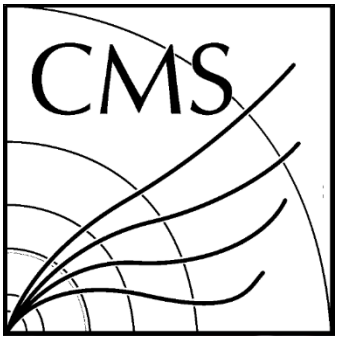


UNIVERSITY OF IOANNINA
PHYSICS DEPARTMENT

A demonstrator system using a Xilinx ZYNQ FPGA and 6.6 Gbps optical links

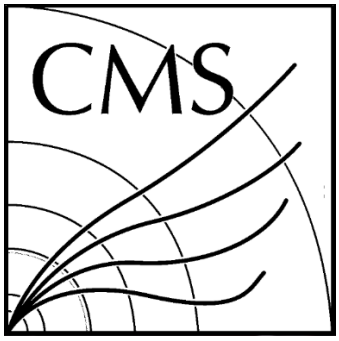
Ioannis Bestintzanos

Physicist

MASTER THESIS

Supervisor: Constantinos Foudas

Ioannina, 2020



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΤΜΗΜΑ ΦΥΣΙΚΗΣ

Σύστημα επίδειξης που χρησιμοποιεί την ZYNQ FPGA της Xilinx και 6.6 Gbps οπτικούς συνδέσμους

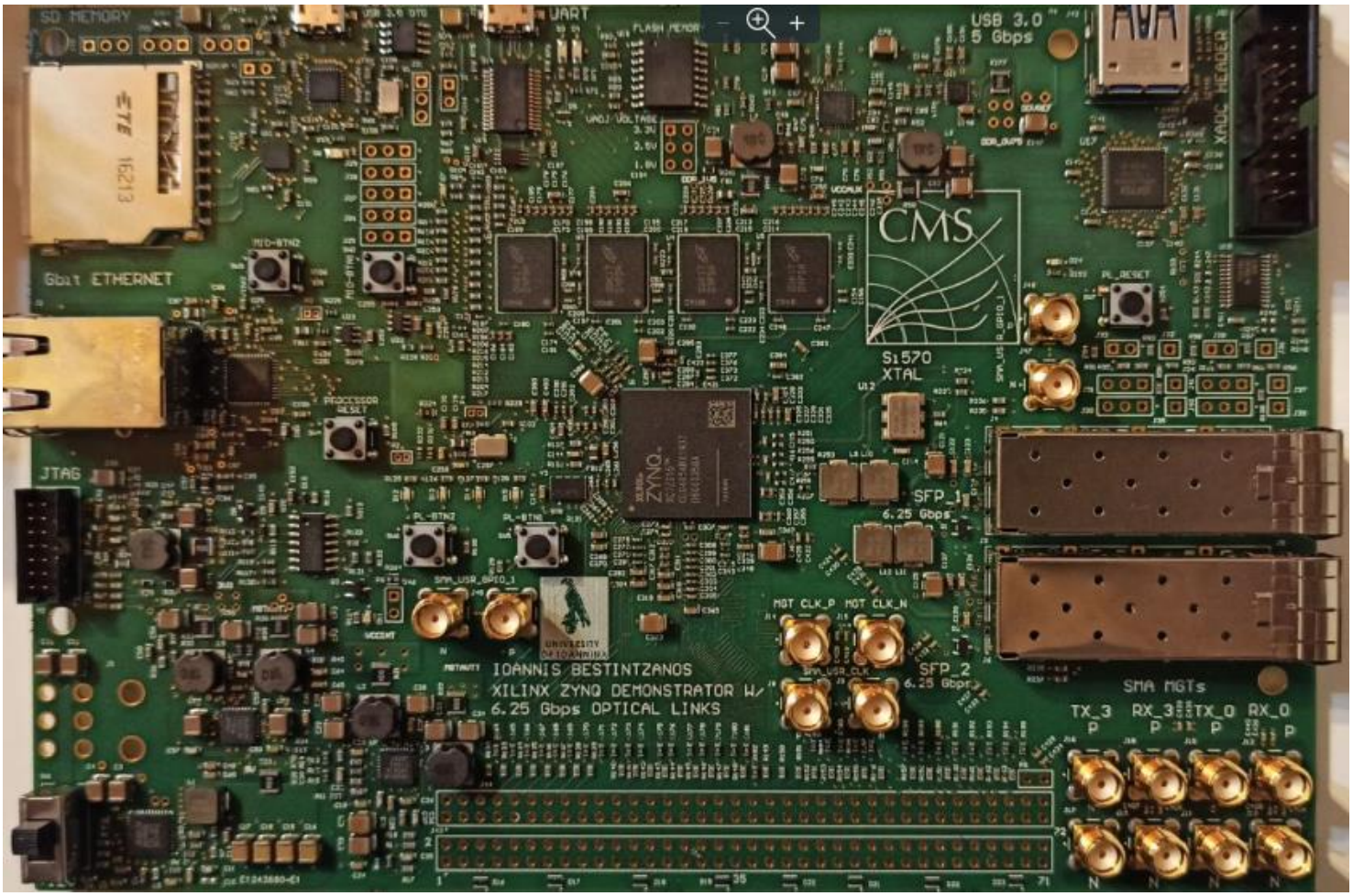
Ιωάννης Μπεστιντζάνος

Φυσικός

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων καθηγητής: Κωνσταντίνος Φουντάς

Ιωάννινα, 2020


SD MEMORY
UART
USB 3.0
5 Gbps
XADC HEADER
Gbit ETHERNET
CMS
Si570
XTAL
PROCESSOR
RESET
JTAG
SFP_1
6.25 Gbps
SFP_2
6.25 Gbps
MGT CLK_P
MGT CLK_N
SMA MGTs
TX_3
RX_3
TX_0
RX_0
IOANNIS BESTINTZANOS
XILINX ZYNQ DEMONSTRATOR W/
6.25 Gbps OPTICAL LINKS

Abstract

In this thesis a complete digital signal processing system is designed, imprinted on a Printed Circuit Board, manufactured and tested. This system uses a novel computing architecture, combining big amounts of reconfigurable logic resources with a capable dual-core processor, tightly coupled in the same silicon die. A goal is to provide the end-user with many interfacing options, aiding flexibility. This flexibility is provided using 6.6 Gbps optical links, USB 2.0 & 3.0, Gigabit Ethernet, as well as a big quantity of high-speed optimized headers. Furthermore, the card should be able to run an operating system, giving the ability to the user to run high-level applications, such as CERN's ROOT in this case, providing a very powerful tool for data processing. Special effort has been put in providing the best signal integrity while keeping PCB complexity and cost low, proving that such systems could start becoming more mainstream. The system was successful, and every subsystem could function as expected from the prototype.

Περίληψη

Σε αυτή τη διπλωματική εργασία ένα σύστημα επεξεργασίας σημάτων σχεδιάζεται, αποτυπώνεται σαν τυπωμένο κύκλωμα, κατασκευάζεται και δοκιμάζεται. Το σύστημα αυτό χρησιμοποιεί καινοτόμα αρχιτεκτονική, συνδυάζοντας μεγάλες ποσότητες επαναπρογραμματίσιμης λογικής, καθώς και έναν ισχυρό, διπύρηνο επεξεργαστή στενά συνδεδεμένα μέσα στο ίδιο ολοκληρωμένο κύκλωμα. Ένας από τους στόχους της παρούσας εργασίας είναι να εφοδιάσει τον τελικό χρήστη με μεγάλη ευελιξία στη συνδεσιμότητα. Για τον λόγο αυτό η κάρτα προσφέρει οπτικούς συνδέσμους στα 6.25 Gbps, USB 2.0 & 3.0, Gigabit Ethernet, καθώς και πληθώρα ηλεκτρικών βυσμάτων, βελτιστοποιημένα για υψίσυχνη λειτουργία. Ακόμα ένας στόχος είναι η κάρτα αυτή να μπορεί να τρέξει λειτουργικό σύστημα, έτσι ώστε να παρέχει στον χρήστη τη δυνατότητα χρήσης εφαρμογών υψηλού επιπέδου. Ένα παράδειγμα αυτών, που πράχθηκε στη συγκεκριμένη περίπτωση, είανι η χρήση του ROOT, που δημιουργήθηκε από το CERN, το οποίο αποτελεί ένα πολύ ικανό εργαλείο για την επεξεργασία δεδομένων. Έμφαση δόθηκε στο να παραμείνει το κόστος της κάρτας μικρό, χωρίς παράλληλα να γίνει κάποια υποβάθμιση στην ακεραιότητα των σημάτων, κάτι που μπορεί να δείξει οτι παρόμοια συστήματα μπορούν να είναι πιο προσιτά. Το σύστημα ήταν επιτυχές, καθώς από το πρωτότυπο η κάρτα είναι πλήρως λειτουργική.

Acknowledgments

Given this chance, I would like to express my gratitude to some people, which have helped me throughout this thesis and throughout my life as well. First of all, to my supervisor, Costas Foudas, who is a very knowledgeable person and offered me the chance to enter the world of high-energy physics and high performance electronics. His precious advices are perhaps the most important resource in this competitive field of science. I would also like to thank my family and Ourania Kyrpianou for the endless support they have given me, and the patience they have shown.

Contents

Abstract.....1

Περίληψη.....2

Acknowledgments.....3

Chapter 1: Introduction.....11

1.1 Embedded systems.....11

1.2 State of the art.....11

1.3 This design.....12

1.4 Possible uses.....12

1.5 Other existing similar systems.....13

1.6 Future implementations.....13

Chapter 2: FPGA Technology.....14

2.1 Configurable Logic Blocks.....15

2.2 Routing and clocking network.....16

2.3 FPGA Features.....16

2.4 Bleeding edge devices.....18

2.5 Hardware Description languages.....19

2.6 Simulation.....22

2.7 High Level Synthesis.....22

Chapter 3: Printed Circuit Boards.....23

3.1 PCB Technology and manufacturing.....23

3.2 PCB Assembly.....27

3.3 Technical specifications.....29

3.4 Signal integrity 34
3.5 Design methods 41
3.6 This PCB 43
Chapter 4: System components 48
4.1 High level block diagram 48
4.2 Power Supply 49
4.3 Buck converter 49
4.4 Power distribution network 51
4.4.1 12V 52
4.4.2 5V 52
4.4.3 3.3V 54
4.4.4 VCCINT 55
4.4.5 VCC1V8 56
4.4.6 MGTAVCC 57
4.4.7 MGTAVTT 58
4.4.8 VCCAUX 59
4.4.9 DDR1V5 60
4.4.10 DDR0V75 61
4.4.11 DDVREF 62
4.4.12 XADC-1V25 63
4.4.13 XADC-1V8 63
4.4.14 Power sequencing 63
4.4.15 Debugging features 65
4.5 ZYNQ Device 66
4.5.1 ZYNQ Technology 66

4.5.2 XC7Z015-2CLG485C 70
4.6 Random Access Memory 73
4.6.1 DDR memory 73
4.6.2 RAM on this board 75
4.6.3 DDR3 interface 76
4.6.4 Fly-By routing topology 77
4.6.5 DDR3 signal integrity requirements 78
4.6.6 DDR3 implementation on this board 79
4.7 Multi Gigabit Transceivers 87
4.7.1 MGT technologies 89
4.7.2 Transceivers on the XC7Z015 device 92
4.7.3 MGT internal structure 93
4.7.4 Characterization 94
4.7.5 MGT implementation on this board 96
4.8 Ethernet 102
4.8.1 Ethernet technology 102
4.8.2 This Ethernet implementation 105
4.9 UART 107
4.9.1 Universal Asynchronous Receiver Transmitter 107
4.9.2 UART implementation on this board 108
4.10 SD 110
4.11 QSPI flash memory 112
4.12 USB 2.0 114
4.13 USB 3.0 118
4.14 Pin header batch I/O 122

4.15 Generic Input / Output 127
4.16 JTAG 128
4.17 I2C Multiplexer 128
4.18 Boot mode configuration 131
4.19 XADC header 131
Chapter 5: Testing and performance 133
5.1 Power supply 133
5.2 Clocks 134
5.3 FPGA 134
5.4 Processor 136
5.5 RAM 136
5.6 SD card & Ethernet 138
5.7 Multi Gigabit Transceivers 138
5.8 Extended testing 140
5.9 Co-operation of the ARM processor and the FPGA for histogramming using root 142
Chapter 6: Conclusion 146
Chapter 7: Appendix 147
7.1 References 147
7.2 Figure list 150
7.3 Schematics 156
7.4 Gerber files 198
7.5 Trace lengths report 218
Address / Clock / Command PP1 218
Address / Clock / Command PP2 219

Address / Clock / Command PP3 220
Address / Clock / Command PP4 221
1st data byte 222
3rd data byte 223
4th data byte 223
7.6 Bill of materials 224
7.7 Power supply measurement data 246
Idle measurements 247
Stressed measurements 256
7.8 Eyescans 264
SFP 1 (X0Y1(TX)/X0Y2(RX)) 264
SFP 2 (X0Y2(TX)/X0Y1(RX)) 268
SMA MGTs 272
7.9 Errata 275

Chapter 1: Introduction

1.1 Embedded systems

On scientific and industrial applications, very often arises the need for a digital system that performs a specific task. Be it experimental high energy physics, telecommunications, astronomy or autonomous driving, these fields cannot perform without specialized digital systems. On high energy physics, the physicists need to be able to process all the data the detector provides, analyze the physics, and trigger on specific events for further processing. The data rates are extraordinary though, in the order of many petabytes per second. Using off-the-shelf systems to perform such tasks is out of discussion. Assembling these systems out of generic, sequential processors is out of discussion as well. These devices cannot handle data rates anywhere near what is needed. The same goes for a cellular 5G basestation, no commercially available systems can be used for such tasks. Or interfacing to a gigapixel camera, to search for the faintest objects in the universe, can be accomplished by off-the-self systems? Or perhaps running real time object recognition for a self-driving car? None of the above applications, and many more, can be served by commercial, off-the-self systems. For them, special systems have to be designed, and usually that does not only mean designing the board to interface the various components, but custom designing the logic as well.

1.2 State of the art

Most state-of-the-art custom systems can be found, as previously noted, in the most demanding and exotic applications. IC prototyping, such as processor prototyping, requires a system to simulate the whole ASIC in real logic. To achieve this, custom systems are created using many, huge FPGAs that work as a single device, the device under test. In CERN, the triggering systems for the experiments require not only a large amount of subsystems co-operating in harmony, not only each of these subsystems requires many units, but each unit needs huge

logic resources and enormous input/output capabilities. It is a common sight on this space seeing cards being capable of Terabytes per second of data throughput. All these systems have a common denominator, and that is FPGA devices, that allow a designer to implement the most complex logic circuits. These FPGA devices offer millions of logic gates, tightly coupled with Digital Signal Processors, hundreds of Multi Gigabit Transceivers, serial processors, huge amounts of memory, analog circuitry such as high end ADCs and DACs, Artificial Intelligence engines and many more integrated subsystems, making them the most important block of any system of this kind.

1.3 This design

This design aims to produce a medium capability system, in relevance to the systems that exist, but nevertheless utilizes a novel approach that is starting to build steam. This approach is to use a heterogeneous computing solution, having a processor on one side, and on the other side having large amounts of reconfigurable logic resources, tightly coupled. This technology is used under the form of the Xilinx ZYNQ device. This system will aim in bringing a custom board with plenty of CPU and logic resources, while offering a high degree of connectivity under various interfaces, to raise its capabilities as a signal processor. New approaches can be used in signal processing, basing the implementations on a "normal" CPU, using a full-blown Operating System, while at the same time huge acceleration can be achieved using logic resources in the FPGA.

1.4 Possible uses

The application space of such systems is huge, but this specific system is built having in mind high-energy physics triggering and possible real-time monitoring of the data coming out of the detectors. Having a processor is very handy in using high level applications such as CERN's root. Other uses could be interfacing to other systems, such as high speed DAQ peripherals. Combined with the large connectivity this system offers, the optimized non-MGT I/O for high-speed applications, together with the MGTs, that can interface directly to another system, or even a computer, the USB 3.0 and USB 2.0 and the gigabit Ethernet cover much of what a user would need.

1.5 Other existing similar systems

There exist systems that aim to perform the same task, however most are found under the form of development boards, which are an all-in-one solution and usually lack in Input / Output capabilities, or are usually targeting very high-speed connectivity lacking more lower-speed, easier to interface ports. Example of these boards are the Xilinx evaluation boards, which usually lack low speed connectivity, except one uses an FMC header, or Digilent's Zedboard, that lacks high-speed connectivity. Other similar systems are various SoMs (Systems on Module), that can usually have many uses and various options for connectivity, whether high speed or low speed, however, in the expense of not being a standalone system. In order to use them, one has to install them in a host system, or install them on a breakout board.

1.6 Future implementations

Future implementations of the system presented on this thesis could be systems that move under the same concept, however use the newer Xilinx ZYNQ Ultrascale+ devices, that offer higher density and logic speed and more processor cores, with these cores being more powerful as well. However, one of the most important differences is the inclusion of MGT transceivers in the processor system as well, offering more powerful storage capabilities using a SATA Solid State Drive, instead of an SD card, and native USB 3.0 support, that could enhance the connectivity of the system without using dedicated transceiver ICs, and provide a higher level of abstraction as well.

Chapter 2: FPGA Technology

FPGAs (Field Programmable Gate Arrays), are integrated circuits that offer a (usually) large amount of generic logic resources accompanied by a complex and capable routing complex on the same die. This combination, tied with the fact that the logic resources are not programmed to perform a specific logic function and the routing network is not programmed beforehand, allow for a very powerful device to be realized. The user can program the logic functions and the routing structure "in the field", and as many times as desired. Other ingredients of this successful recipe are several "hard" IPs, such as PLLs (Phased Locked Loops), memory blocks, DSPs (Digital Signal Processors), Gigabit transceivers (SerDes), Memory Controllers, Processor cores, PCIe blocks and much more.

The nature of FPGAs in inherently concurrent, suggesting that logic blocks are independent of each other. Of course, various logic constructs can run under the same clock and function in a serial manner. Under this, one can argue that FPGA computing is not time-constrained (such as CPU computing), but is area-constrained, implying that if one asks for more performance, all that is needed is a bigger FPGA device. Tasks that are efficiently parallelizable can be executed orders of magnitude faster in an FPGA in comparison with a CPU. Of course, one should not expect the same performance in serial tasks, but many algorithms can realize a lot of performance gains by parallelizing them.

FPGAs are identified by their flexibility, which sits between general processing units and ASICs. Changing the configuration of the device takes only milliseconds, and one can only reconfigure certain parts of the device if needed. That means that not only easier debugging and possible firmware updates are possible, but the same device can perform many different tasks on demand, even if the area does not allow for all of these tasks to exist in parallel the same time in the same chip. Another area that FPGAs shine is the I/O capability. A single device can feature many signaling standards, and a huge amount of IO pins. Under this, very wide (arbitrarily) parallel buses, usually limited by the I/O that the device packaging offers, and very fast serial links can be constructed.

2.1 Configurable Logic Blocks

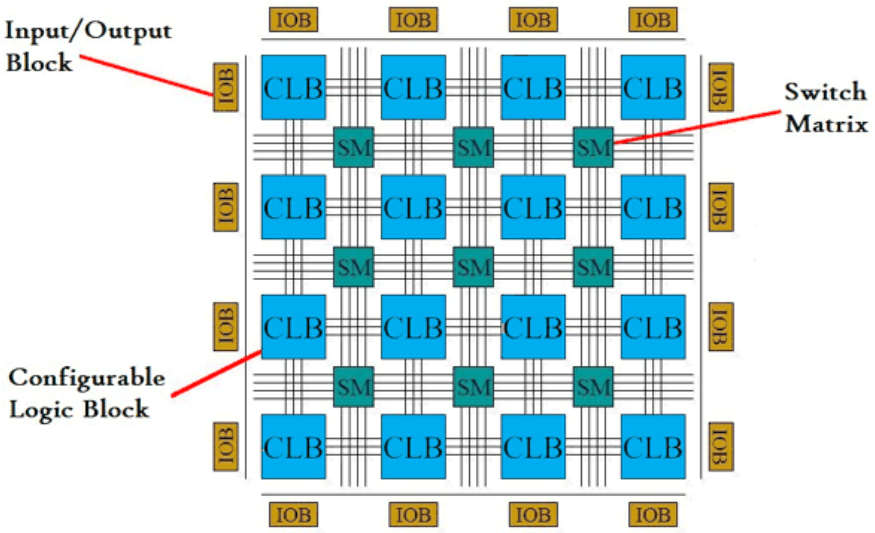


Figure 1. Programmable Logic Architecture

The reconfigurable logic portion of an FPGA consists of the CLBs (Configurable Logic Blocks), Input/Output Buffers, and the routing network. An essential CLB consists of a LUT (Look-Up Table), a Flip-Flop and a multiplexer. A LUT is an SRAM block that contains the truth-table of the specific logic function that is assigned to this CLB. In this way, instead of performing the calculation through gates, the outcome of a possible input combination is predetermined and stored in the SRAM, so the result is known. All state-less logic functions can be represented by a Look-Up Table, and this is the way FPGAs "mimic" logic functions. If a bigger LUT is needed, many CLBs can be combined to obtain the desired function. The Flip-Flop is the smallest storage element possible, storing one bit on its output on the clock edge, and is used to register the result of the logic function. Combining a large number of these Flip-Flops wide registers can be constructed. The multiplexer is essentially a switch, which can select between the inputs to the circuit.

2.2 Routing and clocking network

Interconnecting all these logic resources can be a very challenging task. Responsible for this is the routing network, which consists of a lot of wires and multiplexers. Having a fast routing network but in the same time large enough to not run into congestion is a very important factor to maintain high performance numbers. Often the inability of the routing network to serve a complex design is the reason for timing-closure errors to start appearing, obligating the designer to relax the clocking frequency. A category of signals that are of special handling are the clocking signals, since they have to be noise free, have low jitter, and should be able to cross the whole chip die with the minimum possible delay. For this reason, a special routing network exists especially for them, referred to as the clocking network. The clocking network is split into global clocking network and local clocking entities, making the clocking distribution efficient and clean across the whole device and all processing elements. Special clock buffers exist in the chip as well for this reason.

2.3 FPGA Features

Applications nowadays often require big amounts of memory for the logic to use. FPGAs address this in two ways. The first is to include blocks of memory in the die, scattered between the programmable logic (often referred to as Block Ram). Block ram is multi-kilobit long memory arrays, 8 or more bits wide that usually offer true dual-port capability, and instant access. Although larger FPGAs can contain several megabits of this kind of memory it is usually not enough and including more memory in the same die can eat up space that logic could use. For this reason FPGAs offer DDR memory controllers, that abstract the memory handling, handle the needs of the memory (refresh, etc.), and interface to memory chips that are widely available on the market and of high density, using high speed technologies such as DDR4. In this way, an FPGA can have access to vast amounts of memory, orders of magnitude higher than the GB mark.

A common use case in the computing environment for FPGAs are compute-intensive DSP tasks. Since these tasks can quite often be efficiently parallelized, using an FPGA can be very rewarding performance and/or power wise. For this reason, a special “hard” circuit that is

usually found in FPGAs are the DSP blocks. These blocks have multiple inputs, contain a binary adder and multiplier and flexible output, and since they are not completely reconfigurable have a better performance and power consumption than being implemented on reconfigurable logic. An FPGA can have only a handful of DSPs or multiple thousands of them, and their performance is a large portion of the total GMACS (Giga Multiply-ACcumulates per Second) performance metric of an FPGA device.

Of course, all this outstanding computing performance is nothing if there is no efficient way to input and output data fast enough. This aspect is covered by the MGTs (Multi Gigabit Transceivers). MGTs are "hard" circuits in the FPGA device that can serialize a parallel stream of data and transmit it in rates far exceeding 1 Gbps. The same MGTs are capable of transmitting and receiving in the same time. For receiving, a serial stream of data is fed to the MGT and after sampling of each bit is being parallelized again to form a parallel bus at a much lower frequency. The inverse holds true for transmitting. The MGT circuitry consists of the PCS layer (Physical Coding Sublayer) and the PMA block (Physical Medium Attachment) layer. The PCS layer contains some encoding schemes (8b10, 64b66b, etc.), scrambler and descrambler, link alignment circuitry, and other features, some of which could be implemented by the user in the configurable logic. The PMA layer is essentially the front-end of the transceiver, being responsible for the sampling and the transmission of the data stream into the differential link, CDR (Clock and Data Recovery), eyescan functionality, equalization schemes and more that can't be performed in the configurable logic portion of the device. Data rates for the MGTs can reach above 100 Gbps, and a single device can have more than 100 MGT channels at such high frequencies, making for a very capable means of data transfer.

FPGAs are perfect candidates for parallel computing, but even parallel applications often need a serial processing unit for complementary functions or even control, and reconfigurable logic, while capable, is not the best choice to utilize serial processors. The solution to this comes by including "hard" serial processors in the same die, tightly coupled with the reconfigurable logic part of the device. These processors usually come in the shape of ARM processors, featuring from a single to many cores, dedicated GPU, cache memory and memory controllers, peripherals, and a very fast and wide bus interconnecting the logic with the processor. This combination is very potent, since one overcomes the serial computing restrictions that the FPGA poses, and does not consume programmable logic resources. Under this scheme, an FPGA device can transform to a full-blown computer with impressive computing capabilities.

It is an often sight in the modern FPGA space to see devices running an operating system, and the circuits in the logic appearing as devices in the system. This opens up endless possibilities and offers a new, higher level of abstraction and integration.

2.4 Bleeding edge devices

FPGA devices often include technologies that are not available to the wide market of commercial products. Today, bleeding edge features that are available on FPGA devices, are very high speed serial transceivers, in data rates of 58G per MGT (or 112G for 2 combined MGTs), up to a number of 140 (or 70 for combined) channels per device, using PAM-4 (Pulse Amplitude Modulation) encoding schemes, together with at least 28 lower speed-rate MGTs (32G). Summing up only the MGT connectivity, data rates in excess of 9 Terabits per second can be achieved in one direction, and another 9 Terabits in the other direction, only from a single device. To aid connectivity, PCI-e Gen 5 blocks are provided, 100G and 600G hard Ethernet MACs, and multiple 400G High-Speed Crypto engines. These devices usually offer up to 4 DDR4 memory controllers, with data widths of 256 bits, achieving huge memory data movement capabilities. The offered logic on the devices is huge, exceeding 3 million Look Up Tables and several megabytes of on-chip memory. DSP (Digital Signal Processors) blocks are offered in excess of 14000 per device, making the FPGAs have top signal processing capabilities. On top of this, Artificial Intelligence cores are added, allowing for a fast, low latency, hardware implementation of complex state-of-the art neural network algorithms. All these are often combined with many serial CPU cores and their peripherals, making for a complete SoC (System on Chip) solution. The silicon die area required is huge, and manufacturing such a huge silicon piece without defects would yield a very low production efficiency, so often these devices are manufactured in distinct dies and are then interconnected together, and assembled into a single chip. Every distinct die is referred to by Xilinx as an SLR (Super Logic Region). Chips with up to 4 SLRs exist. Interconnecting SLRs and providing a capable clocking network for the whole chip pose many engineering challenges. Huge amounts of SDRAM memory are included in the chips, using very powerful and fast interconnects, in the order of several Gigabytes. This technology is known as HBM (High Bandwidth Memory), and is achieved by placing memory dies very close to the FPGA die, in the same chip. The signals routing is done through the substrate. Other industry disruptive FPGA technologies are the RFSoC devices, which apart from huge logic resources, MGTs and hard processor cores,

offer a big number of on-chip ADCs and DACs, which offer high sampling rate and precision, making excellent contesters for RF systems.

FPGA devices come from different vendors. The two biggest vendors are Xilinx, capturing almost half of the market share, followed by Intel (formerly Altera), falling just short of Xilinx in sales. Both of these vendors offer a wide variety of devices from low-end to high-end, with similar capabilities along the lines. Microsemi, Lattice and the rest of the vendors capture about 10% of the market share, usually competing in the lower-end and the ultra-low-end of the devices.

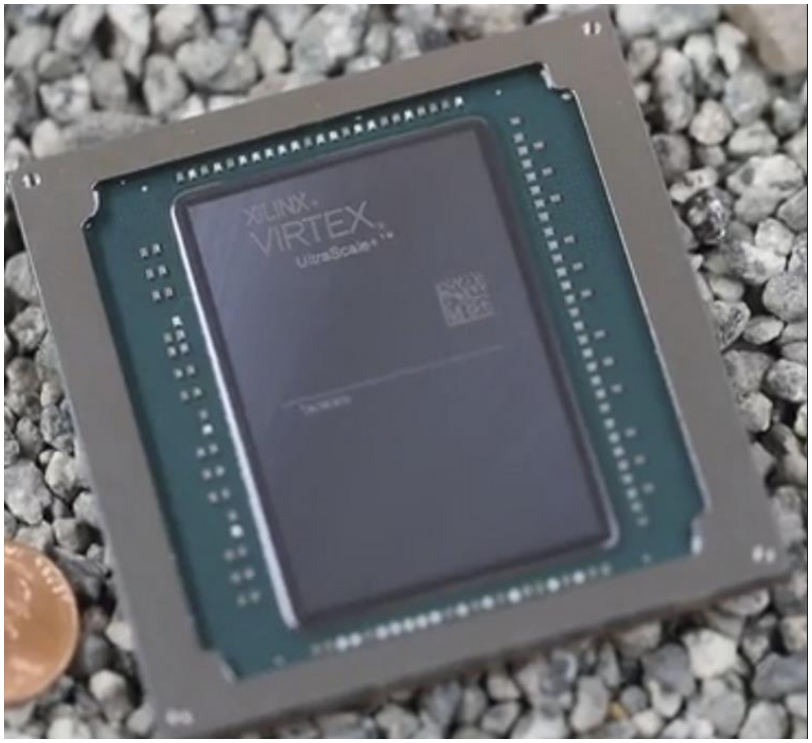


Figure 2. A Xilinx Virtex Ultrascale+ VU19P device. The coin is placed to show how large the silicon area is.

2.5 Hardware Description languages

The amount of logic contained in FPGAs can be huge, and it is a logical result of this to have a way to abstract the design of the circuits contained in the devices. This is done by using Hardware Description Languages (HDL), allowing a designer to describe a logic circuit, its functions and its interfaces using text based description and language constructs. HDLs are usually very strict in syntax and in context and are not to be confused with SPLs (Software

Programming Languages). Circuits constructed using HDLs are inherently parallel and not procedural, unless explicitly defined to. Of course, HDLs are not constrained in use only for FPGA devices, but are used to describe and design all forms of Integrated Circuits. Code written in HDL is used by specialized tools, called synthesizers that transform the code into logic circuits, and then map these circuits on FPGA devices. Usually, but not always, these synthesizers are provided by vendors to be used for their devices.

VHDL (Very high speed integrated circuit Hardware Description Language) is one of the main HDL languages, with the other being Verilog. VHDL was developed in 1983 from the U.S. Department of Defense, as a way to document the behavior of ASICs that supplier companies were including in the equipment. In programming context and syntax, VHDL borrows much form the Ada programming language, used extensively in the time by the U.S. Department of Defense. VHDL is still largely used, and is being continuously updated.

In VHDL, the code is typically typed in a text file, which is used by the synthesizer to construct the logic the code dictates. This step is called compilation (or synthesis). During this step the code is transformed from abstract RTL (Register Transfer Level) to a netlist at the gate level. The step following is the optimization, which can have many objectives, such as speed, area reduction, power consumption etc. The last step before the bitstream generation is the PAR (Place And Route) which maps the synthesis constructs into circuits mapped in a specific device. How the mapping is done, depends on the available logic resources type and quantity each device has, the congestion level of the routing network available, the timing constraints used for the timing closure and more. For example, an operation suited for a DSP will be mapped to a DSP if the specific device targeted contains such hardware blocks, otherwise it will be mapped to regular logic. After the logic placement is done on the device, one can view the populated device, change the placement of logic blocks, and perform post-implementation simulation. The final step is generating the bitstream, which can be loaded to the device. The bitstream contains all the initialization values for the SRAM inside the FPGA.

In VHDL, a design is defined as an entity. In the entity declaration the designer can define the Input and Output ports of the design. In the design components can be included, and a specified hierarchy is used, depending on whether these components are parts of a bigger circuit. If so, these components are instantiated in the architecture body of the entity. Every circuit has ports,

