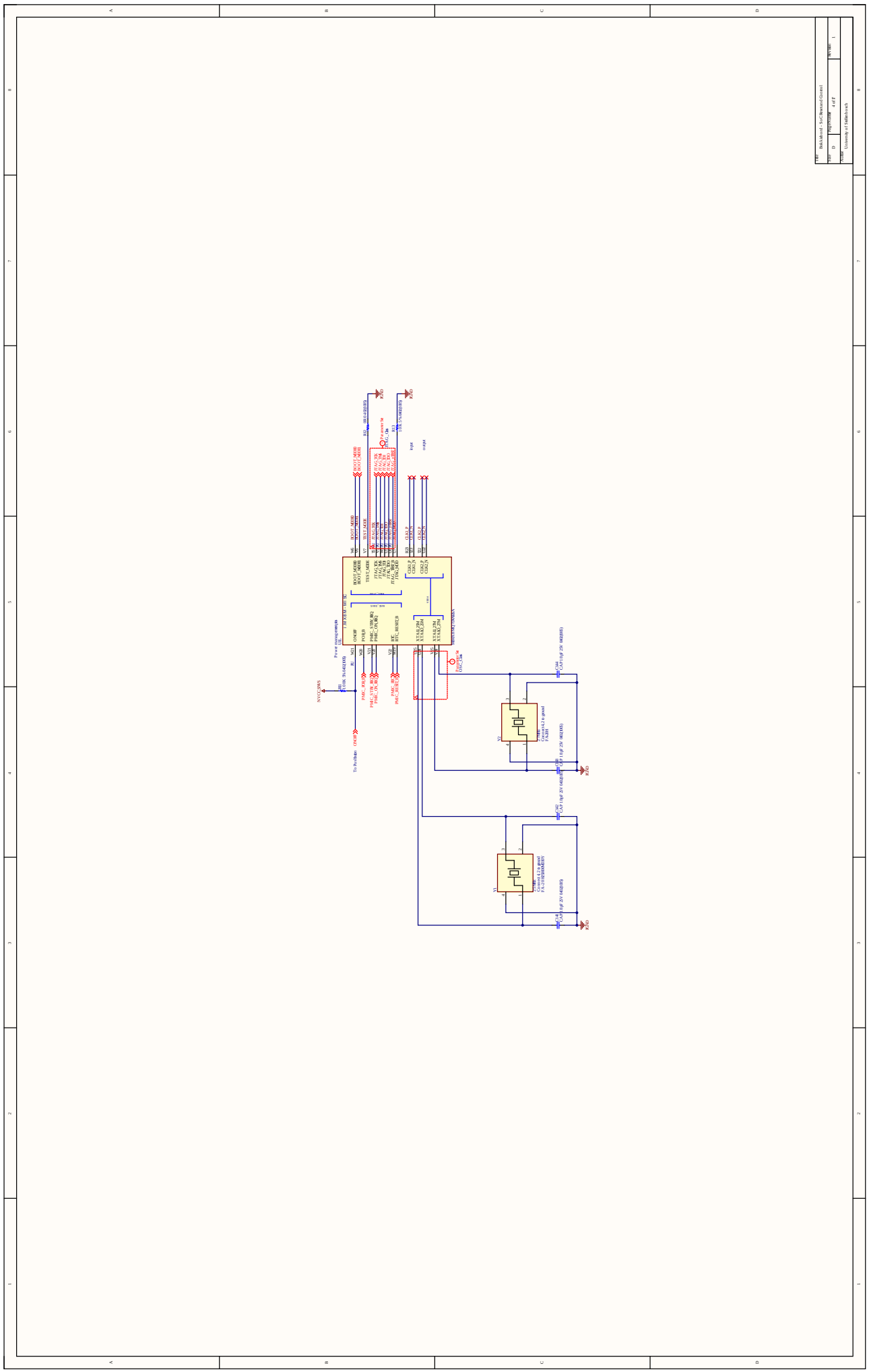
Schematic order:

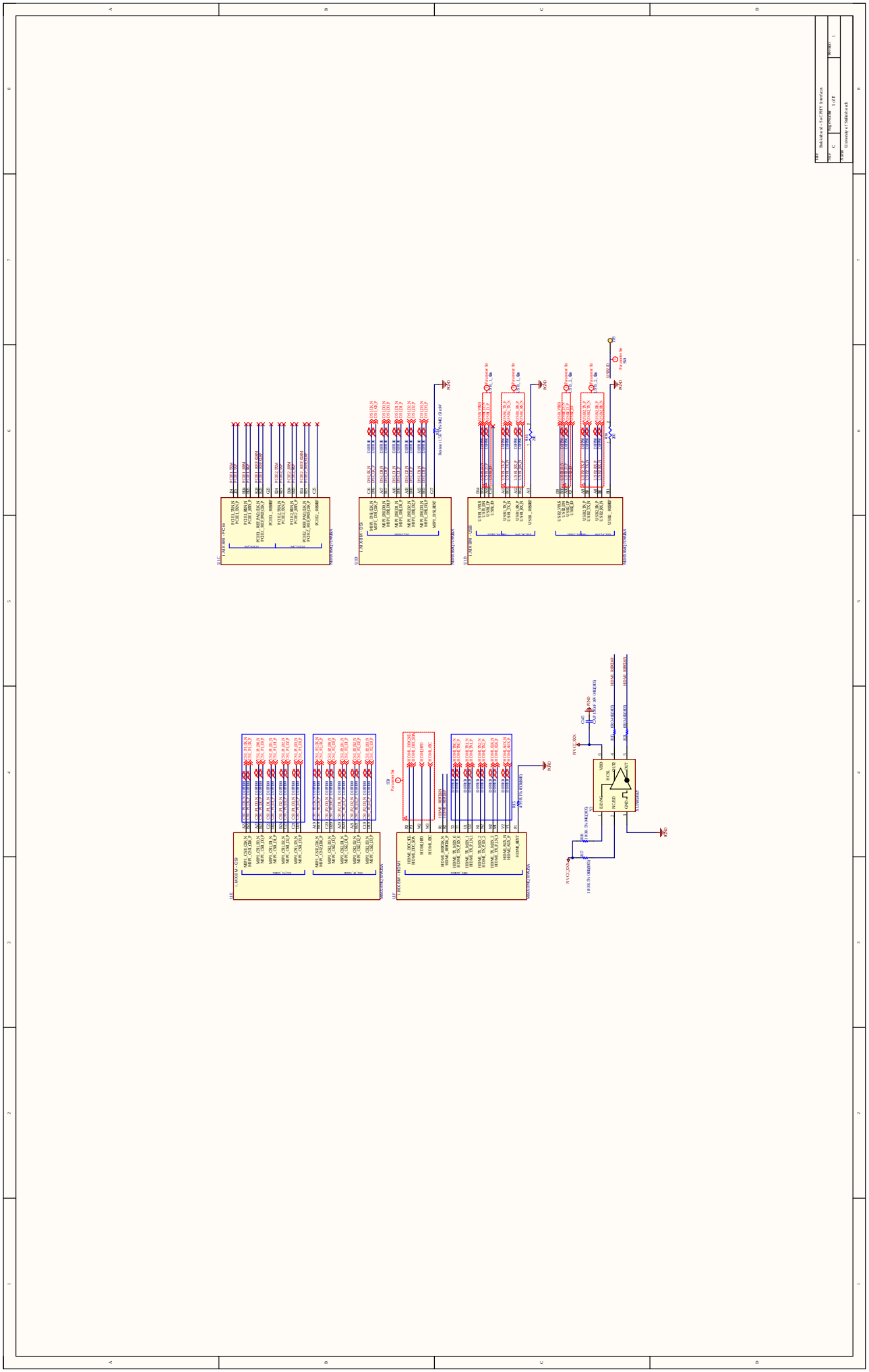
- SoC Power
- SoC Memory
- SoC IO
- SoC Reset and Control
- SoC PHY Interfaces
- SoC Storage
- BB Power System 1/2
- BB Power System 2/2
- BB Boot Configuration
- SoC USB
- SoC HDMI
- BB ADC
- BB MIPI
- BB Debug 1/2
- BB Debug 2/2
- BB Expansion Connectors
- BB Ethernet