which define its interface. The ports of a circuit are connected with ports of other circuits using electrical connections, what VHDL refers to as a signal.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
entity design_1_wrapper is
  port (
    DDR_addr : inout STD_LOGIC_VECTOR ( 14 downto 0 );
    DDR_ba : inout STD_LOGIC_VECTOR ( 2 downto 0 );
    DDR_cas_n : inout STD_LOGIC;
    DDR_ck_n : inout STD_LOGIC;
    DDR_ck_p : inout STD_LOGIC;
    DDR_cke : inout STD_LOGIC;
    DDR_cs_n : inout STD_LOGIC;
    DDR_dm : inout STD_LOGIC_VECTOR ( 3 downto 0 );
    DDR_dq : inout STD_LOGIC_VECTOR ( 31 downto 0 );
    DDR_dqs_n : inout STD_LOGIC_VECTOR ( 3 downto 0 );
    DDR_dqs_p : inout STD_LOGIC_VECTOR ( 3 downto 0 );
    DDR_odt : inout STD_LOGIC;
    DDR_ras_n : inout STD_LOGIC;
    DDR_reset_n : inout STD_LOGIC;
    DDR_we_n : inout STD_LOGIC;
    FIXED_IO_ddr_vrn : inout STD_LOGIC;
    FIXED_IO_ddr_vrp : inout STD_LOGIC;
    FIXED_IO_mio : inout STD_LOGIC_VECTOR ( 53 downto 0 );
    FIXED_IO_ps_clk : inout STD_LOGIC;
    FIXED_IO_ps_porb : inout STD_LOGIC;
    FIXED_IO_ps_srstb : inout STD_LOGIC
  );
end design_1_wrapper;
```

Figure 3. VHDL code defining the ARM processor on a ZYNQ device, showing the exposed RAM interface.

VHDL is an inherently parallel language, but sequential constructs can be defined as well. These constructs are defined as processes. Every process is executed concurrently with every other process on the design, and can be completely independent, in the sense that different clocks can be used for each process.

As the concept of libraries exists in sequential languages, this concept in VHDL is referred to as Package. VHDL can also parse data from text files, which is usually useful for simulation and memory configuration files.

2.6 Simulation

Circuits designed using VHDL can be simulated prior to implementation, using specific VHDL files, containing clocking and I/O for the circuits, referred to as “stimulus”. The circuit that is being simulated is referred to as the UUT (Unit Under test). Simulating circuits can be very useful in validating without the need to have the actual device and expensive testing equipment. Simulation can happen on other phases of the design as well, such as post-synthesis or post-implementation.

2.7 High Level Synthesis

VHDL is a very strict and inherently parallel language, which requires specialized engineers. However, the lack of engineers and the increased interest in FPGA devices for various applications, has risen demand for engineers, and a movement of software engineers to FPGA applications is observed. Furthermore, even for specialized hardware engineers, it can be very time consuming to write HDL code. In order to make the transition viable and the time to market faster, a middle solution has been introduced. This solution offers a higher abstract for VHDL, referred to as HLS (High Level Synthesis). HLS code is usually written in C language, and a wide variety of preprocessor directives can be used to aid the tool in producing the correct circuits. C code is translated in VHDL, and then synthesized to form circuits. Although HLS code offers nowhere near the optimization that can be achieved by RTL, it offers a very fast way to convert algorithms into circuits.

Chapter 3: Printed Circuit Boards

Printed Circuit Boards (PCBs) are found in almost all electronics products, and their use is to provide the mechanical support and the electrical connections between the components of an electronics system. PCBs are made by copper sheets laminated on to a non-conductive substrate, and on top of the copper a protection mask is painted. The copper on the PCB gets etched in order to create tracks, pads and other features, by a method using light and chemicals, called photo chemical etching. Then, components are placed on the PCB by the method of soldering.

The advantage with PCBs is that the manufacturing can be automated, something that helps their dominance as a means of electrical connection between components. However, this requires PCB layout by specialized engineers and perhaps several re-spins of prototypes until a satisfying solution is reached. After this, one can order production runs of as many pieces as he wishes from specialized PCB manufacturing houses. PCB production costs face a steep decline in price per piece as production quantity increases, since the main initial cost driver of the process is the tooling needed to produce each design. When the tooling is verified, only factory production-line time adds up to the cost, and raw materials.

3.1 PCB Technology and manufacturing

There are several technologies that categorize a PCB. The main is the layer count. Since the top and the bottom of the board are the only places that components can be placed on, it is reasonable that the more the components that are placed, the less space is left for routing traces. So, PCB engineers began looking at another direction to increase the density of their designs. By adding more copper layers internally, a lot of routing space is being gained while close to none physical space is being consumed, because the thickness increase is minimal. However, adding more internal layers increases the manufacturing cost by a huge amount, especially the one-time tooling cost. Most PCBs are single-layer (meaning that there is copper only on one side) or doubled sided (meaning that there is copper on the top and the bottom of the PCB). By adding an internal pair of layers one creates a 4-layer PCB. PCB layer count can reach up to

more than 40 layers for exotic applications. However, common complex PCBs such as computer mainboards, mobile phones, etc. use between 6 to 16 layers.

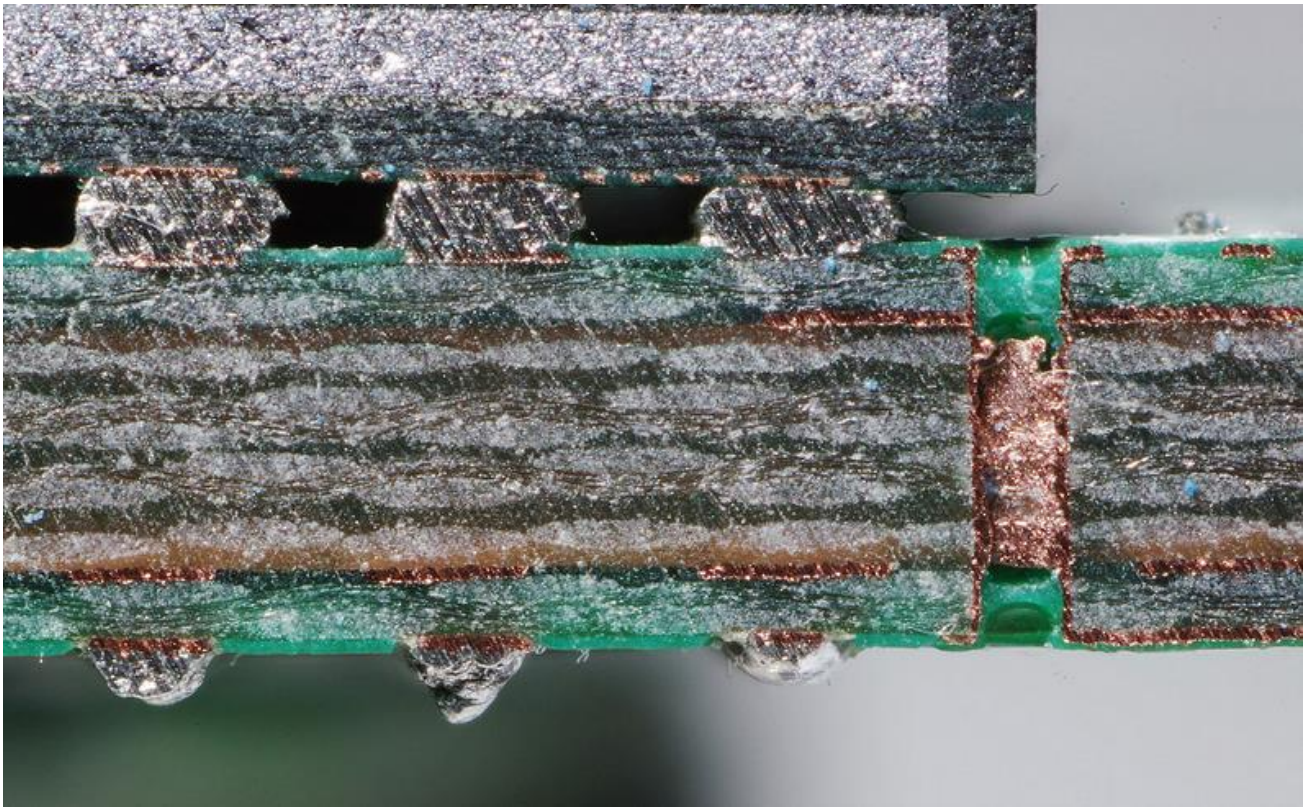


Figure 4. Cut through a multi-layered PCB. Visible is the copper on different layers, as well as the through-hole via on the right.

Multiple layers are useless if there is no means to pass connectivity from one layer to the other. This is achieved using vias. Vias are holes in the PCB that are plated conductive using electrical deposition processes. With vias one can travel from any layer to any other layer, with the expense being the space that the hole itself accommodates, and to a lesser degree, compromised structural integrity.

The etching of the copper is usually done using photoresist coating on the PCB, usually sensitive to UV light. The PCB "artwork" (term referring to the copper layout) is then projected on the photoresist. This process dissolves the photoresist from the spots that the copper has to be removed. After this, the PCB gets submerged in an etching solution, dissolving the copper in the places that the photoresist was gone. In this way, the copper gets split in partitions, which

form tracks, planes, pads etc. In the case of a multi-layer PCB each copper layer is first etched and then “sandwiched” together.

The finish on the two outer-layers of copper plays a vital role on the solderability of the board and the electrical resistance of the connections, as well as their longevity under harsh conditions or if no solder is applied on them. A designer may choose to not finish the pads and the exposed copper of the PCB, that being the cheapest solution. However in quality products, several finishes are available to the designer such as HASL (Hot Air Solder Level), Lead-Free HASL, Gold immersion, Tin immersion, ENIG (Electroless Nickel Immersion Gold), ENEPIG (Electroless Nickel Electroless Palladium Immersion Gold) and more.

In order to avoid corrosion and electrical shorts during soldering, a special thin lacquer-like polymer is applied on the top and the bottom layer of the board. Since it protects against soldering where it is not supposed to be, it is usually called “solder mask”. Solder mask is usually green, but can vary in color as well.

On top of the PCBs text or textures can be printed, in what is called the silkscreen. The silkscreen is useful in placing product codes and serial numbers on the PCB, manufacturer information, directions for usage and more. However, one of the most useful aspects of the silkscreen is that component placement is numbered (such as R60, for the 60th resistor on the board), so the assembly house knows where to place each component visually. Something that is even more important, is that the silkscreen has information about the polarity and the rotation of placement of sensitive components (such as a BGA chip or a diode).

Figure 5. PCB silkscreen and textures showing the polarity of the diode and the electrolytic capacitors.

After the manufacturing of the PCB, if the designer requests it, testing, whether extended or not, can take place on the board. The most common form of PCB testing is visual inspection. This can be done by humans but nowadays is mostly done by computers using automated programs embedded in the production line. A more extended method of testing, which tests the inner layers as well in the case of a multi-layered board is that of the flying probe. In this process, a very fast robot that is able to move electrical probes tests the PCB pads for connectivity, as should be according to the netlist provided to the manufacturer. This allows for much faster and precise testing. Up to this point testing is non-destructive, meaning that the same PCB tested can go ahead to be assembled and become functional. An even more exhaustive form of quality testing is performing cuts on the PCB, a form of destructive testing. With this method, the manufacturer can verify if the specifications for the PCB are met, that is the width of the conductive plating of the vias and more technical parameters.

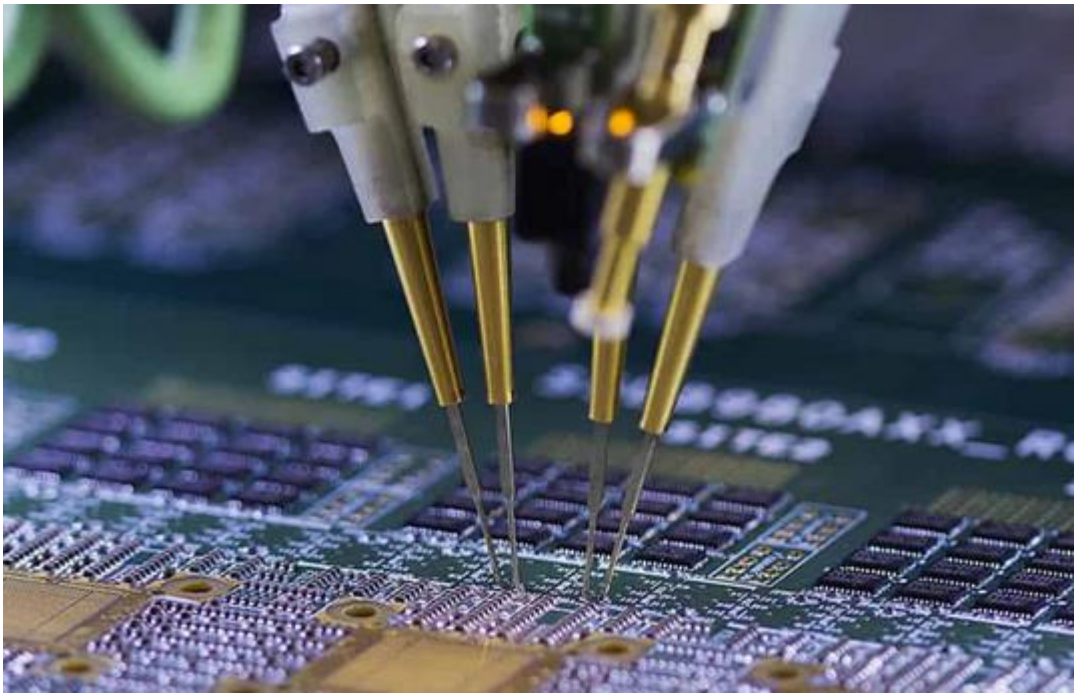


Figure 6. Flying probe test of a semi-populated PCB.

3.2 PCB Assembly

There are two main ways to mount components on a PCB. Through-hole mounting and surface mounting. Through hole mounting, as the name suggest, is a method of mounting the components, using their leads, through holes on the PCB. Most leads on a component form a functioning electrical connection, and they do not usually have only mounting uses. This method of mounting allows for easy hand mounting, and is usually served by single mounted PCBs if there are only though-hole components. This mounting method is very common, especially in simple and low-cost assemblies. The advantages are that the soldering can be done easily without special equipment other than a soldering iron. The disadvantage is that though-hole components are usually physically larger, meaning that they do not allow for dense designs. The way these components are soldered is either manually with a soldering iron, or with wave soldering. Wave soldering is using a bath of molten solder, in which the side of the board that is to be assembled is placed, and the solder soaks the connections and forms a solid joint when the board has cooled down enough.

The second component mounting method is the SMT (Surface-Mount Technology) assembly. In this method the components do not have long leads that can pass through holes, but rather short leads that are soldered on the same layer as the component. Today SMT has the leading role in manufacturing technologies, particularly for two reasons. The first one is that by using this method, the component packaging is much smaller in physical dimension which allows for much denser designs. The second, and perhaps equally if not more important, is that SMT can be fully automated, from component placement by specialized robots, to soldering. This allows for much lower cost of production and much higher quality due to its automated nature. Soldering SMT components is usually done with the method of reflow soldering. This happens by first applying a solder paste on the pads of the component to be placed and then placing the component. The next step is to heat the board high enough to melt the solder paste and soak the connections. After the PCB has cooled down, solid connections are formed. This is usually done by placing the board in a furnace and raising the temperature high enough, until the solder paste is molten, and then let the board cool down.

Most PCBs manufactured today are using both the though-hole and the SMT processes, with the first being used to place bulky electrolytic capacitors, transistors, connectors etc., and the latter being used to place the bulk of the components on the board such as chips, decoupling capacitors, resistors etc.

This reflow process however may prove dangerous for sensitive components that either have their temperature specifications exceeded, or a too high or too slow slope for heating cooling was used. This may cause cracking of the package (and lead to cease of functioning or moisture intruding the package) by the uneven rates of expansion on a given chip. There are many kind of damages that can happen during reflow soldering, and often lead sooner or later to a faulty chip. In order to mitigate this problem, chip manufactures provide reflow curves; that is curves that specify what the rates of heating/cooling and the temperatures in each phase of the reflow process. However, since not every single chip on the PCB has the same specifications, the assembler decides on the final temperature curve that will be followed (after advising the manufacturer's specification) since the assembly house knows the behavior of their soldering equipment the best. It is not uncommon from soldering houses to ask for "test rounds", in order to test their reflow temperature curve on a design.

The solder can be whether leaded (that is, containing lead), or lead-free. Since lead is a hazardous substance, its use is prohibited by many directives. However the advantage of leaded-solder is that of the lower melting temperature, meaning that one does not have to thermally stress the components by a big amount during soldering, and a later rework on the board is made easier. Unleaded solder has a considerably higher melting point, but is much more environmentally friendly than its leaded counterpart.

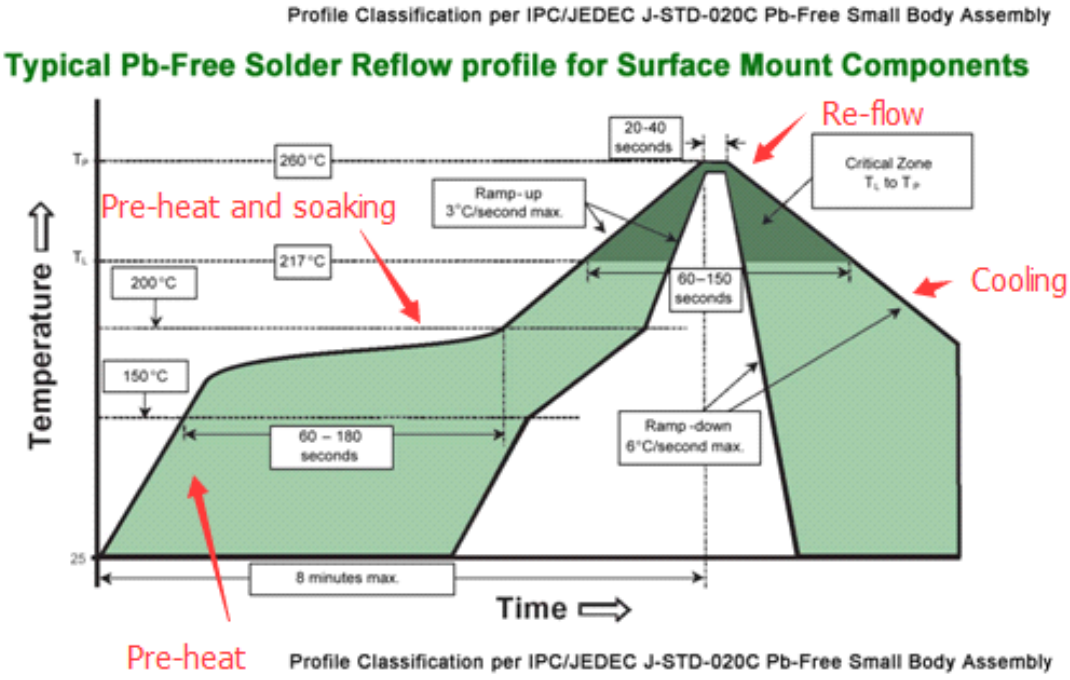


Figure 7. Typical reflow curve.

3.3 Technical specifications

Layer count is a very important factor to categorize the technological effort of each PCB, but not the only one. Technical specifications such as minimum trace width, minimum trace spacing, minimum via size, via types, dielectric materials and more are of equal importance in aiding the PCB designer to achieve the maximum possible design density and the best signal integrity for high speed or high accuracy applications. Some of the most important PCB technical aspects are listed below.

- Material

  There is a wide selection of available materials, such as FR-2 (phenolic cotton paper), FR-3 (cotton paper and epoxy), FR-4 (woven glass and epoxy) and more. Usually not only the material is of interest, but the structure of the material as well.

- Board thickness

  Although board thickness is a function of the materials and the number of the layers it is composed from, it is usually an important technical aspect since various specifications do not allow an arbitrarily large thickness.

- Outer and inner copper weight

  With the term copper weight, in PCB manufacturing one refers to how thick the copper layer is. This number is usually different for internal and external layers, and can differ in between internal layers as well. It is important since it is a factor determining the thickness of the final PCB, but has functional properties as well. For example, in order to carry large amounts of current in a power layer, a thicker copper layer is needed than in an inherent signal layer.

- Via hole size

  Usually a term referring to the minimum via hole size that is supported by the manufacturer, it is a determining factor for the max achievable design density. Hole diameters down to 0.2mm are commonly supported, with many PCB manufacturers willing to go even a bit lower for an added cost. Even lower hole diameters are supported by specialized manufacturers, though since no drill bits of this size exist, this technology requires the use of a laser to open up the holes using an evaporation technique, and this adds up to the cost a lot. Furthermore, electro-plating the holes in a later stage in order to become conductive becomes more difficult with decreasing hole

diameter. Another concern with vias, is that in the multi-gigahertz frequency range, a via can act as an antenna. Take for example a through hole via, travelling all the board from layer 1 to layer 16. If the signal starts its path from layer 1 and uses the via to move to layer 4, then the rest of the via's height (layer 4 to layer 16) is useless, and if the wavelength of the signal is comparable to this length, the via will act as an antenna. This will cause impedance mismatching of the line, signal distortion, as well as radiated noise. A common way to mitigate this problem is to "back-drill" the unused via portion, that had been plated during the plating process. This a very useful technique considering high-speed signals, albeit very costly.

- Minimum annular ring

Again a specification of vias, this time referring to the area between the edge of the drilled via and the copper pad associated with that hole. The greater the width of an annular ring, the greater the copper connection around the drilled via is. The smaller the annular ring, the more density it allows for, especially under devices such as BGA chips of a small pitch.

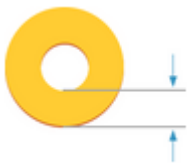


Figure 8. Minimum annular ring.

However, since drilling is not infinitely accurate, the annular ring cannot be very small since the drilling may cut the ring due to inaccuracies.

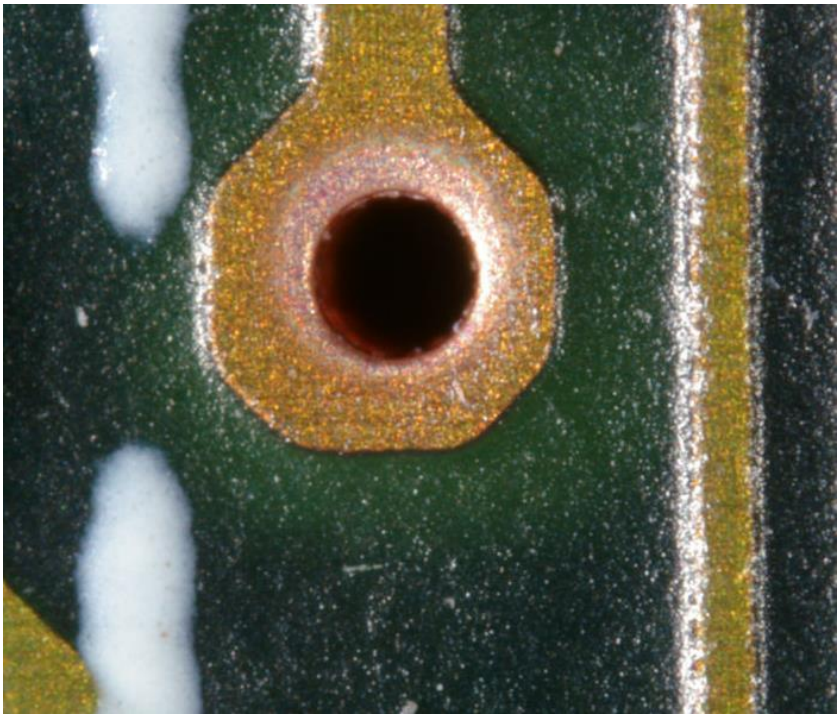


Figure 9. A via that is slightly mis-registered.

- Clearance

Several types of clearance specifications exist, such as Via to Via clearance, Pad to Pad clearance, Via to track clearance, track to track clearance and more. These specifications are critical to the density of the design, since especially track to track and via to track clearances can become prohibitive in the fan out of a big BGA device, in case they are not small enough.

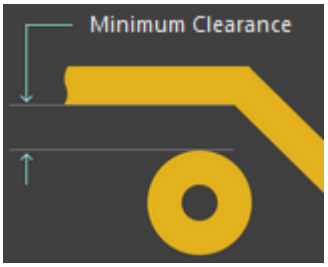


Figure 10. Pad to trace clearance.

- Trace width

  Minimum trace width is very critical, since if it is large enough may prevent the fan-out of a BGA device, however if it is small enough may allow even for two traces to be routed between two BGA balls. It is a limitation coming from the UV light exposure in the etching process. Usually manufacturers are willing to do smaller than standard trace widths for an increased price. The drive for increasing price is because the rate of defect boards during production goes up considerably.

- Solder mask

  Solder mask has various specifications, such as different color and dielectric constant, which comes into play when calculating the impedance of a trace. However other specifications exist as well, such as minimum width and minimum opening around a pad to avoid mis-registration.

- Silkscreen

  Different specifications exist in printing silkscreen, such as minimum line width (which is important for font size or texture details), minimum distance from a pad (since silkscreen on a pad might prevent solderability of this joint) and more.

Figure 11. Textures printed on silkscreen. The green layer is the solder mask.

3.4 Signal integrity

When designing a high speed design, perhaps the most important aspect is signal integrity, and designer effort comes into play in signal integrity in two aspects. First, by minimizing noise and crosstalk, which is mostly a layout issue, and secondly, by providing traces of correct impedance to the various signals on the PCB in order to avoid reflections, distortion etc.

Impedance controlling of traces can be a tedious process. It involves various mathematical calculations approximating the problem, in which many variables exist such as trace width, space between the lines of a differential pair, dielectric constants of substrates, distances between reference planes, etc. So this process is basically iterative, whether changing some variables and solving for impedance, or defining impedance and solving for a parameter. However, the results have to be supported from the PCB's manufacturer. For example, in order to achieve a specified impedance, the calculations may ask for a trace width that is too small to be supported by the manufacturer. Then, one can vary the dielectric constant of the substrate, supposed that the manufacturer supports the required dielectric.

Except for the engineering difficulty it poses, impedance controlling can be a very costly process. Usually the required specifications for the PCB traces or the dielectrics needed to achieve the desired impedance can be high technology and low volume, hence driving the cost strongly upwards. On the other hand, a common pitfall for PCB engineers is over-engineering the impedance specifications of the board, leading to very costly solutions, for minimal gain on what would be an otherwise perfectly working system.

The characteristic impedance of a transmission line is defined as the ratio of the amplitudes of voltage and current of a single electrical wave propagating along the line. Supposing that the length of the line is infinite, so no reflections occur, the characteristic impedance could be defined as the input impedance of the line. Since the line is not infinite, one can terminate the line with a termination resistance of the same value as the characteristic impedance of the line. In this case, the line electrically looks like it was infinite in length, and signal reflections are eliminated.

There are several types of transmission lines for which one can use existing formulas to model the characteristic impedance. The most common perhaps is the microstrip transmission line, which is composed from a conductor and a ground plane, and between them exists the dielectric material. The air above the conductor acts as dielectric as well. This is usually useful for single-ended traces in the outer layers of the PCB.

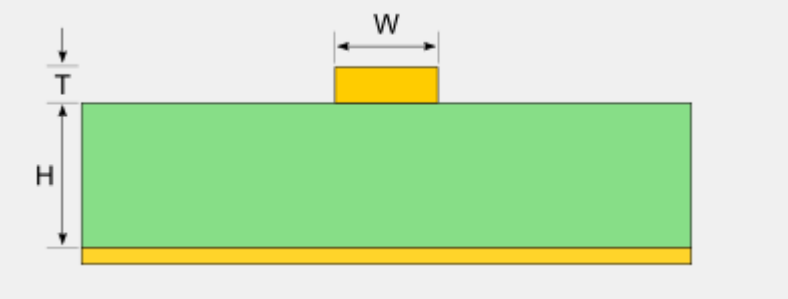


Figure 12. Microstrip transmission line.

There are several models that approximate the characteristic impedance of the microstrip transmission line. A good approximation documented by IPC is summarized below [1].

$$Z_{0,surf} = \frac{\eta_0}{2\sqrt{2\pi}\sqrt{\eta_{r,eff}+1}} ln\left\{1 + 4\frac{h}{w'}\left[4\left(\frac{14\varepsilon_{r,eff}+8}{11\varepsilon_{r,eff}}\right)\frac{h}{w'} + \sqrt{16\left(\frac{14\varepsilon_{r,eff}+8}{11\varepsilon_{r,eff}}\right)^2\left(\frac{h}{w'}\right)^2 + \frac{\varepsilon_{r,eff}+1}{2\varepsilon_{r,eff}}\pi^2}\right]\right\}$$

Where n0 is the impedance of free space. The impedance of free space is otherwise abbreviated as Zo. $Zo = \mu_0 c_0 = \pi \times 119.9168932\ \Omega$. W' is the corrected line width,

$$w' = w + \frac{t}{\pi} ln\left\{\frac{4e}{\sqrt{\left(\frac{t}{h}\right)^2 + \left(\frac{1}{w\pi + 1.1t\pi}\right)^2}}\right\}\left(\frac{\varepsilon_{r,eff}+1}{2\varepsilon_{r,eff}}\right)$$

and for the effective dielectric constant, a discontinuity exists at w/h = 1, where:

$$\varepsilon_{r,eff} = \left\{\frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left\{\sqrt{\frac{w}{w+12h}} + 0.04\left(1 - \frac{w}{h}\right)^2\right\} \quad \frac{w}{h} < 1\right\}$$

$$\varepsilon_{r,eff} = \left\{\frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\sqrt{\frac{w}{w+12h}} \quad \frac{w}{h} \geq 1\right\}$$

Another common type of transmission line is the symmetric stripline, in which now the air above the conductor is exchanged with a ground plane, and between them exists dielectric material as well. The distance between the two planes and the conductor is the same, as the

name symmetric suggests. This type of transmission line is useful for single-ended signals in the inner layers of the PCB.

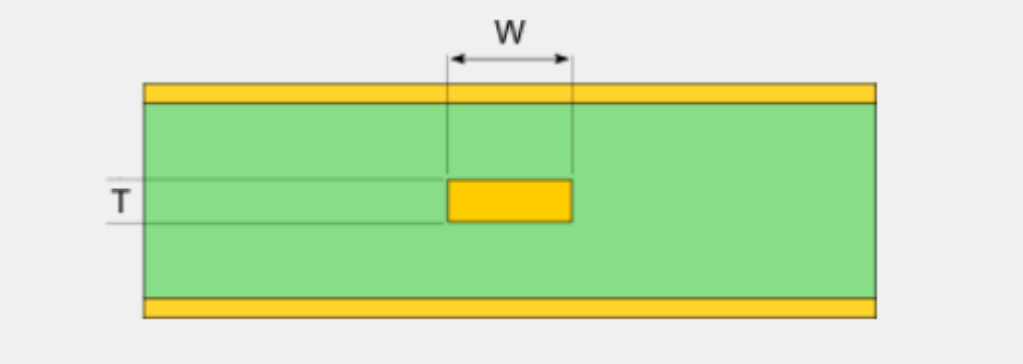


Figure 13. Symmetric stripline conductor.

Again, an approximation from IPC for the modelling of the symmetric stripline is listed below [1]. Note that there exist two distinct cases, one for the "narrow signal" conductor and one for the "wide signal" conductor.

For the narrow signal conductor (w/b < 0.35):

$$Z_{0,SS,t-2} = \frac{60}{\sqrt{\varepsilon_r}} ln\left(\frac{4b}{\pi D}\right)$$

Where b = 2h +t and

$$D = \frac{w}{2}\left\{1 + \frac{1}{\pi w}\left[1 + ln\left(\frac{4\pi w}{t}\right)\right] + 0.551\left(\frac{t}{w}\right)^2\right\}$$

While for the wide signal conductor (w/b >= 0.35), the impedance is approximated as follows:

$$Z_{0,ss,w-2} = \frac{94.15}{\left(\frac{w/b}{1-t/b} + \frac{\theta}{\pi}\right)}$$

Where:

$$\theta = \left(\frac{2b}{b-1}\right) ln\left(\frac{2b-1}{b-1}\right) - \left(\frac{t}{b-1}\right) ln\left[\frac{2bt-t^2}{(b-t)^2}\right]$$

For differential signals, as is the case with many high speed signals, the two above cases turn to edge coupled microstrip and edge coupled stripline respectively. Both of these cases are visible below.

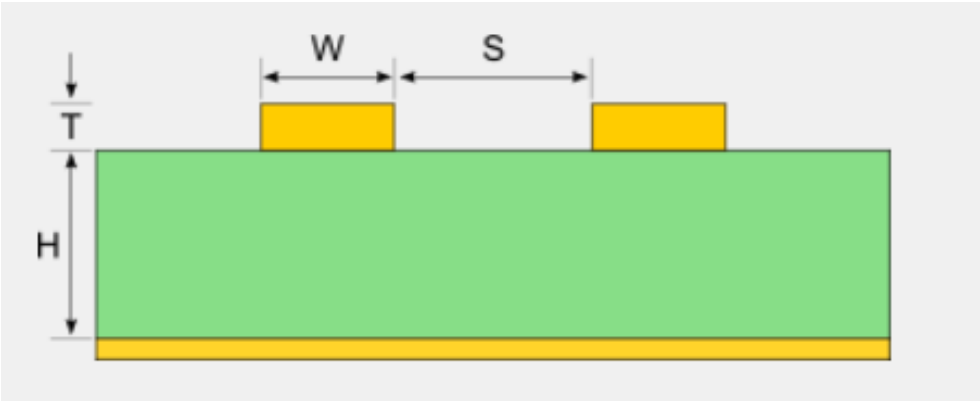


Figure 14. Edge coupled microstrip line.

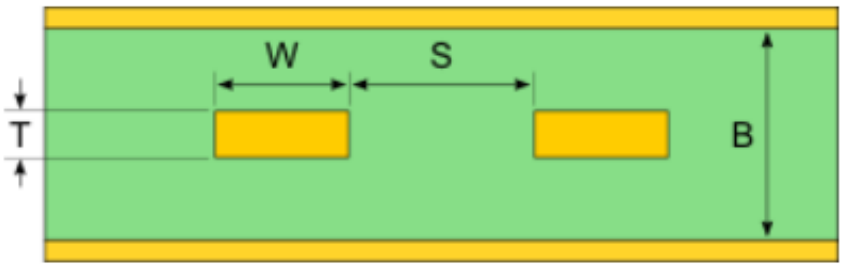


Figure 15. Edge coupled stripline.

Of importance is of course the case where a conductor lays between two planes, however the distance of the conductor from the two planes is not the same. This case of transmission line is referred to as the asymmetric stripline.

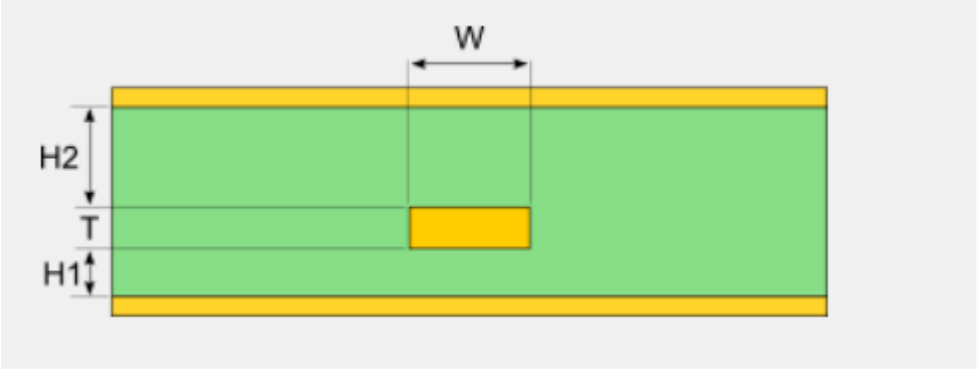


Figure 16. Asymmetric stripline.

As previously stated, the dielectric used can affect the impedance of the conductors and plays a primary role in the formulas approximating the characteristic impedance. However, the effect of the dielectrics is dynamic, meaning that it depends on frequency, and, not only that, but dielectrics suffer from a higher loss as the frequency increases. So dielectric selection gets even trickier. On top of that, since the dielectrics have a defined structure, the angle of this structure in respect to the conductors can have adverse effects on the impedance calculated and the loss due to the dielectrics.

Figure 17. Substrate structure.

In the context of high speed signals, not only impedance matters, but the propagation delay of each transmission line as well, which on parallel buses may cause skew and corrupt data. A common technique to mitigate such problems is to increase the length of the conductors to that of the longest conductor in the same bus, usually using serpentine routing. Serpentine routing is a process of changing the shape of a conductor, in order to increase its length.

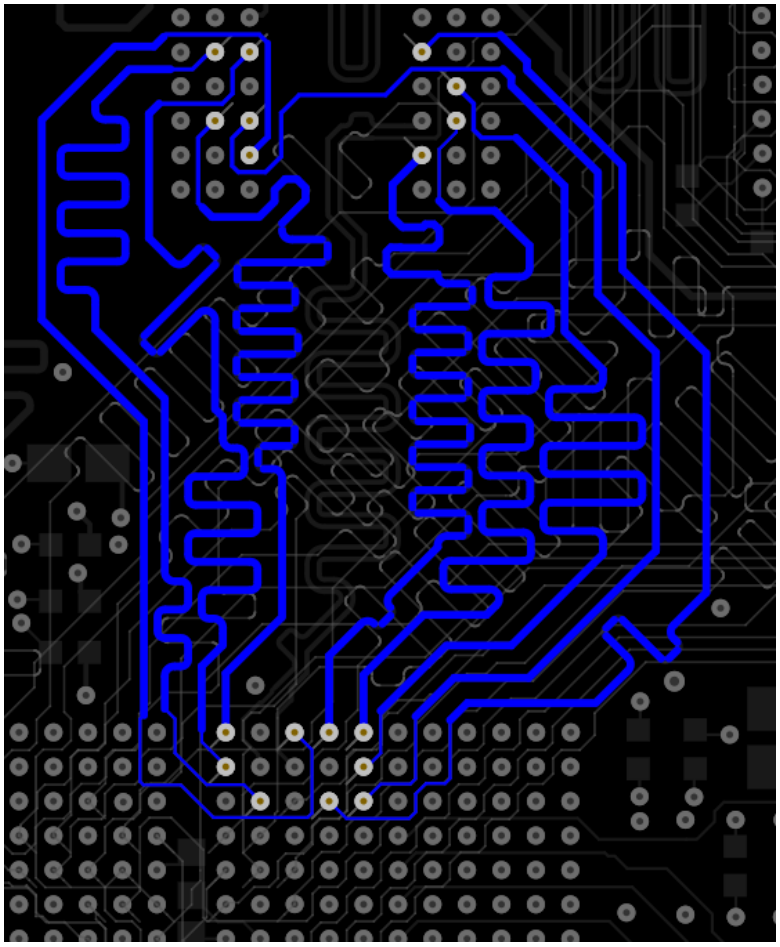


Figure 18. Serpentine routed single - ended signals, matching the delay of the longest line (on the left).

The same technique can be used to delay-match differential lines as well, but in this case special care has to be taken in order to not induce any in-pair skew.

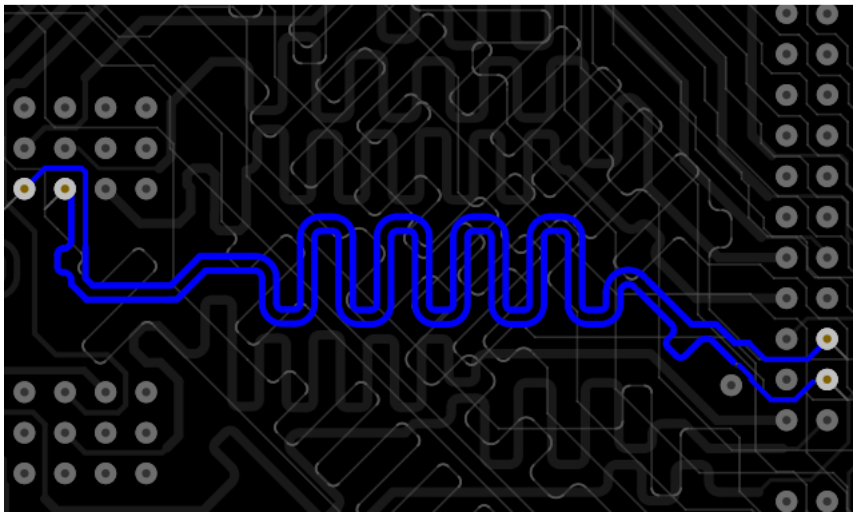


Figure 19. Serpentine routing on a differential pair. Notice the little bumps on the shortest line in order to mitigate skew induced by the serpentine routing process.

While serpentine routing can prove very useful, it has its disadvantages, which a designer has to be fully aware of. The most important is that if sharp turns are used, widening of the pulse's energy spectrum can occur. This is because in the inner side of a sharp turn the signal has to travel a shorter distance, so takes the turn faster than the part of the signal that travels in the outside radius of the turn. After the turn, the pulse edge appears widened in time. Another problem that may arise is that of crosstalk, which can cause transfer of part of the pulse's energy to the next serpentine segment. This can as well cause broadening of the pulse's density in time. A solution to these problems can be mitered serpentine routing which avoids direct 90 degree turns, as well as rounded turns which can help to smoothen the direction change. Furthermore, the designer should leave adequate spacing between the serpentine's segments.

3.5 Design methods

In the earliest stages of electronics, PCB design would be done by hand. Designers would draw the traces on a piece of paper, and later this would become the mask for the PCB etching process. Today though, with high density designs, consisting of multiple layers, thousands of tracks and vias and traces trimmed down to the mil, this is not possible. The advent of modern PCBs has been enabled to a huge degree by the advent of CAD (Computer Aided Design).

There is a large offering of CAD programs today, and they supply the PCB designer with very powerful tools. CAD programs can be used to capture the schematics, transfer the connectivity to a PCB, design the component layout and the traces on the PCB and generate the manufacturing files. The designer specifies an extended set of rules, electrical and mechanical, that the PCB has to obey. Then with the use of the DRC (Design Rule Checker), the CAD tool reports if these rules are obeyed, and if not, where there exist violations. Not only 2D visual capabilities are offered by modern EDA tools, but 3D as well, allowing the design of a complete solution, including cases, connectors, etc.

Many CAD programs offer tools such as auto-routers, which automatically route the board and follow the design constraints, a feature that might prove useful in simpler designs. However, these tools are not capable enough to cope with complex designs, especially if high-speed signals are included and signal integrity is an issue.

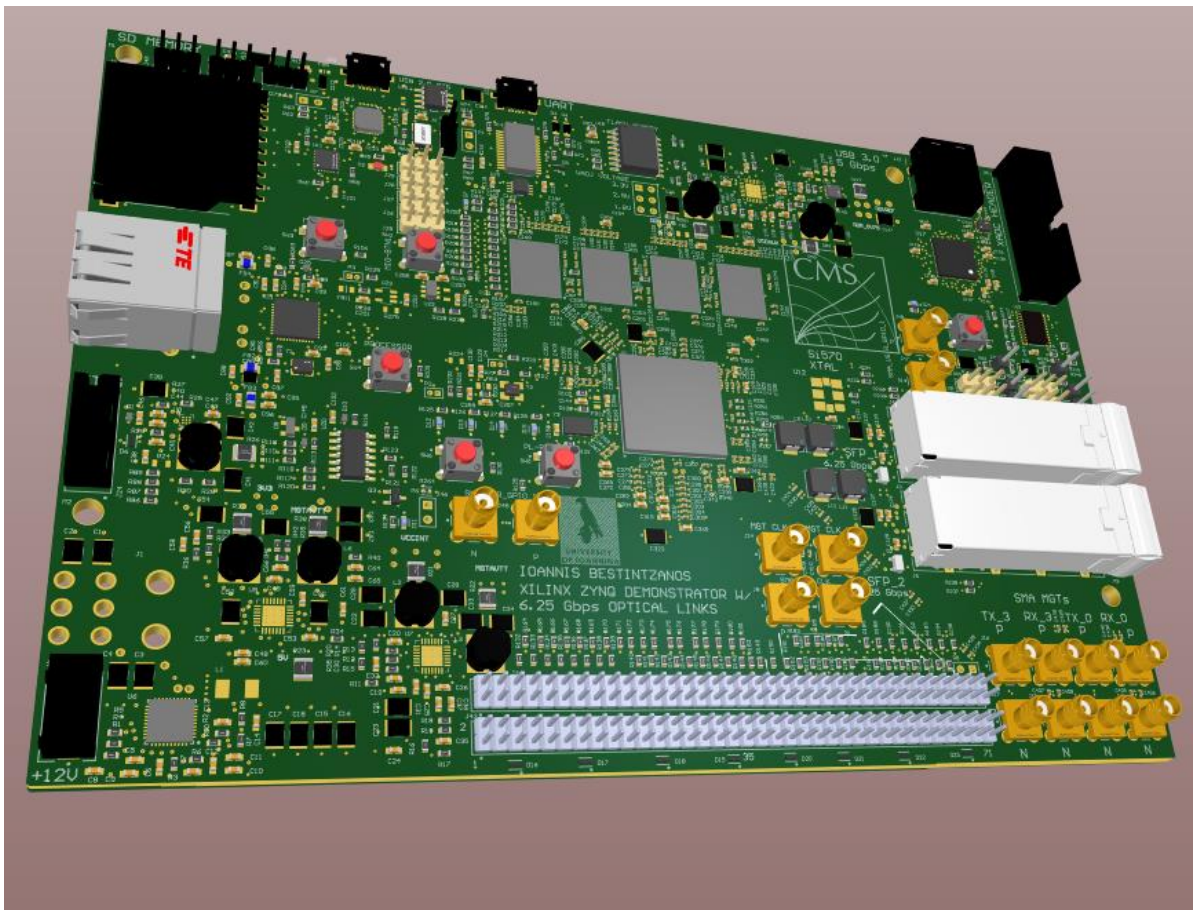


Figure 20. 3D view of the board in Altium Designer.

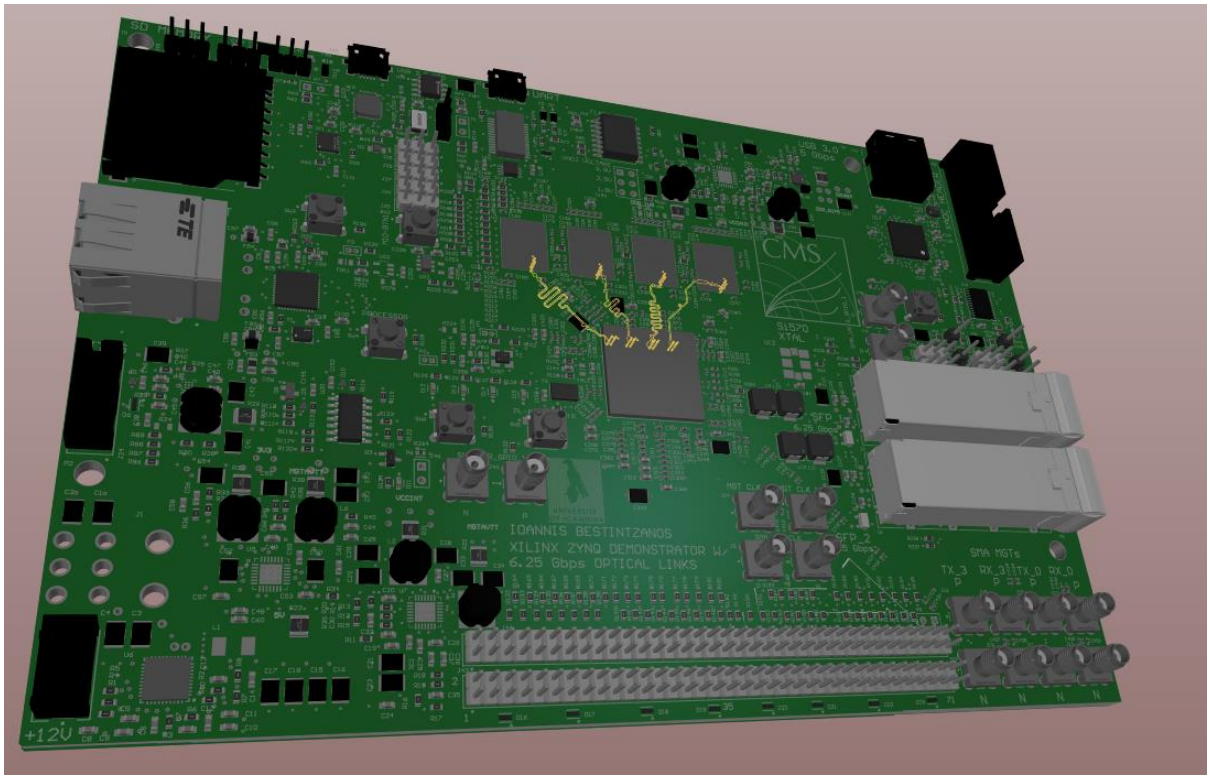


Figure 21. 3D view of the board, with the clock of the byte lanes of the RAM highlighted.

3.6 This PCB

The dimensions of this board are 191.2 mm x 131.88mm. The board features 8 copper layers, which makes routing quite challenging for high density, high speed signals (such as RAM). The final resulting thickness is 1.55mm. The copper thickness for inner and outer layers was both chosen to be 18 μm (or ½ oz). The solder mask color was chosen as green and the silkscreen color as white.

The technological specifications of the PCB are as below (minimum):

- Outer layer trackwidth : 0.100 mm
- Outer layer isolation distance: 0.100 mm
- Outer layer annular ring : 0.100 mm
- Inner layer trackwidth : 0.100 mm

- Inner layer isolation distance : 0.100 mm
- Inner layer annular ring : 0.125 mm
- Smallest final hole : 0.10 mm

These specifications are needed, otherwise several parts of the board (such as BGA devices) could not be fanned out.

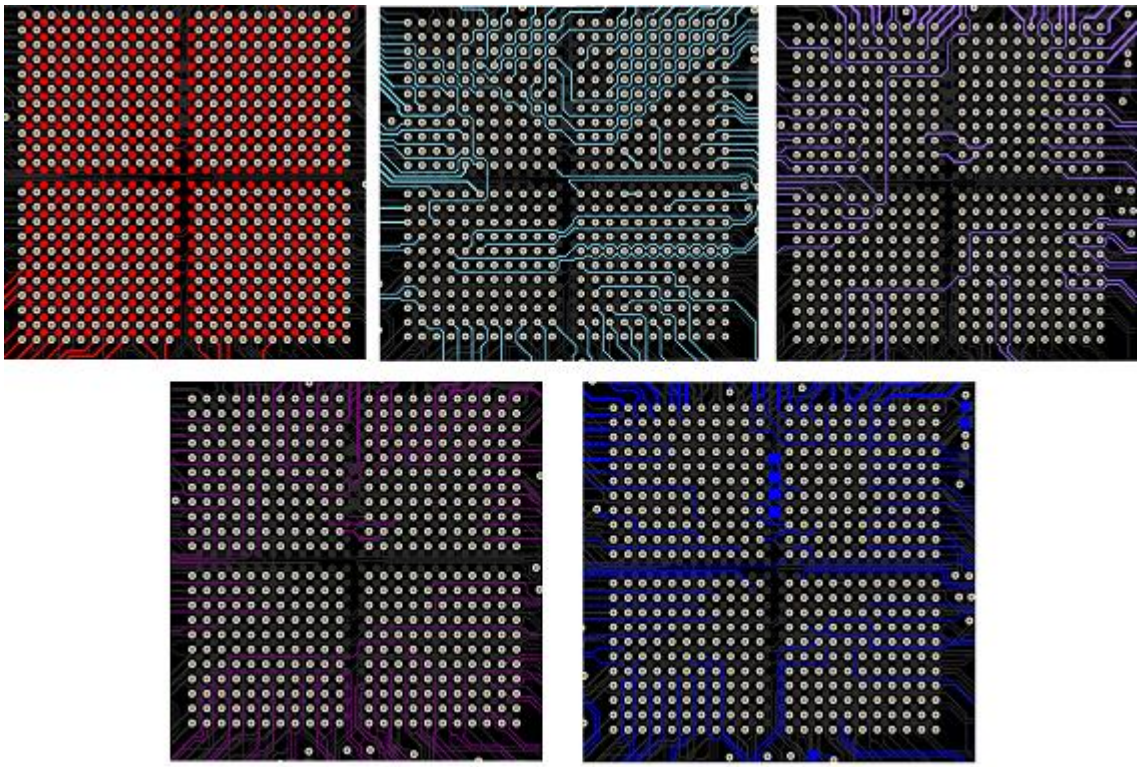


Figure 22. Fanout of the ZYNQ device in the 5 signal layers.

For the base material of the PCB, ISOLA IS400 was used [2]. IS400 is a high performance product, featuring excellent thermal reliability, intended for the Automotive & Transportation market, and offers a good high frequency performance. For the prepreg (the substrate between two copper layers that is pre-impregnated in resin) the 2116 structure was used. 2116 structure has a resin content of 53%, a thickness of 0.119mm and a dielectric constant (Dk)/dissipation factor (Df) [3] of:

Dielectric Constant (DK) / Dissipation Factor (DF)					
100 MHz	500 MHz	1 Ghz	2 GHz	5 Ghz	10 Ghz
4.30 0.0145	4.21 0.0163	4.20 0.0186	4.18 0.0194	4.13 0.0225	4.12 0.0226

Figure 23. IS400 2116 Dk/Df table.

For the core, the structure used is the 7628, which has a resin content of 45%, a thickness of 0.203mm, and the Dk/Df is shown below:

Dielectric Constant (DK) / Dissipation Factor (DF)					
100 MHz	500 MHz	1 Ghz	2 GHz	5 Ghz	10 Ghz
4.48 0.0141	4.40 0.0156	4.42 0.0168	4.37 0.0181	4.35 0.0200	4.32 0.0201

Figure 24. IS400 7628 Dk/Df table.

A defined impedance layer buildup was asked from the manufacturer. Upon this request, the manufacturer pays special care to the thicknesses of the substrates to be as specified, because not only in this way the calculations for the impedance can be valid, but aids to re-manufacturability as well. The PCB build up is as shown below:

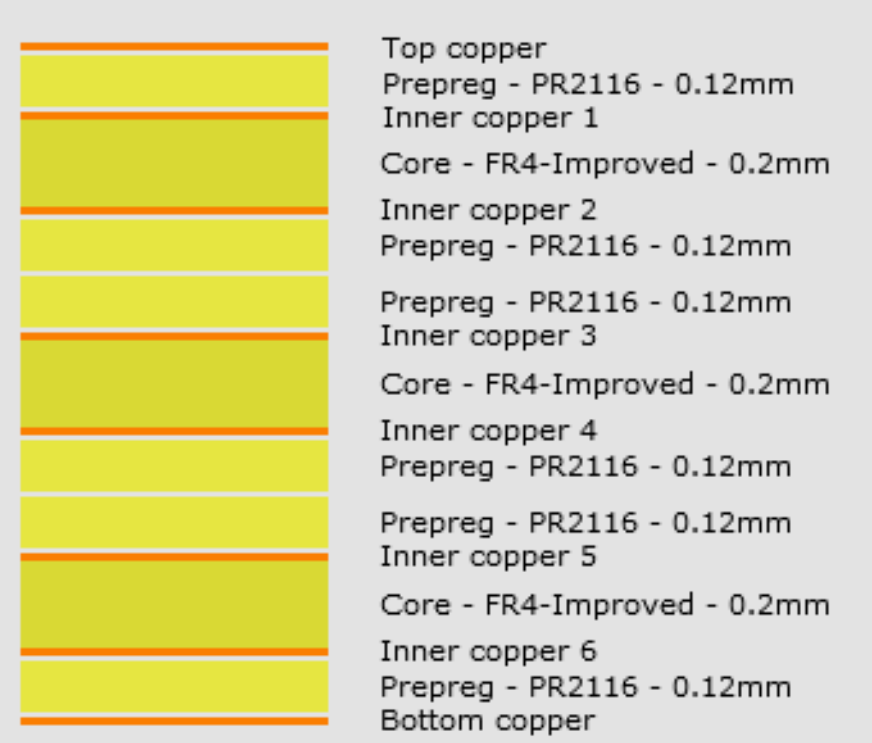


Figure 25. The defined impedance buildup of the PCB.