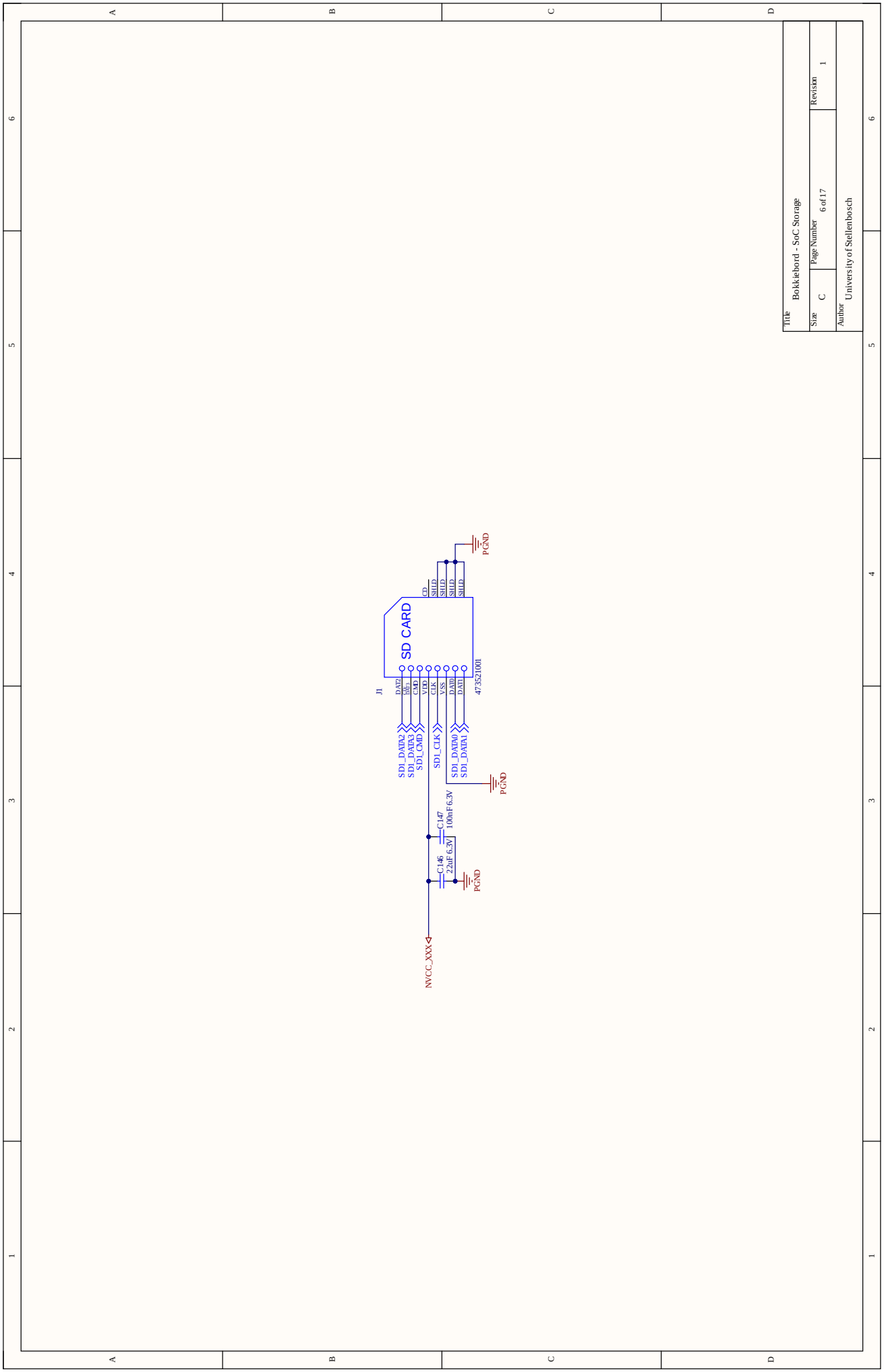


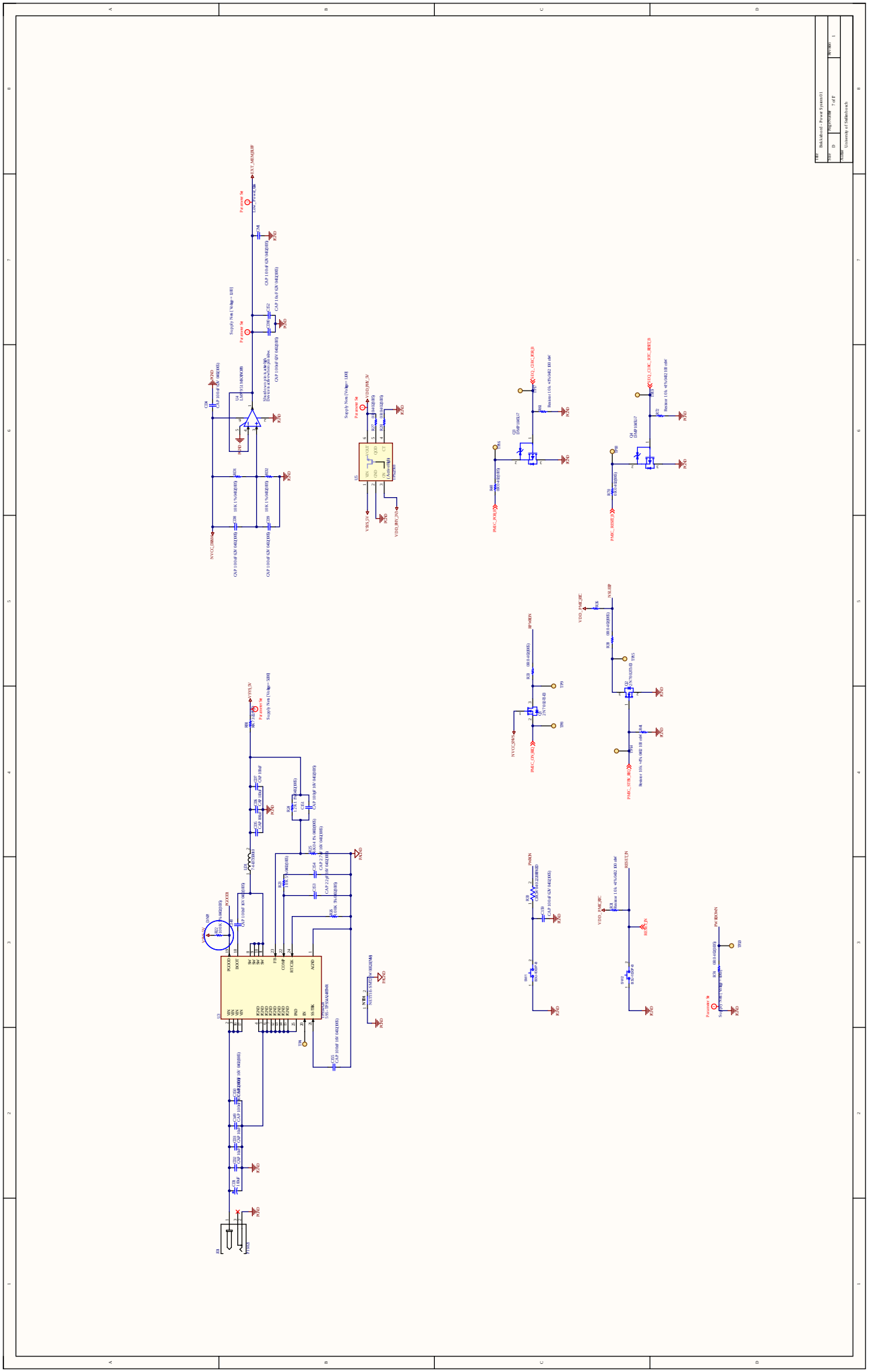


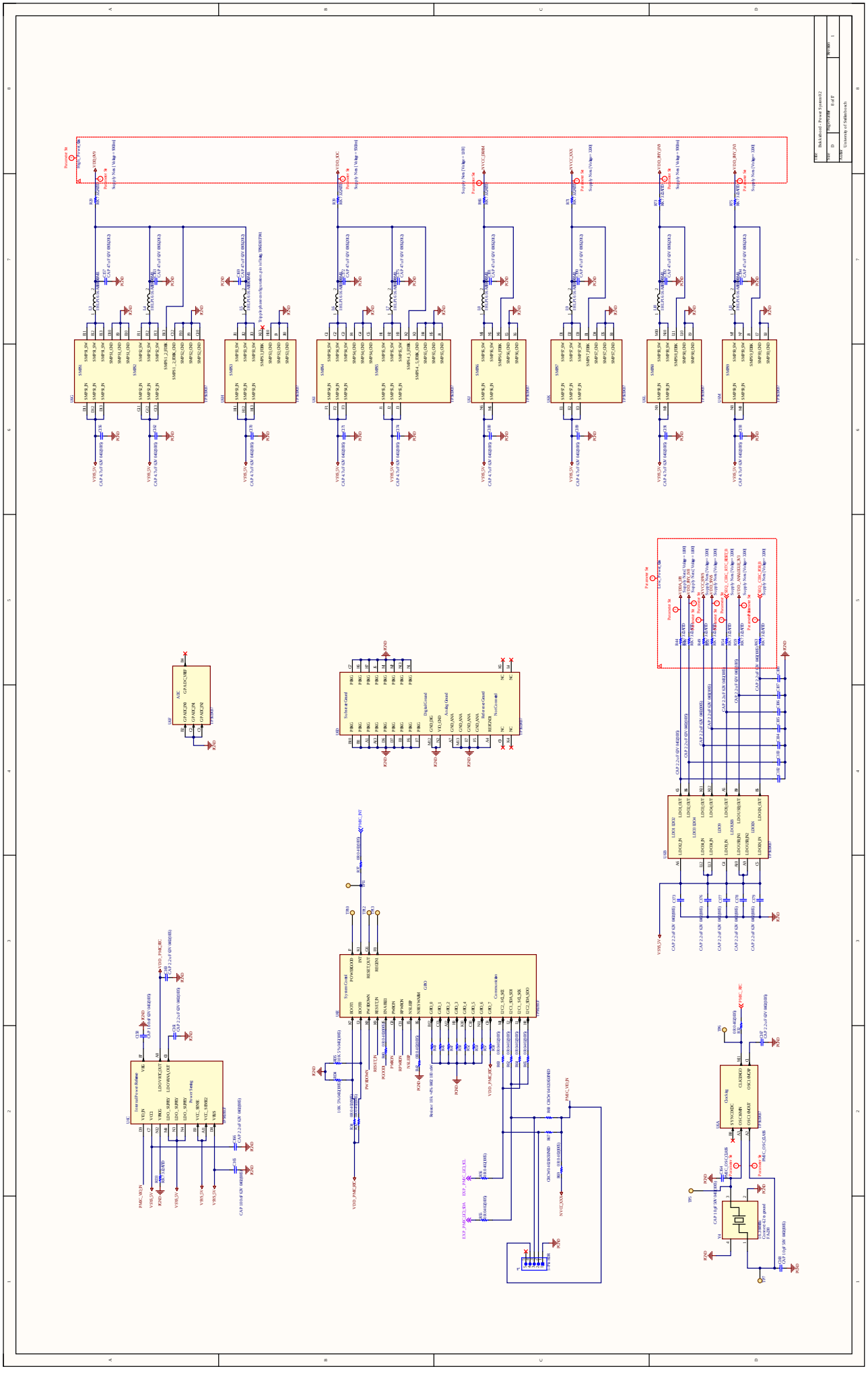


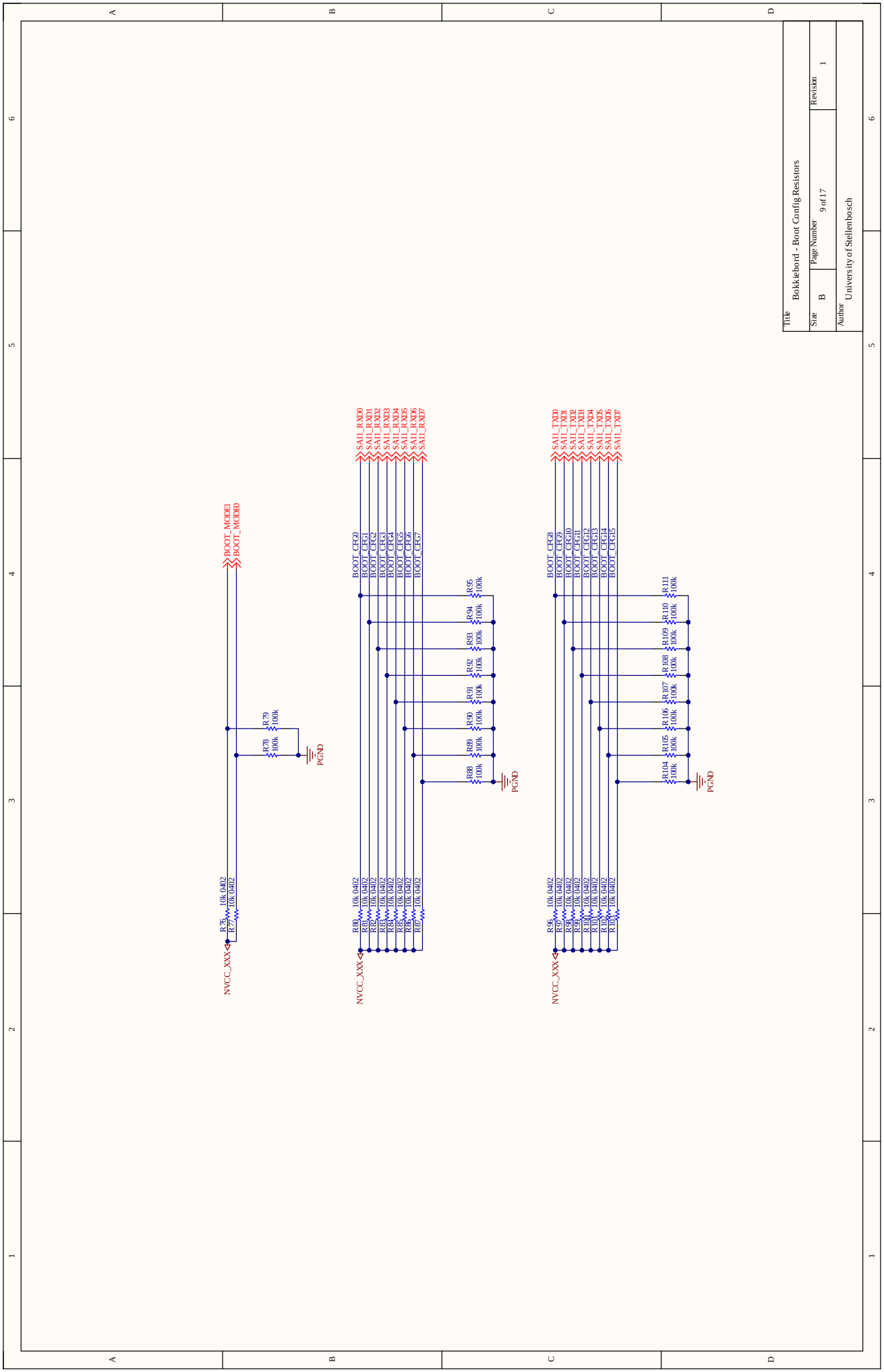


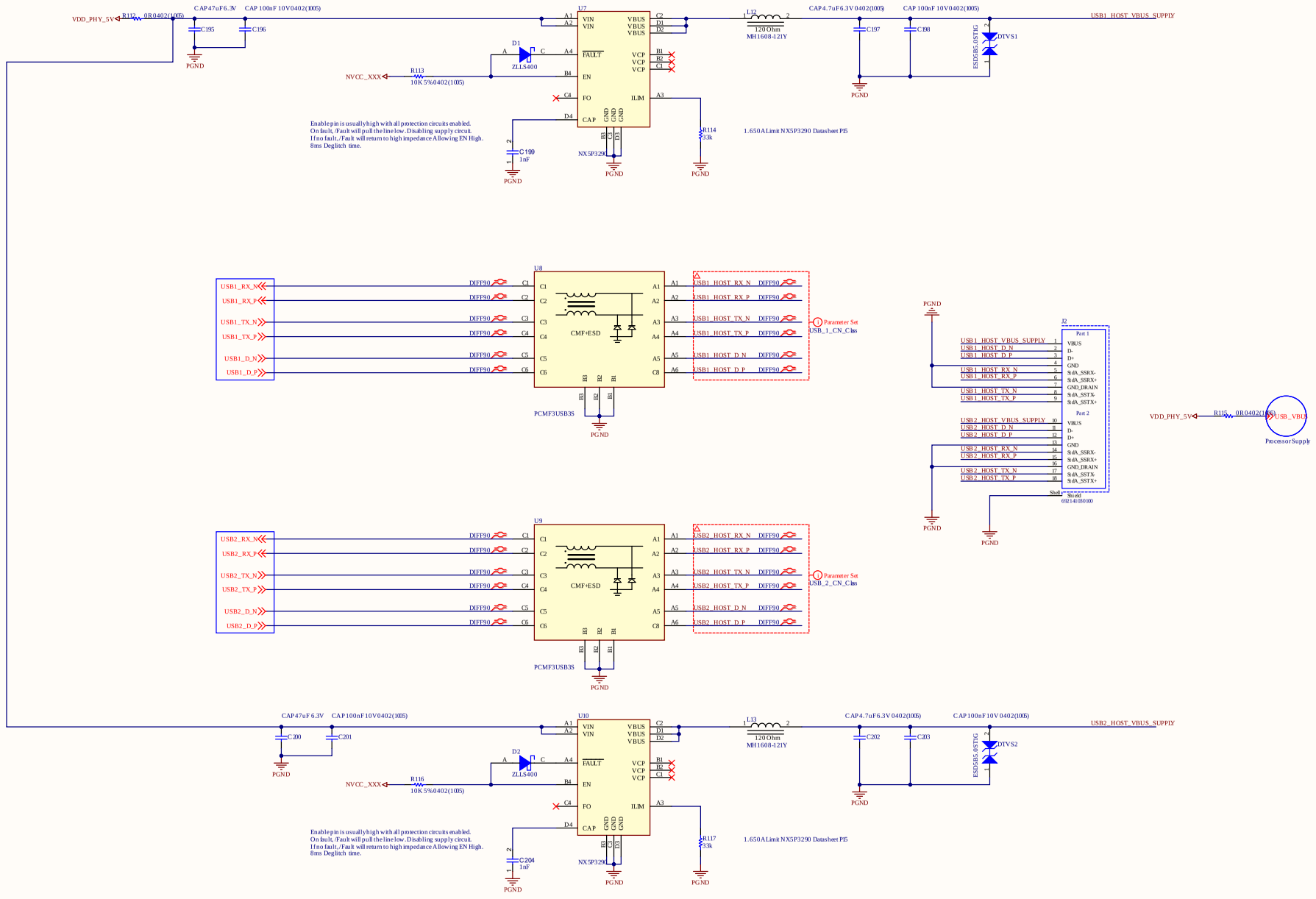




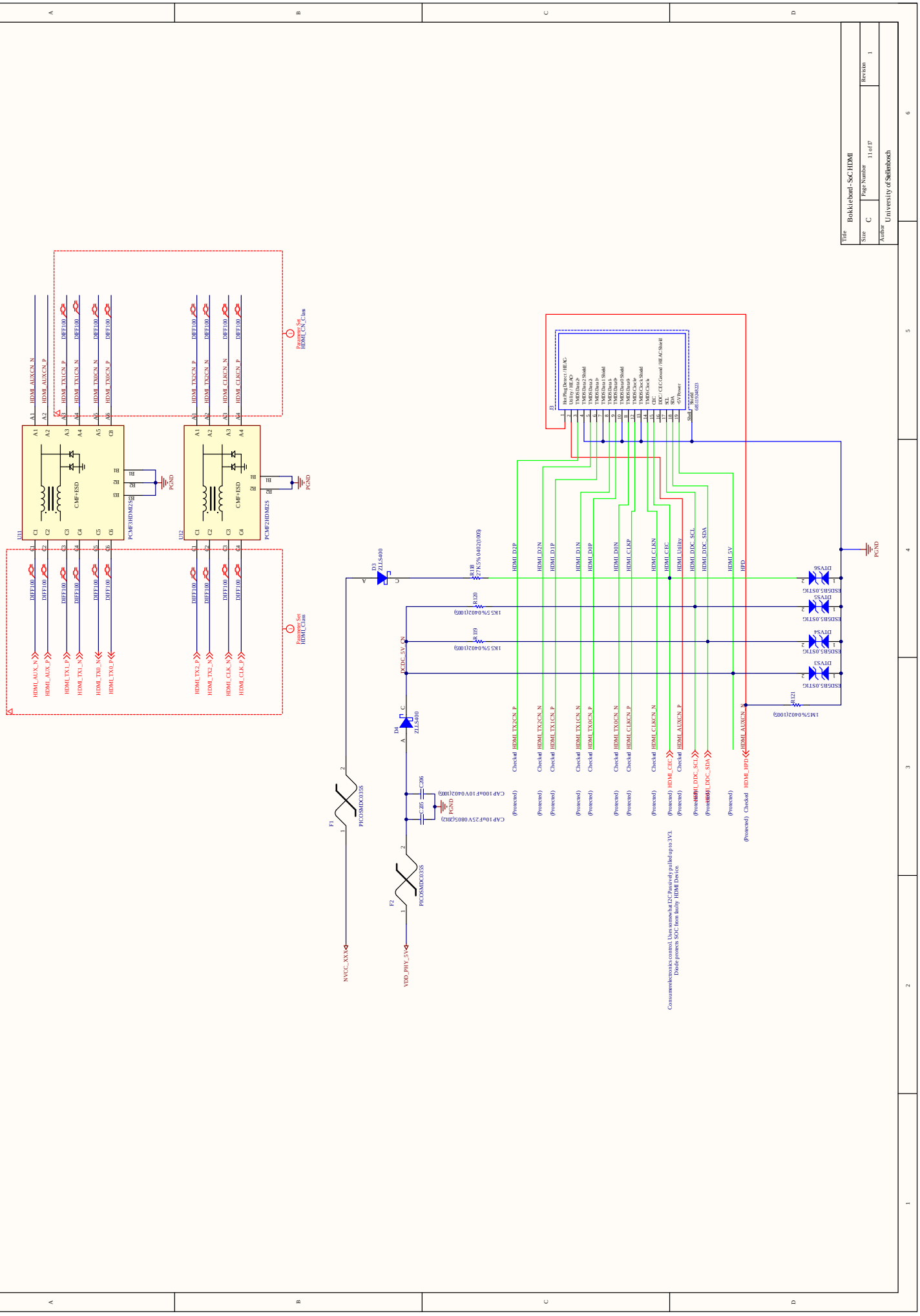


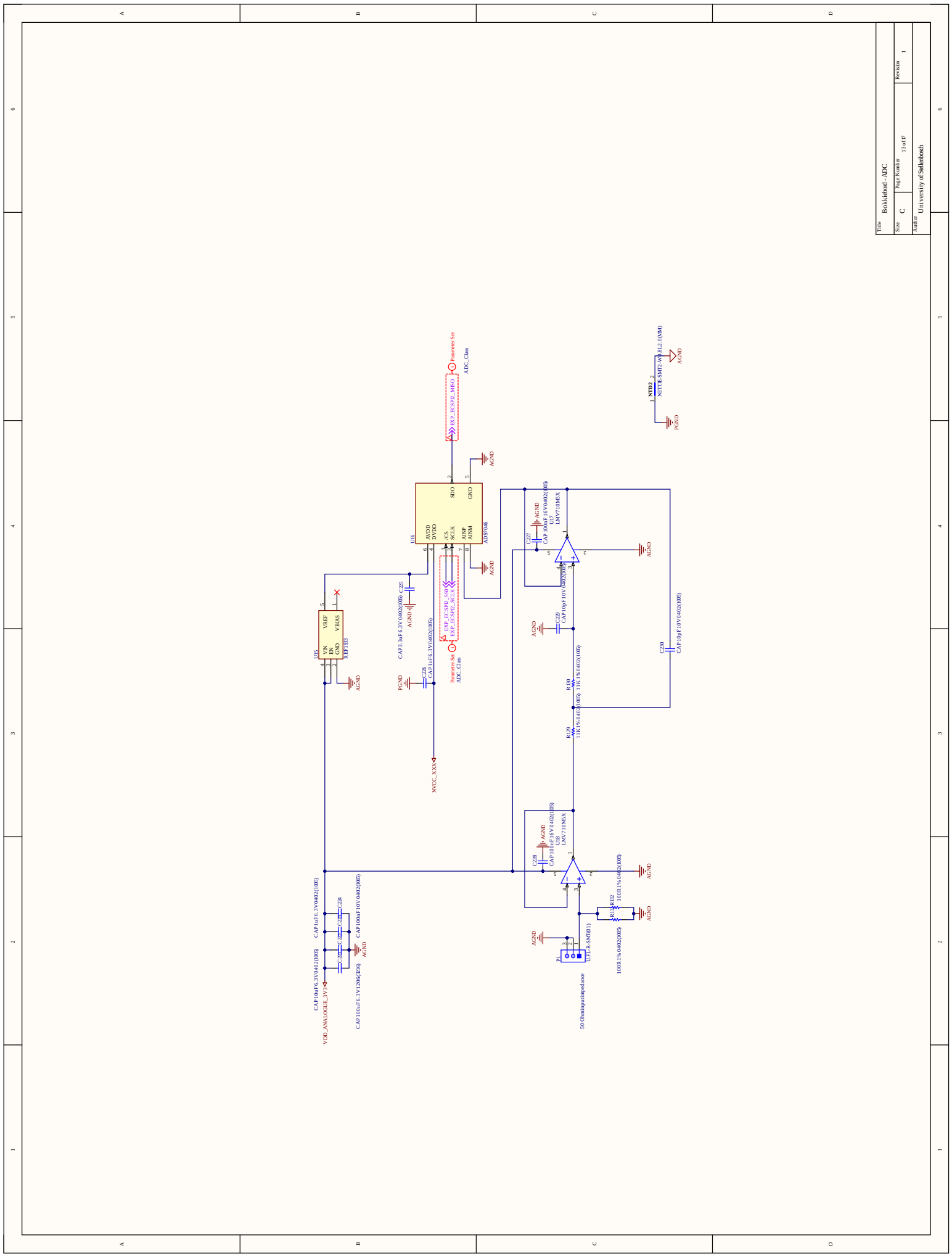


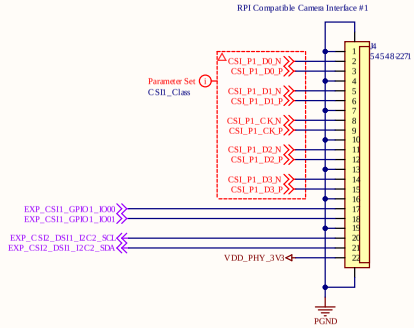


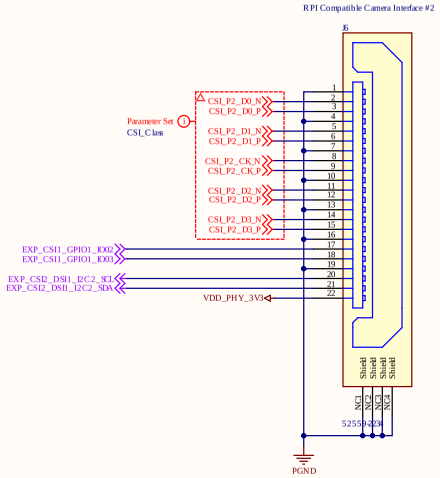


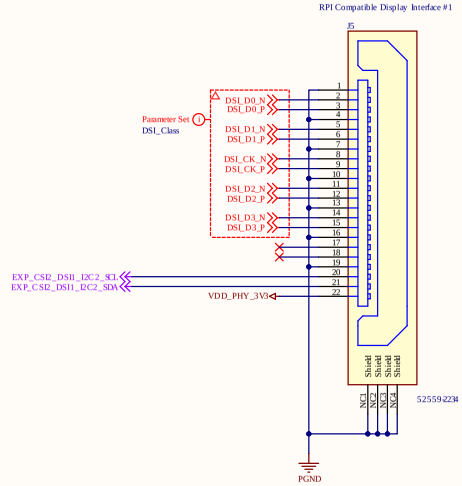
Title	Bokkieboard - SoC USB		
Size	C	Page Number 10 of 17	Revision 1
Author	University of Stellenbosch		



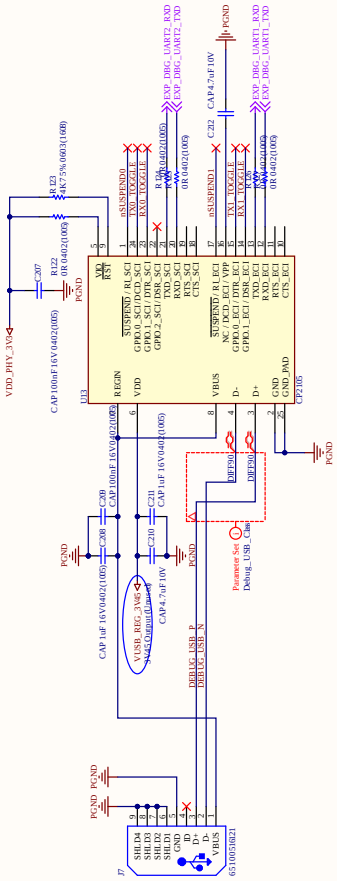




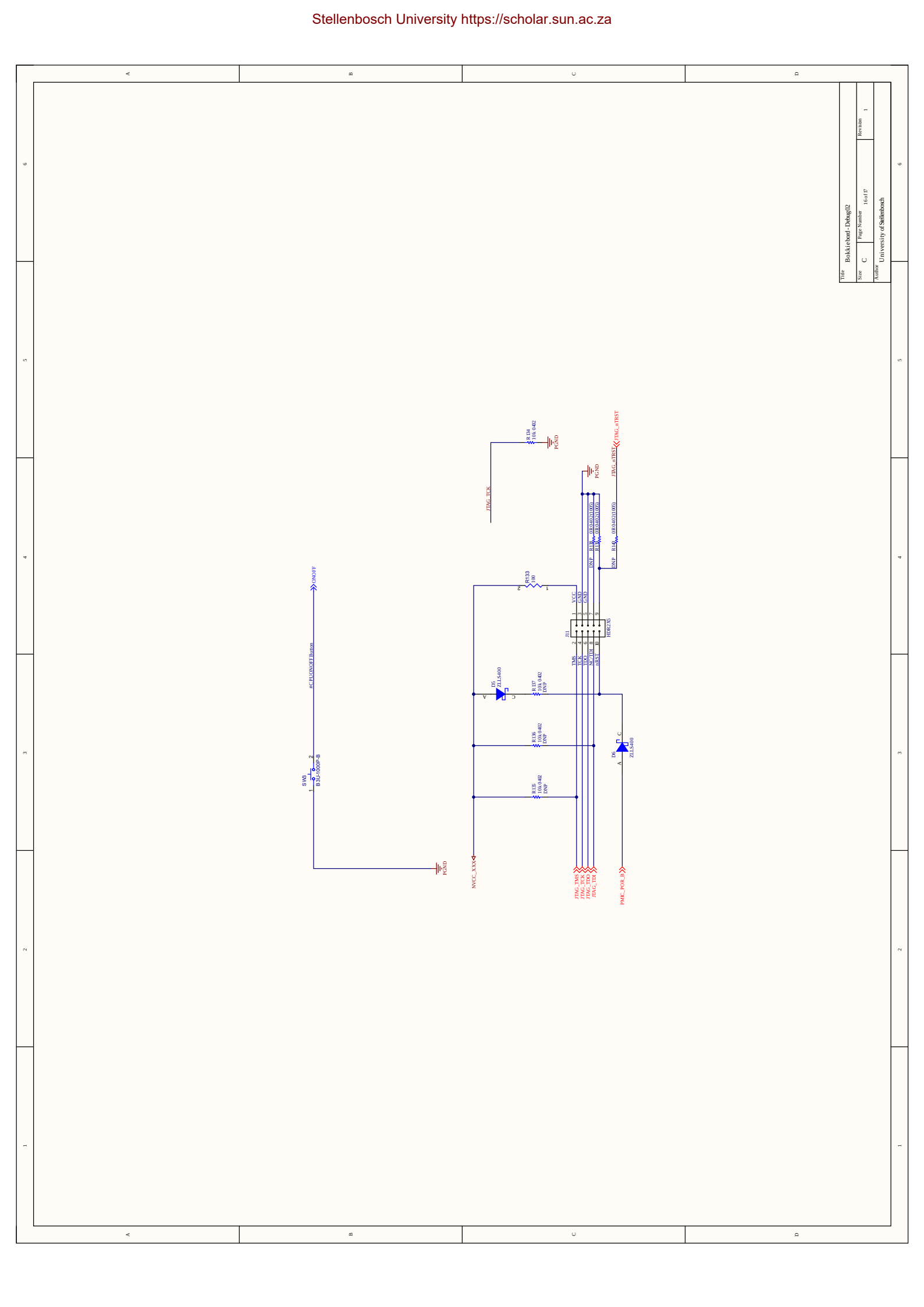


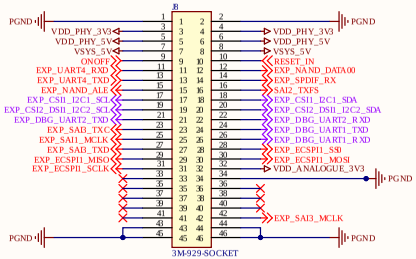


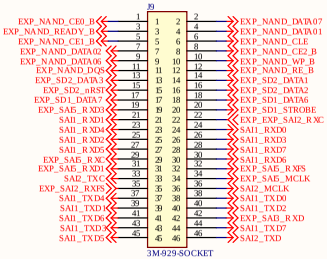
Title: Bokkiebord - MIPI Connectors		
Size: C	Page Number: 14 of 17	Revision: 1
Author: University of Stellenbosch		