Especially, the usage of very low-thickness substrate below the outer copper layers, enabled the board to meet the specifications under an 8 layer design. Due to the little number of layers used, a big number of RAM traces had to be routed on the outer layers (more importantly the bottom layer). Without the low-thickness dielectric, the traces should be very wide in order to obtain the impedance required, and this could not be realized. However, the usage of low-thickness substrate below the outer layers allowed to get to the impedance required while not going to very wide, thus non-realistic, traces.

Autorouter was not used in any point for routing the PCB. The finish used for the PCB is ENIG. For the PCB testing, flying probe testing was used by the manufacturer. In the first production run of 2 boards, one failed to pass the testing, so the manufacturer performed a second production run.

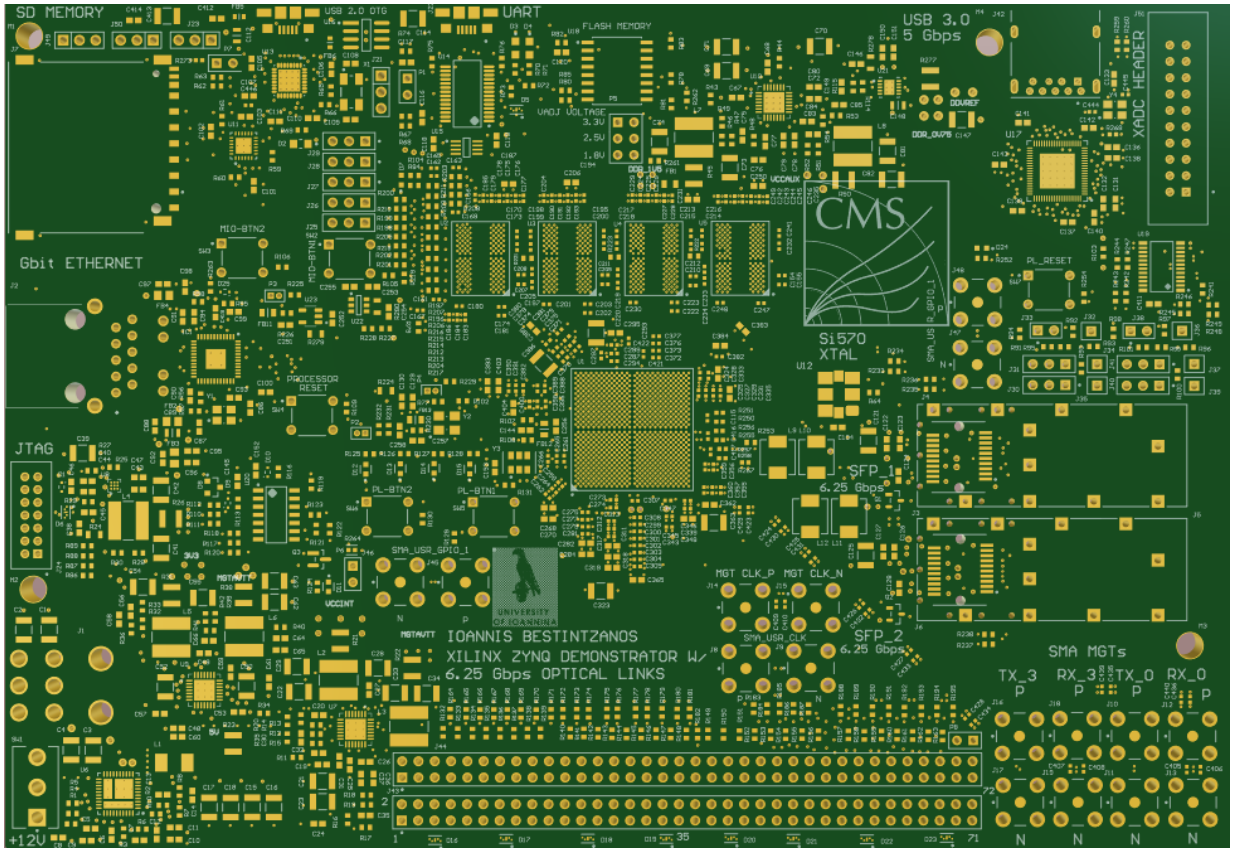


Figure 26. Top view of the PCB.

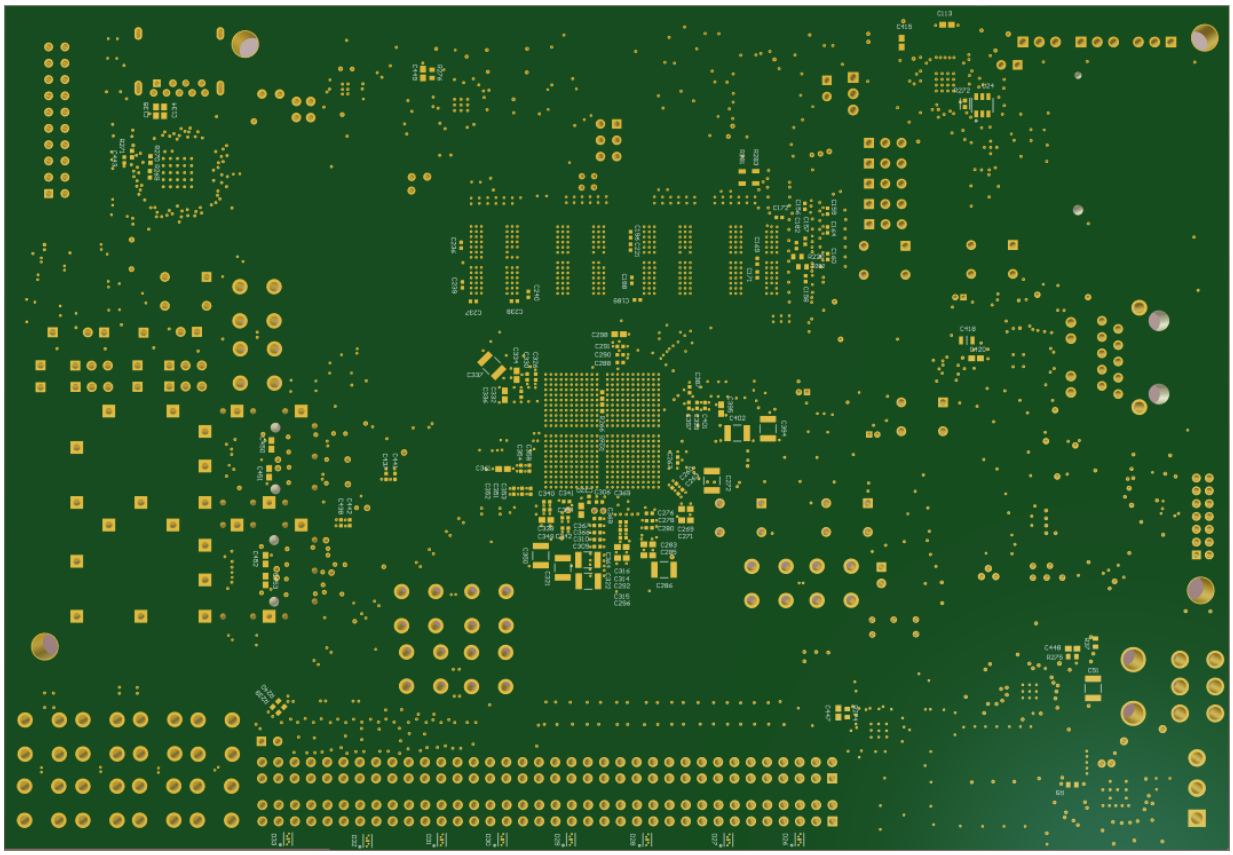


Figure 27. Bottom view of the PCB.

Chapter 4: System components

4.1 High level block diagram

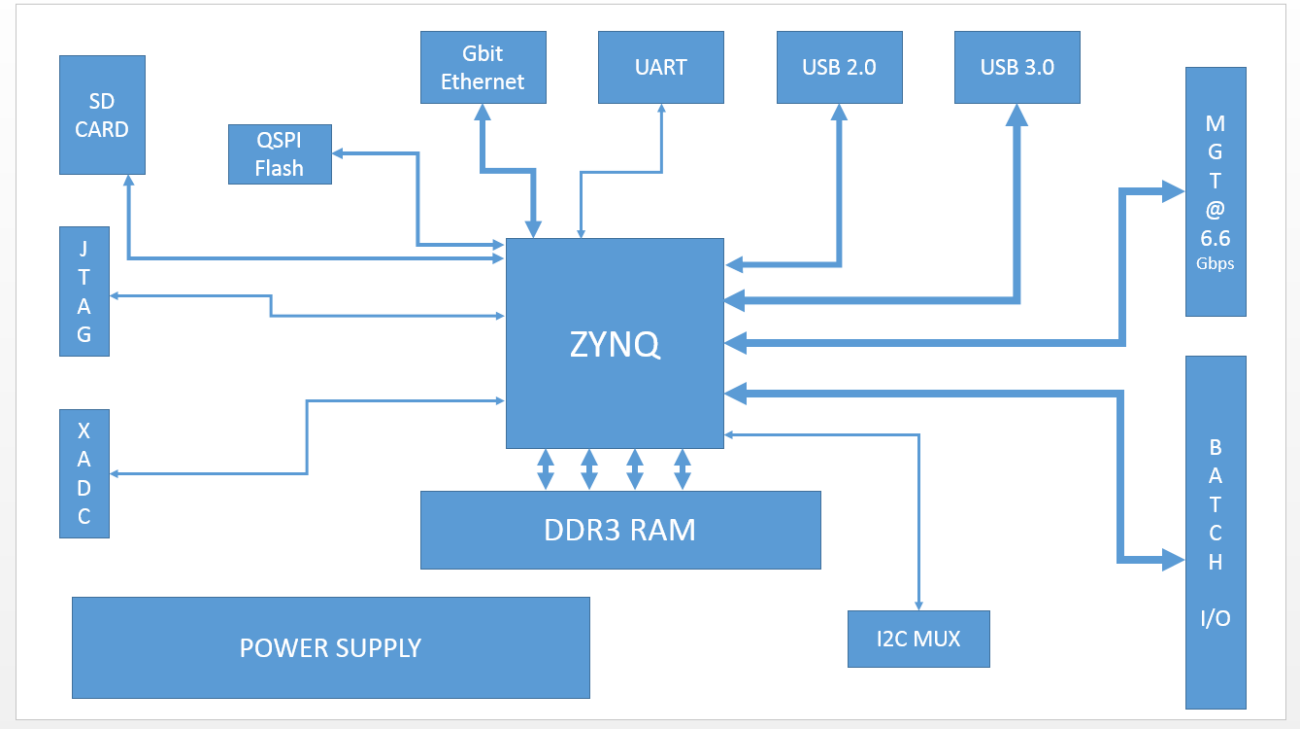


Figure 28. High level block diagram of the system. The width of the arrows corresponds to the interface throughput.

The system designed in this thesis mainly consists of the following subsystems:

- **Power supply**
- **ZYNQ**
- **DDR3 Random Access memory**
- **6.6 Gbps GTP Multi-Gigabit Transceivers**
- **Gbit Ethernet**
- **USB 2.0**
- **USB 3.0**
- **32-bit singled ended optimized pin header**

- **16-bit differential optimized (or 32 bit singled ended) pin header**
- **SD Card memory**
- **QSPI Flash**
- **UART**
- **I2C MUX**
- **JTAG**
- **XADC**

4.2 Power Supply

The card's power supply consists of 13 separate voltage rails. In short, these are: 12V, 5V, 3.3V, VCCINT, MGTAVCC, MGTAVTT, VCC1V8, DDR1V5, VCCAUX, DDR0V75, DDVREF, XADC-1V25, and XADC-1V8. Each power rail has to fulfill different power requirements, and as a consequence the circuits differ. All the power rails, with the exception of 12V, which is externally supplied, are based on the buck topology. The buck converter topology allows us to have a high current supply capability, while at the same time maintaining high conversion efficiency. The integrated circuits that were used as the controllers for the power supply rails are specialized for this kind of application. Characteristic of the power supply system is that on every power rail, there is a shunt resistor right after the converter, so the supply current can be monitored by measuring the voltage drop across the resistor, but to facilitate as well a possible debug on a specific power rail, by cutting the supplied components from the voltage converter.

4.3 Buck converter

This kind of converters is used to provide an output voltage lower than the input voltage, usually accompanied by a higher current output than what is supplied to the converter. The buck converter belongs to the family of switching power supplies. In its simplest form, it consists of a transistor, a diode, an inductor and an output capacitor. They are characterized of very high conversion efficiency values (often exceeding 90%), a factor that makes them widely accepted by the electronics industry. They are the heart of modern electronics power supplies, and they appear in ATX power supplies, motherboards, mobile phones, etc. A buck converter can be characterized by two main states, the On-state and the Off-state. During the On-state, the transistor, which acts like a switch, starts conducting, but due to the current increase, the inductor reacts by creating a reverse voltage difference across its terminals. Since the diode blocks the potential short circuit that could be formed, the current moves towards the output capacitor. So the voltage that the capacitor sees after the inductor is smaller than the input voltage. Due to the inductance of the inductor, an energy is saved in the inductor in the form of a magnetic field, E = ½(LI^2). During the Off phase, the transistor stops conducting, and the magnetic field stored in the inductor starts collapsing. This creates a current towards the output capacitor, but now the circuit is completed through the diode, which functions in forward conduction mode. In this point it is worth noting that the output current is higher than the input current, since it is being supplied to the output during the On and Off states, but the input only supplies current to the converter during the On-state. If we provide current to the converter and never let the magnetic field that exists in the inductor completely collapse, we can then ensure that the output voltage will always be greater than zero. The typical frequency range in which buck converters function is relatively high, in the order of 20KHz – 1MHz. Usually a ferrite core is used at the inductor, because in this way we can achieve a higher inductance, while minimizing the physical footprint – and hence the cost – of the inductors. Ferrite cores have relatively low losses as well at high frequencies, something that makes them even more fit for the application's requirements.

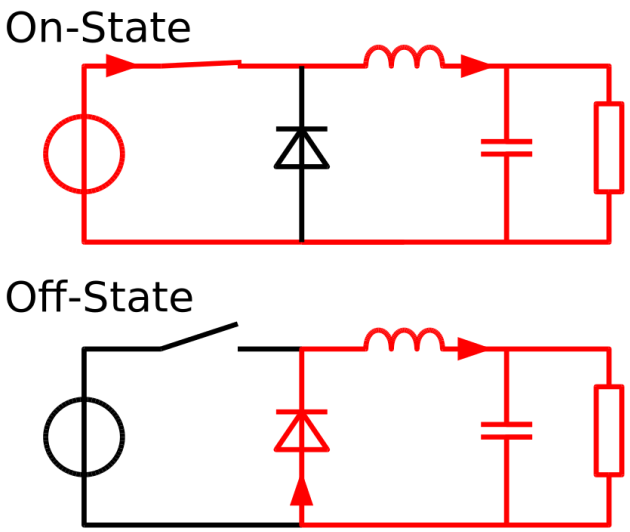


Figure 29. A simple buck converter model, and its two principal states.

4.4 Power distribution network

Of fundamental importance is the power distribution network that is able to supply high currents with low enough impedance in every corner of the card that needs the power. Low impedance of the supply network is imperative for the smooth function of the digital circuits of the card, especially when these systems consist of an ARM processor, a relatively large and high density FPGA, digital multi-gigabit transceivers and many other signals with a high frequency component. To achieve this, two copper layers were dedicated to the power distribution, and a separate copper is dedicated to the ground.

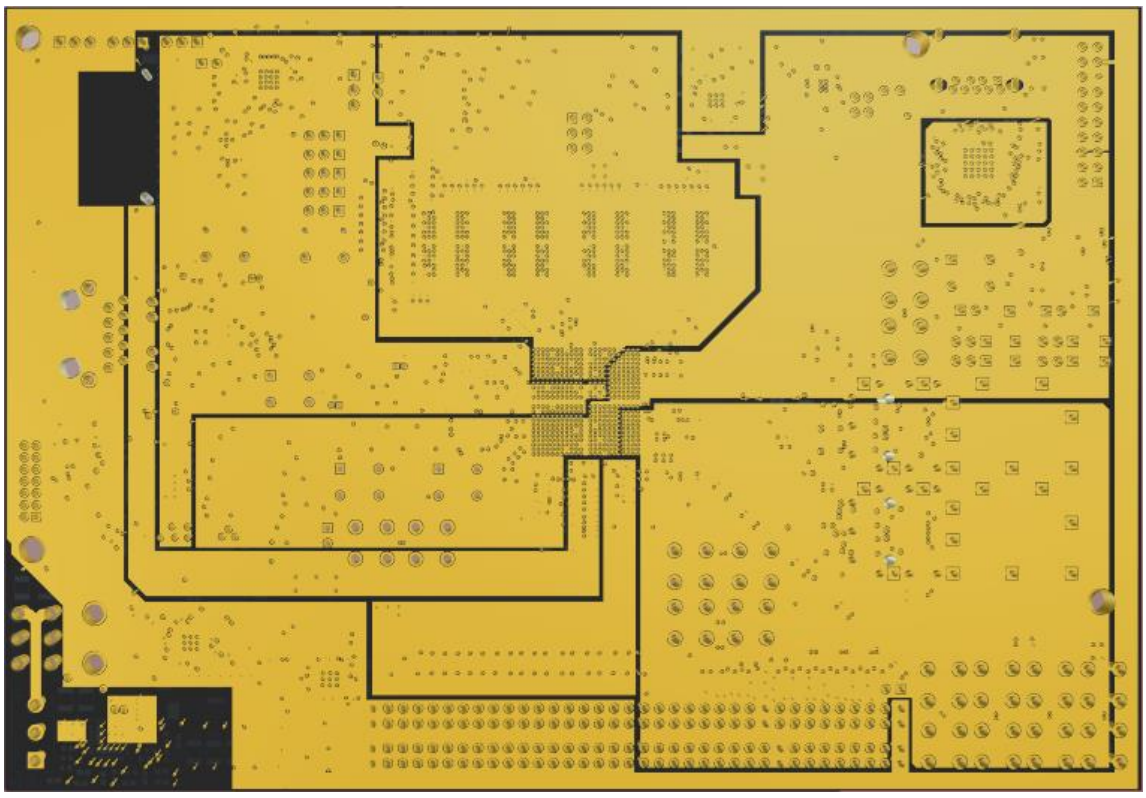


Figure 30. Power plane 1

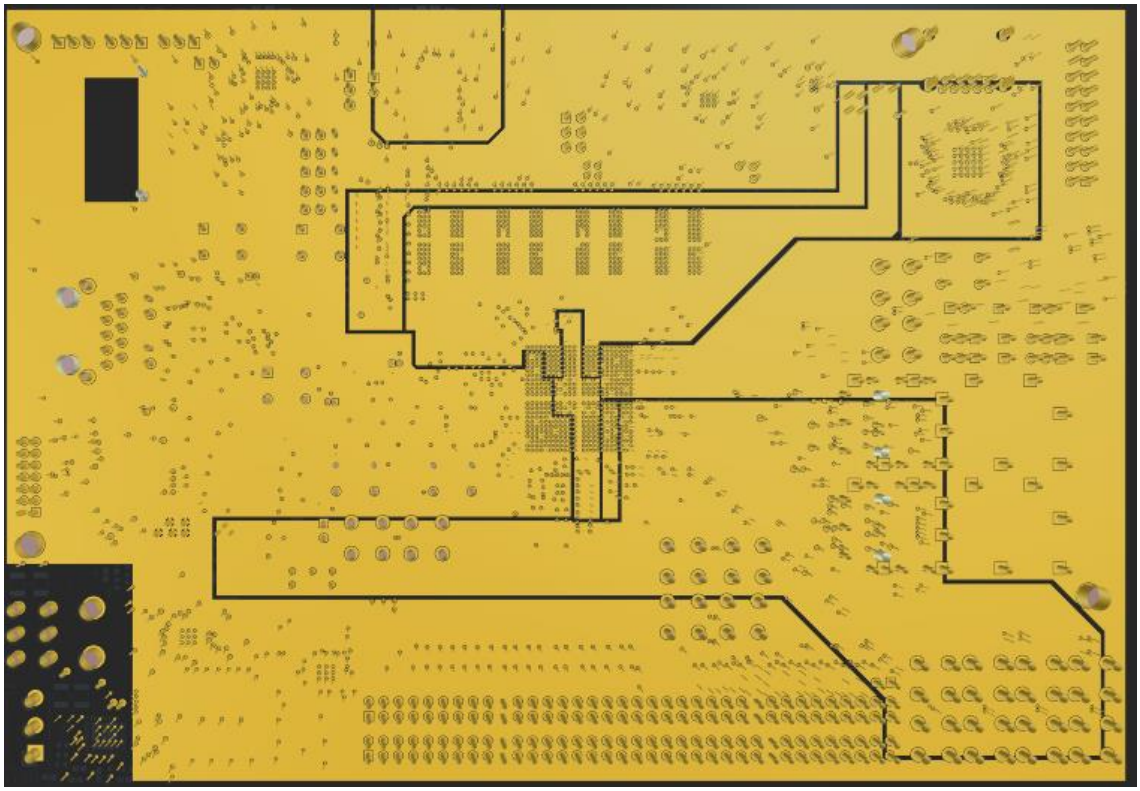


Figure 31. Power plane 2

4.4.1 12V

The 12V input voltage is supplied externally to the card, through the connector 39301060 manufactured my MOLEX. After this stage, 12V are either connected or disconnected to the first conversion stage through a switch (NKK MS12ANW03, [4]) manufactured by NKK. 12V are being handled only at the lower left corner of the card in order to achieve the highest isolation possible.

4.4.2 5V

The first stage of voltage down-conversion is converting the 12V input voltage to 5V. Then, 5V are distributed peripherally to the whole card in order to supply all the next conversion stages. 5V are obtained in a buck topology, using a step-down DC-DC converter controller by MAXIM able to deliver up to 25A (MAX8686, [5]). This IC can function with a wide input voltage (4.5V – 20V), while the output can range from 0.7V to 5.5V. The controller is able to

function in a multiphase configuration tied with other MAX8686 ICs, though for the current needs of this card only 1 phase is needed and utilized. A high precision voltage reference for the feedback loop is integrated in the IC. The output voltage is regulated according to the RS+ input of the regulator, utilizing a feedback loop, whose input is fed by a voltage divider. Varying the output voltage of the voltage divider we control the output voltage of the regulator. In order to achieve high precision in the output voltage, 1% precision resistors are used to create the voltage divider. 1% precision suffices, since 5V are an intermediate stage of power conversion, and not fed directly to a circuit other than the rest of the voltage regulators, the output voltage precision is not of critical importance. An important feature of the MAX8686 is that it uses a High-side and a Low-side mosfet in order to energize the inductor, and thus is capable to isolate the inductor in case a fault happens. Further useful features of the regulator, is the integrated temperature sensor and automatic over-temperature shutdown (>160C), soft-start, current limit, etc. In this case the inductor used is of 1.5uH inductance, ferrite core.

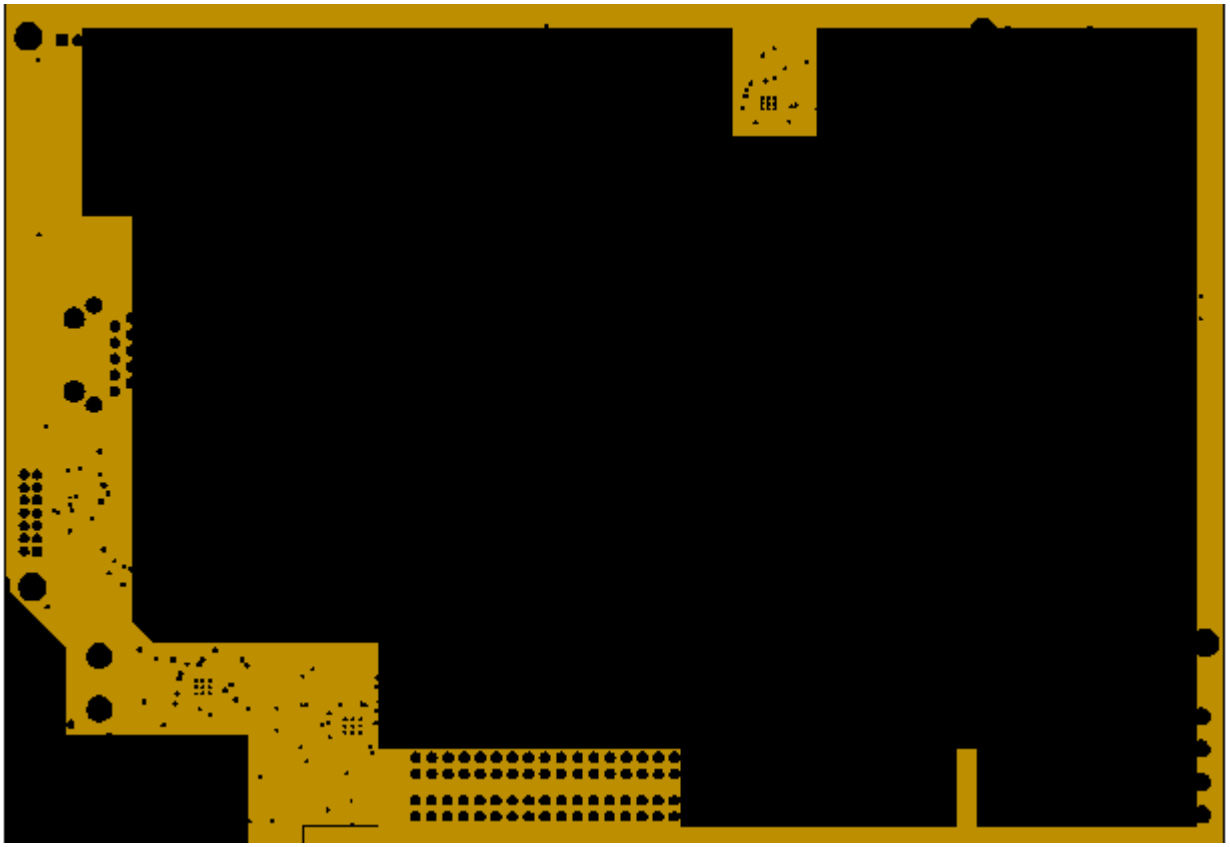


Figure 32. 5V Power plane

4.4.3 3.3V

Of the most important power rails is the 3.3V, since it is being used by many peripheral circuits on the card, and many output buffers of the FPGA use this voltage. This power rail is distributed on the whole card, and for this both of the power rails are used. 3.3V are being generated using the MAX15021 by Maxim Integrated [6]. MAX15021 is a dual-output, pulse-width-modulated (PWM), step-down DC-DC converter with tracking and power sequencing. The input voltage ranges from 2.5V to 5.5V, and the outputs can be adjusted from 0.6V up to the power supply voltage. Output 1 delivers up to 4A of current and output 2 delivers up to 2A of current. The switching frequency ranges from 500 kHz to 4 MHZ, allowing the use of smaller inductors thanks to the high switching frequency. Both output regulators operate 180 degrees out of phase, in order to reduce the input ripple current and thus to ease the input decoupling capacitors requirements. Of big importance are features such as thermal shutdown, under voltage lockout hysteresis, output short-circuit protection and more. 3.3V are generated using the first regulator of the MAX15021, and the current capability of the rail is up to 4A. Again, the output voltage is controlled by the feedback loop, using as an input the output from a voltage divider from the output to the ground, and the IC regulates this voltage to 0.6V. In order to achieve high precision, this time 0.1% resistors are used for the voltage divider. The 3V3 rail is used to supply power to the output buffers of bank 13, bank 35, and the reserved voltage inputs of the FPGA, as well as various peripheral circuits on the card.

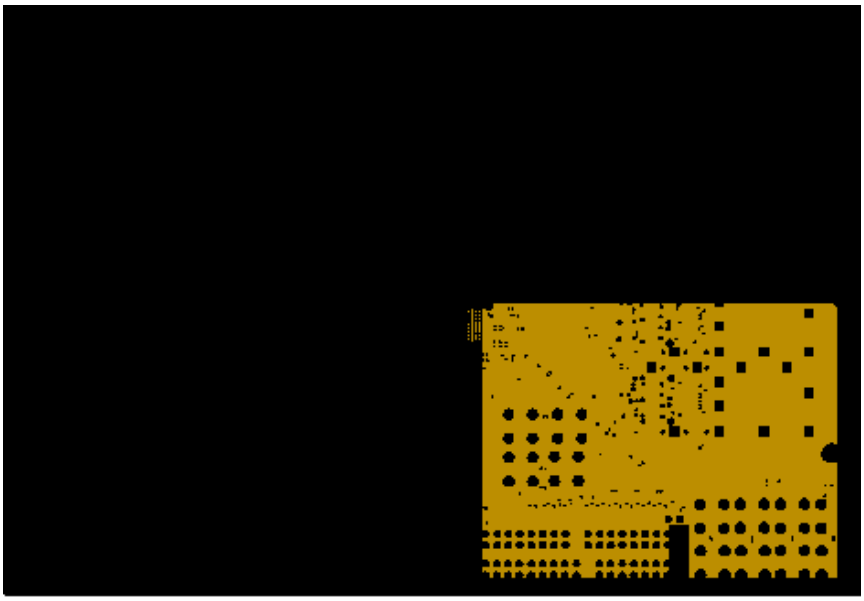


Figure 33. 3.3V plane on power plane 1.

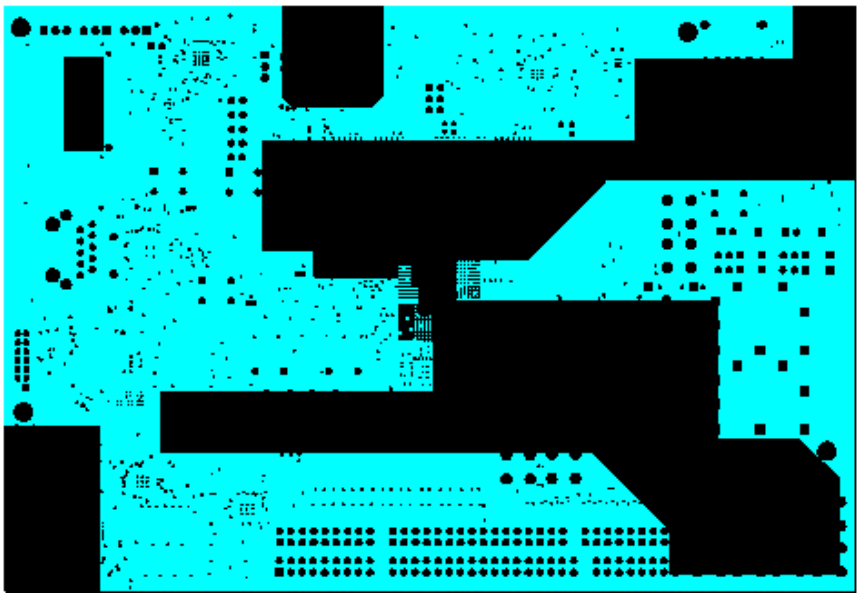


Figure 34. 3V plane on power plane 2

4.4.4 VCCINT

VCCINT is of very high importance, since it feeds the circuitry of the PS and the PL region. The current draw of this rail can get quite high, hence a 4A current capability has been ensured. Again, the first regulator of a MAX15021 has been used to generate this rail. Since any logic switches of the ZYNQ device will cause ripple to VCCINT, special care has been taken in order to decrease the impedance of the rail. More than 40 capacitors of different capacitance and ESR values have been placed in critical spots along the rail in order to achieve the least impedance possible. Again, since the voltage of this rail is critical to be precise, a voltage divider for the feedback loop has been utilized using 0.1% resistors. This rail is fed to three different power rails on the FPGA, namely VCCINT (used for the PL logic), VCCPINT (used for the ARM processor logic) and VCCBRAM (used for the Block Ram in the logic region of the device), and also supplies the core voltage for the Ethernet controller (in series with a ferrite bead, to avoid noise contamination of the rail).

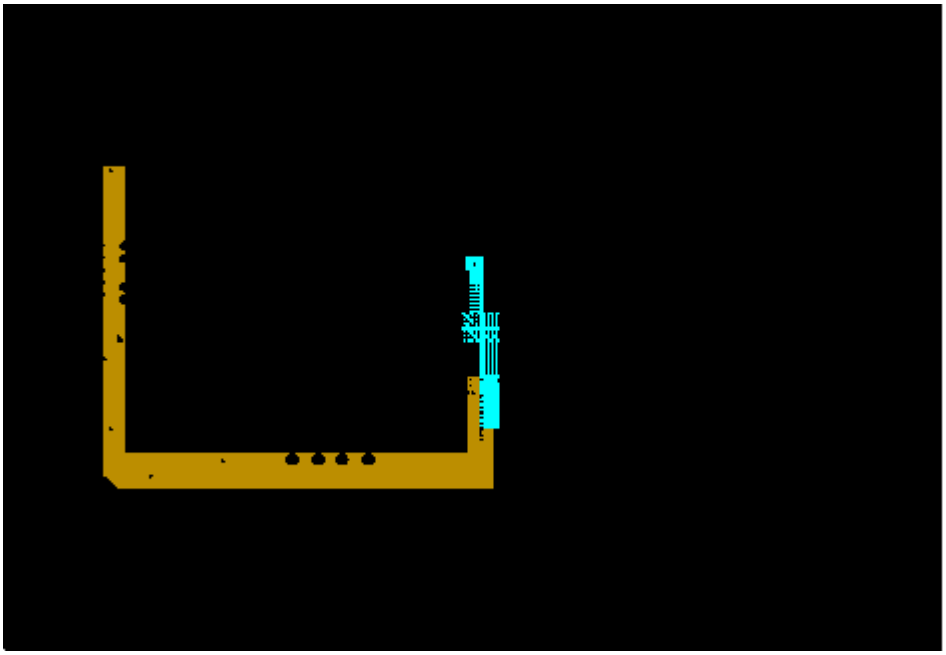


Figure 35. VCCINT planes.

4.4.5 VCC1V8

VCC1V8 is an important power rail that feeds not only some circuits of the FPGA, but a number of peripherals as well. On the FPGA device, it feeds the PLLs (in series with a ferrite bead to remove any noise, and supply the cleanest power possible to the PLLs), the XADC (again using a ferrite), VCCAUX, which is the auxiliary power supply of the FPGA, and bank 1 of the ARM processor. This rail also feeds the Ethernet controller (in series with a ferrite bead), the SD card level shifter, the core voltage and the AND gate of the USB-OTG circuitry and feeds one side of the level shifter for the UART to USB circuitry. The current demands from this rail are not that high, and in order to generate this voltage, MAX15053 has been used. MAX15053 is a high-efficiency, current-mode, synchronous step-down switching regulator with integrated switches by MAXIM Integrated [7]. It is a very compact regulator, in a 9WLP package, obtaining up to 96% efficiency. The output voltage is adjustable from 0.6V up to 0.94 x Vin, and the regulation feedback is on the same principle as previously described, again using 0.1% precision resistors for maximum voltage output accuracy.

Figure 36. VCC1V8 Power plane.

4.4.6 MGTAVCC

Multi-Gigabit Transceivers have their own special needs. These circuits are really fast and need a power rail that is capable of supplying enough current but that at the same time is able to supply noise-free power. Noise addition to these power rails hurts the MGT performance, increases link jitter and should be avoided by all means. MGTAVCC feeds the logic circuitry of the MGTs at 1.0V, same voltage as the VCCINT rail. However, a separate regulator scheme has been chosen for this rail, in order to achieve the least noise coupling from the logic as possible. This is achieved by using the second regulator of the IC that was used to create the VCCINT voltage. Current sourcing ability of this rail can reach 2A and again, special decoupling caution was taken in order to minimize the rail's impedance. The feedback loop again consists of a voltage divider from the voltage output, and of course 0.1% precision resistor are used to ensure that the voltage is spot on.

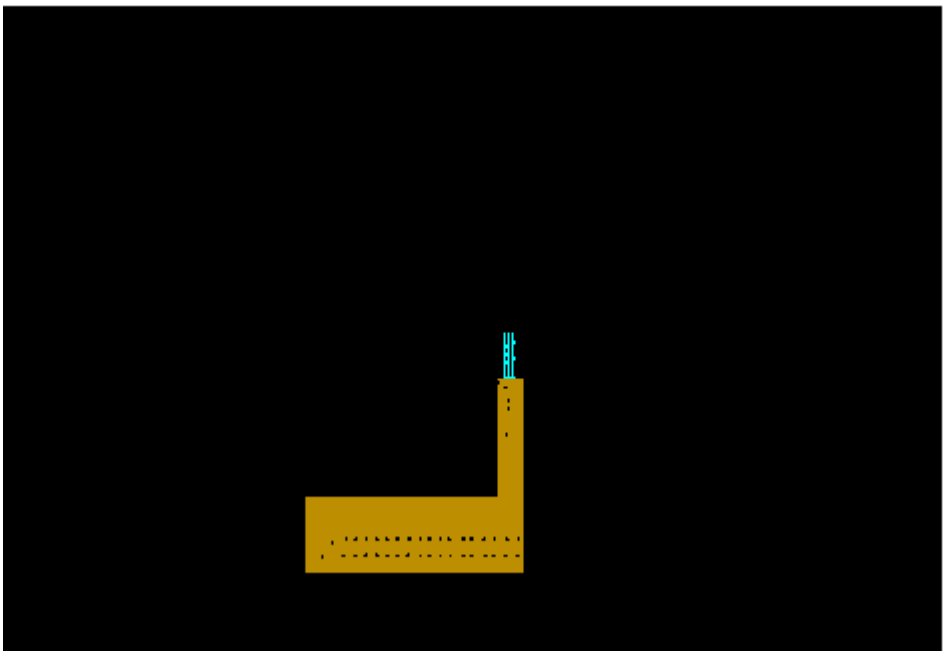


Figure 37. MGTAVCC power plane.

4.4.7 MGTAVTT

MGTAVTT is the second unique voltage rail of the transceivers, and is used in order to terminate the differential signals of the links. Again, precision is of importance in order to ensure signal integrity on the front end of the transceivers. This rail is at 1.2V, current sourcing needs are moderate, so the second regulator of the IC that was used to create 3.3V is utilized. Thorough decoupling again plays an important role in order to achieve the best specification for the rail possible. Since voltage precision here plays a big role as well, 0.1% resistors have been used in the feedback loop.

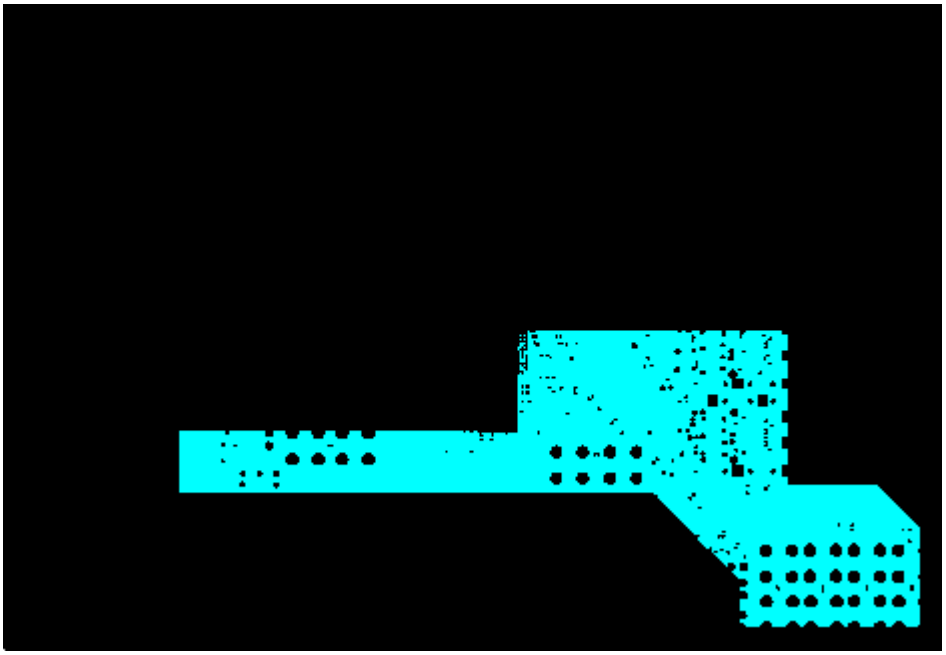


Figure 38. MGTAVTT Power plane. The extra expansion is to create a coherent plane to control the impedance of the MGT tracks.

4.4.8 VCCAUX

VCCAUX feeds output buffers of bank 34 of the FPGA. 2A of source current capability have been granted to this rail since the output buffers of bank 34 alone are not that demanding, but special care has been taken to make this rail flexible. The user can change the voltage of this rail by moving the jumper on P5. The available options are 3.3V, 2.5V and 1.8V. This is a very useful feature since bank 35 is exposed on the batch pin headers, and hence interfacing with circuits of different voltage than 3.3V is made easy. The feedback loop follows the same scheme as the previous rails, and 0.1% resistors are used here as well in order to maintain an acceptable accuracy of the voltage. This voltage rail is generated using the second regulator of the DDR1V5 controller.

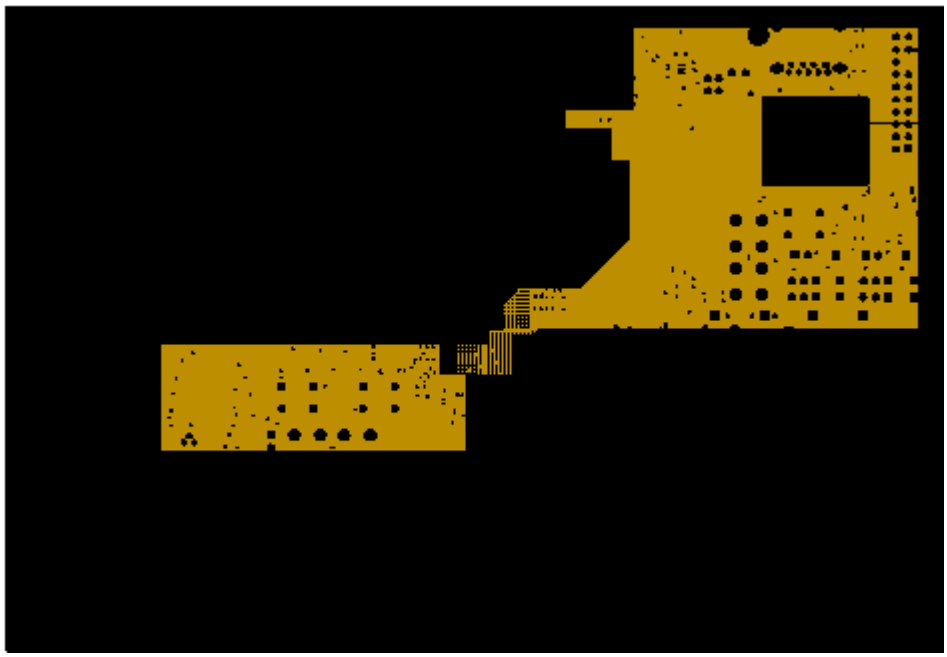


Figure 39. VCCAUX Power plane.

4.4.9 DDR1V5

SDRAM and the ARM memory controller have their own, special power needs. The RAM ICs since being DDR3 and not low power operate at 1.5V. Since 4 RAM chips have been integrated on the board and operate at the maximum frequency the ZYNQ can handle, the power ability of this rail has been enhanced to 4A. For this, the 1st regulator of a MAX15021 is used, whose 2nd regulator powers VCCAUX. Since SDRAM makes a lot of rapid switches at the same time, the voltage ripple that may appear on the power rail has to be handled. An extensive decoupling network has been designed for this rail, to ensure the minimum noise possible. Of course 0.1% resistors have been used on the feedback loop, ensuring voltage accuracy. The regulator is placed on the top of the board, close to where the SRAM is placed, ensuring a minimum distance for the current to travel, and is fed by the peripheral 5V power rail.

Figure 40. DDR1V5 power plane.

4.4.10 DDR0V75

RAM signals are fast, and without proper termination, signal integrity cannot be ensured. In order to terminate these signals correctly, a special voltage is supplied, DDR0V75, at 0.75V, which sits exactly at the middle of the bit transitions. Very important is to keep this rail clean, so the termination voltage remains the same through bit toggling, and this value has to be very accurate in order to be exactly in the middle of the transition. For this reason, 0.1% resistors have been used as previously in the feedback loop. This time, the regulator used is a MAX1510 [8], by Maxim Integrated. MAX1510 is a special DDR linear regulator that can source and sink up to 3A of peak current, and using internal n-channel MOSFETs as switches. This regulator regulates to the voltage of the REFIN pin.

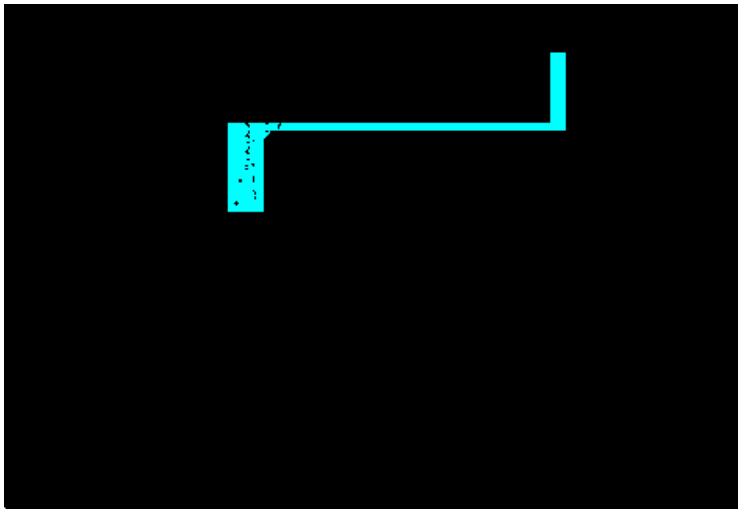


Figure 41. DDR0V75 Power plane.

4.4.11 DDVREF

The MAX1510 regulator has a buffer output that is used as the reference voltage for the SDRAM. This output can source and sink up to 5mA. The output voltage is the same as the output of the regulator of the same chip.

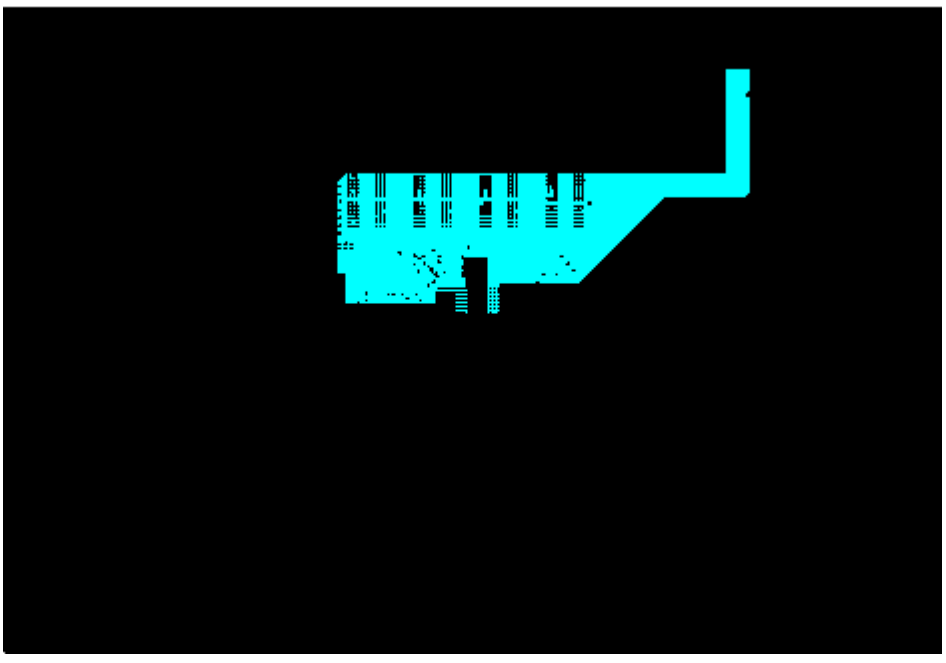


Figure 42. DDVREF Power plane.

4.4.12 XADC-1V25

The XADC integrated in the ZYNQ device needs a special 1.25V, clean voltage for the analog to digital conversions. This voltage acts as a reference and no special current demands exist, but has to be very clean for maximum noise immunity. This voltage is created using MAX6037 [9], which is a low-dropout, micropower voltage reference, with adjustable output. The output can source and sink up to 5mA of current.

4.4.13 XADC-1V8

The integrated XADC also needs a 1.8V source, for the digital power supply. Again, in order to have the cleanest possible power supply for such a sensitive component, an independent rail was designed, based on the MAX1983 [10]. MAX1983 is a low-voltage, low-dropout linear regulator with external bias supply input. The output is set again with a resistive divider, on which 0.1% resistors have been placed for maximum accuracy. Useful features of the regulator are thermal shutdown and current-limit.

4.4.14 Power sequencing

In order to minimize current draw during start-up and to meet manufacturer specifications, FPGA devices have specific power-up sequencing specifications; that meaning that one cannot power the rails in any arbitrary manner [11, p. 8]. A special procedure has to be followed, and in order to do this special circuitry has been implemented, with the “enable” and “Power-OK” feature of the regulators chosen for the power supply coming handy in this case. In order to compare if the voltages have gone above a certain threshold, so they can be considered as live, an LM339M quad-comparator is used. When 12V are switched, the first rail to come alive is the 5V, since it is the one needed for the rest of the regulator to function. When and if 5V have successfully powered up the PGOOD_5V signal generated from the 5V regulator, and indicating that 5V is up and stable, acts as the enable signal for the VCCINT regulator. 5V also acts as a supply for the XADC_1V8 rail which gets powered up as well. VCCINT is compared with a voltage divider output that is fed by a 3V reference voltage created with a precision

Zener diode. The output of the divider is configured to be 0.92V and is fed to the inverting input of the first comparator. When VCCINT, which is fed in the inverting input of the first comparator goes higher than 0.92V, PG-1V0 signal is becoming high, meaning that the VCCINT rail is up. PG-1V0 acts as the enable signal for the VCC1V8 rail. MAX15053, which generates VCC1V8, has a PGOOD output which goes up when the output is up and steady. This signal then acts as the enable signal for 3V3FPGA, MGTAVCC, MGTAVTT, DDR1V5 and VCCAUX. Together with 3V3, RAM reference voltages are being enabled, and their corresponding OK signal (PG-DDR) is going high. Then for the second comparator, on the inverting input is the 0.92V reference voltage, and on the non-inverting input is the PG-DDR flag. This asserts the PG-ALL signal, which lights up the POWER_OK led, and is fed to the inverting input of the third compactor, with a 3V reference on the inverting input, which asserts PS-POR-B, and enables the FPGA to power up. Using the fourth comparator, VCCAUX can pull down PG-ALL, by comparing to a voltage (~0.6V) created with the 3V reference and a voltage divider, in case a fault happens on this rail.

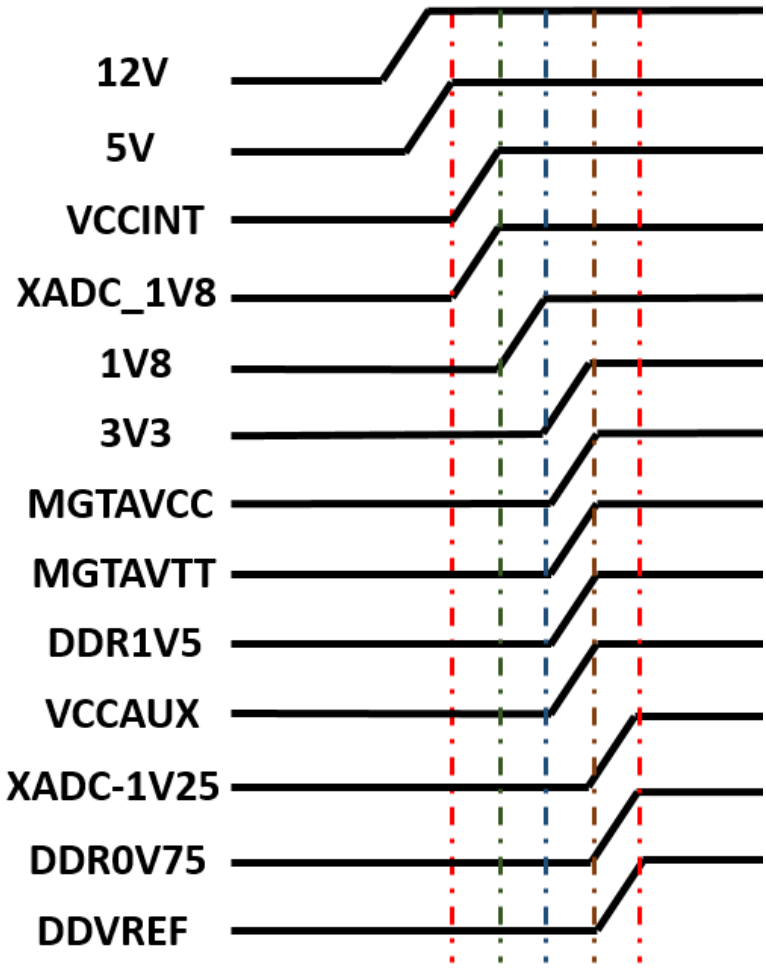


Figure 43. Power sequencing order.

4.4.15 Debugging features

Prototyping a complex power supply can be very demanding on debugging if the designer has not taken this into account. For this reason, every single power rail on this board has a very small series resistor (0.01Ω) right after the regulator, which can serve two purposes. First, measure the current of the rail by measuring the voltage differential across the resistor, and secondly, one can completely remove the resistor in order to isolate a power rail from the board. Furthermore, every enable signal for the regulators has a 0 ohm resistor in series, so one can disable this power rail by unsoldering this resistor.

4.5 ZYNQ Device

The main component of the card is a XILINX ZYNQ device. A ZYNQ is a SoC (System on Chip) manufactured by XILINX, which features an ARM processor tightly coupled, in the same die, with an FPGA. Such integration allows for a lot of flexibility, since a serial processor always comes in handy to execute the serial computing heavy tasks, and act as a controller for the card or operate an OS on the card, which if coupled with Ethernet connectivity can transform the card in a complete computer system. The ZYNQ used on this card is a XC7Z015-2CLG485C, of the 7-Series ZYNQ family, which features a two-core ARM Cortex A-9 processor, capable of running up to 866MHz, and the FPGA part is an Artix-7 equivalent, 74K Logic Cells, and 4 GTP Multi Gigabit Transceivers, with a maximum data rate of 6.6 Gbps. The ARM processor has a dedicated, hard memory controller capable of interfacing DDR2, DDR3 and LPDDR3 SDRAM memory.

4.5.1 ZYNQ Technology

Up to now, needs for having a serial processor in the FPGA were met by instantiating "soft" processors, meaning that they are implemented in the configurable logic, such as Microblaze or PowerPC, often supplied by the FPGA vendors as IP cores. Though a handy solution, these cores are taking up precious configurable logic space inside the FPGA and have performance limitations, since the reconfigurable logic cannot operate as fast as "hard" silicon due to its architecture. Nevertheless, these cores are usually 32-bit processors, often capable of running a full OS. Of course, just including an independent processor in the same card as the FPGA, while a common sight, has its own limitations, since if one asks for a fast and low latency coupling between the FPGA and the processor, in need of big data rates, has to sacrifice a lot of I/Os and make a mess on the PCB board from the traces needed, since these buses in their nature are parallel. More sophisticated solutions, such as interfacing through a PCI-e interface exist, but still one needs to sacrifice scarce and expensive MGTs. This would cover the upper end of the applications, but what about the middle end, which needs a capable enough

processor, but not such a complicated design? The solution came by integrating the processor in the same die such as the FPGA. This move alone sold many problems. First, since in the same die, one can have a big number of wide buses moving to the logic part of the SoC without needing extra expensive PCB layers and worrying about signal integrity. The choice of the processors used, being low-power RISC architectures allows for having a capable, multicore, 64-bit solution, capable of running most tasks in a breeze. The choice of the embedded processors varies widely, with single core, Cortex-A9 Arm processors, clocked at a bit higher than half a GHz, being on the low end of the 7-Series ZYNQ parts. On the upper part of the range, some Ultrascale+ ZYNQ devices feature Quad-core ARM Cortex-A53 MPCores up to 1.5GHz as application processors, with Dual-Core ARM Cortex-R5 MPCores up to 600MHz, and integrated Mali GPUs.

Zynq®-7000 SoC Family

		Cost-Optimized Devices						Mid-Range Devices			
	Device Name	Z-7007S	Z-7012S	Z-7014S	Z-7010	Z-7015	Z-7020	Z-7030	Z-7035	Z-7045	Z-7100
	Part Number	XC7Z007S	XC7Z012S	XC7Z014S	XC7Z010	XC7Z015	XC7Z020	XC7Z030	XC7Z035	XC7Z045	XC7Z100
Processing System (PS)	Processor Core	Single-Core ARM® Cortex™-A9 MPCore™ Up to 766MHz			Dual-Core ARM Cortex-A9 MPCore Up to 866MHz			Dual-Core ARM Cortex-A9 MPCore Up to 1GHz[1]			
	Processor Extensions	NEON™ SIMD Engine and Single/Double Precision Floating Point Unit per processor									
	L1 Cache	32KB Instruction, 32KB Data per processor									
	L2 Cache	512KB									
	On-Chip Memory	256KB									
	External Memory Support[2]	DDR3, DDR3L, DDR2, LPDDR2									
	External Static Memory Support[2]	2x Quad-SPI, NAND, NOR									
	DMA Channels	8 (4 dedicated to PL)									
	Peripherals	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO									
	Peripherals w/ built-in DMA[2]	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO									
	Security[3]	RSA Authentication of First Stage Boot Loader, AES and SHA 256b Decryption and Authentication for Secure Boot									
	Processing System to Programmable Logic Interface Ports (Primary Interfaces & Interrupts Only)	2x AXI 32b Master, 2x AXI 32b Slave 4x AXI 64b/32b Memory AXI 64b ACP 16 Interrupts									
Programmable Logic (PL)	7 Series PL Equivalent	Artix®-7	Artix-7	Artix-7	Artix-7	Artix-7	Artix-7	Kintex®-7	Kintex-7	Kintex-7	Kintex-7
	Logic Cells	23K	55K	65K	28K	74K	85K	125K	275K	350K	444K
	Look-Up Tables (LUTs)	14,400	34,400	40,600	17,600	46,200	53,200	78,600	171,900	218,600	277,400
	Flip-Flops	28,800	68,800	81,200	35,200	92,400	106,400	157,200	343,800	437,200	554,800
	Total Block RAM	1.8Mb	2.5Mb	3.8Mb	2.1Mb	3.3Mb	4.9Mb	9.3Mb	17.6Mb	19.2Mb	26.5Mb
	(# 36Kb Blocks)	(50)	(72)	(107)	(60)	(95)	(140)	(265)	(500)	(545)	(755)
	DSP Slices	66	120	170	80	160	220	400	900	900	2,020
	PCI Express®	—	Gen2 x4	—	—	Gen2 x4	—	Gen2 x4	Gen2 x8	Gen2 x8	Gen2 x8
	Analog Mixed Signal (AMS) / XADC[2]	2x 12 bit, MSPS ADCs with up to 17 Differential Inputs									
	Security[3]	AES & SHA 256b Decryption & Authentication for Secure Programmable Logic Config									
	Speed Grades Commercial	-1			-1			-1			-1
	Speed Grades Extended	-2			-2,-3			-2,-3			-2
	Speed Grades Industrial	-1, -2			-1, -2, -1L			-1, -2, -2L			-1, -2, -2L

Notes:
1. 1 GHz processor frequency is available only for -3 speed grades in Z-7030, Z-7035, and Z-7045 devices. See DS190, *Zynq-7000 SoC Overview* for details.
2. Z-7007S and Z-7010 in CLG225 have restrictions on PS peripherals, memory interfaces, and I/Os. Please refer to UG585, *Zynq-7000 SoC Technical Reference Manual* for more details.
3. Security block is shared by the Processing System and the Programmable Logic.

Page 2

XILINX

Figure 44. Selection of XILINX 7-Series ZYNQ Devices.

The Processor of the ZYNQ can be fully independent from the FPGA. It can have its own configuration, has its own peripherals much like a normal processor, have its own RAM memory, its own clocking circuitry and operate without having the FPGA configured. On the other hand, the same is true for the FPGA. The PL part can operate completely independent from the processor, not relying on any of the processor features.

The ARM processor has independent SRAM and L2 Cache, as well as a memory controller for external memory. Of course, interfacing to an external memory is not mandatory if one does not want to run an OS and the on-chip memory of the processor is enough for the program and the variables. In the processor are embedded DMA channels, which can come really handy when integrating a circuit on the FPGA in the application or the OS. Debugging is easier since the processor is CoreSight enabled. Several components seen on processors and microcontrollers, such as TTCs and SWDT are integrated in the processor.

The processor features many "hard" peripherals, such as Ethernet MACs, USB interfaces, SPI interfaces, SD card interfaces (in which one can save the OS, programs and bitstreams for the FPGA), UART circuits, I2C and CAN controllers and general use Inputs/Outputs. Memory interfaces on which one can load the processor configuration from are among others NOR, NAND, and QUAD SPI. While the Processor has certain pins which are used on the package and these cannot be used from the FPGA, there is a big flexibility offered on to what pins one has to use for every peripheral. The connections are multiplexed internally and the freedom offered is very handy. Also, one can use the pins dedicated to the PL for these peripherals. (referred to by Xilinx as EMIO).

The clocking needs are a single clock, from which the processor can synthesize internally the clocks for the CPU, RAM, USB etc., as well with clock outputs that attach to the PL clocking network.

A very important feature of the processor is the integrated SDRAM memory controller. The processor can have RAM independent from the RAM possibly assigned to the FPGA and using this option, vast amounts of memory can become available to run a full operating system. Memory capacity limitations on the lower end of the ZYNQ devices are at 1GByte using 2 or 4 memory chips, while on the upper end of the ZYNQ spectrum several Gigabytes of RAM can be supported by the processor. The memory controller supports write leveling and training, a feature that can improve a lot the performance of the interface by making sure that the interface bits sampling always happens at the best possible timing.

The interconnect between the processor and the FPGA happens using AXI buses, categorized as General Purpose Slave AXI interface, High Performance Slave AXI interface, and ACP (Accelerator Coherency port) Slave AXI interface. The buses are 32 or 64 bit wide, and are usually more than one in quantity per category. A DMA controller is available as well, and a PS-PL cross trigger interface.

Another important aspect of the PS-PL interconnect is the ACP port. This port allows for cache coherency between the processor cache and the memory in the FPGA, so that one can transfer information directly to the processor's cache without worrying about data synchronization problems. Under this scheme, the processor can assist the FPGA in computing heavy applications with minimal overhead, or vice-versa, the FPGA can act as a very powerful accelerator for a program that is running on the processor, a part of which has been offloaded to the FPGA.

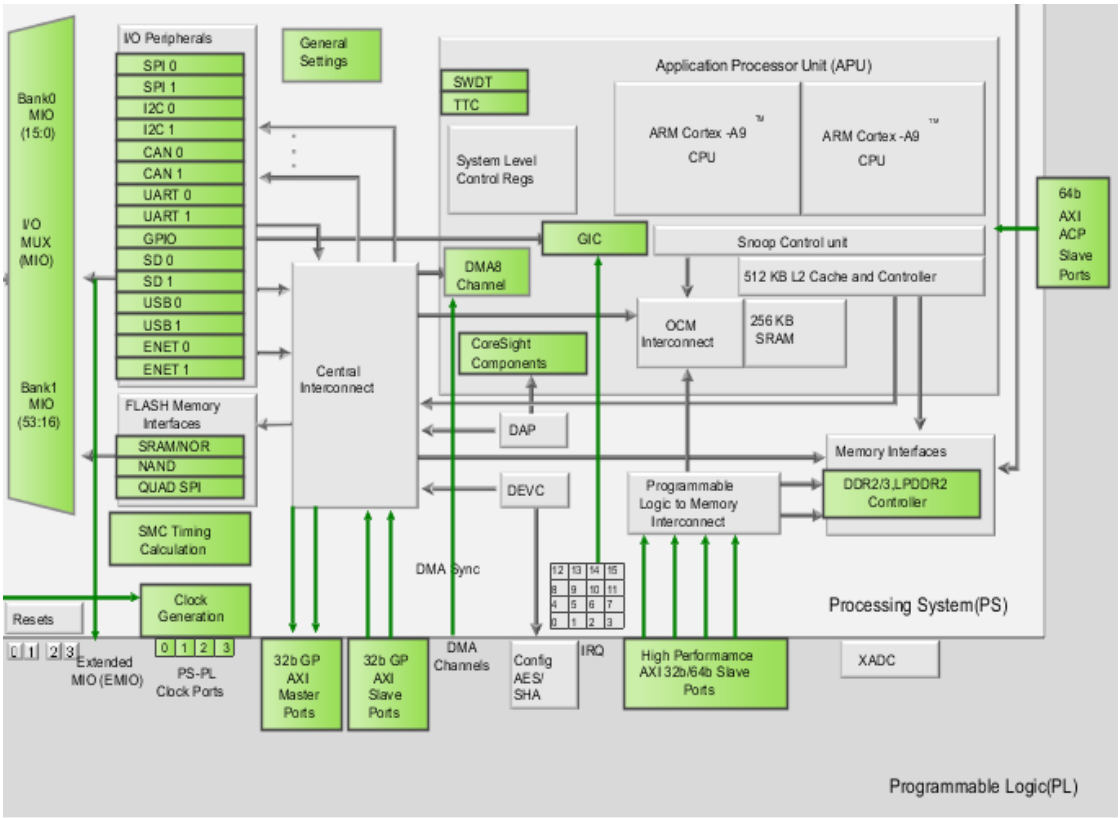


Figure 45. Block Diagram of the ZYNQ device.

4.5.2 XC7Z015-2CLG485C

The ZYNQ device used on this card is the XC7Z015-2CLG485C [12]. It is classified as a mid-range, Cost-Optimized Device, featuring a Dual-Core ARM Cortex-A9 MPCore capable of running at 767MHz, with a NEON SIMD engine per core. The logic part is an Artix-7 equivalent, packing 46,200 Look Up Tables, 92,400 Flip-Flops, a total of 3.3Mb of Block RAM arranged in 95 blocks of 36Kb each, 160 DSP slices, a Gen2 x4 PCI Express block and an XADC. The speed grade is -2, capable of running the MGTs at 6.25 Gbps, but lacks on the processor frequency against its -3 speed grade brother, with the processor running at 767MHz instead of 866 MHz for the -3 counterpart.

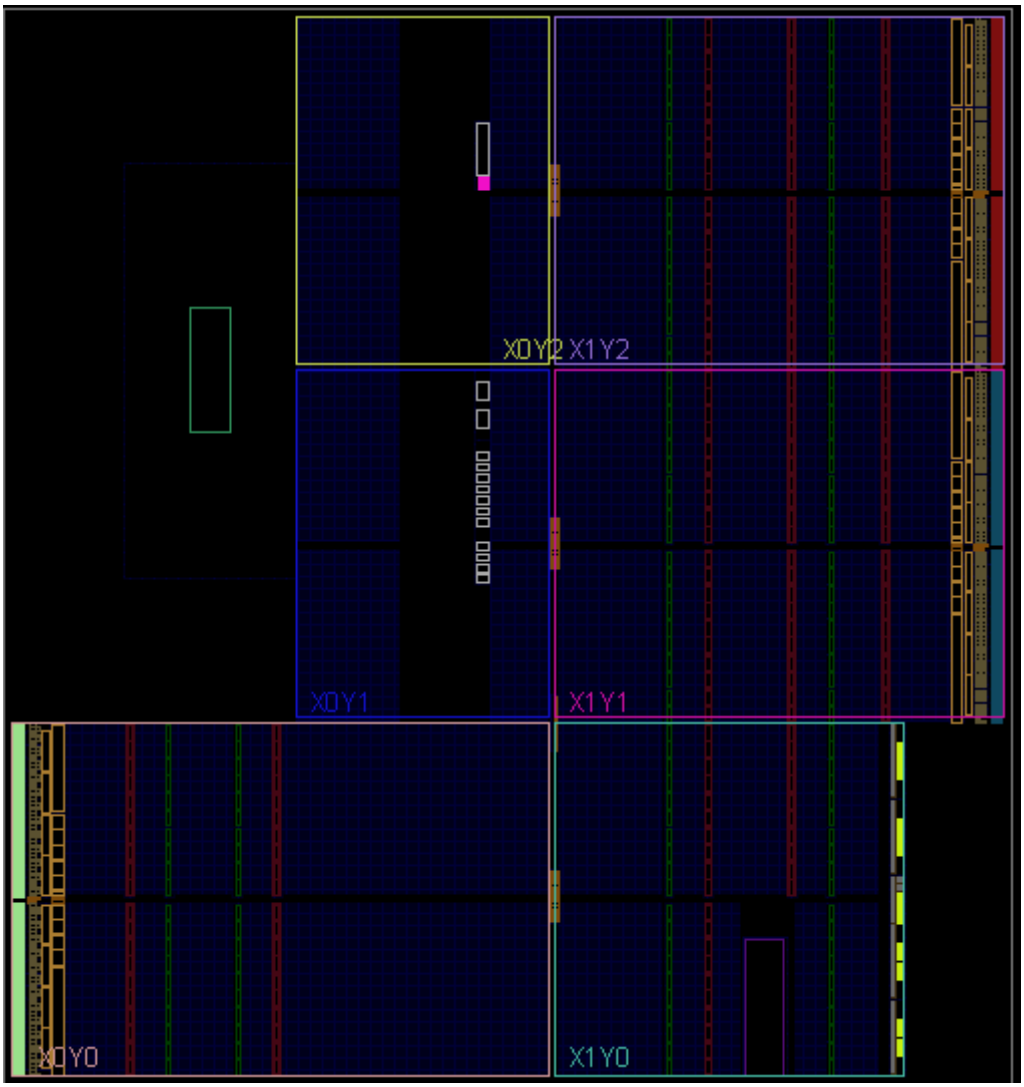


Figure 46. XC7Z015 ZYNQ device view.

This specific device features 4 GTP transceivers at 6.25 Gbps [13]. The transceivers are packed in one quad and can be clocked from 2 dedicated reference clock inputs. The reference clock gets synthesized to the frequency needed for the link by dedicated PLLs included in the MGT quad. Each transceiver has one receiving differential pair signal and one transmitting differential pair. The transmitter and the receiver need 50 ohm impedance traces.

The package of the device is CLG485C [14]. This is a BGA (Ball Grid Array) package, with 485 pins at 1mm pitch. The temperature range is Extended; that is 0 -100C. This ensures that the device can withstand continuous operation at 100C for at least 10 years. However, up to 110C operation is allowed for 10% of the device's lifespan.

Figure 47. Photo of the ZYNQ device. Visible on the right side is the DDR3 memory.

The PS has a dedicated 33.333 MHz clock, which is used by the processor, and is also fed to the processor's PLL which can synthesize various clocks and feed to the PL part. The PL has a dedicated 200 MHz differential output oscillator routed to a global clock input.