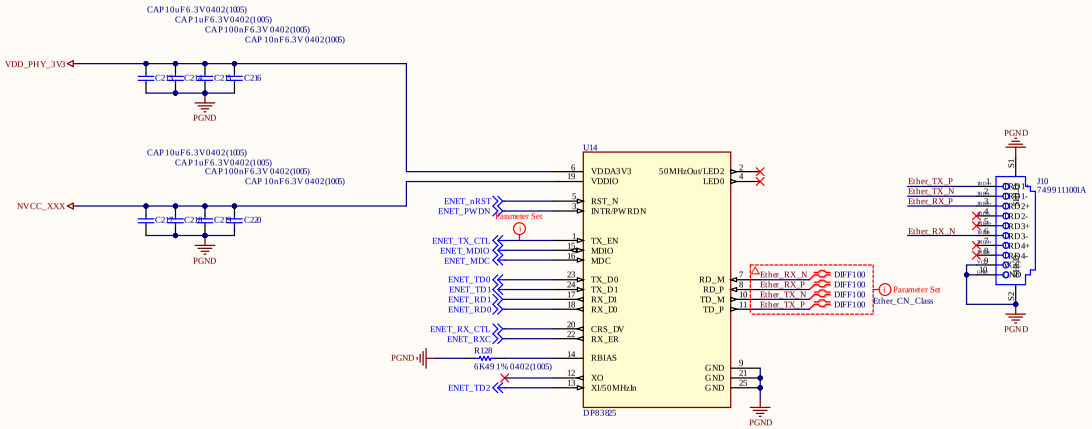
Title	Bokkiebord - Debug01
Size	C
Page Number	15 of 17
Revision	1
Author	University of Stellenbosch







Title: Bokkiebord - Expansion Connectors		
Size: C	Page Number: 17 of 17	Revision: 1
Author: University of Stellenbosch		

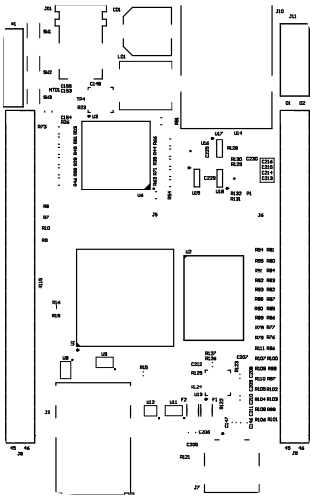


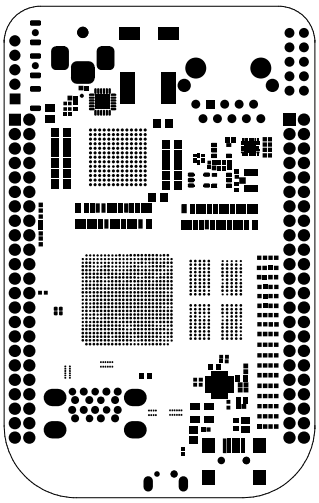
Title	Bokkiebord - RMII Ethernet Interface		
Size	C	Page Number 12 of 17	Revision 1
Author	University of Stellenbosch		

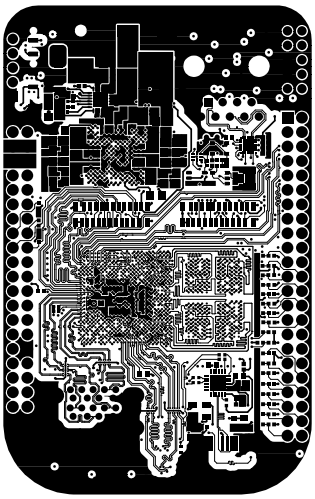
A.2 Standard Bokkiebord Gerbers

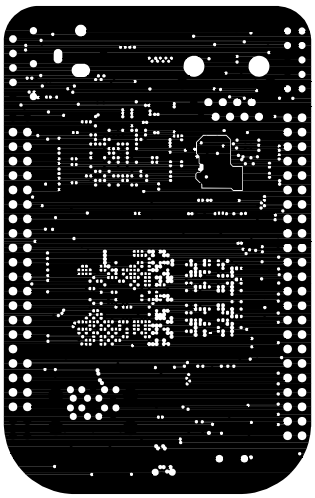
Layer order:

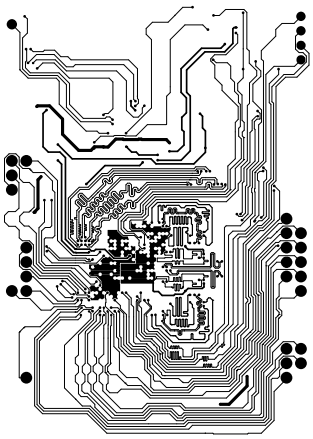
- Top Overlay
- Top Solder mask
- Top Copper Layer
- Ground Plane
- 1st Inner Layer
- 2nd Inner Layer
- Power Plane
- Bottom Copper Layer
- Bottom Solder Mask
- Bottom Overlay
- Drill Layer
- Drill Layer
- Board Outline

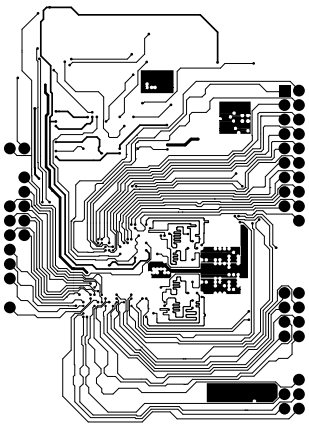


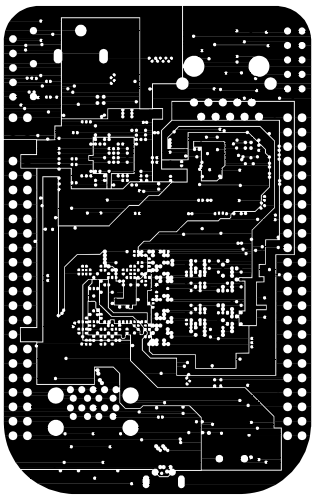


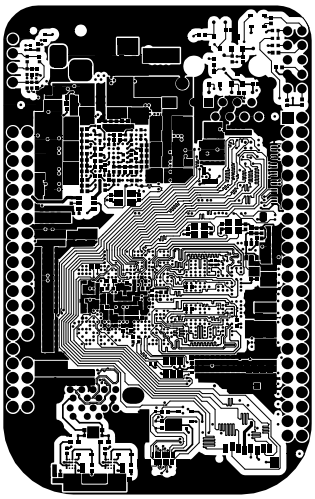


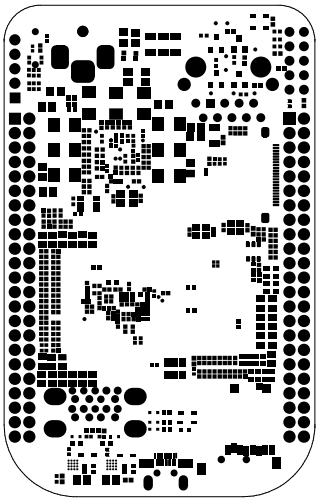


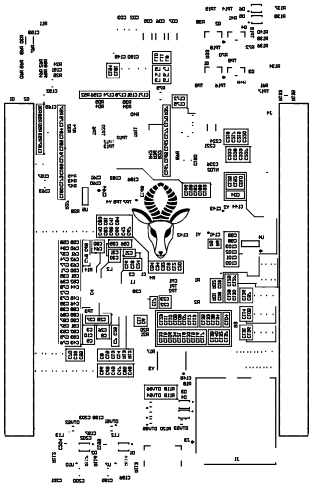


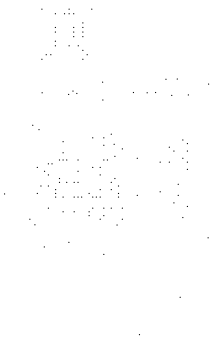


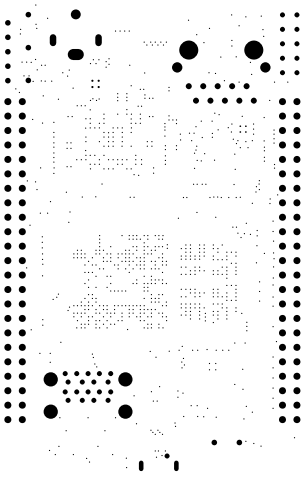


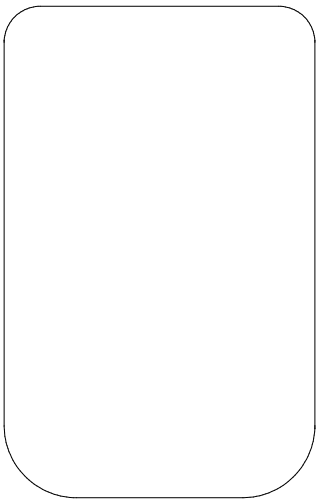












A.3 Standard Bokkiebord Bill of Materials

Designator	Quantity	Mouser part No	Size	Value	Voltage	Tolerance%
		Integrated Circuits				
U1	1	MIMX8MQ6DVAJZAB				
U2	1	MT53E512M32D2NP-046 WT:E				
U7, U10	2	NX5P3290UKZ				
U8, U9	2	PCM3USB3SZ				
U11	1	PCMF3HDMI2SZ				
U12	1	PCMF2HDMI2SZ				
U13	1	CP2105				
U14	1	DP83825IRMQT				
U16	1	ADS7046IRUGR				
		Power IC				
U3	1	TPS54A24				
U6	1	TPS6590378ZWSR				
		Loadswitch				
U5	1	TPS22918TDBVTQ1				
		Opamp				
U4	1	LMV951MK/NOPB				
U17, U18	2	LMV710M5X/NOPB				
		Voltage Reference				
U15	1	REF1933AIDDCT				
		Transistors				
Q1, Q2	2	2N7002-T1-E3				
Q3, Q4	2	DMP1045U-7				
		Diodes				
D1, D2, D3, D4, D5, D6	6	ZLLS400QTC				
DTVS1, DTVS2, DTVS3, DTVS4, DTVS5, DTVS6	6	ESD5B5.0ST1G				
		Crystals				
Y1	1	830108212509				
Y2	1	R2016-27.000-8-F-1010				
Y4	1	FA-238				
		Oscillators				
Y3	1	NX5427001Z				
		Resistors				

R1, R2, R3, RD1, RD2, R13, R23, R113, R116, RD4, RD5, R31, R36, R41, R43, R45, R47, R50, R52, R53, R57, R58, R61, R72, R76, R77, R80, R81, R82, R83, R84, R85, R86, R87, R96, R97, R98, R99, R100, R101, R102, R103, R134, R135, R136, R137	46	CRCW040210K0FKEDC	402	10k		1
R4	1	CRCW0402240RFKED	402	240R		1
R5, R6	2	CRCW0201240RFKED	201	240R		1
R7, R8, R9, R10	4	CRCW04024K70FKED	402	4.7k		1
R11, R17, R18, R22, R26, R78, R79, R88, R89, R90, R91, R92, R93, R94, R95, R104, R105, R106, R107, R108, R109, R110, R111	23	CRCW0402100KFKED	402	100k		1
R12, R20, R21, R27, R28, R32, R33, R34, R35, R37, R38, R40, R42, R48, R55, R56, R60, R62, R64, R65, R69, R70, R74, R112, R115, R122, R124, R125, R126, R127, R138, R138, R139, R140	33	CRCW04020000Z0ED	402	0R		5
R14	1	CRCW040215K0FKED	402	15k		1
R15	1	CRCW0603499RFKEA	603	499R		1
R16, R19	2	CRCW0402200RJNED	402	200R		5
R24	1	CRCW0402220KFKED	402	220k		1
R25	1	CRCW040230K0FKED	402	30k		1

R29, R39, R44, R46, R49, R51, R54, R59, R63, R66, R71, R73, R75, RB1, RD3	15	CRCW08050000Z0EA	805	0R		5
R30	1	CRCW0402220RFKED	402	220R		1
R67, R68, R119, R120	4	CRCW04021K50JNED	402	1k5		5
R114, R117	2	CRCW040233K0JNED	402	33k		5
R118	1	CRCW040227K0JNED	402	27k		5
R121	1	CRCW04021M00JNED	402	1M		5
R123	1	CRCW06034K70JNED	603	4k7		5
R128	1	CRCW04026K49FKED	402	6k49		1
R129, R130	2	CRCW040211K0FKED	402	11k		1
R131, R132, R133	3	CRCW0402100RFKED	402	100R		1
		Capacitors				
C1, C13, C14, C15, C16, C17, C18, C21, C22, C23, C24, C28, C35, C36, C37, C38, C39, C43, C44, C45, C46, C50, C51, C52, C53, C57, C58, C59, C60, C61, C62, C63, C71, C72, C73, C74, C106, C107, C108, C109, C110, C111, C112, C113, C115, C116, C117, C118, C119, C120, C121, C122, C123, C124, C208, C211, C214, C218, C223, C226	60	GRM155R61E105KA12D	402	1uF	10V	10

C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C25, C26, C27, C29, C30, C40, C41, C64, C66, C67, C79, C81, C82, C83, C95, C96, C98, C99, C100, C101, C102, C103, C125, C145, C148, C149, C150, C155, C207, C209, C158, C159, C165, C215, C219, CD4, CD8, CD9, CD10, CM1, C196, C198, C201, C203, C224, C206, C227, C228	58	GRM155R61H104JE14D	402	100nF	10V	10
C12, C19, C20, C32, C33, C34, C47, C48, C49, C54, C55, C56, C69, C70, C85, C97, C104, C105, C114, C146, C126, C131, C136	23	GRM219R61A226MEA0D	805	22uF	6.3V	20
C31, C42, C65, C68, C80, C84, C160, C161, C166, C167, C173, C176, C177, C178, C179, C182, C183, C184, C185, C186, C187, C188	22	GRM155R61C225KE44D	402	2.2uF	6.3V	10
C75, C76, C216, C220	4	GRM155R61A103KA01D	402	10nF	10V	10

C77, C78, C86, C87, C88, C89, C90, C91, C92, C93, C94	11	GRM155R61A472KA01D	402	4.7nF	10V	10
C127, C128, C129, C130, C132, C133, C134, C135, C137, C138, C139, C140, C210, C212	14	GRM185R61C475ME11D	603	4.7uF	10V	20
C141, C142, C143, C144	4	490-GJM1555C1H160FB01JCT-ND	402	16pF	50V	1
C147	1	GRM188R72A104MA35D	603	100nF	6.3V	20
C151	1	GRM1555C1H101JA01D	402	100pF	16V	10
C152, C213, C217, C222	4	GRM155R60J106ME15J	402	10uF	6.3V	20
C153	1	GRM1555C1E220GA01D	402	22pF	16V	10
C154	1	GRM155R71H272KA01J	402	2.7nF	16V	10
C156, C162, C170, C171, C174, C180, C189, C191, C193, C197, C202	11	GRM155R60J475ME47D	402	4.7uF	6.3V	10
C157, C163, C169, C172, C175, C181, C190, C192, C194, C195, C200	11	GRM219R60J476ME44D	805	47uF	6.3V	20
C164, C168, C229, C230	4	GRM1555C1H100FA01D	402	10pF	50V	5
C199, C204	2	GRM155R61E102KA01D	402	1nF	50V	20
C205, CD2, CD3	3	GRM21BR61C106KE15K	805	10uF	25V	10
C221, CD5, CD6, CD7	4	GRM31CR61A107ME05L	1206	100uF	6.3V	20
C225	1	C1005X5R1A335K050BC	402	3.3uF	6.3V	10
CD1	1	UCL1V101MNL1GS		100uF	35V	20
		Inductors				
L3, L4, L5, L6, L7, L8, L9, L10, L11	9	IHLP1616ABER1R0M11				
LD1	1	74437358010				
		Choke				
L1	1	KMZ1608DHR241CTD25				

L2, L12, L13	3	KMZ1608SHR121ATD25				
		Fuses				
F1, F2	2	PICOSMDC035S-2				
		Switches				
SW1, SW2, SW3	3	B3U-1000P-B				
		Connectors				
*1	1					
J1	1	"473521001"				
J2	1	692141030100				
J3	1	"685119248223"				
J4	1	54548-2271				
J5, J6	2	52559-2234				
J7	1	65100516121				
J8,J9	2	SSW-123-01-T-D				
J10	1	7499111001A				
J11	1					
JD1	1	PJ1-021				
P1	1	U.FL-R-SMT(01)				

A.4 Standard Bokkiebord Simulations

Simulation order:

- VIN
- VSYS_5V
- NVCC_DRAM
- VDD_0V9
- VDD_SOC
- NVCC_XXX
- VDD_0V9
- VDD_PHY_3V3
- VDDA_1P8
- NVCC_SNVS
- VDD_SNVS
- VDD_ANALOGUE
- VDD_PHY_5V

A.4.0.1 VIN Power Plane

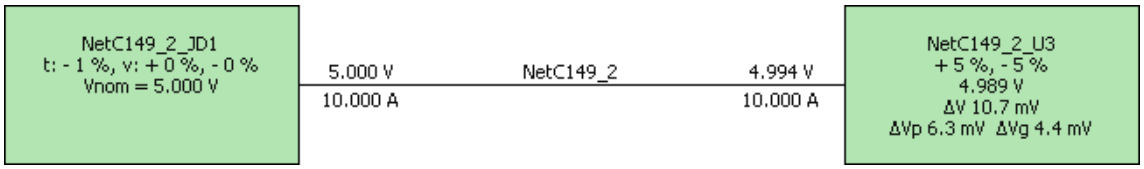


Figure A.1: VIN Power tree from Power Jack to 1st Stage regulator

Left: VIN Linear Voltage Heat Map from Power Jack to 1st stage regulator. 100% red is 4.99980 Volts and 100% blue is 4.99366 Volts. Right: Ground Linear Voltage Heat Map from USB Port to PMIC (Return). 100% red is 0.00435 Volts and 100% blue is 0.00009 volts.

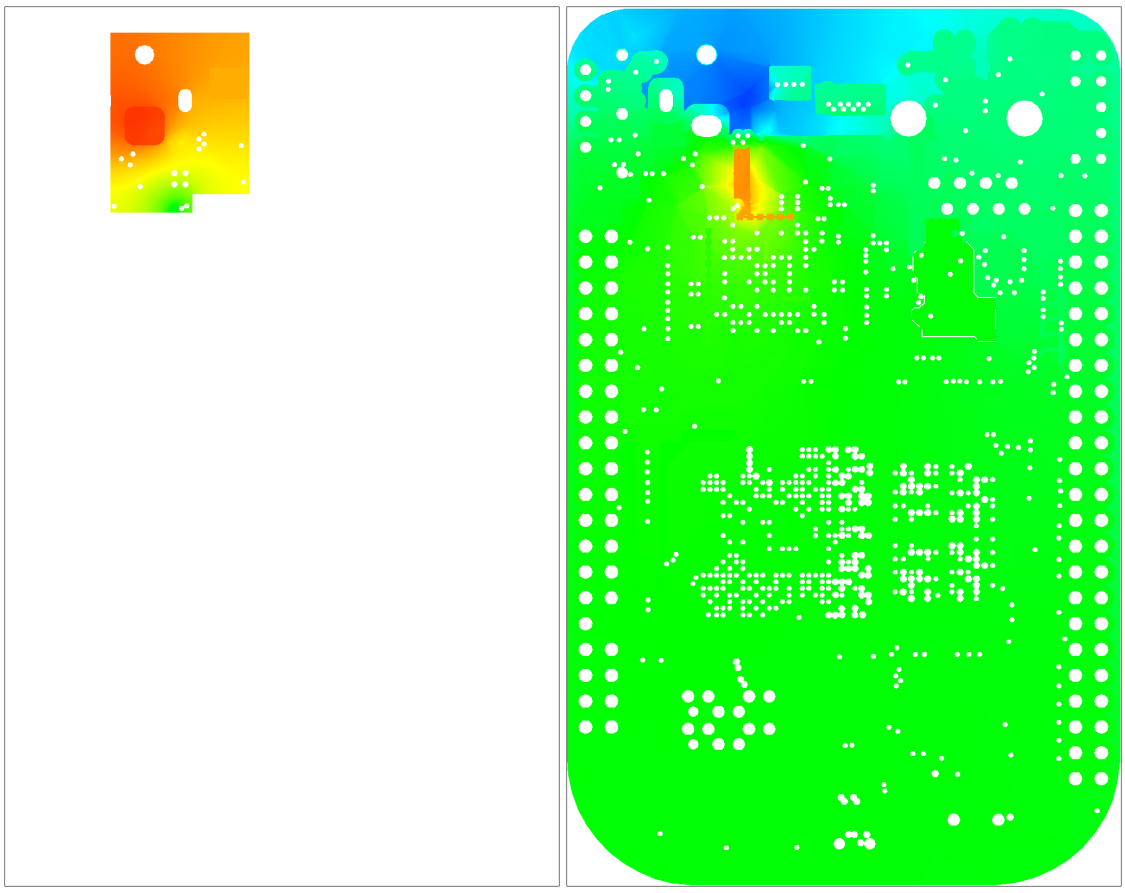


Figure A.2: VIN Linear Voltage Heat Map

A.4.0.2 VSYS_5V Power Plane

Figure A.3: VSYS Power tree from 1st stage regulator to PMIC

Left: VSYS=_5V Linear Voltage Heat Map from 1st stage regulator to PMIC. 100% red is 4.99988 Volts and 100% blue is 4.97037 Volts. Right: Ground Linear Voltage Heat Map from USB Port to PMIC (Return). 100% red is 0.00382 Volts and 100% blue is 0.00000 volts.

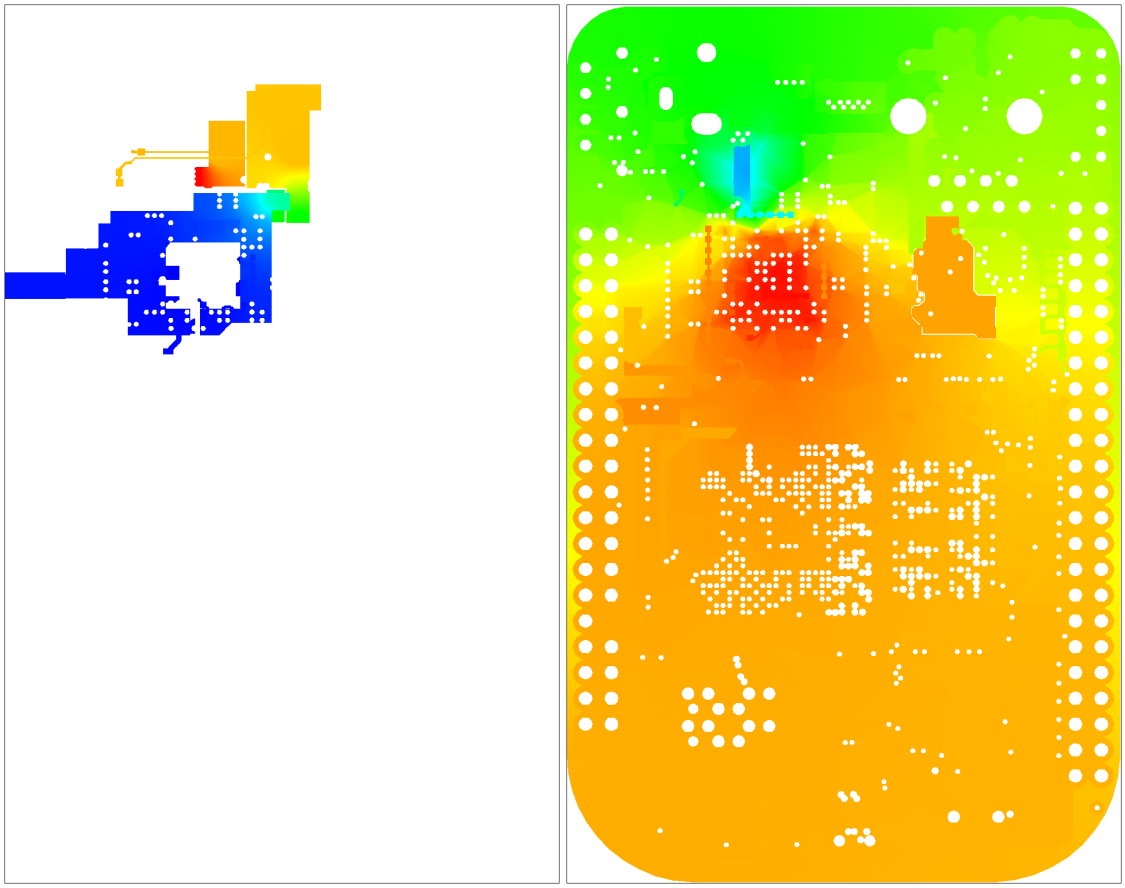


Figure A.4: VSYS_5V Linear Voltage Heat Map

A.4.0.3 NVCC_DRAM Power Plane

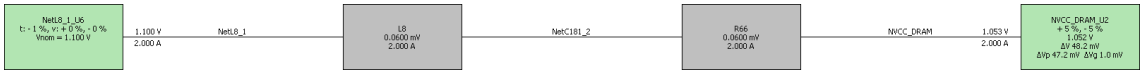


Figure A.5: NVCC_DRAM Power tree from PMIC to Memory

Left: NVCC_DRAM Linear Voltage Heat Map from PMIC to Memory. 100% red is 1.09998 Volts and 100% blue is 1.05277 Volts. Right: Ground Linear Voltage Heat Map from Memory to PMIC (Return). 100% red is 0.00102 Volts and 100% blue is 0.00000 volts.

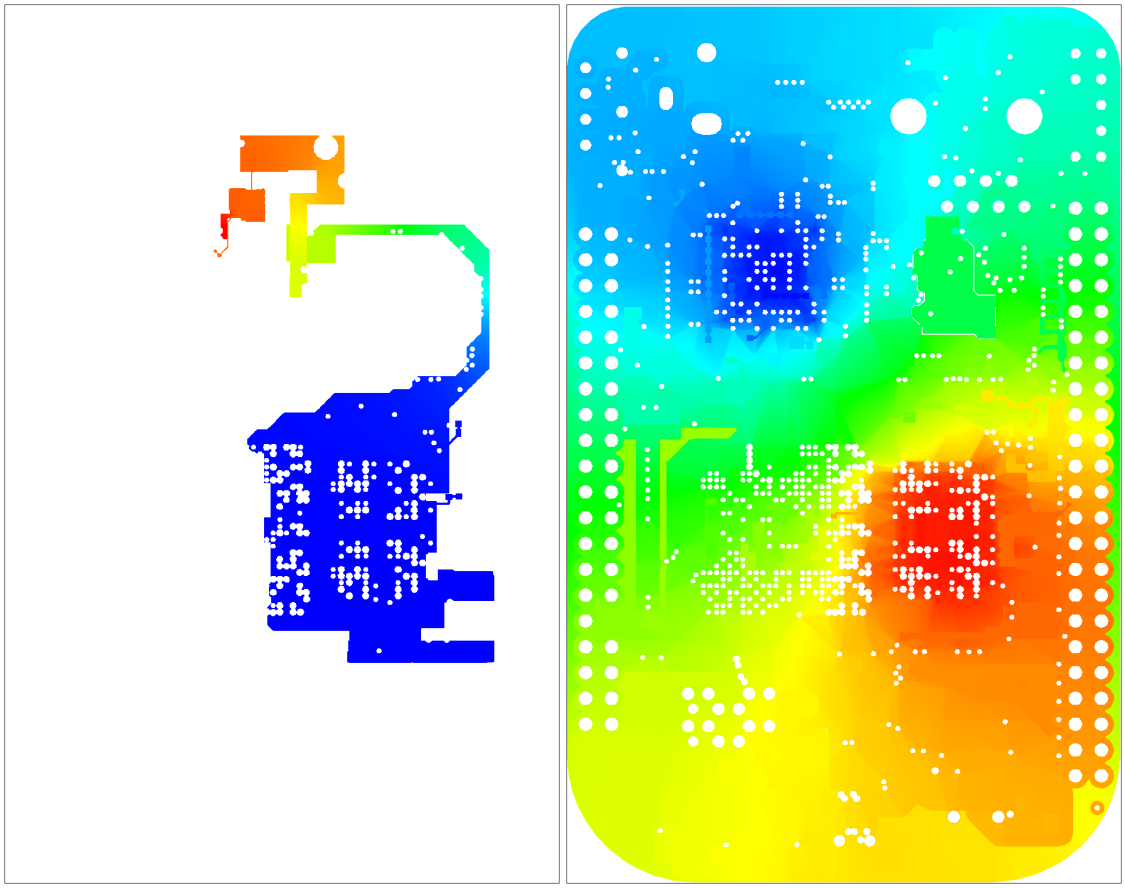


Figure A.6: NVCC_DRAM Linear Voltage Heat Map from PMIC to Memory

A.4.0.4 VDD_0V9 Power Plane

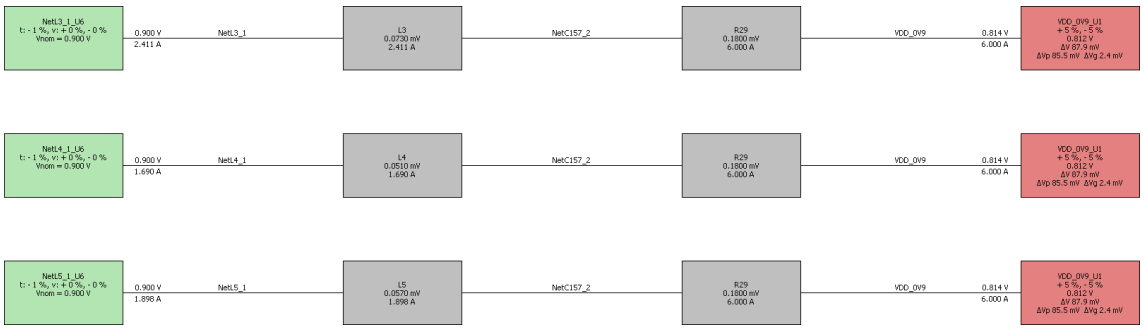


Figure A.7: VDD_0V9 Power tree from PMIC to Processor

Left: VDD_0V9 Linear Voltage Heat Map from PMIC to Processor. 100% red is 0.89998 Volts and 100% blue is 0.81446 Volts. Right: Ground Linear Voltage Heat Map from Processor to PMIC (Return). 100% red is 0.00239 Volts and 100% blue is 0.00000 volts.

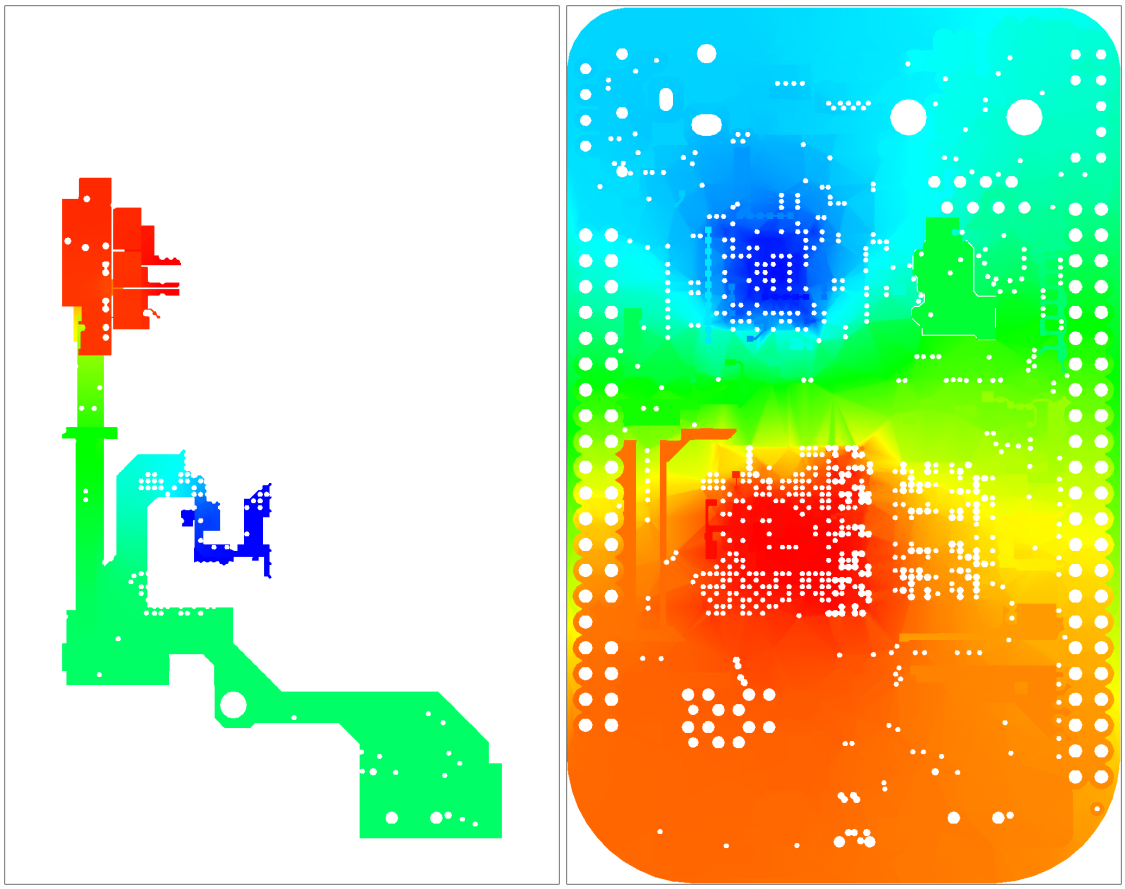


Figure A.8: VDD_0V9 Linear Voltage Heat Map from PMIC to Processor

A.4.0.5 VDD_SOC Power Plane

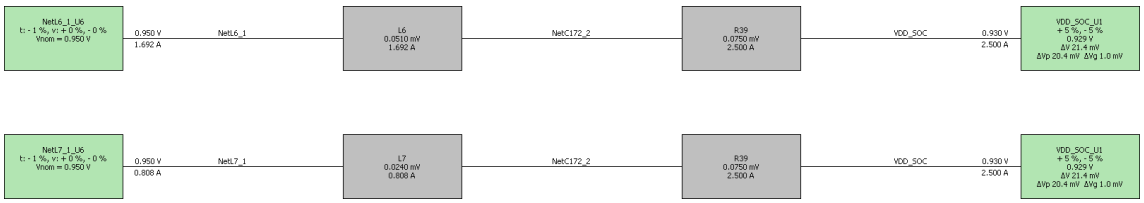


Figure A.9: VDD_SOC Power tree from PMIC to Processor

Left: VDD_SOC Linear Voltage Heat Map from PMIC to Processor. 100% red is 0.94999 Volts and 100% blue is 0.92958 Volts. Right: Ground Linear Voltage Heat Map from Processor to PMIC (Return). 100% red is 0.00100 Volts and 100% blue is 0.00000 volts.

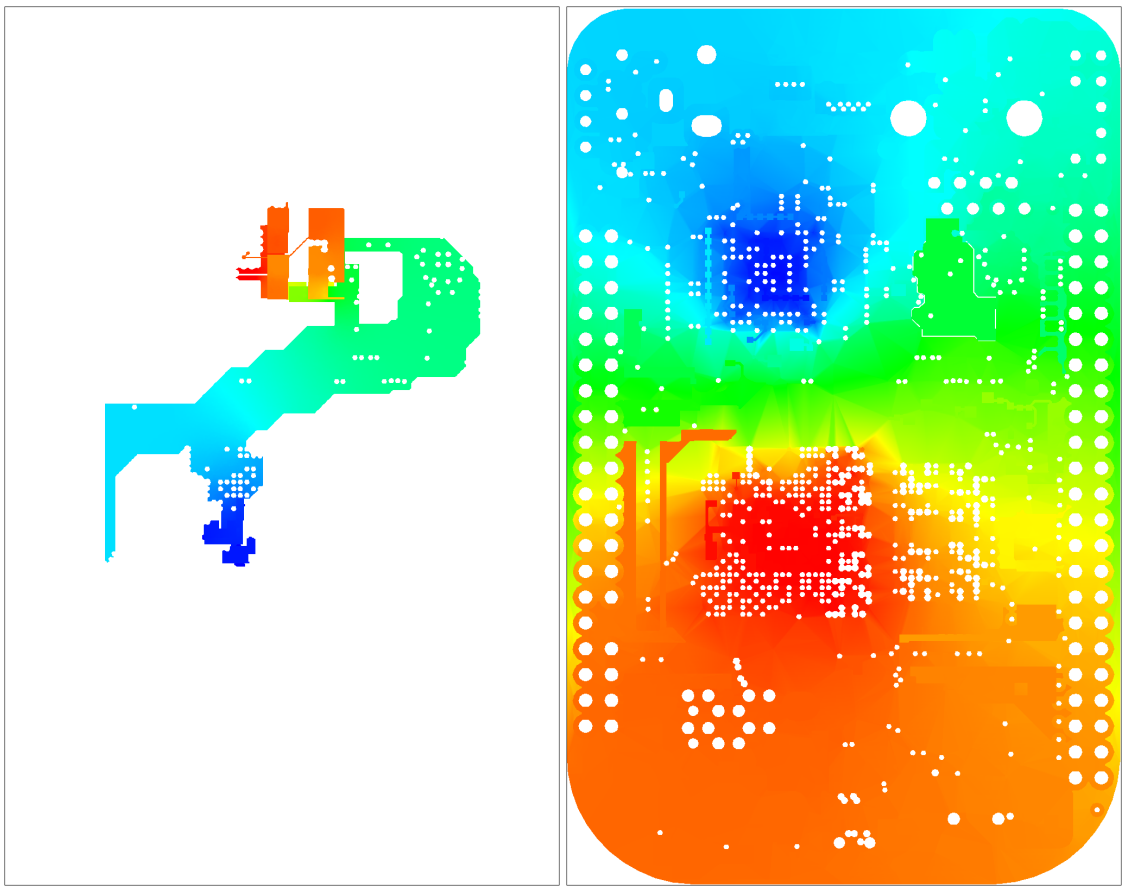


Figure A.10: VDD_SOC Linear Voltage Heat Map from PMIC to Processor

A.4.0.6 NVCC_XXX Power Plane

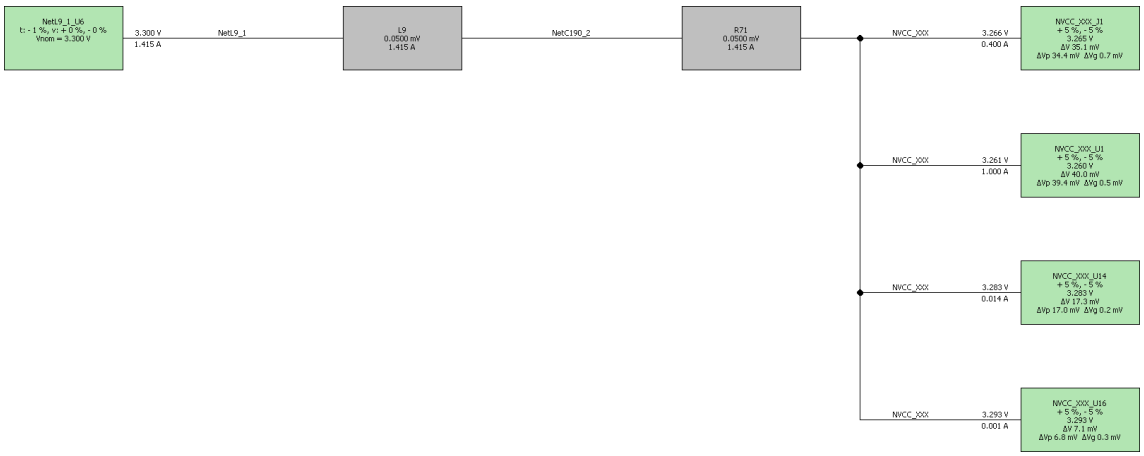


Figure A.11: NVCC_XXX Power tree from PMIC to Processor

Left: NVCC_XXX Linear Voltage Heat Map from PMIC to ADC, Ethernet PHY, SD Card and Processor. 100% red is 3.29999 Volts and 100% blue is 3.26055 Volts. Right: Ground Linear Voltage Heat Map from ADC, Ethernet PHY, SD Card and Processor to PMIC (Return). 100% red is 0.00069 Volts and 100% blue is 0.00000 volts.