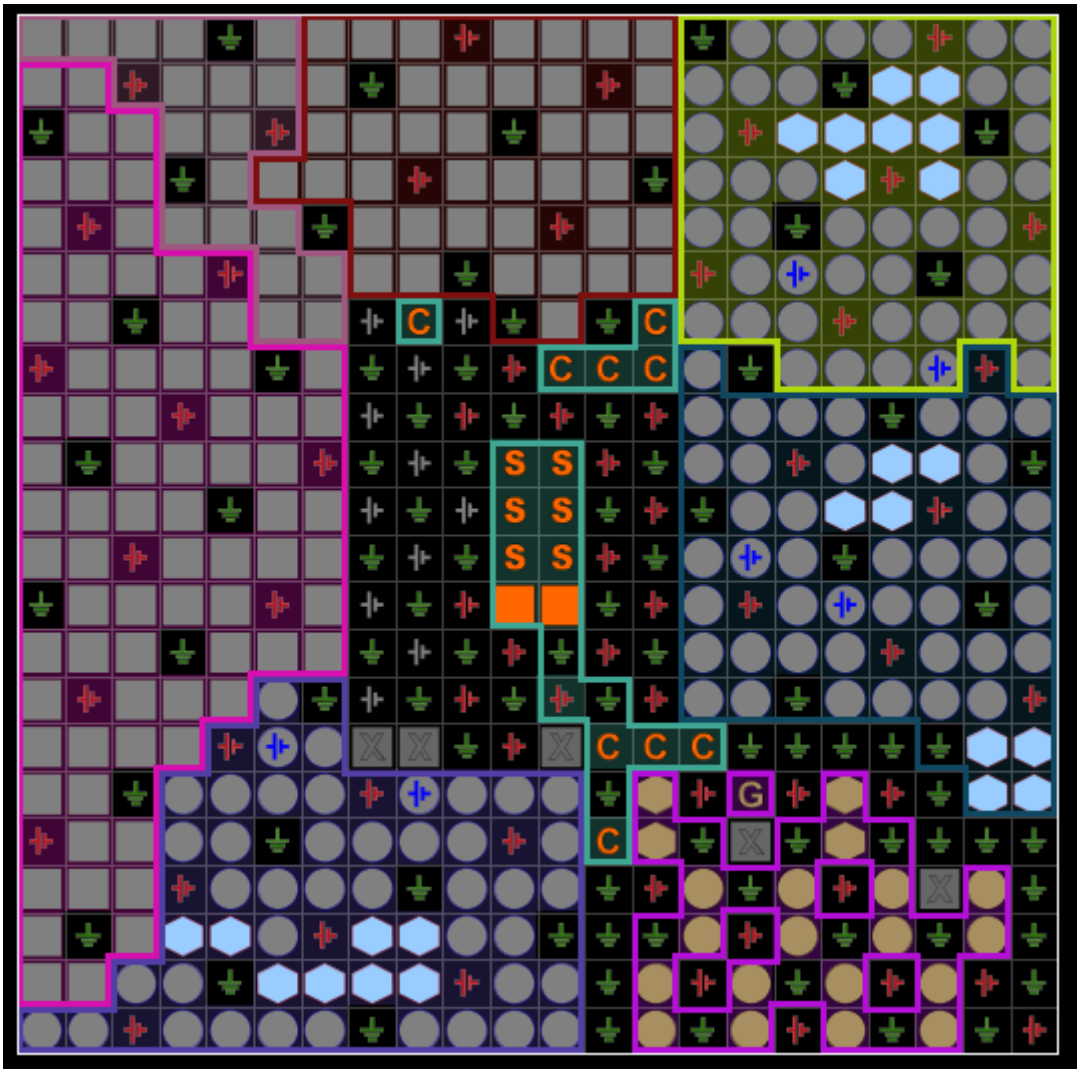


Figure 48. XC7Z015 bottom view.

4.6 Random Access Memory

It is essential for a processor to have some sort of intermediate buffer between the cache and the permanent storage, since cache memory is very little in size and permanent storage mediums are always very slow for a processor's needs. Of course, one can run a program "baremetal"; that is without any Operating System, directly manipulating the registers of the CPU, but in our case running an OS is desired, so we need the extra intermediate memory. This memory is of course SDRAM (Synchronous Dynamic Random Access Memory) for which the ZYNQ ARM Processor has a dedicated controller. Total size of the memory routed on the board is chosen to be 1GB, which is the maximum this specific device can handle. 1 Gigabyte is enough to run Linux and perform many tasks at the same time, without running out of memory. Having to use swap memory because RAM is not enough is a bad idea in this case, since the permanent storage medium of the card is SD storage, whose speed is not suitable to act as swap. For this reason, the maximum capacity of RAM memory was chosen to be installed.

4.6.1 DDR memory

The memory interface implemented is DDR3. DDR stands for Double Data Rate, which means that data are transferred on both the rising and the falling edges of the clock signal, in contrast to SDR (Single Data Rate) on which data are transferred only on one edge of the clock period. This allows for faster data rates, but also puts stricter demands on the timing accuracy. Nevertheless, the frequency for a given data rate is half of what would be needed with SDR, so overall PCB requirements are eased in comparison to an SDR with the same data rate. For example, for a DDR200 device, the data transfer frequency is 200MHz but the bus speed is 100MHz. The ZYNQ processor's memory controller natively supports DDR3 signaling, up to a frequency of a 533MHz.

Data transfer rates can be quite high figures on DDR interfaces, because not only data are sent twice in each clock cycle, and the operating frequencies can be fast, but also because the buses are parallel of many bits wide. For example, the data bus used on the board, is 32 bits wide and operates to a frequency of 533MHz. Hence, the data rate figure is 533 MHz x 2 x 32 bits = 34.1 Gigabits/second or ~4.3 Gigabytes/second.

SDRAM has its own advantages and disadvantages compared with other memory technologies. The data density it offers is quite big compared to the silicon area/power dissipated and part owing to its pipelined architecture, offers for very high frequency for the DDR interface. This allows for DRAM amounts in real word applications (such as servers) in excess of 1TB per CPU. Some its disadvantages though are the high latency, due to their complex operation and the architecture, which keeps on increasing with new technologies (e.g. DDR3 has higher latency in periods, but since a period is smaller, the latency is almost the same as that of DDR2). This makes SDRAM unsuitable for some applications, such as large Look Up Table storage, since usually such applications can't afford the latency overhead of the SDRAM. CPUs however, cope well with this latency overhead, by using usually 2 or 3 levels of intra-die cache memory, and by being able to perform other tasks while waiting for the data to arrive from the memory. Another disadvantage is the lack of data retention. Data can only be stored in an SDRAM memory for a very little amount of time, after which they are lost. The countermeasure to this is the "refresh" procedure, which ensures that the data are always there as long as there is power. Cutting the power, even for a brief amount of time such as milliseconds, will cause data corruption. Data refreshing is often a task assigned to the memory controller, although common sight on the market are chips that offer an auto-Refresh feature, though again their access and control is not as fast and simple as with SRAM.

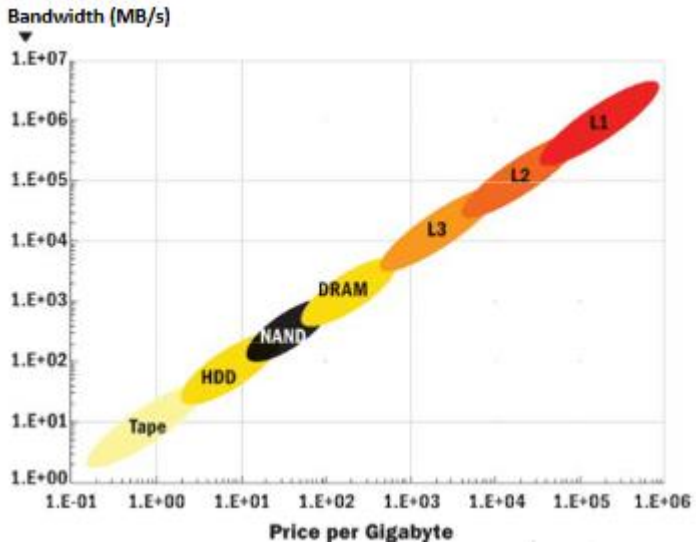


Figure 49. Different memory technologies, their bandwidth and their price. DRAM sits in the middle.

4.6.2 RAM on this board

The RAM chips used on the board are the MT41K256M8DA-125:K [15]. They are manufactured by Micron and the capacity of each chip is 2 Gb. It is configured as 32M x 8 bits x 8 banks. The nominal operating voltage is 1.5V, support write leveling and output driver calibration. The maximum operating frequency of the -125 mark is 800 MHz, and the operating temperature range is 0 to 95C. The DRAM chip package is a 78-ball FBGA, Pb-free package, at 9mm x 14mm dimensions.

In order to reach 1GB of RAM capacity, 4 Micron SDRAM ICs were placed on the board. They were placed with the maximum possible proximity to the FPGA, in order to keep trace lengths minimum for best signal integrity. 4 ICs means that some signals, especially the address signals will have to travel a relatively long way to reach all the chips. This, in addition to the small number of total layers used, makes routing and ensuring RAM performance a challenge.

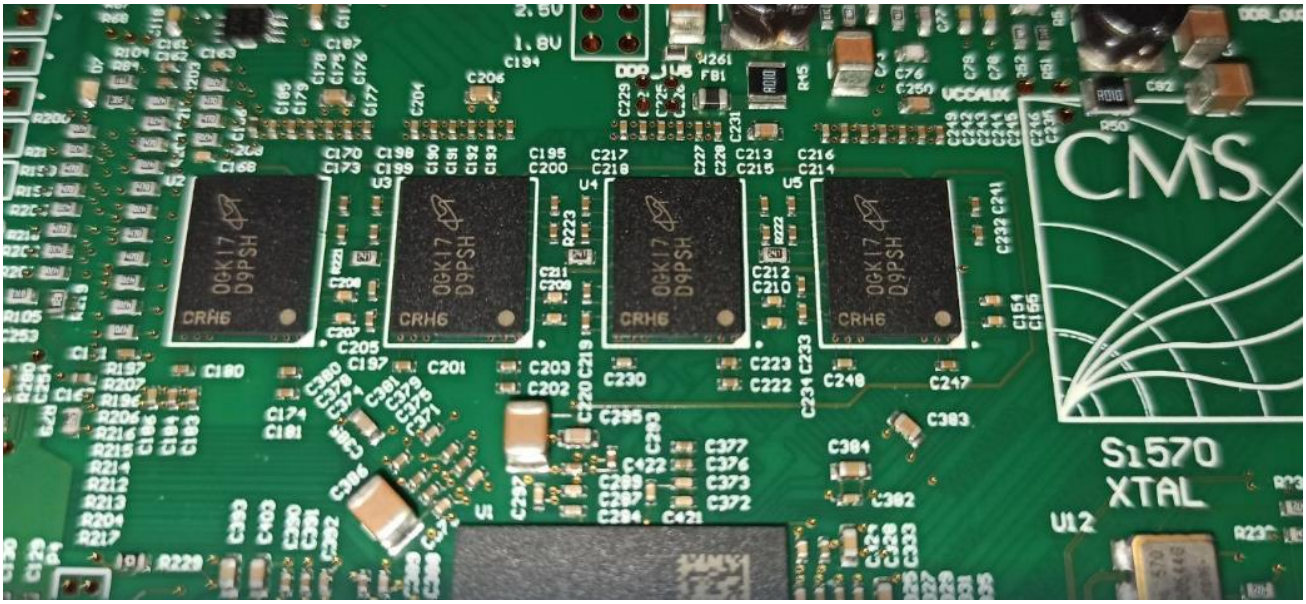


Figure 50. The Micron RAM ICs, placed close to the FPGA.

While 1GB is the maximum capacity the processor's memory controller can handle, this is not the maximum total memory this device can handle. One can route extra memory if needed to the PL part, and instantiate a memory controller in the PL in order to access it. In this way more memory is available to the user, however the latter is only accessible from algorithms in the PL part.

4.6.3 DDR3 interface

DDR3 memory bus consists of the Data and Strobe signals group and the Address, Clock and Command signals group. The data lane is an 8-bit wide, bidirectional, parallel bus for each chip. Bidirectional means that both the RAM and the Memory Controller (ZYNQ) can use this bus to send or receive data. DQS stands for Data Strobe, and is essentially a dedicated clock for each data lane. Every data lane has its own strobe, so 4 data strobes are used for this configuration. Data strobe is a differential signal, whilst each data bit is single ended, and that is needed to keep the best possible signal integrity, since a clock is a signal which faces the most transitions. One data lane with its corresponding DQS/DQS# signal is connected to every DRAM chip. In total, these 4x8 bits form a 32bit word.

On the Address/Clock/Command group, Address is the bus where the address of the memory location we need to access is transmitted. In this case, it is 15bits long, assisted by 3 extra bits for the bank selection. In this belongs the DM pin, which stands for data mask. SDRAM operation fundamentally is burst, usually transferring 4 or 8 data beats per burst. That means, that having a 32-bit wide data bus, transferring 32bit x 8 would be the minimum. However, this burst action is sometimes undesirable, since data will be written to a memory location we wouldn't want to touch. For this reason exists the Data Mask, which if asserted masks the corresponding byte, and this specific IC does not execute the operation. In the same group exist the Command lines, which are the WE (Write Enable), CAS (Column Address Strobe), the RAS (Row Address Strobe) and the CS (Chip Select). Write Enable indicates a read or write operation, CAS indicates whether the address refers to the Column addressing, RAS refers to whether the address refers to the Row addressing, and chip selects enables/disables the chip.

4.6.4 Fly-By routing topology

The routing topology chosen for this card is Fly-By topology. Although the skew in the Address / Clock / Command is small by absolute measure, it can be very important considering the frequencies memory interfaces run to, and becomes increasingly critical when the bus frequency increases, so the timing specifications get stricter. In Fly-By topology, all Address / Clock / Command signals are sourced synchronously from the memory controller. Then as they travel in the traces, they meet one chip after another, instead of arriving at every chip at the same time. This bring a lot of benefits, the first being that one needs the same amount of traces, no matter how many ICs the signal group will meet on its way. If we were to connect each chip to the controller individually, we would need a new set of traces for every single chip. Another great benefit of the Fly-By topology, is that since the signals meet every chip in distinct moments, the capacitive loading of the line is distributed in time. This allows for higher frequencies and requires less output strength which improves the signal integrity and minimizes noise. The Data/Strobe groups however are separate, and each travels to its corresponding IC. Since it is different bytes between each Data/Strobe group, this is something desirable and allows for more freedom between each group.

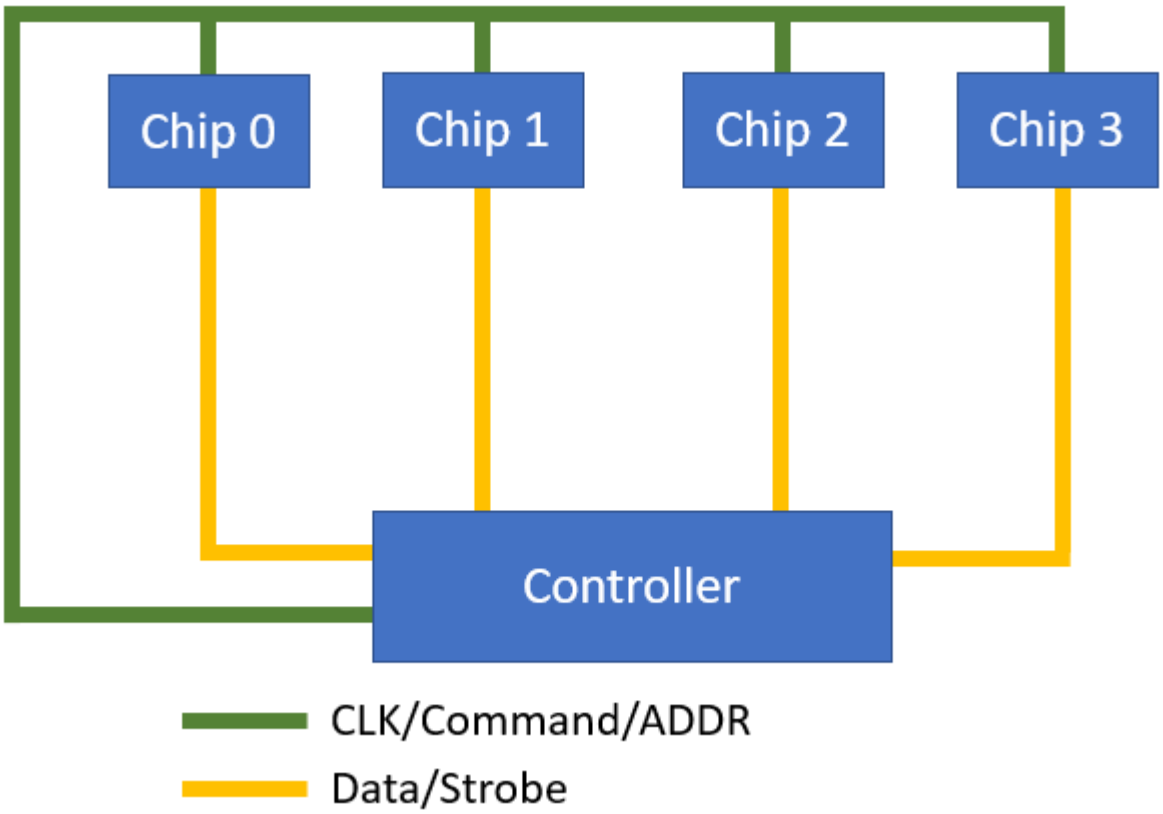


Figure 51. Fly-By topology.

4.6.5 DDR3 signal integrity requirements

DDR3 memory is very strict in timing requirements. These go up as frequency increases, and keeping the timing requirement in specs is of fundamental importance in order to achieve error-free operation. Xilinx publishes the timing requirements for the ZYNQ devices [16], and in this case are as follows:

- The maximum electrical delay between any DQ or DM and its associated DQS/DQS# must be ≤ ±5 ps.
- The maximum electrical delay between any address and control signals and the corresponding CK/CK# must be ≤ ±25 ps.

- CK/CK# signals must arrive at each memory device after the DQS/DQS# signals. The skew allowed between CK/CK# and DQS/DQS# must be bounded between 0 and 1,600 ps.
- CK/CK# must arrive after DQS/DQS# at each memory component to ensure calibration can align DQS/DQS# to the correct CK/CK# clock cycle. Write Calibration failures are seen if this specification is violated.

Keeping the signal timings under these specs is a demanding task and extensive serpentine routing and other techniques have to be used to reach them. Many times it is a compromise between space/layers available, signal integrity and timing.

Differential pairs, such as clocks, are even stricter on the timing specifications, since the least skew can cause unwanted jitter. For this, all differential pairs are matched to have no more than 5ps of skew between the positive and the negative trace.

RAM signals not only have timing specifications, but have impedance specifications as well. Single ended signals need 40 Ohms of impedance, and differential signals need 80 Ohms [16]. For lower data rates, these specifications may be relaxed to 50 and 100 Ohms. Maintaining these specifications is very important in order to achieve acceptable signal integrity and to restrict as much as possible reflection of the signal due to impedance mismatches.

4.6.6 DDR3 implementation on this board

Since this is only an 8-layer board, with only 5 layers available for signal routing (2 of which are Top and Bottom, which already have components and pads on them), routing all the RAM ICs while keeping the timing and signal integrity specs under control becomes challenging. The inherent problem here while calculating the delay of every trace is that not every layer has the same signal propagation velocity. Sure, if one has many layers at his disposal, so he can achieve the same propagation delay on every layer, this becomes easy to solve, but in this case due to the little number of layers, RAM signals have to be routed on the outer layers as well. The layer stack-up was chosen as to allow for at least one coherent plane to be used as a

reference for signals in each layer. Still though, for the signals on the outside layers, which have copper only on one side and on the other side there is air, the difference in propagation velocity becomes remarkable.

To mitigate this problem, RAM signal delays were not measured in units of length but in units of time. By inputting the exact specifications of the board materials and the stack-up, the Simbeor simulator embedded in Altium Designer used the field solver to calculate accurately the propagation velocity on each layer. Having this information, then the propagation delay is being calculated in real time using the results from the 3D solver, adding up for every single mil of trace. This technique gave exact numbers for the propagation delay of each trace, bypassing the problem of different propagation velocities on each layer.

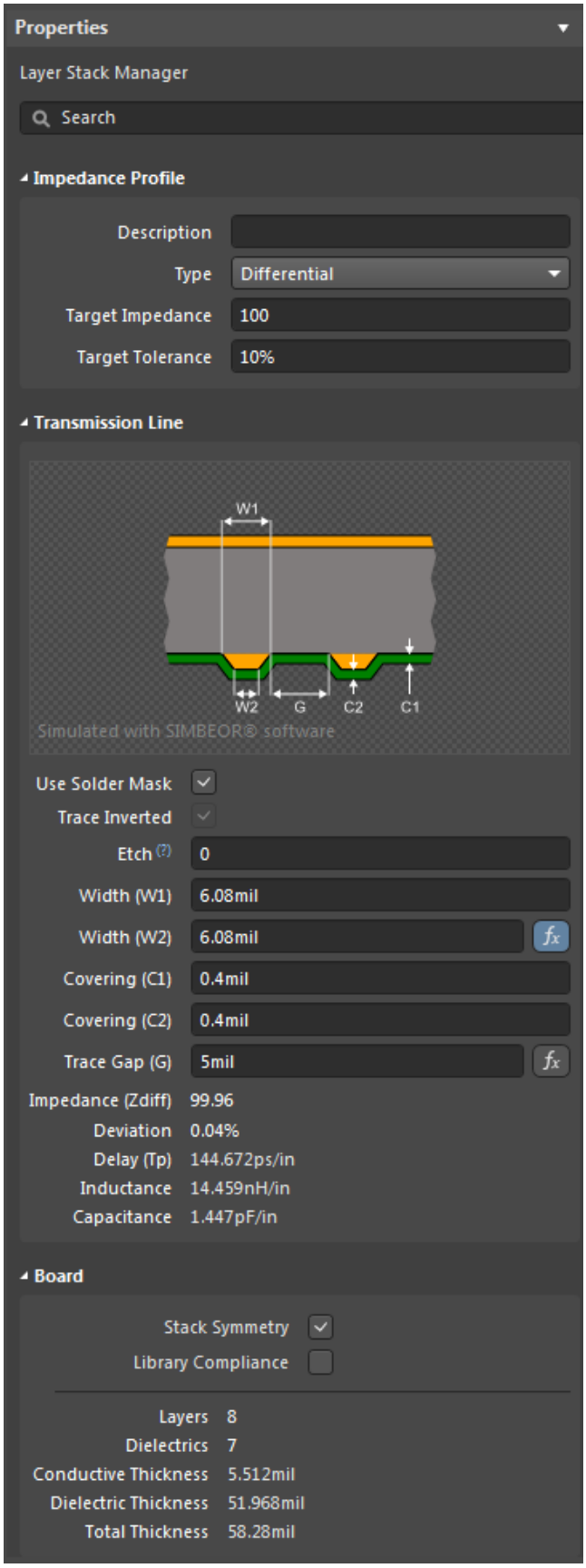


Figure 52.Impedance simulation of the bottom copper plane of the card.

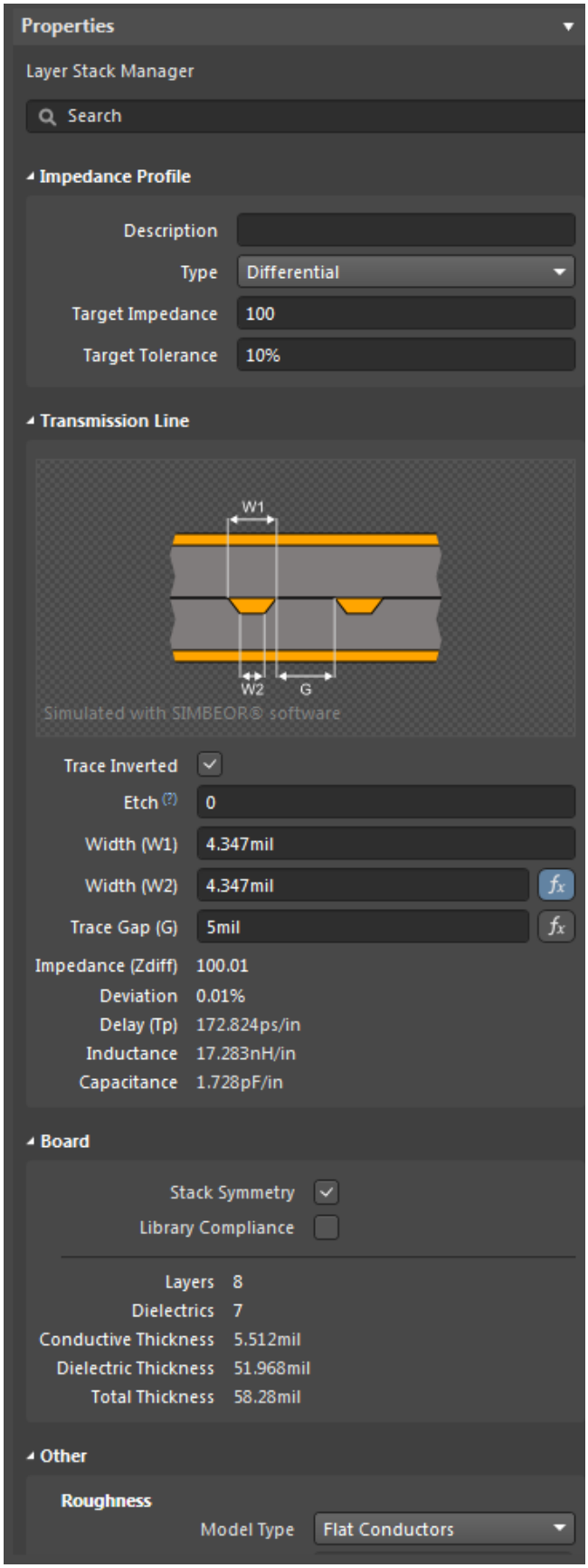


Figure 53. Impedance simulation of an inner copper plane of the card. Visible is the different propagation velocity from the previous figure.

In order to minimize reflections, vias have to be kept at a minimum. Their quite extended use is granted for sure in this case though, due to the limited number of layers. Even then, the vias are used on a same amount on each signal of each signal group that means whether all signals or none of a group are routed through vias.

In these frequencies, every single mil of length matters, and in order to meet the timing specs as much as possible, the intra-chip trace delays were taken into account. In a BGA package, the die is usually much smaller than the whole chip, so signals have to travel a not negligible amount of length from the package balls. This adds delays, sometimes reaching as much as 100ps, but the important thing here is that these delays are not uniform. For example, DQ0 may have an intra-chip delay of 40ps while DQ4 may have an intra-chip delay of 90ps. This makes for a 50ps difference, which otherwise would not be taken into account. XILINX provides the package delays for all the nets, so they are taken into account into all the calculation according the trace delay times.

In order to ensure data integrity on the Address / Clock / Command line, termination resistors are used in the end of the traces. The termination is used in order to avoid reflections in the end of the line. The suggested impedance for these lines are 40Ω. Since the least impedance that could be achieved with this layer stack-up is around 47 Ohms, 47 Ohms termination resistors are used. If no termination exists at the trace end and it is an open circuit, the signal effectively gets reflected back to the transmitter, causing interference and harming signal integrity. By placing termination resistors, the signal trace looks like being electrically infinite in length so no reflection problems occur.

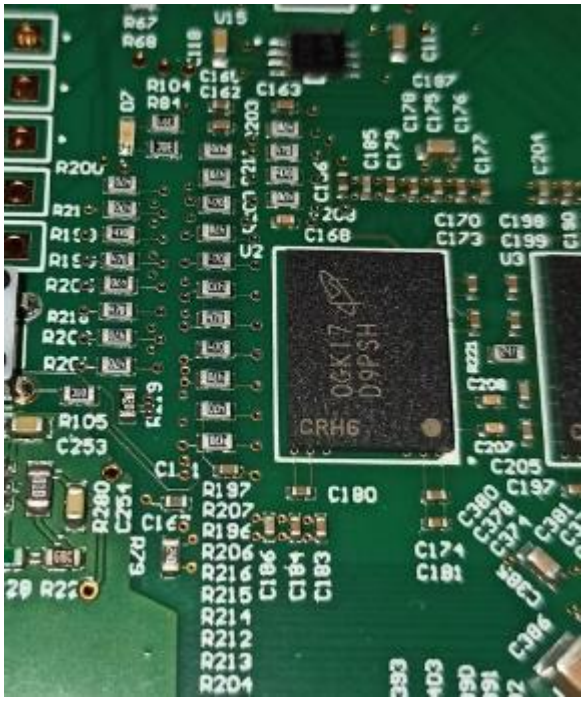


Figure 54. Termination resistors placed right after the last DRAM chip of the Fly-By chain.

The I/O pins of the memory controller on the ZYNQ device are fixed in placement, so they lack the flexibility of usual FPGA design, where the designer picks pins for each signal according to the routing needs. However, all are placed in one side of the device package, in order to assist the routing and the signal integrity. Furthermore, one is still able to use common techniques on DDR routing, such as data pin swapping, inside a specific byte, according to the layout's needs. Pin swapping is used extensively in this board, since routing space is limited in the first place.

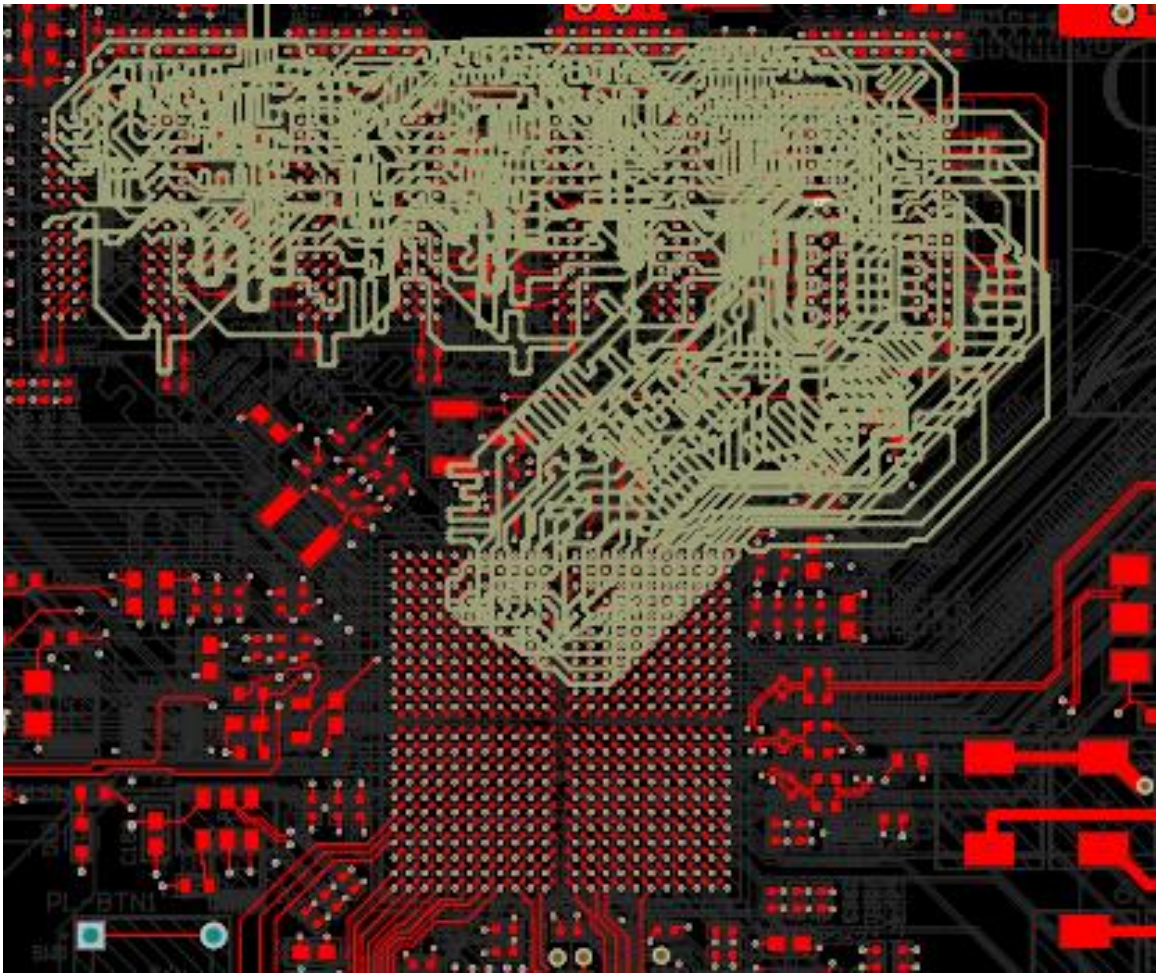


Figure 55. Highlighted are the Address / CLK / Command lanes. Visible is the Fly-By topology, with every signal essentially passing from every RAM chip.

The signal integrity can be verified by taking eye-scans on the memory bits, which functionality is embedded on the memory chips, and the ZYNQ's memory controller. Signal integrity can be assisted by write-leveling, a feature of DDR3 SDRAM, which allows the memory controller to compensate for the skew in DQS, by making small internal delay increments until it finds the correct delay.

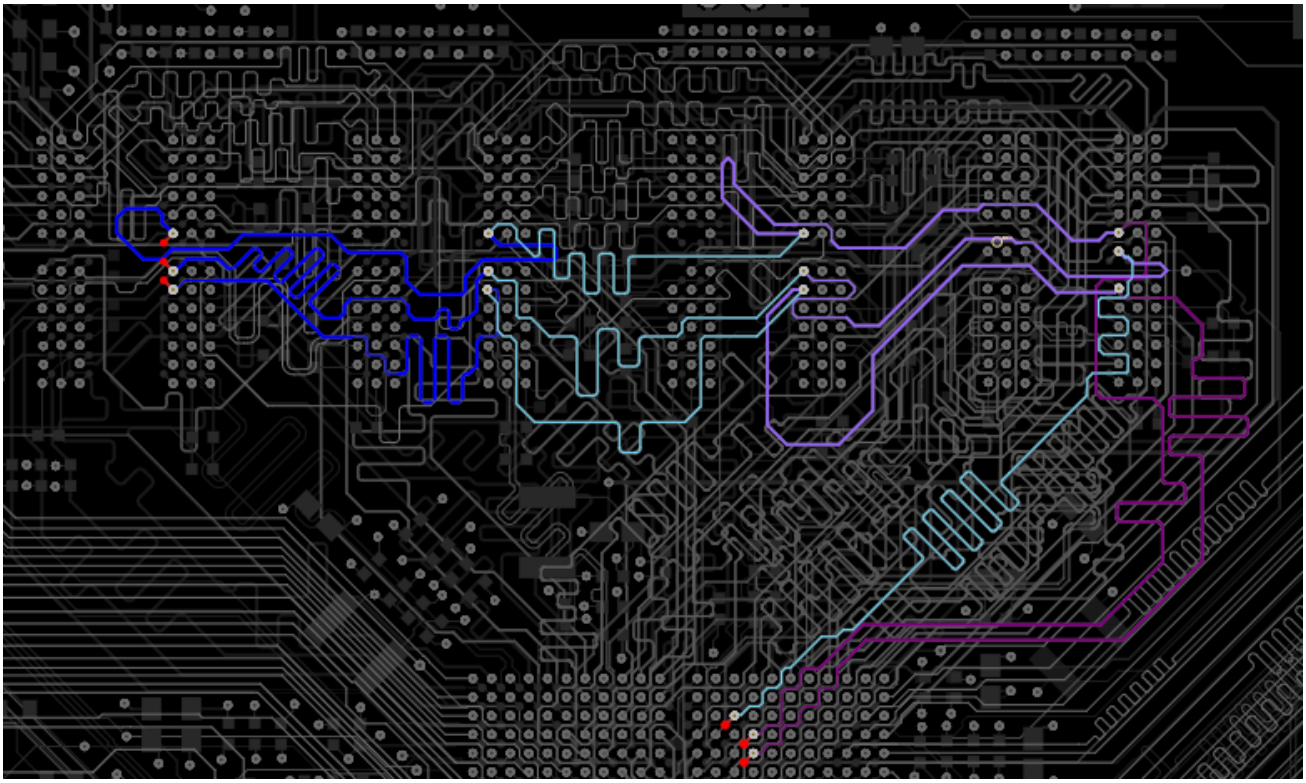


Figure 56. WE, RAS and CAS signals following the Fly-By topology.

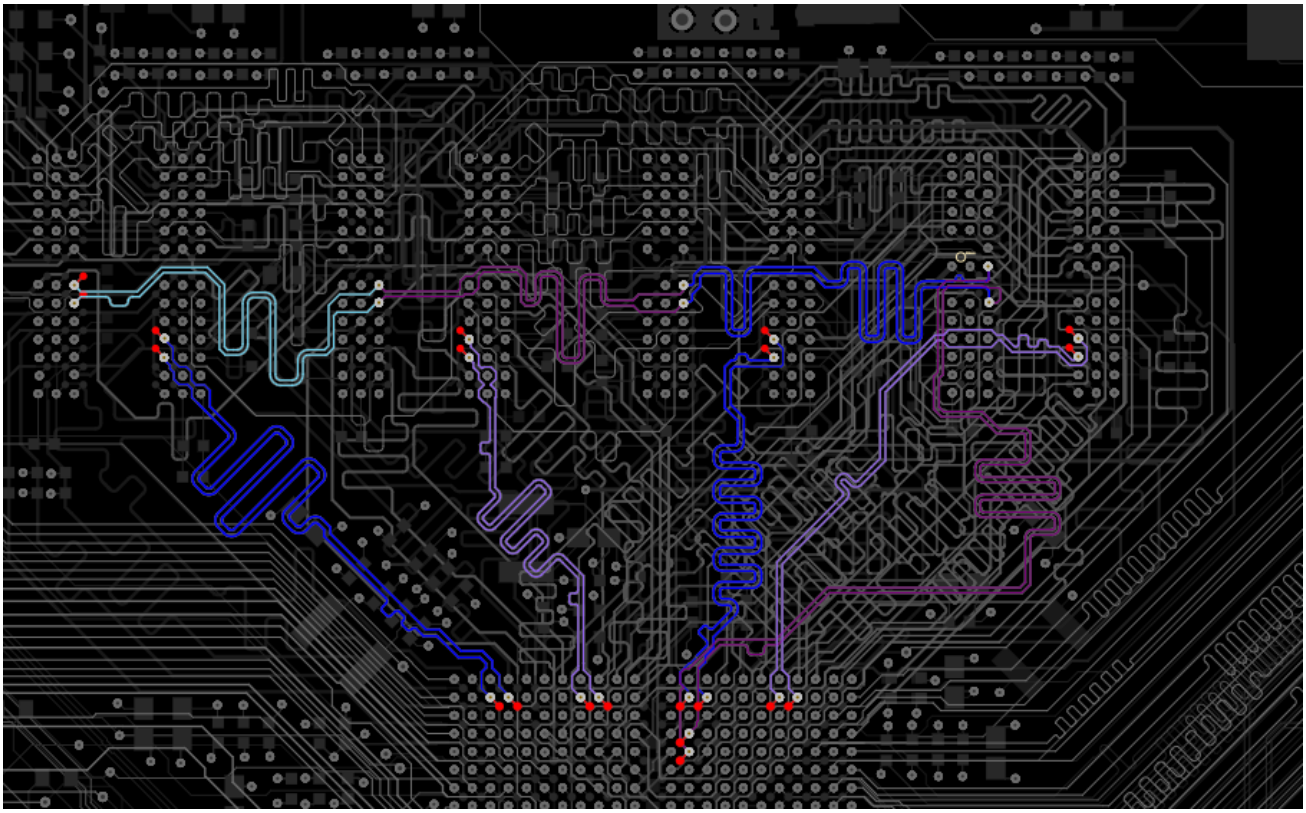


Figure 57. The Clock signal and the Data Strobes differential pairs.

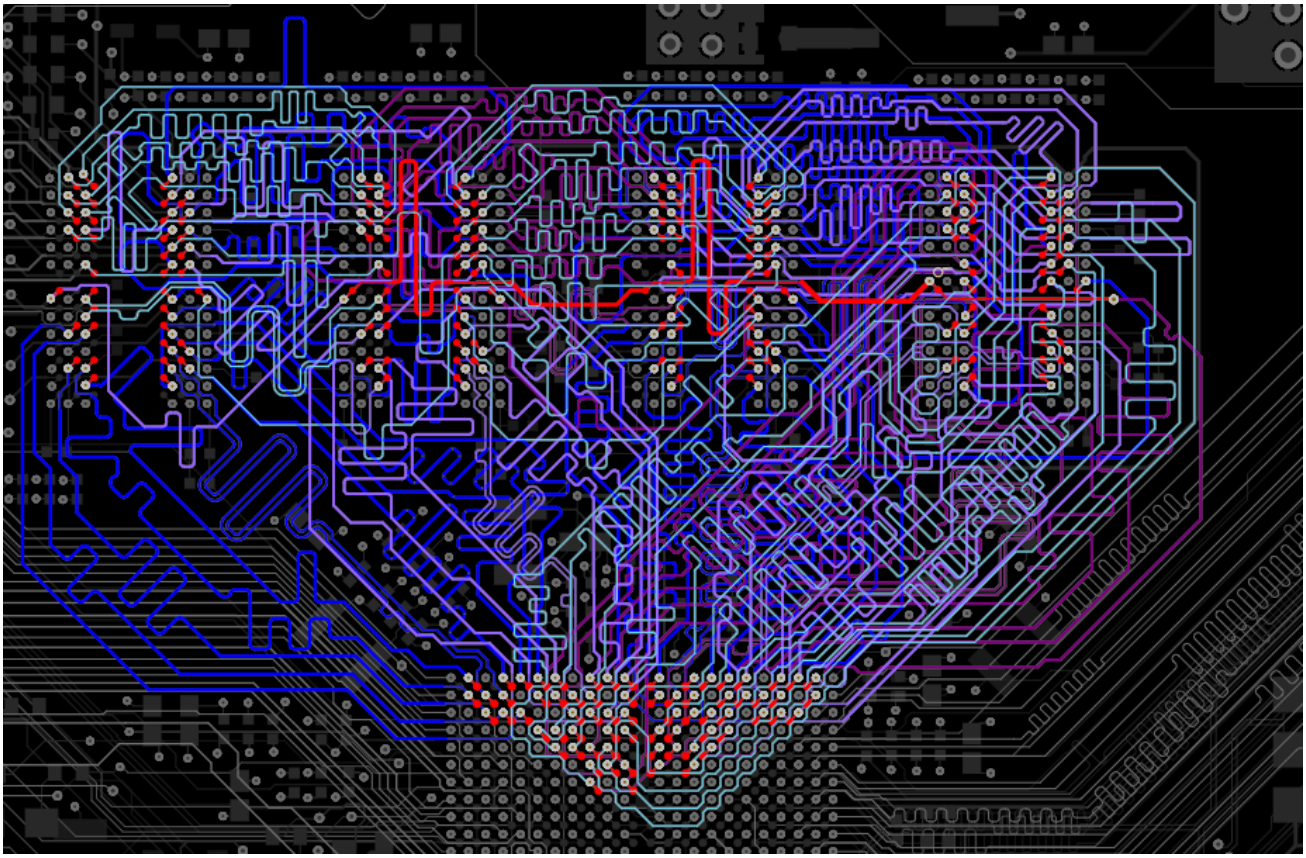


Figure 58. RAM interface signals.

4.7 Multi Gigabit Transceivers

The needs for data transmission are rising continuously, so the desire for faster data transmission always exists. Data transmission needs a medium in order to take place, in which the data are transmitted. A data transmission line essentially consists of a conductor, on which one applies a voltage on one end, and this voltage is read out on the other end. For an elementary digital transmission system, one has to either apply (1) or not apply (0) voltage on one terminal of the conductor, and on the other end interpret the existence of voltage or not as a bit. How many times one can change this voltage and be read out correctly on the other end is the data rate of this system. There are two main things one can do in order to increase the data rate. Whether increase the frequency with which the voltage changes, or bring in more conductors in parallel to transmit more bits at the same time. For example, by having 8 conductors in parallel, one now transmits 1 byte per period, in contrast to one bit if there was only a single line. Of course there are other methods to increase the information encoded in a period, such as AM (Amplitude Modulation). As an example, if one had 4 distinct signal levels, then 2 bits could be encoded as there now exist 4 states (00, 01, 10, 11) as opposed to two states (1, 0). Today's data transmission systems are a superposition of all these principles, and the state of the art of these systems is usually found at the MGTs of FPGA devices.

Going parallel is not the best option if one looks for very high performance. Sure one can use a lot of lanes in parallel (e.g. 384), and using only a moderate clock rate achieve remarkable performance. But the routing effort for such a high number of lanes is immense. Also, packing all these lanes would create remarkable interference and noise problems. And in the end, the the decrease in operating frequency could outweigh the advantage of the wide bus. The biggest problem though lies in synchronization. All these data bits are transmitted in parallel, with the clock travelling with them and are expected to reach the destination at the same time. However, some skew between the lanes is unavoidable. No matter how well one designs the PCB in terms of length matched traces and noise susceptibility, the higher the number of parallel lanes, the more difficult it is to avoid some skew. If the frequency increases, the specifications for the maximum skew become even stricter, because the clock period is now smaller. That is a big problem which limits us from using arbitrarily wide buses.

Going back to the first scenario, let's use only a single lane and increase the frequency. The routing has now become much easier and the frequency can be increased a lot, but there is still a problem. The clock has to be carried in parallel with the data, because even if the same, very precise oscillators are used in both the receiving and transmitting ends, there will be a difference in the frequency of the oscillators. A small difference but existent. This means that after some cycles the two frequencies will be off by an amount big enough to cause corruption on the data received. So there is still one limiting factor: the skew between the clock and the data lane. Of course now it allows for much higher frequency, but again we hit a brick wall.

The solution is simple: Integrate the clock inside the data. In this way there is no skew between different lanes to care about, and the maximum operating frequency of the link is now dependent on the electrical characteristics of the transmitter, the receiver and the transmission line. MGTs use this principle in order to achieve very high data rates. Then one can employ more channels in parallel, now though synchronizing them through the protocol used. These lanes are differential to assist in signal integrity and noise immunity, and now the skew between the positive and the negative signal of the differential pair of the transmission line are of interest and of course is not as big of a deal to get right as dealing with the skew between two or more different lanes. The clock is integrated in the data by using specialized scramblers and data protocols.

Of course though, FPGA logic cannot handle signals toggling at multi-gigabit rates, but, on the other hand, FPGAs are very good in handling large parallel buses. For this reason an essential circuit of every MGT is the SerDes (Serializer - Deserializer). Its task is to take a rather wide stream of data (such as 80 bits wide) and transmit them one by one. On the other end of the link, the DeSerializer does exactly the inverse, transforming the fast serial stream into a relatively slow, parallel bus. MGT SerDes are hardwired blocks in the transceiver circuitry, and usually cannot be used by user logic (though there exists a special kind of SerDes that lies on the non-MGT I/O pins of the FPGA, usually however handling much lower rates such as 1Gbps maximum).

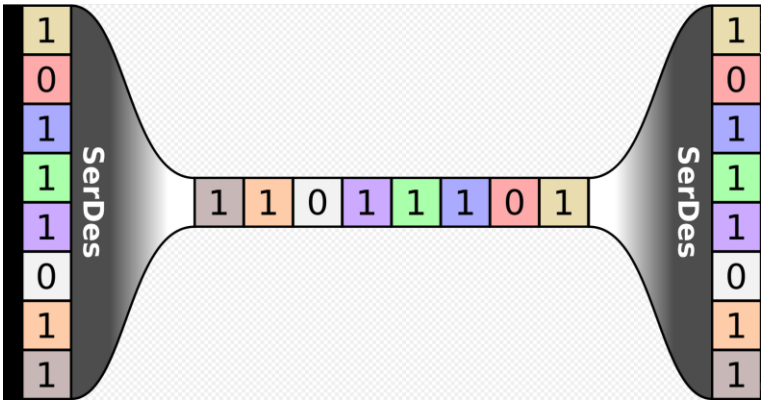


Figure 59. Depiction of SerDes (Serializing on the left, DeSerializing on the right).

4.7.1 MGT technologies

The MGTs and the protocols over them incorporate a number of technologies in order to function. Some of which are listed below.

- Clock Data Recovery (CDR)

  This is one of the most important circuits and is in charge of extracting the clock out of the incoming data by detecting transitions. Using the reference clock provided, a CDR always tries to sample in the middle of the data "eye". By using this technology serial links are enabled to function reliably over a long distance.

- Phase-Locked loops (PLLs)

  The CDR uses a reference clock in order to lock to the clock encoded in the data stream. This reference clock needs to be of high quality, high stability and low jitter in order to support the CDR. However this reference clock is supplied externally, and its frequency is much lower than the link clock. For this reason exist embedded PLLs into the

transceiver circuitry which multiply the reference clock to a frequency close to the link clock, so that the CDR can use it as a reference.

- Encoding/Decoding

In order to have a reliable link we have to ensure that there are enough transitions for the CDR to lock on to. Transmitting a sequence of data that consists of only '0' bits for example, would cause the CDR to lose lock to the link. For this reason special encoding schemes are used that ensure that no matter what the input data stream is, the output data stream will always have enough transitions. These encoders are known as scramblers, and the respective decoder (de-scrambler) exists on the receiving side. Another problem that may appear without the usage of a scrambler is that since most high speed links are using ac-coupling, sustained transmission of 1 or 0 may lead to capacitive loading of the line, causing less than ideal transition swings on the receiving side. An example of such encoding/decoding schemes is 8B/10B which takes as input 8 bits and outputs 10bits, with the drawback of ~20% data overhead for the encoding. Another example is 64/66 which takes 64 bits and outputs 66 bits and in this case the encoding overhead is much smaller, at 3%. Such protocols also include some special reserved words, used for control.

- Differential Signaling

Single ended signals are too vulnerable to noise to be used in such applications. So the electrical interface is low voltage differential, and in this way half of the voltage swing is needed for a transmission, and since most of the times the noise is common to both of the conductors it is being rejected. To be able to provide such fast transition rate specialized differential drivers are used.

- Emphasis

  Since transmission lines form a natural low-pass filter, and this phenomenon is increased with frequency, fast MGT signals tend to be attenuated a lot. To compensate for this, a technology implemented is transmitter emphasis. The transmitter transmits at a lower impedance signals that due to their frequency are to be attenuated the most. Emphasis on the transmitter side is called Pre-Emphasis. The receiver then can use the inverse of this procedure to de-emphasize the signal and reproduce the original waveform.

- Equalization

  The receiver can amplify certain parts of the signal spectrum, which will be the most attenuated in order to construct a waveform clean enough for the CDR and the data sampler. This process is called equalization and is implemented by various mechanisms in the transceiver circuitry.

- Termination impedance

  For the best signal integrity the impedance of the transmitter, the transmission line and the receiver must match. Since no matter the effort put by the designer, miniscule variations in the PCB and in the parts exist, the MGTs have the ability to slightly calibrate the termination resistance internally in order to match the link's impedance.

- Data alignment

  The receiver cannot know where a word begins. This is being solved by including certain words in the data stream, which the receiver identifies. By using these special

words, the receiver can know where a word starts, and in certain intervals these words are inserted again in order for alignment to be maintained.

- Channel bonding

MGTs transmit data at very fast rates, but even then this may not be enough. In order to increase the throughput of the system, one can combine many of these serial channels in parallel to make a higher throughput link. The problem again is the skew, because then every channel needs to have cabling of the exact same length. This problem is solved in the logic. The transmitters, in the same time, transmit a certain alignment word or sequence. This is being identified by the receiving logic, and the received data, stored in FIFOs, are being offset in order for all the channels to be aligned.

4.7.2 Transceivers on the XC7Z015 device

The specific ZYNQ device that is used on this board features 4 GTP transceivers, in one quad. GTP transceivers have a maximum operating frequency of 6.6 Gbps on -2 and -3 speed grade devices, and 3.125 Gbps at -1 speed grade ZYNQ devices. Each 4 transceivers are grouped into a quad, and since only 4 trasnsceivers exist on the device, there is only one quad. There are 2 inputs for reference clock in each quad, which can be multiplexed and the user is free to choose which reference clock will be fed to each of the PLLs. There are 2 PLLs, and being 2 comes in handy in cases where one need for example a different frequency for the transmitters and a different frequency for the receivers.

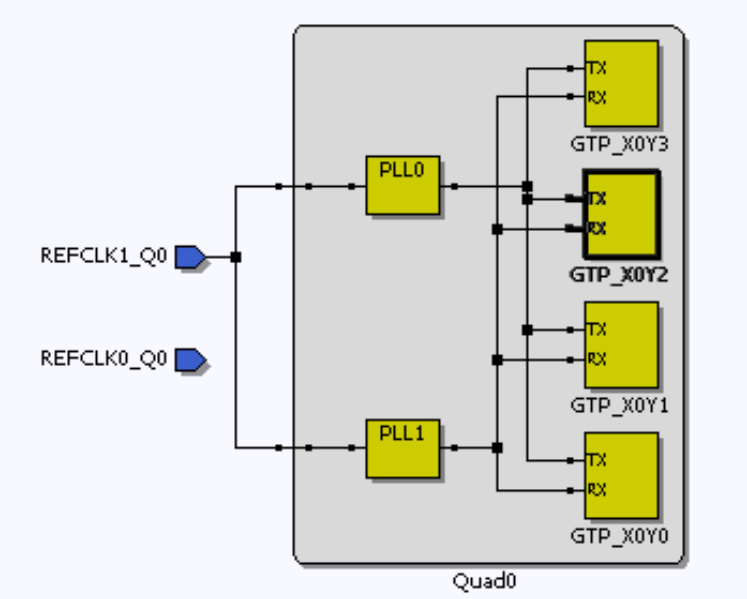


Figure 60. Clocking network of the MGT quad.

4.7.3 MGT internal structure

The internal datapath of the transceivers can be up to 20 bits. Hard blocks for 8B/10B encoding is included in the transceivers, so no additional logic has to be wasted, and also 64B/66B and 64/67B encodings are supported. Included is a PRBS (Pseudo – Random Binary Sequence) generator and checker block, which is very useful for debugging puproses. All the above are included in the PCS (Physical Coding Sublayer) block of the transceivers.

The other block of the transceivers is the PMA (Physical Medium Attachment), that essentially contains the Front-End of the transceivers. PMA includes the PLLs, the multiplexers for the clocking options, equalization such as DFE and LPM, emphasis mechanisms and the eye-scan capture circuitry.

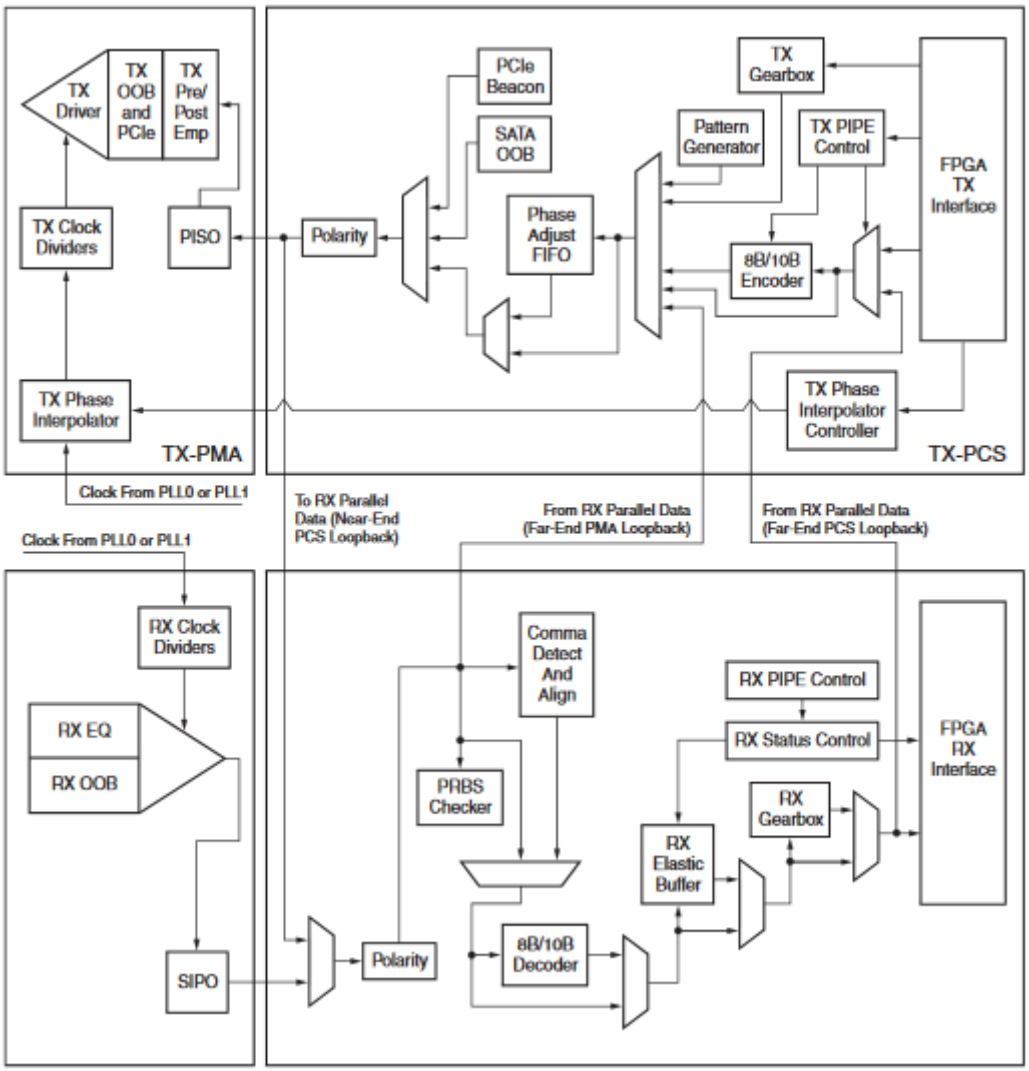


Figure 61. Ug482, p. 15. The Tx/Rx PCS and PMA blocks.

4.7.4 Characterization

The main charectirization of serial links is the BER (Bit Error Ratio). This essentially defines how many bits are received incorrectly out of the number of total bits received. This is usually done by using the PRBS generator to generate a known stream of data, and then compare this stream of data to the expected words, since the sequence can be predicted. If an error is identified, then the errors accumulator increases by one.

Another way to characterize the performance of a link, and especially its jitter performance is by using the eye-scan functionality. An eyescan is obtained by using an oscilloscope to probe

the link, with persistence enabled to infinite, for a long time, and capturing all the waveforms in a single picture. This gives information about the shape of the pulse, the rise/fall time and the jitter. In the middle of the eye is where the sampler should ideally take the data samples.

GTP transceivers have this functionality built-in. The way the eye-scan is taken is by having two independent samplers. One is the data sampler, which samples always in the middle of the eye, as perceived by the CDR. The other sampler is the offset sampler, and can be moved in both dimensions of the eye scan (e.g. horizontally (in time), and in height (in voltage)).

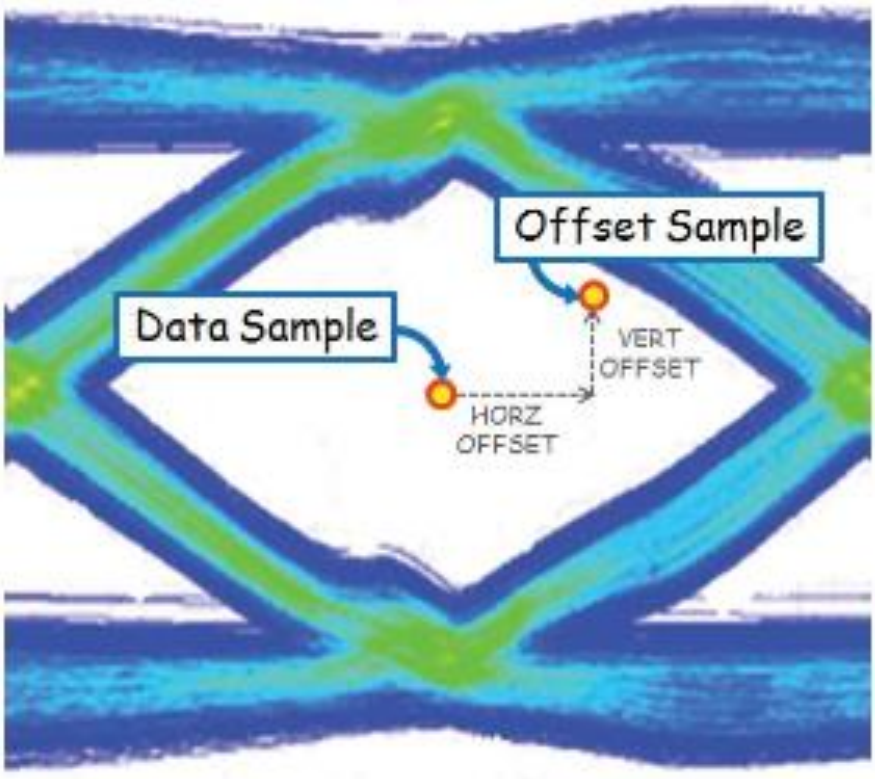


Figure 62. Eyes can functionality. The concepts of the data sampler and the offset sampler are visible.

The eye-scan functionality, with many other elements, is accessible from the DRP (Dynamic Reconfiguration Port) of the transceiver. This port is asynchronous to the transceivers and can be used to manipulate registers in the transceivers, access the eye scan functionality, tweak the values of the PLLs, reset portions of the Tx/Rx and much more.

Another useful debugging technique is the loopback. In loopback mode, the receiver receives the data of the transmitter of the same channel (or of a different channel in external mode). The data reversal can happen in several regions, for example, in PCS loopback the data never leave the transmitter PCS and get transferred directly to PCS of the receiver. There also exists the PMA loopback, in which the data are taken from the front-end of the transmitter and travel to the PMA of the receiver, and the Far-End loopback mode, in which the data enter another device and their direction is inversed in there (in PCS or in PMA as well).

4.7.5 MGT implementation on this board.

The reference clock used in this case is an Si570 Programmable oscillator [17]. Si570 offers very high performance, identical for applications such as MGTs and is programmable with a frequency ranging from 10MHz to 945 MHz and <1 ppb resolution. The programming is done via I2C, and the user manipulates some registers that set the multiplication and the division for the PLL. In order to find these coefficients, the user has to implement an algorithm in order to find the correct solution for the required frequency. The default oscillator frequency is programmed by the factory and can vary according to the user's needs. In this case it is 156.25MHz, a frequency which is handy for the PLLs in the MGT quad to produce an output of 6.25GHz. The output of the oscillator is differential, and feeds directly one of the two reference clock ports of the MGT quad (MGTREFCLK0).

The second reference clock input of the MGT quad is connected to a pair of SMA connectors, on to which can be connected an external differential clock to be supplied to the transceivers.

Figure 63. The Si570 oscillator.

Of the 4 MGT channels on the board, 2 are exposed to SFP interfaces, and two are exposed to SMA connectors. SFP (Small Form-factor Pluggable) is a compact, hot-pluggable, modular interface for networking applications, usually connecting a fiber-optic cable and sometimes copper cables. SFP+ allows for speeds up to 10 Gb/s and is what is used on this board. Other specifications exist, such as QSFP-DD which allows for data rates up to 400 Gb/s (however utilizing 4 channels, instead of 1 in SFP+). The SFP module gets inserted in what is called the SFP cage, which is a metallic enclosure which offers EMI protection and casing. In the end of this cage is the connector, with copper pads, on which the SFP module attaches to.

The SFP transceivers can be configured using an I2C bus interface. For this reason, both the SFPs are connected to the I2c bus that originates from the hard I2C peripheral in the ARM processor. However, since both SFPs appear as having the same I2C address, the bus is multiplexed from the I2C mux. Furthermore, some signals of the SFPs are connected to pin headers above the SFPs, and can be useful for debugging. The power to the SFP modules passes first from choke inductors, in order to decrease any noise coupling with the rest of the system.

Figure 64. A fiber optic SFP module

Optics have many advantages over copper cable. The data, instead of being transmitted through electrical pulses, are now being transmitted with pulses of light. The light then, by taking

advantage of the total internal reflection inside the optic fiber cabling, can travel long distances, in the order for 100s of Km, with very little attenuation. Using light for transmission makes the system immune to Electro Magnetic Interference. Furthermore, the transmission line has no capacitance, hence no negative effects take place on high frequency signals. Also, there is practically no limitation to the bandwidth available to the user, since the frequency of the light that is used is usually very high (close to the visible spectrum).

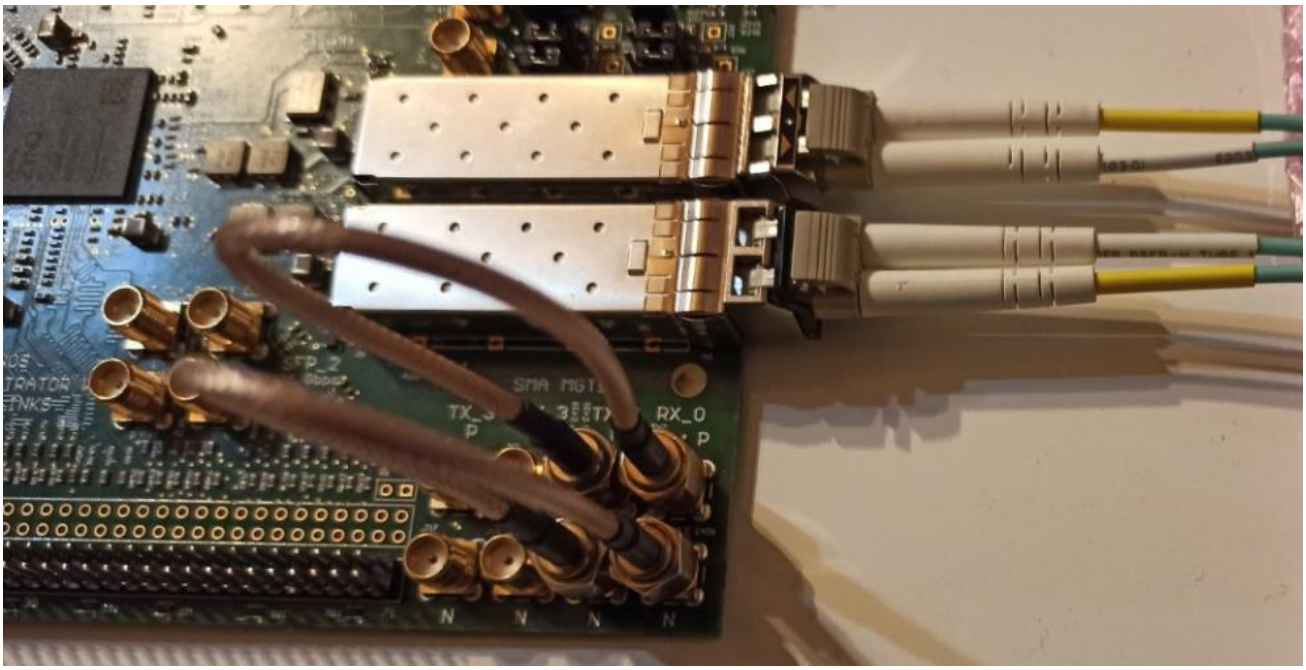


Figure 65. The SFP optical assembly

The other two channels are being exposed to SMA connectors, on to which SMA, shielded copper cables can be attached. The main advantage of this is the simplicity, since no extra modules are required, other than a pair of cables. Of course this solution carries all the disadvantages of the copper cables, and though it can be used for high data rates, the price of the cables goes up since manufacturing of the cable gets more difficult. Even then, long distances cannot be supported reliably by copper transmission on such high frequencies.

Figure 66. The MGT SMA connectors on the board.

Routing of the MGT traces from the FPGA to the SFPs and the SMAs is of vital importance for the performance of the links. Even the slightest skew has to be avoided in the Positive/Negative pair of the channel, the impedance of the differential traces need to be very close to 100 Ohms and the dielectric should not show high losses in the operation frequencies. To define the impedance with accuracy, the designer needs to make sure that adjacent to the MGT traces is at least 1 (above or below), or two (above and below) coherent copper planes which will be used as reference in the calculations. The voltage on these planes does not have to be 0, but has to be DC, and these planes need to be able to act as an effective return path for the AC current. For this reason, all the way along the MGT traces, extensive coupling has been used between the planes and the ground. For high-speed AC currents, this acts as a return path to ground.

3	POWER 1	CF-004	Signal	1/2oz	0.689mil		
	Dielectric 2	PP-006	Prepreg		9.449mil	4.1	0.02
4	MGTs 1	CF-004	Signal	1/2oz	0.689mil		
	Dielectric 1	Core-009	Core		7.874mil	4.04	0.02
5	GND	CF-004	Signal	1/2oz	0.689mil		
	Dielectric 3	PP-006	Prepreg		9.449mil	4.1	0.02
6	MGTs 2	CF-004	Signal	1/2oz	0.689mil		
	Dielectric 5	Core-009	Core		7.874mil	4.04	0.02
7	POWER 2	CF-004	Signal	1/2oz	0.689mil		

Figure 67. PCB Layer Stack for the MGT layers.

The MGT channels in this board are routed in two different layers. On one layer are the transmitting channels, and on the other layer are the receiving channels. The first copper layer on which transceivers are routed, has above it the Power Layer 1, a copper pour of the 3V3 rail. Below it, is the ground layer. Both of these layers on this region have no discontinuities. The second copper layer on which transceivers are routed has the ground layer above, and below it lies the Power Layer 2 carrying the MGTAVTT voltage. Again, no discontinuities exist over this region.

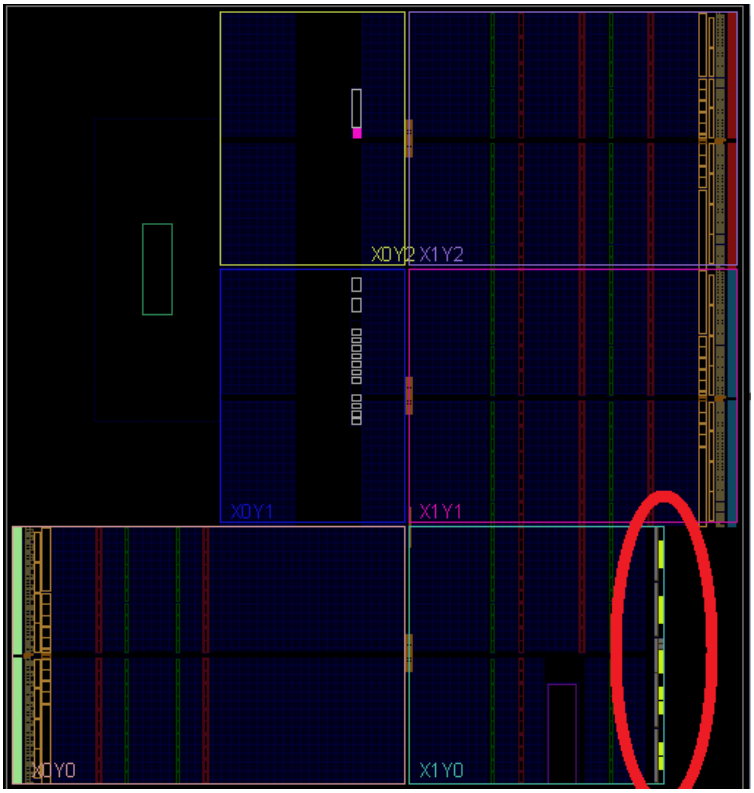


Figure 68. The MGTs location on the device die.

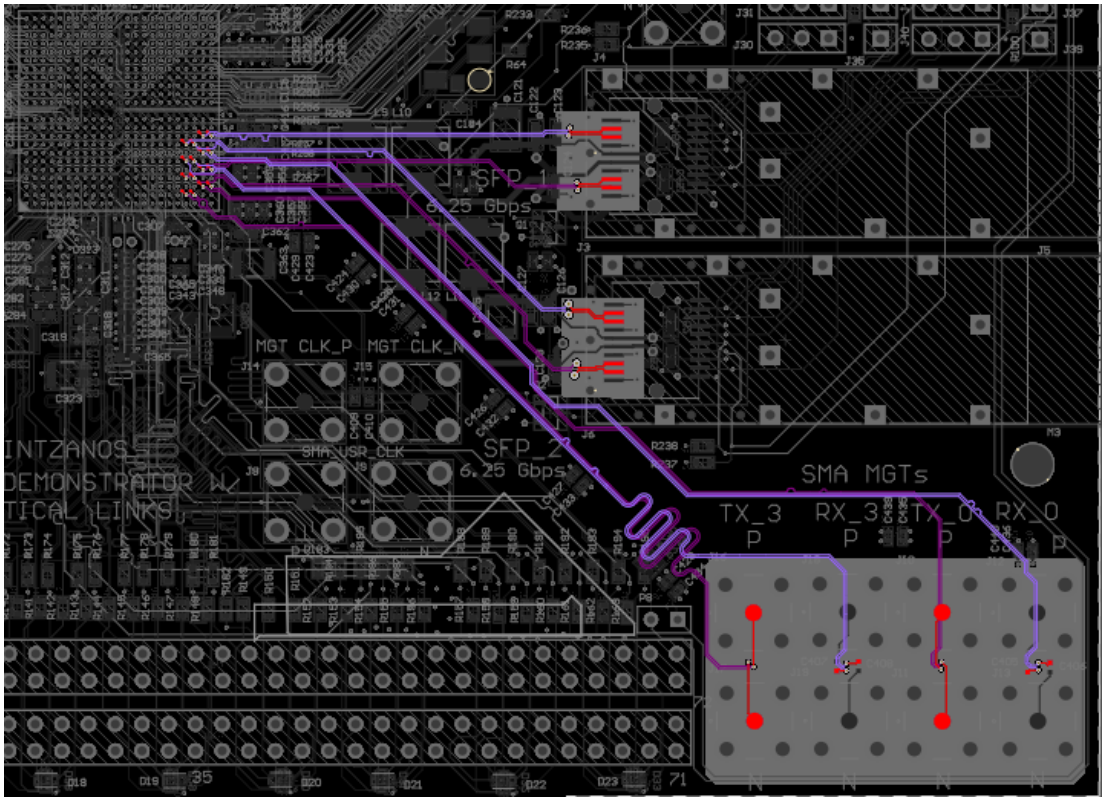


Figure 69. The MGT signals on the PCB.

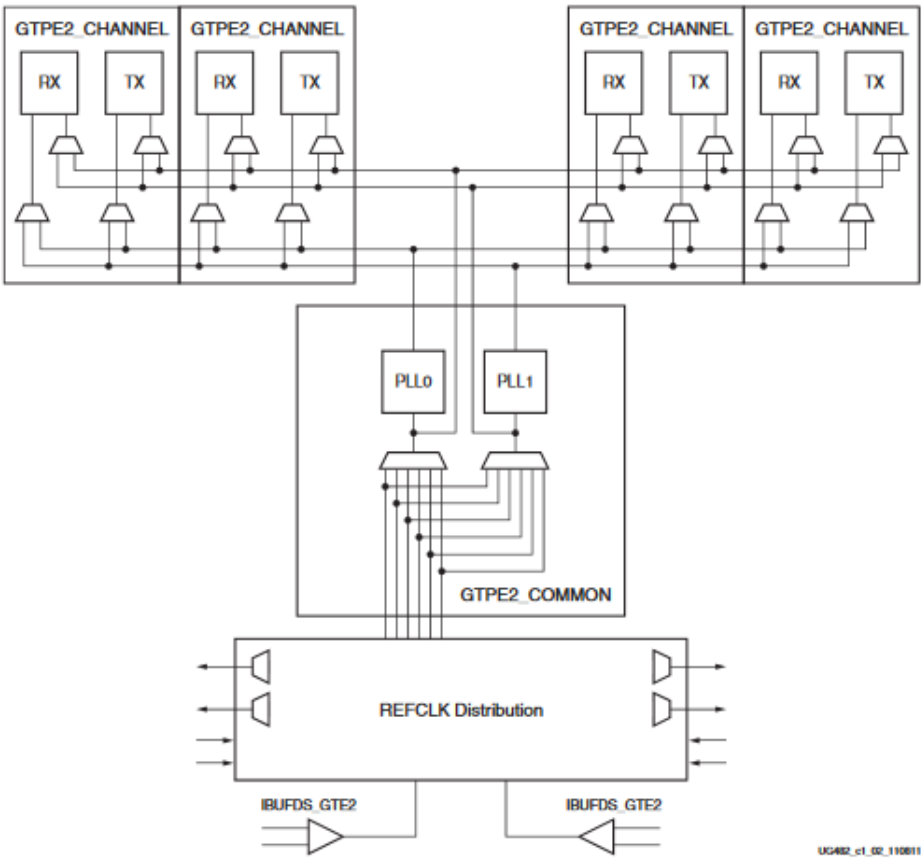


Figure 70. Reference clock distribution inside the MGT quad.

4.8 Ethernet

4.8.1 Ethernet technology

Ethernet is one of the most common networking technologies used nowadays. It is found from Personal Computers to embedded systems due to its versatility and flexibility. It was commercially introduced in 1980 and was standardized in 1983. Ethernet has uses ranging from bulk data transfer, equipment control, to carrying the internet protocol.

An Ethernet frame is called a "datagram". A datagram contains the preamble (a known sequence to synchronize the receivers to the data transmission), the SFD (Start Frame Delimiter), the frame header, the payload and the CRC (Cyclic Redundancy Check). The header contains information such as the source and destination MAC addresses, the protocol type (TCP or UDP), and the length of the payload. The payload carries the data and its size can vary between 42 and 1500 bytes. Some Ethernet MACs (Media Access Controllers) support jumbo frames, which can be up to 9000 bytes in size. The CRC is 32-bits wide and used by the receiver to check if there has happened a data corruption during the transfer.

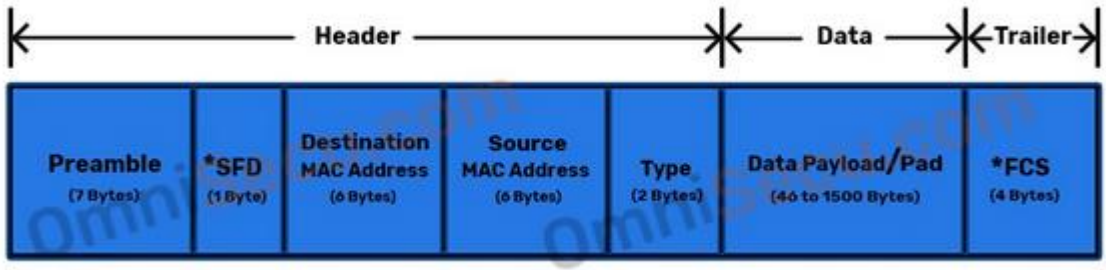


Figure 71. An Ethernet frame.

The responsible system for the correct assembly of the Ethernet frame is the MAC. ZYNQ-7000 devices contain two hard MACs in the Processor Section, capable of speeds up to 1 Gbps. Allowing Ethernet connectivity to the ZYNQ can assist tasks such as data transfer from the FPGA to a PC (for which 1Gbps is a enough for many applications), board to board

communication, gain access to the ZYNQ though the local network, especially if there is an Operating System running on the ZYNQ, and finally connecting the ZYNQ to the Internet.

Ethernet communication is usually carried by copper wires with most common the UTP (Unshielded Twisted Pair) wire. UTP wire consists of 4 wire pairs, of which 100BASE-T Ethernet (100 Mbps) uses 2 pairs, one for each direction, at a maximum distance of 100 meters per specification. 1000BASE-T Ethernet (1 Gbps) uses all 4 pairs, with PAM-5 (Pulse Amplitude Modulation) to increase the symbol rate for a maximum distance of 100 meters. UTP cabling exists in many categories (Cat), such as Cat 5, Cat 6 and Cat 7), with an increasing cost as category increases. Both 100BASE-T and 1000BASE-T need at least Cat5 cables in order to function properly.

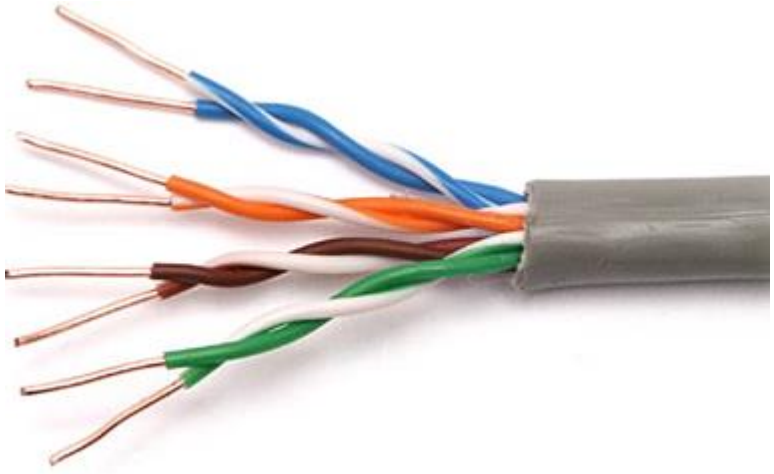


Figure 72. Twister Pair (UTP) cable.

Having an embedded MAC into the ZYNQ device is very useful since otherwise one would have to implement it using logic resources in the FPGA, but it is not enough alone to achieve Ethernet connectivity. For this, one needs a PHY, which is a device that is in charge of converting the logic signals to signals capable of travelling adequately on the UTP cable. 1000BASE-T signals are differential signals with a voltage swing of +- 2.5V. All four pairs of the UTP cable are used and every pair is bidirectional instead of serving only one direction as in 100BASE-T. PAM-5 modulation allows for sending multi-bit symbols in one clock cycle, due to different voltage levels allowed for signaling. The link frequency is 125MHz, and since 2 bits can be transmitted per clock per lane, results in 125 Mbaud per lane or 250 Mbits per

lane. So in total 8 bits per clock cycle are transmitted, accounting for the four lanes, and 125 MHz x 8 bits = 1000 Mbits/s.

Since an FPGA is not capable (some are, but for much higher performance applications) to transmit in PAM-5, the PHY does all this work. Another important task of the Ethernet PHY is the auto-negotiation. Since Ethernet is generally backwards compatible, connecting a UTP cable between two devices does not say anything about the maximum data rate supported by the link. One device may support Gigabit Ethernet while the other one may support only 100Mbps. Since it is not only a matter of adjusting the frequency in order to trim the data rate, a special process of auto-negotiation has to happen. The PHYs of both sides agree to the maximum supported speed of the link, and settle on this.

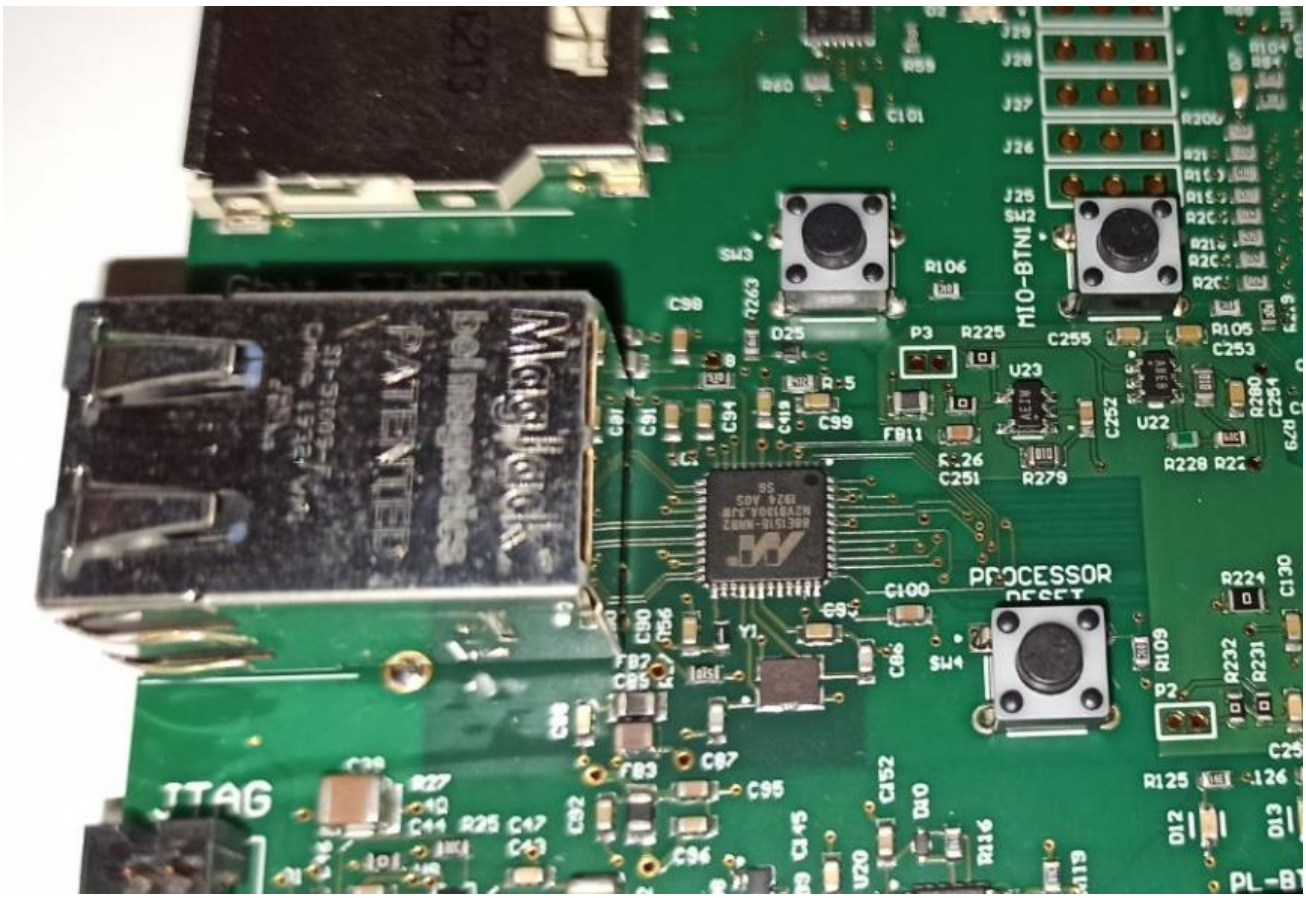


Figure 73. The Ethernet PHY and RJ-45 Jack on the board.

The connector for the UTP cable is the RJ-45 connector. The jack for these connectors often feature built-in leds, informing the user for Link-Up status, and/or instantaneous transmit/receive. The signals then are AC coupled from the PHY to the line, and this happens with special magnetics. These magnetics may be embedded in the connector or placed on the PCB board.

4.8.2 This Ethernet implementation

The PHY controller used for this board is a Marvell Alaska 8E1518 [18]. Alaska 88E1518 is an integrated 10/100/1000 Mbps Energy Efficient Ethernet Transceiver supporting RGMII Interface (Reduced Gigabit Media-Independent Interface). It is manufactured using standard CMOS process and contains all the active circuity required to implement the physical layer functions to transmit and receive data on standard CAT 5 cable. The MDI termination resistors are integrated into the IC, so the board components count is reduced.

The power supply requirements are a single 3.3V supply, and the on-chip regulator is capable of producing the core voltage. However, in this case all the voltages are supplied externally to the chip, using choke inductors to contain any RF noise. These voltage are ETH_1V0, derived from the VCCINT rail, analog circuitry supply AVDD, derived from the VCC1V8 rail, and the ETHAVCC, derived from the 3.3V supply rail on the card.

The PHY needs an external oscillator in order to derive the clocking for the interfaces, and that is usually a division of the 125 MHz clock. In this case, the oscillator's frequency is 25MHz. The circuitry is configured such as to reset the PHY every time a board reset happens, and to not allow the PHY to come out of reset, unless all power rails are up and steady.

The interface between the ZYNQ and the PHY in this case is RGMII, which is natively supported by the two hard MACs on the ZYNQ device. The pins for the RGMII interface are taken directly from the MIO pins of the ARM processor on the ZYNQ. RGMII consists of 4 parallel lanes for receive, 4 parallel lanes for transmit, a clocking signal, a control signal and the MDIO and MDC signals.

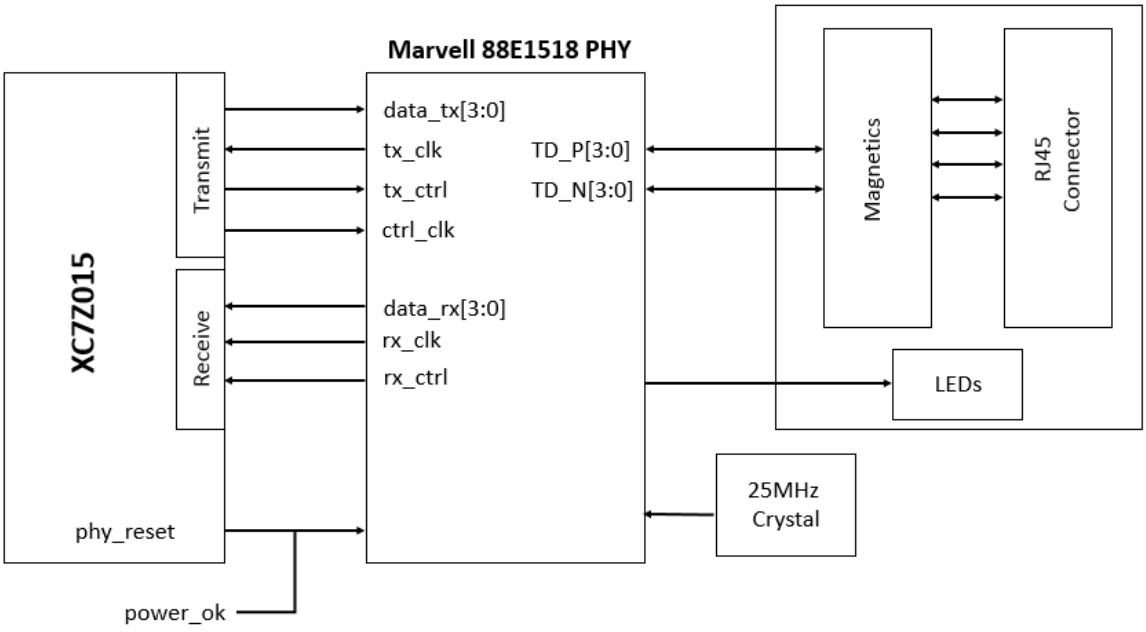


Figure 74. High-level diagram of the Ethernet interface.

Since the RGMII parallel interface is quite fast, the bus is vulnerable to skew. This many times can prevent the interface from functioning properly, especially if the lanes on the PCB are not length-matched as per specifications. In this case, the lanes are matched to within 50 mils, so no additional tweaking was required to get the interface up and running. However, in case that there is excessive skew on some of the lanes, there are variable delay taps on the PHY, so one can align the lanes. This is done by configuring the respective registers in the PHY from the controller (the ZYNQ, in this case).

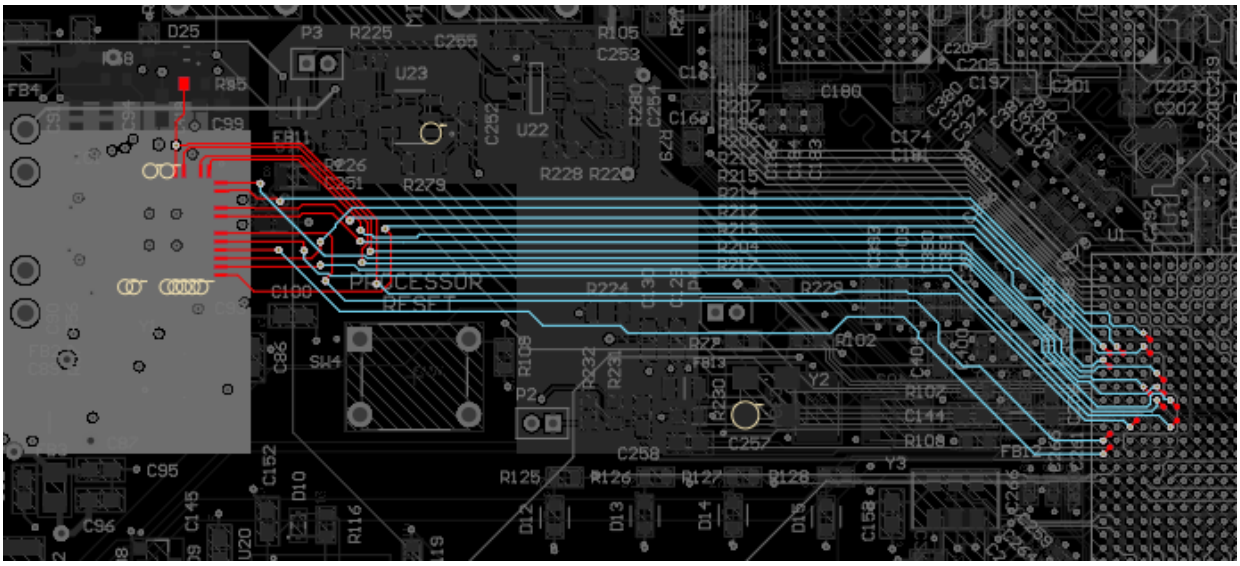


Figure 75. The RGMII interface between the ZYNQ and the PHY.

4.9 UART

4.9.1 Universal Asynchronous Receiver Transmitter

UART (Universal Asynchronous Receiver Transmitter) is a device used for asynchronous serial communication in which the format and data transmission rates are configurable. UART circuits are found today almost in all microcontrollers, in many embedded systems, even in Personal Computers, and it still retains its place among the peripheral circuits because of its simplicity and wide adoption. The bits are transmitted one by one, and each frame has its own Start and Stop conditions, such as the receiver can extract the timing information precisely. The frequency (usually referred to as the Baud Rate) of the channel has to be agreed beforehand from both the receiver and the transmitter. However, there are UART circuits that can extract the baud rate out of the incoming data stream. Due to the no need for clock, only one signal (usually single ended) is needed for the transmitting channel and one for the receiving channel. The frames are usually consisted from 1 start bit, 8 data bits and one stop bit, however several variations exist, such as including parity bits to implement a data correction mechanism.

UART is usually preferred for debugging purposes and for slow control or slow data transmission, since the baud rates supported are low by today's standards. Common baud rates for UART can range from 1200 bps to 115200 bps. However, many microcontrollers support up to 1Mbaud/s and custom implementations may reach even further. There are several signaling specifications for UART, with the most common one the RS-232 which specifies voltage swings of $\pm$12V. Such high voltage swings aid in noise immunity, since UART can quite commonly be found in industrial environments.

For the needs of this board, UART is very useful in debugging, especially since it can provide a means to communicate with the processor before for example an OS and a network connection have been established, so more sophisticated methods of communication can be used.

4.9.2 UART implementation on this board

The ZYNQ device features 2 hard UART circuits on the processor side, one of which is used in this board. Of course, implementing a UART controller in the Programmable Logic is trivial, but the ease that the hard controller provides is that it can be used no matter of the bitstream programmed to the FPGA. Of course, using RS-232 signaling is out of the boundaries of this board's use, so for the transmission of the UART data stream a USB (Universal Serial Bus) solution has been chosen.

The Integrated Circuit used on the board, that is in charge of converting the UART signals to USB signals is a Cypress CY7C46225 [19]. CY7C46225 is a UART to USB Bridge, with integrated USB transceiver, supporting the USB 2.0 specification. On the computer side, drivers for this chip are installed, so it appears as a COM port.

Since the bank on the ZYNQ that the UART signals are pinned out to is powered by 1.8V, there needs to be some voltage translation between this voltage level, and the 3.3V that the UART to USB Bridge expects to see. This is happening by using a TXS0102 [20], a 2 port level translator by national instruments.

Two LEDs are placed on the board, driven by the UART to USB Bridge, one indicating when there is data transmission happening, and the other one indicating whether there is data reception.

From the USB to UART bridge a differential pair of signals travels to the USB-micro connector placed on the top of the board. To protect the card from ESD events, an ESD protection diode array of ultra-low capacitance [21] to not distort the USB signals has been placed on the data lines. The diode is rated for up to 20kV contact or air ESD discharge, with a low clamping voltage of 9.2V.

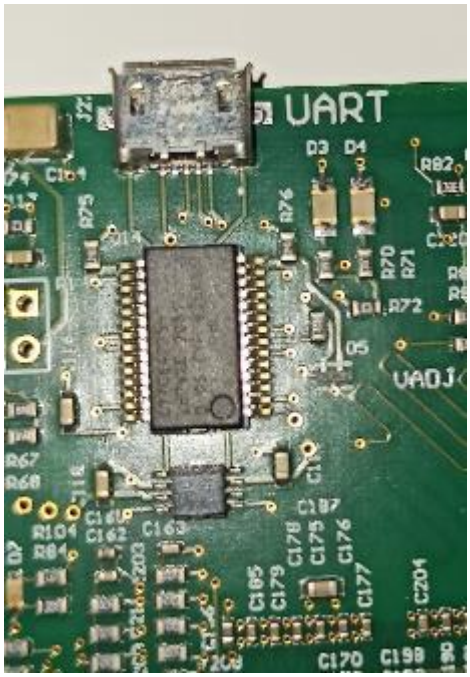


Figure 76. The UART circuit. On the top is the USB-micro connector, in the middle the USB to UART Bridge, and in the bottom the level shifter IC.

4.10 SD

In order to run an operating system on the board, not only RAM memory is required but also a storage medium to store the OS, the applications and the user's files. Is this board the storage medium is an SD (Secure Digital) card, which allows for moderately large storage space (>32GB), has low physical space needs and allows for moderate data transfer rates.

The ZYNQ device has integrated 2 hard SD card controllers, which can be used to load the processor ELF (Executable Linkable Format) configuration file, the bitstream and the OS. The board can be configured to boot from the SD card. One of these controllers is used on this board, and is connected to an SD card connector.

The SD card interface consists of a 4 bit-wide data bus, a command signal and the clock signal. The operating frequency of the ZYNQ SD card controller supports up to 125 MHz, so one has to pay care to the length and the delay of the lines, in order to avoid troubles during operation. The clock signal needs a 50 Ohm in-line termination. The interface also includes the SD-Command signal.

The physical connection and the level shifting of the signals is achieved by using the TXS02612 SD port expander [22]. The port expander allows for 2 ports for SD connections, of which only is one used. The level translation is from 1.8V, which is the voltage of the ZYNQ MIO bank to 3.3V that is used by the SD card.

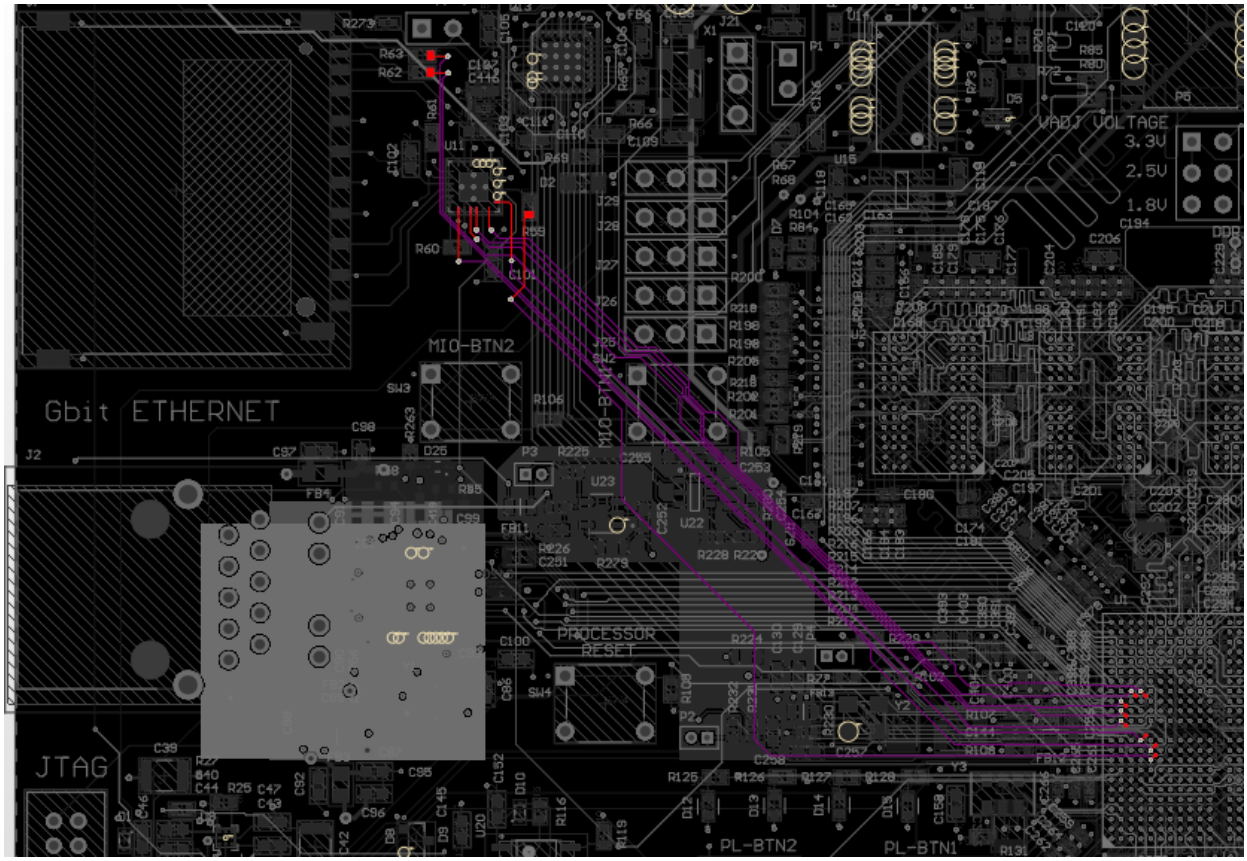


Figure 77. The SD card nets.

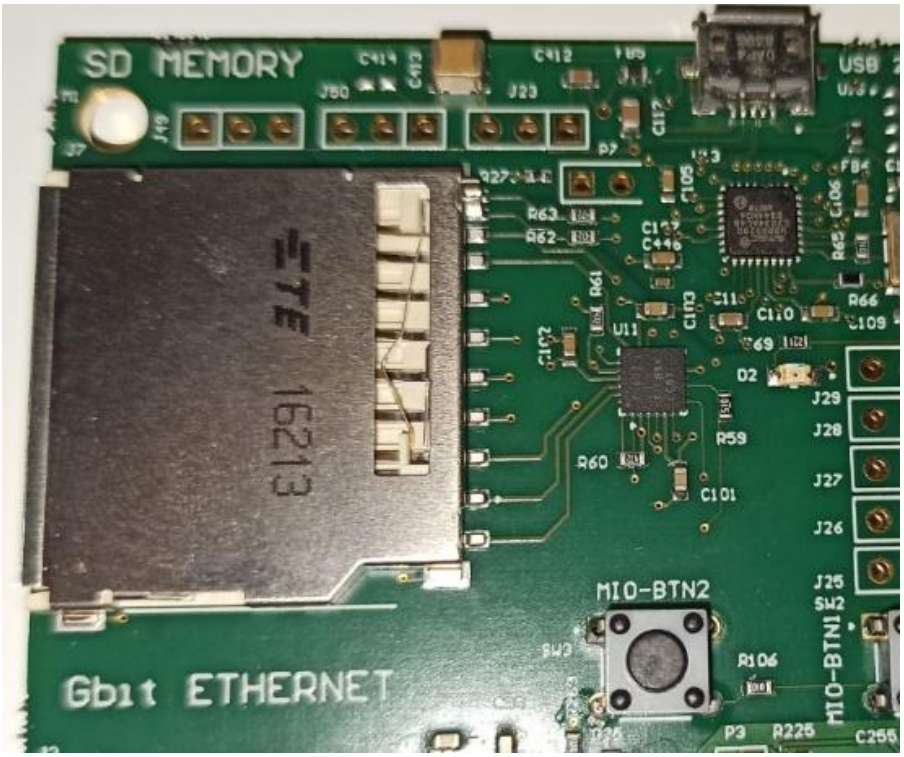


Figure 78. The SD card circuitry and the respective connector.

4.11 QSPI flash memory

It is useful to have a means on the board to store a bitstream and/or a bare metal application for the ZYNQ processor, without having to use the SD memory. For this reason, a QSPI flash memory is placed on the board. The advantage of loading the bitstream from a QSPI flash is the remarkable speed. In the case of this board, an S25FL256SAGMFI001 QSPI Flash memory from Cypress is used. The size of the memory is 256 Mbits, which allows for storage of the bitstream and there is a lot of space left for PS applications. The interface of the memory is QSPI (Quad Serial Peripheral Interface), which is based on the SPI (Serial Peripheral Interface), but uses two more lanes, and due to this is able to transmit 4 bytes in a single clock. The Quad interface in combination to the high bus frequencies supported (100MHz), allows for fast data transfer with very little latency overhead. This means that configuration times after power-on are nearly instant.

The flash memory is powered from 3.3V. Due to the high frequencies involved, extra care has been given to trace length matching on the PCB, in order to ensure reliable operation. On this board, the traces are length matched within 20 mils. The bitstream on the flash memory can be programmed from Vivado. This specific part guarantees at least 100,000 Program-Erase cycles, and the data retention is guaranteed to be 20 years on minimum.

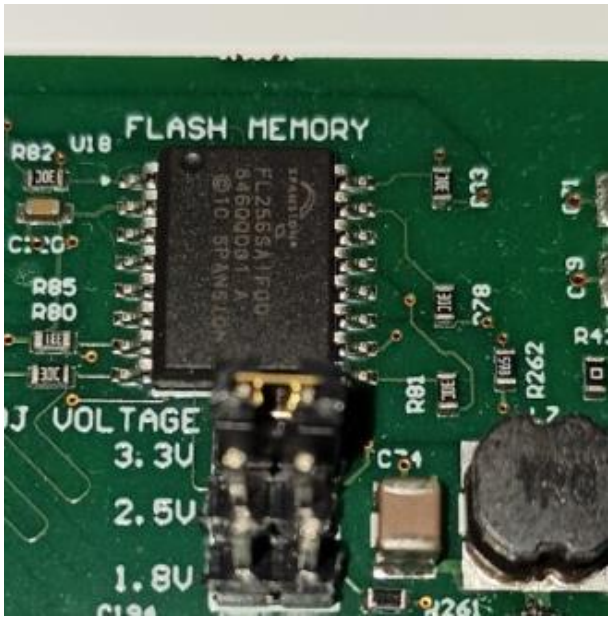


Figure 79. The flash memory circuitry on the board.

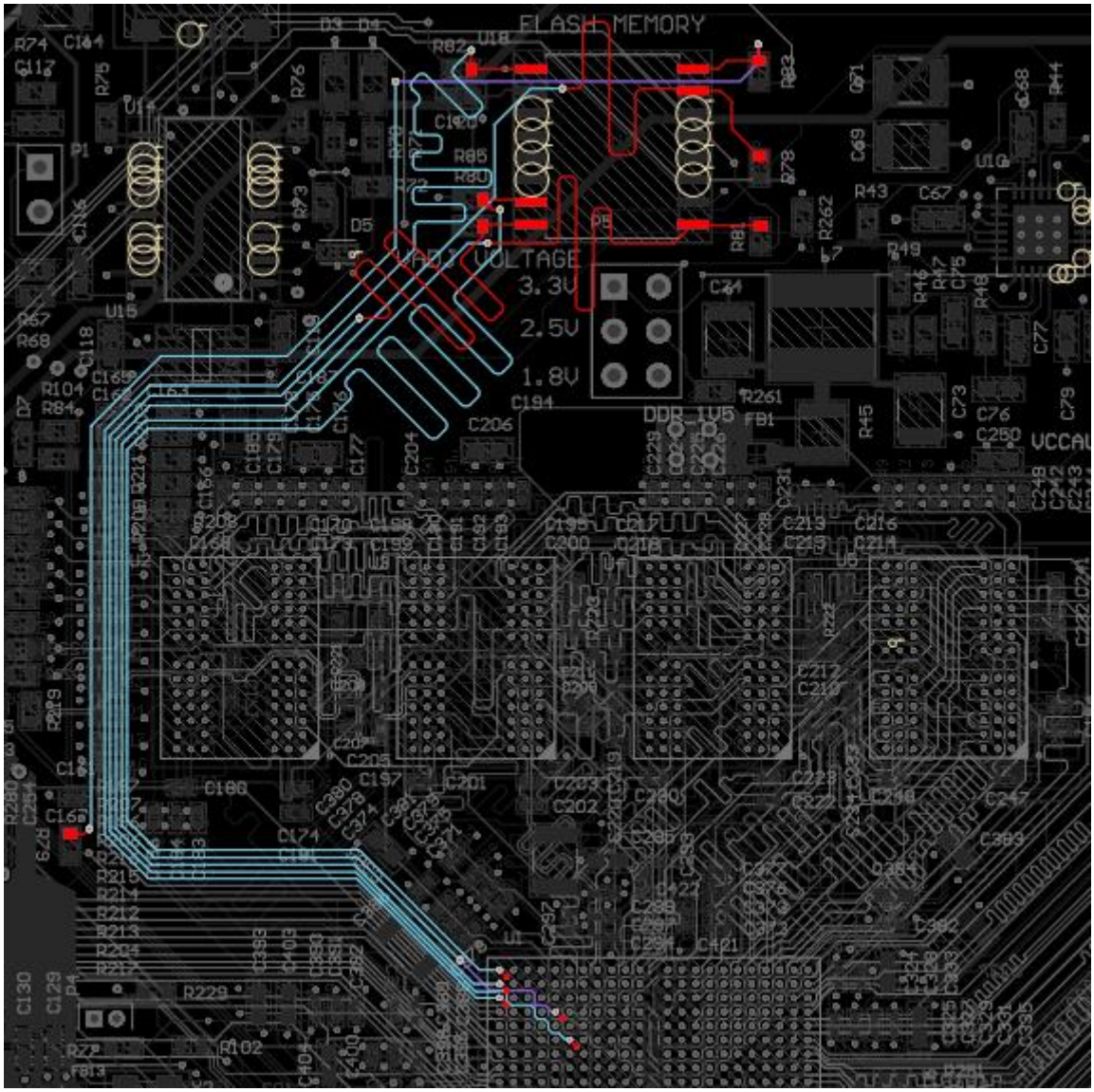


Figure 80. The QSPI flash memory nets.

4.12 USB 2.0

USB (Universal Serial Bus) is an industry standard which specifies cables, connectors and interfacing between computers and peripherals. It is widely adopted today, both in industrial and commercial applications. Several USB standards have been released, starting from USB 1.x, USB 2.0, USB 3.x and lately, USB 4. These standards feature backwards compatibility (except the cables for USB 4), something that aids their widespread adoption. There is a variety of USB connectors, such as Mini-A, Mini-B, USB B type, etc.

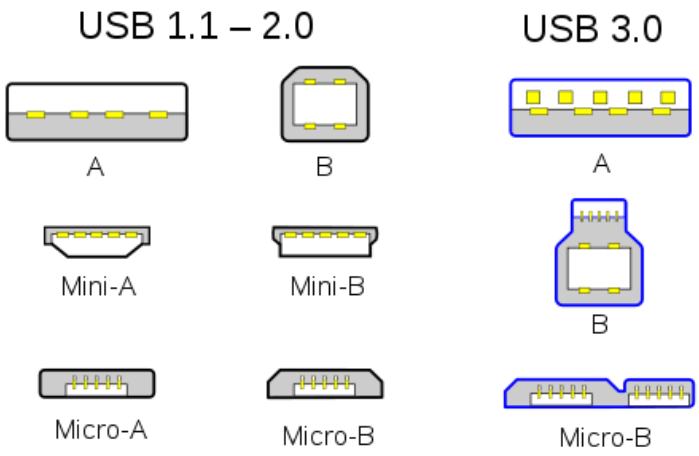


Figure 81. Different USB connectors.

The first USB standard (USB 1), was specifying speeds of 1.5Mbps at the Low Speed mode, and 12 Mbps at the Full Speed mode. USB 2.0 made a huge leap, offering 480 Mbps at the High Speed mode. Most peripheral devices during the write up of this thesis are using the USB 2.0 specification, from computer mice and keyboard to mobile phones and cameras. Power transfer is possible though USB, so many devices use this standard (or only the cables and the connectors) for charging. A common example is mobile phones. The ability to transfer adequate amount of power through USB is a big advantage of the bus. The voltage rail of the bus is at 5V and per specification the maximum current is 500 mA.

ZYNQ-7000 devices feature 2 USB OTG (On The Go) hard controllers on the chip, interfaced to the ARM processor. These controllers support the USB 2.0 specification, and their addition is very important, since it allows the connection of peripheral devices to the processor (such as a mouse or a USB camera), but can be also very good solutions to transfer data at adequate data rates. Had these controllers not been integrated to the ZYNQ, one would have to implement them on custom logic, and this is often a complex job. However, many companies offer IPs for USB controllers.

USB 2.0 uses a differential pair for the data connection. The signal level on this differential pair is $\pm$ 400mV. Since only one differential pair is available for the data, the communication is half-duplex (i.e. transmit or receive only in a specific moment), and it arbitrated by the host. In total, 4 wires exist in a USB 2.0 cable, and the other two wires are VCC and Ground. Using these two wires one can transfer energy from the host to the device.

Many devices can be connected on a single USB controller by using a hub. So for example, one USB controller does not mean only one possible USB peripheral attached. However, by using a hub the bandwidth is shared between the USB devices. Since USB transactions are complex, there is always the need for a driver on the host side to perform the arbitration to the user space. In case of commercial devices, this driver is usually provided by the vendor.

USB On-The-Go allows the device to act as a host or a device on demand. This is useful, since one can connect the board to a computer as a device, while, in another scenario, a USB mouse could be connected on the board, and in this case the board would function as a host.

While USB 2.0 permits relatively high speeds, there is a limitation on the maximum length of the cable. The 2.0 specification limits the length to 5 meters. This can be mitigated by using active repeaters on the cable, but usually the case is to not use USB if long distance wiring is needed. When no device is connected on the bus, the controller enters the so-called single-ended zero state. The host includes 15 kΩ resistors on each data line, so when no device is connected the lines are pulled down and the host identifies that no device is connected.

The USB controller embedded in the ZYNQ device cannot provide the signaling needs of USB and for this reason a PHY is needed. In this board a USB3320 PHY is used [23]. USB3320 is a highly integrated fully featured Hi-Speed USB transceiver, supporting ULPI interface between the controller and the PHY. ULPI stands for UTMI Low Pin Interface, and is an interface which allows the use of less pins for USB controller – PHY communication in comparison to the traditional interface.

The signals needed for the ULPI interface are 8 data pins, STP (Stop) which stops the data stream currently on the bus, DIR, which controls the direction of the data, NXT which indicates acceptance of the current byte received from the PHY, and when the PHY is sending data indicated when a new byte is available to consume. Of course a clock is sent parallel to the link, and all interface signals are synchronous to this clock. Both directions are allowed for the clock.

Figure 82. USB 2.0 circuitry on the board.

A signal pair laves the PHY and travels to the peripheral device though the cable. The impedance of this pair should be 90 Ω. Care has to be taken to ensure correct impedance for this signal, but also to ensure that no excessive delays appear in the ULPI lines, though the requirement is more relaxed in this case due to the lower operating frequency.

The PHY in order to function has to be provided with an external clock. On this board, the external clock is provided by a 24 MHz crystal oscillator.

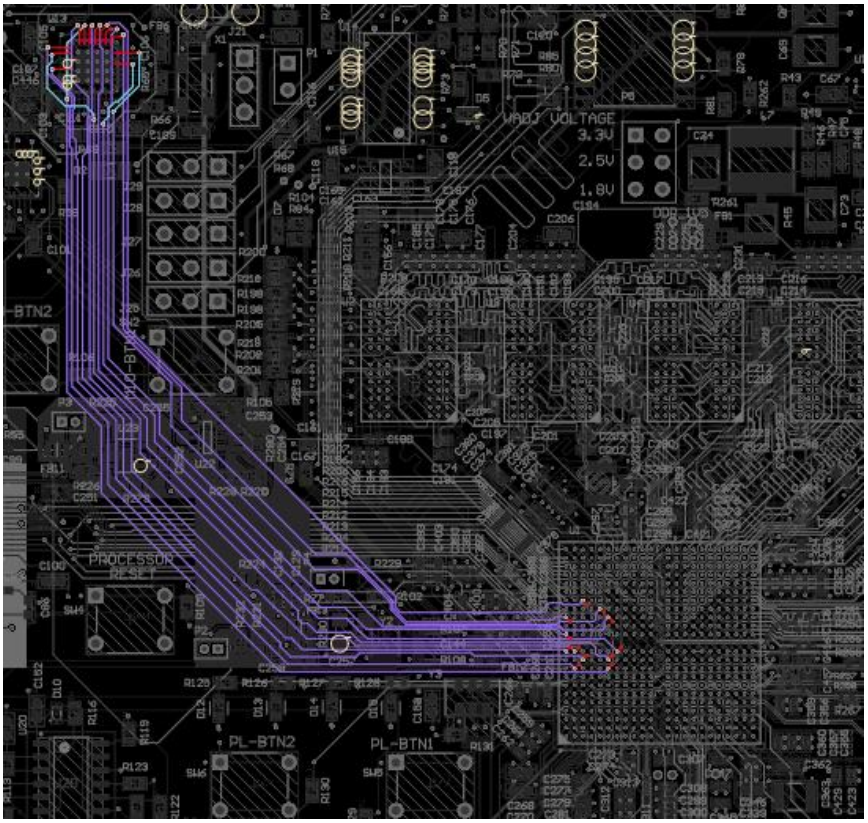


Figure 83. ZYNQ to USB 2.0 PHY nets.

4.13 USB 3.0

While USB 2.0 can serve a wide variety of needs, it lacks high data rates, such that one could use to transfer raw data from an FPGA-based data acquisition system to a Personal Computer for further processing. This would normally be done by using the PCI-Express lanes on the computer, which offer multi-gigabit capability. However, this would be a very complex and costly solution. Fortunately, USB 3.0 (Super Speed) exists nowadays, which offers high speed functionality at 5 Gbps, and since most computer motherboards now support USB 3.0 connectivity, the cost is only a cable.

USB 3.0 has many similarities with USB 2.0, the main being it is backwards compatible. However, there are many differences as well. In order for USB 3.0 to achieve its specified data rate, a new cable is needed in comparison to the cables that were the same up to the 2.0 spec. This is essential because USB 3.0 uses 2 extra differential lanes. Electrically, USB 3.0 is more similar to Multi-Gigabit transceivers such as PCI-Express or SATA than USB 2.0.

Another great advantage of USB 3.0, in comparison to USB 2.0, is that it is full-duplex, meaning that the user can receive and transmit with a data rate of 5 Gbps at the same time, making the bandwidth gap with the previous USB generation even larger. The available current for power through the USB 3.0 is now almost doubled, at 900 mA per specification.

The data in USB 3.0 are scrambled, in a similar manner as described in the MGT transceivers section, using 8b10b encoding. This adds up to about 20% encoding overhead, so the real data rate is more about 4 Gbps. The specification considers it reasonable to achieve 3.2 Gbps or a little more in real life operation. No maximum cable length is specified for USB 3.0, but 3 meters is considered reasonable for AWG (American Wire Gauge) 26 copper wires.

To achieve USB 3.0 connectivity, a transceiver is embedded on the board. This is FT601, a USB 3.0 to FIFO Bridge from FTDI Chip [24]. The transceiver is connected to the Programmable Logic of the ZYNQ device. FT601 can be handled as a FIFO interface from the

FPGA, so complex USB controller logic is avoided, making the interface easier and saving logic resources.

FT601 supports USB 2.0 High Speed (480 Mbps) and USB 2.0 Full Speed (12 Mbps), so it can be used with older technologies, if the data rate requirement permits it. It has built-in 16kB FIFO data buffer RAM, which comes useful when transmitting large chunks of data because transmission from the FPGA is not interrupted, in contrast to if there was a much smaller FIFO RAM.

The power supply needs for the FT601 is 3.3V, the I/O buffers voltage (which in this case is 3.3V) and the core voltage which is 1.0V. The FT601 has an internal LDO regulator, and is able to generate the 1.0V core voltage by the 3.3V supplied, and this regulator is used, to which only external decoupling is provided. FT601 needs a clock source, and this clock is provided by an external 30 MHz crystal oscillator placed adjacent to the chip. The package of the chip is a QFN-76, and its temperature range specifications are -40C to +85C.

The interface used to connect the FT601 to the FPGA is a 32-bit wide data interface, 4 byte enable bits for the FIFO data stream (BE [3:0]), TXE signal, which indicates whether the transmit FIFO on the FT601 is empty, RXF signal which indicates whether the receive FIFO on the FT601 is full, WR which is an active low input to the FT601 and is a write enable signal, RD which is an active low read enable signal, OE being an active low data output enable, RESET which is an active low RESET signal and WAKE_UP which can be driven by the application to send a remote wake up command on the host.

This bus has been routed to bank 13 of the FPGA. Since it is a wide, high frequency parallel bus, correct delay matching on the traces is critical to ensure reliable operation. The traces have been routed to within 80 mils of trace length, and all signal layers have been used for the bus routing.

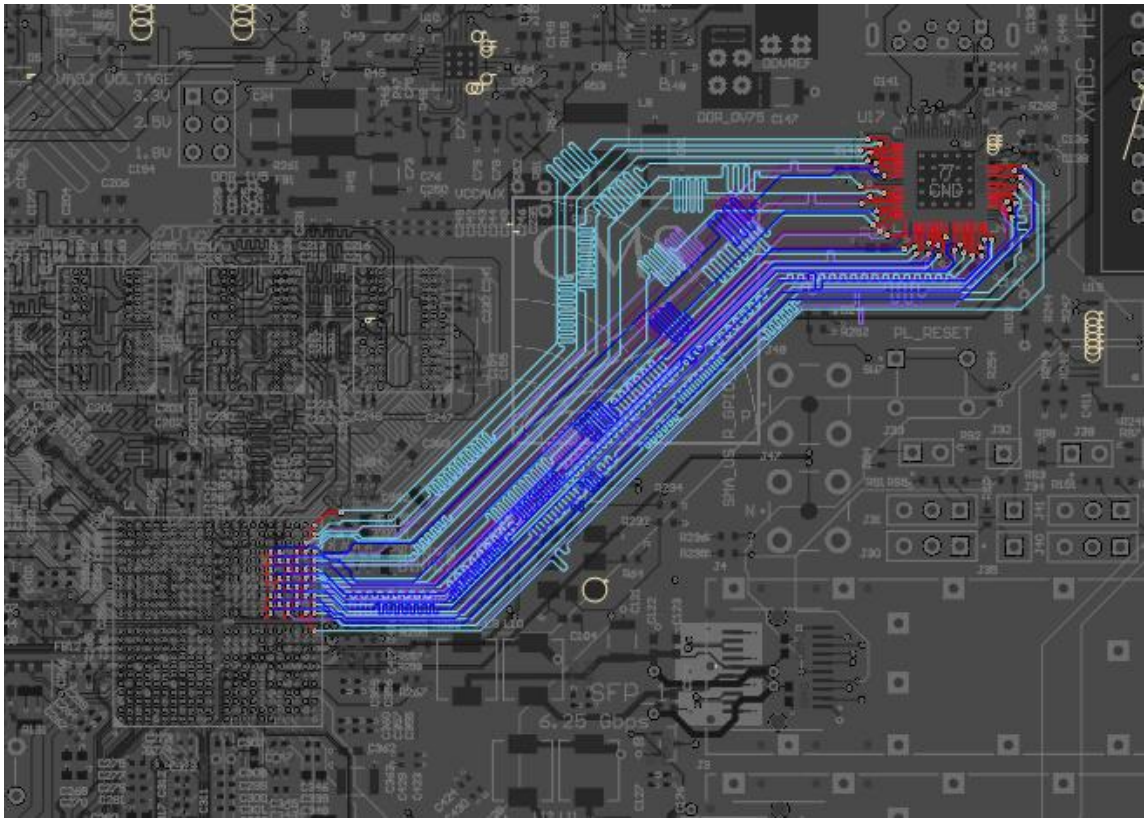


Figure 84. USB 3.0 bus signals.

USB 3.0 signals are high frequency transceiver signals so their length has been kept the lowest possible, by placing the USB 3.0 circuitry as close as possible to the USB 3.0 connector. These signals are differential, with an impedance specification of 90 ohms. The transceiver's transmit differential pair is AC coupled using capacitors.

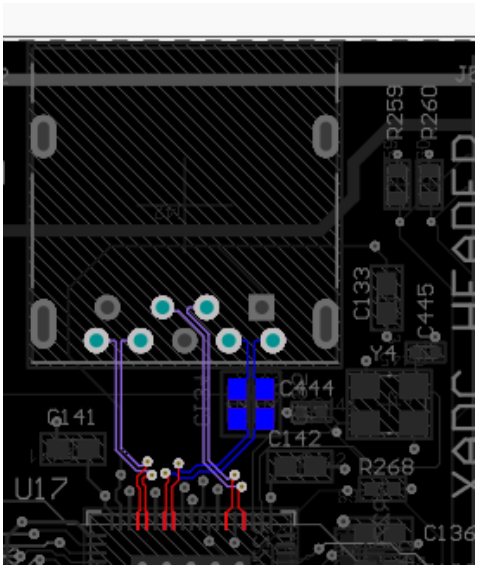


Figure 85. USB 3.0 transceiver high speed lanes.

The USB connector used is a USB 3.0 A Type Female Connector by Molex, part no. : 48393-0003 [25].

Figure 86. The USB 3.0 circuitry on the board.

4.14 Pin header batch I/O

If a data acquisition component is not embedded on the board (such as an ADC), there have to be means to interface to it. In order to aid connectivity of the board in this manner, an extensive number of pin headers have been included in the board design. These pin headers can be used to interface to almost anything, can provide adequate amounts of power to peripherals, and special care is taken in order to ensure that high speed connectivity can be sustained through these headers.

The pin header I/O have been placed in the bottom of the board, and in total 64 I/O connections are offered plus 2 lanes for an I2C bus. The pin header spacing is 2.54mm, since that is compatible with many peripherals, and cables such as ribbon cables can be used, that offer good signal integrity, and have the advantage of being widely available (such as IDE hard drive cables).

While pin headers are not known for their high-speed connectivity, with special care in the design phase one can ensure adequate performance. Examples of moderately high speed connectivity using pin headers are IDE hard drives, USB 2.0 and more. Furthermore, attention has to be given to whether a signal is preferred to be differential or single ended.

On the board, 2 pin header connectors have been used, each consisting of 72 pins in a 36 x 2 configuration. One pin header connector has been designed for differential operation, and the other has been designed for single ended operation. Each has its own needs, since for a differential pair one has to ensure that on the FPGA side, the pins chosen are capable of driving the Positive/Negative signals of the differential pair and that the skew within the pair is as little as possible. Of course, pin header pairs optimized for differential use can be used as single-ended signals, and vice versa, pin headers optimized for single-ended use can be used as differential pairs where applicable. However, under this case, performance will not be optimum and the user has to consider using the optimized configuration if the frequency needs of a particular device connected is high.

For the differential pin header connector 32 FPGA pins are exposed, or 16 differential pairs. For the single ended connector, 32 FPGA pins are exposed as well. A series resistor has been used on every lane to protect against wrong voltage levels applied to the pins. On the single ended signals 200 Ohm resistors have been placed in series, however in the differential pairs the resistors placed are 0 Ohms (practically a short), to aid in signal integrity for high-speed applications. However, if there is a safety concern when using a specific device, these resistors can be de-soldered, and resistors of higher value can be placed in this position.

Since pin headers is something that users usually touch with their hands, whether on purpose or by mistake, there is a huge danger of an ESD discharge event. This event can prove fatal to the FPGA's I/O buffers, or worse can damage the whole FPGA device permanently. To mitigate this problem, every single pin header has been protected with ESD protection diodes [21], reverse biased to the ground, with an ESD event handling capability of up to 20kV, 9V threshold voltage and ultra-low capacitance to avoid any signal distortion in the case of high speed signals.

Care has been taken to give the ability to the user to power the connected devices by the pin header connector if needed. So, 4 pins on each 72-pin connector are dedicated to supplying the 5V power, which thanks for its high output current capability can be used to power the connected devices, or supplied to a voltage regulator on the connected PCBs to generate one or more voltages on the connected system. Furthermore, each 72-pin connector features 4 dedicated pins to distributing the 3.3V power supply rail, in order to be used directly on the connected system.

Since different voltage levels when interfacing to external systems can be an issue, because for example this board has 3.3V available on the I/O buffers of the FPGA, and the connected board needs 2.5V. In this case a voltage level translator must be used which may add unnecessary complexity to the system. Having this case in mind, almost half of the pin-header I/O buffers on the FPGA are supplied from the VCCAUX power rail, which the user is able to configure to generate whether 1.8V, 2.5V or a 3.3V voltage level. So, interfacing to a circuit of different voltage is only a matter of changing the VCCAUX voltage configuration jumpers on the board. In the single-ended optimized pin-header connector, 16 out of the 32 bits are connected to Bank 34 on the FPGA, which I/O buffers are supplied by the flexible VCCAUX power supply.