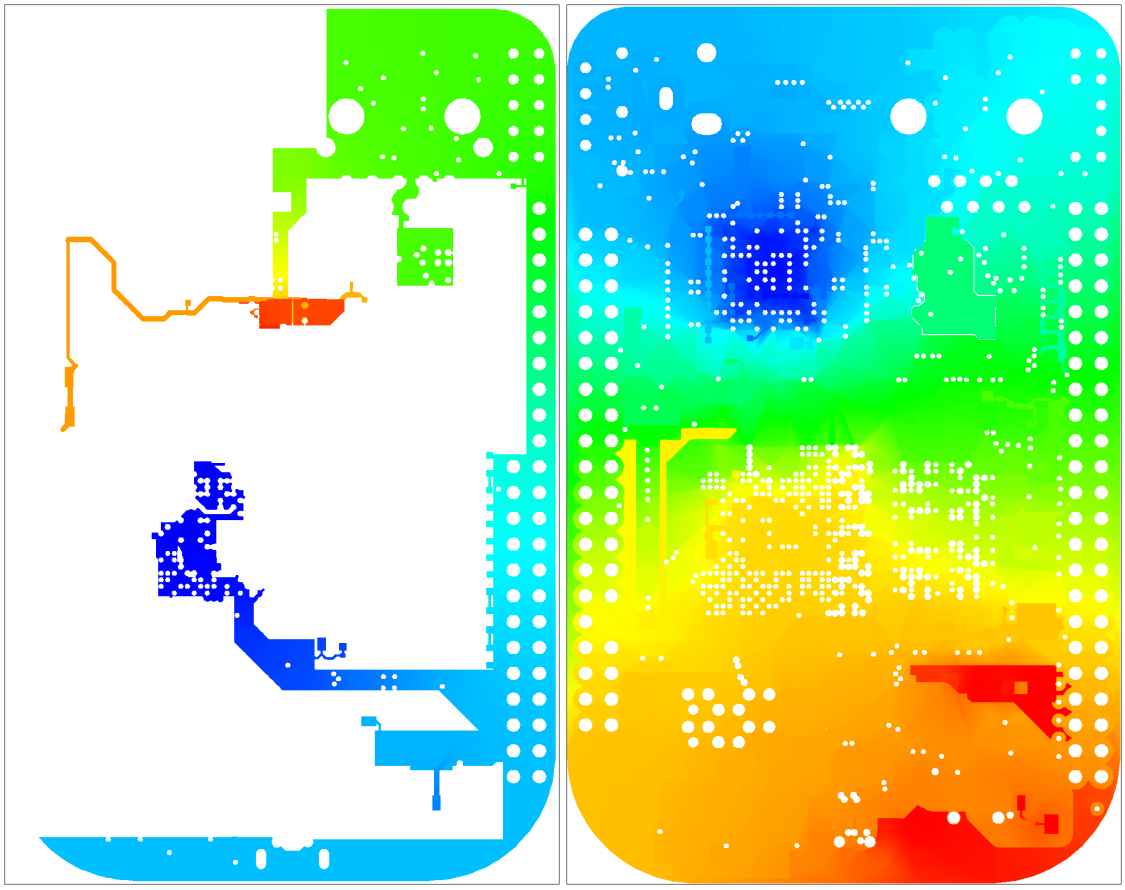


Figure A.12: NVCC_XXX Linear Voltage Heat Map from PMIC to ADC, Ethernet PHY, SD Card and Processor

A.4.0.7 VDD_PHY_0V9 Power Plane

Figure A.13: VDD_PHY_0V9 Power tree from PMIC to Processor

Left: VDD_PHY_0V9 Linear Voltage Heat Map from PMIC Processor. 100% red is 0.90000 Volts and 100% blue is 0.89369 Volts. Right: Ground Linear Voltage Heat Map from Processor to PMIC (Return). 100% red is 0.00005 Volts and 100% blue is 0.00000 volts.

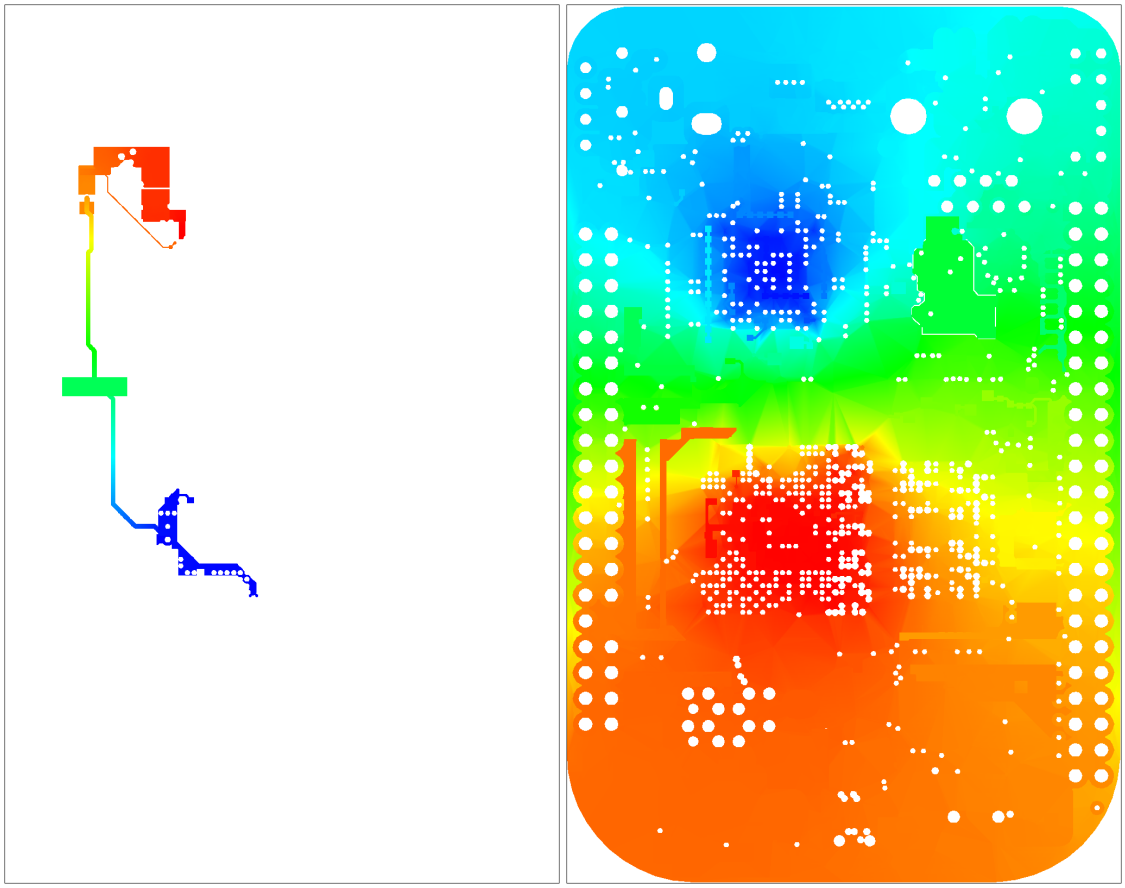


Figure A.14: VDD_PHY_0V9 Linear Voltage Heat Map from PMIC Processor

A.4.0.8 VDD_PHY_3V3 Power Plane

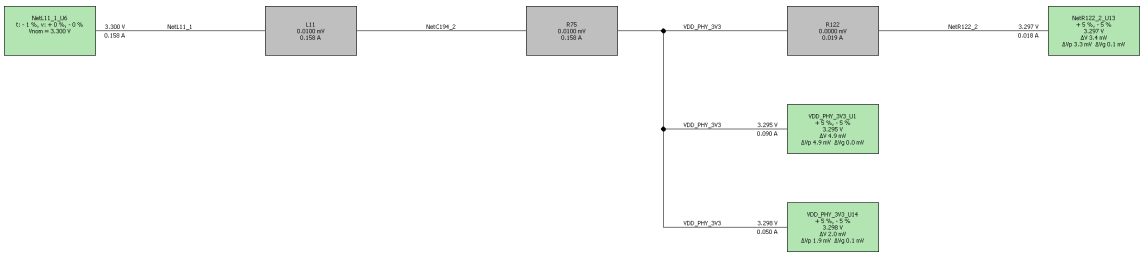


Figure A.15: VDD_PHY_3V3 Power tree from PMIC to Processor

Left: VDD_PHY_3V3 Linear Voltage Heat Map from PMIC to Ethernet PHY, UART USB Bridge and Processor. 100% red is 3.30000 Volts and 100% blue is 3.29513 Volts. Right: Ground Linear Voltage Heat Map from Ethernet PHY, UART USB Bridge and Processor to PMIC (Return). 100% red is 0.00006 Volts and 100% blue is 0.00000 volts.

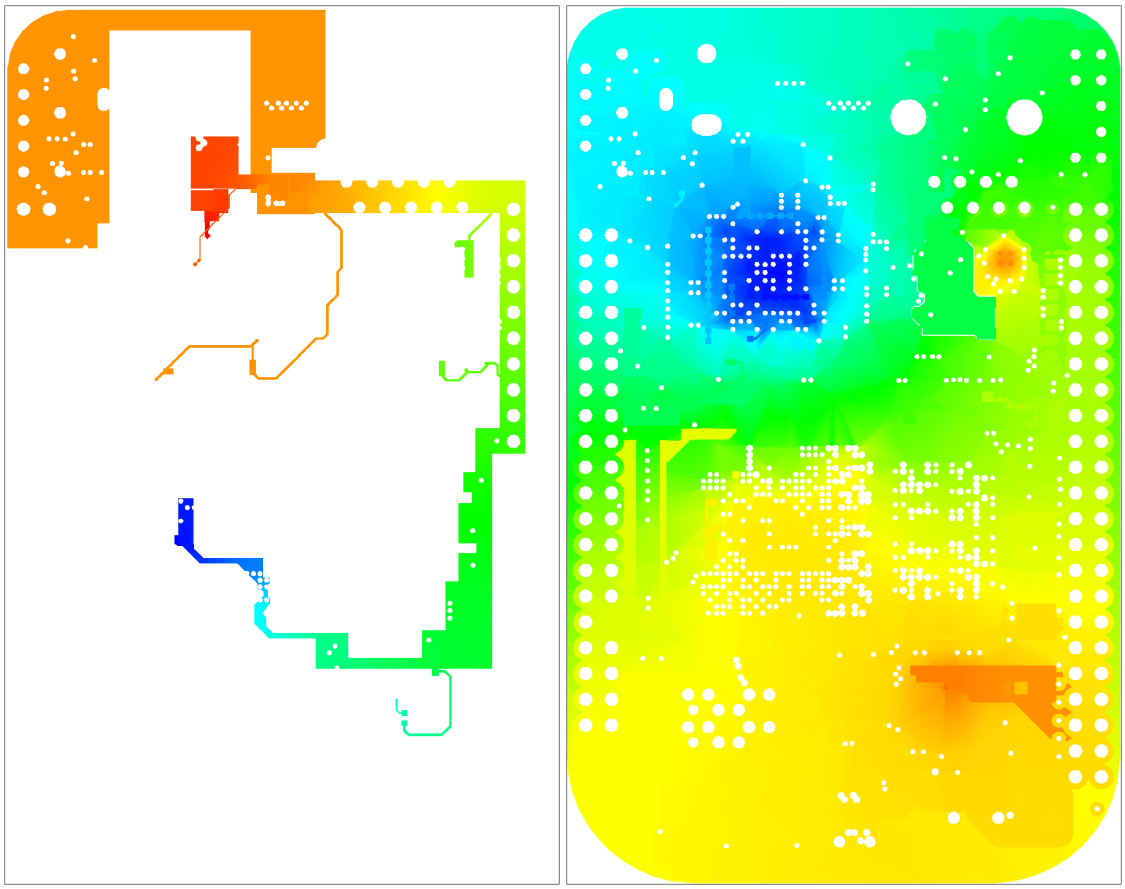


Figure A.16: VDD_PHY_3V3 Linear Voltage Heat Map from PMIC to Ethernet PHY, UART USB Bridge and Processor

A.4.0.9 VDDA_1P8 Power Plane

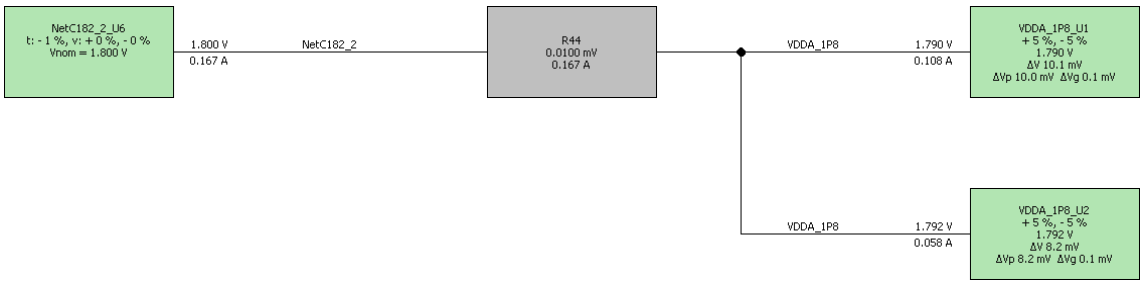


Figure A.17: VDDA_1P8 Power tree from PMIC to Processor

Left: VDDA_1P8 Linear Voltage Heat Map from PMIC to Memory and Processor. 100% red is 1.80000 Volts and 100% blue is 1.79001 Volts. Right: Ground Linear Voltage Heat Map from Memory and Processor to PMIC (Return). 100% red is 0.00006 Volts and 100% blue is 0.00000 volts.

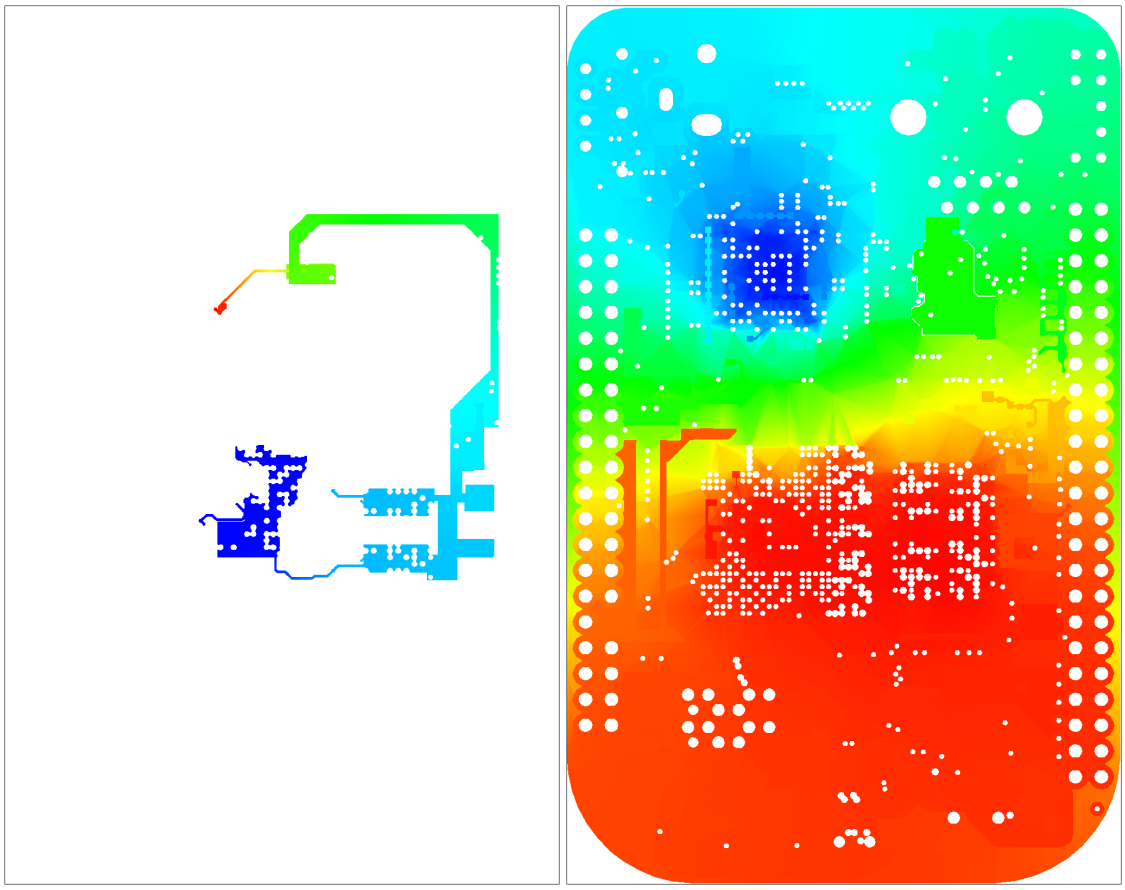


Figure A.18: VDDA_1P8 Linear Voltage Heat Map from PMIC to Memory and Processor

A.4.0.10 NVCC_SNVS Power Plane

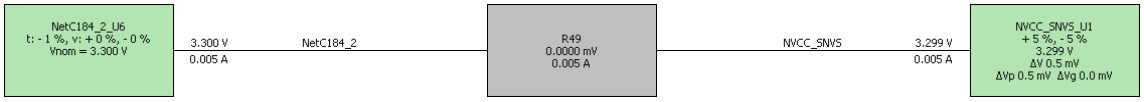


Figure A.19: NVCC_SNVS Power tree from PMIC to Processor

Left: NVCC_SNVS Linear Voltage Heat Map from PMIC to Processor. 100% red is 3.30000 Volts and 100% blue is 3.29948 Volts. Right: Ground Linear Voltage Heat Map from Processor to PMIC (Return). 100% red is 0.000001 Volts and 100% blue is 0.00000 volts.

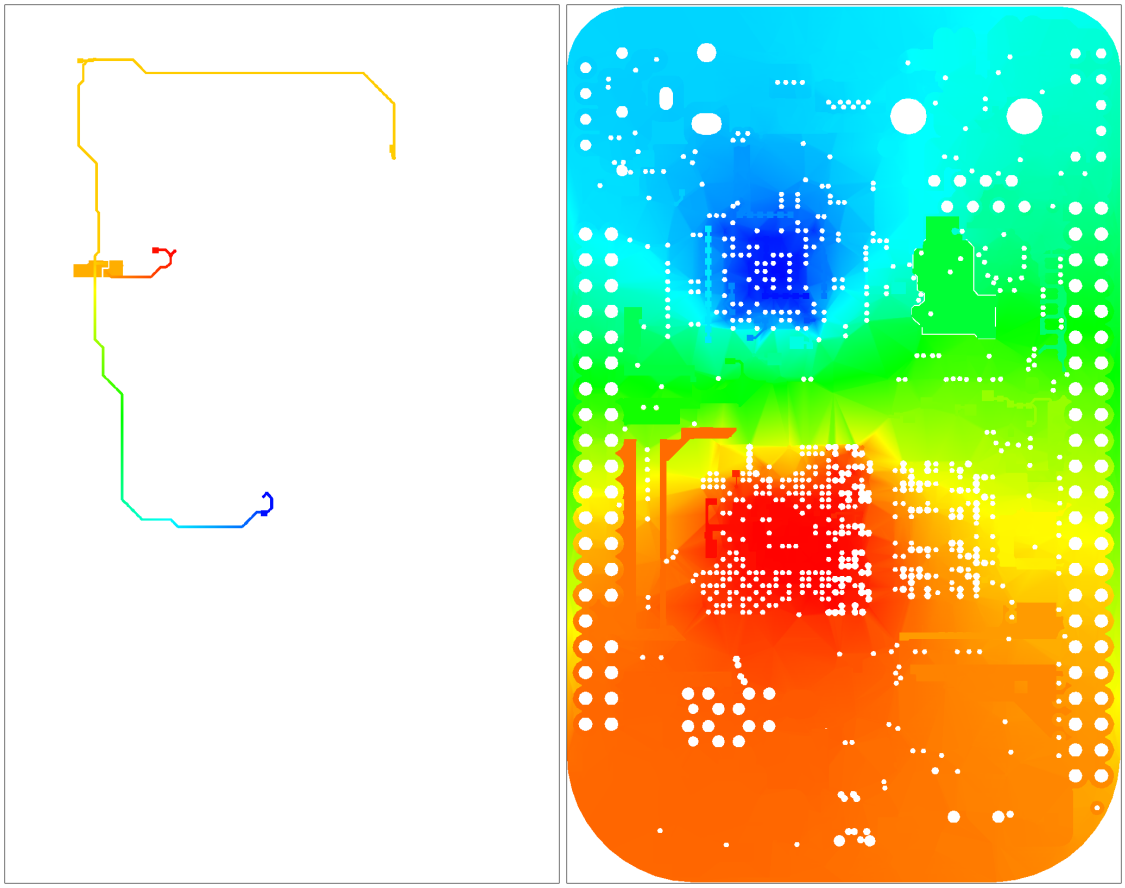


Figure A.20: Left: NVCC_SNVS Linear Voltage Heat Map from PMIC to Processor.