On the differential signaling optimized pin-header connector, 7 differential pairs are connected to the adjustable voltage bank 34.

Since an externally connected system may need an I2C bus for configuration and/or control, 2 pin headers are dedicated to I2C. These I2C lanes are connected to the I2C mux, which bus is in the end connected to the hard I2C peripheral of the ARM processor in the ZYNQ device.

Since these pin-headers may be used for high-speed connections, special care is to be given to the trace lengths of the traces, as to not produce any unwanted skew between the lanes in parallel buses. For this, the traces from the FPGA to the pin header have all been length matched. All the bits within the first two bytes of the single-ended optimized connector are length-matched to within 50 mils, all the bits of the next two bytes of the single-ended optimized connector are length matched to within 50 mils, and all the differential pairs off the differentially optimized connector are matched to within 50 mils. The intra pair skew of the differential pairs is kept to < 5 mils.

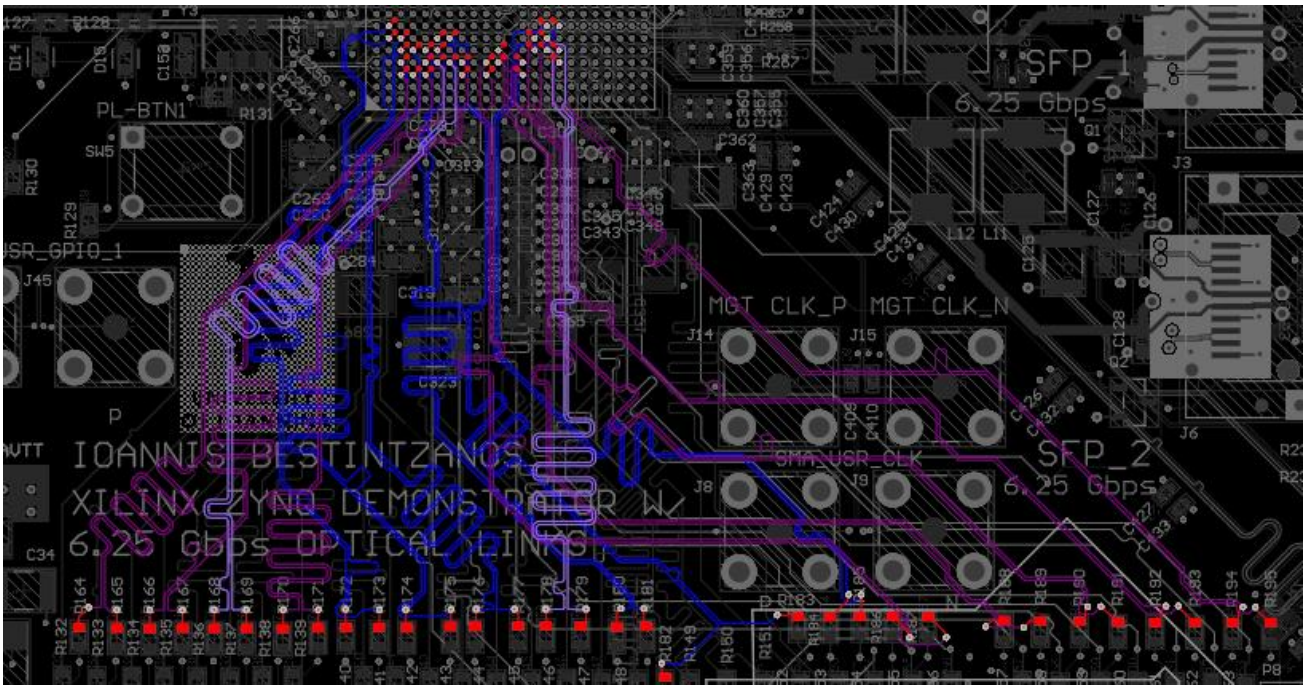


Figure 87. The differential pair lanes travelling from the FPGA to the pin header connector.

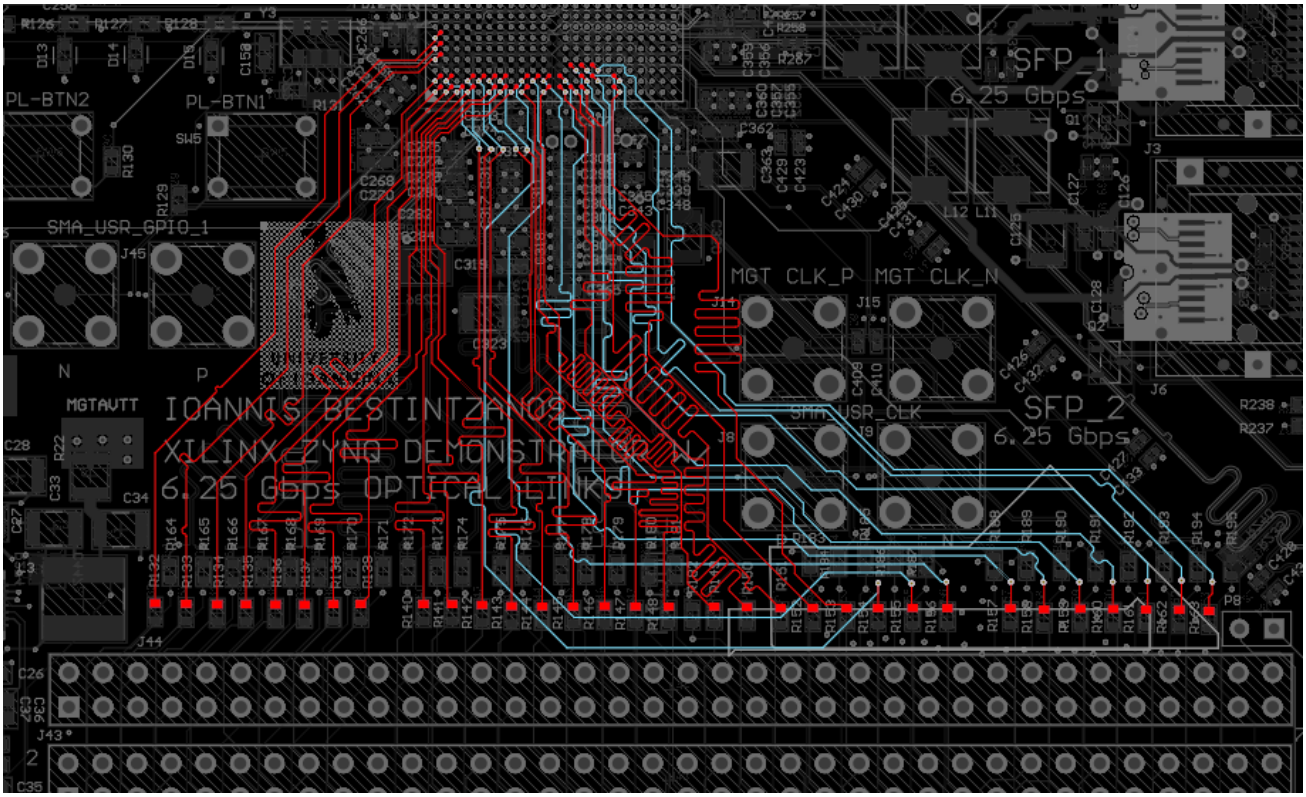


Figure 88. The single-ended lanes travelling from the FPGA to the single-ended optimized connector.

In order to aid for single-ended or differential connectivity, special patterns have been used for the connectors pin-out. This is mostly useful when the signals are connected to a ribbon cable and travel for quite a long distance. For the single ended signals, it has been ensured that all the signals are shielded using ground (or DC voltage) lines around them.

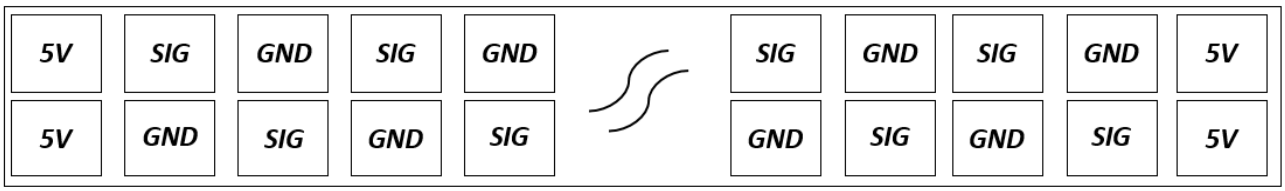


Figure 89. Single-ended optimized connector signal pattern. Notice the ground shielding around the signals.

For differential signals a similar pattern has been used, but now shielding both the lanes of the differential pair. This aid in keeping a constant impedance throughout the travel of the signal, but also the shielding of the differential pair increases a lot the noise immunity of this lane.

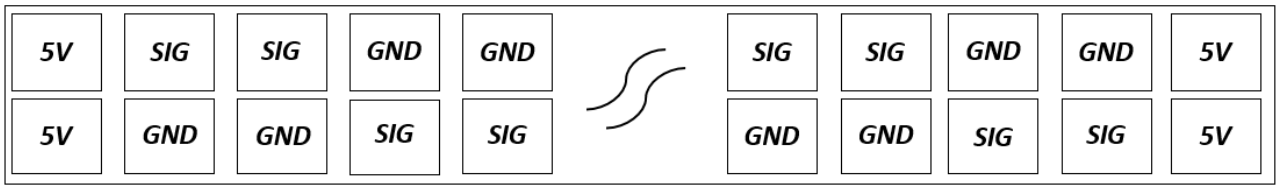


Figure 90. Differential-pair optimized connector signal pattern. Notice that now both the Positive and the negative lines of the differential pair are shielded, while maintaining the differential line coupling.

Figure 91. The pin-header connectors on the board. Only the single-ended connector is populated here. Note the series resistors above the pin header connectors and the ESD protection diodes below the connectors.

4.15 Generic Input / Output

A number of miscellaneous I/O have been added to the board to assist debugging and connectivity. These are various indicator or user programmable LEDs, general or specific purpose buttons and a number of SMA connectors to aid in high-speed connectivity. Furthermore, a pair of SMA connectors have been added to provide external clock to the FPGA.

The are 2 indicator LEDs on the board, a green one indicating whether the power supply circuitry of the board is up and steady and a blue one which indicates if lit whether the PL has been programmed. On top of that, there are 4 green user programmable LEDs in a row, which can be used by the user for various indicating functions or for assisting in debugging the logic. Furthermore, there is an LED which is connected to the Arm Processor, and can be programmed from the applications running on the ZYNQ's processor.

There exist 6 buttons on the boards. One is connected to reset pin of the ARM processor, so the user can execute a reset when needed. Another one is connected to the PL PROG pin, which if pressed erases the current configuration from the PL and the PL loads again the bitstream file if configured to load from a medium (i.e. The QSPI flash or the SD memory).

Another 4 user-programmable buttons are provided, 2 of them being connected to the FPGA and are accessible from within the programmable logic, and 2 of them connected to and accessible by the ARM processor.

4 SMA connectors have been included on the board, which are connected to the programmable logic of the ZYNQ device. The SMAs come in pairs and form two differential pairs to assist high speed connectivity. Again, they can be used as 4 individual single-ended signals. Since

these SMA connectors are purposed for high-speed applications, the two traces of each differential pair have been length-matched to avoid any timing skew.

Furthermore, an SMA pair is connected to a global clock input, on to which an external clock can be connected and supplied to the whole FPGA device die, since it is being fed directly to a global clock buffer. Again, these pins can be used for regular Input/Output functions if desired, and their traces haven been traced for minimum skew.

4.16 JTAG

The download of the bitstream and the processor configuration is both done through JTAG, as well as run-time access to the FPGA. Furthermore, the user can use the JTAG to program the on-board QSPI flash. For this reason, a JTAG header is provided on the left of the board. All the JTAG signals are protected against ESD discharges by ESD protection diodes.

4.17 I2C Multiplexer

I2C (Inter-Integrated Circuit) is a serial communication bus invented in 1982 by Phillips Semiconductor. It is widely adopted, and is used to attach low-speed peripherals to processors and microcontrollers in short-distance, for intra-board communication. Common uses are the interconnect of small memories, holding configuration information, RTC (Real Time Clock) connection, low-speed ADCs or DACs, camera control, screens etc. To a huge degree, its wide adoption is due to its simplicity. Two cables are enough for duplex communication, SDA (1-bit bi-directional data) and SCL (clock).

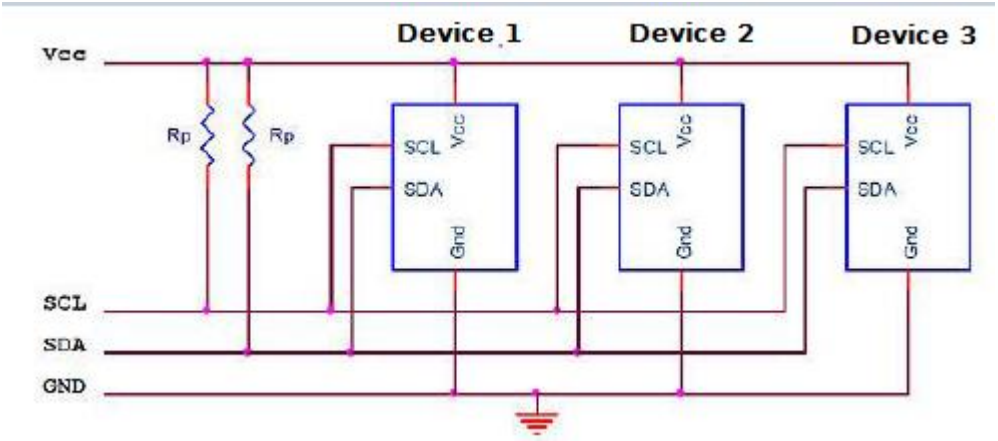


Figure 92. Typical I2C topology.

In I2C context, a device can function as a Master or as a Slave. The master device can start a transaction and is responsible for the clock distribution. Each data bit is transmitted with each clock cycle. Since the bus lines are open-collector or open-drain, pull up resistors are used on the lines.

The Start condition that initiates a transaction, happens by pulling SDA low while SCL is high. After this follows the payload and then a Stop condition, which happens by pulling SDA high while SCL is on the VCC level.

In a transaction, the first byte transmitted is the address of the device that is the recipient, and the last bit indicates whether this is a read or write request. The second byte contains the register that is about to be manipulated or read, and the rest of the bytes following are the data to be written, if about a write transaction, or then the sender stops and the corresponding device has to respond with the data requested, for a read request.

The device which is receiving the message, has to respond with an ACK (ACKnowledge) after each byte received. Failing to do so means the transaction has failed and this is referred to as NACK (Not ACKnowledge). The ACK condition happens on the 9th bits of the transmission, where the transmitter leaves the SDA lane and the recipient pulls it down. The line pull down is detected by the transmitter.

Speed wise, I2C is a slow protocol. Usual speeds are 100-400 KHz, while the last specification allows up to 5 MHz if the devices support operating that high (quite unusual, though). There are arbitrations of the I2C bus used for specific applications. An example of this is PMBus (Power Management Bus), which is used for controlling and monitoring power supply controllers.

The ZYNQ device has 2 I2C controller peripherals embedded on the ARM Processor, one of which is used on the board. However, due to the address collision that may happen on an I2C bus due to the limited address space (7 or 8 bits), an I2C mux has been placed on the board to mitigate any problems that might occur. Devices that use I2C for configuration on the board are the Si570 clock synthesized used for the MGTs, the 2 optical SFP transceivers, and also the I2C bus is exposed to the pin header connectors and on the XADC header.

The I2C mux used is the PCA9548 [26], which has an input of one I2C bus and up to 8 outputs, of which 4 are used. The user has to first communicate with the I2C mux though I2C to configure the multiplexer, and then the mux is transparent for the next transactions. The address of the I2C mux is configurable by 3 pins on the mux IC. The address has been fixed using resistors, and can be changed by relocating these resistors.

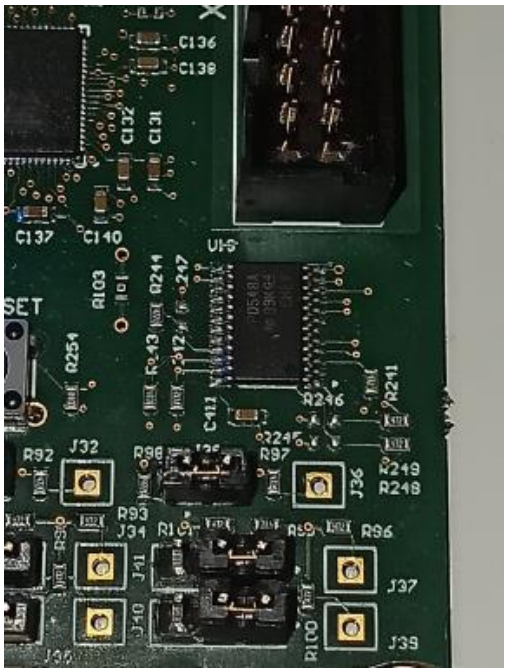


Figure 93. The I2C mux circuitry.

4.18 Boot mode configuration

The ZYNQ device supports several boot modes. On power-up or after reset, the ARM processor is the master of the booting process, and the mode it follows to boot depends on the configuration the user has chosen. This configuration is selectable by the user, by setting the appropriate jumpers on the pin headers J25, J26, J27, J28 and J29. MIO pins [2:6] set the boot mode, depending on the state they have on power. For this, the configuration pin headers provide either a pull-up or pull-down to these pins. The modes that the user can choose from are:

- Boot from JTAG
- Boot from SD
- Boot from QSPI flash

Furthermore, the user is able through these headers to enable or bypass the internal PLL, and to select whether the JTAG chain will be cascaded between the PL and the PS.

4.19 XADC header

The ZYNQ device contains an integrated dual ADC, capable of 1 Msps (Mega samples per second), with 12 bits of measurement precision [27]. This ADC is capable of accessing up to 17 external input channels. Such embedded ADC capability is very useful in monitoring, such as power supply voltage and current monitoring, or temperature measurements on the board. However, it is also useful for measurements out of the board, if the sampling rate and accuracy make the ADC suitable.

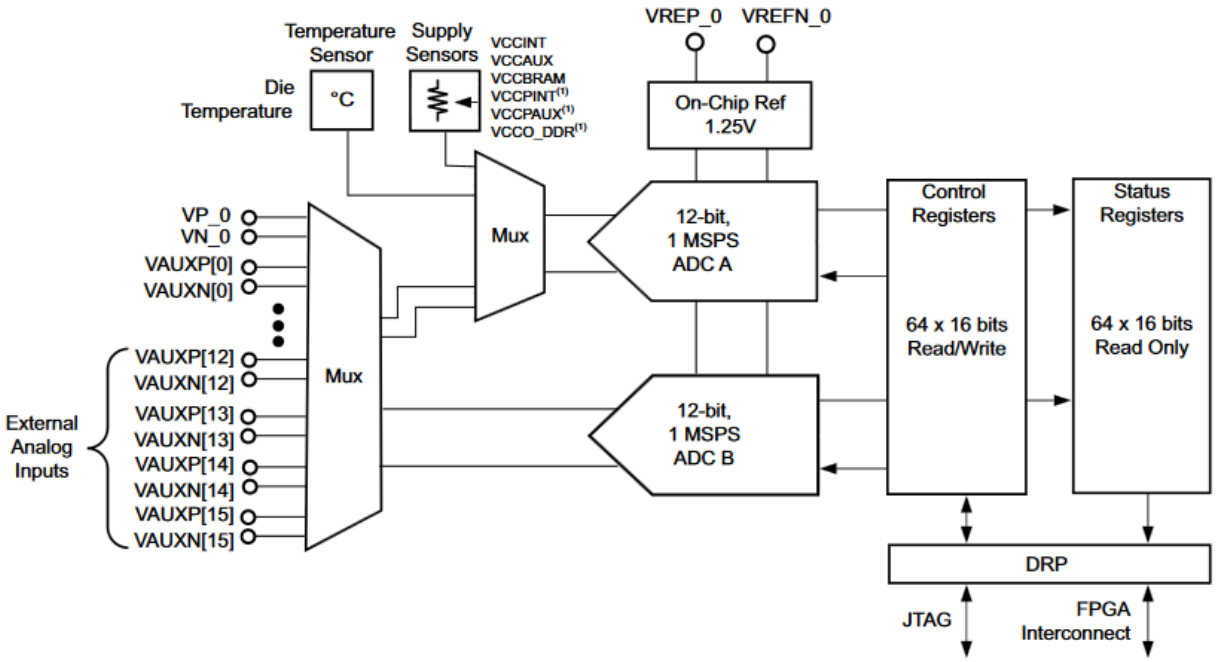


Figure 94. Typical XADC Block Diagram.

The XADC is accessible through JTAG or the programmable logic. In the case of this board, external reference voltage is provided, and an independent power supply is provided to the ADC in order to minimize noise coupling as much as possible. Two ADC channels (Channel 0 and Channel 8) are exposed to the XADC header, as well as the XADC reference and power rails of the card, in case there is a peripheral connected on this header. Furthermore, an I2C interface is provided on the same header. The XADC header is placed on the top right side of the card.

Chapter 5: Testing and performance

5.1 Power supply

First test in the testing procedure of the assembled boards is to verify against shorts that should not exist. This is done with a multimeter, paying special attention to the power rails for any shorts. If shorts are found in this stage of testing, they could indicate that the connectivity is wrong, or that a device was damaged during the assembly. No shorts that shouldn't be there were found.

The next step is the verification of the power supply circuitry. To avoid damage in components in case something went wrong, all the power supply rails were isolated by the series resistors existing in the output of every regulator for this very reason. First the power up of the 5V supply took place. The power up happened using a bench power supply in the 12V input, using a current limit of just a few mA. The input capacitors take some time to charge since the capacitance is quite high. The 5V are probed with an oscilloscope to confirm that the power rail is up and steady.

After verifying that 5V power is ok, the rest of the power rails have to be checked. Since there is a specific voltage sequencing on the board, first have to be checked and powered the rails that are first in the sequencing order, otherwise the rest of the rails will not be enabled. The verification of the next power rails happens by placing again the removed resistor in place, so that the voltage reaches the next conversion stages. Next in testing order is the VCCINT power rail, and after this one is verified as well, the same exact procedure happens for the VCC1V8 power rail. After this one is verified as well, the same procedure takes place for 3V3FPGA, MGTAVCC, MGTAVTT, DDR1V5 and VCCAUX. In the end, this happens again for the RAM reference voltage rails and the XADC dedicated rails. No faults were found in the power supply and everything is functioning according to specifications. With all the rails power up, the current draw is in the order of 100mA, indicating that nothing is drawing excess current. Extended measurements on the power rails are provided in the appendix.

5.2 Clocks

After verifying the power supply for correct operation, all the oscillators and the programmable clock generator were probed using an oscilloscope. These peripherals should function and output the clock signal, without having anything programmed on the FPGA. All the clocks were found to operate nicely within their specified operating frequency.

Figure 95. The 33.33MHz processor clock.

5.3 FPGA

Having verified the power supply and the clocks, the next step is to check the FPGA. This is done by connecting to the JTAG header a Xilinx Programmer Cable, and check if Vivado can recognize the FPGA, after configuring the on-board configuration headers. Vivado successfully recognized both the FPGA and the ARM processor on the ZYNQ.

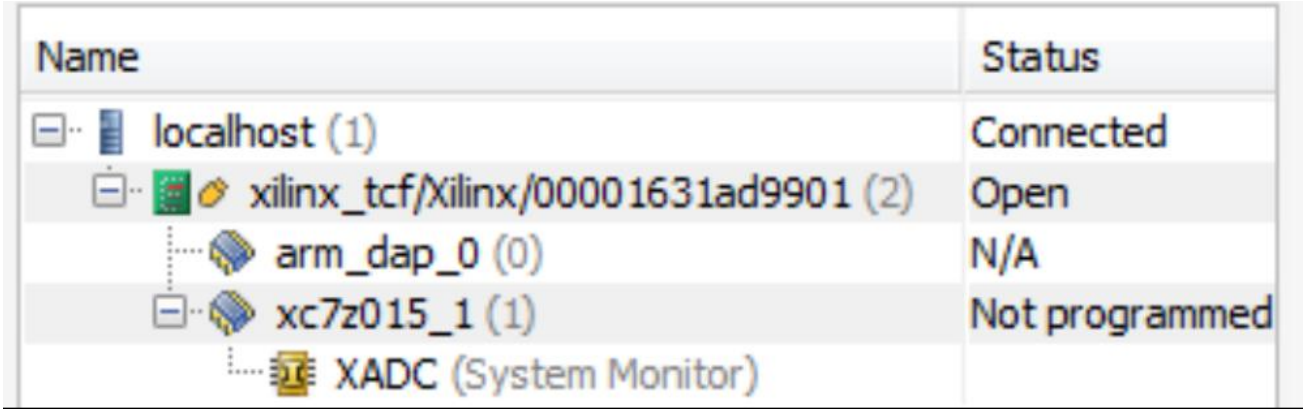


Figure 96. JTAG connection to the ZYNQ.

Since the FPGA is recognized and the clocks are there, the next step is to test the logic inside the FPGA. A simple VHDL circuit is written to light 2 two of the 4 user leds on the board, and after some time turn them off and light the other 2, in a loop. Also, a process that generates a slow clock by the on-board 200 MHz oscillator fed to the FPGA is written (fed to the LED process), testing logic, global clock distribution network, connectivity and the LEDs at the same time. After the bitstream is generated and downloaded to the board, the LEDs light up as expected.

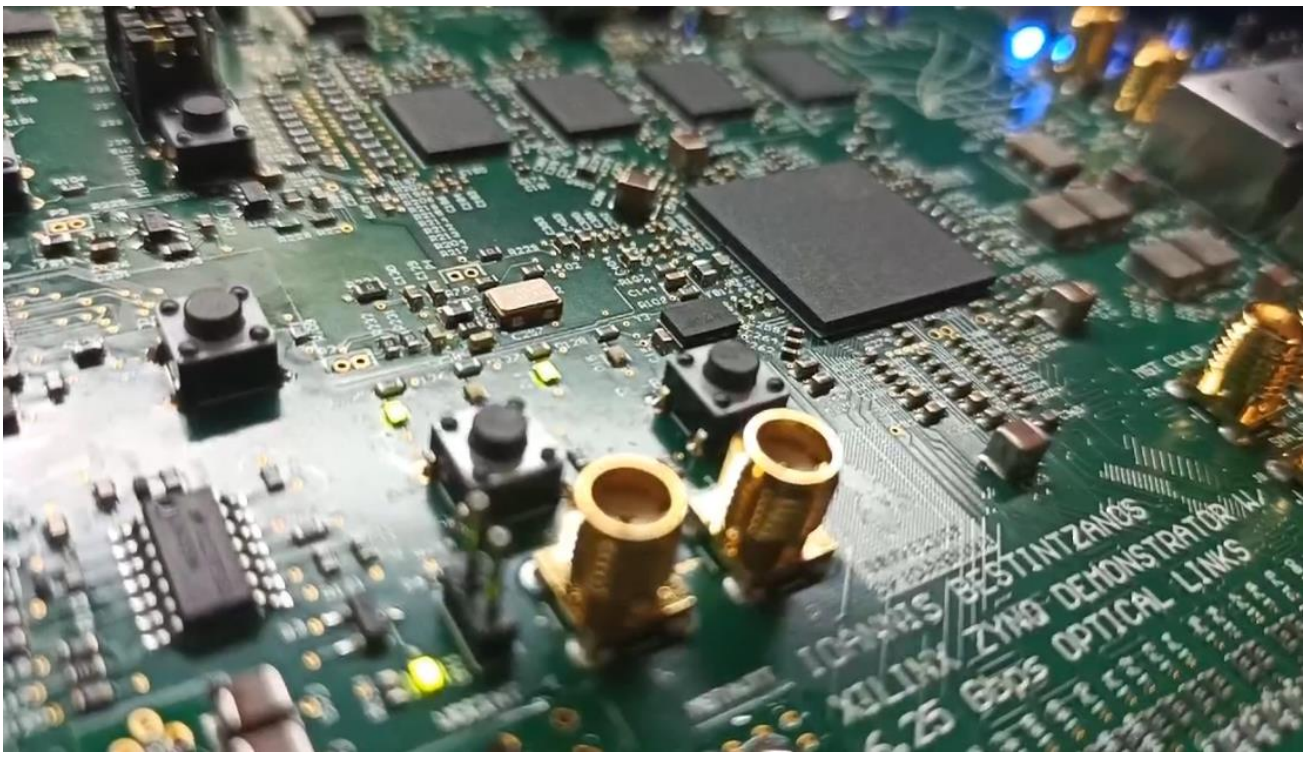


Figure 97. The LEDs lit up and blinking.

5.4 Processor

Since the FPGA seems to be functioning as expected, the ARM processor is to be tested next. The fastest way to do so is to instantiate the processor in a design, and configure the UART peripheral of the controller, as a means of communication with the outside world. Then, using the Xilinx SDK the processor can be configured and programmed. After generating the bitstream with the ARM processor instantiated, a simple C program is written to output an increasing counter in the UART. In this way the processor is tested, the PS oscillator is tested, and the UART circuitry is tested as well. After the processor was programmed, UART output was as expected, thus confirming the functionality of UART and the processor.

5.5 RAM

With the processor functional, the next step is to test the RAM. To do this, a bitstream has to be created, in which the processor is instantiated, together with the memory controller which gets configured according to the memory placed on board. Then, using the Xilinx Software Development Kit, a new project is created, with a template to test the memory by Xilinx. The template "Zynq DRAM tests", runs out of the processor's memory and performs memory tests on the DRAM connected on the ZYNQ, and outputs using UART. After compiling and downloading to the processor, the RAM is recognized by the processor successfully. All the tests are run successfully with 0 errors, and memory eye scans are performed, which report very good results.

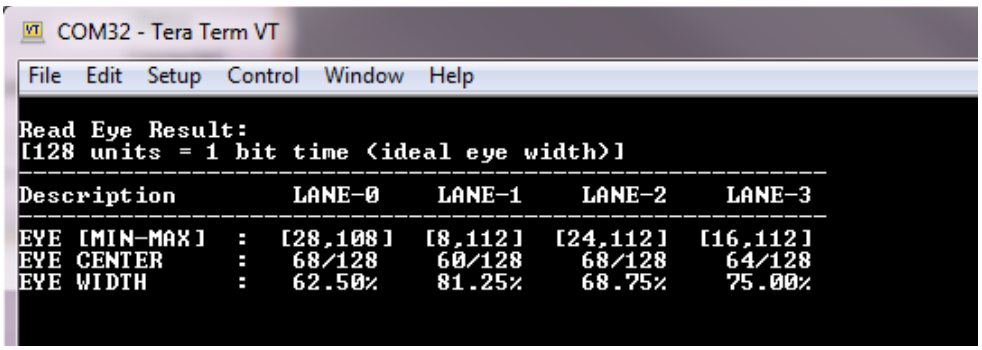


Figure 98. SDRAM read eye measurements.

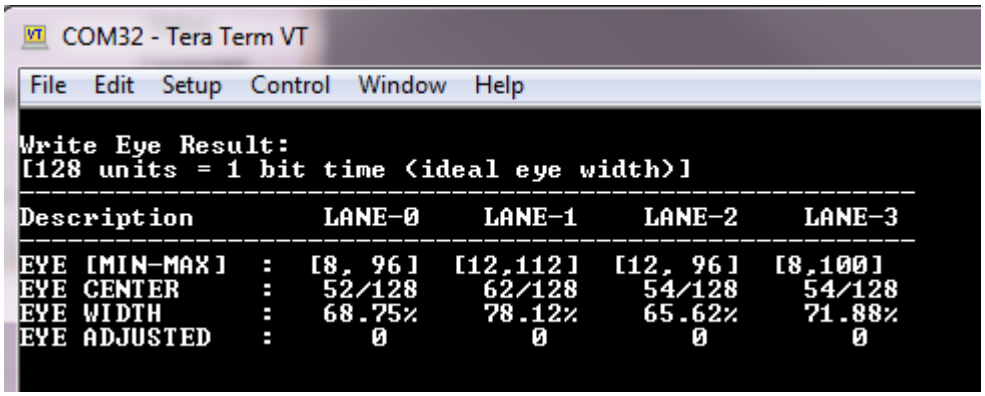


Figure 99.SDRAM write eye measurements.

The memory eyescans show excellent memory performance. For comparison, eyescans captured from Digilent's Zedboard are given:

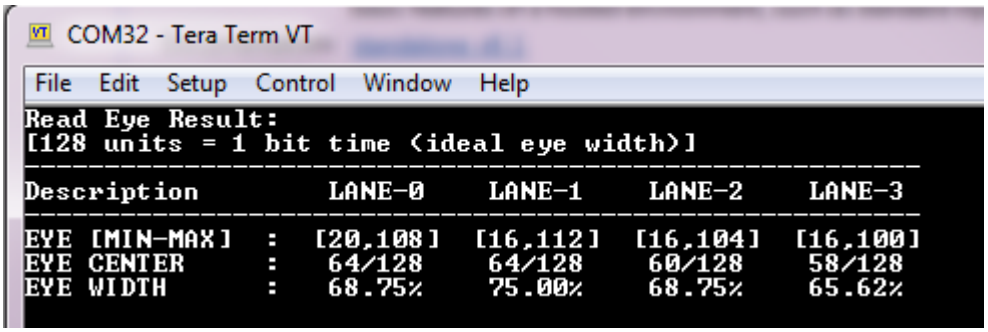


Figure 100. Zedboard's read eye measurements.

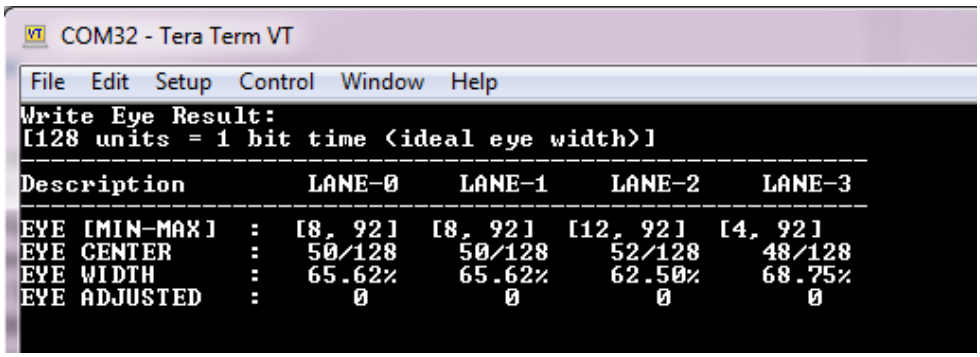


Figure 101. Zedboard's write eye measurements.

The comparison shows that this board has very good performance, especially considering that less layers are used on the PCB, and double the amount of RAM and ICs are installed on this board.

5.6 SD card & Ethernet

Next step would be to test the SD card and the Ethernet circuitry for functionality. There are many ways to do that, but a complete one is to create a Linux image, burn in to an SD card, and load it to the processor. In this way, Ethernet connectivity could be tested as well.

The Linux kernel image is created using Xilinx Petalinux toolkit, and an Ubuntu rootfs for the ARMv7 architecture is attached. Next, the files are transferred to the SD card and the ZYNQ is configured to boot from SD. The UART serial output will now show the boot progress and provide shell access to the system.

The board booted up fine, the Ethernet auto-negotiation mechanism initiated the connection, and the board is now able to access the internet. After downloading some essential packages, and installing the Xfce lightweight desktop environment, a VNC server is set up. Using the VNC server, the desktop of the OS running on the ZYNQ is now accessible from anywhere on the network. The memtester package is downloaded that can aid in testing the random access memory inside from the operating system.

5.7 Multi Gigabit Transceivers

A very important part of this thesis is the functionality and the performance of the multi gigabit transceivers. Essential in the characterization of the performance of the MGTs is the metrics of BER (Bit Error Ratio) and the eyescans. In order to calculate BER, one has to send a huge amount of data through a link, and test on the receiver side if the data received are the same with the data transmitted. If one bit is found to be wrong, the error counter increases by one. For example, if one transmits 10^9 bits and finds 5 wrong bits the BER is $\frac{5}{10^9} = 5\ x\ 10^{-9}$. If 0 errors are found, the one can calculate the BER floor from this measurement, which is an upper limit on the BER.

To do this, Xilinx's IBERT IP core is used. Configuration of the IP core is done for operation at 6.25 Gbps per link, feeding in as the reference clock for the PLLs the Si570 clock, which defaults at 156.25 MHz is done. The test is as follows: A PRBS sequence is generated on the transmitter, using the hard PRBS block of the MGTs and is then serialized and sent in the channel. The PRBS data are random so they should produce an AC balanced signal, much like a scrambler would do. The receiver, on the other hand, receives the data and checks if the data received match the data expected from the PRBS sequence. For every bit that does not match, an error is accumulated in the error counter. There is a variety of options offered for data to be transmitted, such as PRBS7, PRBS15, PRBS23, PRBS31, Fast Clk and Slow Clk. The transmitter configuration has to match the receiver configuration, in respect to the data. The user can modify several options for the transceivers, such as TX Pre-Cursor, TX Post-Cursor, TX Differential Swing, RX termination voltage, reset the TX or the RX portion of every MGT, and more. Furthermore, the user can inject errors to verify the system functionality, on whether these are detected by the receiver or not.

One of the most important techniques when performing eyescans is the loopback. The loopback can be done externally, as simple as connecting the TX to the RX, be it copper cables or fibers optics, to forcing a harsh environment for the MGTs to operate in, such as very long cables, signal attenuators, etc. However, the GTP MGTs offer more loopback modes, the PCS loopback mode and the PMA loopback mode. In the first mode, the data never leave the digital domain of the MGT and are loop backed from the TX to the TX right before the digital domain boundary. This tests the logic circuitry and node of the analog parts. The PMA loopback tests the signal in the analog domain, and the TX to RX transition of the signal happens right before it leaves the chip. This serves in testing the analog circuitry of the MGT, along with power supply noise, clock jitter, etc. Finally, the external loopback mode can be used to test everything else, such as PCB traces, SFP modules, fibers or cables, etc.

After the design was implemented and the bitstream was downloaded to the FPGA, the link's TX/RX pairs were configured and the tests were run. The copper links were externally looped back using a pair of shielded SMA copper wire for each P/N pair of the MGTs.

In all loopback modes, no errors were detected, after running several terabytes of data. Eyescans were performed on each receiver, with excellent results; the eyes were very open,

indicating an electrically good link, with little jitter on the clock and little noise in the power supply rails. Though the GTP MGTs are specified for a data rate up to 6.6 Gbps, error-free performance was demonstrated for considerably higher data rates. Extended testing data can be found in the appendix.

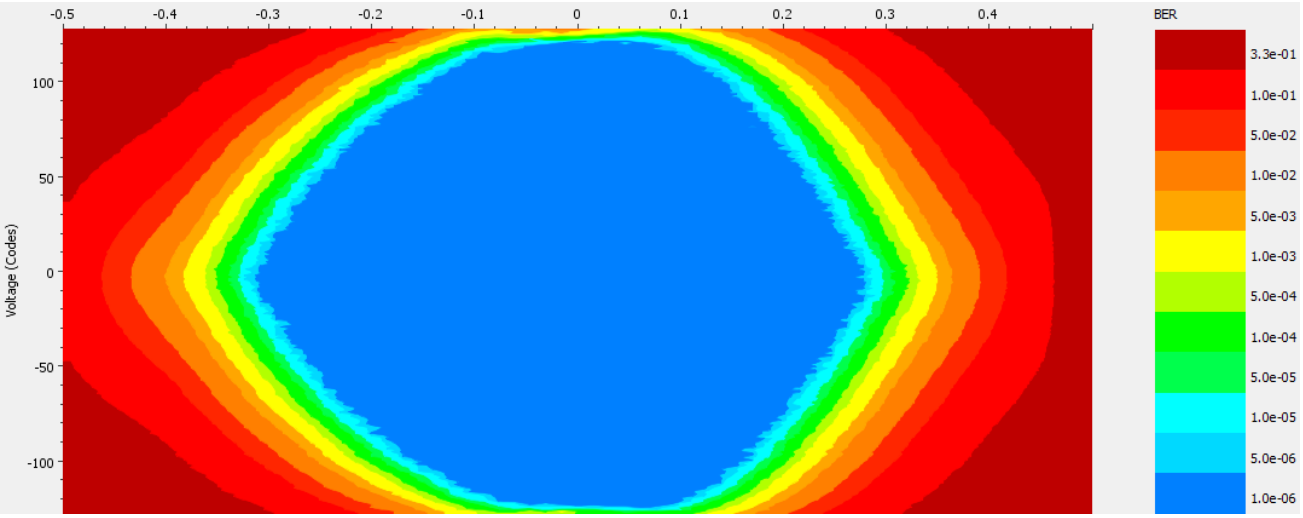


Figure 102. Eyescan on an MGT with an optic SFP attached, on external loopback mode.

5.8 Extended testing

After the initial testing, a series of extended testing was performed. In order to surface any errors, elevated temperatures were used. A fault has a much higher probability of occurring near the edge of the operational envelope of the device, rather at perfect conditions. So, the ZYNQ device was kept at a steady ~90C junction temperature.

The tests lasted for more than a week of continuous stressing. In this way, not only the components under stress are being tested, but the whole board as well, including circuits of high importance such as the power supply. A single glitch in the power supply could starve the FPGA from power for a little amount of time, enough though for the configuration to be lost. The tests were aimed mainly at the RAM and the MGTs.

An OS was running on the board, and at the same time the MGTs operated at external loopback at their highest data rate. From inside the OS, memory tests were running during the testing period.

The tests yielded excellent results, with 0 errors on all the MGT channels, sending over 2x10e15 bits per channel, achieving a BER floor of lower than 5x10e-16.

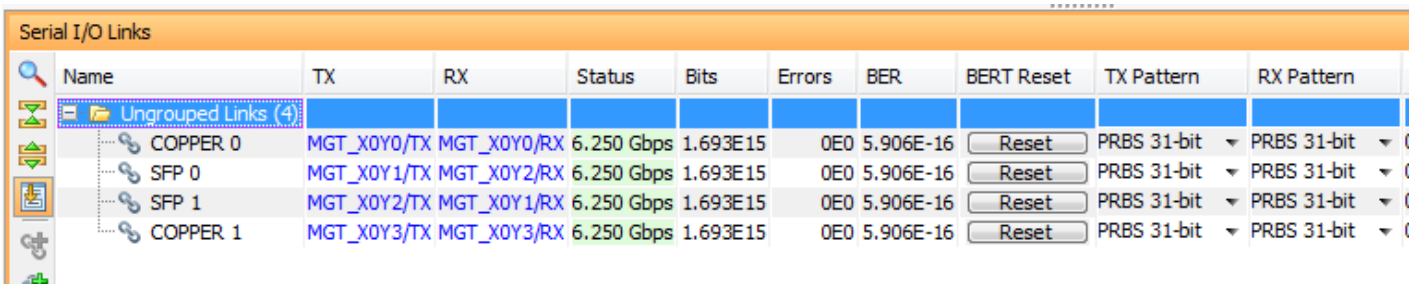


Figure 103. The BER extended testing.

Similar results were observed for the RAM, since no errors were detected over all this testing course.

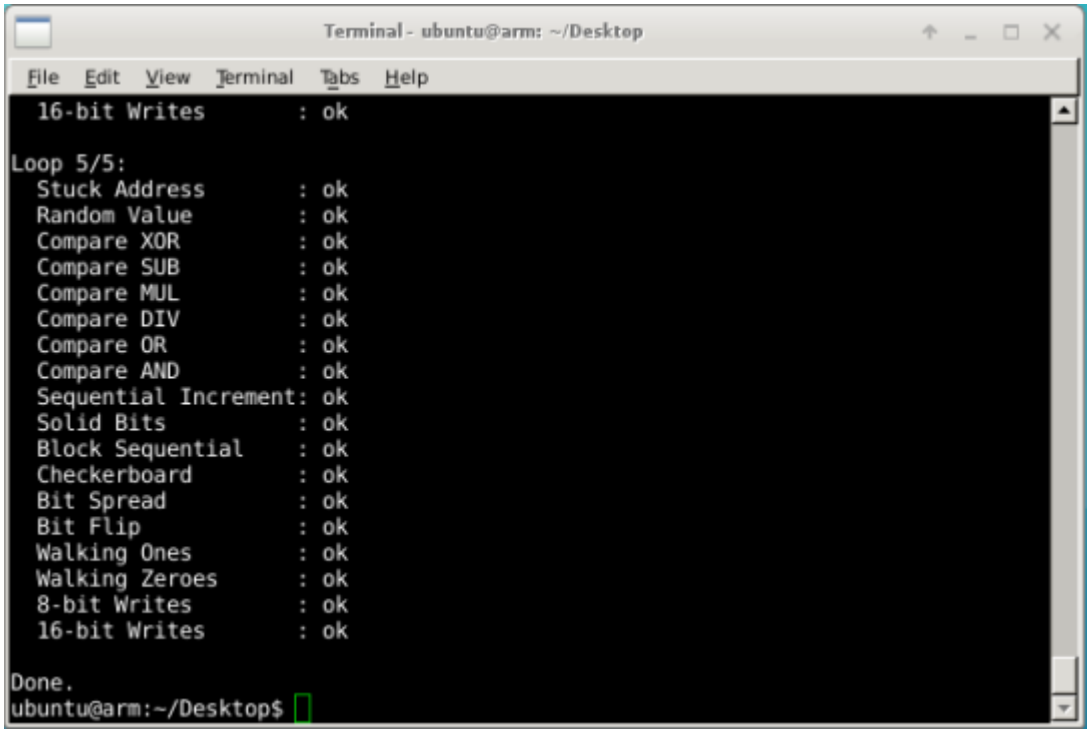


Figure 104. Memtester terminal interface.

5.9 Co-operation of the ARM processor and the FPGA for histogramming using root

As a proof of concept, and a demonstration of the ZYNQ's capabilities, a simple scenario mimicking live histogramming for the CMS's Level-1 triggering system was developed. In such experiments, where huge amounts of data come into play, one cannot store and process all of them. Processing is done in real time with systems utilizing large FPGAs, running custom algorithms with very little latency and very high throughput performance. These algorithms form a "filter", in the sense that they get to choose which events are interesting physics-wise, decimating the data rate by orders of magnitude. This hardware trigger usually consists of one or two layers, after which the data passed are forwarded to a high level trigger which runs on commercial CPUs and GPUs and performs extended processing, again decimating the data rate. In the end, only a miniscule fraction of the generated data are saved long-term to be used for analysis world-wide.

While an absolute necessity, this leaves a huge amount of generated data not accessible to the physicists. Furthermore, it is difficult to get an insight on what is being detected by the sub-systems in any given time. This could be mitigated by having a means to get statistics on the current data fed to the trigger, but not necessarily passed to the triggers downstream. This could be done by having a system very close to the trigger processor, to which a portion of the input data is fed to. Then using an FPGA, low latency and high performance pre-processing could be offloaded to the programmable logic, while a serial processor could use a high level program such as CERN's root to perform the high level processing. These statistics could then be accessed by the users, and gain insight on the data. The ZYNQ is a perfect candidate for such a computing scheme.

The way that this idea is validated in this thesis, is by generating random numbers in the Programmable Logic of the ZYNQ device, doing the high performance, low-level preprocessing in the FPGA, and then handle the data to the processor, running an operating system and ROOT to perform the high-level processing.

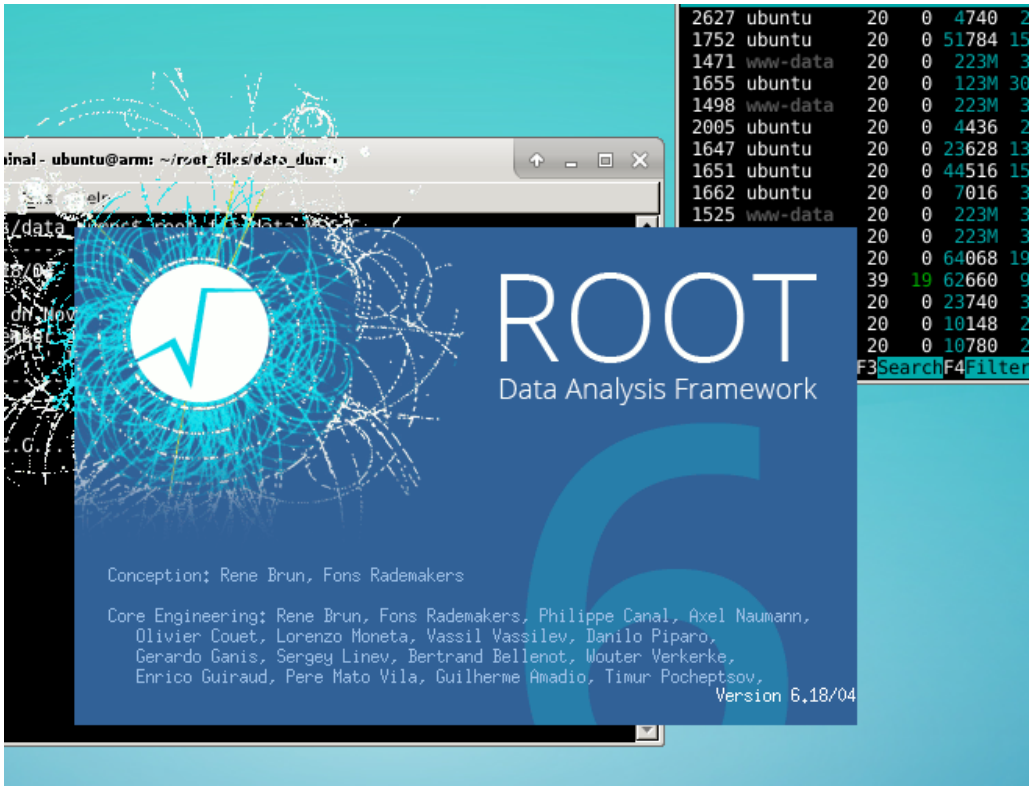


Figure 105. ROOT's splash screen running on the ZYNQ.

The way the "random" numbers are generated is by using LFSRs (Linear Feedback Shift Registers) in the FPGA. In fact, the generated numbers are pseudo-random numbers, because the circuit is purely deterministic, but makes a good approximation of a random number generator, especially if not applied to critical applications such as cryptography. In this case, it would be better to utilize an entropy source. The shift register is fed with a seed, and then the input is fed with the XOR-ed value of some bits in the register, called the taps. If the feedback function is well chosen, then the LSFR can have a very long repetition period.

These numbers would be uniformly distributed, but in this case data following a Gaussian distribution will be created. In order to create a Gaussian distribution, four of the outcomes of the LSFRs are summed each time. The Central Limit Theorem establishes that when independent random variables are added, their normalized sum tends towards a normal distribution. So, the numbers should follow a kind of (because this is not a perfect method, but rather a crude one to produce the data needed) Gaussian distribution. Every circuit that does this exact thing is operated at a frequency of 135 MHz. In total 9 such circuits are used and to total data-rate is ~5 Gbps.

After the data production, the output from all the 9 random number producer blocks are fed to another circuit, which bins them in real time. The binned data are then available in the

operating system running on the ARM processor, using DMA (Direct Memory Access) transactions. The ROOT, after compilation for the ARMv7 architecture, is fully functional on the ZYNQ. ROOT can then access the preprocessed data by just reading data from the memory locations that the data lie. After accessed by root the data are histogrammed. A lot more can be done, such as fitting of the data, and all these can be done in real time. Below is the histograms generated by root after starting the generation circuit, and thereafter sampled every few seconds. The results resemble a bell-like curve.

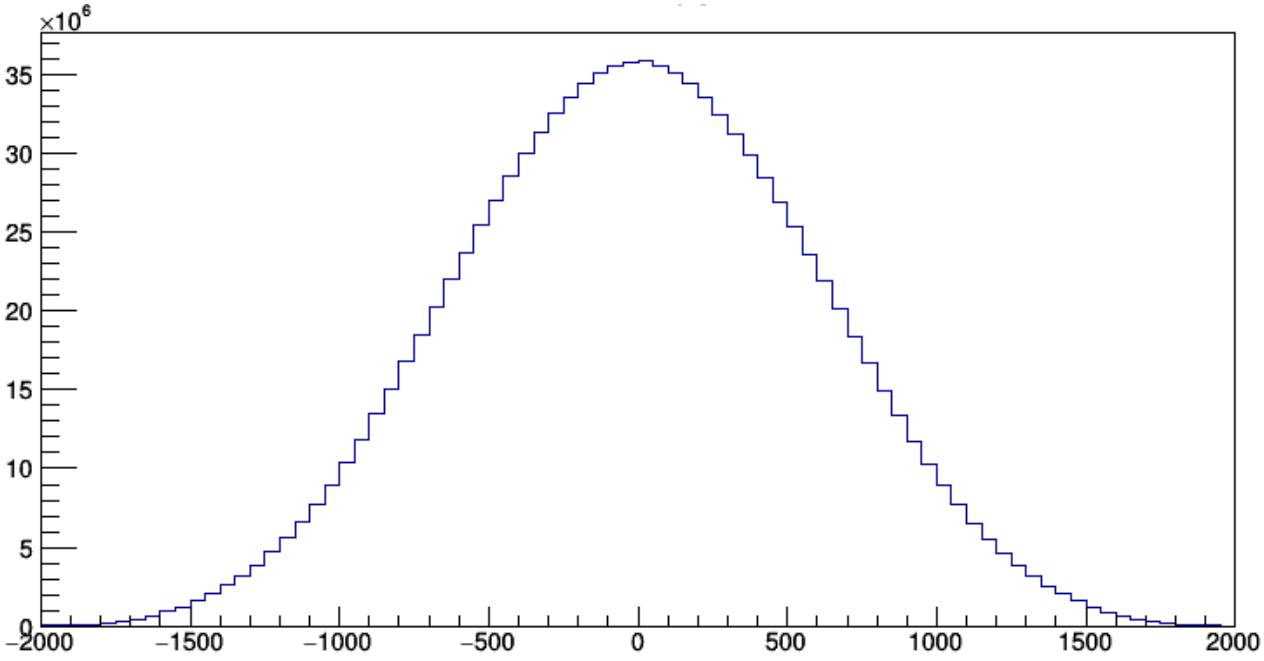


Figure 106. First sample of the random data.

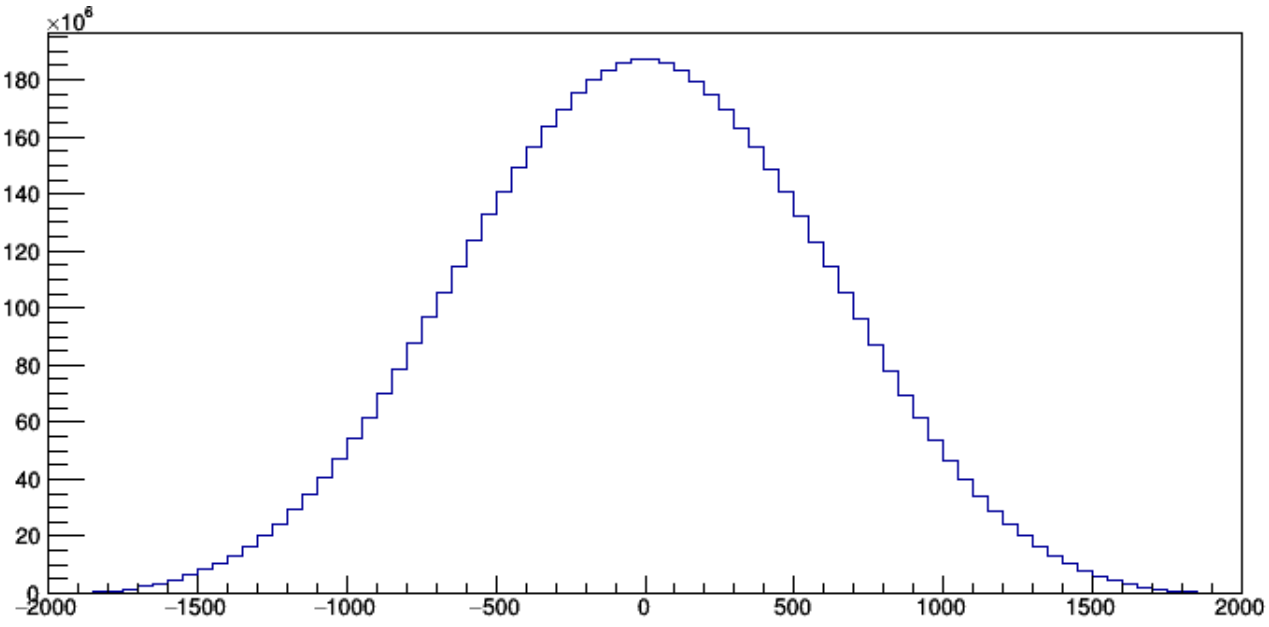


Figure 107. Second sample of the random data.

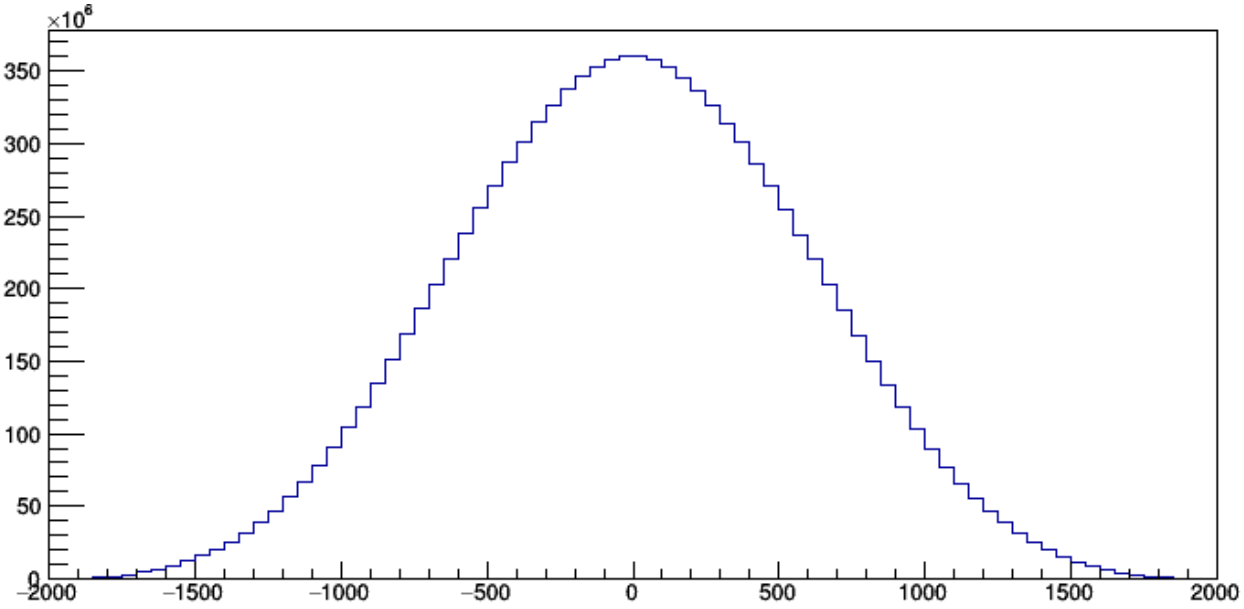


Figure 108. Third sample of the random data.

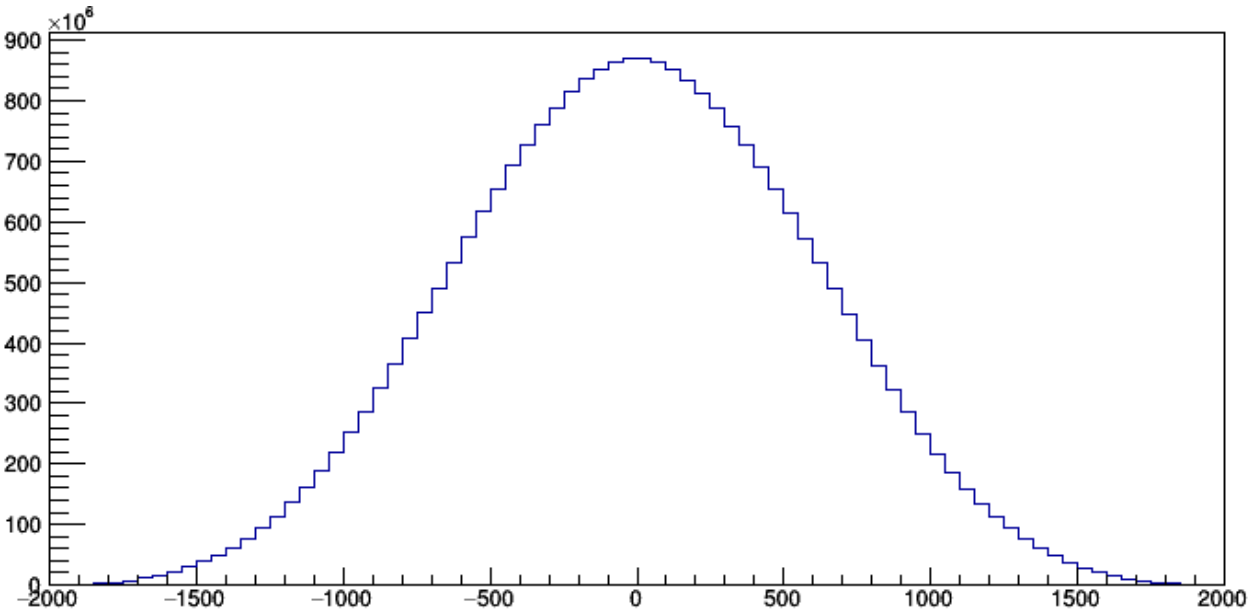


Figure 109. Fourth sample of the random data.

Chapter 6: Conclusion

A system that features novel computing architectures was designed and manufactured, while the cost was kept low. In total two boards were manufactured in the first production run, which were successful, and no re-spin of the design was needed. Even though the PCB was designed as a low cost, covering the absolutely minimum specifications, the signal integrity was kept at very good levels, often exceeding commercial boards of the same type, with higher PCB specifications which serve a less complex system. The ZYNQ device used on the board has been utilized to its limits on many aspects, such as I/O capabilities, MGT connectivity, USB connectivity, and the processor has been provided with the maximum memory possible.

There was no subsystem on the board that failed to operate as intended, however a small number of resistors were installed with the wrong value, due to wrong values provided on the BOM to the manufacturer, and were corrected by hand. Furthermore, due to a CAD error a connection was missing, which was again fixed by soldering, as well as two connections on the I2C muc. An errata is provided in the appendix.

Chapter 7: Appendix

7.1 References

[1] IPC, IPC-2141A, 2004.

[2] ISOLA, "IS400, High Performace Laminate and Prepreg," [Online]. Available: https://www.isola-group.com/wp-content/uploads/data-sheets/is400.pdf?v=1609968269.

[3] ISOLA, "IS400 Dk/Df table," [Online]. Available: https://www.isola-group.com/wp-content/uploads/data-sheets/IS400__Dk_Df_Tables.pdf?v=1609968269.

[4] NKK, "Series MS datasheet," [Online]. Available: https://www.nkkswitches.com/pdf/MSnonilluminated.pdf.

[5] MAXIM Integrated, "Single/Multiphase, Step-Down,DC-DC Converter," [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX8686.pdf.

[6] MAXIM Integrated, "Dual, 4A/2A, 4MHz, Step-DownDC-DC Regulator withTracking/Sequencing Capability," [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX15021.pdf.

[7] MAXIM Integrated, "High-Efficiency, 2A, Current-Mode Synchronous, Step-Down Switching Regulator," [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX15053.pdf.

[8] MAXIM Integrated, "Low-Voltage DDR Linear Regulator," [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX1510-MAX17510.pdf.

[9] MAXIM Integrated, "Low-Power, Fixed and Adjustable Reference," [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX6037-MAX6037C.pdf.

[10] MAXIM Integrated, "Low-Voltage, Low-Dropout Linear Regulator," [Online]. Available: https://datasheets.maximintegrated.com/en/ds/MAX1982-MAX1983.pdf.

[1 XILINX, "DS187, Zynq-7000 SoC DC and AC Switching Characteristics," [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds187-XC7Z010-XC7Z020-Data-Sheet.pdf.

[1 XILINX, "DS190, Zynq-7000 SoC Data Sheet: Overview," [Online]. Available: https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf.

[1 XILINX, "UG482, 7 Series FPGAs GTP Transceivers," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug482_7Series_GTP_Transceivers.pdf.

[1 XILINX, "Zynq-7000 SoC Packaging and Pinout," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug865-Zynq-7000-Pkg-Pinout.pdf.

[1 Micron, "2Gb: x4, x8, x16 DDR3 SDRAMFeatures," [Online]. Available: https://media-www.micron.com/-/media/client/global/documents/products/data-sheet/dram/ddr3/2gb_ddr3_sdram.pdf?rev=4bc67ac3a6f34250a2b73cb9db8c5502.

[1 XILINX, "Zynq-7000 SoCPCB Design Guide," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug933-Zynq-7000-PCB.pdf.

[1 Silicon Labs, "Si570/Si571 10 MHZ TO 1.4 GHZ I2C PROGRAMMABLE XO/VCXO," [Online]. Available: https://www.silabs.com/documents/public/data-sheets/si570.pdf.

[1 Marvell, "Integrated 10/100/1000 Mbps Energy Efficient Ethernet Transceiver," [Online]. Available: https://www.marvell.com/content/dam/marvell/en/public-collateral/transceivers/marvell-phys-transceivers-alaska-88e151x-datasheet.pdf.

[1 Cypress Semiconductor, "CY7C64225 USB-to-UART Bridge Controller," [Online]. Available: https://www.cypress.com/file/139421/download.

[2 Texas Instruments, "TXS01022-Bit BidirectionalVoltage-LevelTranslatorfor Open-Drainand Push-PullApplications," [Online]. Available: https://www.ti.com/lit/ds/symlink/txs0102.pdf?ts=1609968219372&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTXS0102.

[21] Littelfuse, "SESD Series Ultra Low Capacitance Diode Arrays," [Online]. Available: https://m.littelfuse.com/~/media/electronics/datasheets/tvs_diode_arrays/littelfuse_tvs_diode_array_sesd_ultra_low_capacitance_diode_arrays_datasheet.pdf.pdf.

[22] Texas Instruments, "SDIO PORT EXPANDER WITH VOLTAGE-LEVEL TRANSLATION," [Online]. Available: https://www.ti.com/lit/ds/sces682c/sces682c.pdf?ts=1609969834500&ref_url=https%253A%252F%252Fwww.google.com%252F.

[23] Microchip, "USB3320 Highly Integrated Full Featured Hi-Speed USB 2.0 ULPI Transceiver," [Online]. Available: https://cdn.hackaday.io/files/19354828041536/00001792E.pdf.

[24] FTDI, "FT600Q-FT601Q IC Datasheet(USB 3.0 to FIFO Bridge)," [Online]. Available: https://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT600Q-FT601Q%20IC%20Datasheet.pdf.