A.4.0.11 VDD_SNVS Power Plane

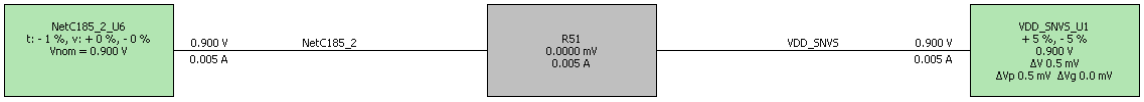


Figure A.21: VDD_SNVS Power tree from PMIC to Processor

Left: VDD_SNVS Linear Voltage Heat Map from PMIC to Processor. 100% red is 0.90000 Volts and 100% blue is 0.89954 Volts. Right: Ground Linear Voltage Heat Map from Processor to PMIC (Return). 100% red is 0.000001 Volts and 100% blue is 0.00000 volts.

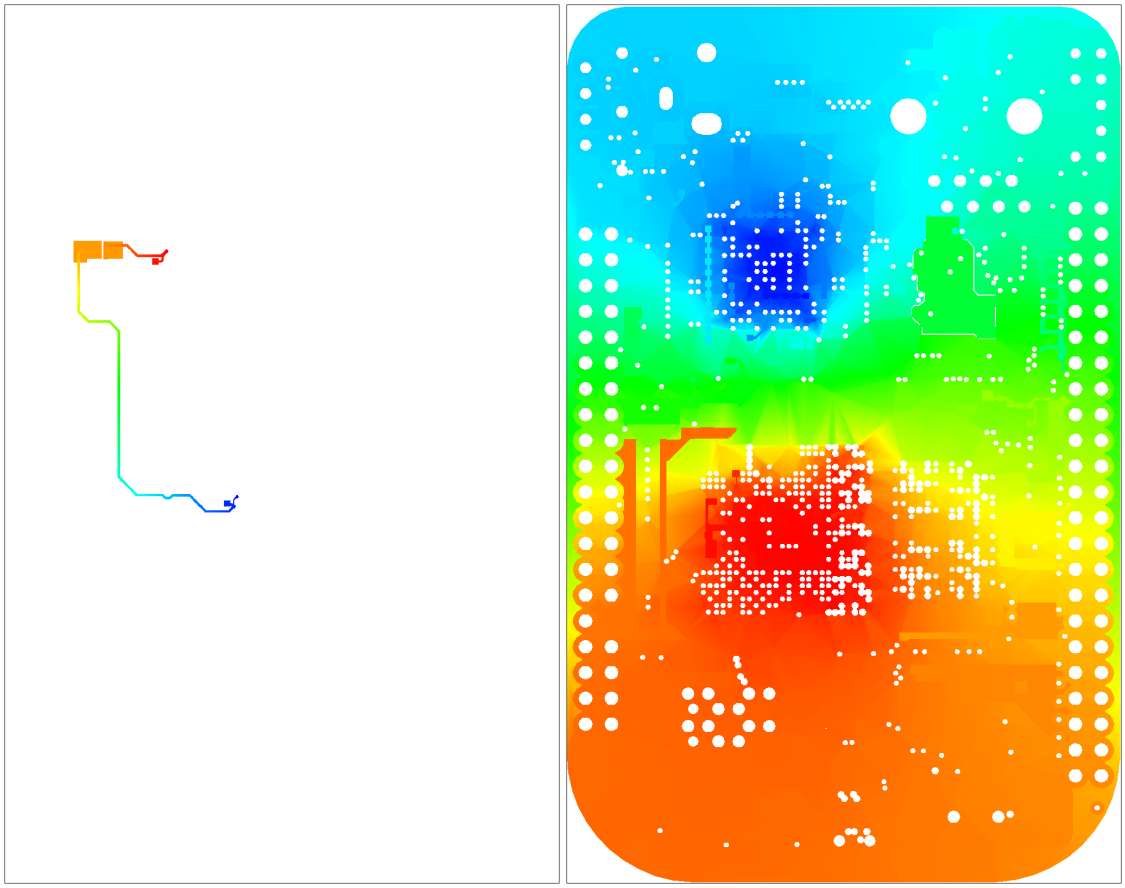


Figure A.22: Left: VDD_SNVS Linear Voltage Heat Map from PMIC to Processor.

A.4.0.12 VDD_ANALOGUE_3V3 Power Plane

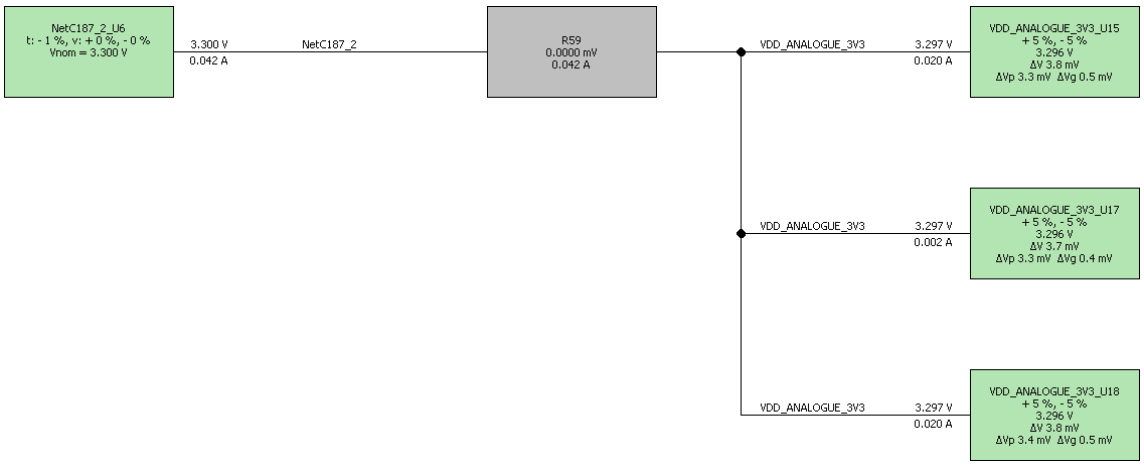


Figure A.23: VDD_ANALOGUE_3V3 Power tree from PMIC to Processor

Left: VDD_ANALOGUE_3V3 Linear Voltage Heat Map from PMIC to Voltage Reference and Op-amps. 100% red is 3.30000 Volts and 100% blue is 3.29664 Volts. Right: Ground Linear Voltage Heat Map from Processor to PMIC (Return). 100% red is 0.00049 Volts and 100% blue is 0.00000 volts.

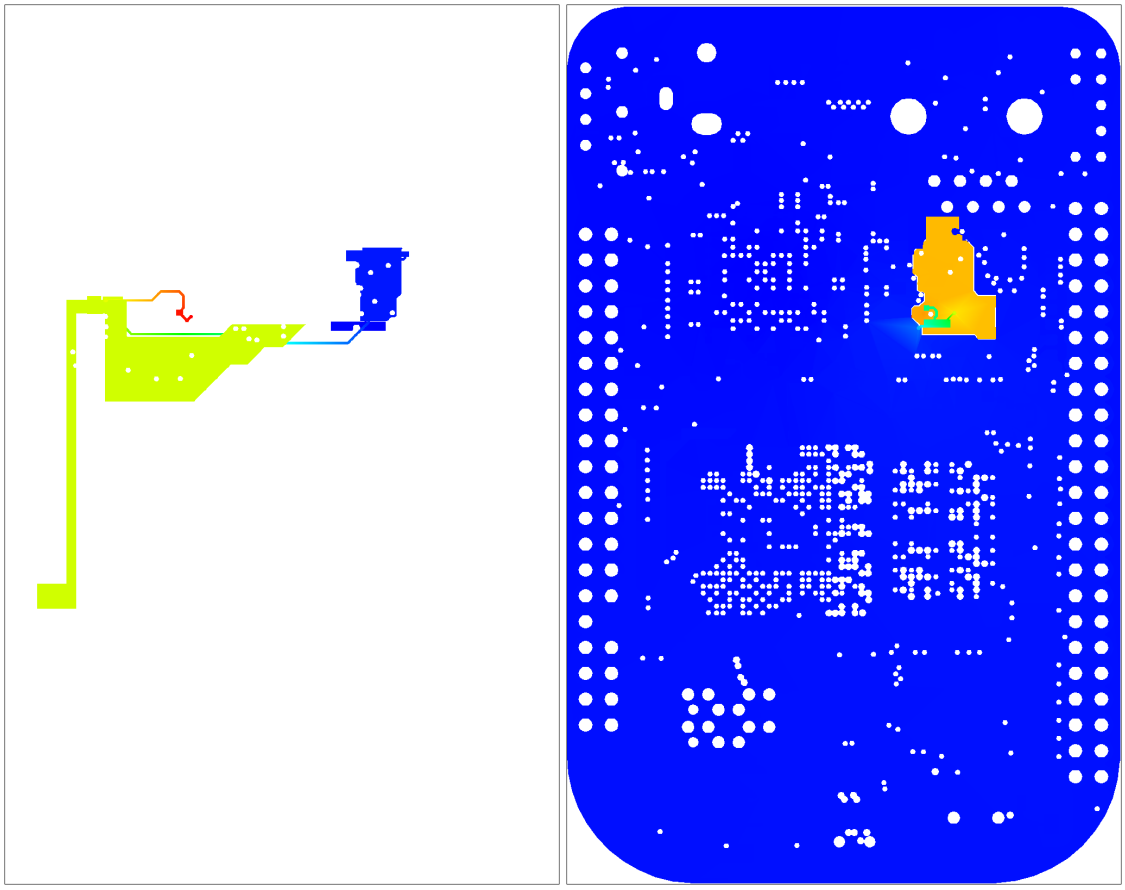


Figure A.24: VDD_ANALOGUE_3V3 Linear Voltage Heat Map from PMIC to Voltage Reference and Op-amps

A.4.0.13 VDD_PHY_5V Power Plane

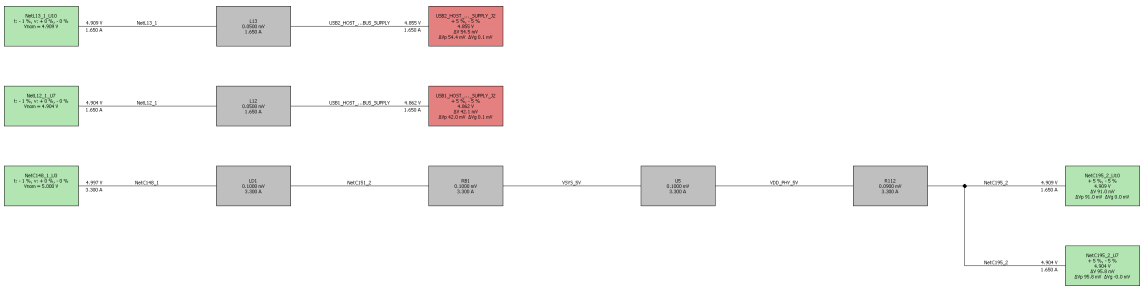


Figure A.25: VDD_PHY_5V Power tree from PMIC to Processor

Left: VDD_PHY_5V Linear Voltage Heat Map from PMIC to USB port. 100% red is 4.99677 Volts and 100% blue is 4.85458 Volts. Right: Ground Linear Voltage Heat Map from USB Port to PMIC (Return). 100% red is 0.00006 Volts and 100% blue is 0.00000 volts.

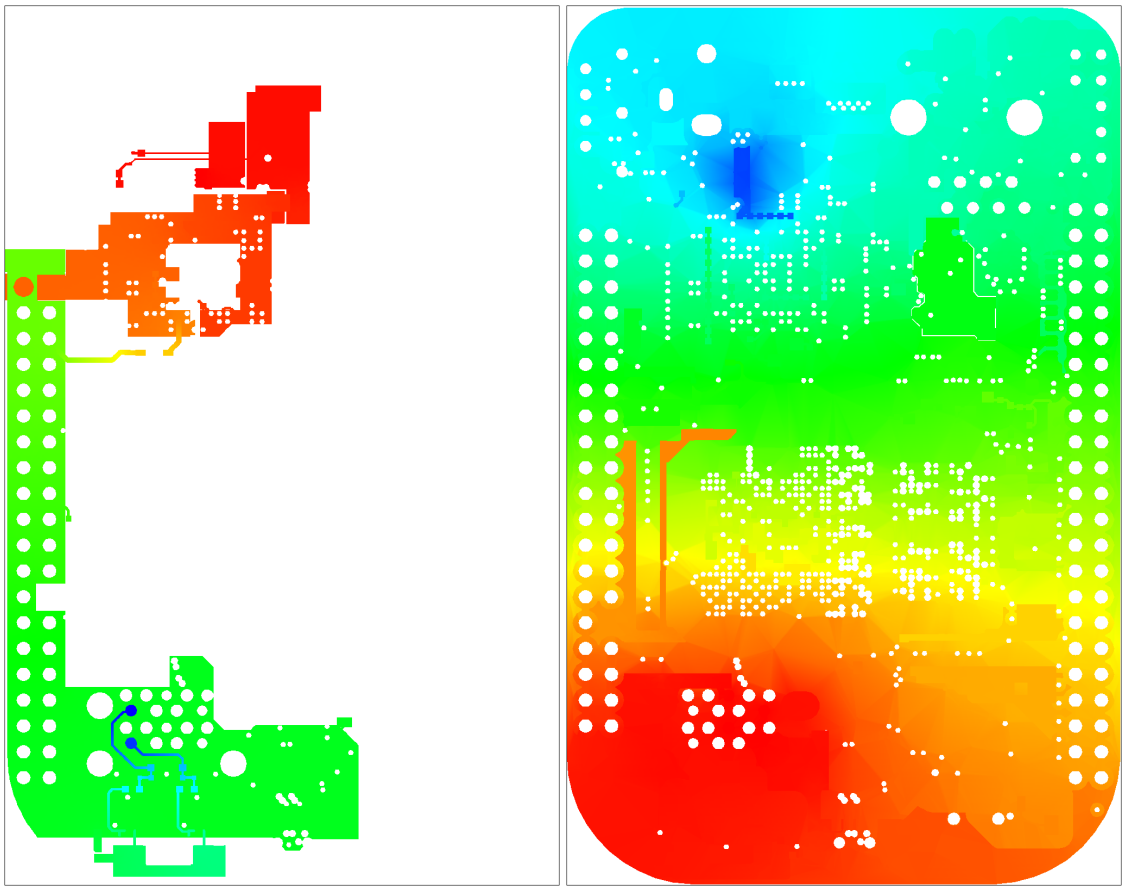
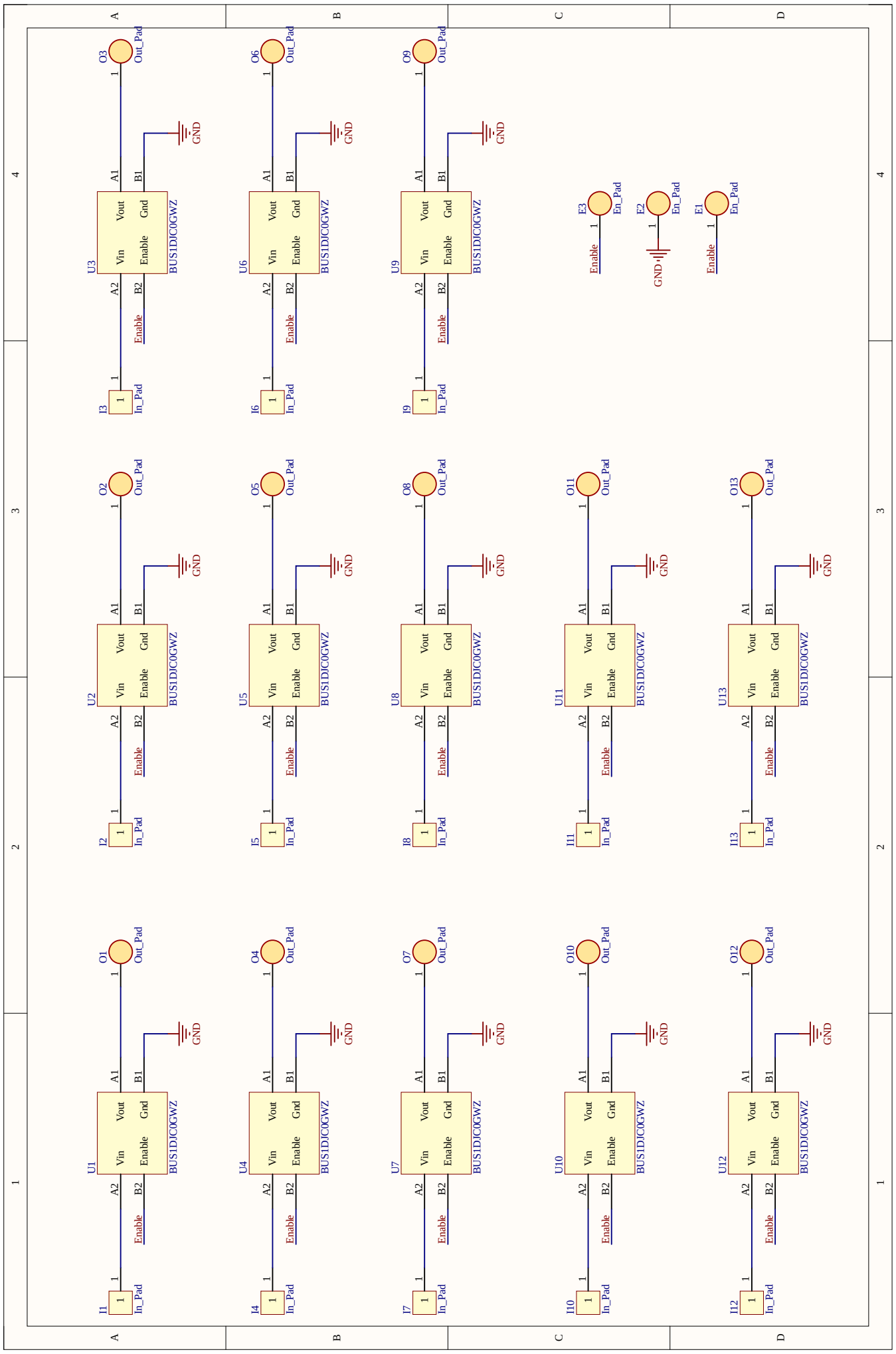


Figure A.26: VDD_PHY_5V Linear Voltage Heat Map from PMIC to USB port

Appendix B

Load Switch Mod-Board

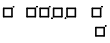
B.1 Load Switch Mod-Board Design

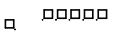


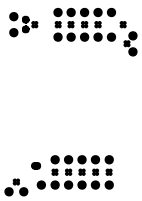
B.2 Load Switch Mod-Board Gerbers

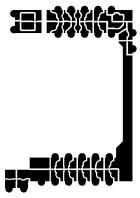
Layer order:

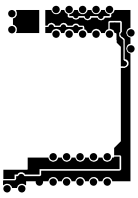
- Top Overlay
- Top Solder mask
- Top Copper Layer
- Bottom Copper Layer
- Bottom Overlay
- Drill Layer

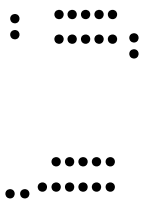


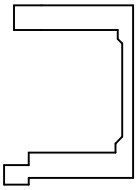












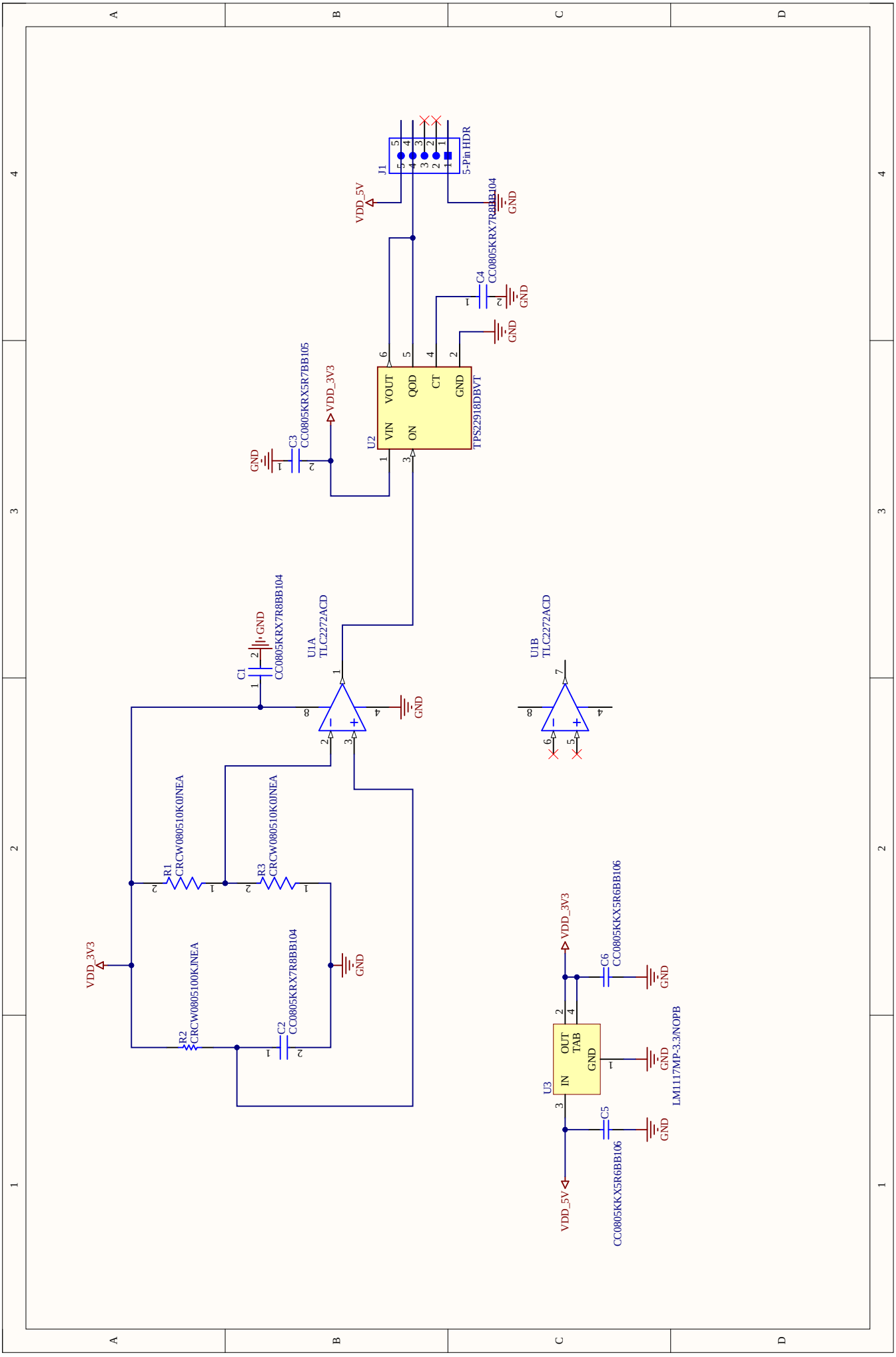
B.3 Load Switch Mod-Board Bill of Materials

Designator	Quantity	Mouser part No	Size	Value	Voltage	Tolerance%
U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13	13	TPS22914BYFPR	0.78mm x 0.78mm	2A	6V	-

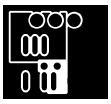
Appendix C

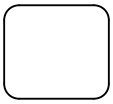
External Regulator Module

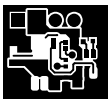
C.1 External Regulator Module Design



C.2 External Regulator Module Gerbers







C.3 External Regulator Module Bill of Materials

Designator	Quantity		Size	Value	Voltage	Tolerance%
		Integrated Circuits				
U1	1	TLC2272ACD				
U2	1	TPS22918DBVT				
U3	1	LM1117MP-3.3/NOPB				
		Resistors				
R1, R3	2	CRCW080510K0JNEA	805	10k	-	5
R2	1	CRCW0805100KJNEA	805	100k	-	5
		Capacitors				
C1, C2, C4	3	GRM21B7U1A104JA01L	805	100nF	10V	5
C3	1	GRM21BC72A105KE01L	805	1uF	100V	10
C5	1	GRM21BR61C106KE15L	805	10uF	16V	10
C6	1	GRM21BR61C106KE15L	805	10uF	16V	10
		Connectors				
J1	1	NA				