[25] Molex, "48393 USB 3.0 connector, product specifications," [Online]. Available: https://www.molex.com/pdm_docs/ps/PS-48393-001-001.pdf.

[26] NXP semiconductor, "PCA9548A 8-channel I2C-bus switch with reset, Rev. 5.1," [Online]. Available: https://www.nxp.com/docs/en/data-sheet/PCA9548A.pdf.

[27] XILINX, "7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter," [Online]. Available: https://www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf.

7.2 Figure list

Figure 1. Programmable Logic Architecture 15
Figure 2. A Xilinx Virtex Ultrascale+ VU19P device. The coin is placed to show how large the silicon area is. 19
Figure 3. VHDL code defining the ARM processor on a ZYNQ device, showing the exposed RAM interface. 21
Figure 4. Cut through a multi-layered PCB. Visible is the copper on different layers, as well as the through-hole via on the right. 24
Figure 5. PCB silkscreen and textures showing the polarity of the diode and the electrolytic capacitors. 26
Figure 6. Flying probe test of a semi-populated PCB 27
Figure 7. Typical reflow curve. 29
Figure 8. Minimum annular ring. 31
Figure 9. A via that is slightly mis-registered. 32
Figure 10. Pad to trace clearance. 32
Figure 11. Textures printed on silkscreen. The green layer is the solder mask. . 34
Figure 12. Microstrip transmission line. 35
Figure 13. Symmetric stripline conductor. 37
Figure 14. Edge coupled microstrip line. 38
Figure 15. Edge coupled stripline. 39
Figure 16. Asymmetric stripline. 39
Figure 17. Substrate structure. 40
Figure 18. Serpentine routed single - ended signals, matching the delay of the longest line (on the left). 40
Figure 19. Serpentine routing on a differential pair. Notice the little bumps on the shortest line in order to mitigate skew induced by the serpentine routing process. 41
Figure 20. 3D view of the board in Altium Designer. 42

Figure 21. 3D view of the board, with the clock of the byte lanes of the RAM highlighted. 43
Figure 22. Fanout of the ZYNQ device in the 5 signal layers. 44
Figure 23. IS400 2116 Dk/Df table. 45
Figure 24. IS400 7628 Dk/Df table. 45
Figure 25. The defined impedance buildup of the PCB. 46
Figure 26. Top view of the PCB..... 47
Figure 27. Bottom view of the PCB. 47
Figure 28. High level block diagram of the system. The width of the arrows corresponds to the interface throughput. 48
Figure 29. A simple buck converter model, and its two principal states. 50
Figure 30. Power plane 1..... 51
Figure 31. Power plane 2..... 52
Figure 32. 5V Power plane..... 53
Figure 33. 3.3V plane on power plane 1..... 54
Figure 34. 3V plane on power plane 2 55
Figure 35. VCCINT planes. 56
Figure 36. VCC1V8 Power plane..... 57
Figure 37. MGTAVCC power plane. 58
Figure 38. MGTAVTT Power plane. The extra expansion is to create a coherent plane to control the impedance of the MGT tracks. 59
Figure 39. VCCAUX Power plane. 60
Figure 40. DDR1V5 power plane..... 61
Figure 41. DDR0V75 Power plane. 62
Figure 42. DDVREF Power plane..... 62
Figure 43. Power sequencing order. 65
Figure 44. Selection of XILINX 7-Series ZYNQ Devices. 67
Figure 45. Block Diagram of the ZYNQ device..... 69
Figure 46. XC7Z015 ZYNQ device view. 70
Figure 47. Photo of the ZYNQ device. Visible on the right side is the DDR3 memory. 71

Figure 48. XC7Z015 bottom view. 72
Figure 49. Different memory technologies, their bandwidth and their price. DRAM sits in the middle. 75
Figure 50. The Micron RAM ICs, placed close to the FPGA. 76
Figure 51. Fly-By topology. 78
Figure 52.Impedance simulation of the bottom copper plane of the card. 81
Figure 53. Impedance simulation of an inner copper plane of the card. Visible is the different propagation velocity from the previous figure. 81
Figure 54. Termination resistors placed right after the last DRAM chip of the Fly-By chain. 83
Figure 55. Highlighted are the Address / CLK / Command lanes. Visible is the Fly-By topology, with every signal essentially passing from every RAM chip. 84
Figure 56. WE, RAS and CAS signals following the Fly-By topology. 85
Figure 57. The Clock signal and the Data Strobes differential pairs. 85
Figure 58. RAM interface signals. 86
Figure 59. Depiction of SerDes (Serializing on the left, DeSerializing on the right). 89
Figure 60. Clocking network of the MGT quad. 93
Figure 61. Ug482, p. 15. The Tx/Rx PCS and PMA blocks. 94
Figure 62. Eyes can functionality. The concepts of the data sampler and the offset sampler are visible. 95
Figure 63. The Si570 oscillator. 96
Figure 64. A fiber optic SFP module 97
Figure 65. The SFP optical assembly 98
Figure 66. The MGT SMA connectors on the board. 99
Figure 67. PCB Layer Stack for the MGT layers. 99
Figure 68. The MGTs location on the device die. 100
Figure 69. The MGT signals on the PCB. 101
Figure 70. Reference clock distribution inside the MGT quad. 101
Figure 71. An Ethernet frame. 102
Figure 72. Twister Pair (UTP) cable. 103

Figure 73. The Ethernet PHY and RJ-45 Jack on the board. 104
Figure 74. High-level diagram of the Ethernet interface. 106
Figure 75. The RGMII interface between the ZYNQ and the PHY. 106
Figure 76. The UART circuit. On the top is the USB-micro connector, in the middle the USB to UART Bridge, and in the bottom the level shifter IC. 109
Figure 77. The SD card nets. 111
Figure 78. The SD card circuitry and the respective connector. 111
Figure 79. The flash memory circuitry on the board. 112
Figure 80. The QSPI flash memory nets. 113
Figure 81. Different USB connectors. 114
Figure 82. USB 2.0 circuitry on the board. 116
Figure 83. ZYNQ to USB 2.0 PHY nets. 117
Figure 84. USB 3.0 bus signals. 120
Figure 85. USB 3.0 transceiver high speed lanes. 120
Figure 86. The USB 3.0 circuitry on the board. 121
Figure 87. The differential pair lanes travelling from the FPGA to the pin header connector. 124
Figure 88. The single-ended lanes travelling from the FPGA to the single-ended optimized connector. 125
Figure 89. Single-ended optimized connector signal pattern. Notice the ground shielding around the signals. 125
Figure 90. Differential-pair optimized connector signal pattern. Notice that now both the Positive and the negative lines of the differential pair are shielded, while maintaining the differential line coupling. 126
Figure 91. The pin-header connectors on the board. Only the single-ended connector is populated here. Note the series resistors above the pin header connectors and the ESD protection diodes below the connectors. 126
Figure 92. Typical I2C topology. 129
Figure 93. The I2C mux circuitry. 130
Figure 94. Typical XADC Block Diagram. 132
Figure 95. The 33.33MHz processor clock. 134

Figure 96. JTAG connection to the ZYNQ. 135
Figure 97. The LEDs lit up and blinking. 135
Figure 98. SDRAM read eye measurements. 136
Figure 99.SDRAM write eye measurements. 137
Figure 100. Zedboard's read eye measurements. 137
Figure 101. Zedboard's write eye measurements. 137
Figure 102. Eyescan on an MGT with an optic SFP attached, on external loopback mode 140
Figure 103. The BER extended testing. 141
Figure 104. Memtester terminal interface. 141
Figure 105. ROOT's splash screen running on the ZYNQ. 143
Figure 106. First sample of the random data. 144
Figure 107. Second sample of the random data. 144
Figure 108. Third sample of the random data. 145
Figure 109. Fourth sample of the random data. 145
Figure 110. The low inductance ground lead used to probe the power rails. 246
Figure 111. SFP1 @ 1.25 Gbps, eyescan dwell BER = $10-5$ 264
Figure 112. SFP1 @ 2.5 Gbps, eyescan dwell BER = $10-6$ 265
Figure 113. SFP1 @ 3.906 Gbps, eyescan dwell BER = $10-6$ 265
Figure 114. SFP1 @ 5 Gbps, eyescan dwell BER = $10-6$ 266
Figure 115. SFP1 @ 6.25 Gbps, eyescan dwell BER = $10-6$ 266
Figure 116. SFP1 @ 7.5 Gbps, eyescan dwell BER = $10-6$ 267
Figure 117. SFP1 @ 9.375 Gbps, eyescan dwell BER = $10-6$ 267
Figure 118. SFP1 @ 10 Gbps, eyescan dwell BER = $10-6$ 268
Figure 119. SFP2 @ 1.25 Gbps, eyescan dwell BER = $10-5$ 268
Figure 120. SFP2 @ 2.5 Gbps, eyescan dwell BER = $10-6$ 269
Figure 121. SFP2 @ 3.906 Gbps, eyescan dwell BER = $10-6$ 269
Figure 122. SFP2 @ 5 Gbps, eyescan dwell BER = $10-6$ 270
Figure 123. SFP2 @ 6.25 Gbps, eyescan dwell BER = $10-6$ 270
Figure 124. SFP2 @ 7.5 Gbps, eyescan dwell BER = $10-6$ 271

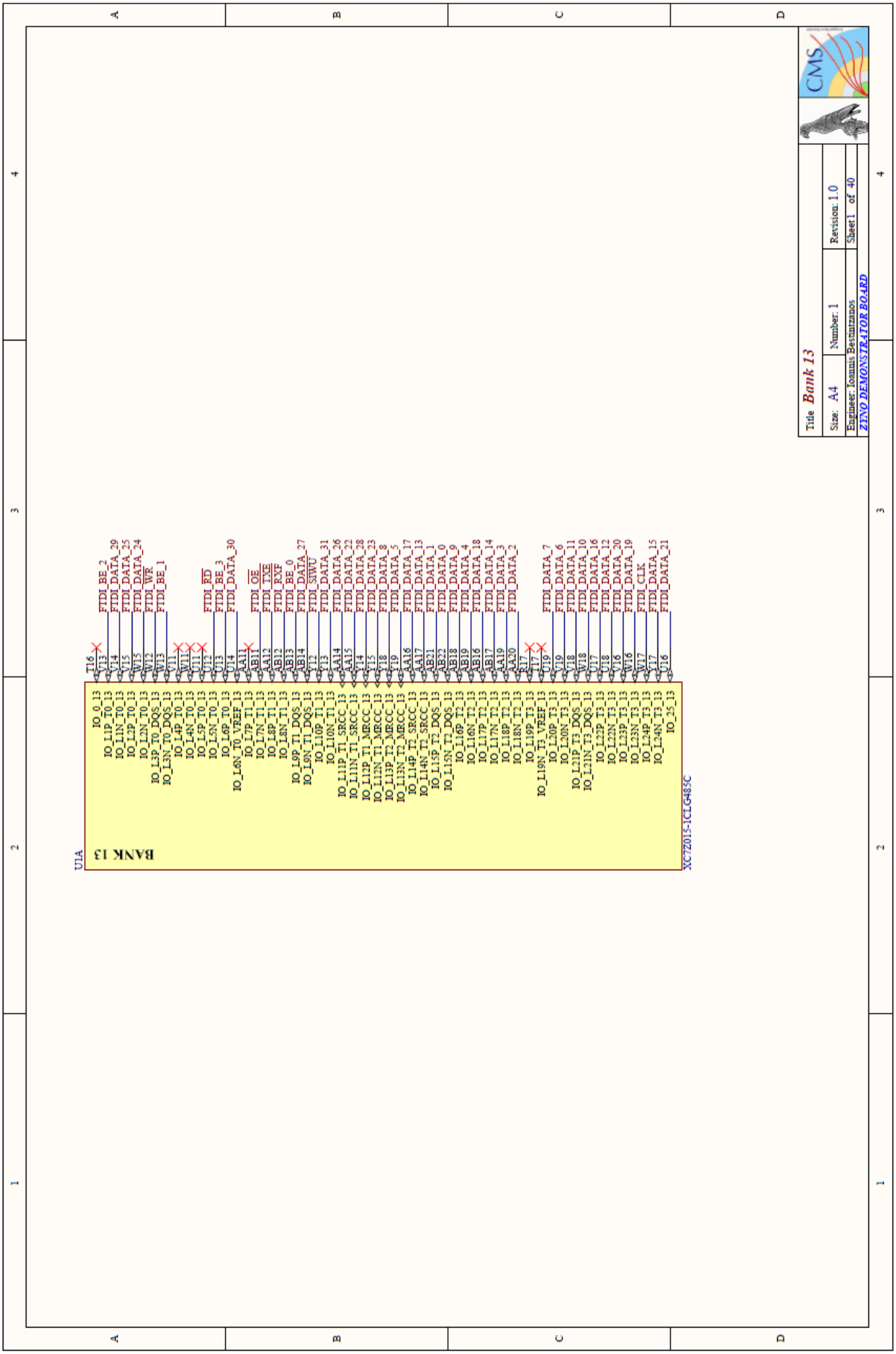
Figure 125. SFP2 @ 9.375 Gbps, eyescan dwell BER = $10 - 6$ 271
Figure 126. SFP2 @ 10 Gbps, eyescan dwell BER = $10 - 6$ 272
Figure 127. SMA1 @ 3.25 Gbps, eyescan dwell BER = $10 - 6$ 273
Figure 128. SMA1 @ 6.25 Gbps, eyescan dwell BER = $10 - 6$ 273
Figure 129. SMA2 @ 3.25 Gbps, eyescan dwell BER = $10 - 6$ 274
Figure 130. SMA1 @ 6.25 Gbps, eyescan dwell BER = $10 - 6$ 274

7.3 Schematics

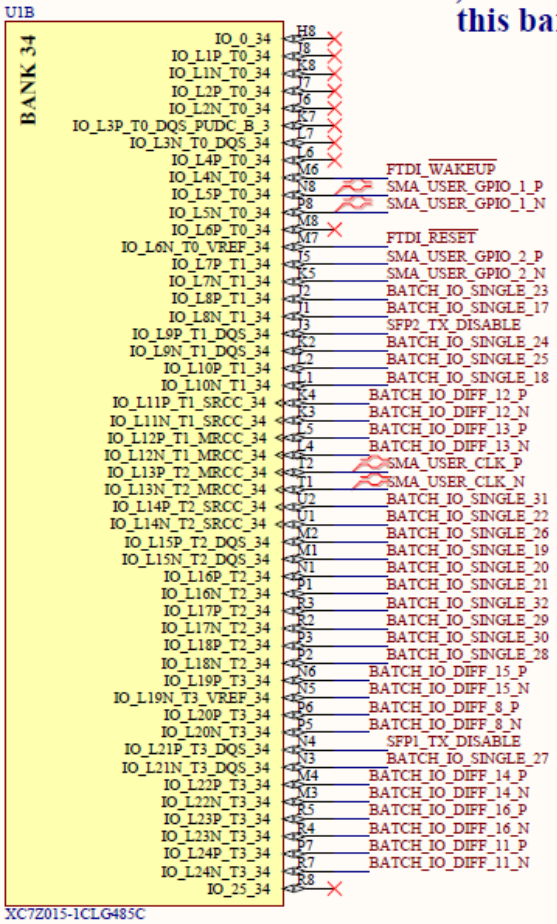
Below are attached the schematics of the board. In total there are 40 schematic sheets. The schematic sheets are as follows:

- ***Bank 13***
- ***Bank 34***
- ***Bank 35***
- ***PS MIO***
- ***PS DDR***
- ***MGTs***
- ***JTAG***
- ***FPGA power***
- ***FPGA decoupling***
- ***FPGA ground***
- ***RAM data***
- ***RAM termination***
- ***RAM power***
- ***RAM decoupling***
- ***Power 5V***
- ***Power VCCINT / MGTAVCC***
- ***Power VCC1V8***
- ***Power 3V3 / MGTAVTT***
- ***Power DDR1V5 / VCCAUX***
- ***Power sequencing***
- ***Gigabit Ethernet***
- ***SD Memory Card***
- ***Clocks***
- ***Optics***
- ***MGT SMAs***
- ***MGT reference plane decoupling***
- ***USB 2.0***

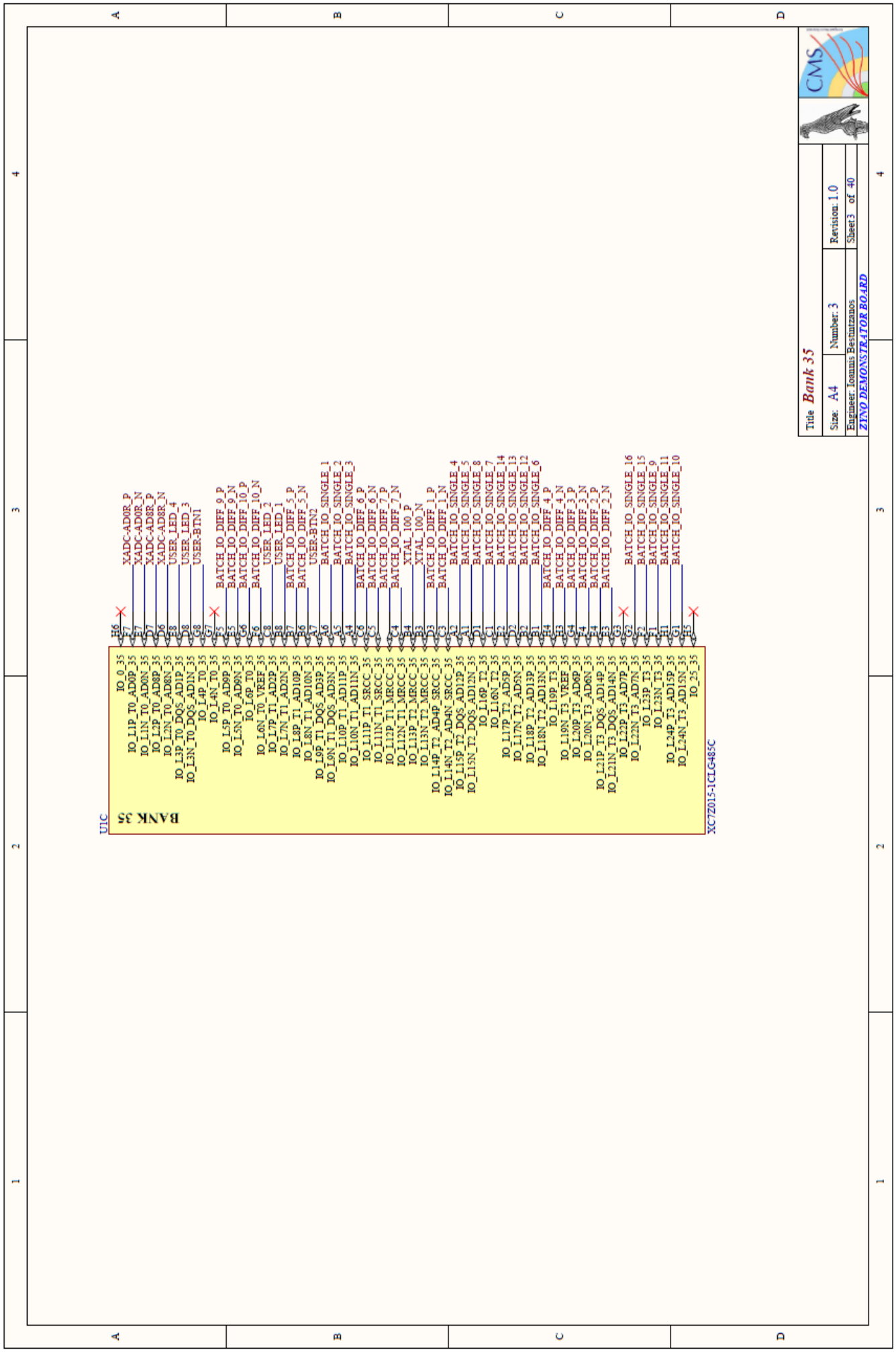
- ***USB 3.0***
- ***UART***
- ***QSPI flash memory***
- ***I2C Multiplexer***
- ***User LEDs***
- ***User buttons***
- ***Batch I/O single ended bits 1 to 16***
- ***Batch I/O single ended bits 17 to 32***
- ***Batch I/O differential bits 1 to 8***
- ***Batch I/O differential bits 9 to 16***
- ***SMA GPIO***

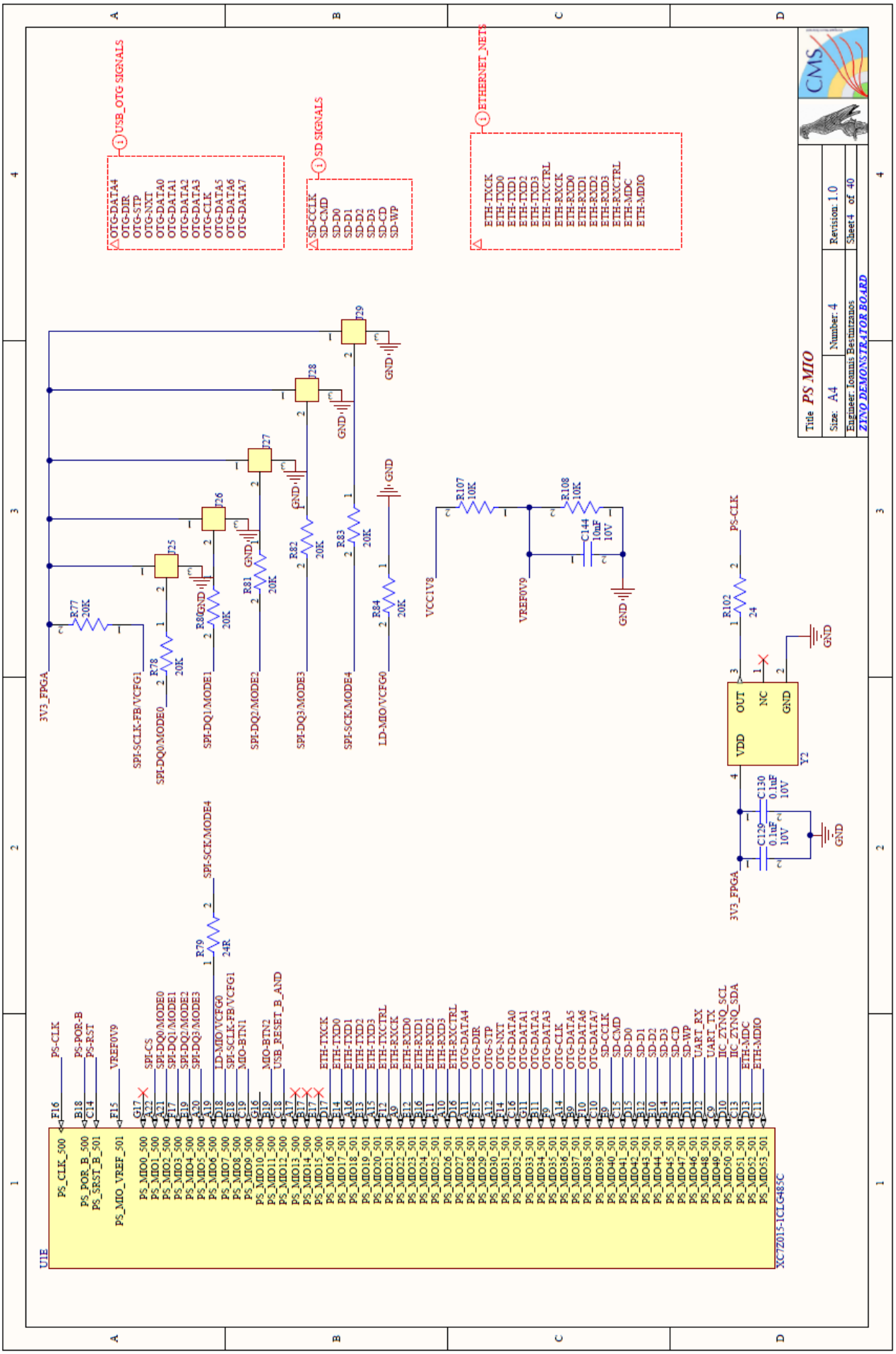

Title Bank 13
Size: A4
Number: 1
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 1 of 40
ZYNQ DEMONSTRATOR BOARD
U1A
BANK 13
XC7Z015-1CLG485C
FTDI_BE_2
FTDI_DATA_29
FTDI_DATA_25
FTDI_DATA_24
FTDI_WR
FTDI_BE_1
FTDI_RD
FTDI_BE_3
FTDI_DATA_30
FTDI_OE
FTDI_TXE
FTDI_RXF
FTDI_BE_0
FTDI_DATA_27
FTDI_SIWU
FTDI_DATA_31
FTDI_DATA_26
FTDI_DATA_22
FTDI_DATA_28
FTDI_DATA_23
FTDI_DATA_8
FTDI_DATA_5
FTDI_DATA_17
FTDI_DATA_13
FTDI_DATA_1
FTDI_DATA_0
FTDI_DATA_9
FTDI_DATA_4
FTDI_DATA_18
FTDI_DATA_14
FTDI_DATA_3
FTDI_DATA_2
FTDI_DATA_7
FTDI_DATA_6
FTDI_DATA_11
FTDI_DATA_10
FTDI_DATA_16
FTDI_DATA_12
FTDI_DATA_20
FTDI_DATA_19
FTDI_CLK
FTDI_DATA_15
FTDI_DATA_21

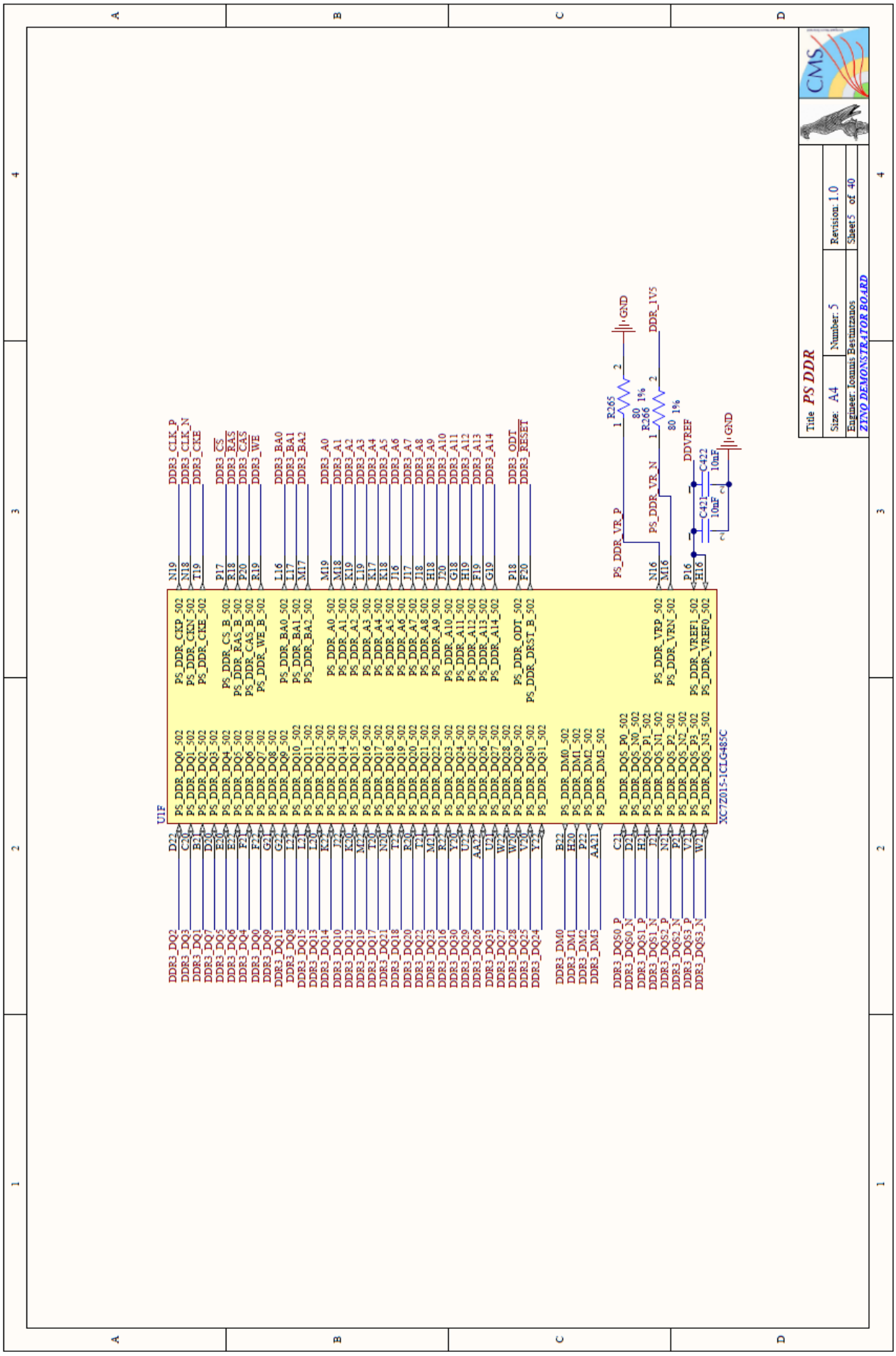
1.8V, 2.5V or 3.3V can be selected as the voltage of this bank via the configuration header.

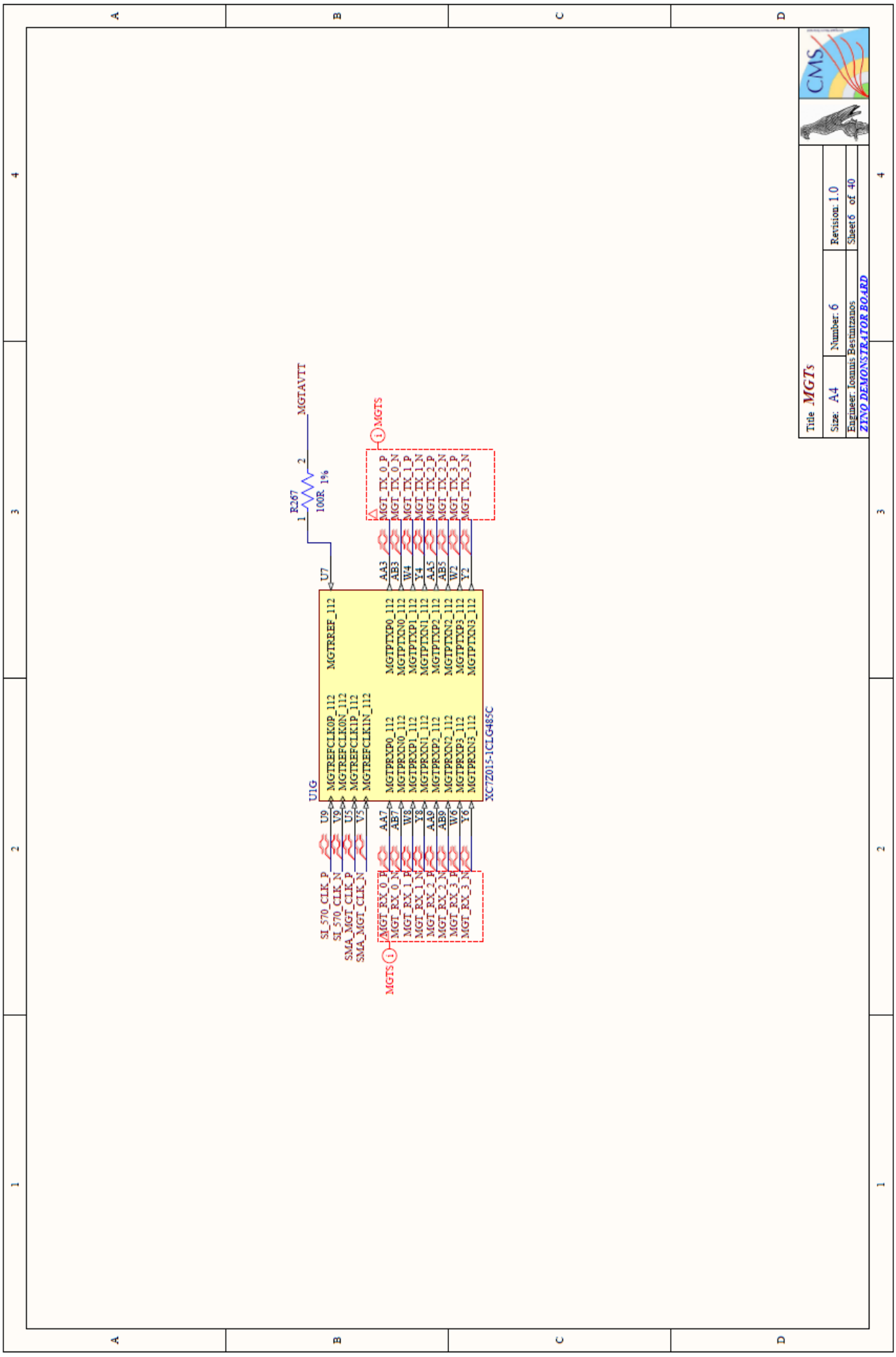


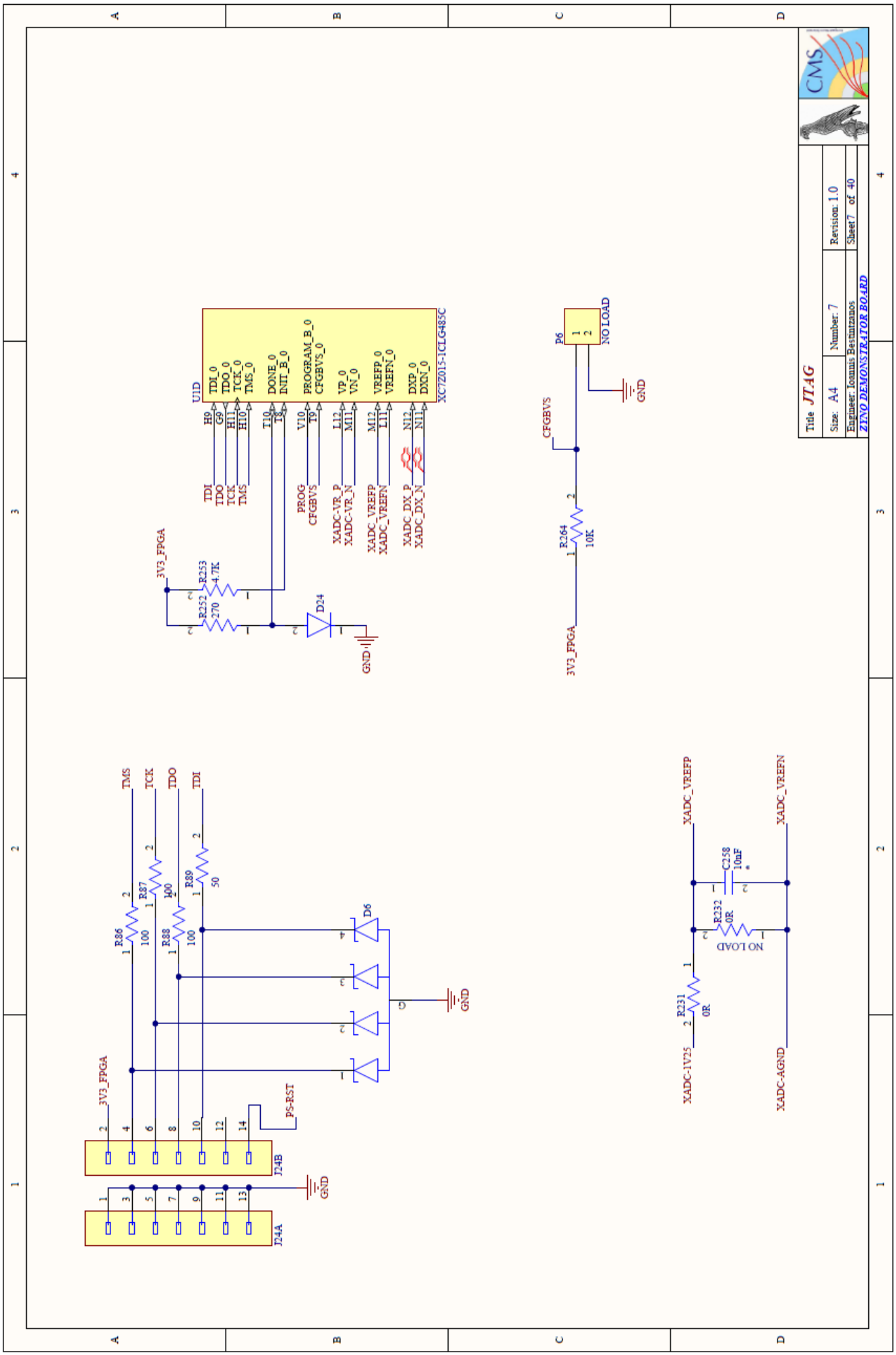
Title *Bank 34*		
Size: A4	Number: 2	Revision: 1.0
Engineer: Ioannis Bestintzanos		Sheet 2 of 40
ZYNQ DEMONSTRATOR BOARD		

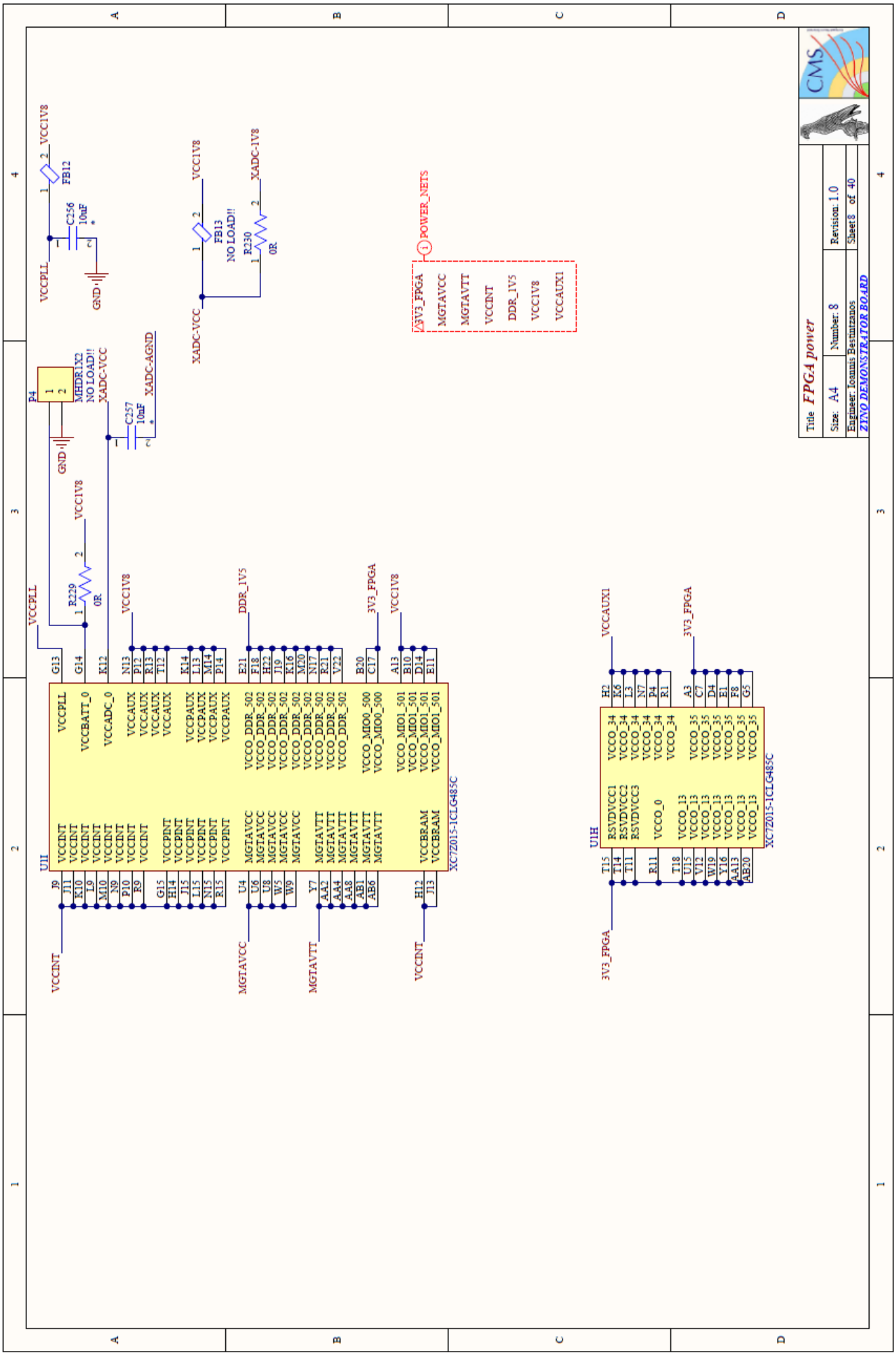

Title Bank 35
Size: A4
Number: 3
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 3 of 40
ZYNQ DEMONSTRATOR BOARD
U1C
BANK 35
XC7Z015-1CLG485C
IO_0_35
IO_L1P_T0_AD0P_35
IO_L1N_T0_AD0N_35
IO_L2P_T0_AD8P_35
IO_L2N_T0_AD8N_35
IO_L3P_T0_DQS_AD1P_35
IO_L3N_T0_DQS_AD1N_35
IO_L4P_T0_35
IO_L4N_T0_35
IO_L5P_T0_AD9P_35
IO_L5N_T0_AD9N_35
IO_L6P_T0_35
IO_L6N_T0_VREF_35
IO_L7P_T1_AD2P_35
IO_L7N_T1_AD2N_35
IO_L8P_T1_AD10P_35
IO_L8N_T1_AD10N_35
IO_L9P_T1_DQS_AD3P_35
IO_L9N_T1_DQS_AD3N_35
IO_L10P_T1_AD11P_35
IO_L10N_T1_AD11N_35
IO_L11P_T1_SRCC_35
IO_L11N_T1_SRCC_35
IO_L12P_T1_MRCC_35
IO_L12N_T1_MRCC_35
IO_L13P_T2_MRCC_35
IO_L13N_T2_MRCC_35
IO_L14P_T2_AD4P_SRCC_35
IO_L14N_T2_AD4N_SRCC_35
IO_L15P_T2_DQS_AD12P_35
IO_L15N_T2_DQS_AD12N_35
IO_L16P_T2_35
IO_L16N_T2_35
IO_L17P_T2_AD5P_35
IO_L17N_T2_AD5N_35
IO_L18P_T2_AD13P_35
IO_L18N_T2_AD13N_35
IO_L19P_T3_35
IO_L19N_T3_VREF_35
IO_L20P_T3_AD6P_35
IO_L20N_T3_AD6N_35
IO_L21P_T3_DQS_AD14P_35
IO_L21N_T3_DQS_AD14N_35
IO_L22P_T3_AD7P_35
IO_L22N_T3_AD7N_35
IO_L23P_T3_35
IO_L23N_T3_35
IO_L24P_T3_AD15P_35
IO_L24N_T3_AD15N_35
IO_25_35
XADC-AD0R_P
XADC-AD0R_N
XADC-AD8R_P
XADC-AD8R_N
USER_LED_4
USER_LED_3
USER-BTN1
BATCH_IO_DIFF_9_P
BATCH_IO_DIFF_9_N
BATCH_IO_DIFF_10_P
BATCH_IO_DIFF_10_N
USER_LED_2
USER_LED_1
BATCH_IO_DIFF_5_P
BATCH_IO_DIFF_5_N
USER-BTN2
BATCH_IO_SINGLE_1
BATCH_IO_SINGLE_2
BATCH_IO_SINGLE_3
BATCH_IO_DIFF_6_P
BATCH_IO_DIFF_6_N
BATCH_IO_DIFF_7_P
BATCH_IO_DIFF_7_N
XTAL_100_P
XTAL_100_N
BATCH_IO_DIFF_1_P
BATCH_IO_DIFF_1_N
BATCH_IO_SINGLE_4
BATCH_IO_SINGLE_5
BATCH_IO_SINGLE_8
BATCH_IO_SINGLE_7
BATCH_IO_SINGLE_14
BATCH_IO_SINGLE_13
BATCH_IO_SINGLE_12
BATCH_IO_SINGLE_6
BATCH_IO_DIFF_4_P
BATCH_IO_DIFF_4_N
BATCH_IO_DIFF_3_P
BATCH_IO_DIFF_3_N
BATCH_IO_DIFF_2_P
BATCH_IO_DIFF_2_N
BATCH_IO_SINGLE_16
BATCH_IO_SINGLE_15
BATCH_IO_SINGLE_9
BATCH_IO_SINGLE_11
BATCH_IO_SINGLE_10

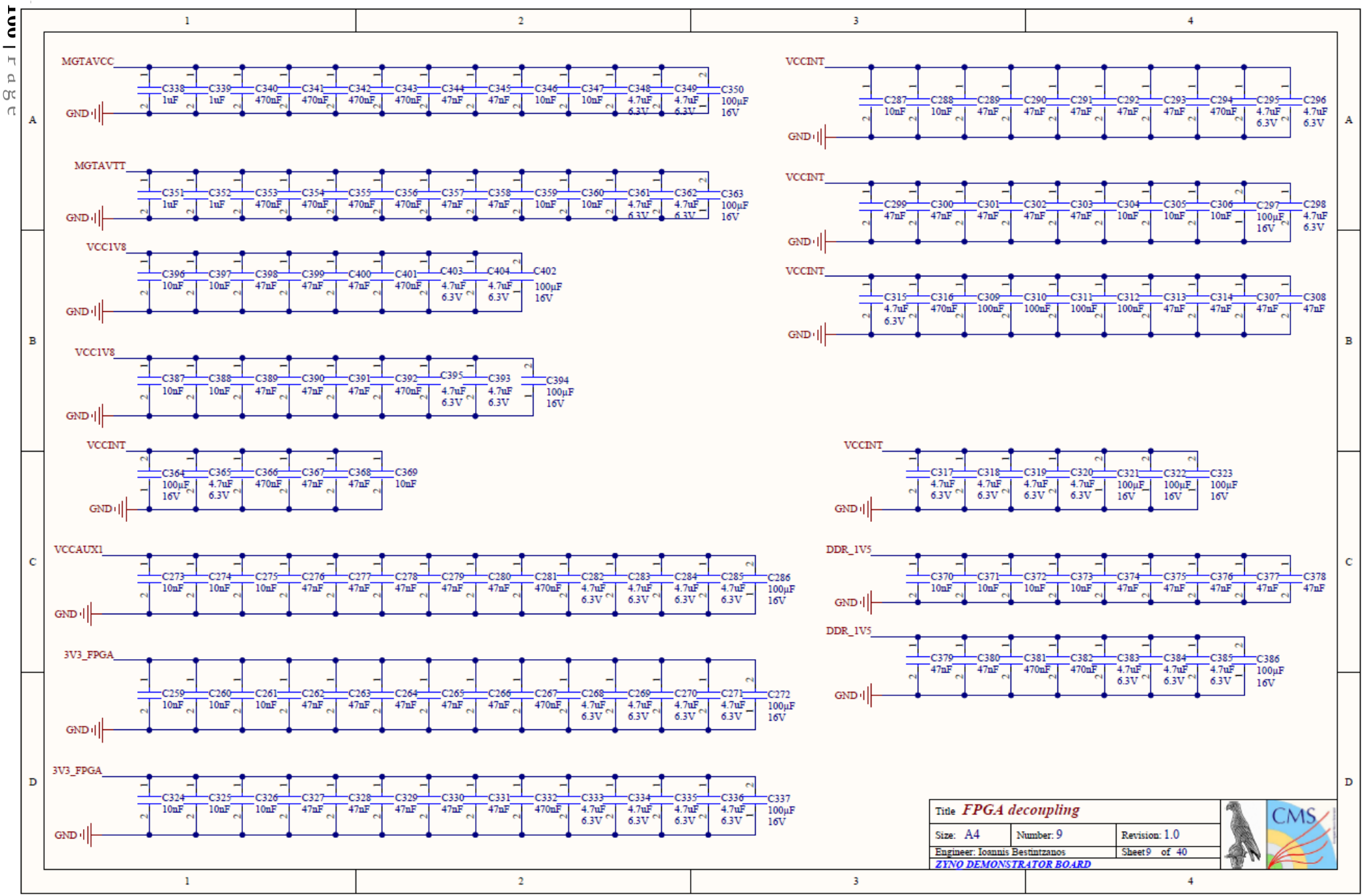

USB_OTG SIGNALS
OTG-DATA4
OTG-DIR
OTG-STP
OTG-NXT
OTG-DATA0
OTG-DATA1
OTG-DATA2
OTG-DATA3
OTG-CLK
OTG-DATA5
OTG-DATA6
OTG-DATA7
SD SIGNALS
SD-CCLK
SD-CMD
SD-D0
SD-D1
SD-D2
SD-D3
SD-CD
SD-WP
ETHERNET_NETS
ETH-TXCK
ETH-TXD0
ETH-TXD1
ETH-TXD2
ETH-TXD3
ETH-TXCTRL
ETH-RXCK
ETH-RXD0
ETH-RXD1
ETH-RXD2
ETH-RXD3
ETH-RXCTRL
ETH-MDC
ETH-MDIO
Title PS MIO
Size: A4
Number: 4
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 4 of 40
ZYNQ DEMONSTRATOR BOARD
U1E
XC7Z015-1CLG485C
3V3_FPGA
VCC1V8
VREF0V9
GND
PS-CLK
PS-POR-B
PS-SRST
VREF0V9
SPI-CS
SPI-DQ0/MODE0
SPI-DQ1/MODE1
SPI-DQ2/MODE2
SPI-DQ3/MODE3
SPI-SCK/MODE4
LD-MIO/VCFG0
SPI-SCLK-FB/VCFG1
MIO-BTN1
MIO-BTN2
USB_RESET_B_AND
ETH-TXCK
ETH-TXD0
ETH-TXD1
ETH-TXD2
ETH-TXD3
ETH-TXCTRL
ETH-RXCK
ETH-RXD0
ETH-RXD1
ETH-RXD2
ETH-RXD3
ETH-RXCTRL
OTG-DATA4
OTG-DIR
OTG-STP
OTG-NXT
OTG-DATA0
OTG-DATA1
OTG-DATA2
OTG-DATA3
OTG-CLK
OTG-DATA5
OTG-DATA6
OTG-DATA7
SD-CCLK
SD-CMD
SD-D0
SD-D1
SD-D2
SD-D3
SD-CD
SD-WP
UART_RX
UART_TX
IIC_ZYNQ_SCL
IIC_ZYNQ_SDA
ETH-MDC
ETH-MDIO
R77 20K
R78 20K
R80 20K
R81 20K
R82 20K
R83 20K
R84 20K
R79 24R
R107 10K
R108 10K
C144 10nF 10V
R102 24
C130 0.1uF 10V
C129 0.1uF 10V
Y2
VDD
OUT
NC
GND
J25
J26
J27
J28
J29

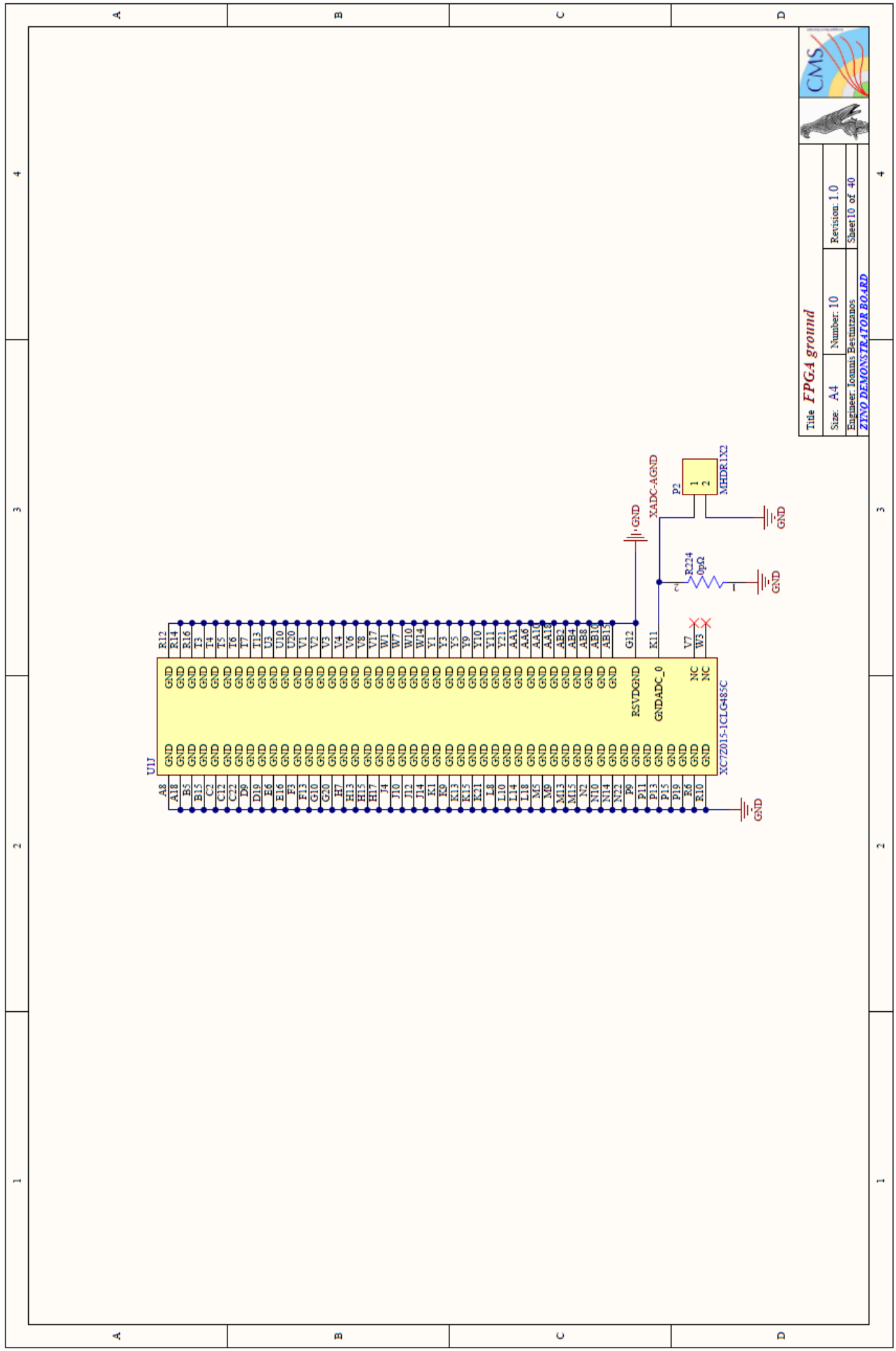

PS DDR
Size: A4
Number: 5
Revision: 1.0
Engineer: Ioannis Besnurzanos
Sheet 5 of 40
ZYNQ DEMONSTRATOR BOARD
U1F
XC7Z015-1CLG485C
PS_DDR_CKP_502
PS_DDR_CKN_502
PS_DDR_CKE_502
PS_DDR_CS_B_502
PS_DDR_RAS_B_502
PS_DDR_CAS_B_502
PS_DDR_WE_B_502
PS_DDR_BA0_502
PS_DDR_BA1_502
PS_DDR_BA2_502
PS_DDR_A0_502
PS_DDR_A1_502
PS_DDR_A2_502
PS_DDR_A3_502
PS_DDR_A4_502
PS_DDR_A5_502
PS_DDR_A6_502
PS_DDR_A7_502
PS_DDR_A8_502
PS_DDR_A9_502
PS_DDR_A10_502
PS_DDR_A11_502
PS_DDR_A12_502
PS_DDR_A13_502
PS_DDR_A14_502
PS_DDR_ODT_502
PS_DDR_DRST_B_502
PS_DDR_VRP_502
PS_DDR_VRN_502
PS_DDR_VREF1_502
PS_DDR_VREF0_502
PS_DDR_DQ0_502
PS_DDR_DQ1_502
PS_DDR_DQ2_502
PS_DDR_DQ3_502
PS_DDR_DQ4_502
PS_DDR_DQ5_502
PS_DDR_DQ6_502
PS_DDR_DQ7_502
PS_DDR_DQ8_502
PS_DDR_DQ9_502
PS_DDR_DQ10_502
PS_DDR_DQ11_502
PS_DDR_DQ12_502
PS_DDR_DQ13_502
PS_DDR_DQ14_502
PS_DDR_DQ15_502
PS_DDR_DQ16_502
PS_DDR_DQ17_502
PS_DDR_DQ18_502
PS_DDR_DQ19_502
PS_DDR_DQ20_502
PS_DDR_DQ21_502
PS_DDR_DQ22_502
PS_DDR_DQ23_502
PS_DDR_DQ24_502
PS_DDR_DQ25_502
PS_DDR_DQ26_502
PS_DDR_DQ27_502
PS_DDR_DQ28_502
PS_DDR_DQ29_502
PS_DDR_DQ30_502
PS_DDR_DQ31_502
PS_DDR_DM0_502
PS_DDR_DM1_502
PS_DDR_DM2_502
PS_DDR_DM3_502
PS_DDR_DQS_P0_502
PS_DDR_DQS_N0_502
PS_DDR_DQS_P1_502
PS_DDR_DQS_N1_502
PS_DDR_DQS_P2_502
PS_DDR_DQS_N2_502
PS_DDR_DQS_P3_502
PS_DDR_DQS_N3_502
DDR3_CLK_P
DDR3_CLK_N
DDR3_CKE
DDR3_CS
DDR3_RAS
DDR3_CAS
DDR3_WE
DDR3_BA0
DDR3_BA1
DDR3_BA2
DDR3_A0
DDR3_A1
DDR3_A2
DDR3_A3
DDR3_A4
DDR3_A5
DDR3_A6
DDR3_A7
DDR3_A8
DDR3_A9
DDR3_A10
DDR3_A11
DDR3_A12
DDR3_A13
DDR3_A14
DDR3_ODT
DDR3_RESET
DDR3_DQ0
DDR3_DQ1
DDR3_DQ2
DDR3_DQ3
DDR3_DQ4
DDR3_DQ5
DDR3_DQ6
DDR3_DQ7
DDR3_DQ8
DDR3_DQ9
DDR3_DQ10
DDR3_DQ11
DDR3_DQ12
DDR3_DQ13
DDR3_DQ14
DDR3_DQ15
DDR3_DQ16
DDR3_DQ17
DDR3_DQ18
DDR3_DQ19
DDR3_DQ20
DDR3_DQ21
DDR3_DQ22
DDR3_DQ23
DDR3_DQ24
DDR3_DQ25
DDR3_DQ26
DDR3_DQ27
DDR3_DQ28
DDR3_DQ29
DDR3_DQ30
DDR3_DQ31
DDR3_DM0
DDR3_DM1
DDR3_DM2
DDR3_DM3
DDR3_DQS0_P
DDR3_DQS0_N
DDR3_DQS1_P
DDR3_DQS1_N
DDR3_DQS2_P
DDR3_DQS2_N
DDR3_DQS3_P
DDR3_DQS3_N
R265
R266
80 1%
PS_DDR_VR_P
PS_DDR_VR_N
DDR_1V5
GND
DDVREF
C421
C422
10nF

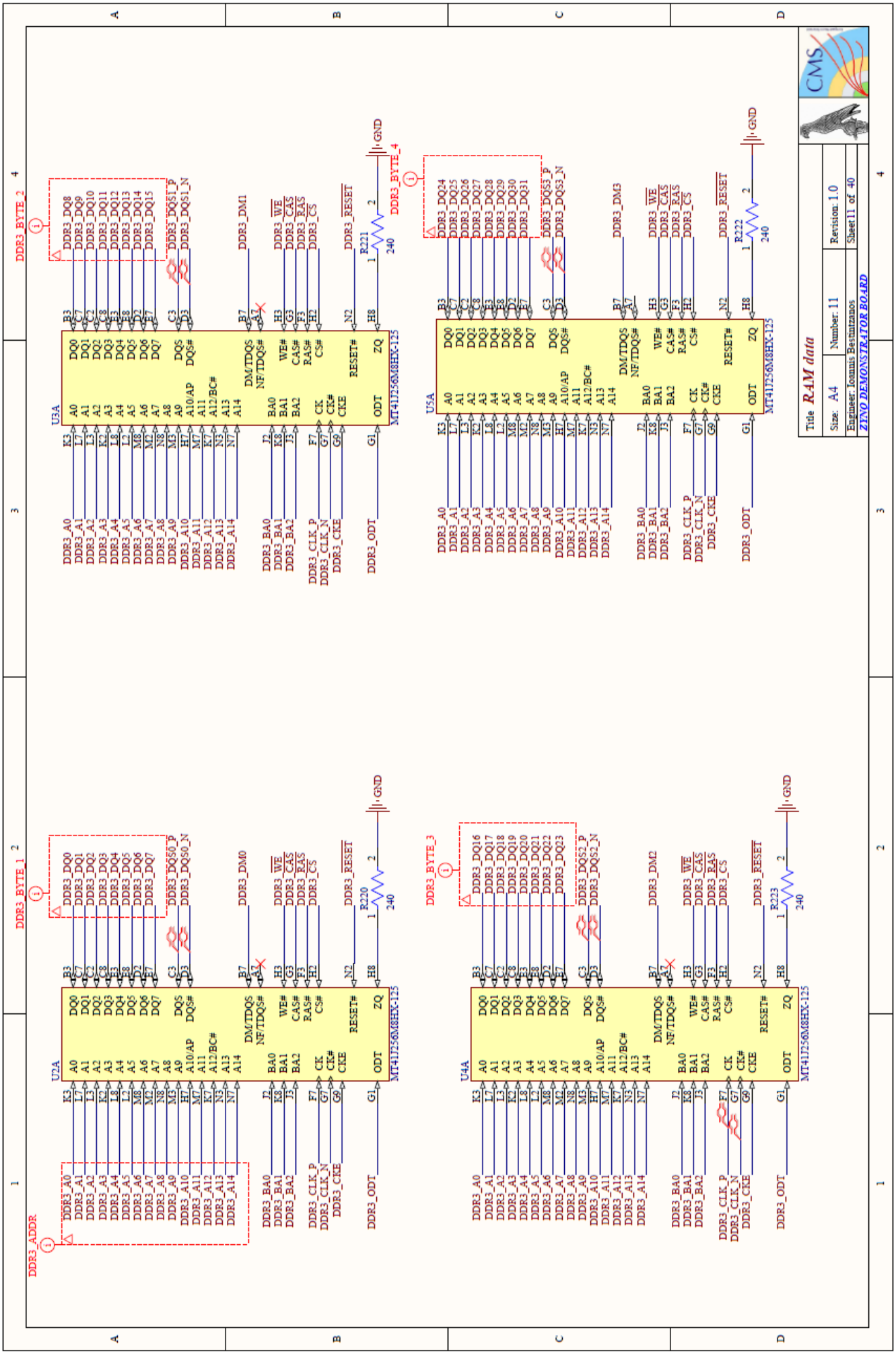

MGTAVTT
R267
100R 1%
U7
MGTRREF_112
MGTREFCLK0P_112
MGTREFCLK0N_112
MGTREFCLK1P_112
MGTREFCLK1N_112
MGTPRXP0_112
MGTPRXN0_112
MGTPRXP1_112
MGTPRXN1_112
MGTPRXP2_112
MGTPRXN2_112
MGTPRXP3_112
MGTPRXN3_112
MGTPTXP0_112
MGTPTXN0_112
MGTPTXP1_112
MGTPTXN1_112
MGTPTXP2_112
MGTPTXN2_112
MGTPTXP3_112
MGTPTXN3_112
U1G
XC7Z015-1CLG485C
SI_570_CLK_P
SI_570_CLK_N
SMA_MGT_CLK_P
SMA_MGT_CLK_N
MGTS
MGTS
MGT_RX_0_P
MGT_RX_0_N
MGT_RX_1_P
MGT_RX_1_N
MGT_RX_2_P
MGT_RX_2_N
MGT_RX_3_P
MGT_RX_3_N
MGT_TX_0_P
MGT_TX_0_N
MGT_TX_1_P
MGT_TX_1_N
MGT_TX_2_P
MGT_TX_2_N
MGT_TX_3_P
MGT_TX_3_N
Title MGTs
Size: A4
Number: 6
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 6 of 40
ZYNQ DEMONSTRATOR BOARD

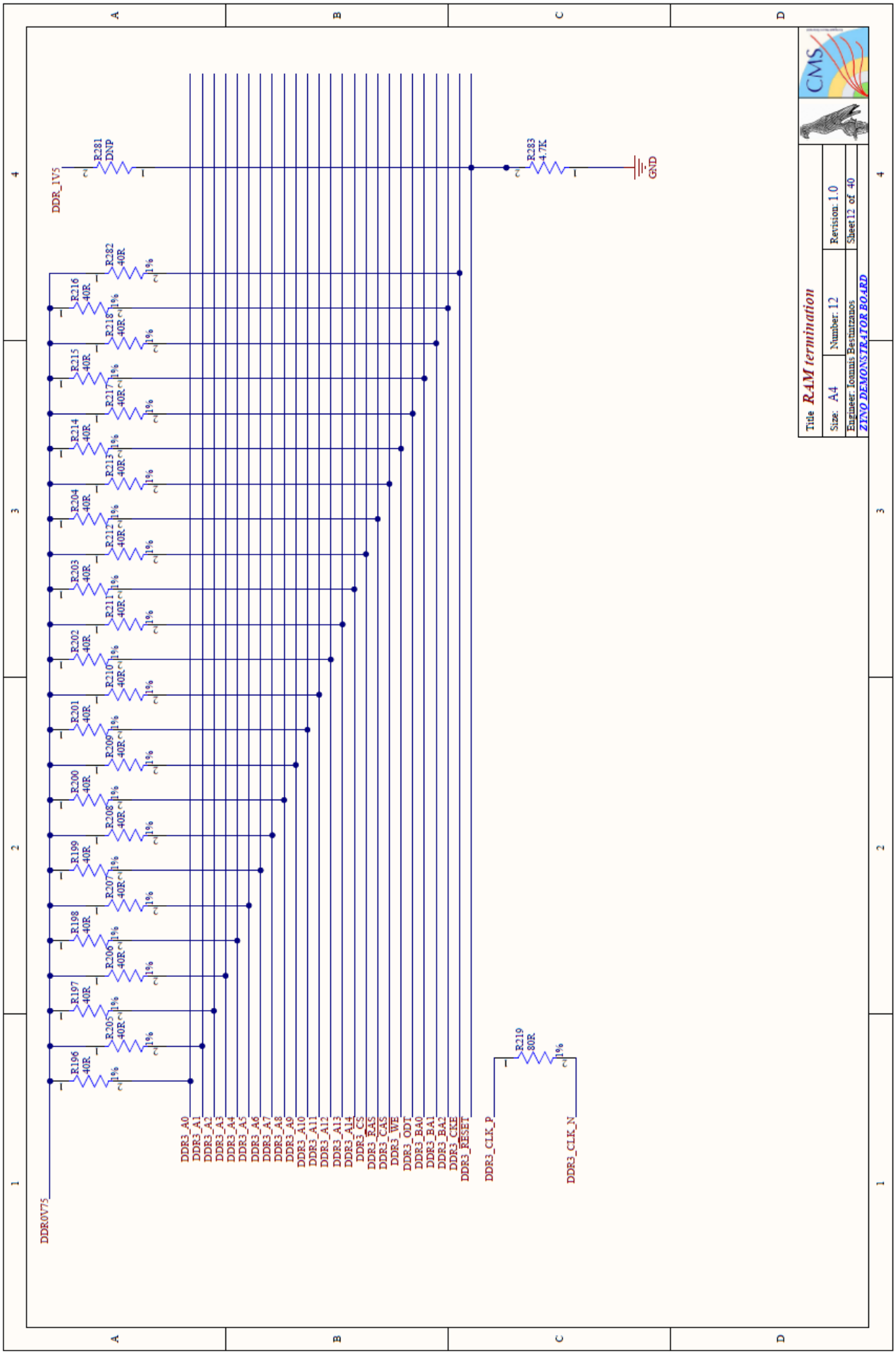

U1D
XC7Z015-1CLG485C
TDI_0
TDO_0
TCK_0
TMS_0
DONE_0
INIT_B_0
PROGRAM_B_0
CFGBVS_0
VP_0
VN_0
VREFP_0
VREFN_0
DXP_0
DXN_0
TDI
TDO
TCK
TMS
PROG
CFGBVS
XADC-VR_P
XADC-VR_N
XADC_VREFP
XADC_VREFN
XADC_DX_P
XADC_DX_N
3V3_FPGA
R253
4.7K
R252
270
D24
GND
P6
NO LOAD
CFGBVS
R264
10K
R86
100
R87
100
R88
100
R89
50
D6
J24A
J24B
PS-RST
C258
10uF
R232
0R
NO LOAD
R231
0R
XADC-1V25
XADC-AGND
Title JTAG
Size: A4
Number: 7
Revision: 1.0
Engineer Ioannis Bestnizanos
Sheet 7 of 40
ZYNQ DEMONSTRATOR BOARD

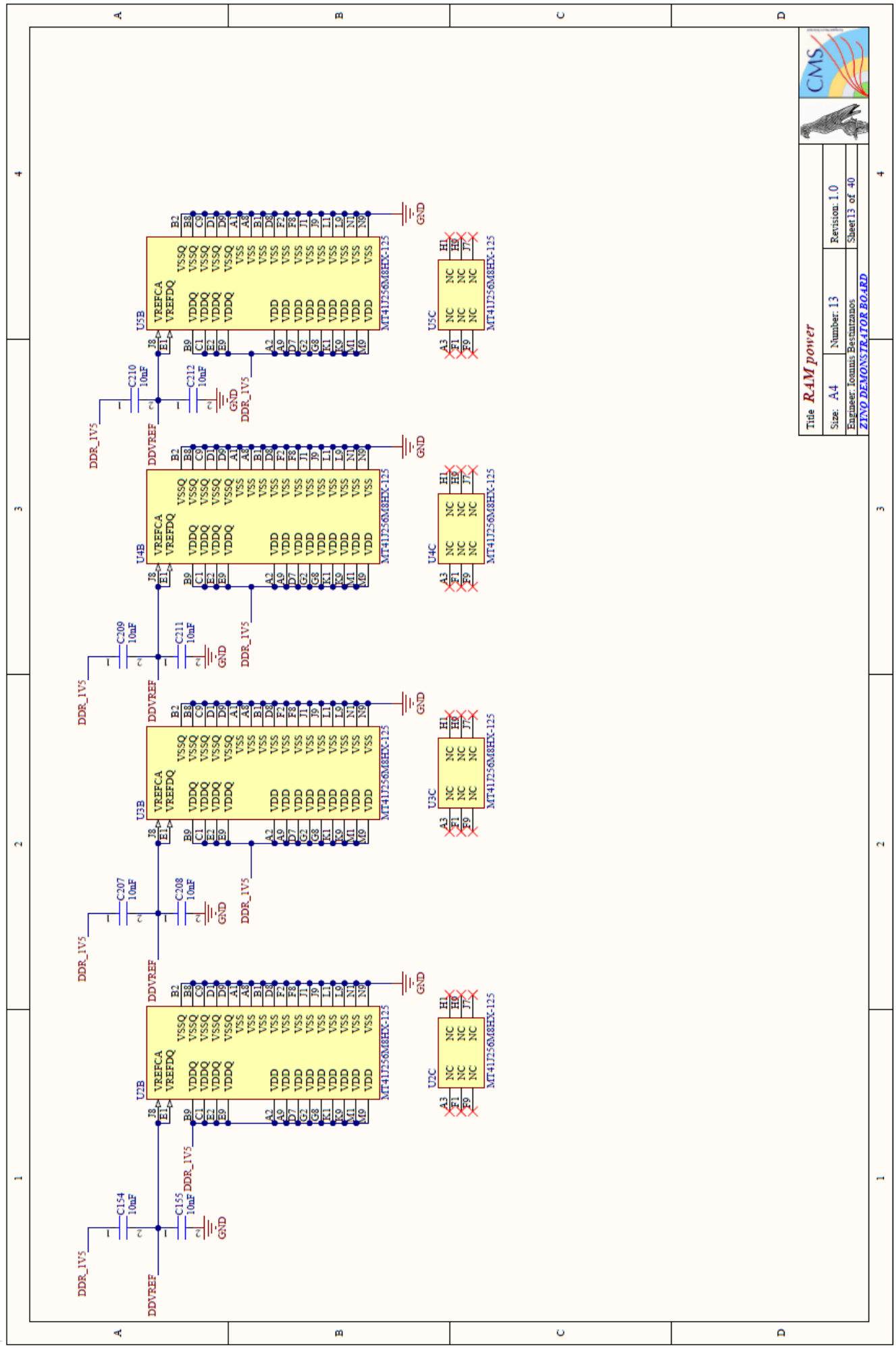

Title FPGA power
Size: A4
Number: 8
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 8 of 40
ZYNQ DEMONSTRATOR BOARD
POWER_NETS
3V3_FPGA
MGTAVCC
MGTAVTT
VCCINT
DDR_1V5
VCC1V8
VCCAUX1
U1I
XC7Z015-1CLG485C
U1H
XC7Z015-1CLG485C
VCCPLL
VCC1V8
XADC-VCC
XADC-AGND
XADC-1V8
VCCAUX1
3V3_FPGA
DDR_1V5
MGTAVCC
MGTAVTT
VCCINT
P4
MHDR1X2
NO LOAD!!
R229
0R
R230
0R
C256
10uF
C257
10uF
FB12
FB13
NO LOAD!!


MGTAVCC
GND
C338 1uF
C339 1uF
C340 470nF
C341 470nF
C342 470nF
C343 470nF
C344 47nF
C345 47nF
C346 10nF
C347 10nF
C348 4.7uF 6.3V
C349 4.7uF 6.3V
C350 100µF 16V
MGTAVTT
GND
C351 1uF
C352 1uF
C353 470nF
C354 470nF
C355 470nF
C356 470nF
C357 47nF
C358 47nF
C359 10nF
C360 10nF
C361 4.7uF 6.3V
C362 4.7uF 6.3V
C363 100µF 16V
VCC1V8
GND
C396 10nF
C397 10nF
C398 47nF
C399 47nF
C400 47nF
C401 470nF
C403 4.7uF 6.3V
C404 4.7uF 6.3V
C402 100µF 16V
VCC1V8
GND
C387 10nF
C388 10nF
C389 47nF
C390 47nF
C391 47nF
C392 470nF
C395 4.7uF 6.3V
C393 4.7uF 6.3V
C394 100µF 16V
VCCINT
GND
C364 100µF 16V
C365 4.7uF 6.3V
C366 470nF
C367 47nF
C368 47nF
C369 10nF
VCCAUX1
GND
C273 10nF
C274 10nF
C275 10nF
C276 47nF
C277 47nF
C278 47nF
C279 47nF
C280 47nF
C281 470nF
C282 4.7uF 6.3V
C283 4.7uF 6.3V
C284 4.7uF 6.3V
C285 4.7uF 6.3V
C286 100µF 16V
3V3_FPGA
GND
C259 10nF
C260 10nF
C261 10nF
C262 47nF
C263 47nF
C264 47nF
C265 47nF
C266 47nF
C267 470nF
C268 4.7uF 6.3V
C269 4.7uF 6.3V
C270 4.7uF 6.3V
C271 4.7uF 6.3V
C272 100µF 16V
3V3_FPGA
GND
C324 10nF
C325 10nF
C326 10nF
C327 47nF
C328 47nF
C329 47nF
C330 47nF
C331 47nF
C332 470nF
C333 4.7uF 6.3V
C334 4.7uF 6.3V
C335 4.7uF 6.3V
C336 4.7uF 6.3V
C337 100µF 16V
VCCINT
GND
C287 10nF
C288 10nF
C289 47nF
C290 47nF
C291 47nF
C292 47nF
C293 47nF
C294 470nF
C295 4.7uF 6.3V
C296 4.7uF 6.3V
VCCINT
GND
C299 47nF
C300 47nF
C301 47nF
C302 47nF
C303 47nF
C304 10nF
C305 10nF
C306 10nF
C297 100µF 16V
C298 4.7uF 6.3V
VCCINT
GND
C315 4.7uF 6.3V
C316 470nF
C309 100nF
C310 100nF
C311 100nF
C312 100nF
C313 47nF
C314 47nF
C307 47nF
C308 47nF
VCCINT
GND
C317 4.7uF 6.3V
C318 4.7uF 6.3V
C319 4.7uF 6.3V
C320 4.7uF 6.3V
C321 100µF 16V
C322 100µF 16V
C323 100µF 16V
DDR_1V5
GND
C370 10nF
C371 10nF
C372 10nF
C373 10nF
C374 47nF
C375 47nF
C376 47nF
C377 47nF
C378 47nF
DDR_1V5
GND
C379 47nF
C380 47nF
C381 470nF
C382 470nF
C383 4.7uF 6.3V
C384 4.7uF 6.3V
C385 4.7uF 6.3V
C386 100µF 16V
Title FPGA decoupling
Size: A4
Number: 9
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 9 of 40
ZYNO DEMONSTRATOR BOARD
CMS

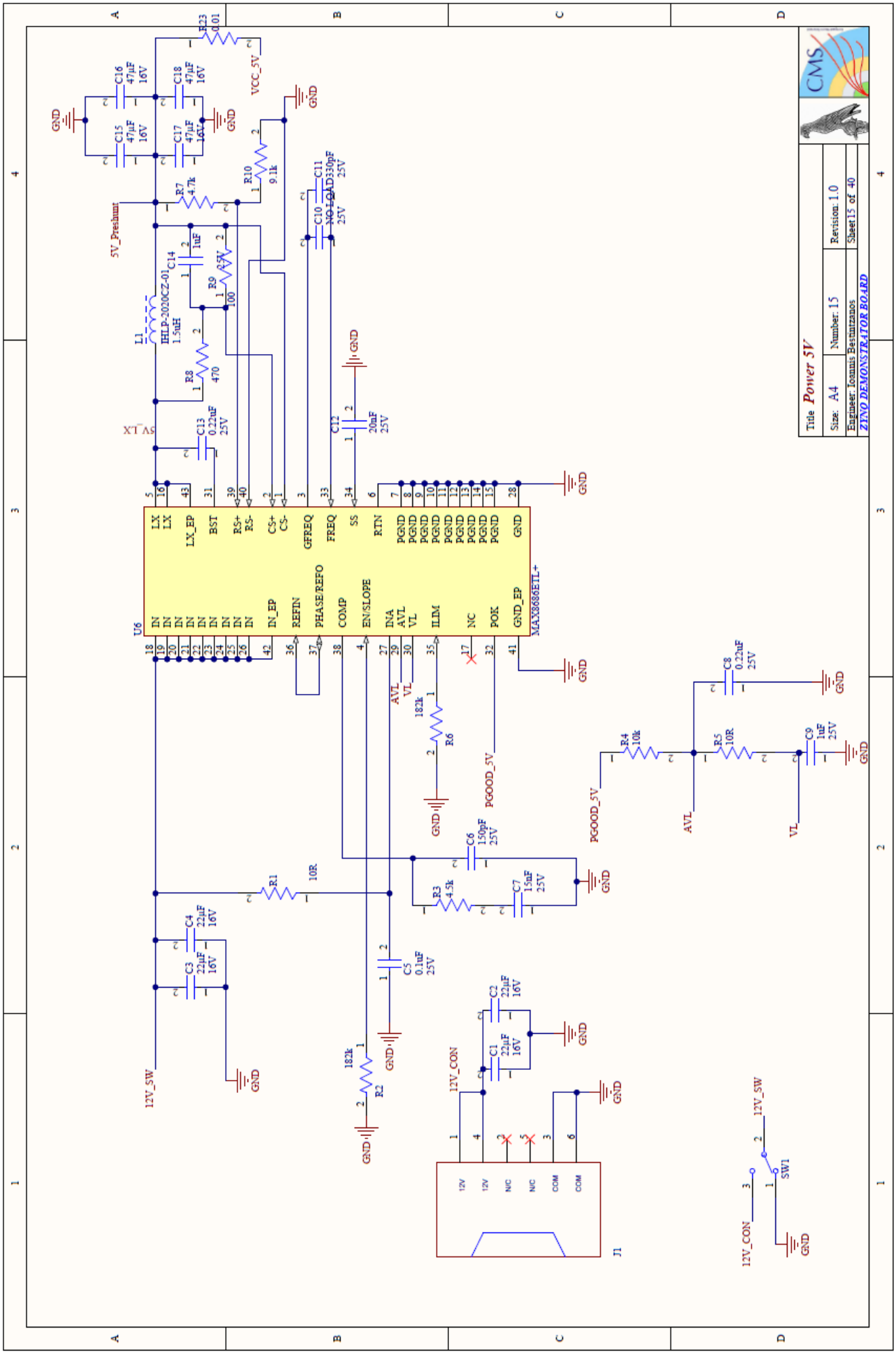

Title FPGA ground
Size: A4
Number: 10
Revision: 1.0
Engineer: Ioannis Bastiranes
Sheet 10 of 40
ZYNQ DEMONSTRATOR BOARD
U1J
XC7Z015-1CLG485C
XADC-AGND
GNDADC_0
RSVDGND
P2
MHDR1X2
R224
0pΩ
NC
GND

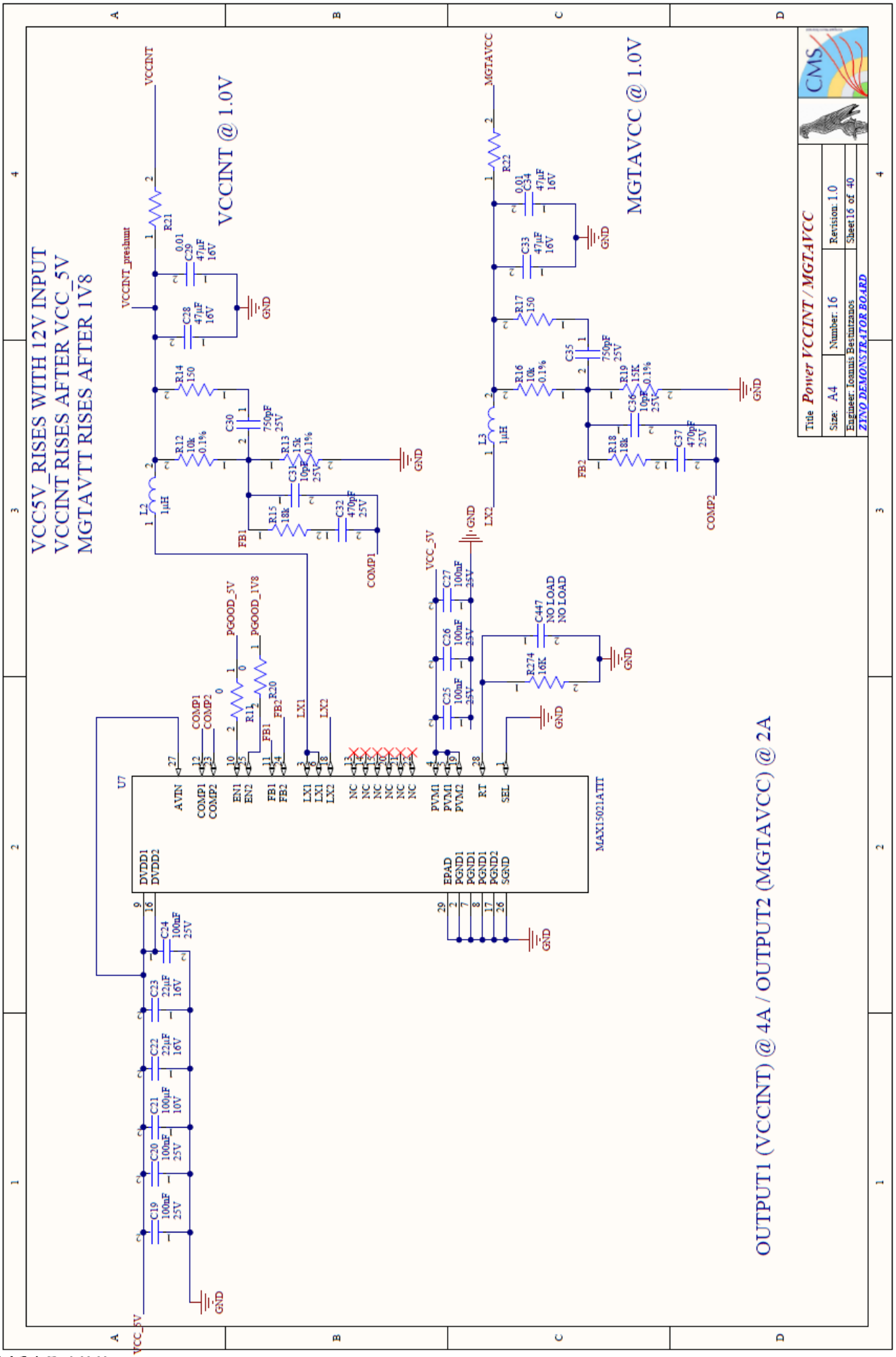

DDR3_ADDR
DDR3_BYTE_1
DDR3_BYTE_2
DDR3_BYTE_3
DDR3_BYTE_4
U2A
U3A
U4A
U5A
MT41J256M8HX-125
GND
Title RAM data
Size: A4
Number: 11
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 11 of 40
ZYNQ DEMONSTRATOR BOARD

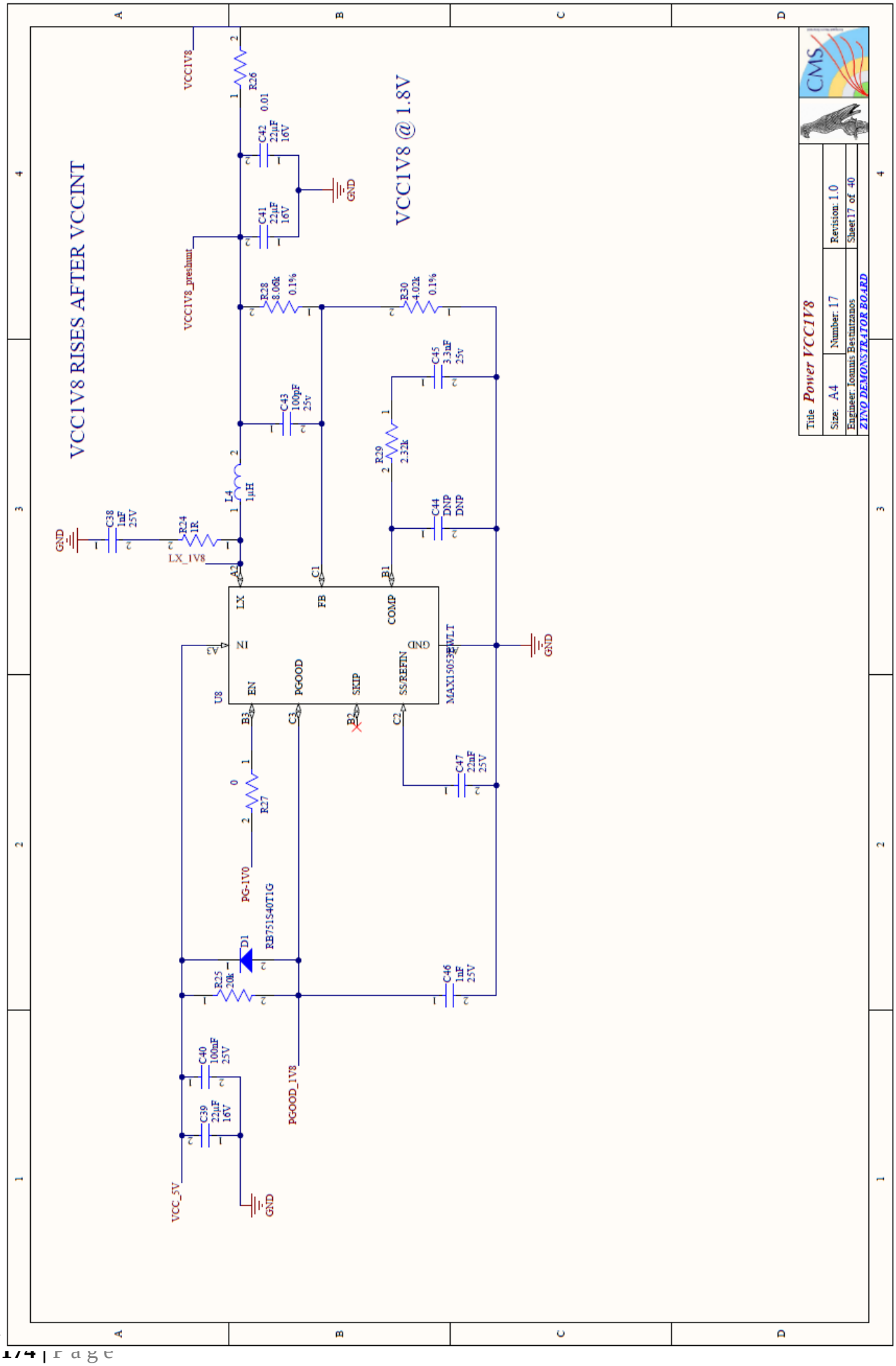

DDR_1V5
R281
DNP
R283
4.7K
GND
R282
40R
1%
R216
R218
R215
R217
R214
R215
R204
R212
R203
R211
R202
R210
R201
R209
R200
R208
R199
R207
R198
R206
R197
R205
R196
DDR0V75
DDR3_A0
DDR3_A1
DDR3_A2
DDR3_A3
DDR3_A4
DDR3_A5
DDR3_A6
DDR3_A7
DDR3_A8
DDR3_A9
DDR3_A10
DDR3_A11
DDR3_A12
DDR3_A13
DDR3_A14
DDR3_CS
DDR3_RAS
DDR3_CAS
DDR3_WE
DDR3_ODT
DDR3_BA0
DDR3_BA1
DDR3_BA2
DDR3_CKE
DDR3_RESET
DDR3_CLK_P
DDR3_CLK_N
R219
80R
Title R4M termination
Size: A4
Number: 12
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 12 of 40
ZINQ DEMONSTRATOR BOARD

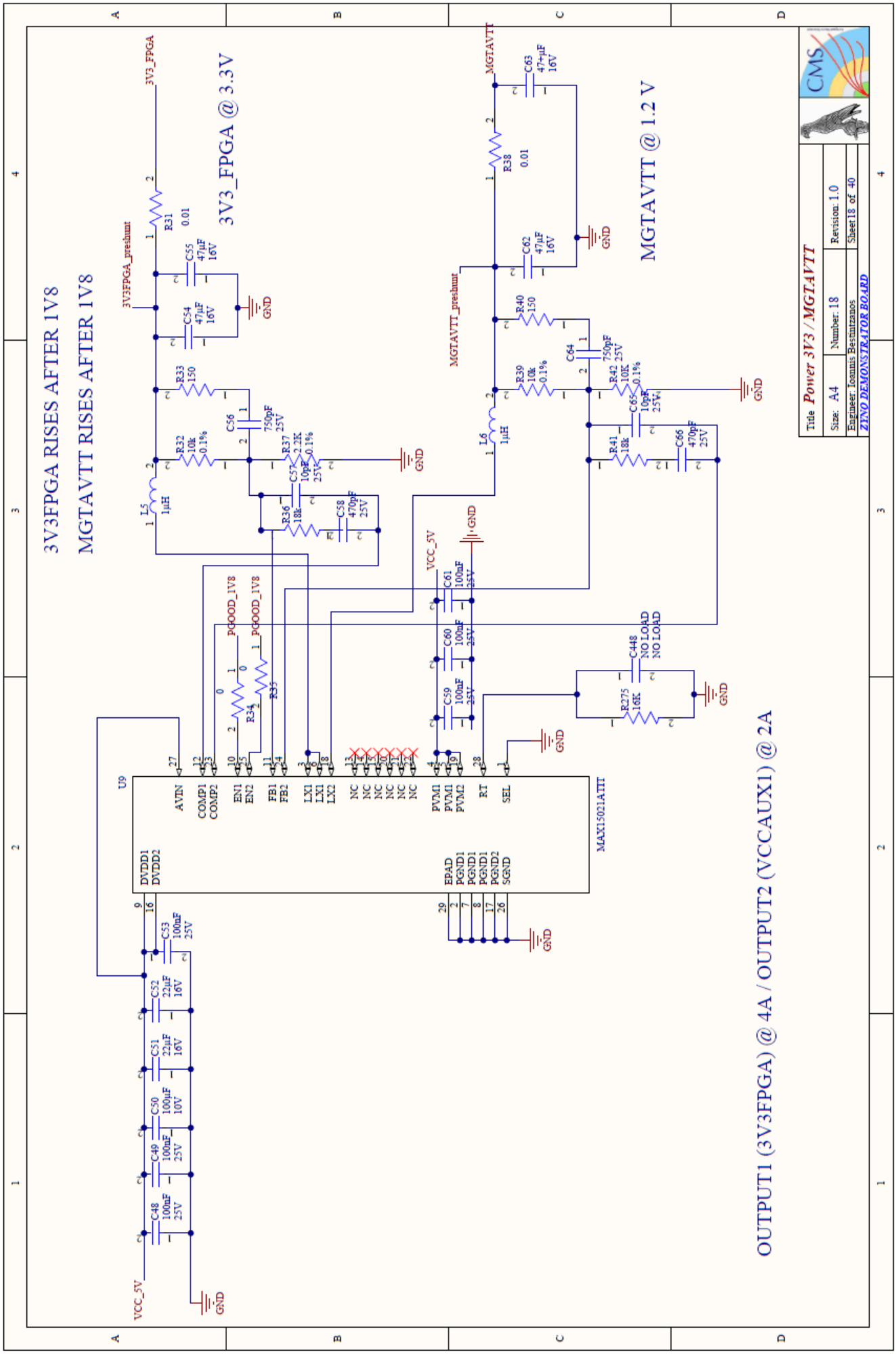

Title RAM power
Size: A4
Number: 13
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 13 of 40
ZINQ DEMONSTRATOR BOARD
U2B
U2C
U3B
U3C
U4B
U4C
U5B
U5C
MT41J256M8HX-125
DDR_1V5
DDVREF
GND
C154
C155
C207
C208
C209
C210
C211
C212
10uF

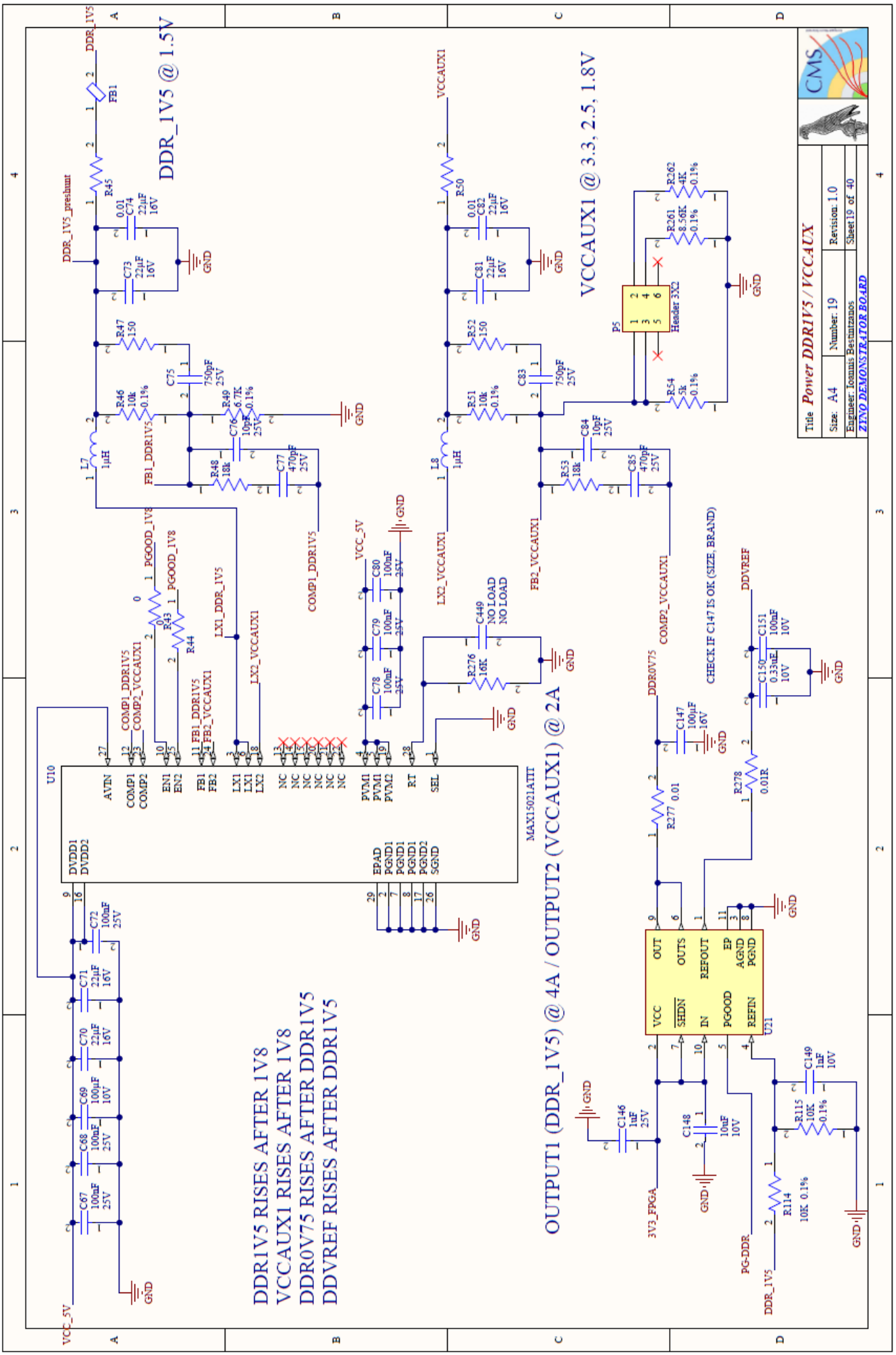

DDR_1V5
C169 47nF
C170 47nF
C171 47nF
C172 47nF
C173 47nF
C174 47nF
C175 47nF
C176 47nF
C177 47nF
C178 47nF
C179 47nF
C180 47nF
C181 47nF
C182 47nF
C183 47nF
C184 47nF
C185 470nF
C186 470nF
C187 4.7uF
GND
DDR_1V5
C188 47nF
C189 47nF
C190 47nF
C191 47nF
C192 47nF
C193 47nF
C194 47nF
C195 47nF
C196 47nF
C197 47nF
C198 47nF
C199 47nF
C200 47nF
C201 47nF
C202 47nF
C203 47nF
C204 470nF
C205 470nF
C206 4.7uF
GND
DDR_1V5
C213 47nF
C214 47nF
C215 47nF
C216 47nF
C217 47nF
C218 47nF
C219 47nF
C220 47nF
C221 47nF
C222 47nF
C223 47nF
C224 47nF
C225 47nF
C226 47nF
C227 47nF
C228 47nF
C229 470nF
C230 470nF
C231 4.7uF
GND
DDR_1V5
C232 47nF
C233 47nF
C234 47nF
C235 47nF
C236 47nF
C237 47nF
C238 47nF
C239 47nF
C240 47nF
C241 47nF
C242 47nF
C243 47nF
C244 47nF
C245 47nF
C246 47nF
C247 47nF
C248 470nF
C249 470nF
C250 4.7uF
GND
DDR0V75
DDR_1V5
C168 10nF
C156 10nF
C157 10nF
C158 100nF
C159 100nF
C160 100nF
C161 100nF
C162 100nF
C163 100nF
C164 1uF
C165 1uF
C166 2.2uF
C167 2.2uF
Title RAM decoupling
Size: A4
Number: 14
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 14 of 40
ZYNQ DEMONSTRATOR BOARD
CMS

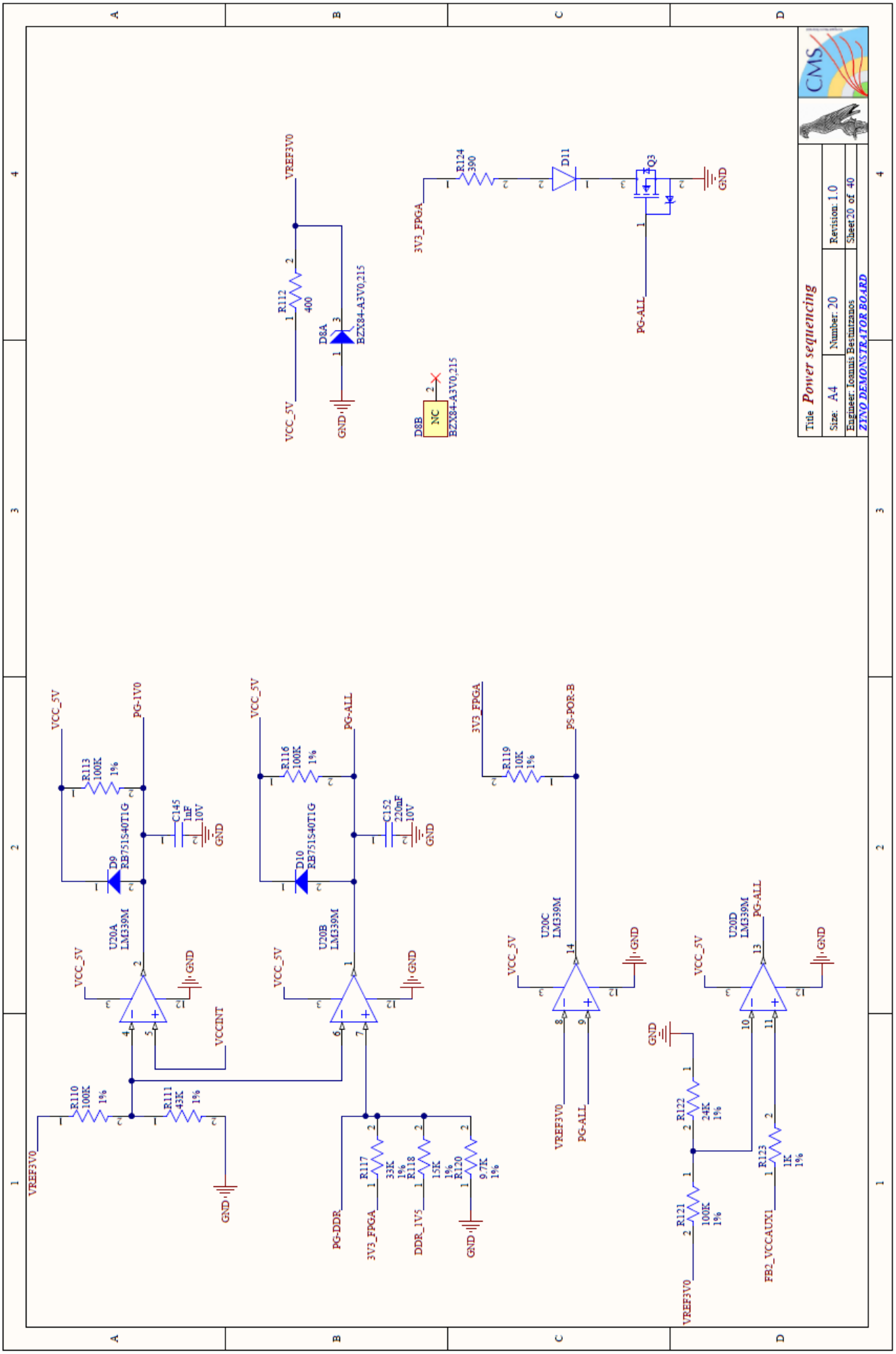

Title Power 5V
Size: A4
Number: 15
Revision: 1.0
Sheet 15 of 40
Engineer: Ioannis Bestnizanos
ZYNQ DEMONSTRATOR BOARD
MAX8686ETL+
U6
J1
SW1
L1
IHLP-2020CZ-01
1.5uH
5V_Presburst
VCC_5V
12V_SW
12V_CON
PGOOD_5V
AVL
VL
GND

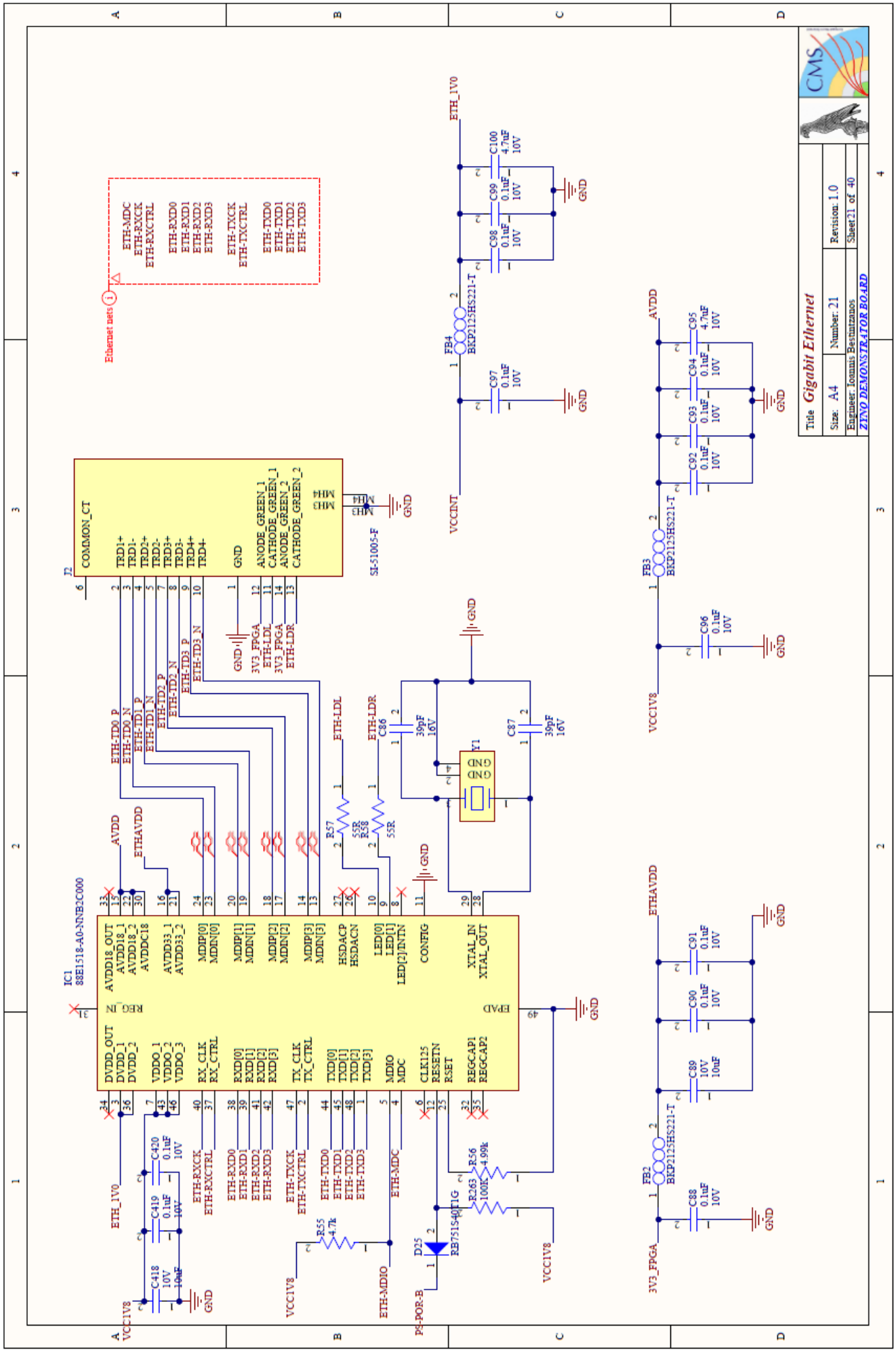

VCCINT @ 1.0V
MGTAVCC @ 1.0V
VCC5V_RISES WITH 12V INPUT
VCCINT_RISES AFTER VCC_5V
MGTAVTT RISES AFTER 1V8
OUTPUT1 (VCCINT) @ 4A / OUTPUT2 (MGTAVCC) @ 2A
MAX15021ATIT
Power VCCINT / MGTAVCC
Size: A4
Number: 16
Revision: 1.0
Sheet 16 of 40
ZYNQ DEMONSTRATOR BOARD

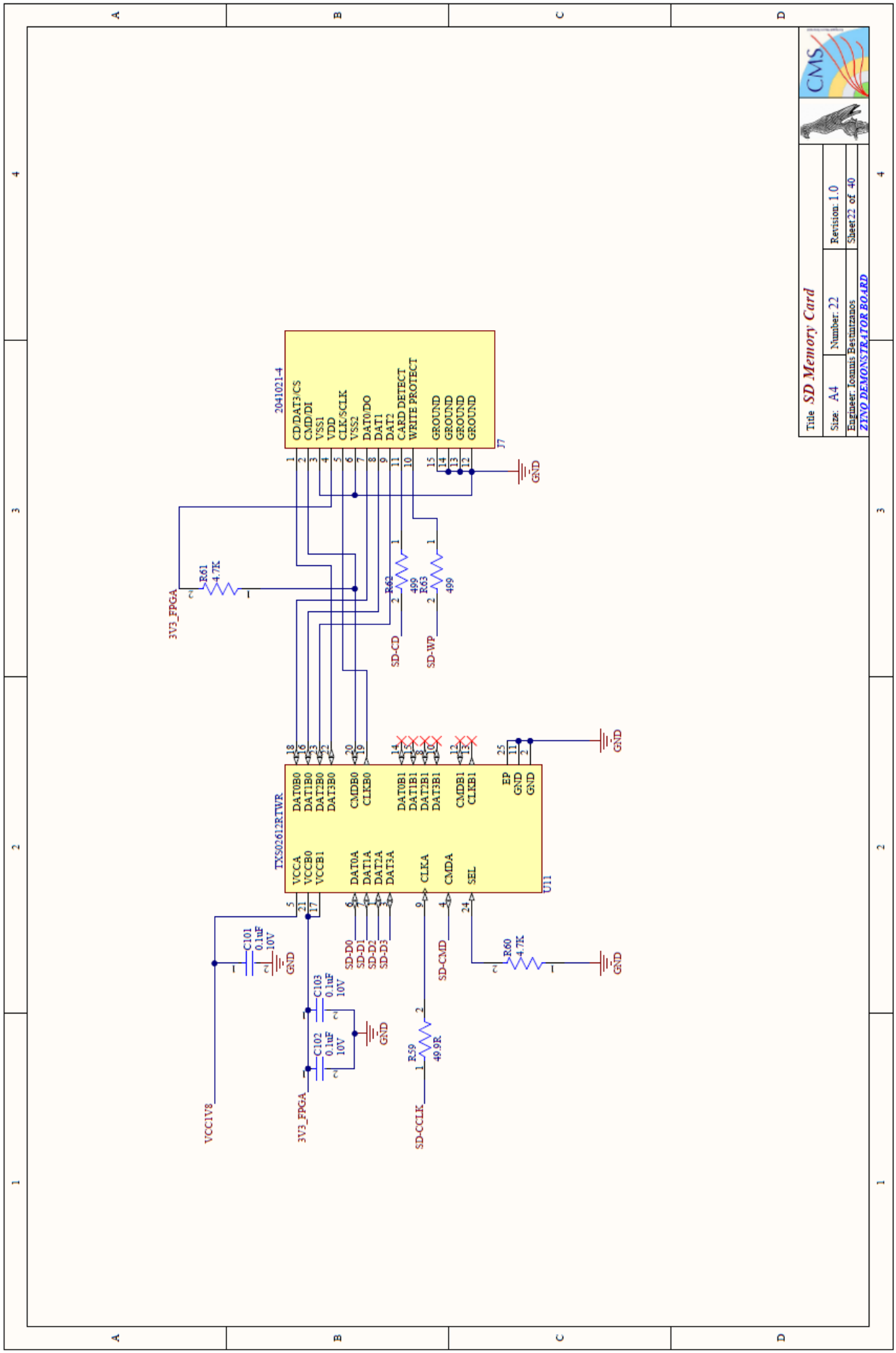

VCC1V8 RISES AFTER VCCINT
VCC1V8 @ 1.8V
VCC1V8
VCC1V8_preshunt
LX_1V8
PG-1V0
PGOOD_1V8
VCC_5V
GND
R26 0.01
C42 22µF 16V
C41 22µF 16V
R28 8.06k 0.1%
R30 4.02k 0.1%
C45 3.3nF 25v
C43 100pF 25v
R29 2.32k
C44 DNP DNP
L4 1µH
C38 1nF 25V
R24 1R
U8
LX
FB
COMP
IN
GND
EN
PGOOD
SKIP
SS/REFIN
MAX15053BWLT
C47 22nF 25V
R27 0
D1 RB751S40T1G
R25 20k
C46 1nF 25V
C40 100nF 25V
C39 22µF 16V
Title Power VCC1V8
Size: A4
Number: 17
Revision: 1.0
Engineer: Ioannis Bastimtzanos
Sheet 17 of 40
ZINQ DEMONSTRATOR BOARD


3V3FPGA RISES AFTER 1V8
MGTAVTT RISES AFTER 1V8
3V3_FPGA @ 3.3V
MGTAVTT @ 1.2 V
OUTPUT1 (3V3FPGA) @ 4A / OUTPUT2 (VCCAUX1) @ 2A
MAX15021ATIT
Power 3V3 / MGTAVTT
Number: 18
Revision: 1.0
Sheet 18 of 40
ZYNQ DEMONSTRATOR BOARD

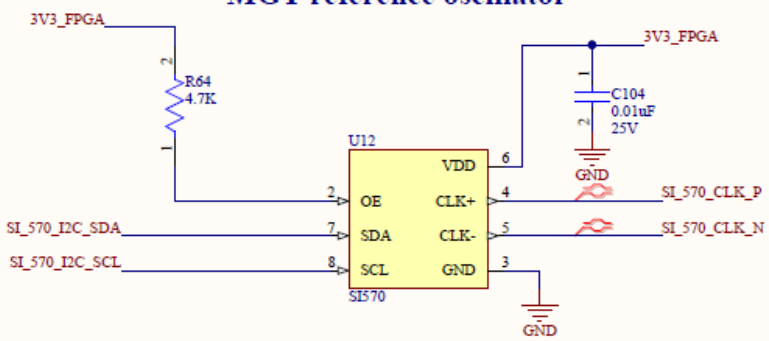

DDR_1V5 @ 1.5V
VCCAUX1 @ 3.3, 2.5, 1.8V
OUTPUT1 (DDR_1V5) @ 4A / OUTPUT2 (VCCAUX1) @ 2A
DDR1V5 RISES AFTER 1V8
VCCAUX1 RISES AFTER 1V8
DDR0V75 RISES AFTER DDR1V5
DDVREF RISES AFTER DDR1V5
CHECK IF C147 IS OK (SIZE, BRAND)
Title Power DDR1V5 / VCCAUX
Size: A4
Number: 19
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 19 of 40
ZINO DEMONSTRATOR BOARD
MAX15021ATIT
Header 3X2


Title Power sequencing
Size: A4
Number: 20
Revision: 1.0
Engineer: Ioannis Besmitzanos
Sheet 20 of 40
ZYNQ DEMONSTRATOR BOARD
VREF3V0
R112 400
D8A BZX84-A3V0,215
VCC_5V
GND
D8B NC BZX84-A3V0,215
3V3_FPGA
R124 390
D11
Q3
PG-ALL
R113 100K 1%
C145 1nF 10V
D9 RB751S40T1G
PG-1V0
U20A LM339M
R116 100K 1%
C152 220nF 10V
D10 RB751S40T1G
U20B LM339M
R119 10K 1%
PS-POR-B
U20C LM339M
U20D LM339M
R110 100K 1%
R111 43K 1%
VCCINT
R117 33K 1%
R118 15K 1%
R120 9.7K 1%
PG-DDR
DDR_1V5
R122 24K 1%
R121 100K 1%
R123 1K 1%
FB2_VCCAUX1

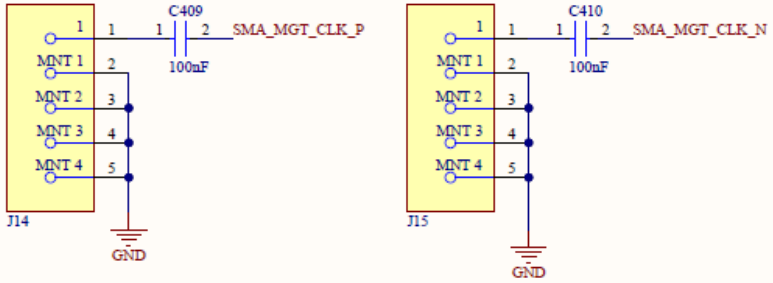

Title Gigabit Ethernet
Size: A4
Number: 21
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 21 of 40
ZTYO DEMONSTRATOR BOARD
Ethernet nets
ETH-MDC
ETH-RXCK
ETH-RXCTRL
ETH-RXD0
ETH-RXD1
ETH-RXD2
ETH-RXD3
ETH-TXCK
ETH-TXCTRL
ETH-TXD0
ETH-TXD1
ETH-TXD2
ETH-TXD3
IC1
88E1518-A0-NNB2C000
J2
COMMON_CT
SI-51005-F
FB2
FB3
FB4
BKP2125HS221-T
Y1
D25
RB751S40T1G
ETH_1V0
VCC1V8
VCCINT
ETHAVDD
AVDD
3V3_FPGA
GND


Title SD Memory Card
Size: A4
Number: 22
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 22 of 40
ZYNQ DEMONSTRATOR BOARD
2041021-4
CD/DAT3/CS
CMD/DI
VSS1
VDD
CLK/SCLK
VSS2
DAT0/DO
DAT1
DAT2
CARD DETECT
WRITE PROTECT
GROUND
J7
TXS02612RTWR
U11
R61 4.7K
R62 499
R63 499
R60 4.7K
R59 49.9R
C101 0.1uF 10V
C102 0.1uF 10V
C103 0.1uF 10V
3V3_FPGA
VCC1V8
GND
SD-CD
SD-WP
SD-D0
SD-D1
SD-D2
SD-D3
SD-CMD
SD-CCLK

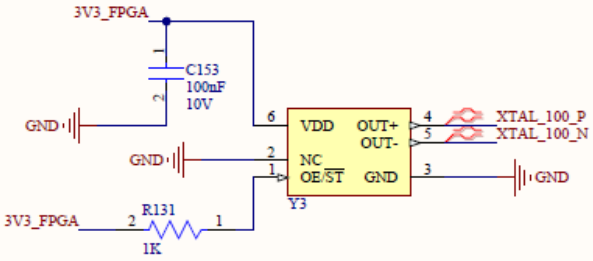
MGT reference oscillator



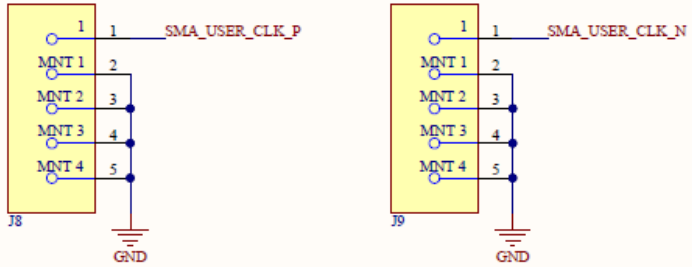
MGT SMA reference clock input



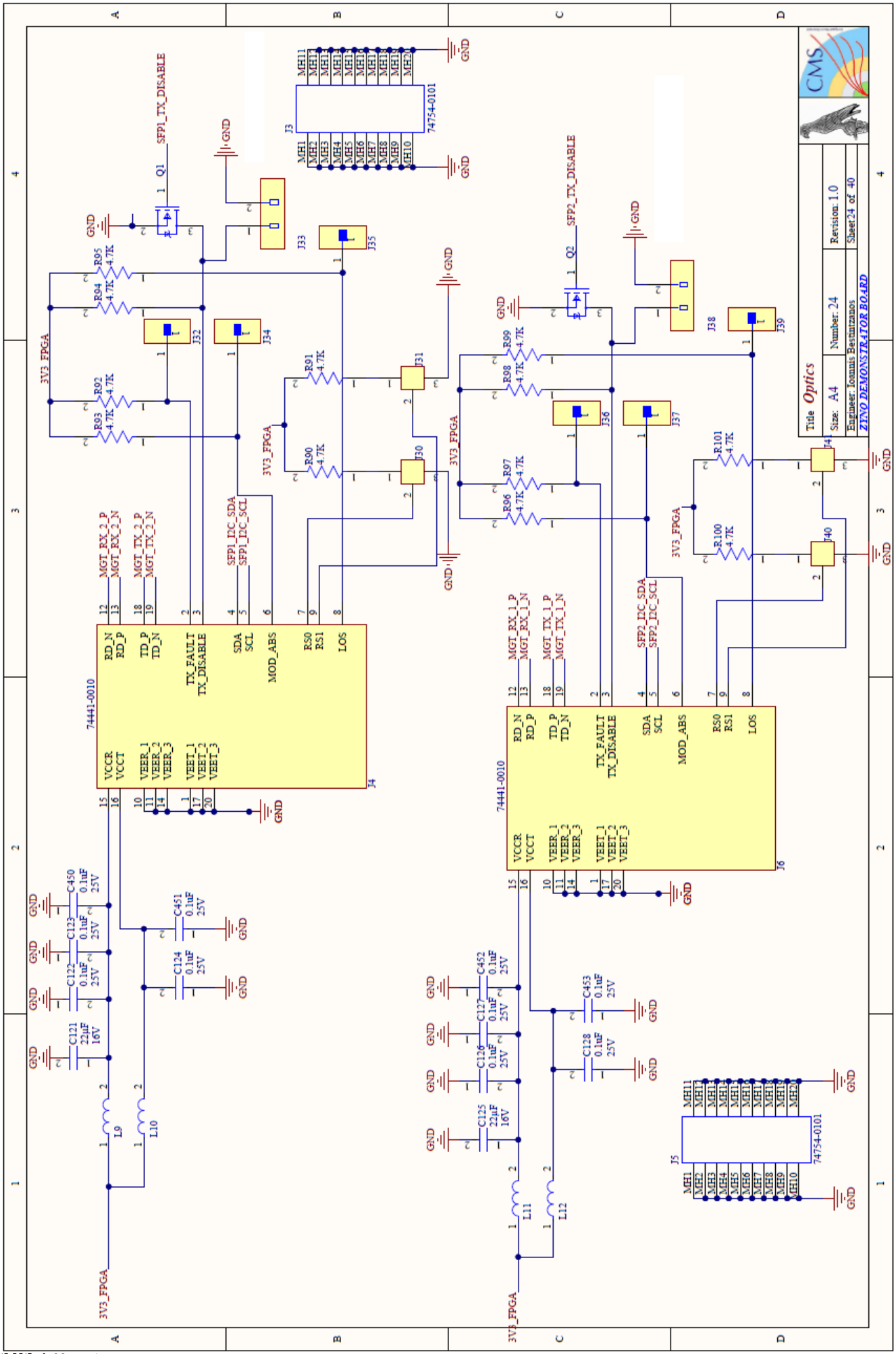
PL oscillator

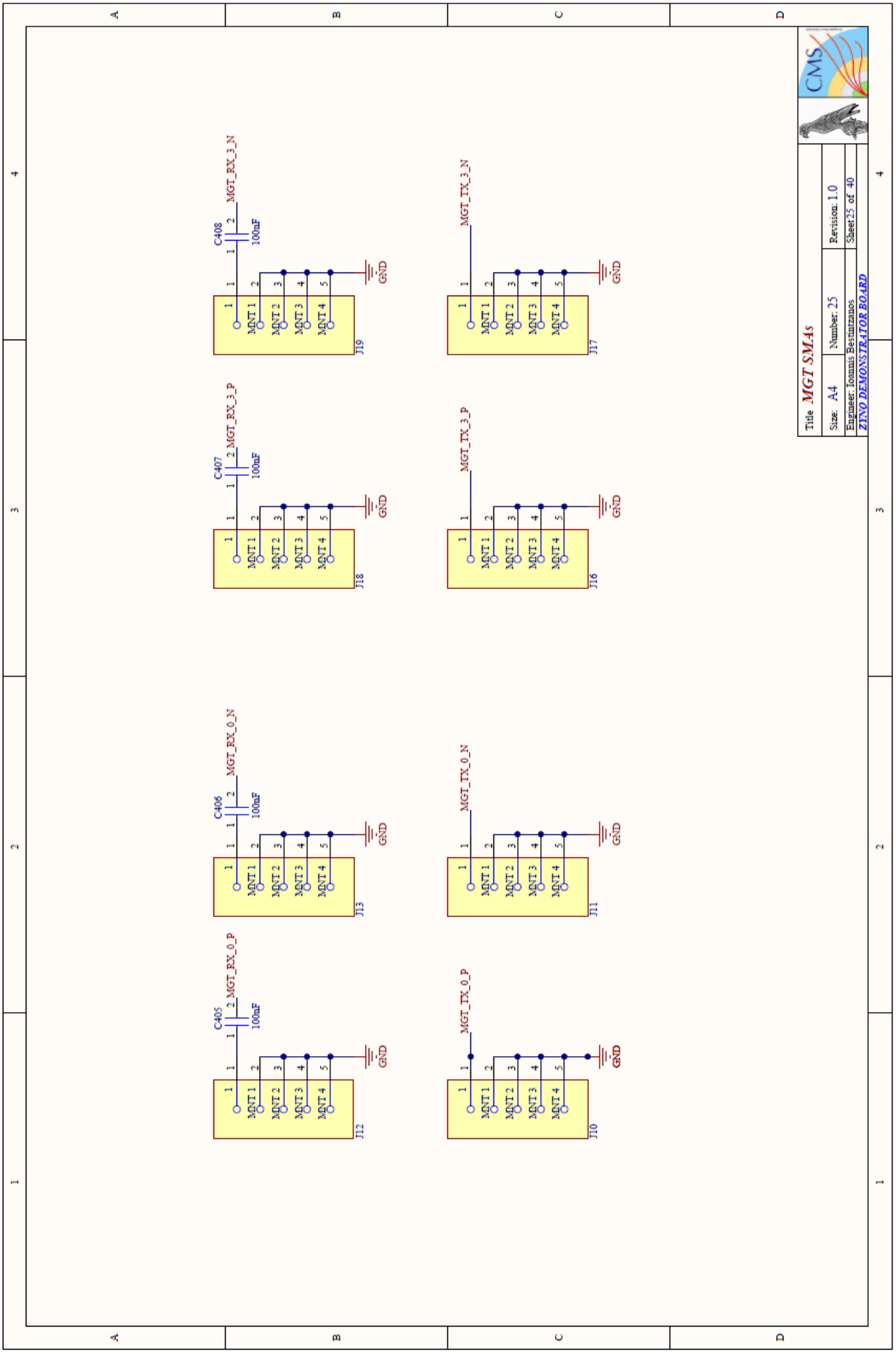


User SMA global clock input

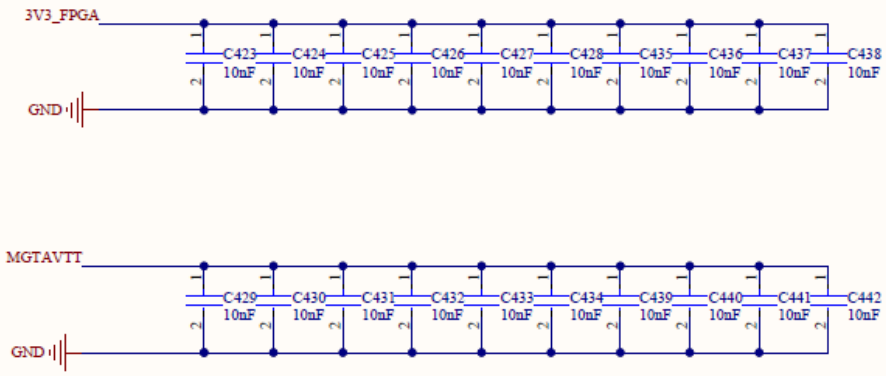


Title *Clocks*		
Size: A4	Number: 23	Revision: 1.0
Engineer: Ioannis Bestintzanos		Sheet 23 of 40
ZYNQ DEMONSTRATOR BOARD		

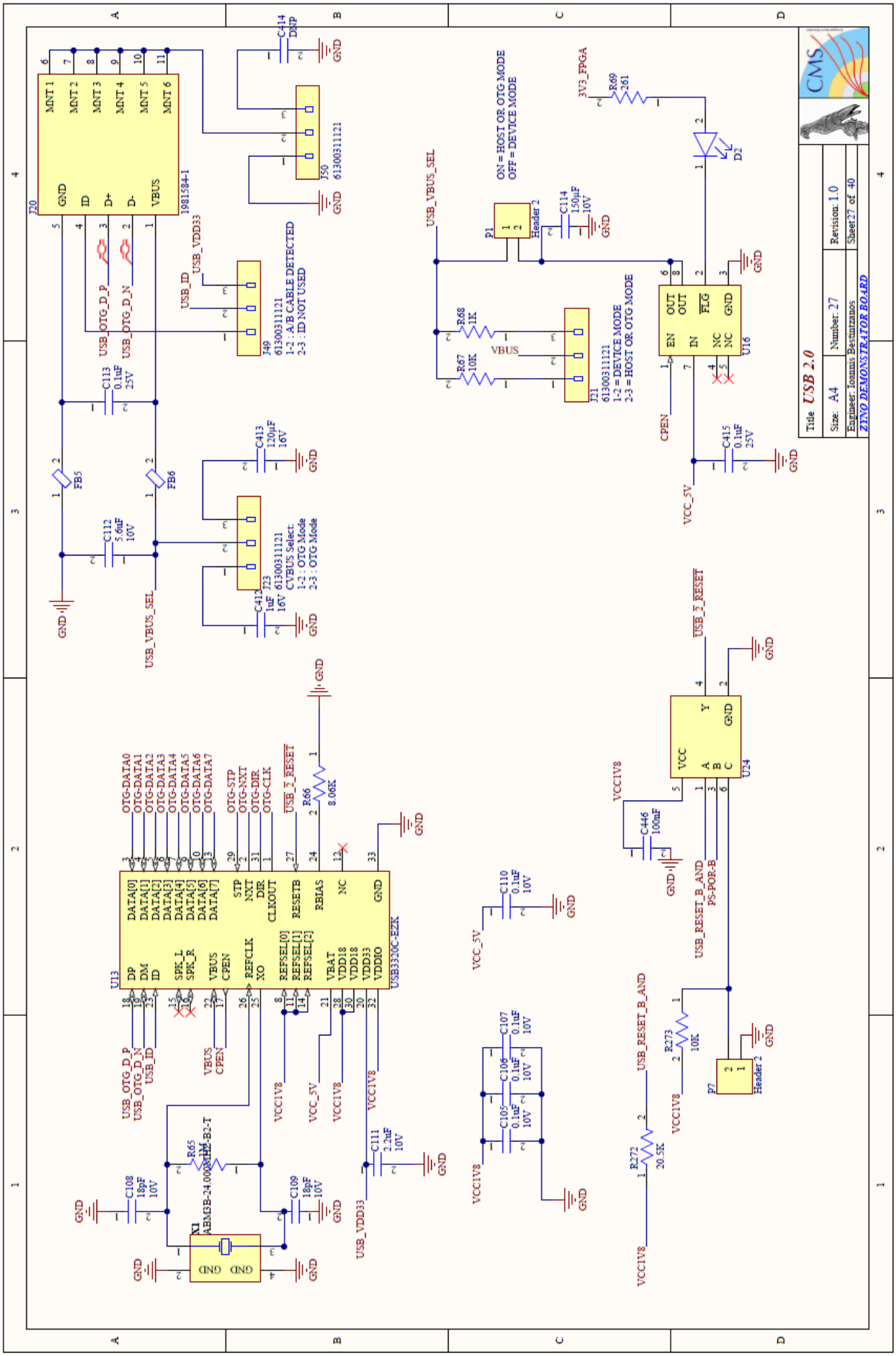

Title Optics
Size: A4
Number: 24
Revision: 1.0
Sheet 24 of 40
Engineer: Ioannis Bestmizanos
ZTNQ DEMONSTRATOR BOARD
SFP1_TX_DISABLE
SFP2_TX_DISABLE
3V3_FPGA
GND
74441-0010
74754-0101
MGT_RX_2_P
MGT_RX_2_N
MGT_TX_2_P
MGT_TX_2_N
SFP1_I2C_SDA
SFP1_I2C_SCL
MGT_RX_1_P
MGT_RX_1_N
MGT_TX_1_P
MGT_TX_1_N
SFP2_I2C_SDA
SFP2_I2C_SCL

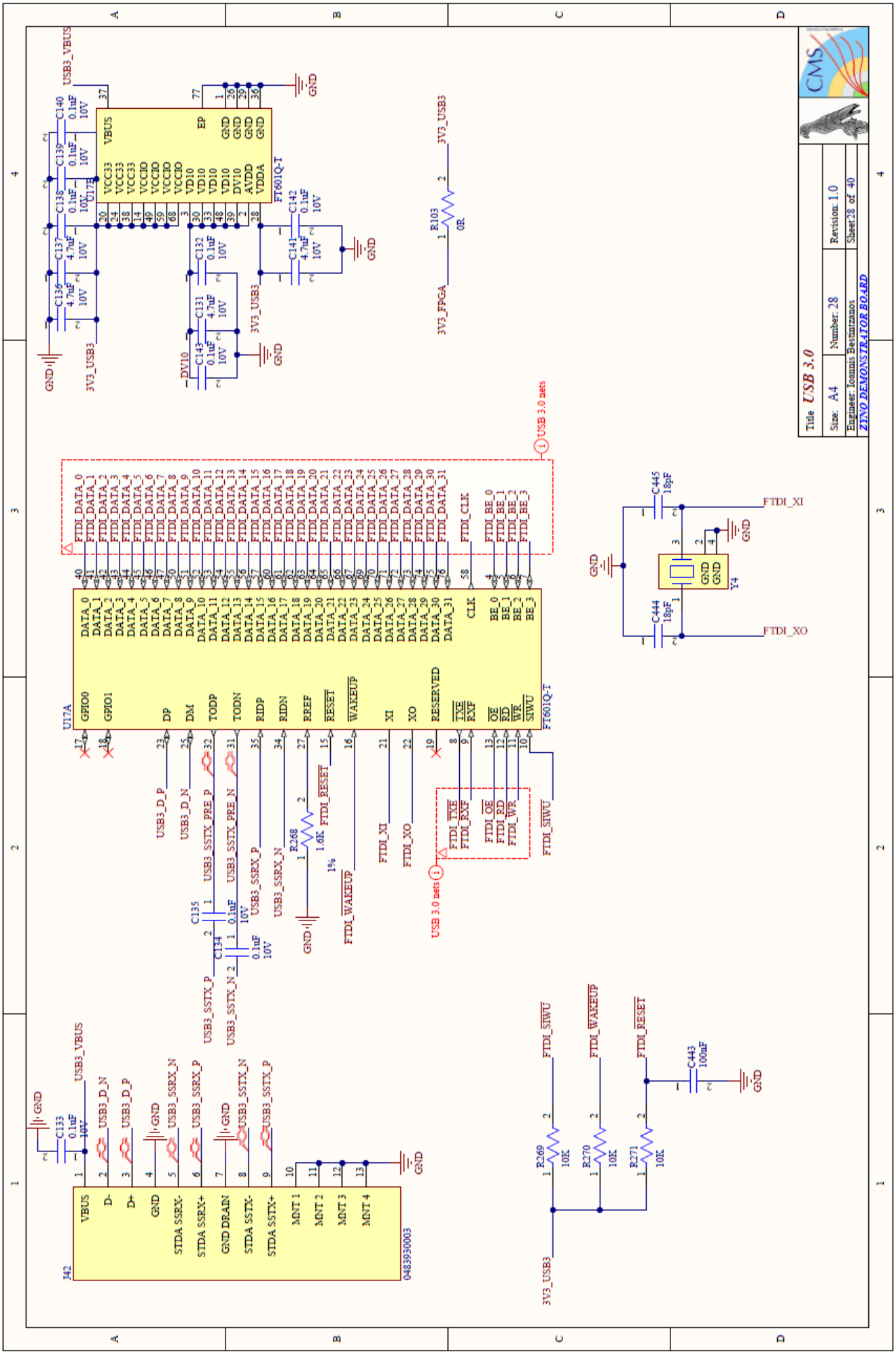

MGT SMAs
Size: A4
Number: 25
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 25 of 40
ZYNQ DEMONSTRATOR BOARD
C408
100nF
MGT_RX_3_N
C407
100nF
MGT_RX_3_P
C406
100nF
MGT_RX_0_N
C405
100nF
MGT_RX_0_P
MGT_TX_3_N
MGT_TX_3_P
MGT_TX_0_N
MGT_TX_0_P
GND
MNT 1
MNT 2
MNT 3
MNT 4
J19
J18
J13
J12
J17
J16
J11
J10

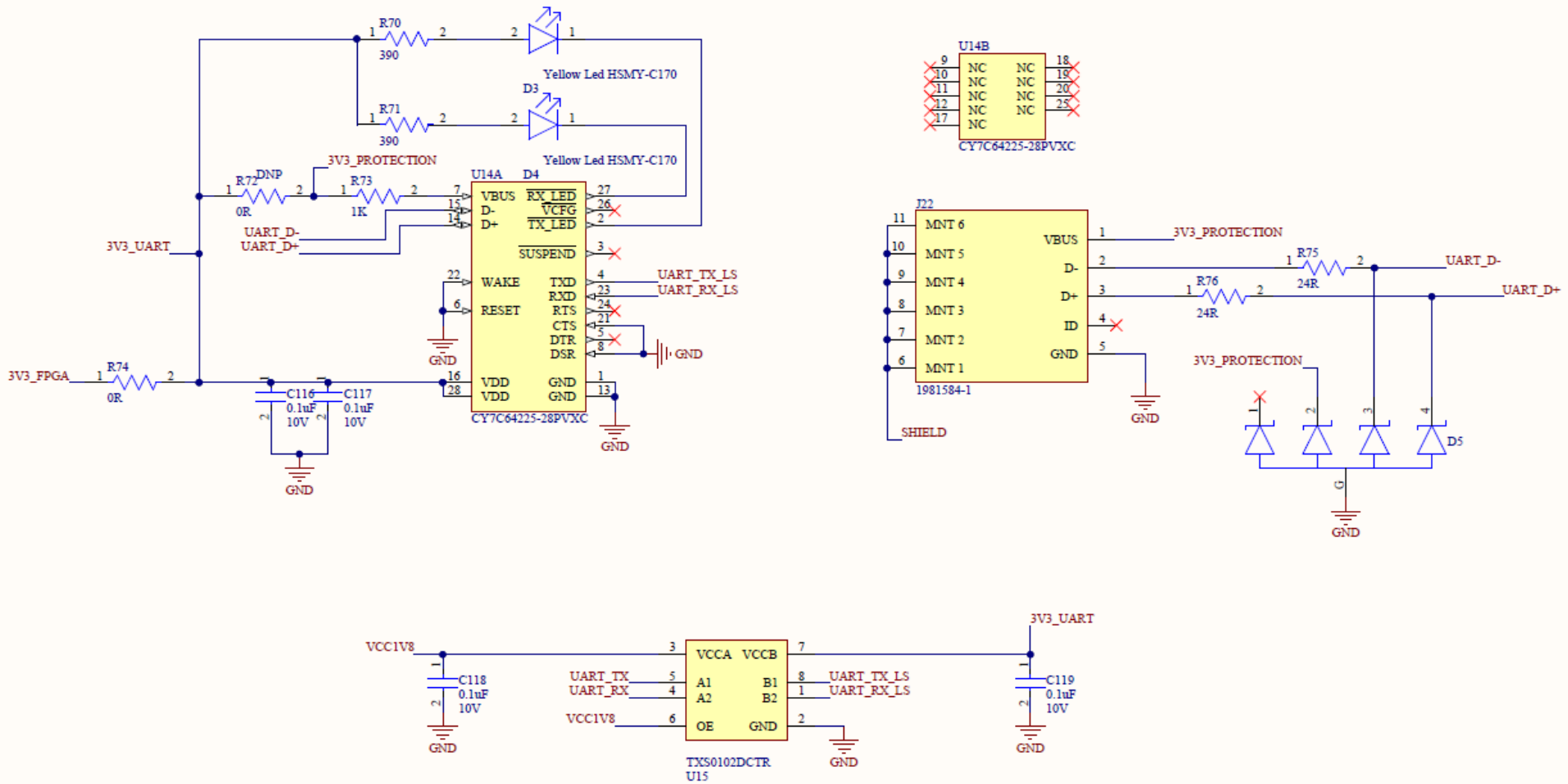
MGT DC reference planes decoupling to ground along the MGT traces



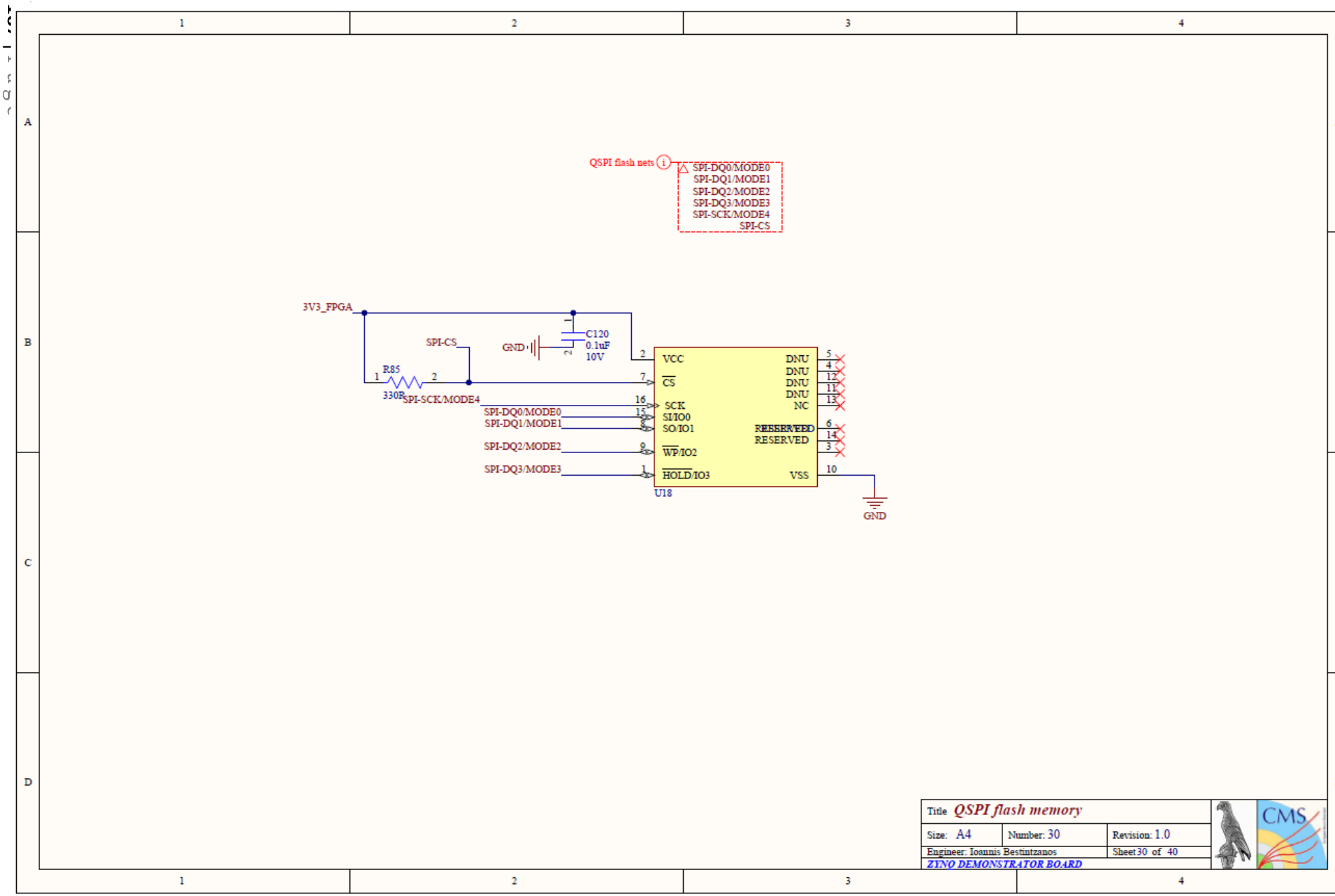
Title *MGT reference plane decoupling*		
Size: A4	Number: 26	Revision: 1.0
Engineer: Ioannis Bestintzanos		Sheet 26 of 40
ZYNQ DEMONSTRATOR BOARD		

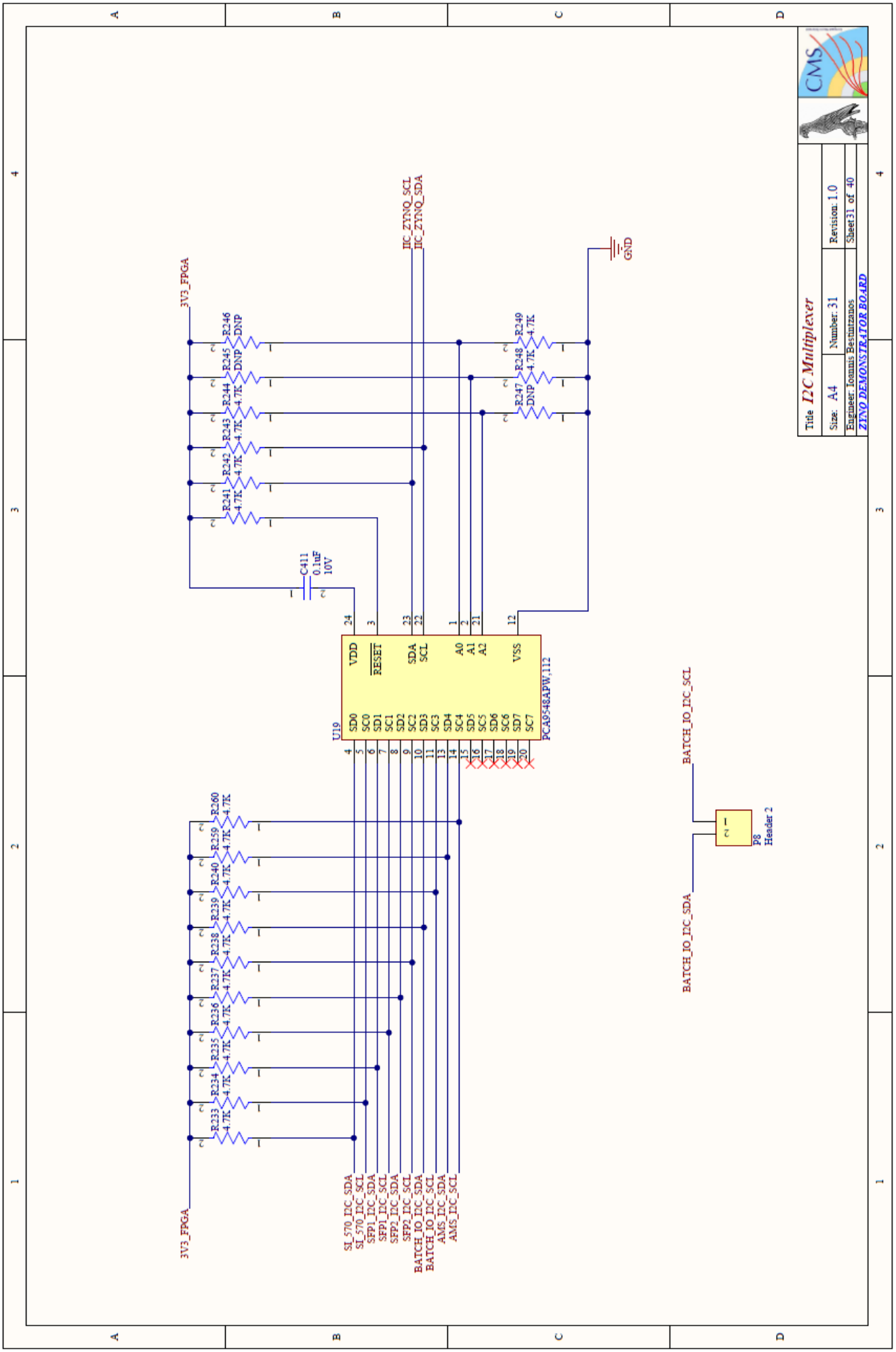

Title USB 2.0
Size: A4
Number: 27
Revision: 1.0
Engineer: Ioannis Bestnitzanos
Sheet 27 of 40
ZYNQ DEMONSTRATOR BOARD
USB_VBUS_SEL
ON = HOST OR OTG MODE
OFF = DEVICE MODE
3V3_FPGA
USB_VDD33
USB_ID
USB_OTG_D_P
USB_OTG_D_N
VBUS
CPEN
USB_2_RESET
USB_RESET_B_AND
PS-POR-B
VCC_5V
VCC1V8
GND
J20
1981384-1
J40
61300311121
1-2 : A/B CABLE DETECTED
2-3 : ID NOT USED
J23
CVBUS Select:
1-2 : OTG Mode
2-3 : OTG Mode
J21
1-2 = DEVICE MODE
2-3 = HOST OR OTG MODE
J50
P1
Header 2
P7
U13
USB3320C-EZK
U16
U24
X1
ABM3B-24.000MHZ-B2-T
D2
FB5
FB6

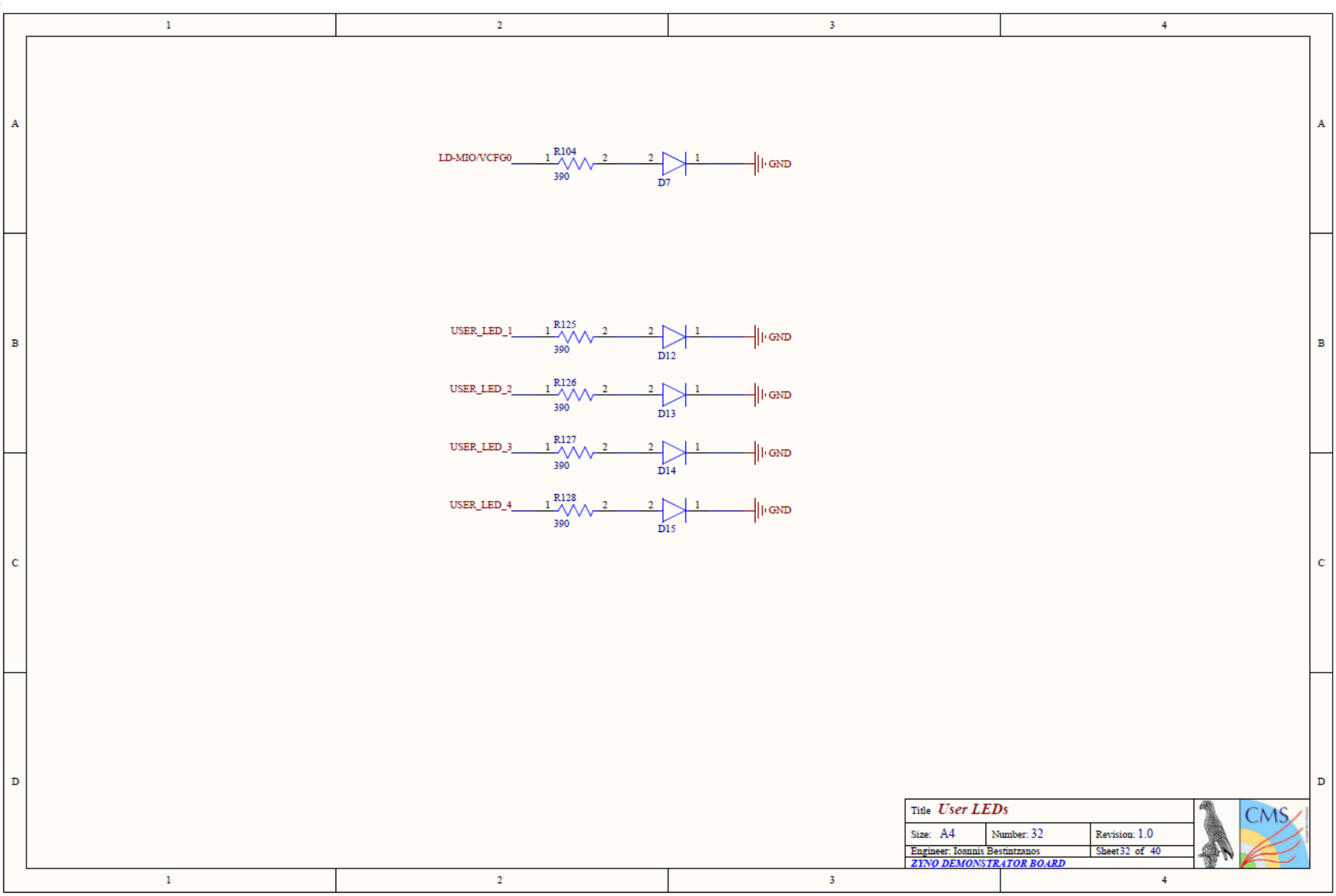

USB 3.0
Size: A4
Number: 28
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 28 of 40
ZINQ DEMONSTRATOR BOARD
FT601Q-T
J42
0483930003
Y4
FTDI_XI
FTDI_XO
3V3_USB3
3V3_FPGA
USB3_VBUS
USB3_D_P
USB3_D_N
USB3_SSTX_P
USB3_SSTX_N
USB3_SSRX_P
USB3_SSRX_N
FTDI_WAKEUP
FTDI_RESET
FTDI_SIWU
FTDI_CLK
FTDI_TXE
FTDI_RXF
FTDI_OE
FTDI_RD
FTDI_WR
FTDI_BE_0
FTDI_BE_1
FTDI_BE_2
FTDI_BE_3
USB 3.0 nets



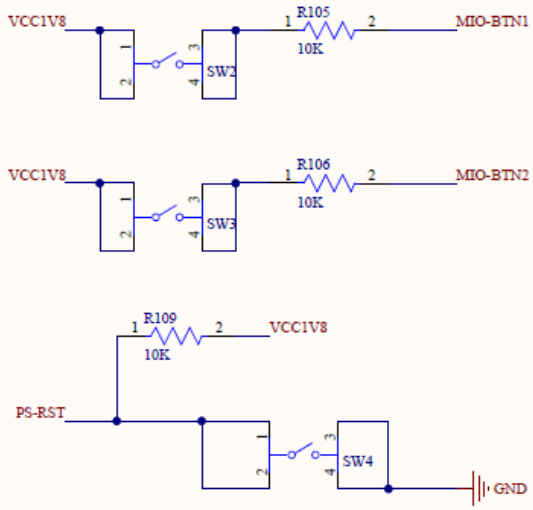
Title *UART*		
Size: A4	Number: 29	Revision: 1.0
Engineer: Ioannis Bestintzanos		Sheet 29 of 40
ZYNQ DEMONSTRATOR BOARD		


QSPI flash nets
SPI-DQ0/MODE0
SPI-DQ1/MODE1
SPI-DQ2/MODE2
SPI-DQ3/MODE3
SPI-SCK/MODE4
SPI-CS
3V3_FPGA
SPI-CS
GND
C120
0.1uF
10V
R85
330R
SPI-SCK/MODE4
SPI-DQ0/MODE0
SPI-DQ1/MODE1
SPI-DQ2/MODE2
SPI-DQ3/MODE3
VCC
CS
SCK
SI/IO0
SO/IO1
WP/IO2
HOLD/IO3
DNU
DNU
DNU
DNU
NC
RESERVED
RESERVED
VSS
U18
GND
Title QSPI flash memory
Size: A4
Number: 30
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 30 of 40
ZYNQ DEMONSTRATOR BOARD
CMS

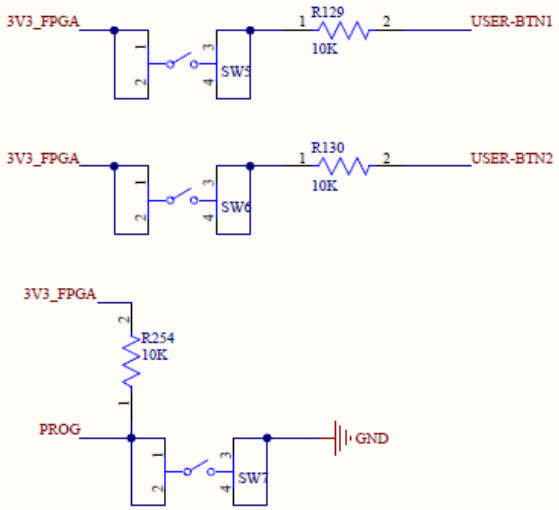

Title I2C Multiplexer
Size: A4
Number: 31
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 31 of 40
ZYNQ DEMONSTRATOR BOARD
3V3_FPGA
IIC_ZYNQ_SCL
IIC_ZYNQ_SDA
GND
R246 DNP
R245 DNP
R244 4.7K
R243 4.7K
R242 4.7K
R241 4.7K
R249 4.7K
R248 4.7K
R247 DNP
C411 0.1uF 10V
U19
PCA9548APW,112
VDD
RESET
SDA
SCL
A0
A1
A2
VSS
SD0
SC0
SD1
SC1
SD2
SC2
SD3
SC3
SD4
SC4
SD5
SC5
SD6
SC6
SD7
SC7
R260 4.7K
R259 4.7K
R240 4.7K
R239 4.7K
R238 4.7K
R237 4.7K
R236 4.7K
R235 4.7K
R234 4.7K
R233 4.7K
SL_570_IIC_SDA
SL_570_IIC_SCL
SFP1_IIC_SDA
SFP1_IIC_SCL
SFP2_IIC_SDA
SFP2_IIC_SCL
BATCH_IO_IIC_SDA
BATCH_IO_IIC_SCL
AMS_IIC_SDA
AMS_IIC_SCL
BATCH_IO_IIC_SCL
BATCH_IO_IIC_SDA
P8
Header 2


1
2
3
4
A
B
C
D
LD-MIO/VCFG0
R104
390
D7
GND
USER_LED_1
R125
390
D12
GND
USER_LED_2
R126
390
D13
GND
USER_LED_3
R127
390
D14
GND
USER_LED_4
R128
390
D15
GND
Title User LEDs
Size: A4
Number: 32
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 32 of 40
ZYNQ DEMONSTRATOR BOARD
CMS

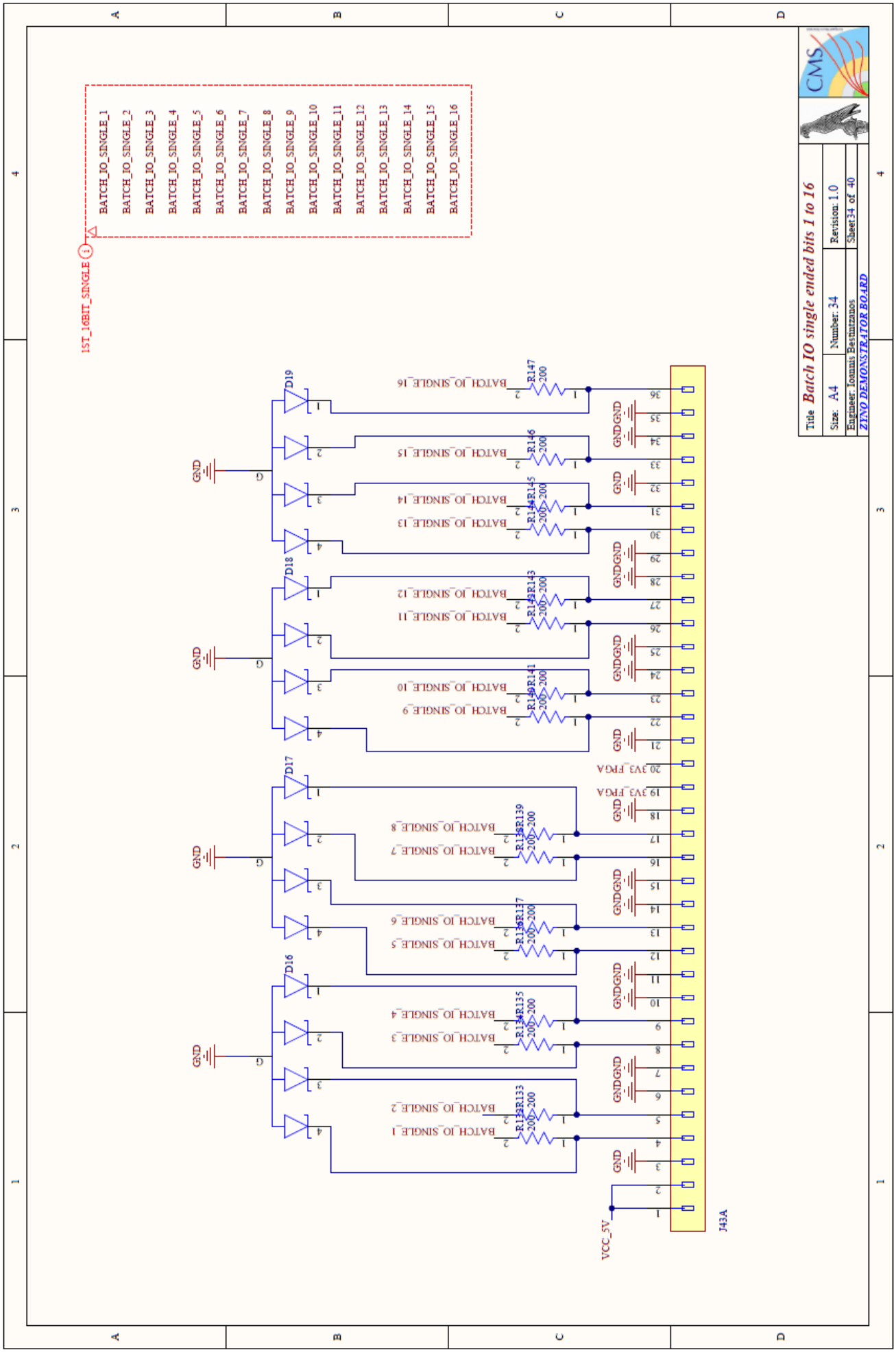
MIO SECTION

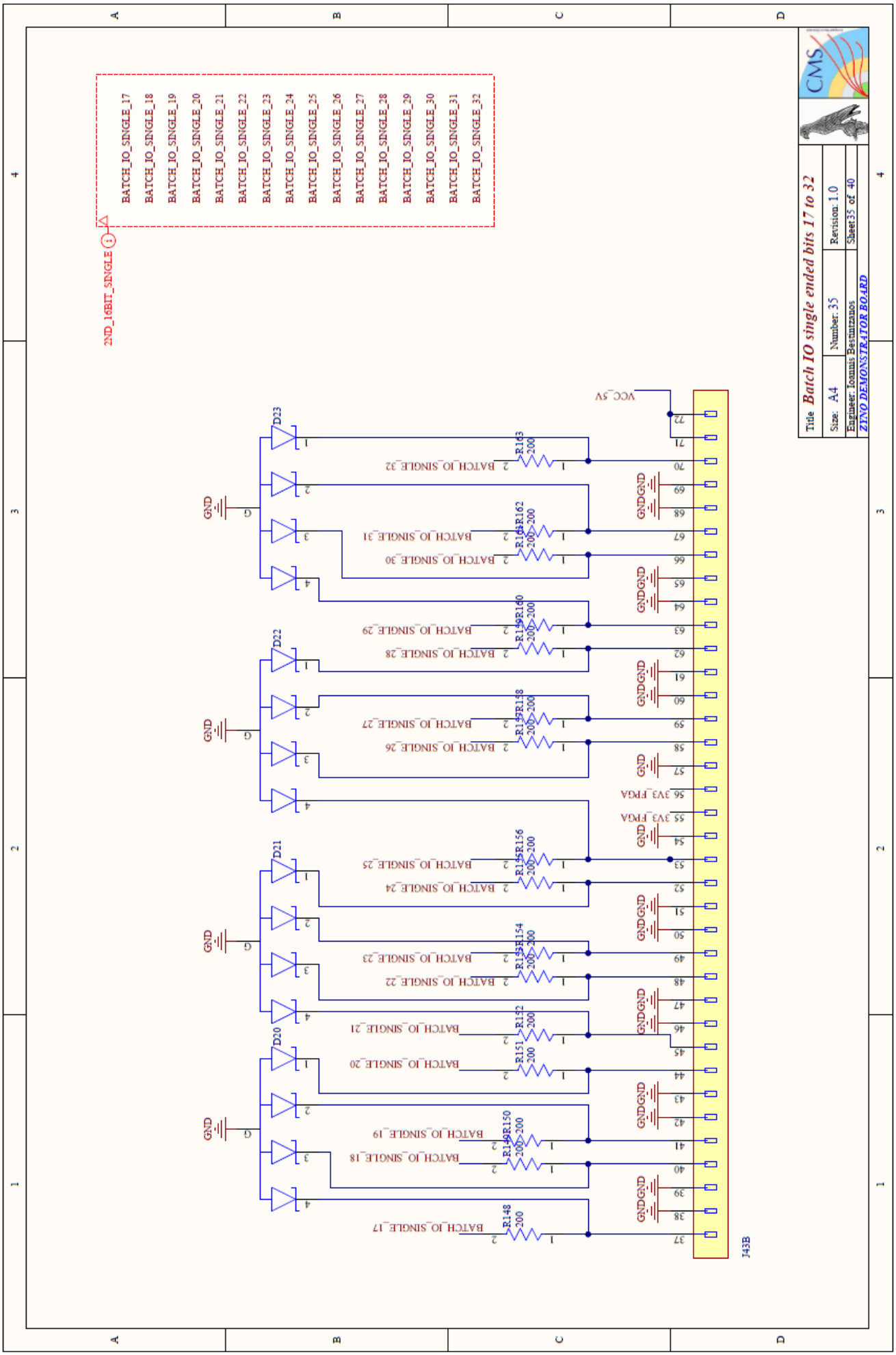


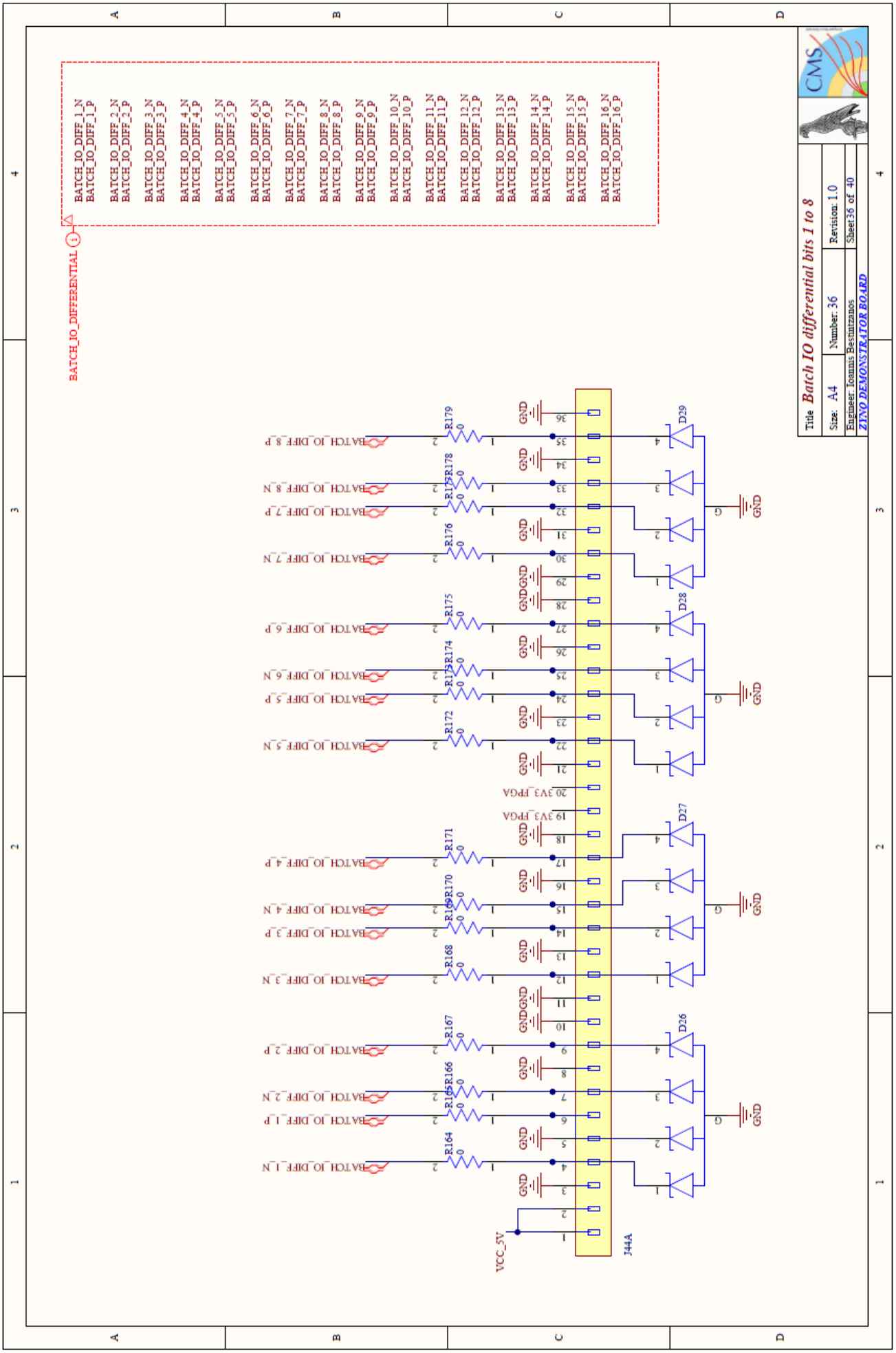
FPGA SECTION

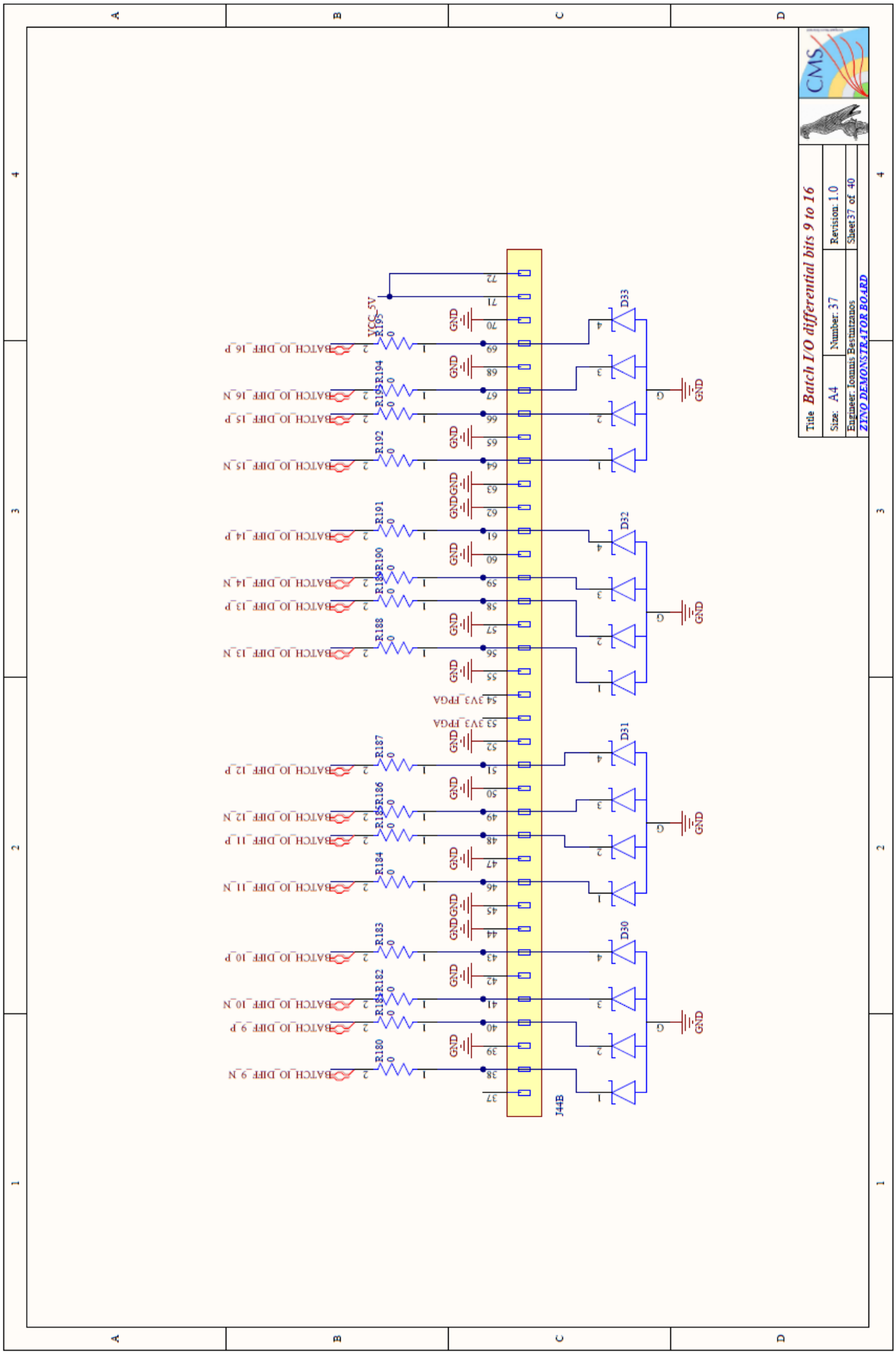


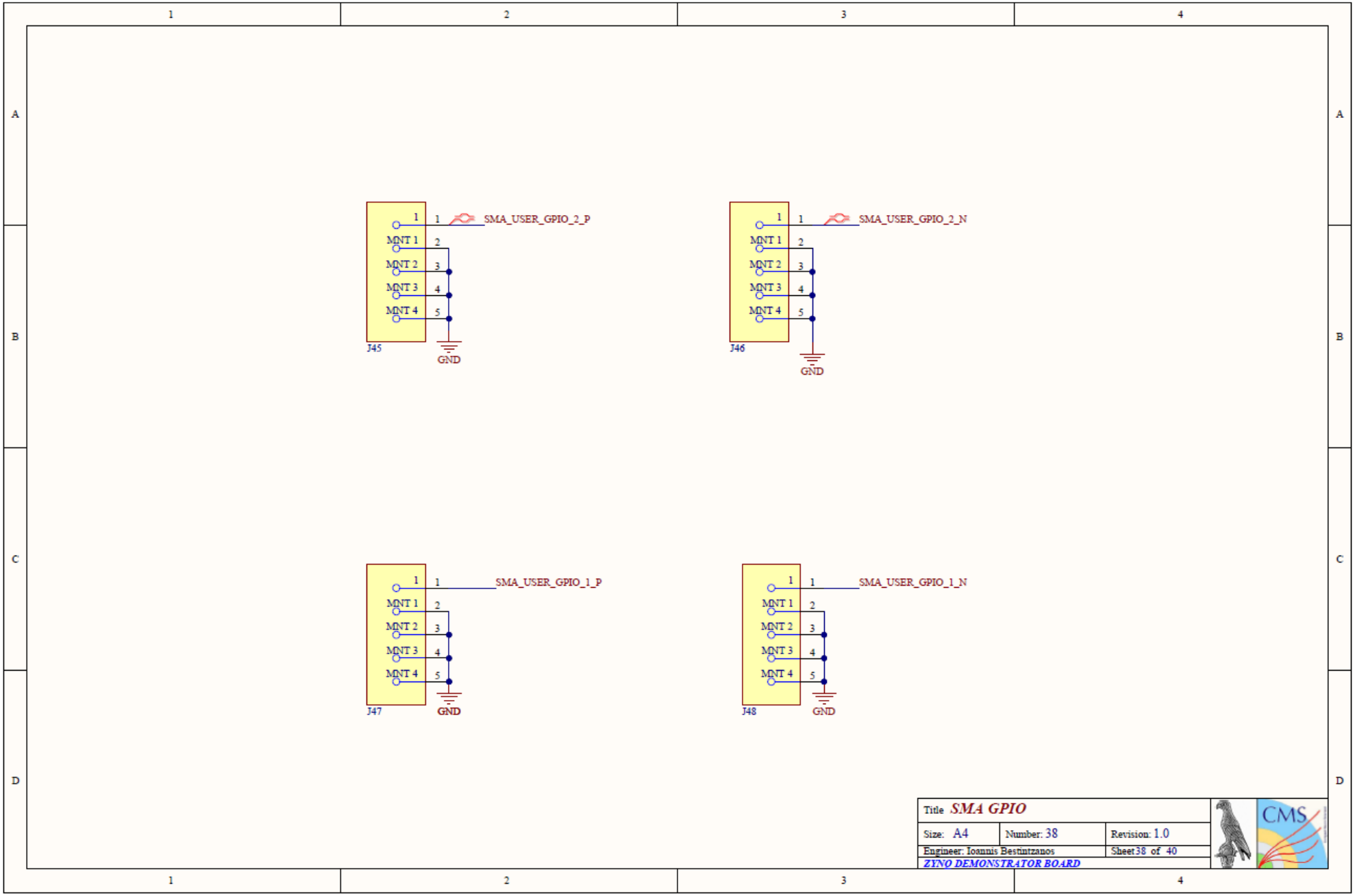
Title *User buttons*		
Size: A4	Number: 33	Revision: 1.0
Engineer: Ioannis Bestintzanos		Sheet 33 of 40
ZYNQ DEMONSTRATOR BOARD		

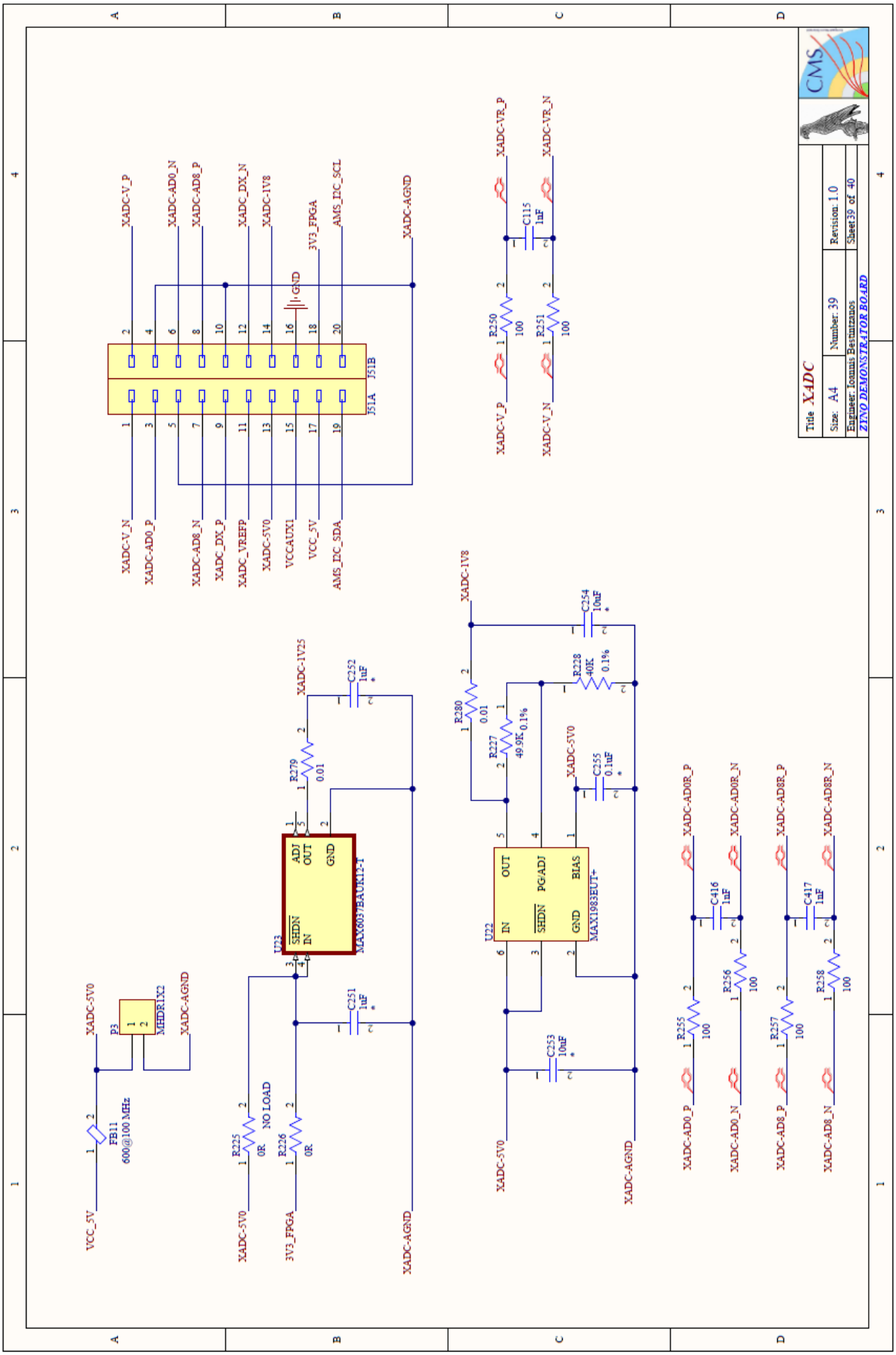

1ST_16BIT_SINGLE
BATCH_IO_SINGLE_1
BATCH_IO_SINGLE_2
BATCH_IO_SINGLE_3
BATCH_IO_SINGLE_4
BATCH_IO_SINGLE_5
BATCH_IO_SINGLE_6
BATCH_IO_SINGLE_7
BATCH_IO_SINGLE_8
BATCH_IO_SINGLE_9
BATCH_IO_SINGLE_10
BATCH_IO_SINGLE_11
BATCH_IO_SINGLE_12
BATCH_IO_SINGLE_13
BATCH_IO_SINGLE_14
BATCH_IO_SINGLE_15
BATCH_IO_SINGLE_16
Title Batch IO single ended bits 1 to 16
Size: A4
Number: 34
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 34 of 40
ZYNQ DEMONSTRATOR BOARD
J43A
VCC_5V
GND
3V3_FPGA
D16
D17
D18
D19
R133
R134
R135
R136
R137
R138
R139
R140
R141
R142
R143
R144
R145
R146
R147
200


2ND_16BIT_SINGLE
BATCH_IO_SINGLE_17
BATCH_IO_SINGLE_18
BATCH_IO_SINGLE_19
BATCH_IO_SINGLE_20
BATCH_IO_SINGLE_21
BATCH_IO_SINGLE_22
BATCH_IO_SINGLE_23
BATCH_IO_SINGLE_24
BATCH_IO_SINGLE_25
BATCH_IO_SINGLE_26
BATCH_IO_SINGLE_27
BATCH_IO_SINGLE_28
BATCH_IO_SINGLE_29
BATCH_IO_SINGLE_30
BATCH_IO_SINGLE_31
BATCH_IO_SINGLE_32
Title Batch IO single ended bits 17 to 32
Size: A4
Number: 35
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 35 of 40
ZYNQ DEMONSTRATOR BOARD
VCC_5V
GND
3V3_FPGA
J43B
D23
D22
D21
D20
R163
R162
R161
R160
R159
R158
R157
R156
R155
R154
R153
R152
R151
R150
R149
R148

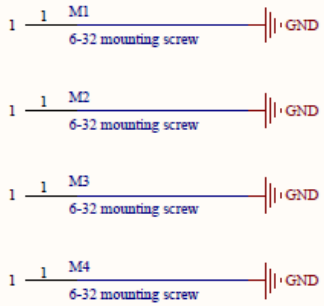

BATCH_IO_DIFFERENTIAL
BATCH_IO_DIFF_1_N
BATCH_IO_DIFF_1_P
BATCH_IO_DIFF_2_N
BATCH_IO_DIFF_2_P
BATCH_IO_DIFF_3_N
BATCH_IO_DIFF_3_P
BATCH_IO_DIFF_4_N
BATCH_IO_DIFF_4_P
BATCH_IO_DIFF_5_N
BATCH_IO_DIFF_5_P
BATCH_IO_DIFF_6_N
BATCH_IO_DIFF_6_P
BATCH_IO_DIFF_7_N
BATCH_IO_DIFF_7_P
BATCH_IO_DIFF_8_N
BATCH_IO_DIFF_8_P
BATCH_IO_DIFF_9_N
BATCH_IO_DIFF_9_P
BATCH_IO_DIFF_10_N
BATCH_IO_DIFF_10_P
BATCH_IO_DIFF_11_N
BATCH_IO_DIFF_11_P
BATCH_IO_DIFF_12_N
BATCH_IO_DIFF_12_P
BATCH_IO_DIFF_13_N
BATCH_IO_DIFF_13_P
BATCH_IO_DIFF_14_N
BATCH_IO_DIFF_14_P
BATCH_IO_DIFF_15_N
BATCH_IO_DIFF_15_P
BATCH_IO_DIFF_16_N
BATCH_IO_DIFF_16_P
Title Batch IO differential bits 1 to 8
Size: A4
Number: 36
Revision: 1.0
Engineer: Ioannis Besiurzanos
Sheet 36 of 40
ZYNQ DEMONSTRATOR BOARD
J44A
VCC_5V
3V3_FPGA
GND
D29
D28
D27
D26
R164
R165
R166
R167
R168
R169
R170
R171
R172
R173
R174
R175
R176
R177
R178
R179


Title Batch I/O differential bits 9 to 16
Size: A4
Number: 37
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 37 of 40
ZYNQ DEMONSTRATOR BOARD
J44B
D33
D32
D31
D30
R195
R194
R193
R192
R191
R190
R189
R188
R187
R186
R185
R184
R183
R182
R181
R180
VCC_5V
GND
3V3_FPGA
BATCH_IO_DIFF_16_P
BATCH_IO_DIFF_16_N
BATCH_IO_DIFF_15_P
BATCH_IO_DIFF_15_N
BATCH_IO_DIFF_14_P
BATCH_IO_DIFF_14_N
BATCH_IO_DIFF_13_P
BATCH_IO_DIFF_13_N
BATCH_IO_DIFF_12_P
BATCH_IO_DIFF_12_N
BATCH_IO_DIFF_11_P
BATCH_IO_DIFF_11_N
BATCH_IO_DIFF_10_P
BATCH_IO_DIFF_10_N
BATCH_IO_DIFF_9_P
BATCH_IO_DIFF_9_N


1
2
3
4
A
B
C
D
1
SMA_USER_GPIO_2_P
MNT 1
MNT 2
MNT 3
MNT 4
J45
GND
SMA_USER_GPIO_2_N
J46
SMA_USER_GPIO_1_P