Appendix D

Bokkiebord Boot Log

```
U-Boot SPL 2020.04-5.4.24-2.1.0+ga80d22a3f2 (Nov 29 2021 - 19:11:45 +0000)
DDRINFO: start DRAM init
DDRINFO: DRAM rate 2400MTS
DDRINFO:ddrphy calibration done
DDRINFO: ddrmix config done
Normal Boot
Trying to boot from USB SDP
SDP: initialize...
SDP: handle requests...
Downloading file of size 1013184 to 0x40400000... done
Jumping to header at 0x40400000
Header Tag is not an IMX image
Found header at 0x40425fa0


U-Boot 2020.04-5.4.24-2.1.0+ga80d22a3f2 (Nov 29 2021 - 19:11:45 +0000)

CPU:   i.MX8MQ rev2.1 1500 MHz (running at 1000 MHz)
CPU:   Commercial temperature grade (0C to 95C) at 35C
Reset cause: POR
Model: NXP i.MX8MQ EVK
DRAM:  1 GiB
Bokkiebord: Setting up RMII Ethernet Clock
Bokkiebord: Resettig Ethernet PHY Reset is 1
gpio@30200000: dir_output: error: gpio GPIO1_14 not reserved
Bokkiebord: Resettig Ethernet PHY Reset is 0
gpio@30200000: dir_output: error: gpio GPIO1_14 not reserved
Bokkiebord: Resettig Ethernet PHY Reset is 1
gpio@30200000: dir_output: error: gpio GPIO1_14 not reserved
MMC:   FSL_SDHC: 0
Loading Environment from MMC... *** Warning - bad CRC, using default environment

[*]-Video Link 0imx8m_hdmi_probe
cdn_api_checkalive: keep-alive counter did not increment for 10us...
HDMI enable failed, ret -1!
 (1280 x 720)
        [0] display-controller@32e00000, video
        [1] hdmi@32c00000, display
In:    serial
Out:   serial
Err:   serial

 BuildInfo:
  - ATF b0a00f2
  - U-Boot 2020.04-5.4.24-2.1.0+ga80d22a3f2

switch to partitions #0, OK
mmc0 is current device
Detect USB boot. Will enter fastboot mode!
Net:
```

```
Error: ethernet@30be0000 address not set.

Error: ethernet@30be0000 address not set.
No ethernet found.

Fastboot: Normal
Boot from USB for mfgtools
*** Warning - Use default environment for                                    mfgtools
, using default environment

Run bootcmd_mfg: run mfgtool_args;if iminfo ${initrd_addr}; then if test ${tee} =
yes; then bootm ${tee_addr} ${initrd_addr} ${fdt_addr}; else booti ${loadaddr}
${initrd_addr} ${fdt_addr}; fi; else echo "Run fastboot ..."; fastboot 0; fi;
Hit any key to stop autoboot:  0
u-boot=>
u-boot=>
u-boot=>
u-boot=>
u-boot=>
u-boot=>
u-boot=>
u-boot=> setenv mmcroot "/dev/mmcblk0p2 rootwait rw"
u-boot=> boot
switch to partitions #0, OK
mmc0 is current device
28021248 bytes read in 1203 ms (22.2 MiB/s)
Booting from mmc ...
31404 bytes read in 17 ms (1.8 MiB/s)
## Flattened Device Tree blob at 43000000
   Booting using the fdt blob at 0x43000000
   Using Device Tree in place at 0000000043000000, end 000000004300aaab
Found /sound-hdmi node
Modify /sound-hdmi:status disabled
Found /sound-hdmi-arc node
Modify /sound-hdmi-arc:status disabled
Found /soc@0/bus@32c00000/display-controller@32e00000 node
Modify /soc@0/bus@32c00000/display-controller@32e00000:status disabled
Found /soc@0/bus@32c00000/hdmi@32c00000 node
Modify /soc@0/bus@32c00000/hdmi@32c00000:status disabled
Found /soc@0/usb@38100000 node
Modify /soc@0/usb@38100000:maximum-speed = high-speed

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0000000000 [0x410fd034]
[    0.000000] Linux version 5.4.24-2.1.0+gbce1862c0114 (oe-user@oe-host) (gcc
version 9.2.0 (GCC)) #1 SMP PREEMPT Fri Nov 26 03:48:18 UTC 2021
[    0.000000] Machine model: NXP i.MX8MQ EVK
[    0.000000] efi: Getting EFI parameters from FDT:
[    0.000000] efi: UEFI not found.
```

```
[    0.000000] cma: Reserved 320 MiB at 0x000000006c000000
[    0.000000] NUMA: No NUMA configuration found
[    0.000000] NUMA: Faking a node at [mem 0x0000000040000000-0x000000007fffffff]
[    0.000000] NUMA: NODE_DATA [mem 0x6bddd500-0x6bddefff]
[    0.000000] Zone ranges:
[    0.000000]   DMA32    [mem 0x0000000040000000-0x000000007fffffff]
[    0.000000]   Normal   empty
[    0.000000] Movable zone start for each node
[    0.000000] Early memory node ranges
[    0.000000]   node   0: [mem 0x0000000040000000-0x000000007fffffff]
[    0.000000] Initmem setup node 0 [mem 0x0000000040000000-0x000000007fffffff]
[    0.000000] psci: probing for conduit method from DT.
[    0.000000] psci: PSCIv1.1 detected in firmware.
[    0.000000] psci: Using standard PSCI v0.2 function IDs
[    0.000000] psci: MIGRATE_INFO_TYPE not supported.
[    0.000000] psci: SMC Calling Convention v1.1
[    0.000000] percpu: Embedded 24 pages/cpu s58904 r8192 d31208 u98304
[    0.000000] Detected VIPT I-cache on CPU0
[    0.000000] CPU features: detected: ARM erratum 845719
[    0.000000] CPU features: detected: GIC system register CPU interface
[    0.000000] Speculative Store Bypass Disable mitigation not required
[    0.000000] Built 1 zonelists, mobility grouping on.  Total pages: 258048
[    0.000000] Policy zone: DMA32
[    0.000000] Kernel command line: console=ttymxc0,115200 root=/dev/mmcblk0p2
rootwait rw
[    0.000000] Dentry cache hash table entries: 131072 (order: 8, 1048576 bytes,
linear)
[    0.000000] Inode-cache hash table entries: 65536 (order: 7, 524288 bytes,
linear)
[    0.000000] mem auto-init: stack:off, heap alloc:off, heap free:off
[    0.000000] Memory: 671964K/1048576K available (16444K kernel code, 1444K rwdata,
6472K rodata, 2944K init, 1017K bss, 48932K reserved, 327680K cma-reserved)
[    0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=4, Nodes=1
[    0.000000] rcu: Preemptible hierarchical RCU implementation.
[    0.000000] rcu:     RCU restricting CPUs from NR_CPUS=256 to nr_cpu_ids=4.
[    0.000000]  Tasks RCU enabled.
[    0.000000] rcu: RCU calculated value of scheduler-enlistment delay is 25
jiffies.
[    0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=4
[    0.000000] NR_IRQS: 64, nr_irqs: 64, preallocated irqs: 0
[    0.000000] GICv3: GIC: Using split EOI/Deactivate mode
[    0.000000] GICv3: 128 SPIs implemented
[    0.000000] GICv3: 0 Extended SPIs implemented
[    0.000000] GICv3: Distributor has no Range Selector support
[    0.000000] GICv3: 16 PPIs implemented
[    0.000000] GICv3: no VLPI support, no direct LPI support
[    0.000000] GICv3: CPU0: found redistributor 0 region 0:0x0000000038880000
[    0.000000] ITS: No ITS available, not enabling LPIs
[    0.000000] random: get_random_bytes called from start_kernel+0x2b8/0x44c with
crng_init=0
```

```
[    0.000000] arch_timer: cp15 timer(s) running at 8.33MHz (phys).
[    0.000000] clocksource: arch_sys_counter: mask: 0xffffffffffffff max_cycles:
0x1ec0311ec, max_idle_ns: 440795202152 ns
[    0.000003] sched_clock: 56 bits at 8MHz, resolution 120ns, wraps every
2199023255541ns
[    0.000382] Console: colour dummy device 80x25
[    0.000448] Calibrating delay loop (skipped), value calculated using timer
frequency.. 16.66 BogoMIPS (lpj=33333)
[    0.000458] pid_max: default: 32768 minimum: 301
[    0.000539] LSM: Security Framework initializing
[    0.000599] Mount-cache hash table entries: 2048 (order: 2, 16384 bytes, linear)
[    0.000611] Mountpoint-cache hash table entries: 2048 (order: 2, 16384 bytes,
linear)
[    0.024017] ASID allocator initialised with 32768 entries
[    0.032017] rcu: Hierarchical SRCU implementation.
[    0.040677] EFI services will not be available.
[    0.048049] smp: Bringing up secondary CPUs ...
[    0.080191] Detected VIPT I-cache on CPU1
[    0.080219] GICv3: CPU1: found redistributor 1 region 0:0x00000000388a0000
[    0.080243] CPU1: Booted secondary processor 0x0000000001 [0x410fd034]
[    0.112221] Detected VIPT I-cache on CPU2
[    0.112237] GICv3: CPU2: found redistributor 2 region 0:0x00000000388c0000
[    0.112254] CPU2: Booted secondary processor 0x0000000002 [0x410fd034]
[    0.144279] Detected VIPT I-cache on CPU3
[    0.144295] GICv3: CPU3: found redistributor 3 region 0:0x00000000388e0000
[    0.144311] CPU3: Booted secondary processor 0x0000000003 [0x410fd034]
[    0.144377] smp: Brought up 1 node, 4 CPUs
[    0.144400] SMP: Total of 4 processors activated.
[    0.144408] CPU features: detected: 32-bit EL0 Support
[    0.144415] CPU features: detected: CRC32 instructions
[    0.152740] CPU: All CPU(s) started at EL2
[    0.152766] alternatives: patching kernel code
[    0.154062] devtmpfs: initialized
[    0.159118] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff,
max_idle_ns: 7645041785100000 ns
[    0.159135] futex hash table entries: 1024 (order: 4, 65536 bytes, linear)
[    0.168452] pinctrl core: initialized pinctrl subsystem
[    0.169189] DMI not present or invalid.
[    0.169482] NET: Registered protocol family 16
[    0.178860] DMA: preallocated 256 KiB pool for atomic allocations
[    0.178874] audit: initializing netlink subsys (disabled)
[    0.179001] audit: type=2000 audit(0.176:1): state=initialized audit_enabled=0
res=1
[    0.179778] cpuidle: using governor menu
[    0.180197] hw-breakpoint: found 6 breakpoint and 4 watchpoint registers.
[    0.180959] Serial: AMBA PL011 UART driver
[    0.181029] imx mu driver is registered.
[    0.181053] imx rpmsg driver is registered.
[    0.184670] imx8mq-pinctrl 30330000.iomuxc: initialized IMX pinctrl driver
[    0.207079] HugeTLB registered 1.00 GiB page size, pre-allocated 0 pages
```

```
[    0.207090] HugeTLB registered 32.0 MiB page size, pre-allocated 0 pages
[    0.207096] HugeTLB registered 2.00 MiB page size, pre-allocated 0 pages
[    0.207102] HugeTLB registered 64.0 KiB page size, pre-allocated 0 pages
[    0.209048] cryptd: max_cpu_qlen set to 1000
[    0.214816] ACPI: Interpreter disabled.
[    0.215372] iommu: Default domain type: Translated
[    0.215519] vgaarb: loaded
[    0.215858] SCSI subsystem initialized
[    0.216212] usbcore: registered new interface driver usbfs
[    0.216255] usbcore: registered new interface driver hub
[    0.216317] usbcore: registered new device driver usb
[    0.216895] mc: Linux media interface: v0.10
[    0.216923] videodev: Linux video capture interface: v2.00
[    0.217004] pps_core: LinuxPPS API ver. 1 registered
[    0.217009] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti
<giometti@linux.it>
[    0.217025] PTP clock support registered
[    0.217137] EDAC MC: Ver: 3.0.0
[    0.217805] No BMan portals available!
[    0.218083] QMan: Allocated lookup table at (____ptrval____), entry count 65537
[    0.218309] No QMan portals available!
[    0.218645] No USDPAA memory, no 'fsl,usdpaa-mem' in device-tree
[    0.218977] FPGA manager framework
[    0.219067] Advanced Linux Sound Architecture Driver Initialized.
[    0.219500] Bluetooth: Core ver 2.22
[    0.219527] NET: Registered protocol family 31
[    0.219532] Bluetooth: HCI device and connection manager initialized
[    0.219542] Bluetooth: HCI socket layer initialized
[    0.219550] Bluetooth: L2CAP socket layer initialized
[    0.219564] Bluetooth: SCO socket layer initialized
[    0.220184] clocksource: Switched to clocksource arch_sys_counter
[    0.220327] VFS: Disk quotas dquot_6.6.0
[    0.220379] VFS: Dquot-cache hash table entries: 512 (order 0, 4096 bytes)
[    0.220570] pnp: PnP ACPI: disabled
[    0.226438] thermal_sys: Registered thermal governor 'step_wise'
[    0.226441] thermal_sys: Registered thermal governor 'power_allocator'
[    0.226732] NET: Registered protocol family 2
[    0.227067] tcp_listen_portaddr_hash hash table entries: 512 (order: 1, 8192
bytes, linear)
[    0.227091] TCP established hash table entries: 8192 (order: 4, 65536 bytes,
linear)
[    0.227163] TCP bind hash table entries: 8192 (order: 5, 131072 bytes, linear)
[    0.227300] TCP: Hash tables configured (established 8192 bind 8192)
[    0.227383] UDP hash table entries: 512 (order: 2, 16384 bytes, linear)
[    0.227413] UDP-Lite hash table entries: 512 (order: 2, 16384 bytes, linear)
[    0.227536] NET: Registered protocol family 1
[    0.227920] RPC: Registered named UNIX socket transport module.
[    0.227926] RPC: Registered udp transport module.
[    0.227930] RPC: Registered tcp transport module.
[    0.227934] RPC: Registered tcp NFSv4.1 backchannel transport module.
```

```
[    0.228449] PCI: CLS 0 bytes, default 64
[    0.229138] hw perfevents: enabled with armv8_cortex_a53 PMU driver, 7 counters
available
[    0.229398] kvm [1]: IPA Size Limit: 40bits
[    0.230105] kvm [1]: vgic-v2@31020000
[    0.230126] kvm [1]: GIC system register CPU interface enabled
[    0.230190] kvm [1]: vgic interrupt IRQ1
[    0.230298] kvm [1]: Hyp mode initialized successfully
[    0.236487] Initialise system trusted keyrings
[    0.236609] workingset: timestamp_bits=44 max_order=18 bucket_order=0
[    0.243161] squashfs: version 4.0 (2009/01/31) Phillip Lougher
[    0.243832] NFS: Registering the id_resolver key type
[    0.243854] Key type id_resolver registered
[    0.243859] Key type id_legacy registered
[    0.243870] nfs4filelayout_init: NFSv4 File Layout Driver Registering...
[    0.243896] jffs2: version 2.2. (NAND) © 2001-2006 Red Hat, Inc.
[    0.244319] 9p: Installing v9fs 9p2000 file system support
[    0.259772] Key type asymmetric registered
[    0.259779] Asymmetric key parser 'x509' registered
[    0.259815] Block layer SCSI generic (bsg) driver version 0.4 loaded (major 244)
[    0.259821] io scheduler mq-deadline registered
[    0.259827] io scheduler kyber registered
[    0.263789] EINJ: ACPI disabled.
[    0.271431] imx-sdma 302c0000.sdma: Direct firmware load for
imx/sdma/sdma-imx7d.bin failed with error -2
[    0.271445] imx-sdma 302c0000.sdma: Falling back to sysfs fallback for:
imx/sdma/sdma-imx7d.bin
[    0.276074] mxs-dma 33000000.dma-apbh: initialized
[    0.279645] Bus freq driver module loaded
[    0.283963] Serial: 8250/16550 driver, 4 ports, IRQ sharing enabled
[    0.285864] 30860000.serial: ttymxc0 at MMIO 0x30860000 (irq = 33, base_baud =
1562500) is a IMX
[    1.226552] printk: console [ttymxc0] enabled
[    1.243809] loop: module loaded
[    1.248950] imx ahci driver is registered.
[    1.255161] spi_imx 30830000.spi: dma setup error -19, use pio
[    1.261483] spi_imx 30830000.spi: probed
[    1.266635] libphy: Fixed MDIO Bus: probed
[    1.271556] tun: Universal TUN/TAP device driver, 1.6
[    1.277228] thunder_xcv, ver 1.0
[    1.280526] thunder_bgx, ver 1.0
[    1.283787] nicpf, ver 1.0
[    1.287167] pps pps0: new PPS source ptp0
[    1.296008] Freescale FM module, FMD API version 21.1.0
[    1.301457] Freescale FM Ports module
[    1.305129] fsl_mac: fsl_mac: FSL FMan MAC API based driver
[    1.310836] fsl_dpa: FSL DPAA Ethernet driver
[    1.315289] fsl_advanced: FSL DPAA Advanced drivers:
[    1.320261] fsl_proxy: FSL DPAA Proxy initialization driver
[    1.325927] fsl_oh: FSL FMan Offline Parsing port driver
```

```
[    1.332034] hclge is initializing
[    1.335360] hns3: Hisilicon Ethernet Network Driver for Hip08 Family - version
[    1.342587] hns3: Copyright (c) 2017 Huawei Corporation.
[    1.347947] e1000: Intel(R) PRO/1000 Network Driver - version 7.3.21-k8-NAPI
[    1.355004] e1000: Copyright (c) 1999-2006 Intel Corporation.
[    1.360796] e1000e: Intel(R) PRO/1000 Network Driver - 3.2.6-k
[    1.366634] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[    1.372595] igb: Intel(R) Gigabit Ethernet Network Driver - version 5.6.0-k
[    1.379562] igb: Copyright (c) 2007-2014 Intel Corporation.
[    1.385175] igbvf: Intel(R) Gigabit Virtual Function Network Driver - version
2.4.0-k
[    1.393010] igbvf: Copyright (c) 2009 - 2012 Intel Corporation.
[    1.399077] sky2: driver version 1.30
[    1.403575] VFIO - User Level meta-driver version: 0.3
[    1.410919] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    1.417467] ehci-pci: EHCI PCI platform driver
[    1.421959] ehci-platform: EHCI generic platform driver
[    1.427328] ohci_hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
[    1.433526] ohci-pci: OHCI PCI platform driver
[    1.438009] ohci-platform: OHCI generic platform driver
[    1.443711] usbcore: registered new interface driver usb-storage
[    1.449798] usbcore: registered new interface driver usbserial_generic
[    1.456353] usbserial: USB Serial support registered for generic
[    1.462394] usbcore: registered new interface driver ftdi_sio
[    1.468164] usbserial: USB Serial support registered for FTDI USB Serial Device
[    1.475506] usbcore: registered new interface driver usb_serial_simple
[    1.482057] usbserial: USB Serial support registered for carelink
[    1.488182] usbserial: USB Serial support registered for zio
[    1.493865] usbserial: USB Serial support registered for funsoft
[    1.499897] usbserial: USB Serial support registered for flashloader
[    1.506280] usbserial: USB Serial support registered for google
[    1.512224] usbserial: USB Serial support registered for libtransistor
[    1.518778] usbserial: USB Serial support registered for vivopay
[    1.524813] usbserial: USB Serial support registered for moto_modem
[    1.531101] usbserial: USB Serial support registered for motorola_tetra
[    1.537739] usbserial: USB Serial support registered for novatel_gps
[    1.544118] usbserial: USB Serial support registered for hp4x
[    1.549887] usbserial: USB Serial support registered for suunto
[    1.555830] usbserial: USB Serial support registered for siemens_mpi
[    1.564400] input: 30370000.snvs:snvs-powerkey as
/devices/platform/soc@0/soc@0:bus@30000000/30370000.snvs/30370000.snvs:snvs-powerkey
/input/input0
[    1.579395] snvs_rtc 30370000.snvs:snvs-rtc-lp: registered as rtc0
[    1.585663] i2c /dev entries driver
[    1.591752] imx2-wdt 30280000.watchdog: timeout 60 sec (nowayout=0)
[    1.598279] Bluetooth: HCI UART driver ver 2.3
[    1.602738] Bluetooth: HCI UART protocol H4 registered
[    1.607886] Bluetooth: HCI UART protocol BCSP registered
[    1.613232] Bluetooth: HCI UART protocol LL registered
[    1.618380] Bluetooth: HCI UART protocol ATH3K registered
```

```
[    1.623805] Bluetooth: HCI UART protocol Three-wire (H5) registered
[    1.630158] Bluetooth: HCI UART protocol Broadcom registered
[    1.635846] Bluetooth: HCI UART protocol QCA registered
[    1.642875] sdhci: Secure Digital Host Controller Interface driver
[    1.649074] sdhci: Copyright(c) Pierre Ossman
[    1.653609] Synopsys Designware Multimedia Card Interface Driver
[    1.660128] sdhci-pltfm: SDHCI platform and OF driver helper
[    1.699400] mmc0: SDHCI controller on 30b40000.mmc [30b40000.mmc] using ADMA
[    1.709067] ledtrig-cpu: registered to indicate activity on CPUs
[    1.716263] caam 30900000.crypto: device ID = 0x0a16040100000000 (Era 9)
[    1.722994] caam 30900000.crypto: job rings = 3, qi = 0
[    1.756652] caam algorithms registered in /proc/crypto
[    1.761828] mmc0: host does not support reading read-only switch, assuming
write-enable
[    1.766109] caam 30900000.crypto: caam pkc algorithms registered in /proc/crypto
[    1.777913] mmc0: new high speed SDHC card at address e624
[    1.779440] caam_jr 30901000.jr: registering rng-caam
[    1.784536] mmcblk0: mmc0:e624 SS16G 14.8 GiB
[    1.794144] caam-snvs 30370000.caam-snvs: can't get snvs clock
[    1.800085] caam-snvs 30370000.caam-snvs: violation handlers armed - non-secure
state
[    1.806999]  mmcblk0: p1 p2
[    1.808593] usbcore: registered new interface driver usbhid
[    1.816282] usbhid: USB HID core driver
[    1.821892] No fsl,qman node
[    1.824799] Freescale USDPAA process driver
[    1.828990] fsl-usdpaa: no region found
[    1.832835] Freescale USDPAA process IRQ driver
[    1.841411] galcore: clk_get 2d core clock failed, disable 2d/vg!
[    1.847757] Galcore version 6.4.0.234062
[    1.982656] hantrodec: module inserted. Major = 236
[    2.054615] imx-spdif sound-spdif: snd-soc-dummy-dai <-> 30810000.spdif mapping
ok
[    2.062238] imx-spdif sound-spdif: ASoC: no DMI vendor name!
[    2.073126] pktgen: Packet Generator for packet performance testing. Version:
2.75
[    2.088377] NET: Registered protocol family 26
[    2.093465] NET: Registered protocol family 10
[    2.102548] Segment Routing with IPv6
[    2.106271] NET: Registered protocol family 17
[    2.111330] Bluetooth: RFCOMM TTY layer initialized
[    2.116232] Bluetooth: RFCOMM socket layer initialized
[    2.121403] Bluetooth: RFCOMM ver 1.11
[    2.125180] Bluetooth: BNEP (Ethernet Emulation) ver 1.3
[    2.130500] Bluetooth: BNEP filters: protocol multicast
[    2.135734] Bluetooth: BNEP socket layer initialized
[    2.140707] Bluetooth: HIDP (Human Interface Emulation) ver 1.2
[    2.146638] Bluetooth: HIDP socket layer initialized
[    2.151643] 8021q: 802.1Q VLAN Support v1.8
[    2.155854] lib80211: common routines for IEEE802.11 drivers
```

```
[    2.161643] 9pnet: Installing 9P2000 support
[    2.165949] tsn generic netlink module v1 init...
[    2.170721] Key type dns_resolver registered
[    2.176435] registered taskstats version 1
[    2.180552] Loading compiled-in X.509 certificates
[    2.211400] imx8mq-usb-phy 381f0040.usb-phy: 381f0040.usb-phy supply vbus not
found, using dummy regulator
[    2.221349] imx8mq-usb-phy 382f0040.usb-phy: 382f0040.usb-phy supply vbus not
found, using dummy regulator
[    2.233307] pps pps0: new PPS source ptp0
[    2.244083] fec 30be0000.ethernet: Invalid MAC address: 00:00:00:00:00:00
[    2.250936] fec 30be0000.ethernet: Using random MAC address: 6a:46:24:fc:a1:7f
[    2.259337] libphy: fec_enet_mii_bus: probed
[    2.265951] fec 30be0000.ethernet eth0: registered PHC device 0
[    2.272978] OF: graph: no port node found in /soc@0/usb-phy@381f0040
[    2.280380] xhci-hcd xhci-hcd.1.auto: xHCI Host Controller
[    2.285904] xhci-hcd xhci-hcd.1.auto: new USB bus registered, assigned bus number
1
[    2.293704] xhci-hcd xhci-hcd.1.auto: hcc params 0x0220fe6c hci version 0x110
quirks 0x0000000001010010
[    2.303154] xhci-hcd xhci-hcd.1.auto: irq 44, io mem 0x38200000
[    2.310280] hub 1-0:1.0: USB hub found
[    2.314076] hub 1-0:1.0: 1 port detected
[    2.318263] xhci-hcd xhci-hcd.1.auto: xHCI Host Controller
[    2.323772] xhci-hcd xhci-hcd.1.auto: new USB bus registered, assigned bus number
2
[    2.331446] xhci-hcd xhci-hcd.1.auto: Host supports USB 3.0 SuperSpeed
[    2.338036] usb usb2: We don't know the algorithms for LPM for this host,
disabling LPM.
[    2.347592] hub 2-0:1.0: USB hub found
[    2.351377] hub 2-0:1.0: 1 port detected
[    2.355833] imx-cpufreq-dt imx-cpufreq-dt: cpu speed grade 3 mkt segment 0
supported-hw 0x8 0x1
[    2.364911] cpu cpu0: _opp_is_duplicate: duplicate OPPs detected. Existing: freq:
800000000, volt: 900000, enabled: 1. New: freq: 800000000, volt: 900000, enabled: 1
[    2.379767] cpufreq: cpufreq_online: CPU0: Running at unlisted freq: 1000000 KHz
[    2.387188] cpu cpu0: dev_pm_opp_set_rate: failed to find current OPP for freq
1000000000 (-34)
[    2.396009] cpufreq: cpufreq_online: CPU0: Unlisted initial frequency changed to:
800000 KHz
[    2.404663] hantro receive hot notification event: 0
[    2.412312] snvs_rtc 30370000.snvs:snvs-rtc-lp: setting system clock to
1970-01-01T00:00:00 UTC (0)
[    2.421704] cfg80211: Loading compiled-in X.509 certificates for regulatory
database
[    2.434781] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[    2.441417] platform regulatory.0: Direct firmware load for regulatory.db failed
with error -2
[    2.444990] ALSA device list:
[    2.450062] platform regulatory.0: Falling back to sysfs fallback for:
```

```
regulatory.db
[    2.453035]   #0: imx-spdif
[    2.484709] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts:
(null)
[    2.492866] VFS: Mounted root (ext4 filesystem) on device 179:2.
[    2.502742] devtmpfs: mounted
[    2.507343] Freeing unused kernel memory: 2944K
[    2.524820] Run /sbin/init as init process
[    2.888280] random: fast init done
[    2.903954] systemd[1]: System time before build time, advancing clock.
[    2.948881] systemd[1]: systemd 243.2+ running in system mode. (+PAM -AUDIT
-SELINUX +IMA -APPARMOR +SMACK +SYSVINIT +UTMP -LIBCRYPTSETUP -GCRYPT -GNUTLS +ACL
+XZ -LZ4 -SECCOMP +BLKID -ELFUTILS +KMOD -IDN2 -IDN -PCRE2 default-hierarchy=hybrid)
[    2.971026] systemd[1]: Detected architecture arm64.

Welcome to NXP i.MX Release Distro 5.4-zeus (zeus)!

[    3.032097] systemd[1]: Set hostname to <imx8mqevk>.
[    3.052818] random: systemd: uninitialized urandom read (16 bytes read)
[    3.059500] systemd[1]: Initializing machine ID from random generator.
[    3.403386] systemd[1]: /lib/systemd/system/dbus.socket:5: ListenStream=
references a path below legacy directory /var/run/, updating
/var/run/dbus/system_bus_socket → /run/dbus/system_bus_socket; please update the
unit file accordingly.
[    3.586036] systemd[1]: /lib/systemd/system/rpcbind.socket:5: ListenStream=
references a path below legacy directory /var/run/, updating /var/run/rpcbind.sock →
/run/rpcbind.sock; please update the unit file accordingly.
[    3.625218] random: systemd: uninitialized urandom read (16 bytes read)
[    3.632060] systemd[1]: system-getty.slice: unit configures an IP firewall, but
the local system does not support BPF/cgroup firewalling.
[    3.644487] systemd[1]: (This warning is only shown for the first unit using IP
firewalling.)
[    3.655941] systemd[1]: Created slice system-getty.slice.
[  OK  ] Created slice system-getty.slice.
[    3.680345] random: systemd: uninitialized urandom read (16 bytes read)
[    3.688021] systemd[1]: Created slice system-serial\x2dgetty.slice.
[  OK  ] Created slice system-serial\x2dgetty.slice.
[  OK  ] Created slice User and Session Slice.
[  OK  ] Started Dispatch Password …ts to Console Directory Watch.
[  OK  ] Started Forward Password R…uests to Wall Directory Watch.
[  OK  ] Reached target Host and Network Name Lookups.
[  OK  ] Reached target Paths.
[  OK  ] Reached target Remote File Systems.
[  OK  ] Reached target Slices.
[  OK  ] Reached target Swap.
[  OK  ] Listening on Syslog Socket.
[  OK  ] Listening on initctl Compatibility Named Pipe.
[  OK  ] Listening on Journal Audit Socket.
[  OK  ] Listening on Journal Socket (/dev/log).
[  OK  ] Listening on Journal Socket.
```

```
[  OK  ] Listening on Network Service Netlink Socket.
[  OK  ] Listening on udev Control Socket.
[  OK  ] Listening on udev Kernel Socket.
         Mounting Huge Pages File System...
         Mounting POSIX Message Queue File System...
         Mounting Kernel Debug File System...
         Mounting Temporary Directory (/tmp)...
         Starting Create list of st…odes for the current kernel...
         Starting Journal Service...
         Mounting Kernel Configuration File System...
         Starting Remount Root and Kernel File Systems...
[    4.136146] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
         Starting Apply Kernel Variables...
         Starting udev Coldplug all Devices...
[  OK  ] Started Journal Service.
[  OK  ] Mounted Huge Pages File System.
[  OK  ] Mounted POSIX Message Queue File System.
[  OK  ] Mounted Kernel Debug File System.
[  OK  ] Mounted Temporary Directory (/tmp).
[  OK  ] Started Create list of sta… nodes for the current kernel.
[  OK  ] Mounted Kernel Configuration File System.
[  OK  ] Started Remount Root and Kernel File Systems.
[  OK  ] Started Apply Kernel Variables.
         Starting Flush Journal to Persistent Storage...
         Starting Create System Users...
[    4.366131] systemd-journald[361]: Received client request to flush runtime
journal.
[  OK  ] Started Flush Journal to Persistent Storage.
[  OK  ] Started Create System Users.
         Starting Create Static Device Nodes in /dev...
[  OK  ] Started Create Static Device Nodes in /dev.
[  OK  ] Reached target Local File Systems (Pre).
         Mounting /var/volatile...
         Starting udev Kernel Device Manager...
[  OK  ] Mounted /var/volatile.
         Starting Load/Save Random Seed...
[  OK  ] Reached target Local File Systems.
         Starting Rebuild Dynamic Linker Cache...
         Starting Create Volatile Files and Directories...
[  OK  ] Started Create Volatile Files and Directories.
[  OK  ] Started udev Coldplug all Devices.
         Starting Run pending postinsts...
         Starting Rebuild Journal Catalog...
         Starting Network Time Synchronization...
         Starting Update UTMP about System Boot/Shutdown...
[  OK  ] Started udev Kernel Device Manager.
[  OK  ] Started Update UTMP about System Boot/Shutdown.
[  OK  ] Started Rebuild Journal Catalog.
[  OK  ] Started Network Time Synchronization.
[  OK  ] Reached target System Time Set.
```

```
[  OK  ] Reached target System Time Synchronized.
[  OK  ] Started Run pending postinsts.
[  OK  ] Created slice system-systemd\x2dfsck.slice.
[  OK  ] Found device /dev/mmcblk0p1.
[    6.860228] random: crng init done
[    6.863645] random: 7 urandom warning(s) missed due to ratelimiting
         Starting File System Check on /dev/mmcblk0p1...
[  OK  ] Started Load/Save Random Seed.
[  OK  ] Started File System Check on /dev/mmcblk0p1.
         Mounting /run/media/mmcblk0p1...
[  OK  ] Mounted /run/media/mmcblk0p1.
[  OK  ] Started Rebuild Dynamic Linker Cache.
         Starting Update is Completed...
[  OK  ] Started Update is Completed.
[  OK  ] Reached target System Initialization.
[  OK  ] Started Daily apt download activities.
[  OK  ] Started Daily rotation of log files.
[  OK  ] Started Daily Cleanup of Temporary Directories.
[  OK  ] Reached target Timers.
[  OK  ] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[  OK  ] Listening on D-Bus System Message Bus Socket.
[  OK  ] Listening on RPCbind Server Activation Socket.
[  OK  ] Reached target Sockets.
[  OK  ] Reached target Basic System.
         Starting Save/Restore Sound Card State...
[  OK  ] Started Job spooling tools.
[  OK  ] Started Kernel Logging Service.
[  OK  ] Started System Logging Service.
[  OK  ] Started Periodic Command Scheduler.
[  OK  ] Started D-Bus System Message Bus.
[  OK  ] Started Configuration for i.MX GPU (Former rc_gpu.S).
         Starting Packet Filtering Framework...
         Starting Network Time Service (one-shot ntpdate mode)...
         Starting Telephony service...
[  OK  ] Started User Space Regulatory Firmware Loading.
[  OK  ] Started User Space SDMA Firmware Loading.
         Starting Login Service...
[  OK  ] Started Save/Restore Sound Card State.
[  OK  ] Started Packet Filtering Framework.
[    8.212718] imx-sdma 30bd0000.sdma: loaded firmware 4.5
[  OK  ] Started Network Time Service (one-shot ntpdate mode).
[  OK  ] Reached target Network (Pre).
[  OK  ] Reached target Sound Card.
         Starting Connection service...
         Starting Network Service...
[  OK  ] Started Telephony service.
[  OK  ] Started Login Service.
[  OK  ] Started Network Service.
[    8.477476] Generic PHY 30be0000.ethernet-1:00: attached PHY driver [Generic PHY]
(mii_bus:phy_addr=30be0000.ethernet-1:00, irq=POLL)
```

```
[  OK  ] Started Connection service.
[  OK  ] Reached target Network.
         Starting Avahi mDNS/DNS-SD Stack...
         Starting /etc/rc.local Compatibility...
         Starting Permit User Sessions...
[  OK  ] Started /etc/rc.local Compatibility.
[  OK  ] Started Permit User Sessions.
[  OK  ] Started Avahi mDNS/DNS-SD Stack.
[  OK  ] Started Getty on tty1.
[  OK  ] Started Serial Getty on ttymxc0.
[  OK  ] Reached target Login Prompts.
[  OK  ] Reached target Multi-User System.
         Starting Hostname Service...
         Starting Update UTMP about System Runlevel Changes...
         Starting WPA supplicant...
[  OK  ] Started Update UTMP about System Runlevel Changes.
[  OK  ] Started Hostname Service.
[  OK  ] Started WPA supplicant.

NXP i.MX Release Distro 5.4-zeus imx8mqevk ttymxc0

imx8mqevk login: root
[   13.569740] audit: type=1006 audit(1637892246.663:2): pid=579 uid=0
old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=1 res=1
root@imx8mqevk:~# ls
root@imx8mqevk:~#
```