J47
SMA_USER_GPIO_1_N
J48
Title SMA GPIO
Size: A4
Number: 38
Revision: 1.0
Engineer: Ioannis Bestintzanos
Sheet 38 of 40
ZYNQ DEMONSTRATOR BOARD
CMS


XADC
Size: A4
Number: 39
Revision: 1.0
Sheet 39 of 40
ZYNQ DEMONSTRATOR BOARD
J51A
J51B
XADC-V_P
XADC-V_N
XADC-AD0_P
XADC-AD0_N
XADC-AD8_P
XADC-AD8_N
XADC_DX_P
XADC_DX_N
XADC_VREFP
XADC-5V0
XADC-1V8
VCCAUX1
VCC_5V
3V3_FPGA
AMS_I2C_SDA
AMS_I2C_SCL
XADC-AGND
XADC-VR_P
XADC-VR_N
R250
R251
C115
1nF
100
XADC-1V25
C252
R279
0.01
U23
MAX6037BAUK12-T
ADJ
OUT
GND
SHDN
IN
C251
R225
0R
NO LOAD
R226
0R
P3
MHDR1X2
FB11
600@100 MHz
VCC_5V
R280
0.01
R227
49.9K 0.1%
R228
40K 0.1%
C254
10nF
C255
0.1nF
C253
10nF
U22
MAX1983EUT+
IN
OUT
PG/ADJ
SHDN
GND
BIAS
R255
R256
R257
R258
C416
C417
1nF
XADC-AD0R_P
XADC-AD0R_N
XADC-AD8R_P
XADC-AD8R_N

Mounting holes

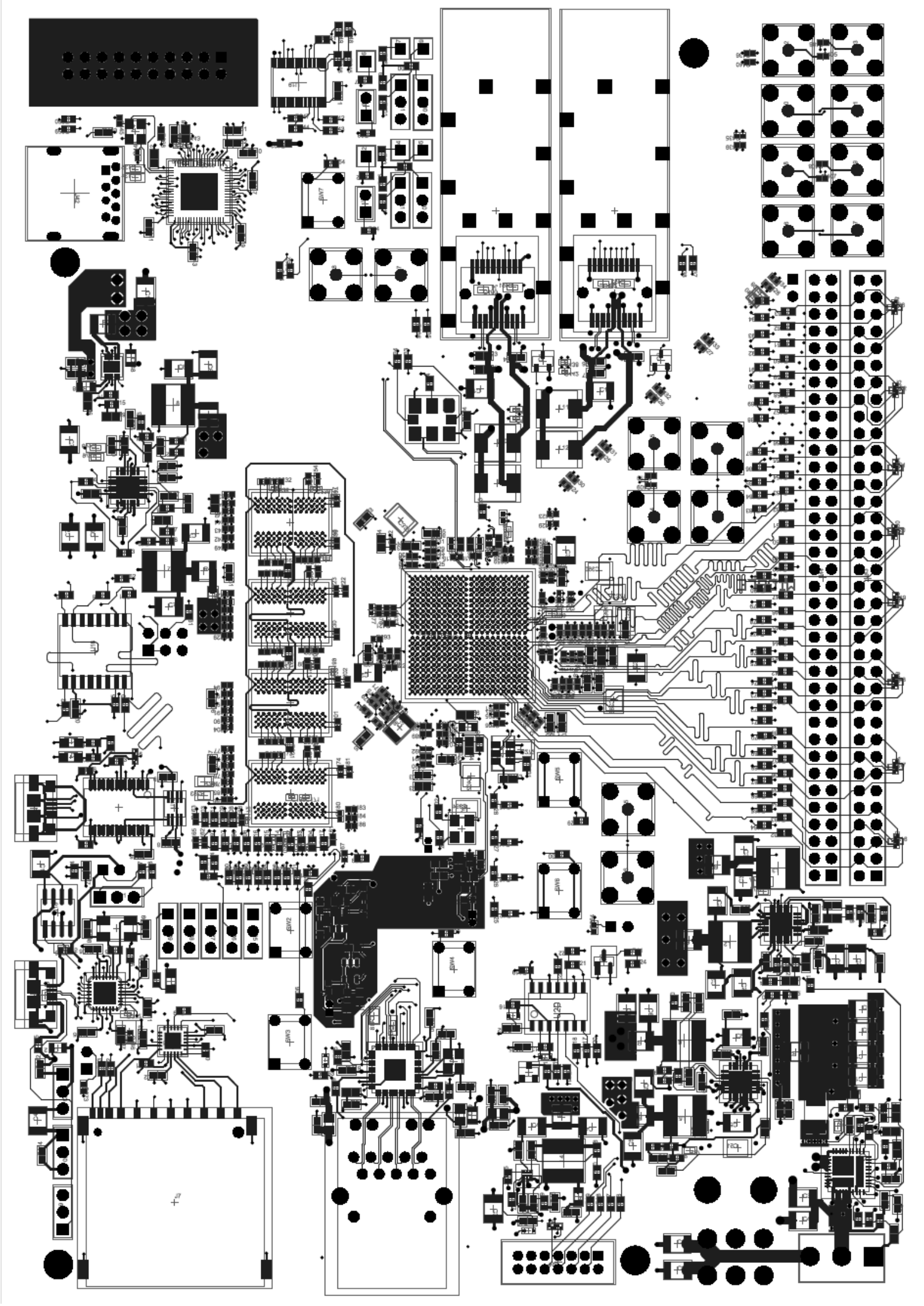


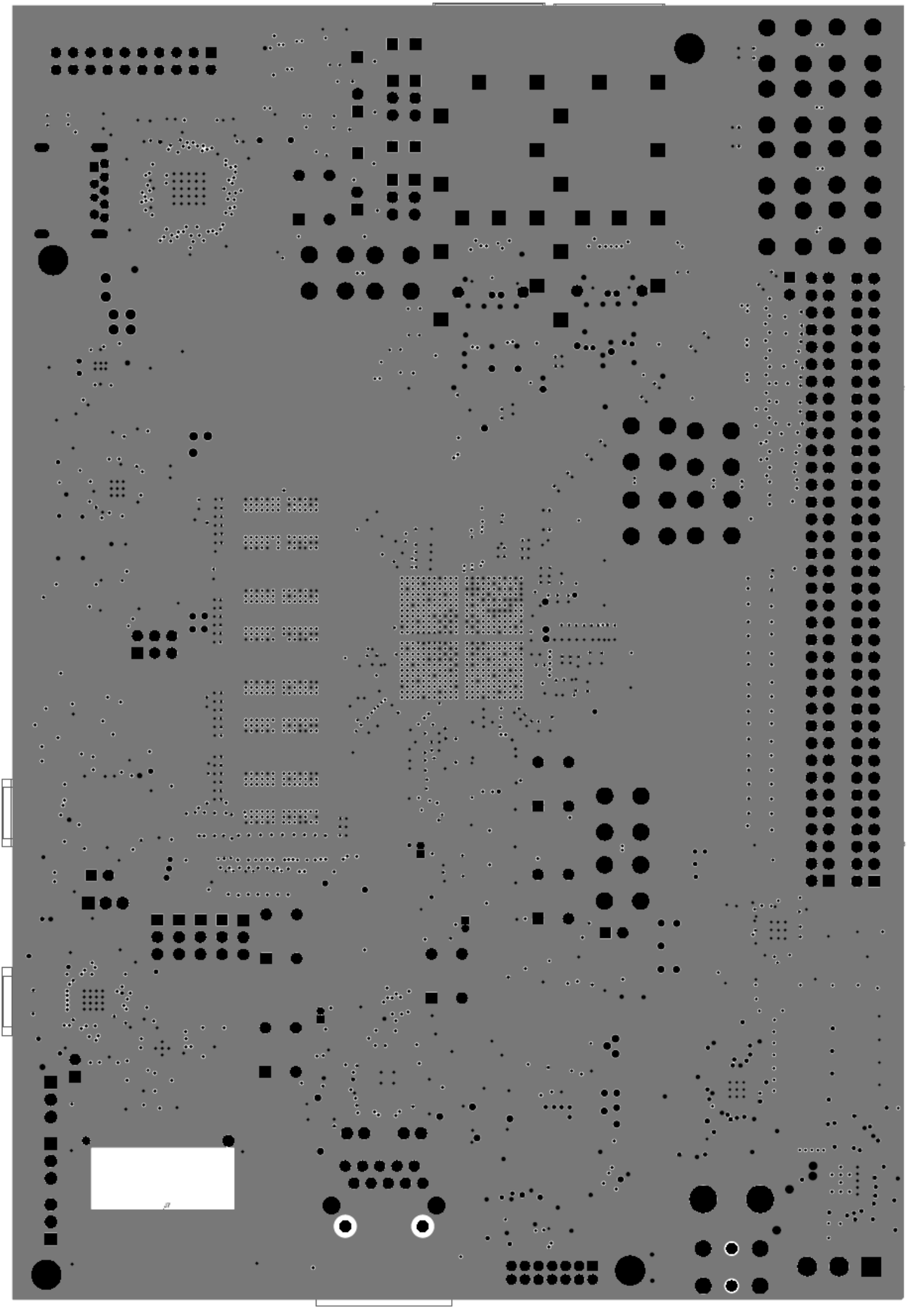
Title *Mechanicals*		
Size: A4	Number: 40	Revision: 1.0
Engineer: Ioannis Bestintzanos		Sheet 40 of 40
ZYNQ DEMONSTRATOR BOARD		

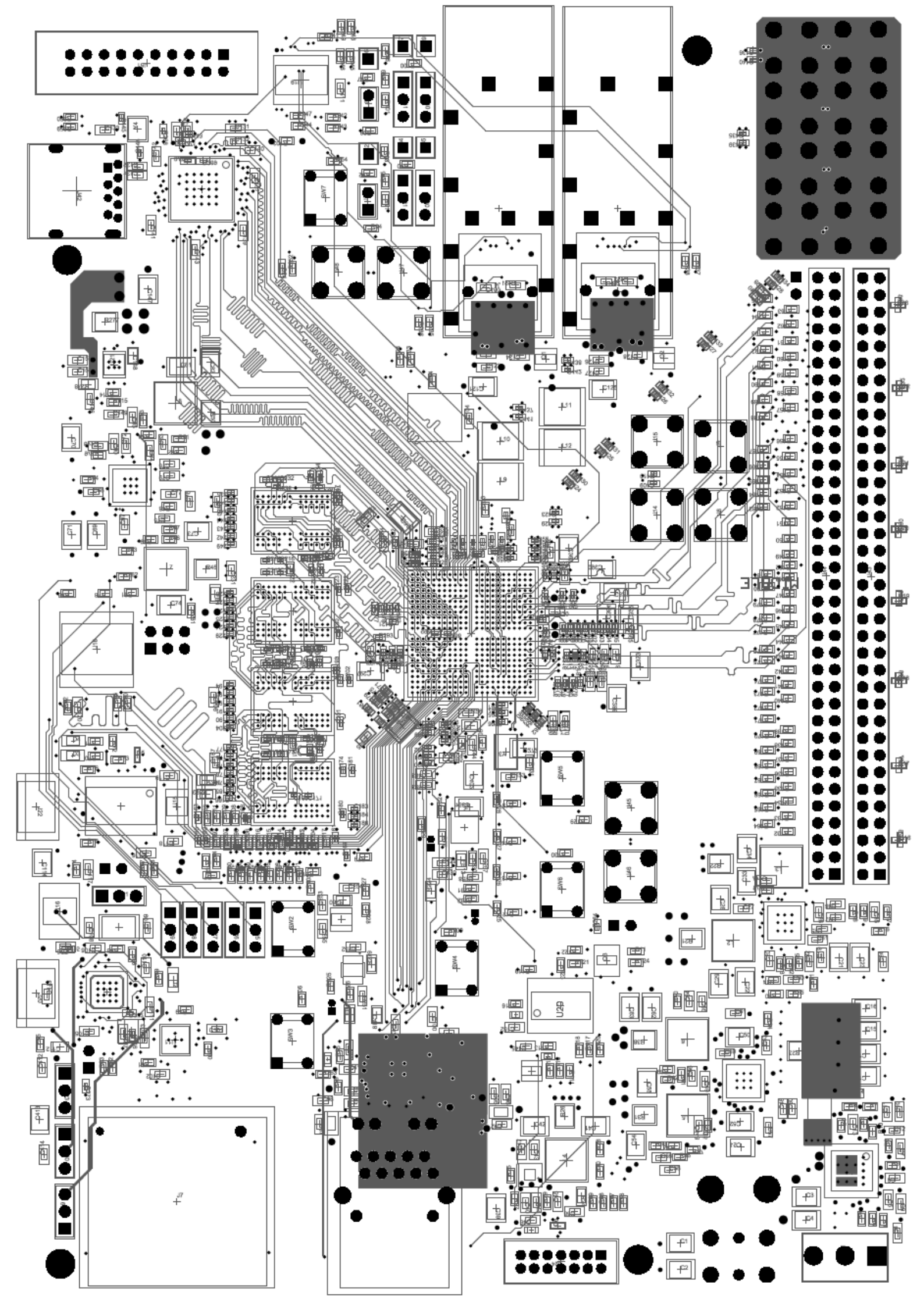
7.4 Gerber files

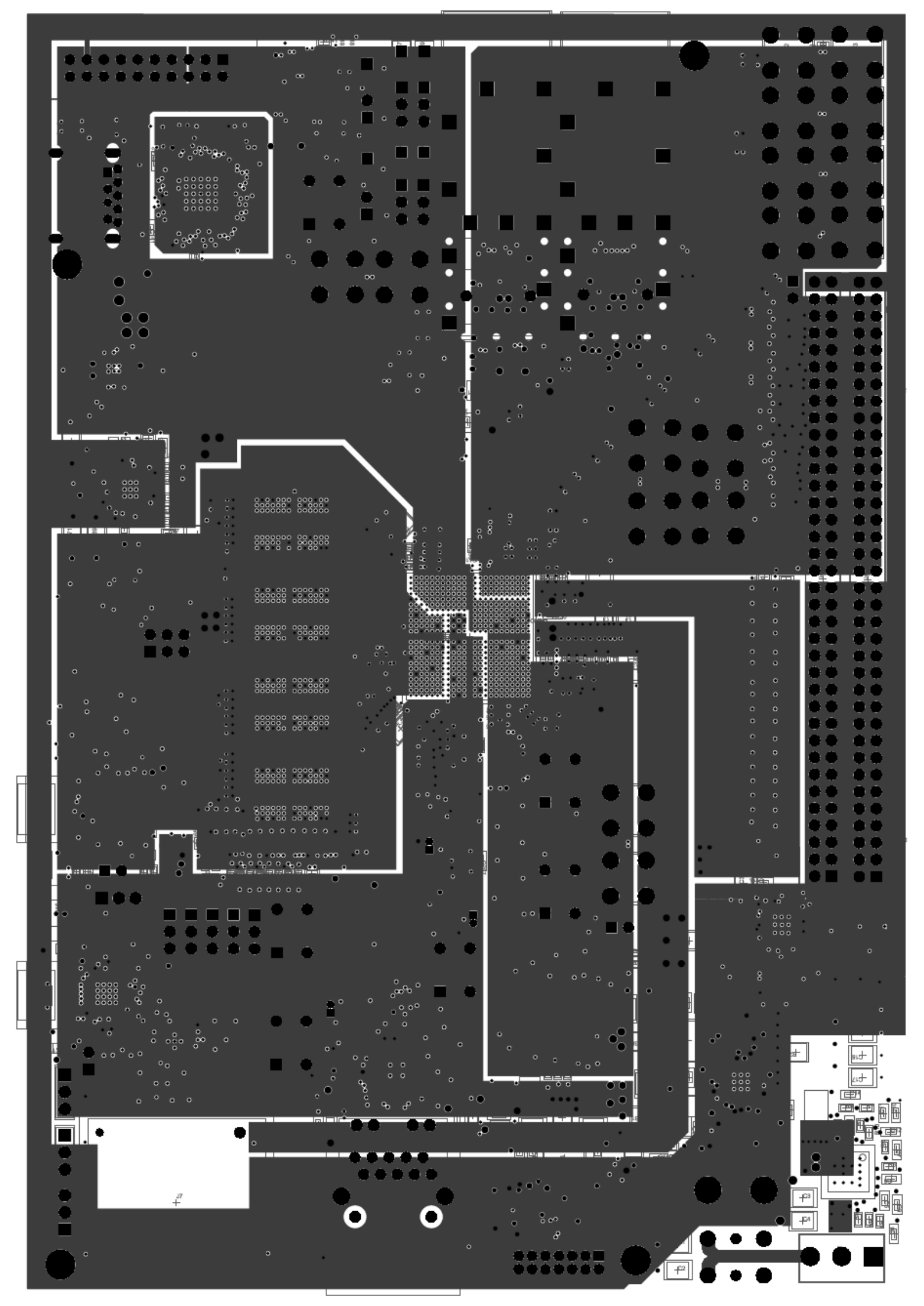
The manufacturing files are shown below. The files included are:

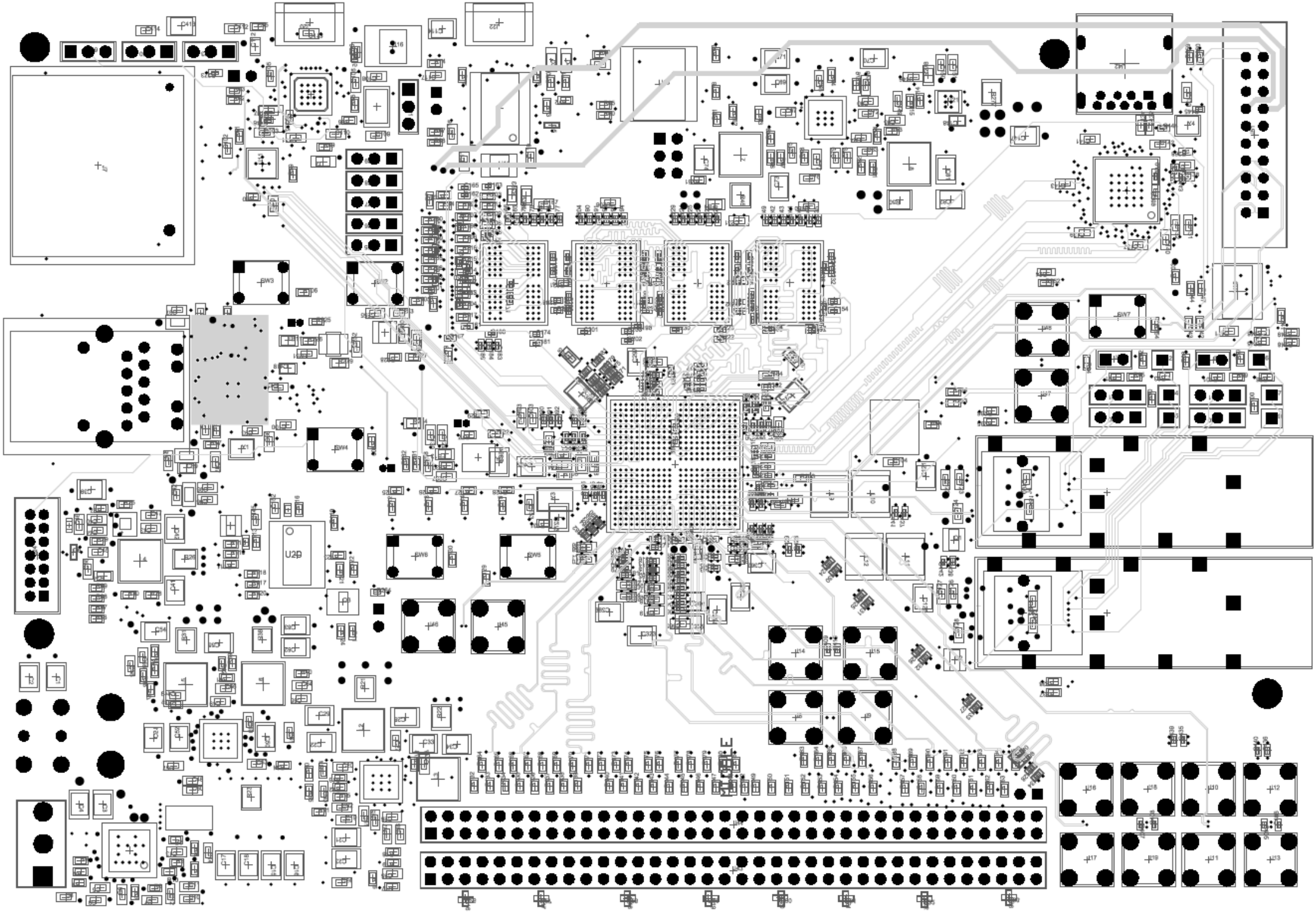
- **Layer 1 – Signal**
- **Layer 2 – GND**
- **Layer 3 – Signal**
- **Layer 4 – Power**
- **Layer 5 – Signal**
- **Layer 6 – Power**
- **Layer 7 – Signal**
- **Layer 8 – Signal**
- **Top Silkscreen**
- **Bottom Silkscreen**
- **Top Paste**
- **Bottom Paste**
- **Top Solder Mask**
- **Bottom Solder Mask**
- **Mechanical**
- **Top Pad Master**
- **Bottom Pad Master**
- **Drill Drawing**
- **Drill Guide**

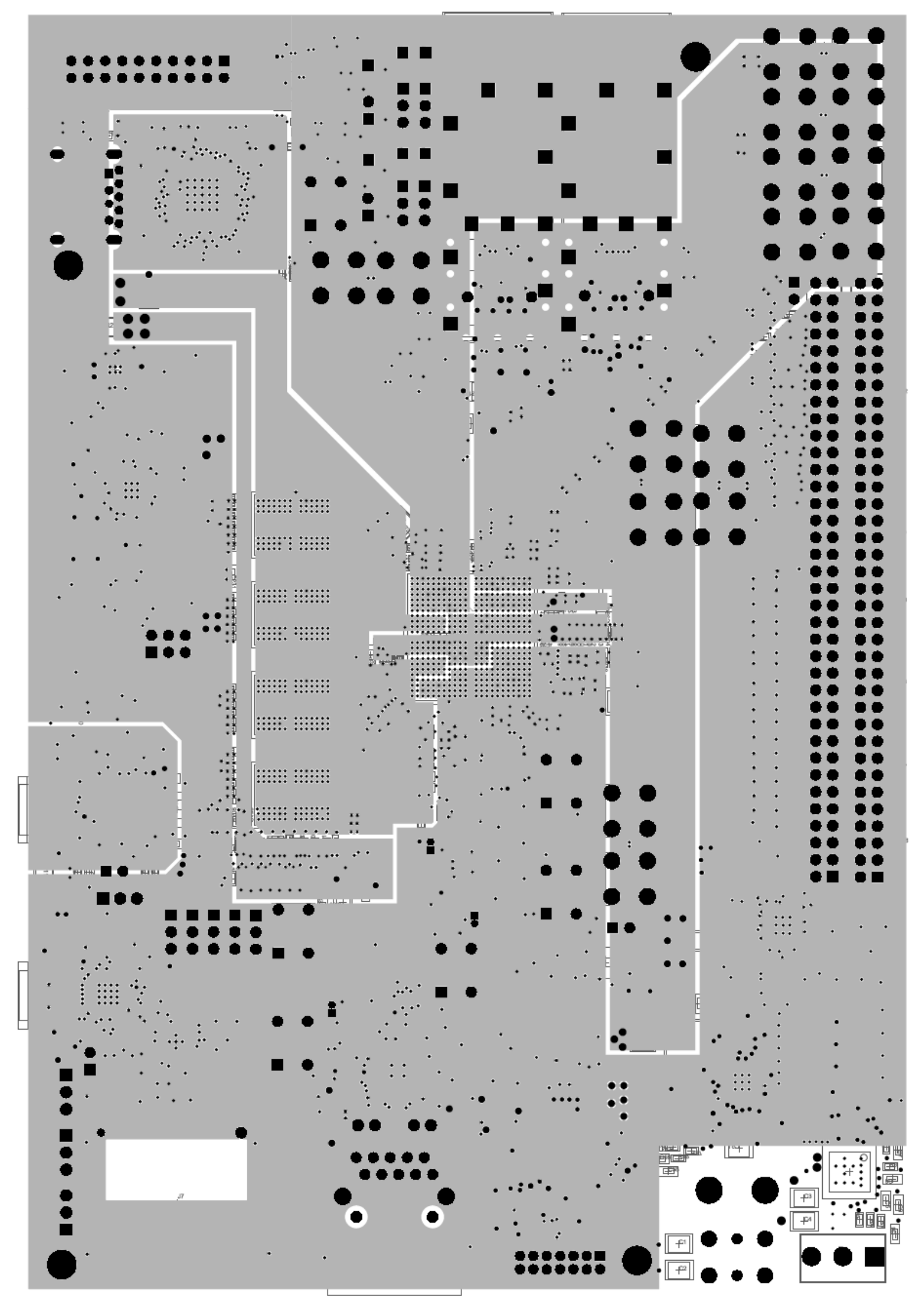


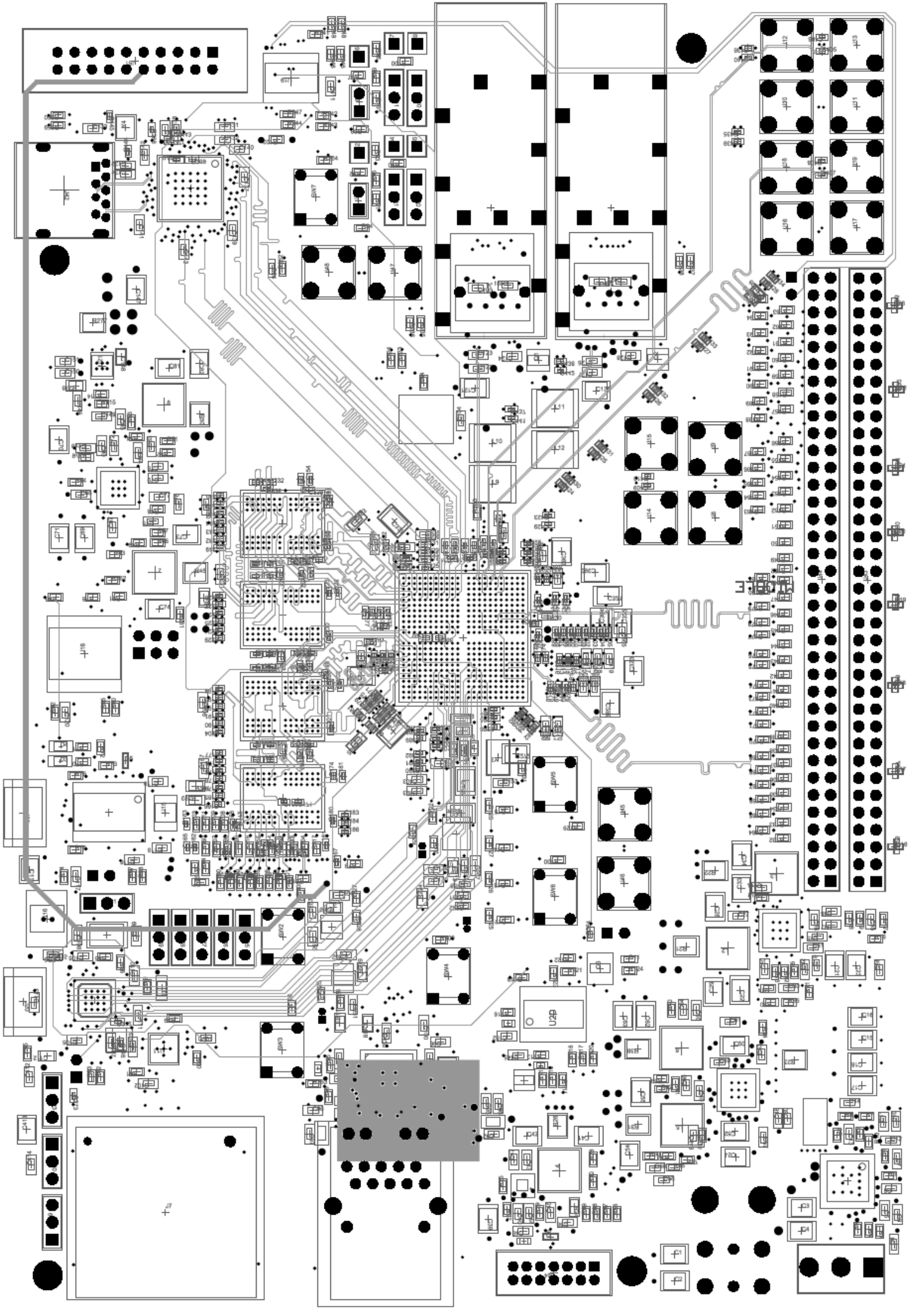


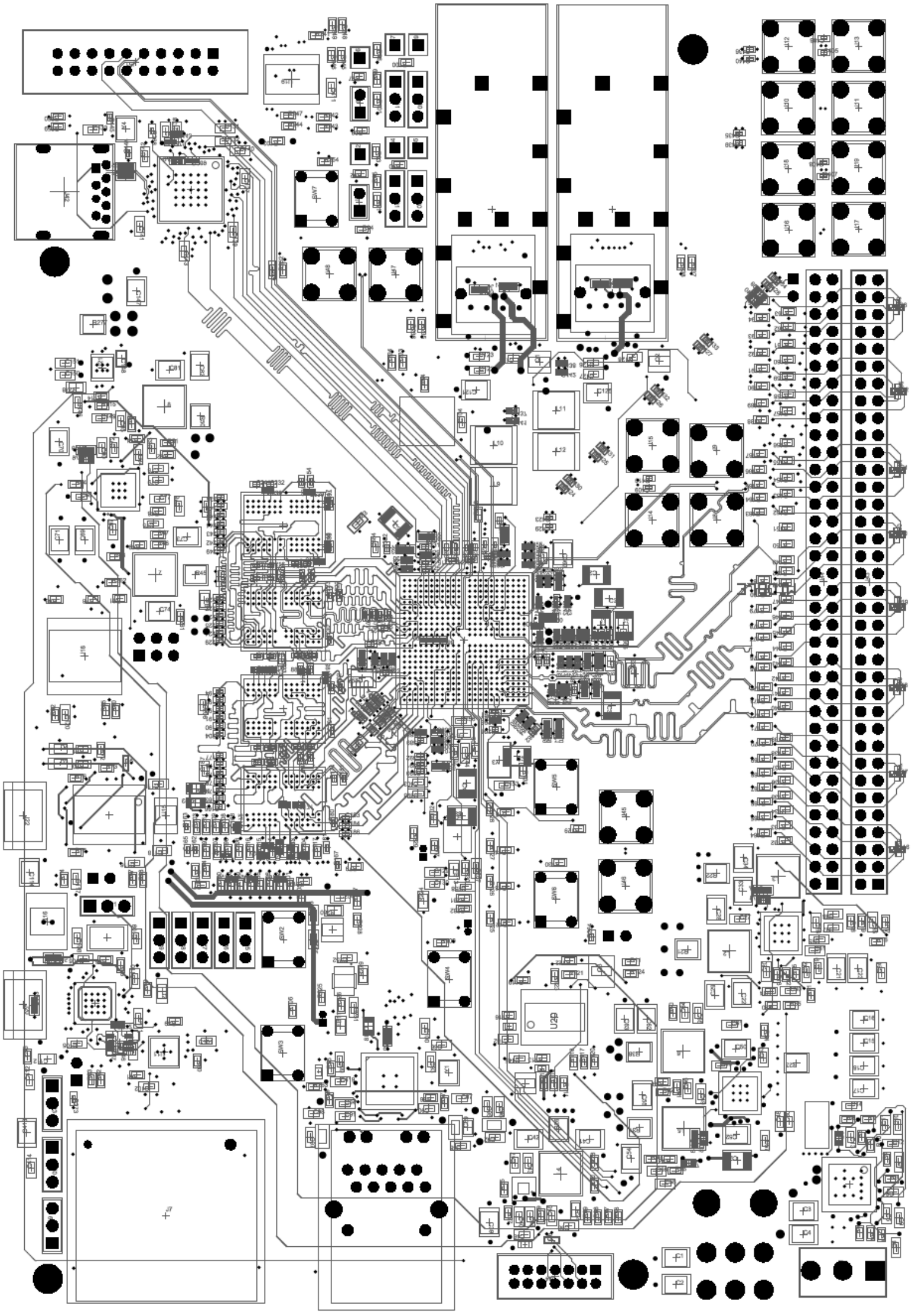


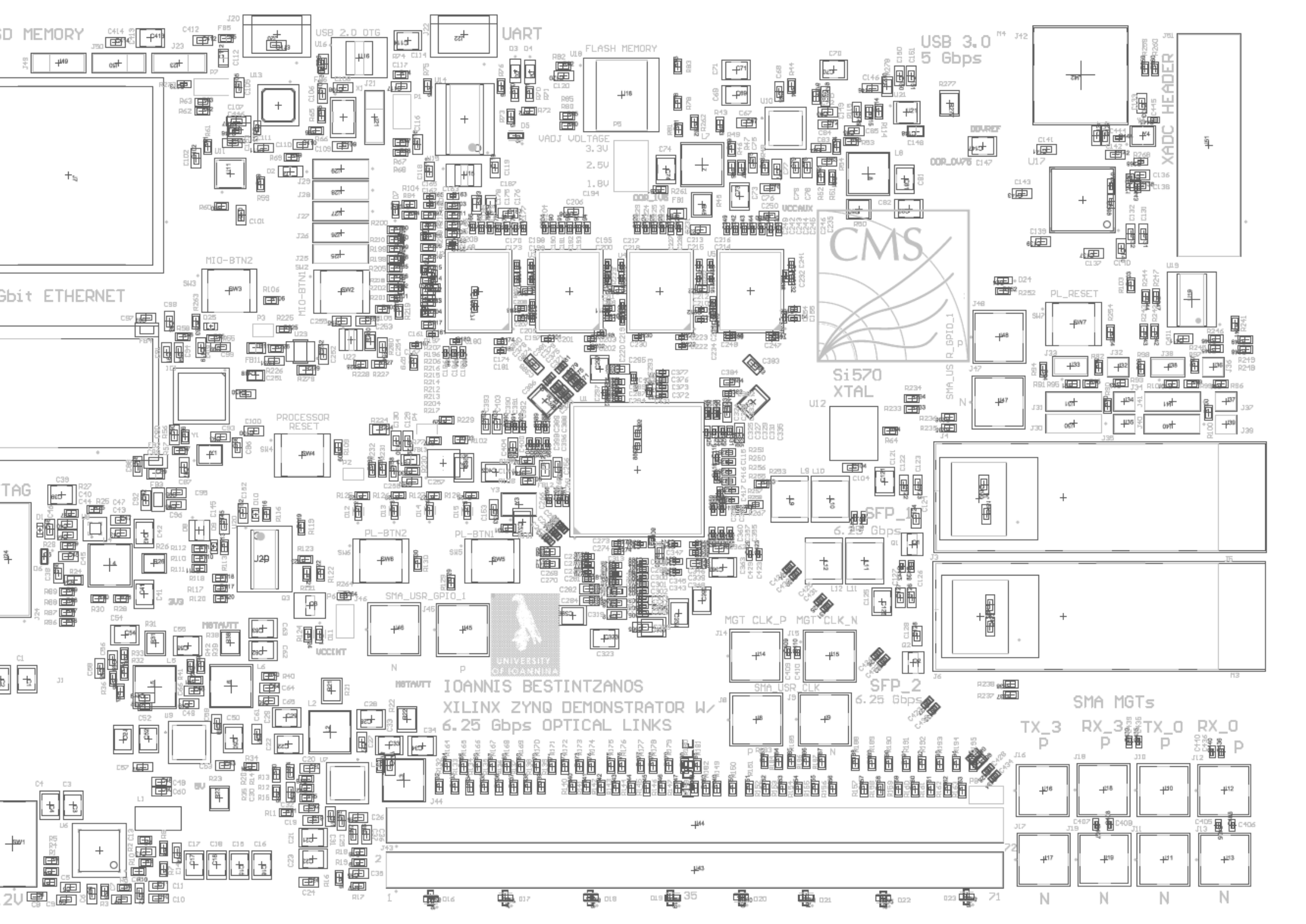


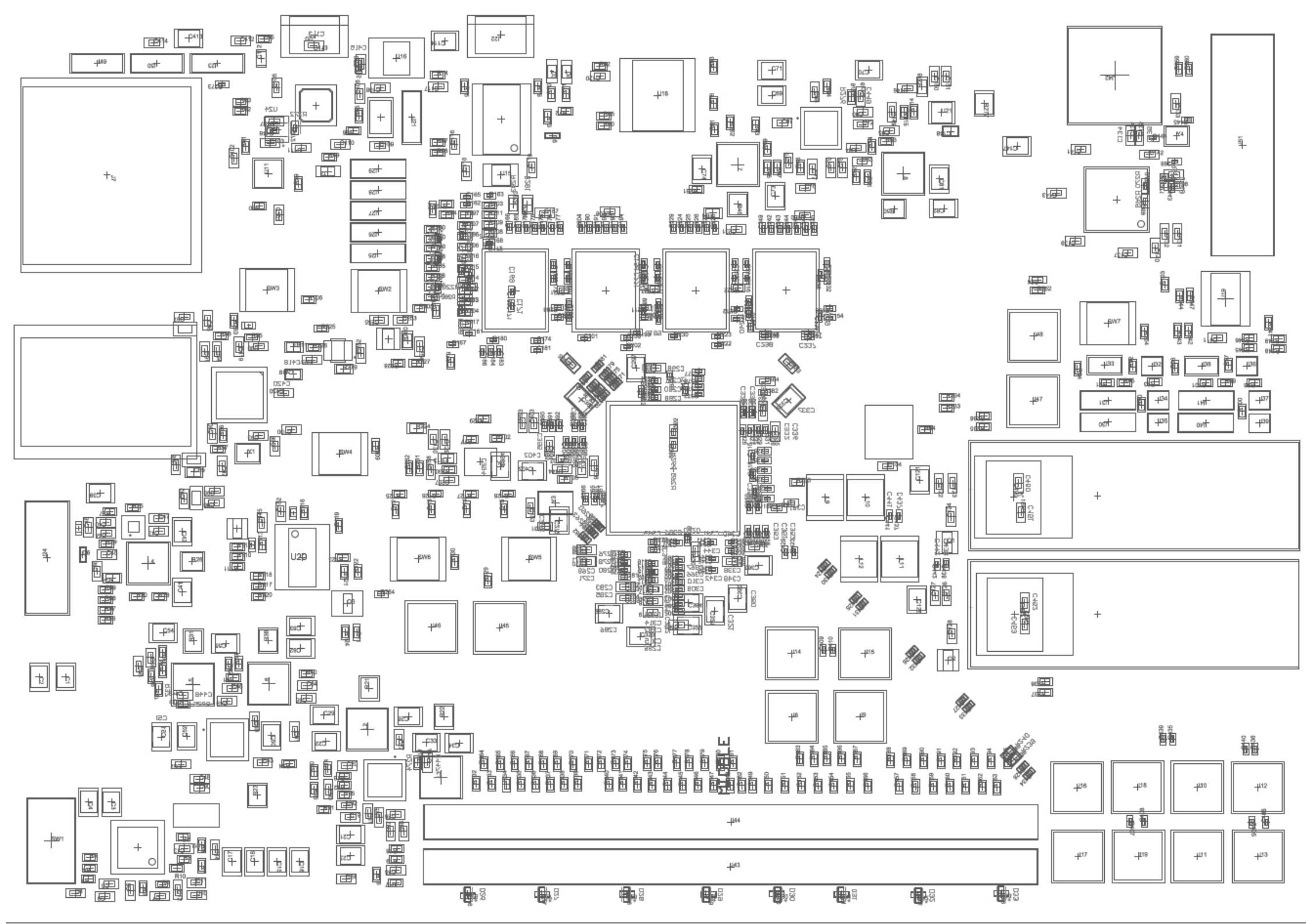


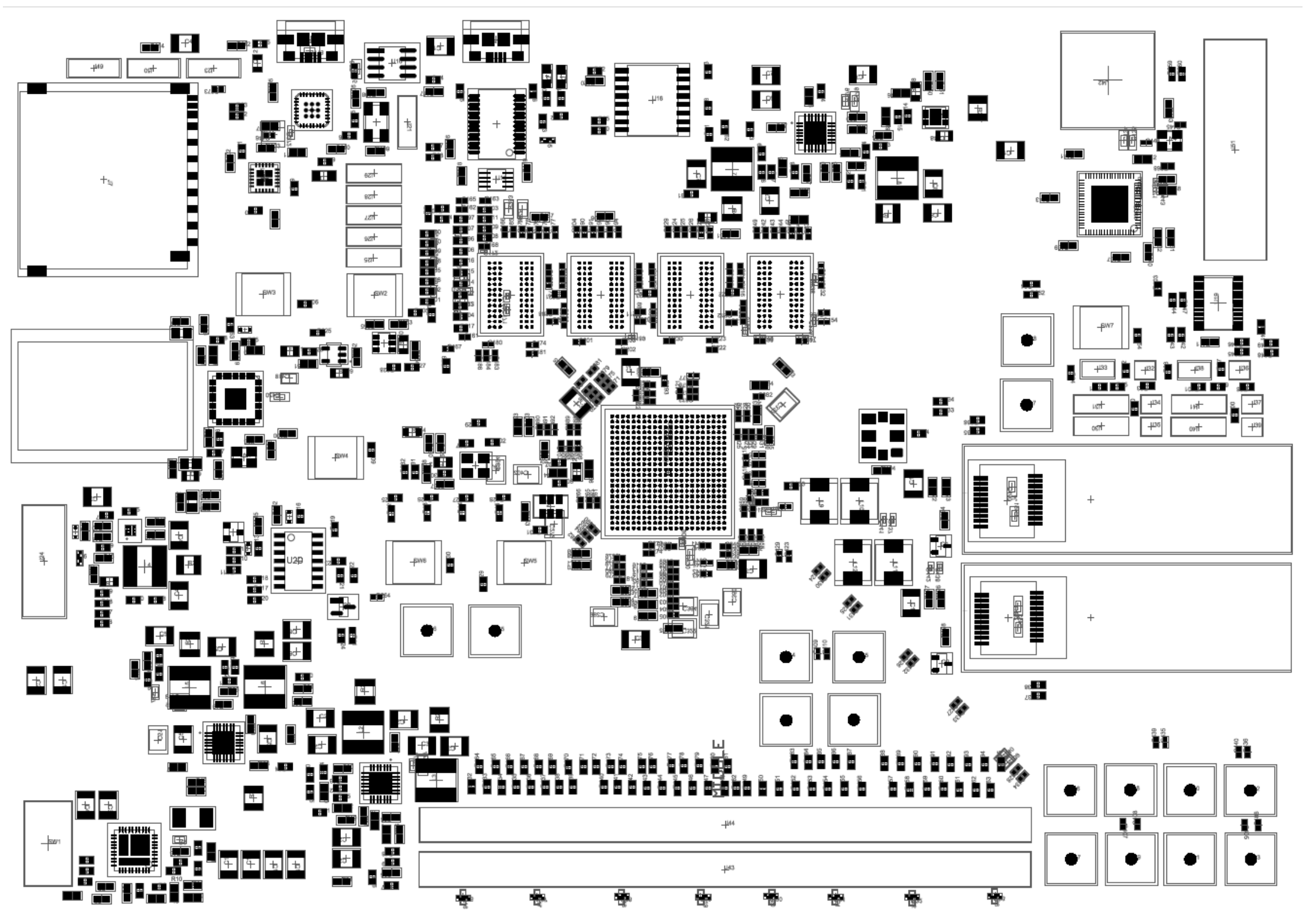


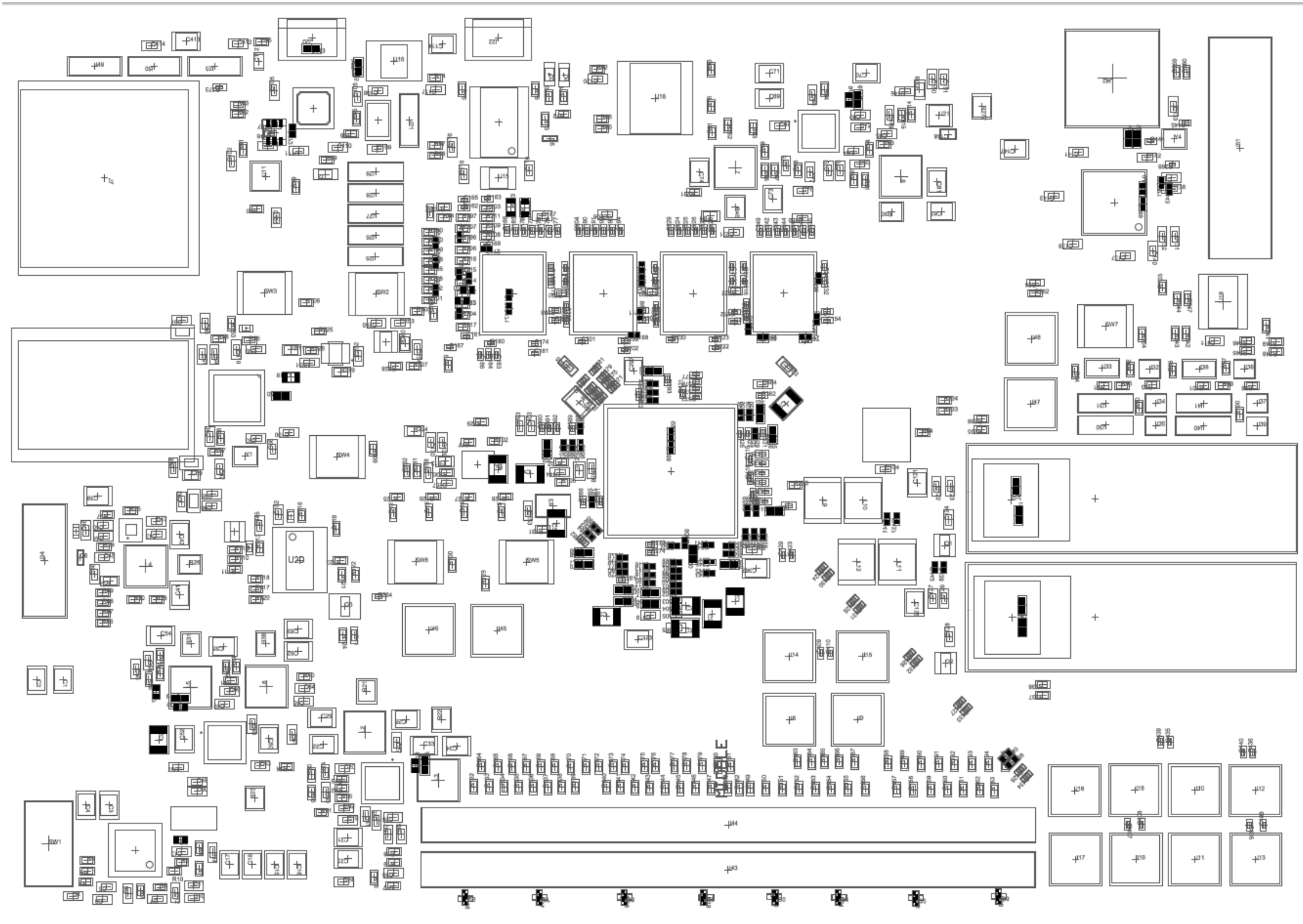


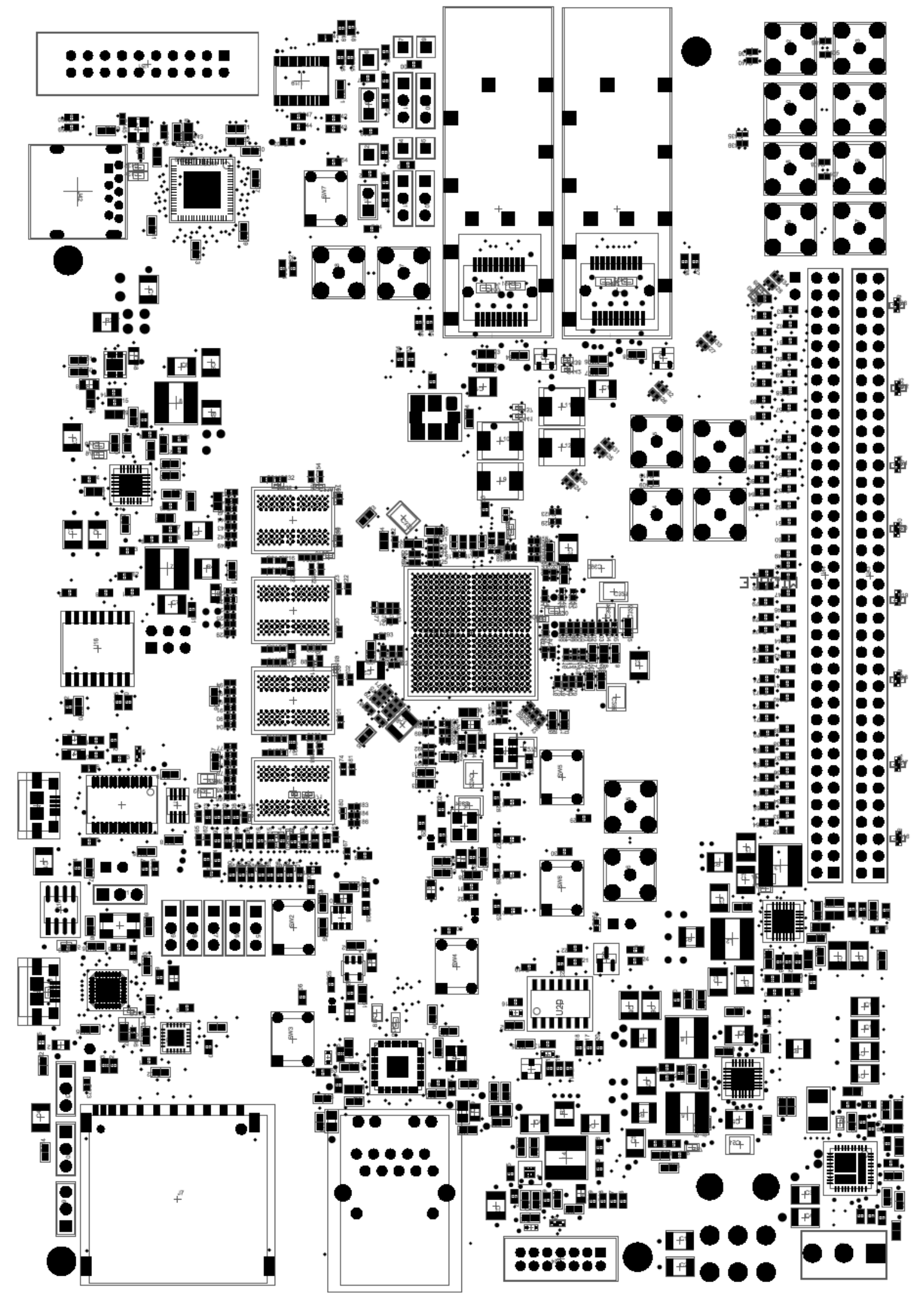


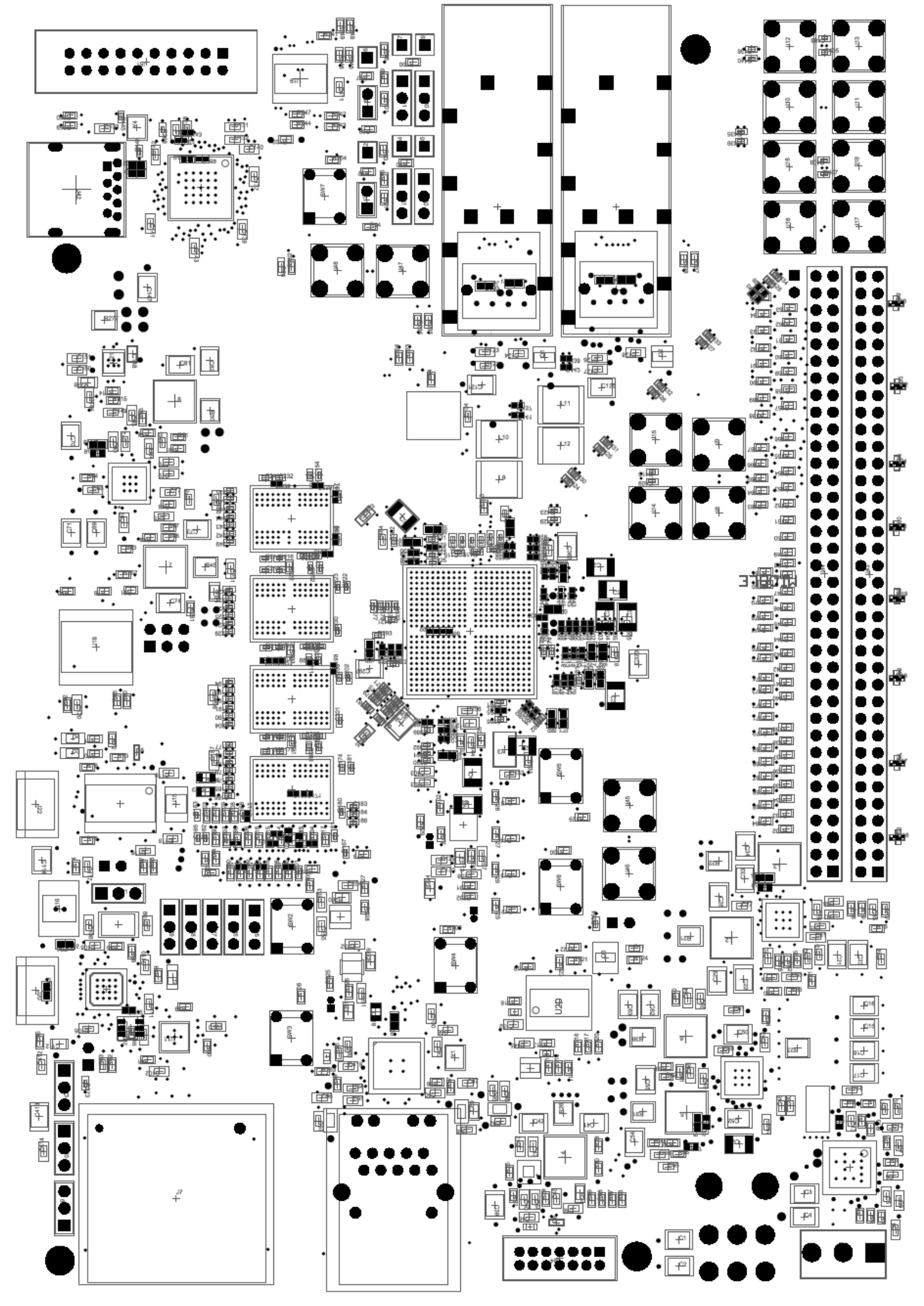

SD MEMORY
USB 2.0 OTG
UART
FLASH MEMORY
USB 3.0
5 Gbps
XADC HEADER
Gbit ETHERNET
MIO-BTN2
MIO-BTN1
VADJ VOLTAGE
3.3V
2.5V
1.8V
PROCESSOR
RESET
PL_RESET
Si570
XTAL
SFP_1
6.25 Gbps
SFP_2
6.25 Gbps
PL-BTN2
PL-BTN1
SMA_USR_GPIO_1
MGT CLK_P
MGT CLK_N
SMA_USR_CLK
SMA MGTs
TX_3 RX_3 TX_0 RX_0
TAG
IOANNIS BESTINTZANOS
XILINX ZYNQ DEMONSTRATOR W/
6.25 Gbps OPTICAL LINKS
UNIVERSITY OF IOANNINA
CMS

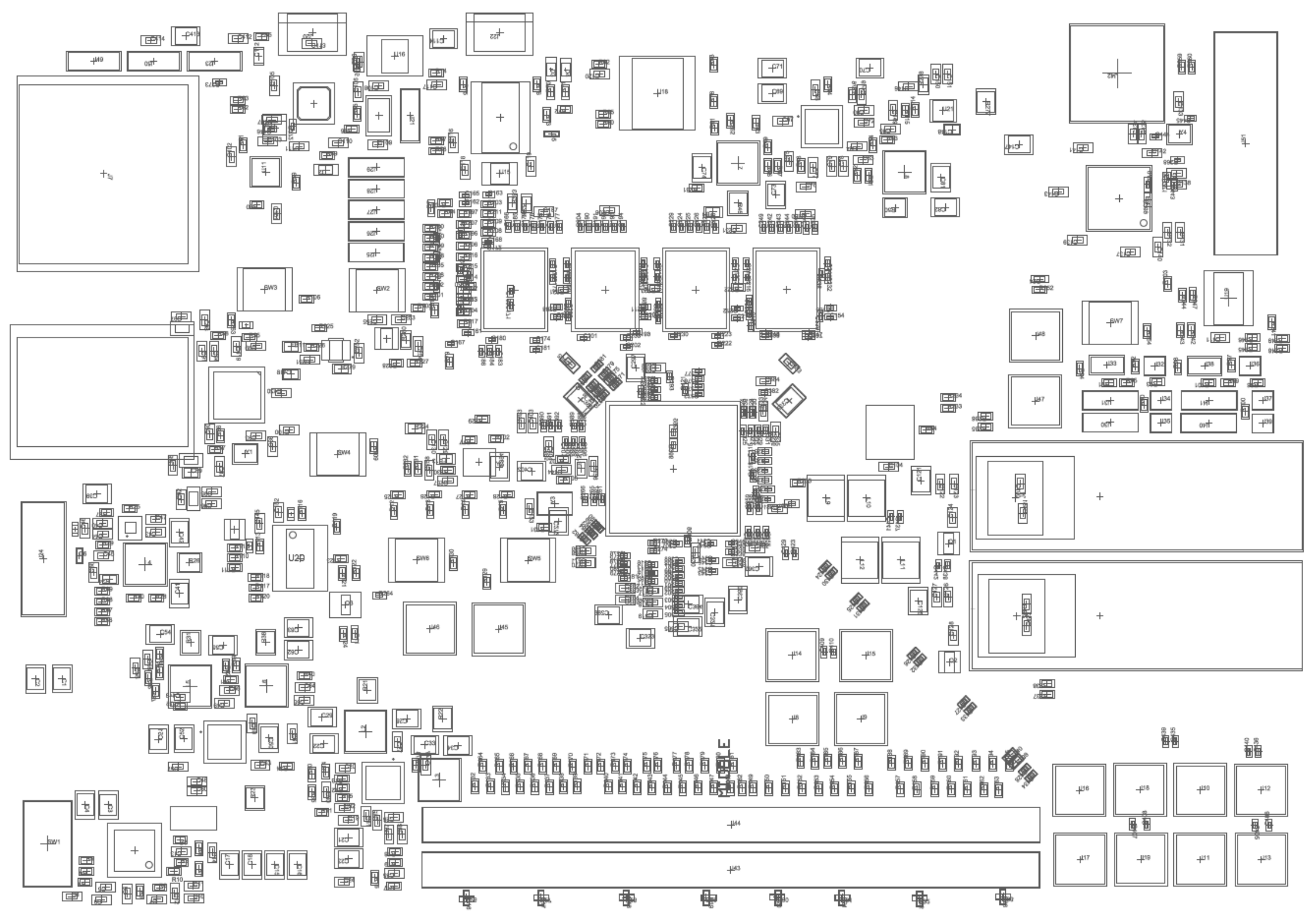


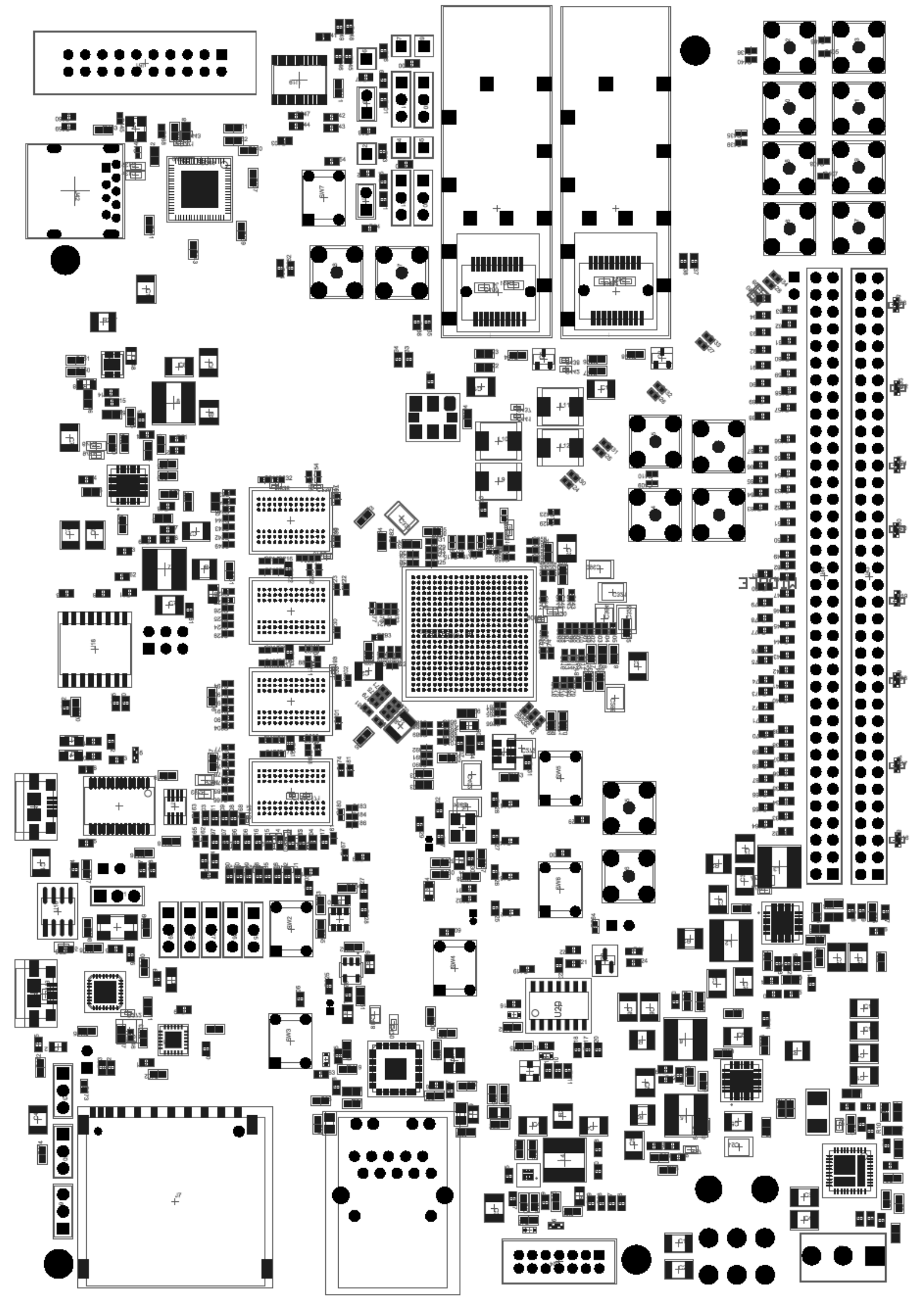


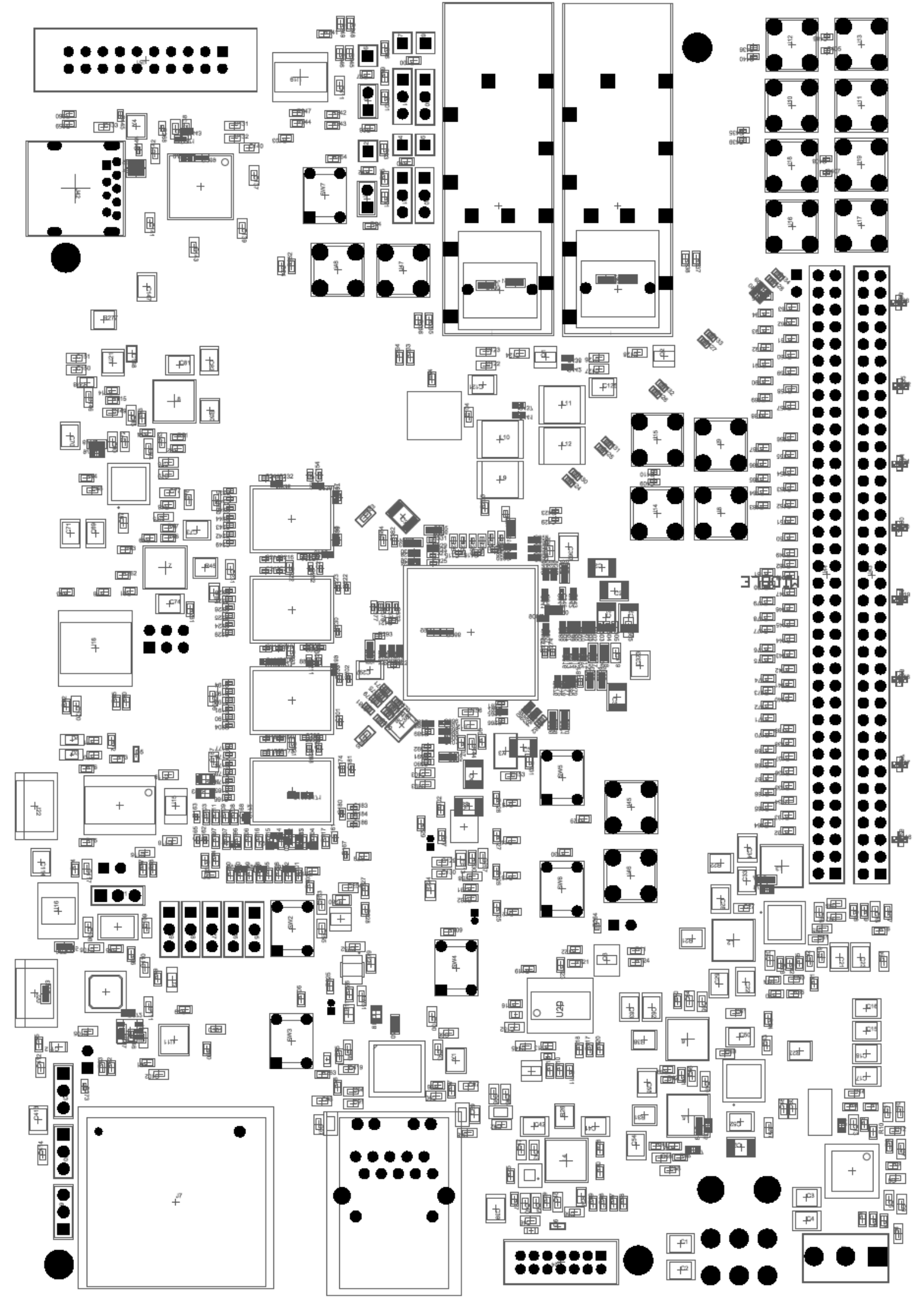


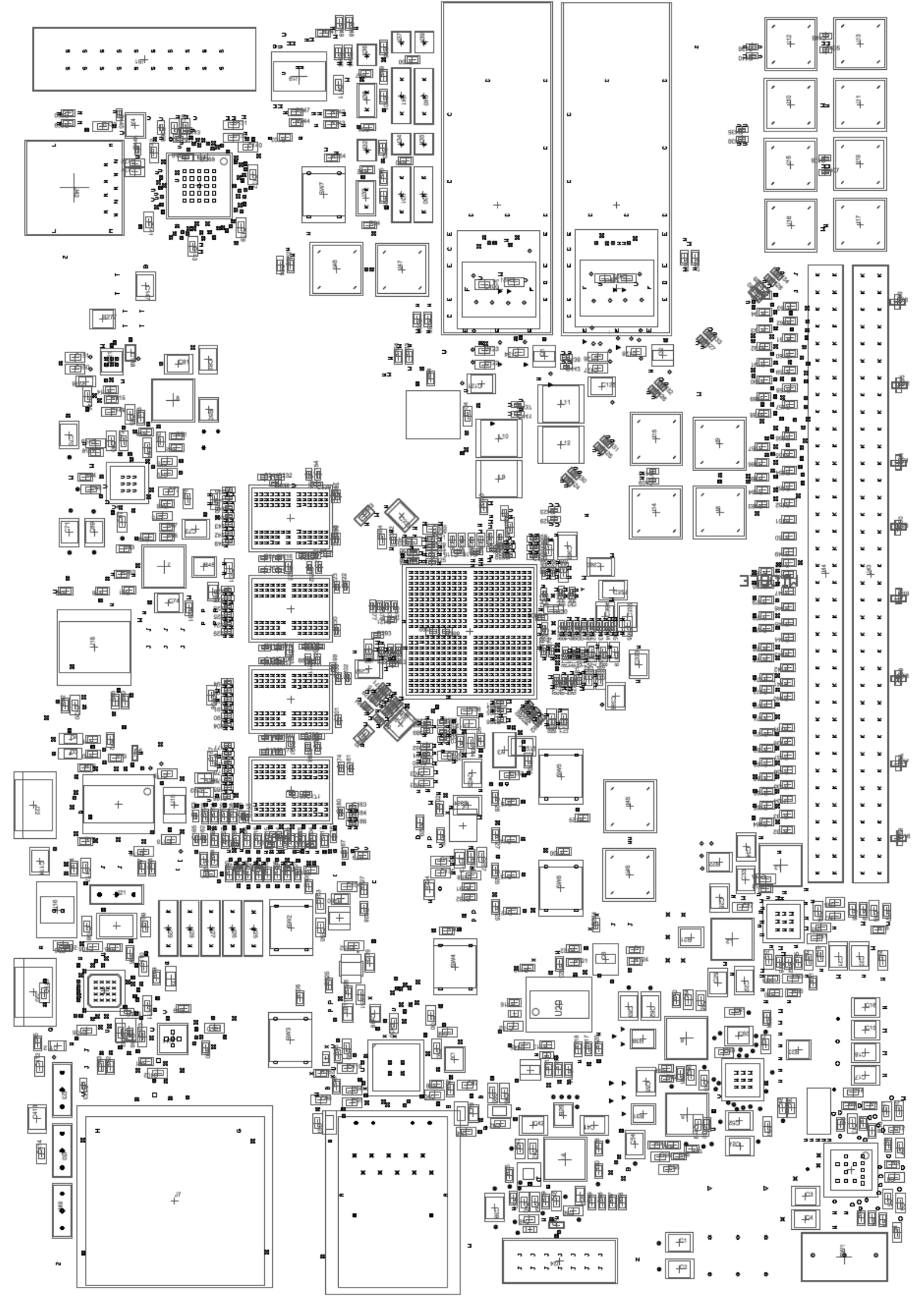


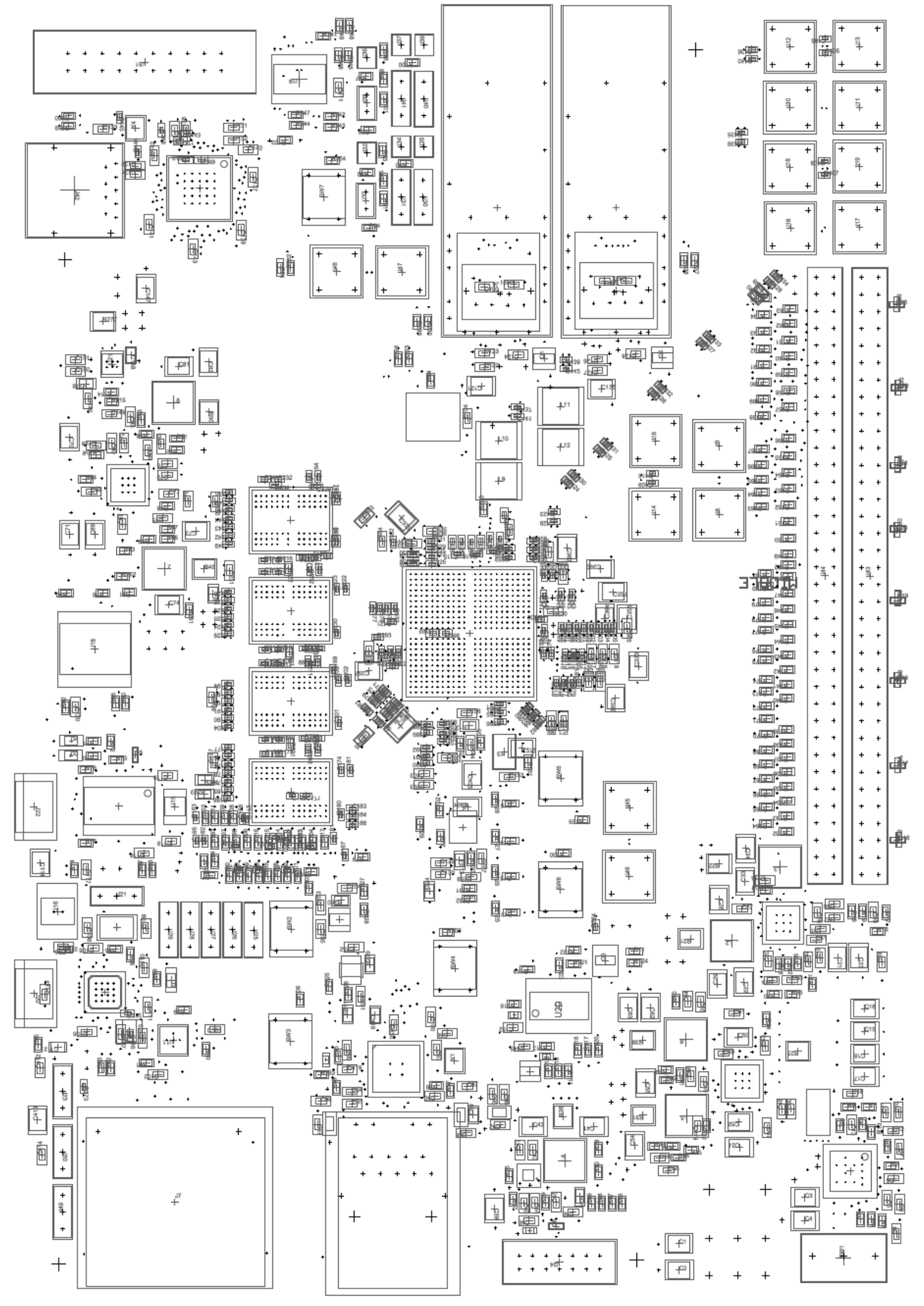












7.5 Trace lengths report

Address / Clock / Command PP1

Below is the report for the trace lengths from the address / clock / command bus of the DDR interface. The traces are from the memory controller on the ZYNQ to the first DRAM chip.

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_A0_PP1	2	2609.734	370.92	2238.814	443.540
DDR3_A1_PP1	2	2720.532	372.87	2347.662	448.389
DDR3_A2_PP1	2	2609.875	346.28	2263.595	443.880
DDR3_A3_PP1	2	2609.986	340.9	2269.086	444.000
DDR3_A4_PP1	2	2624.142	398.9	2225.242	445.698
DDR3_A5_PP1	2	2613.687	388.81	2224.877	444.000
DDR3_A6_PP1	2	2610.129	340.57	2269.559	444.000
DDR3_A7_PP1	2	2614.716	323.23	2291.486	445.052
DDR3_A8_PP1	2	2698.739	417.8	2280.939	444.000
DDR3_A9_PP1	2	2609.336	329.4	2279.936	444.000
DDR3_A10_PP1	2	2692.034	324.83	2367.204	444.000
DDR3_A11_PP1	2	2610.127	340.21	2269.917	444.000
DDR3_A12_PP1	2	2694.055	350.9	2343.155	443.999
DDR3_A13_PP1	2	2618.969	458.96	2160.009	444.000
DDR3_A14_PP1	2	2612.363	369.6	2242.762	444.000
DDR3_BA0_PP1	2	2697.347	341.52	2355.827	444.702
DDR3_BA1_PP1	2	2699.309	442.14	2257.169	443.838
DDR3_BA2_PP1	2	2691.822	321.9	2369.922	444.000
DDR3_CKE_PP1	2	2749.594	600.99	2148.604	444.210
DDR3_CLK_N_PP1	2	2698.831	473	2225.831	444.362
DDR3_CLK_P_PP1	2	2699.85	482	2217.85	444.444
DDR3_C\A\S_PP1	2	2625.836	466.87	2158.966	445.080
DDR3_C\S_PP1	2	2626.938	547.33	2079.608	444.248
DDR3_ODT_PP1	2	2621.949	472.17	2149.779	444.349
DDR3_R\A\S_PP1	2	2707.887	561	2146.887	443.853
DDR3_W\E_PP1	2	2721.438	739.59	1981.848	444.000

Address / Clock / Command PP2

Below is the report for the trace lengths from the address / clock / command bus of the DDR interface. The traces are from the first DRAM to the second DRAM chip.

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_A0_PP2	2	3705.081	370.92	3334.161	625,768
DDR3_A1_PP2	2	3923.729	372.87	3550.859	627,595
DDR3_A2_PP2	2	3697.099	346.28	3350.819	624,702
DDR3_A3_PP2	2	3697.854	340.9	3356.954	624,942
DDR3_A4_PP2	2	3929.063	398.9	3530.163	629,165
DDR3_A5_PP2	2	3715.966	388.81	3327.157	627,412
DDR3_A6_PP2	2	3902.407	340.57	3561.837	624,823
DDR3_A7_PP2	2	3706.025	323.23	3382.795	626,577
DDR3_A8_PP2	2	3913.384	417.8	3495.584	625,179
DDR3_A9_PP2	2	3721.799	329.4	3392.399	629,165
DDR3_A10_PP2	2	3919.235	324.83	3594.405	626,810
DDR3_A11_PP2	2	3912.295	340.21	3572.085	626,886
DDR3_A12_PP2	2	3928.317	350.9	3577.417	627,476
DDR3_A13_PP2	2	3707.296	458.96	3248.336	624,722
DDR3_A14_PP2	2	3915.485	369.6	3545.885	627,340
DDR3_BA0_PP2	2	3757.861	341.52	3416.341	627,396
DDR3_BA1_PP2	2	3930.255	442.14	3488.115	627,249
DDR3_BA2_PP2	2	3919.909	321.9	3598.009	627,002
DDR3_CKE_PP2	3	3926.46	600.99	3325.47	624,841
DDR3_CLK_N_PP2	2	3996.137	473	3523.137	628,001
DDR3_CLK_P_PP2	2	3999.958	482	3517.958	628,488
DDR3_C\A\S_PP2	2	3721.895	466.87	3255.025	627,462
DDR3_C\S_PP2	2	3731.21	547.33	3183.88	628,009
DDR3_ODT_PP2	2	3914.412	472.17	3442.242	626,146
DDR3_R\A\S_PP2	2	3777.133	561	3216.133	628,013
DDR3_W\E_PP2	2	3768.397	739.59	3028.807	624,337

0Address / Clock / Command PP3

Below is the report for the trace lengths from the address / clock / command bus of the DDR interface. The traces are from the second DRAM to the third DRAM chip.

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_A0_PP3	2	4668.76	370.92	4297.84	792.376
DDR3_A1_PP3	2	4887.435	372.87	4514.565	793.595
DDR3_A2_PP3	2	4690.545	346.28	4344.265	796.457
DDR3_A3_PP3	2	4665.311	340.9	4324.411	792.204
DDR3_A4_PP3	2	4903.713	398.9	4504.813	797.046
DDR3_A5_PP3	2	4670.81	388.81	4282	792.493
DDR3_A6_PP3	2	4878.811	340.57	4538.241	793.000
DDR3_A7_PP3	2	4673.039	323.23	4349.81	793.762
DDR3_A8_PP3	2	4887.755	417.8	4469.955	793.000
DDR3_A9_PP3	2	4693.31	329.4	4363.91	797.127
DDR3_A10_PP3	2	4882.761	324.83	4557.931	792.768
DDR3_A11_PP3	2	4876.479	340.21	4536.269	792.956
DDR3_A12_PP3	2	4889.269	350.9	4538.369	793.000
DDR3_A13_PP3	2	4679.232	458.96	4220.272	792.758
DDR3_A14_PP3	2	4877.345	369.6	4507.745	793.000
DDR3_BA0_PP3	2	4718.533	341.52	4377.013	793.484
DDR3_BA1_PP3	2	4892.396	442.14	4450.256	792.994
DDR3_BA2_PP3	2	4883.612	321.9	4561.712	793.000
DDR3_CKE_PP3	4	5023.399	600.99	4422.409	792.964
DDR3_CLK_N_PP3	2	4974.662	473	4501.662	797.161
DDR3_CLK_P_PP3	2	4975.837	482	4493.837	797.203
DDR3_C\A\S_PP3	2	4680.233	466.87	4213.363	793.147
DDR3_C\S_PP3	2	4690.3	547.33	4142.97	793.824
DDR3_ODT_PP3	2	4883.189	472.17	4411.019	793.000
DDR3_R\A\S_PP3	2	4737.515	561	4176.515	794.076
DDR3_W\E_PP3	2	4751.023	739.59	4011.433	794.221

Address / Clock / Command PP4

Below is the report for the trace lengths from the address / clock / command bus of the DDR interface. The traces are from the third DRAM to the fourth DRAM chip. Shortly after the last DRAM chip the trace are terminated

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_A0_PP4	2	5950.575	370.92	5579.655	972.304
DDR3_A1_PP4	2	5926.851	372.87	5553.981	973.297
DDR3_A2_PP4	2	5751.827	346.28	5405.547	972.819
DDR3_A3_PP4	2	5956.391	340.9	5615.491	973.363
DDR3_A4_PP4	2	5938.901	398.9	5540.001	976.017
DDR3_A5_PP4	2	5971.859	388.81	5583.05	975.385
DDR3_A6_PP4	2	5920.444	340.57	5579.874	973.095
DDR3_A7_PP4	2	5974.137	323.23	5650.907	976.641
DDR3_A8_PP4	2	5928.954	417.8	5511.154	973.000
DDR3_A9_PP4	2	5750.912	329.4	5421.512	972.844
DDR3_A10_PP4	2	5925.241	324.83	5600.411	973.000
DDR3_A11_PP4	2	5918.325	340.21	5578.115	973.078
DDR3_A12_PP4	2	5934.263	350.9	5583.363	973.000
DDR3_A13_PP4	2	5978.499	458.96	5519.539	975.595
DDR3_A14_PP4	2	5919.606	369.6	5550.006	973.000
DDR3_BA0_PP4	2	5800.188	341.52	5458.668	973.349
DDR3_BA1_PP4	2	5933.534	442.14	5491.394	972.994
DDR3_BA2_PP4	2	5925.6	321.9	5603.7	973.147
DDR3_CKE_PP4	5	6143.903	600.99	5542.913	976.640
DDR3_CLK_N_PP4	2	6007.609	473	5534.609	976.018
DDR3_CLK_P_PP4	2	6009.657	482	5527.657	976.198
DDR3_C\A\S_PP4	2	5982.038	466.87	5515.168	976.147
DDR3_C\S_PP4	2	5772.822	547.33	5225.492	973.824
DDR3_ODT_PP4	2	5924.328	472.17	5452.158	973.000
DDR3_R\A\S_PP4	2	6039.761	561	5478.761	976.607
DDR3_W\E_PP4	2	6024.958	739.59	5285.368	973.187

1st data byte

Below is the report for the trace lengths from the first byte of the bus of the DDR interface. The traces are from the ZYNQ memory controller to the first DRAM chip.

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_DM3_BL0	2	1215.531	0	1215.531	200.011
DDR3_DQ24_BL0	2	1220.718	0	1220.718	200.936
DDR3_DQ25_BL0	2	1215.093	0	1215.093	200.011
DDR3_DQ26_BL0	2	1212.894	0	1212.894	199.601
DDR3_DQ27_BL0	2	1215.336	0	1215.336	199.995
DDR3_DQ28_BL0	2	1216.296	0	1216.296	200.159
DDR3_DQ29_BL0	2	1215.352	0	1215.352	200.011
DDR3_DQ30_BL0	2	1197.69	0	1197.69	197.000
DDR3_DQ31_BL0	2	1203.857	0	1203.857	198.038
DDR3_DQS3_N_BL0	2	1217.48	0	1217.48	200.757
DDR3_DQS3_P_BL0	2	1217.48	0	1217.48	200.761

2nd data byte

Below is the report for the trace lengths from the first byte of the bus of the DDR interface. The traces are from the ZYNQ memory controller to the second DRAM chip.

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_DM2_BL1	2	1408.545	0	1408.545	196.412
DDR3_DQ16_BL1	2	1380.298	0	1380.298	193.256
DDR3_DQ17_BL1	2	1412.844	0	1412.844	197.233
DDR3_DQ18_BL1	2	1406.908	0	1406.908	197.233
DDR3_DQ19_BL1	2	1385.464	0	1385.464	194.018
DDR3_DQ20_BL1	2	1403.89	0	1403.89	195.619
DDR3_DQ21_BL1	2	1396.116	0	1396.116	195.274
DDR3_DQ22_BL1	2	1407.685	0	1407.685	197.233
DDR3_DQ23_BL1	2	1393.21	0	1393.21	194.668
DDR3_DQS2_N_BL1	2	1478.883	0	1478.883	197.295
DDR3_DQS2_P_BL1	2	1478.875	0	1478.875	197.378

3rd data byte

Below is the report for the trace lengths from the first byte of the bus of the DDR interface. The traces are from the ZYNQ memory controller to the third DRAM chip.

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_DM1_BL2	2	1277.655	0	1277.655	210.802
DDR3_DQ8_BL2	2	1267.496	0	1267.496	208.952
DDR3_DQ9_BL2	2	1267.792	0	1267.792	209.058
DDR3_DQ10_BL2	2	1275.704	0	1275.704	210.405
DDR3_DQ11_BL2	2	1263.118	0	1263.118	208.253
DDR3_DQ12_BL2	2	1268.048	0	1268.048	209.136
DDR3_DQ13_BL2	2	1277.133	0	1277.133	210.722
DDR3_DQ14_BL2	2	1259.446	0	1259.446	207.460
DDR3_DQ15_BL2	2	1272.592	0	1272.592	209.860
DDR3_DQS1_N_BL2	2	1247.177	0	1247.177	205.890
DDR3_DQS1_P_BL2	2	1247.177	0	1247.177	205.890

4th data byte

Below is the report for the trace lengths from the first byte of the bus of the DDR interface. The traces are from the ZYNQ memory controller to the fourth DRAM chip.

Name	Node	Signal Length	Total Pin/Package	Routed	Delay (ps)
DDR3_DM0_BL3	2	1722.506	0	1722.506	245.717
DDR3_DQ0_BL3	2	1709.194	0	1709.194	243.333
DDR3_DQ1_BL3	2	1714.752	0	1714.752	244.075
DDR3_DQ2_BL3	2	1696.903	0	1696.903	241.697
DDR3_DQ3_BL3	2	1724.1	0	1724.1	245.508
DDR3_DQ4_BL3	2	1724.687	0	1724.687	245.718
DDR3_DQ5_BL3	2	1726.902	0	1726.902	245.717
DDR3_DQ6_BL3	2	1723.234	0	1723.234	245.717
DDR3_DQ7_BL3	2	1724.701	0	1724.701	245.506
DDR3_DQS0_N_BL3	2	1813.575	0	1813.575	245.718
DDR3_DQS0_P_BL3	2	1813.459	0	1813.459	245.718

7.6 Bill of materials

Below is listed the Bill of Materials of the board. Part that their “MPN” starts with GP and the “Manufacturer” field are parts like resistor and capacitors which are generic and not any special specification is needed. Then it is left to the manufacturer to place the part that they perhaps have in stock, or already placed in their pick and place machines, as a way to reduce manufacturer effort, BOM and assembly cost, because no new installation of components on the Pick and Place machines happens from the manufacturer and save manufacturing time as well.

Reference designators	Quantity	MPN	Manufacturer	Description
C89, C418	2	GPC0805106-35	Generic part	10µF ±10% 35V Ceramic Capacitor X6S 0805 (2012 Metric)
C112, C148	2	GPC0805475-25	Generic part	4.7µF ±10% 25V Ceramic Capacitor X5R 0805 (2012 Metric)
C115, C416, C417	3	GPC0402102	Generic part	0402, Cap 1nF, 50V, 10.0%, X7R
C154, C155, C156, C157, C168, C207, C208, C209, C210, C211, C212, C259, C260, C261, C273, C274, C275, C287, C288, C304, C305, C306, C324, C325, C326, C346, C347, C359, C360, C369, C370, C371, C372, C373, C387, C388, C396, C397, C421, C422, C423, C424, C425, C426, C427, C428, C429, C430, C431, C432, C433, C434, C435, C436, C437, C438, C439, C440, C441, C442	60	GPC0402103	Generic part	0402, Cap 10.0nF, 50V, 10.0%, X7R

C158, C159, C160, C161, C162, C163, C309, C310, C311, C312, C405, C406, C407, C408, C409, C410, C443, C446	18	GPC0402104	Generic part	0.1µF ±10% 50V Ceramic Capacitor X7R 0402 (1005 Metric)
C164, C165, C338, C339, C351, C352	6	GPC0402105-16	Generic part	1µF ±10% 16V Ceramic Capacitor X5R 0402 (1005 Metric)
C185, C186, C204, C205, C229, C230, C248, C249, C267, C281, C294, C316, C332, C340, C341, C342, C343, C353, C354, C355, C356, C366, C381, C382, C392, C401	26	GPC0402474-25	Generic part	0.47µF ±10% 25V Ceramic Capacitor X5R 0402 (1005 Metric)
C444, C445	2	GPC0402180	Generic part	0402, Cap 18.0pF, 50V, 5%, C0G

C5, C88, C90, C91, C92, C93, C94, C96, C97, C98, C99, C101, C102, C103, C105, C106, C107, C110, C113, C116, C117, C118, C119, C120, C122, C123, C124, C126, C127, C128, C129, C130, C132, C133, C134, C135, C138, C139, C140, C142, C143, C255, C411, C415, C419, C420, C450, C451, C452, C453	50	GPC0603104	Generic part	0603, Cap 100.0nF, 50V, 10.0%, X7R
C19, C20, C24, C25, C26, C27, C40, C48, C49, C53, C59, C60, C61, C67, C68, C72, C78, C79, C80, C151, C153	21	GPC0603104	Generic part	0603, Cap 100.0nF, 50V, 10.0%, X7R
C6	1	GPC0603151	Generic part	0603, Cap 150.0pF, 50V, 10.0%, C0G
C7	1	GPC0603153	Generic part	0603, Cap 15.0nF, 50V, 10.0%, X7R
C8, C13	2	GPC0603224	Generic part	0.22µF ±10% 50V Ceramic Capacitor X7R 0603 (1608 Metric)
C152	1	GPC0603224	Generic part	0.22µF ±10% 50V Ceramic Capacitor X7R 0603 (1608 Metric)

C9, C14, C146, C251, C252, C412	6	GPC0603105	Generic part	0603, Cap 1uF, 50V, 10.0%, X5R
R55, R60, R61, R62, R63, R64, R90, R91, R92, R93, R94, R95, R96, R97, R98, R99, R100, R101, R233, R234, R235, R236, R237, R238, R239, R240, R241, R242, R243, R244, R248, R249, R253, R259, R260	35	GPR06034K7	Generic part	0603, Res 4.7kOhm, 50V, 1.0%, 100mW
R8	1	GPR0603470R	Generic part	0603, Res 470.0Ohm, 75V, 1.0%, 100mW
R9, R86, R87, R88	4	GPR0603100R	Generic part	0603, Res 100.0Ohm, 50V, 1.0%, 100mW
R10	1	GPR06039K1	Generic part	0603, Res 9.1kOhm, 75V, 1.0%, 100mW

R11, R20, R27, R34, R35, R43, R44, R72, R74, R103, R164, R165, R166, R167, R168, R169, R170, R171, R172, R173, R174, R175, R176, R177, R178, R179, R180, R181, R182, R183, R184, R185, R186, R187, R188, R189, R190, R191, R192, R193, R194, R195, R225, R226, R229, R230, R231, R232	48	GPR06030R	Generic part	0 Ohms Jumper 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Moisture Resistant Thick Film
R14, R17, R33, R40, R47, R52	6	GPR0603150R	Generic part	0603, Res 150.0Ohm, 50V, 1.0%, 100mW
R15, R18, R36, R41, R48, R53	6	GPR060318K	Generic part	0603, Res 18.0kOhm, 50V, 1.0%, 100mW
R24	1	GPR08051R	Generic part	0805, Res 1.0Ohm, 150V, 1.0%, 125mW
R25, R77, R78, R80, R81, R82, R83, R84	8	GPR060320K	Generic part	0603, Res 20.0kOhm, 75V, 1.0%, 100mW
R57, R58	2	GPR060351R	Generic part	0603, Res 51.0Ohm, 75V, 1.0%, 100mW
R59, R89	2	GPR060351R	Generic part	0603, Res 51.0Ohm, 75V, 1.0%, 100mW

R65	1	GPR06031M	Generic part	0603, Res 1.0MOhm, 50V, 1.0%, 100mW
R68, R73, R123, R131	4	GPR06031K	Generic part	0603, Res 1.0kOhm, 50V, 1.0%, 100mW
R70, R71, R104, R124, R125, R126, R127, R128	8	GPR0603390R	Generic part	0603, Res 390.0Ohm, 50V, 1.0%, 100mW
R75, R76, R79, R102	4	GPR060324R	Generic part	0603, Res 24.0Ohm, 50V, 1.0%, 100mW
R85	1	GPR0603330R	Generic part	0603, Res 330.0Ohm, 50V, 1.0%, 100mW
R110, R113, R116, R121	4	GPR0603100K	Generic part	0603, Res 100.0kOhm, 50V, 1.0%, 100mW
R263	1	GPR0603100K	Generic part	0603, Res 100.0kOhm, 50V, 1.0%, 100mW
R111	1	GPR060343K	Generic part	0603, Res 43.0kOhm, 75V, 1.0%, 100mW
R117	1	GPR060333K	Generic part	0603, Res 33.0kOhm, 75V, 1.0%, 100mW
R118	1	GPR060315K	Generic part	0603, Res 15.0kOhm, 75V, 1.0%, 100mW
R122	1	GPR060324K	Generic part	0603, Res 24.0kOhm, 50V, 1.0%, 100mW

R132, R133, R134, R135, R136, R137, R138, R139, R140, R141, R142, R143, R144, R145, R146, R147, R148, R149, R150, R151, R152, R153, R154, R155, R156, R157, R158, R159, R160, R161, R162, R163	32	GPR0603200R	Generic part	0603, Res 200.0Ohm, 50V, 1.0%, 100mW
R196, R197, R198, R199, R200, R201, R202, R203, R204, R205, R206, R207, R208, R209, R210, R211, R212, R213, R214, R215, R216, R217, R218, R282	24	GPR060347R	Generic part	0603, Res 47.0Ohm, 50V, 1.0%, 100mW
R219	1	GPR060382R	Generic part	0603, Res 82.0Ohm, 75V, 1.0%, 100mW
R220, R221, R222, R223	4	GPR0603240R	Generic part	0603, Res 240.0Ohm, 50V, 1.0%, 100mW
R283	1	GPR08054K7	Generic part	0805, Res 4.7kOhm, 150V, 1.0%, 125mW
J1	1	39-30-1060	Molex	6 Positions Header Connector 0.165" (4.20mm) Through Hole, Right Angle Tin

IC1	1	88E1518-A0-NNB2C000	Marvell	Ethernet-ICs Single-port Gigabit Ethernet PHY with EEE; RGMII; 1.8V IO only in 48-pin QFN package
D2	1	LTST-C171KSKT	Lite-On	Yellow 587nm LED Indication - Discrete 2V 0805 (2012 Metric)
J3, J5	2	74754-0101	Molex	Position SFP+ Cage Connector Press-Fit Through Hole, Right Angle
J24	1	87831-1420	Molex	14 Positions Header, Shrouded Connector 0.079" (2.00mm) Through Hole Gold
J8, J9, J10, J11, J12, J13, J14, J15, J16, J17, J18, J19, J45, J46, J47, J48	16	SMA-J-P-H-ST-MT1	Samtec Inc.	SMA Connector Jack, Female Socket 50Ohm Surface Mount, Through Hole Solder
SW2, SW3, SW4, SW5, SW6, SW7	6	1825910-6	TE Connectivity	Tactile Switch SPST-NO Top Actuated Through Hole
J7	1	2041021-4	TE Connectivity	11 (9 + 2) Position Card Connector Secure Digital - SD Surface Mount, Right Angle Gold
J51	1	5103308-5	TE Connectivity	20 Positions Header Connector 0.100" (2.54mm) Through Hole Gold or Gold-Palladium
J4, J6	2	74441-0010	Molex	20 Position SFP+ Receptacle Connector Solder Surface Mount, Right Angle
X1	1	ABM3B-24.000MHZ-B2-T	Abracon	24MHz ±20ppm Crystal 18pF 50 Ohm -20°C ~ 70°C Surface Mount 4-SMD, No Lead

Y1	1	ABM8G-25.000MHZ-18-D2Y-T	Abracon	25MHz ±20ppm Crystal 18pF 60 Ohm -40°C ~ 85°C Surface Mount 4-SMD, No Lead
Y4	1	ABM8G-30.000MHZ-18-D2Y-T	Abracon	30MHz ±20ppm Crystal 18pF 50 Ohms 4-SMD, No Lead
FB2, FB3, FB4	3	BKP2125HS221-T	Taiyo Yuden	220 Ohms @ 100MHz 1 Power Line Ferrite Bead 0805 (2012 Metric) 2A 50mOhm
D8	1	BZX84-A3V0,215	Nexperia USA Inc.	Diodes - Zener 3V 250mW ± 1% Surface Mount TO-236AB
C21, C50, C69	3	CL32A107MPVNNNE	Samsung	100µF ±20% 10V Ceramic Capacitor X5R 1210 (3225 Metric)
C147, C272, C286, C297, C321, C322, C323, C337, C350, C363, C364, C386, C394, C402	14	CL32A107MPVNNNE	Samsung	100µF ±20% 10V Ceramic Capacitor X5R 1210 (3225 Metric)
C1, C2, C3, C4, C22, C23, C28, C29, C33, C34, C39, C41, C42, C51, C52, C54, C55, C62, C63, C70, C71, C73, C74, C81, C82, C121, C125	27	CL32A226KAJNNNE	Samsung	22µF ±10% 25V Ceramic Capacitor X5R 1210 (3225 Metric)
C15, C16, C17, C18	4	GRM32ER61C476KE15L	Murata	47µF ±10% 16V Ceramic Capacitor X5R 1210 (3225 Metric)
C114	1	GRM32ER61A107ME20L	Murata	100µF ±20% 10V Ceramic Capacitor X5R 1210 (3225 Metric)

C413	1	GRM32ER61A107ME20L	Murata	100µF ±20% 10V Ceramic Capacitor X5R 1210 (3225 Metric)
C166, C167	2	EMK105ABJ225KV-F	Taiyo Yuden	2.2µF ±10% 16V Ceramic Capacitor X5R 0402 (1005 Metric)

C169, C170, C171, C172, C173, C174, C175, C176, C177, C178, C179, C180, C181, C182, C183, C184, C188, C189, C190, C191, C192, C193, C194, C195, C196, C197, C198, C199, C200, C201, C202, C203, C213, C214, C215, C216, C217, C218, C219, C220, C221, C222, C223, C224, C225, C226, C227, C228, C232, C233, C234, C235, C236, C237, C238, C239, C240, C241, C242, C243, C244, C245, C246, C247, C262, C263, C264, C265, C266, C276, C277, C278, C279, C280, C289, C290, C291, C292, C293, C299, C300, C301, C302, C303,	112	GRM155R71A473KA01D	Murata	SMD Multilayer Ceramic Capacitor, 0402 [1005 Metric], 0.047 µF, 10 V, ± 10%, X7R, GRM Series

C307, C308, C313, C314, C327, C328, C329, C330, C331, C344, C345, C357, C358, C367, C368, C374, C375, C376, C377, C378, C379, C380, C389, C390, C391, C398, C399, C400				
C10, C414	2			
C11	1	GPC0603331	Generic part	0603, Cap 330pF, 50V, 5.0%, C0G

C12	1	GPC0603223	Generic part	0603, Cap 22.0nF, 50V, 10.0%, X7R
C44, C47	2	GPC0603223	Generic part	0603, Cap 22.0nF, 50V, 10.0%, X7R
C31, C36, C57, C65, C76, C84	6	GPC0603100	Generic part	0603, Cap 10.0pF, 50V, 5.0%, C0G
C32, C37, C58, C66, C77, C85	6	GPC0603471	Generic part	0603, Cap 470pF, 50V, 5.0%, C0G
C38, C46, C145, C149	4	GPC0603102	Generic part	0603, Cap 1nF, 50V, 10.0%, X7R
C43	1	GPC0603101	Generic part	0603, Cap 100.0pF, 50V, 5.0%, C0G
C45	1	GPC0603332	Generic part	0603, Cap 3.3nF, 50V, 10.0%, X7R
C95, C100, C131, C136, C137, C141, C187, C206, C231, C250, C268, C269, C270, C271, C282, C283, C284, C285, C295, C296, C298, C315, C317, C318, C319, C320, C333, C334, C335, C336, C348, C349, C361, C362, C365, C383, C384, C385, C393, C395, C403, C404	42	GPC0603475-10	Generic part	4.7µF ±10% 10V Ceramic Capacitor X5R 0603 (1608 Metric)
C111	1	GPC0603475-10	Generic part	4.7µF ±10% 10V Ceramic Capacitor X5R 0603 (1608 Metric)
C104, C144, C257, C258	4	GPC0603103	Generic part	0603, Cap 10.0nF, 50V, 10.0%, X7R

C108, C109	2	GPC0603180	Generic part	0603, Cap 18.0pF, 50V, 5.0%, C0G
C150	1	GPC0603334	Generic part	0603, Cap 330nF, 50V, 10.0%, X7R
C253, C254, C256	3	GPC0603106-10	Generic part	10 uF +/- 10% 10 V X5R 0603 Multilayer Ceramic Capacitors MLCC - SMD/SMT T&R GP
R250, R251, R255, R256, R257, R258	6	GPR0402100R	Generic part	0402, Res 100.0Ohm, 50V, 1.0%, 62.5mW
R267	1	GPR0402100R	Generic part	0402, Res 100.0Ohm, 50V, 1.0%, 62.5mW
R264, R269, R270, R271, R273	5	GPR040210K	Generic part	0402, Res 10.0kOhm, 50V, 1.0%, 62.5mW
R265, R266	2	GPR040282R	Generic part	0402, Res 82.0Ohm, 50V, 1.0%, 62.5mW
R268	1	GPR04021K6	Generic part	0402, Res 1.6kOhm, 50V, 1.0%, 62.5mW
R272	1	GPR040220K	Generic part	0402, Res 20.0kOhm, 50V, 1.0%, 62.5mW
C30, C35, C56, C64, C75, C83	6	GRM1885C1H751JA01D	Murata	750pF ±5% 50V Ceramic Capacitor C0G, NP0 0603 (1608 Metric)
C86, C87	2	MC0603N390J500CT	Multicomp	SMD Multilayer Ceramic Capacitor, 0603 [1608 Metric], 39 pF, 50 V, ± 5%, C0G / NP0, MC Series
J42	1	48393-0003	Molex	USB - A USB 3.0 (USB 3.1 Gen 1, Superspeed) Receptacle Connector 9 Position Through Hole, Right Angle

J20, J22	2	1981584-1	TE Connectivity	USB - micro AB USB 2.0 Receptacle Connector 5 Position Surface Mount, Right Angle
J21, J23, J49, J50	4	61300311121	Würth Elektronik	3 Positions Header Connector 0.100" (2.54mm) Through Hole Gold
U14	1	CY7C64225-28PVXC	Cypress Semiconductor	USB Bridge, USB to UART USB 2.0 USB Interface 28-SSOP
FB1, FB11, FB12, FB13	4	MPZ2012S601AT000	TDK	Ferrite Beads 600ohms 2A 100mOhms 0805 Ferrite Chip
L9, L10, L11, L12	4	IHLP2020BZER4R7M11	Vishay	4.7µH Shielded Molded Inductor 3.2A 81.3 mOhm Max Nonstandard
Q3	1	FDV301N	Fairchild Semiconductor	IC, Digital FET, N-channel
U17	1	FT601Q-T	FTDI	USB Bridge, USB to FIFO USB 3.0 FIFO Interface 76-QFN (9x9)
P1, P2, P3, P4, P7, P8	6			
P6	1			
P5	1			
D3, D4	2	HSMY-C170	Avago	Yellow 586nm LED Indication - Discrete 2.1V
U19	1	PCA9548APWR	Texas Instruments	IC I2C SW 8CH W/RESET Translating Switch Interface 24-TSSOP
L1	1	IHLP2020CZER1R5M01	Vishay	1.5µH Shielded Molded Inductor 7.2A 20.7mOhm Max Nonstandard
L2, L3, L4, L5, L6, L7, L8	7	SRN6045-1R0Y	Bourns	1µH Semi-Shielded Wirewound Inductor 4.2A 13.9 mOhm Max Nonstandard

U20	1	LM339MX/NOPB	Texas Instruments	Comparator General Purpose CMOS, DTL, ECL, MOS, Open-Collector, TTL 14-SOIC
U21	1	MAX1510ETB+T	Maxim Integrated	Converter, DDR Voltage Regulator IC 1 Output 10-TDFN-EP (3x3)
U22	1	MAX1983EUT+T	Maxim Integrated	Linear Voltage Regulator IC Positive Adjustable 1 Output 0.8 V ~ 2 V 300mA SOT-23-6
U23	1	MAX6037BAUK12-T	Maxim Integrated	Series Voltage Reference IC Â±0.3% SOT-23-5
U6	1	MAX8686ETL+	Maxim Integrated	Buck Switching Regulator IC Positive Adjustable 0.7V 1 Output 25A 40-WFQFN Exposed Pad
U7, U9, U10	3	MAX15021ATI+T	Maxim Integrated	Buck Switching Regulator IC Positive Adjustable 0.6V 2 Output 2A, 4A 28-WFQFN Exposed Pad
U8	1	MAX15053EWL+T	Maxim Integrated	Buck Switching Regulator IC Positive Adjustable 0.6V 1 Output 2A 9-WFBGA, WLBGA
U16	1	MIC2025-1YM-TR	Microchip Technology	Power switch / driver 1: 1 N-channel 500mA 8-SOIC
U2, U3, U4, U5	4	MT41K256M8DA-125:K	Micron	DRAM DDR3 2G 256MX8 FBGA
Q1, Q2	2	NDS331N	Fairchild Semiconductor	Transistor, N-Channel Logic Level Enhancement Mode FET, SMD
J25, J26, J27, J28, J29, J30, J31, J40, J41	9			

J32, J34, J35, J36, J37, J39	6			
J33, J38	2			
R1	1	GPR060310R	Generic part	0603, Res 10.0Ohm, 50V, 1.0%, 100mW
R4, R5, R105, R106, R107, R108, R109, R129, R130, R254	10	GPR060310K	Generic part	0603, Res 10.0kOhm, 50V, 1.0%, 100mW
R67, R119	2	GPR060310K	Generic part	0603, Res 10.0kOhm, 50V, 1.0%, 100mW
R7	1	GPR06034K7	Generic part	0603, Res 4.7kOhm, 50V, 1.0%, 100mW
D1, D9, D10, D25	4	RB751S40T1G	ON Semiconductor	Diode Schottky 30V 30mA (DC) Surface Mount SOD-523
R2, R6	2	RC0603FR-07182KL	Yageo	182k Ohm ±1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Moisture Resistant Thick Film
R3	1	RK73H1JTTD4531F	KOA Speer Electronics	4.53 kOhms ±1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200, Moisture Resistant Thick Film
R12, R16, R32, R39, R42, R46, R51, R114, R115	9	RT0603BRD0710KL	Yageo	10 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Thin Film
R13, R19	2	ERA-3AEB153V	Panasonic	15 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film

R28	1	ERA-3AEB8061V	Panasonic	8.06 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film
R66	1	ERA-3AEB8061V	Panasonic	8.06 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film
R29	1	AC0603FR-072K32L	Yageo	2.32 kOhms ±1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200, Moisture Resistant Thick Film
R30	1	ERA-3AEB4021V	Panasonic	Resistor Thin Film 4.02K Ohm 0.1% 25ppm 1/10W
R37	1	ERA3AEB222V	Panasonic	2.2k Ohm ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film
R49	1	ERA3AEB6651V	Panasonic Electronic Components	6.65 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film
R54	1	ERA-3AEB4991V	Panasonic	4.99 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film
R56	1	ERJ-PB3B4991V	Panasonic Electronic Components	4.99 kOhms ±0.1% 0.2W, 1/5W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200, Pulse Withstanding Thick Film
R112	1	ERA3AEB4020V	Panasonic Electronic Components	402 Ohms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film

R120	1	CRCW06039K76FKEA	Vishay	9.76 kOhms ±1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thick Film
R227	1	6-2176089-3	TE Connectivity Passive Product	49.9 kOhms ±0.1% 0.167W, 1/6W Chip Resistor 0603 (1608 Metric) Thin Film
R228	1	PAT0603E4002BST1	Vishay	40 kOhms ±0.1% 0.15W Chip Resistor 0603 (1608 Metric) Anti-Sulfur, Automotive AEC-Q200, Moisture Resistant Thin Film
R245, R246, R247	3			
R69, R252	2	GPR0603270R	Generic part	0603, Res 270.0Ohm, 75V, 1.0%, 100mW
R274, R275, R276	3	GPR060316K	Generic part	0603, Res 16.0kOhm, 50V, 1.0%, 100mW
R224	1	GPR08050R	Generic part	0 Ohms Jumper 0.125W, 1/8W Chip Resistor 0805 (2012 Metric) Moisture Resistant Thick Film
R261	1	ERA-3AEB8661V	Panasonic	8.66 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Automotive AEC-Q200 Thin Film
R262	1	RT0603BRE074K02L	Yageo	4.02 kOhms ±0.1% 0.1W, 1/10W Chip Resistor 0603 (1608 Metric) Moisture Resistant Thin Film
R21, R22, R23, R26, R31, R38, R45, R50, R277	9	RL1210FR-070R01L	Yageo	10 mOhms ±1% 0.5W, 1/2W Chip Resistor 1210 (3225 Metric) Automotive AEC-Q200, Current Sense, Moisture Resistant Thick Film

R278, R279, R280	3	CRF0805-FZ-R010ELF	Bourns	0.01 Ohm ±1% 0.5W, 1/2W Chip Resistor 0805 (2012 Metric) Automotive AEC-Q200, Current Sense, Moisture Resistant Metal Film
U18	1	S25FL256SAGMFI001	Cypress Semiconductor Corp	NOR Flash 256Mb 3V 133MHz Serial NOR Flash
D5, D6, D16, D17, D18, D19, D20, D21, D22, D23, D26, D27, D28, D29, D30, D31, D32, D33	18	SESD0802Q4UG-0020-090	Littelfuse	ESD Suppressors / TVS Diodes 4-CH MINI 9V Uni-Di .20pF 20kV SESD
U12	1	570BAB000544DGR	Silicon Labs	156.25MHz XO (Standard) LVDS Oscillator 3.3V Enable/Disable (Reprogrammable) 8-SMD, No Lead
J2	1	SI-51009-F	Stewart Connector	1 Port RJ45 Magjack Connector Through Hole 10/100/1000 Base-T, AutoMDIX
Y2	1	FO5HSCBE33.333-T1	Fox Electronics	33.333MHz XO (Standard) HCMOS Oscillator 3.3V Standby (Power Down) 4-SMD, No Lead
Y3	1	SIT9120AI-2C2-33E100.000000X	SiTIME	100MHz XO (Standard) LVDS Oscillator 3.3V Enable/Disable 6-SMD, No Lead
D24	1	LTST-C193TBKT-5A	Lite-On	Blue 470nm LED Indication - Discrete 2.8V 0603 (1608 Metric)
D7, D11, D12, D13, D14, D15	6	SML-LX0603GW-TR	Lumax	Green LED Indication - Discrete 2.2V 0603 (1608 Metric)
U24	1	SN74LVC1G11DBVR	Texas Instruments	AND Gate IC 1 Channel SOT-23-6

SW1	1	MS12ANW03	NKK Switches	Slide Switch SPDT Through Hole
J43, J44	2			
U15	1	TXS0102DCTR	Texas Instruments	IC, 2-Bit Bidirectional Voltage-Level Translator for Open-Drain and Push-Pull
U11	1	TXS02612RTWR	Texas Instruments	Cell Phone Interface 24-WQFN (4x4)
U13	1	USB3320C-EZK-TR	Microchip Technology	1/1 Transceiver Half USB 2.0 32-QFN (5x5)
U1	1	XC7Z015-2CLG485I	Xilinx Inc	Dual ARM® Cortex®-A9 MPCore™ with CoreSight™ System On Chip (SOC) IC Zynq®-7000 Artix™-7 FPGA, 74K Logic Cells 766MHz 485-CSBGA (19x19)
FB5, FB6	2	MPZ1608S221ATA00	TDK	FERRITE BEAD 220 OHM 0603 1LN

7.7 Power supply measurement data

The power supply was probed with an oscilloscope to observe the noise on the rails. Two measurements were performed, one with the board idling, the processor not instantiated and the FPGA not programmed, and one with the processor running Linux and running memory tests, while the FPGA was transmitting and receiving on all MGT channels, and the PL was programmed and running intensive tasks as well. The rails were probed close to the consumer device, and a low inductance probe was used, basically eliminating the ground lead of the probe, which would not allow for measurements such low in the mV range. In general, the power supply showed very good performance, with noise on most rails being no more than 5mV P-P.

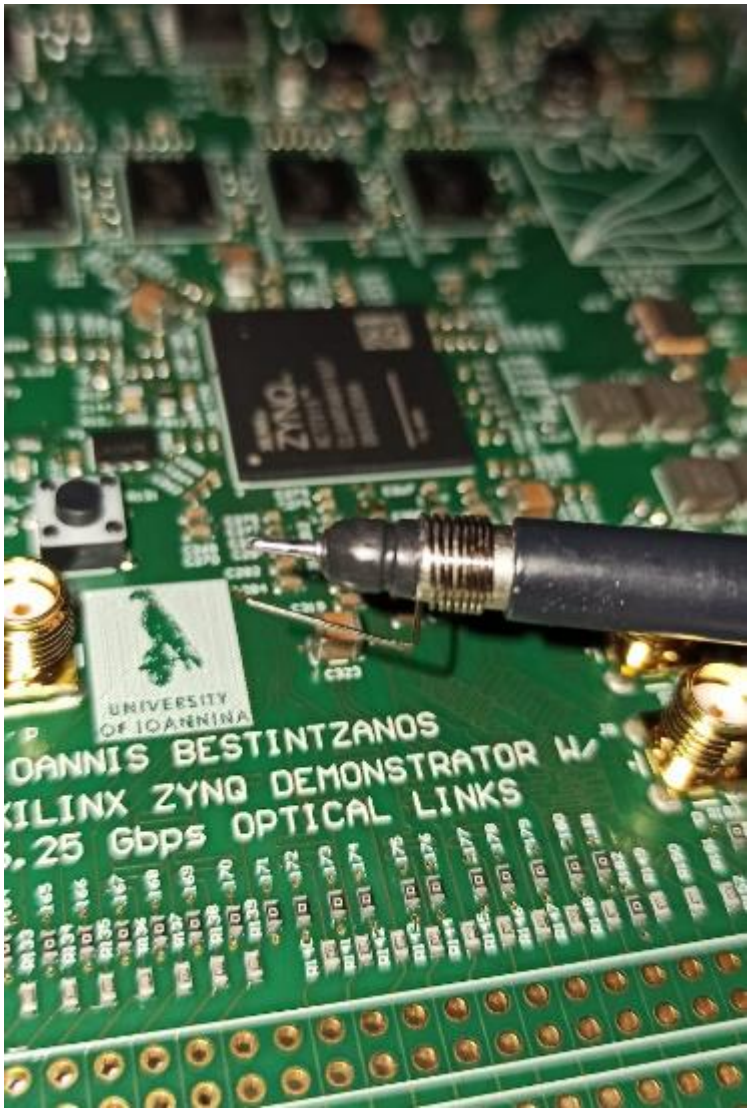
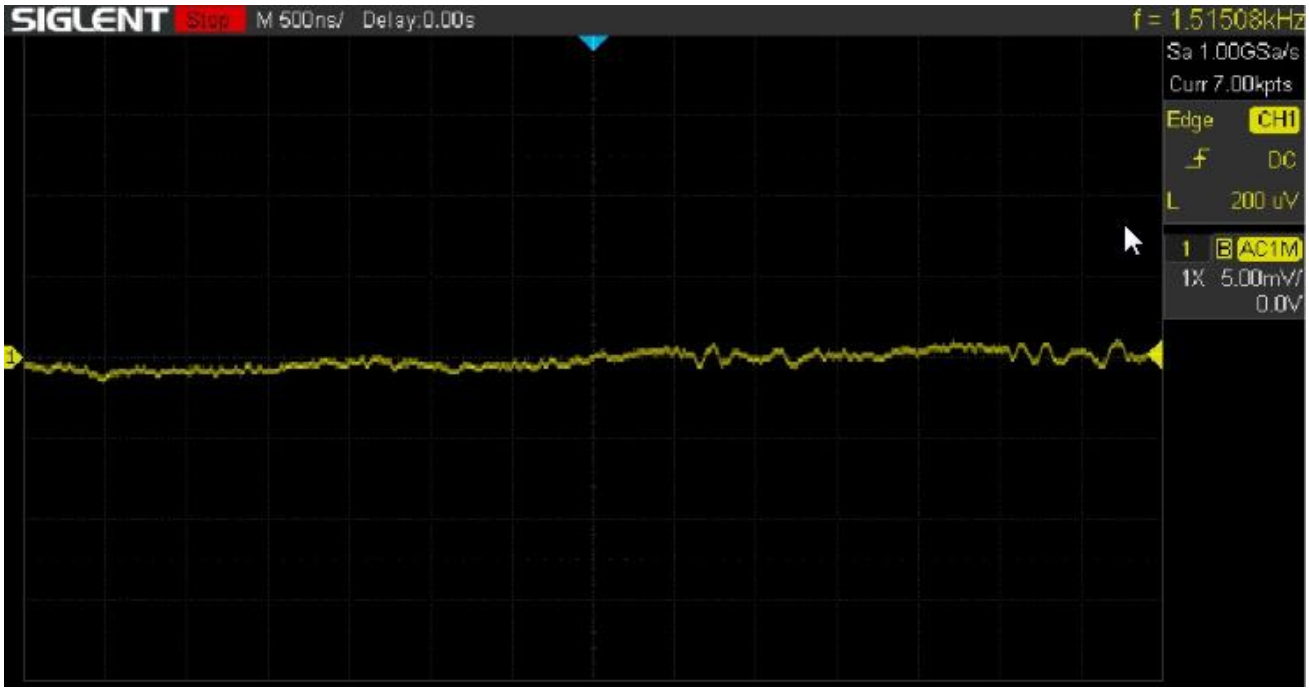


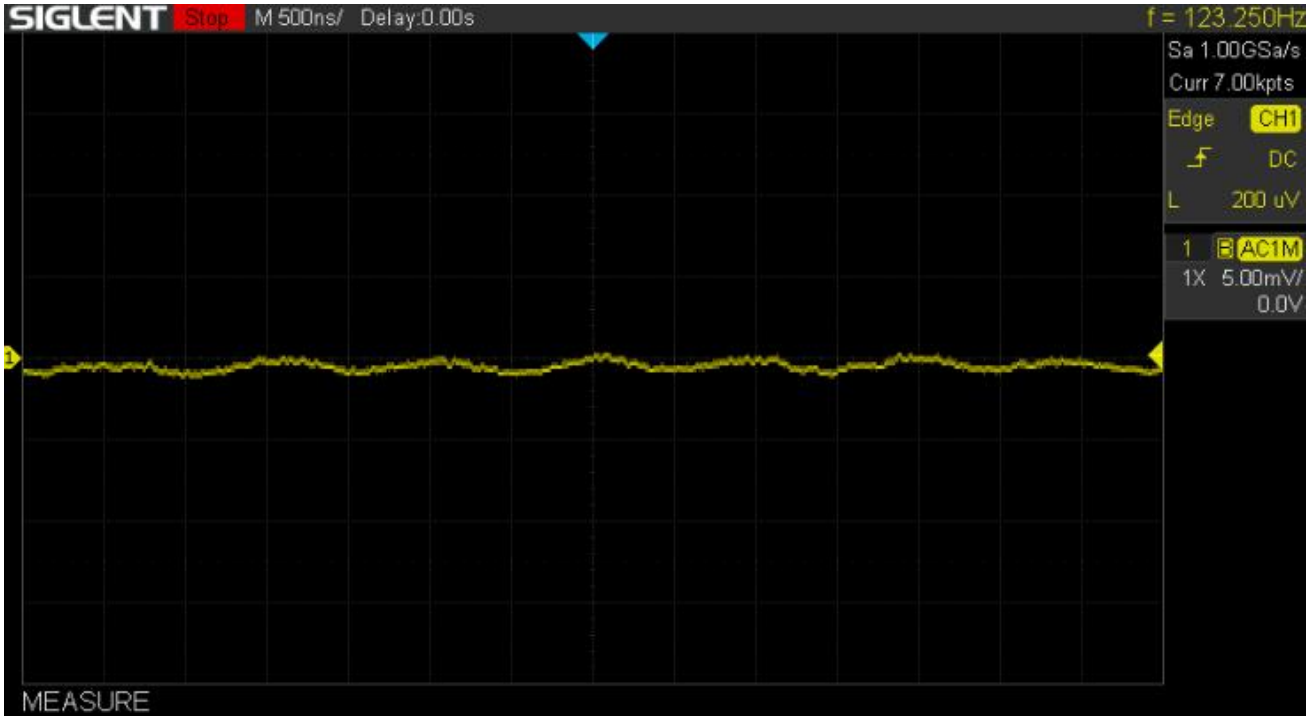
Figure 110. The low inductance ground lead used to probe the power rails.

Idle measurements

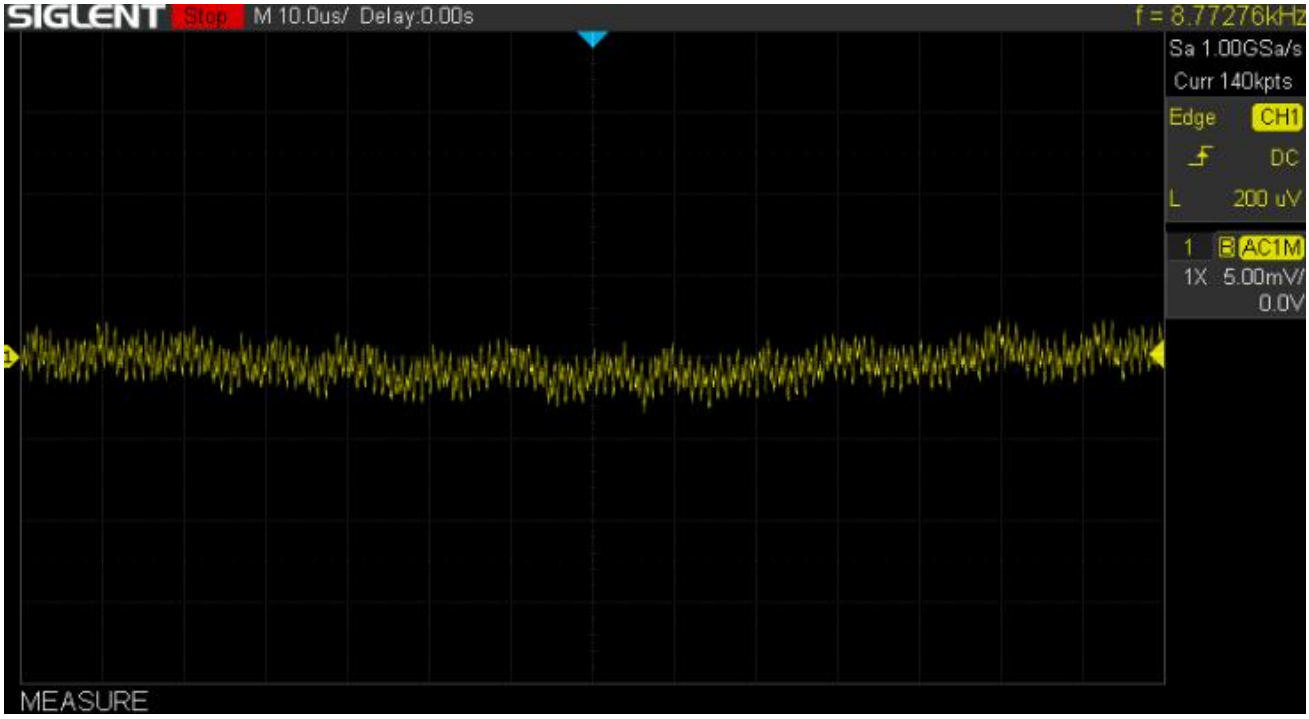
VCCINT

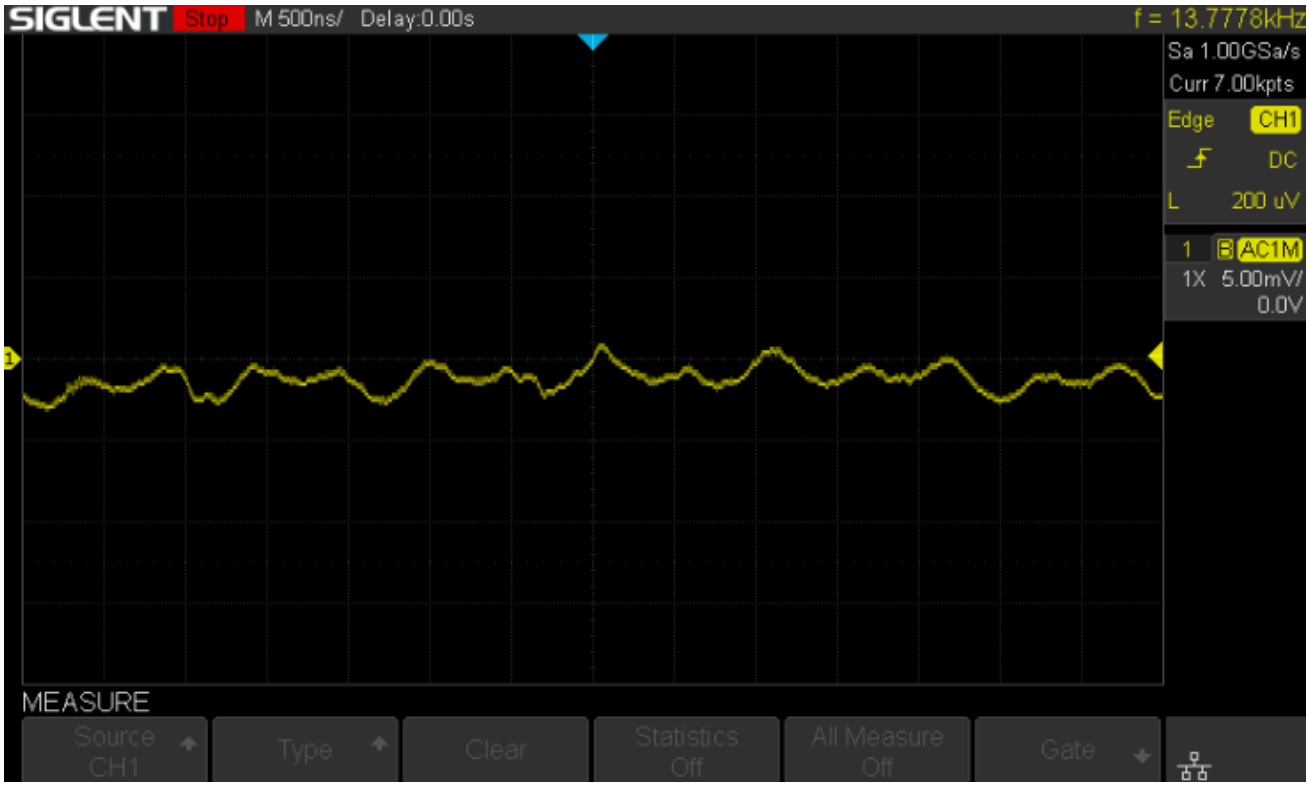


VCC1V8

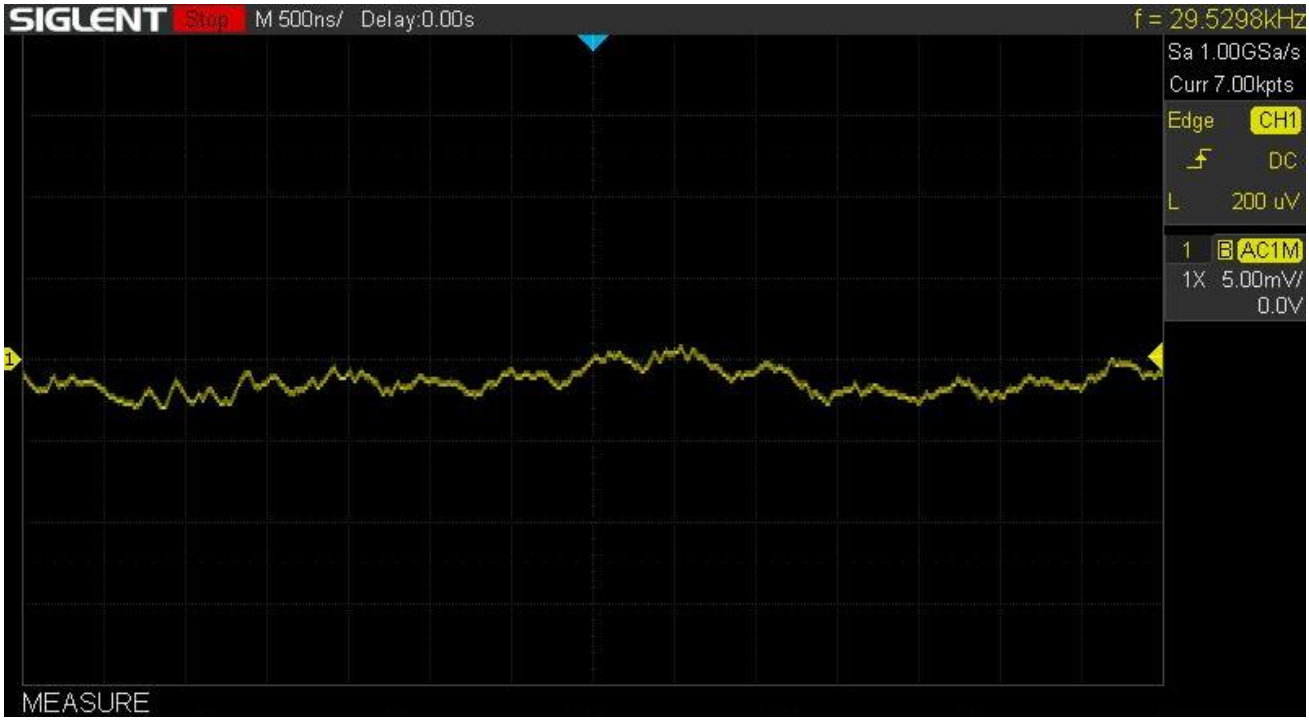

SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 123.250Hz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L
200 uV
1
B
AC1M
1X
5.00mV/
0.0V
MEASURE

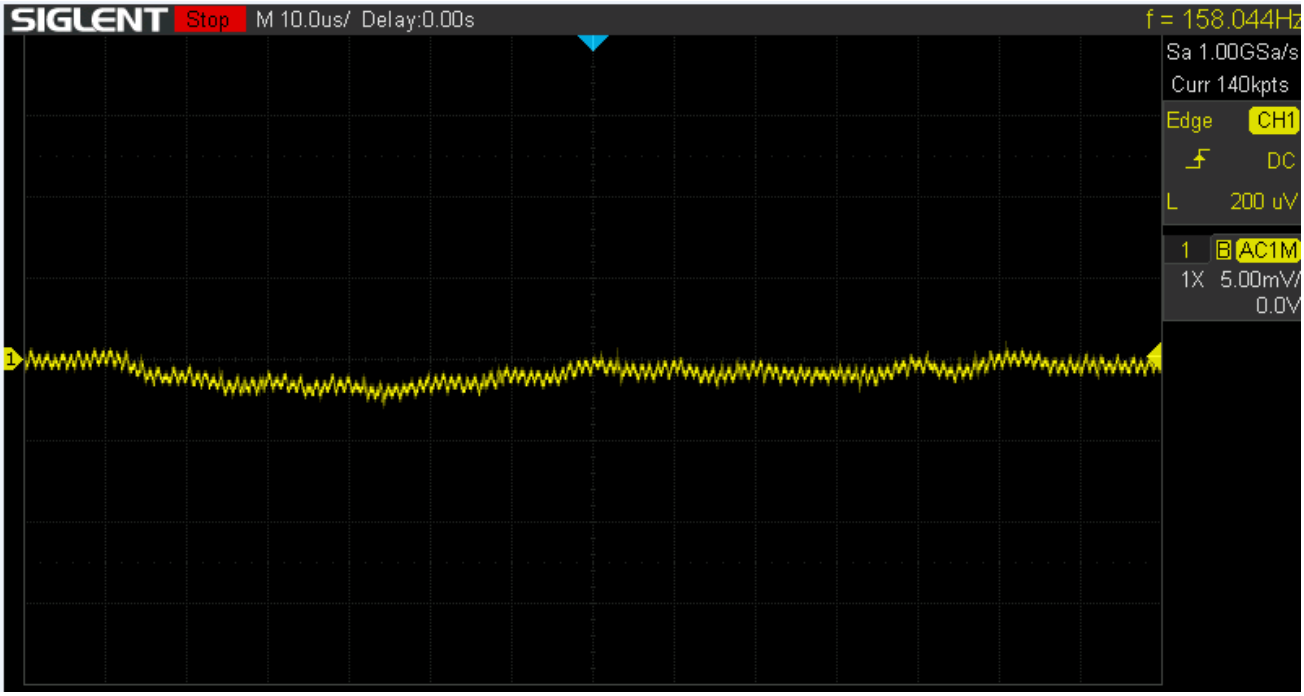
3V3


SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 8.77276kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L
200 uV
1
B
AC1M
1X
5.00mV/
0.0V
MEASURE

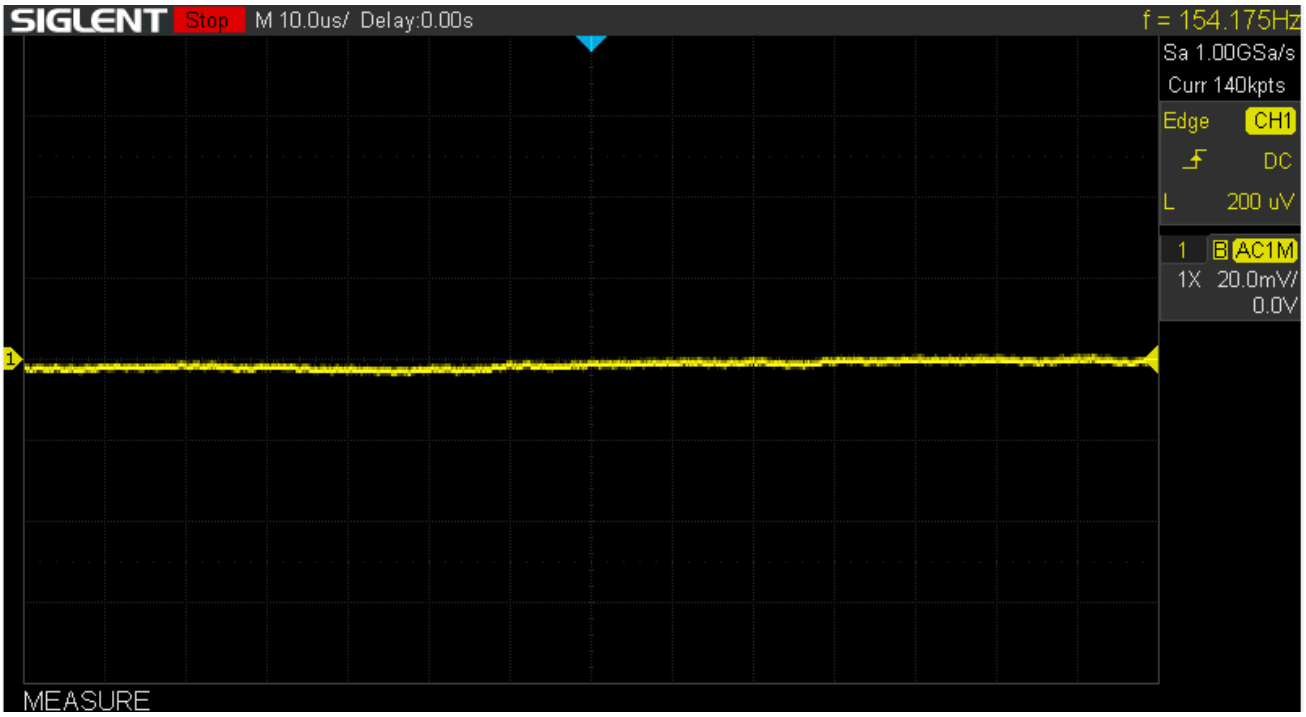

SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 13.7778kHz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L
200 uV
1
B
AC1M
1X
5.00mV/
0.0V
MEASURE
Source
CH1
Type
Clear
Statistics
Off
All Measure
Off
Gate

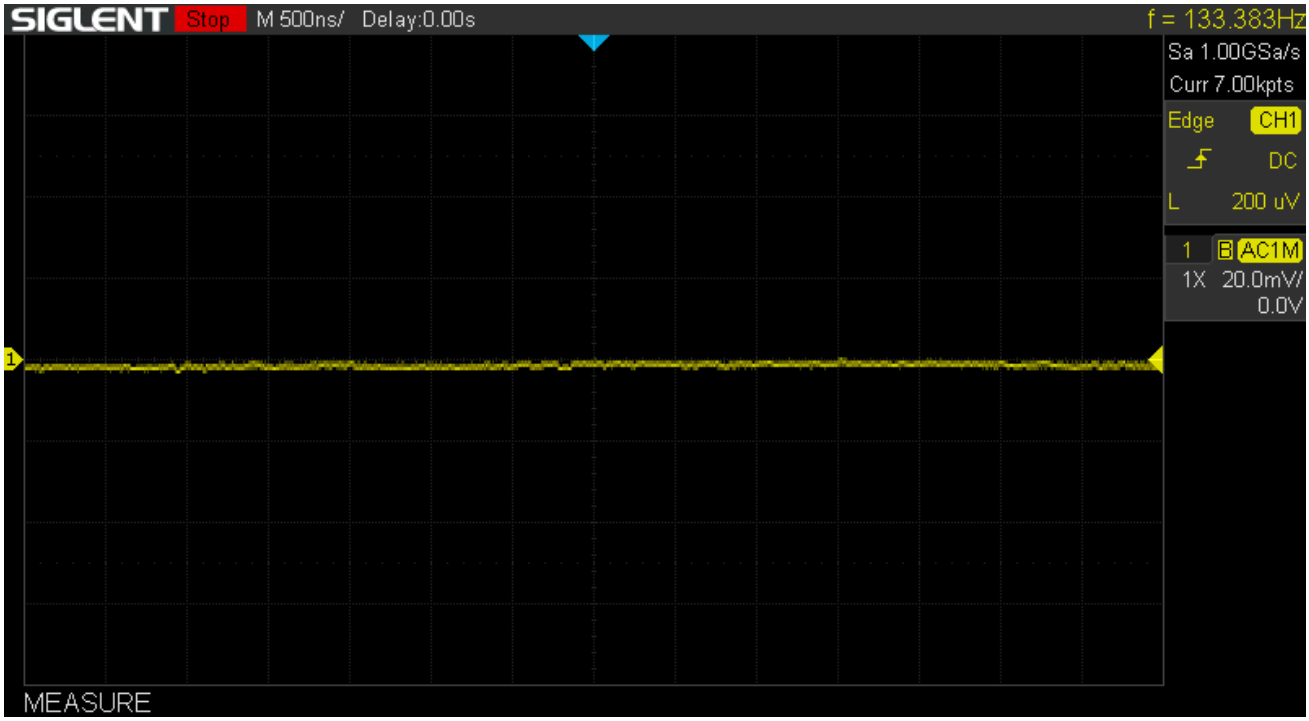
VCCAUX

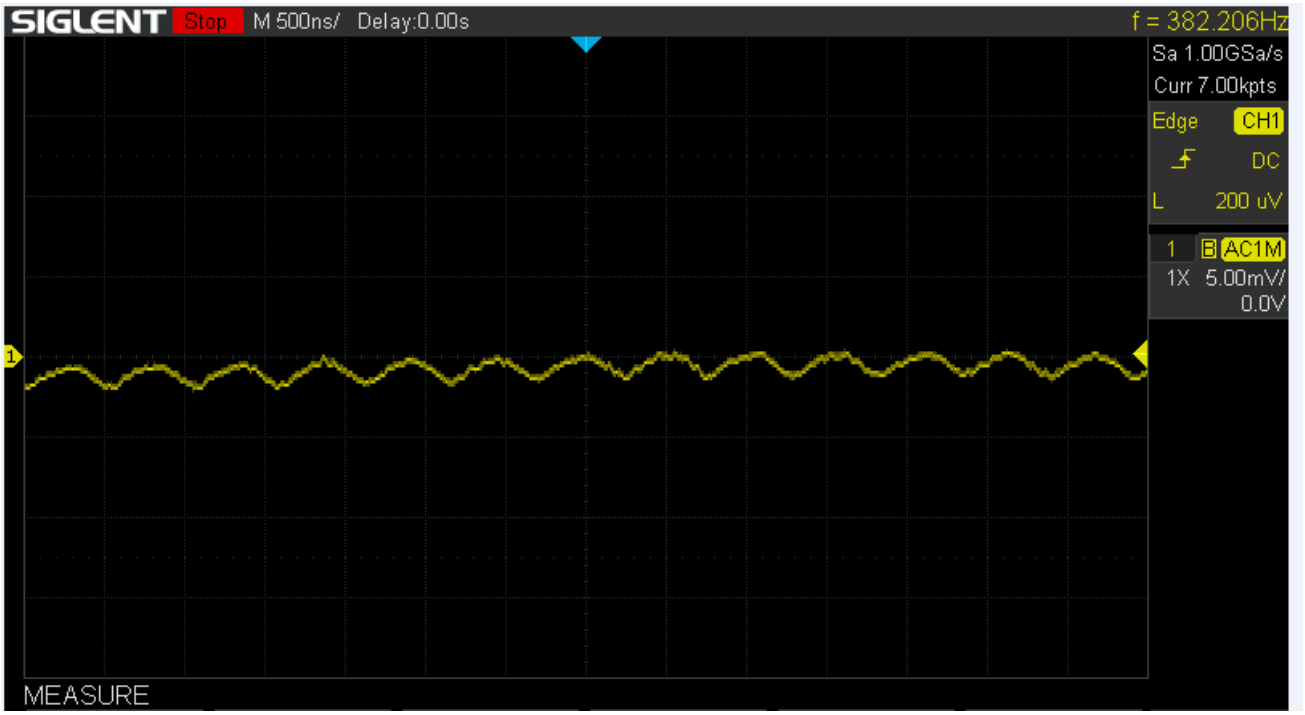

SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 29.5298kHz
Sa 1.00GSa/s
Curr 7.00kpts
Edge CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
MEASURE

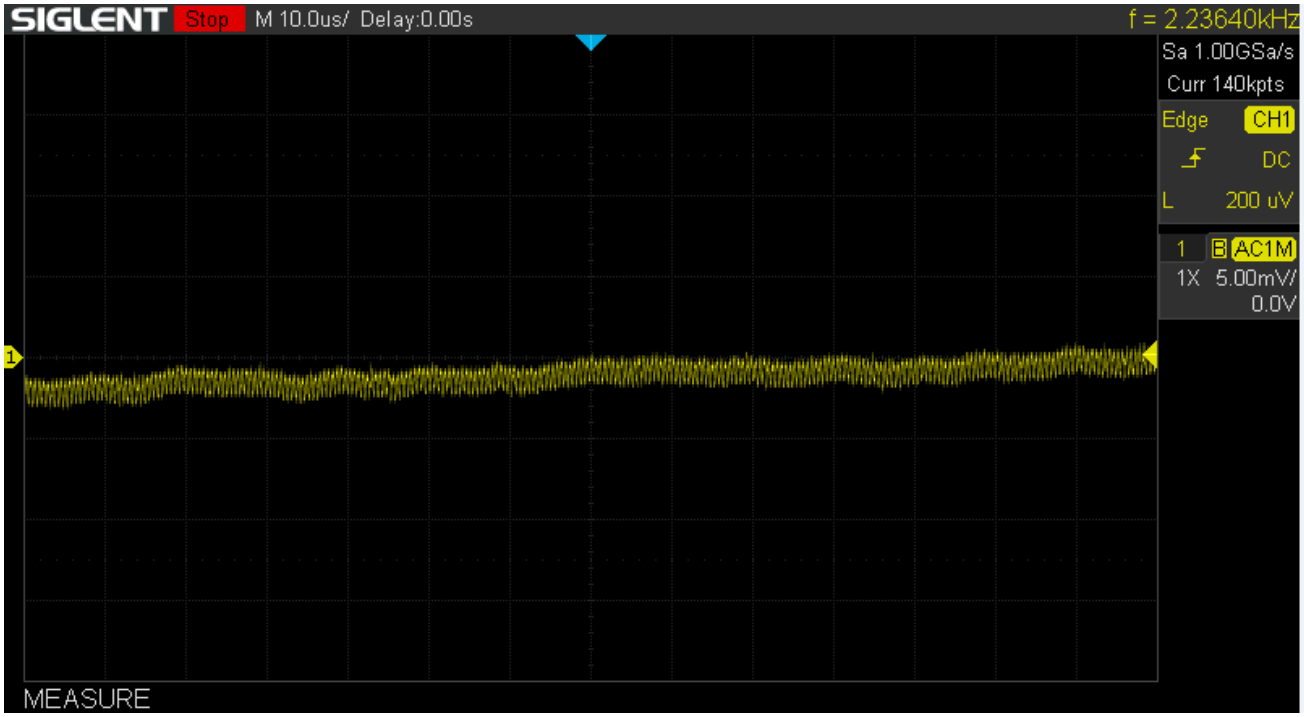

SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 158.044Hz
Sa 1.00GSa/s
Curr 140kpts
Edge CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V

MGTAVCC


SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 154.175Hz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L
200 uV
1
B
AC1M
1X
20.0mV/
0.0V
MEASURE

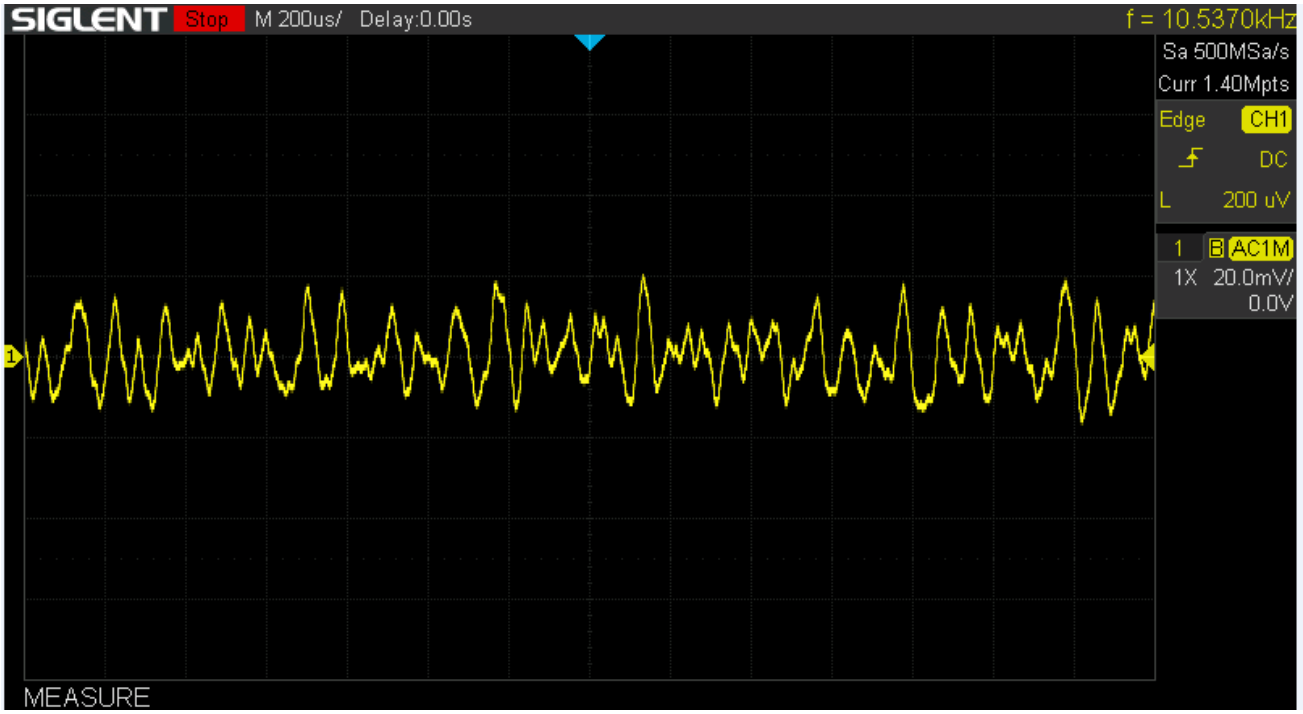

SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 133.383Hz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L
200 uV
1
B
AC1M
1X
20.0mV/
0.0V
MEASURE


SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 382.206Hz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
MEASURE


SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 2.23640kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
MEASURE

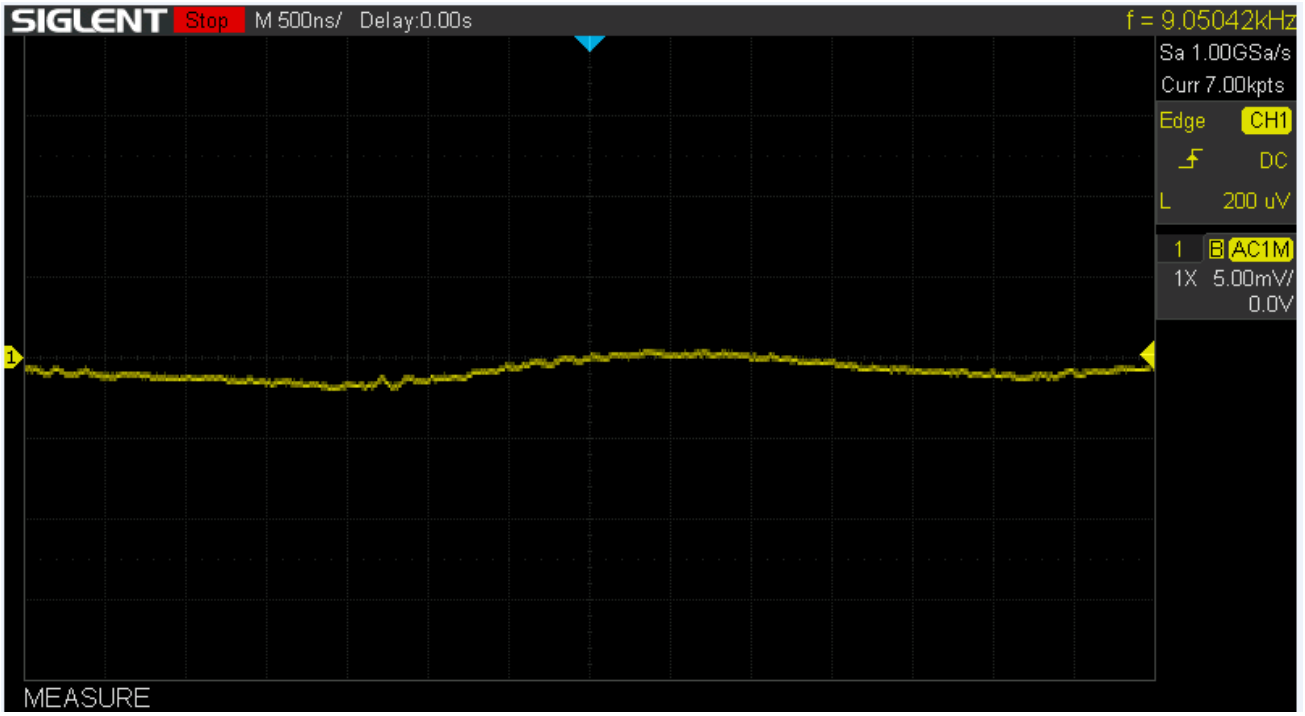
DDR1V5

Close to the FPGA:



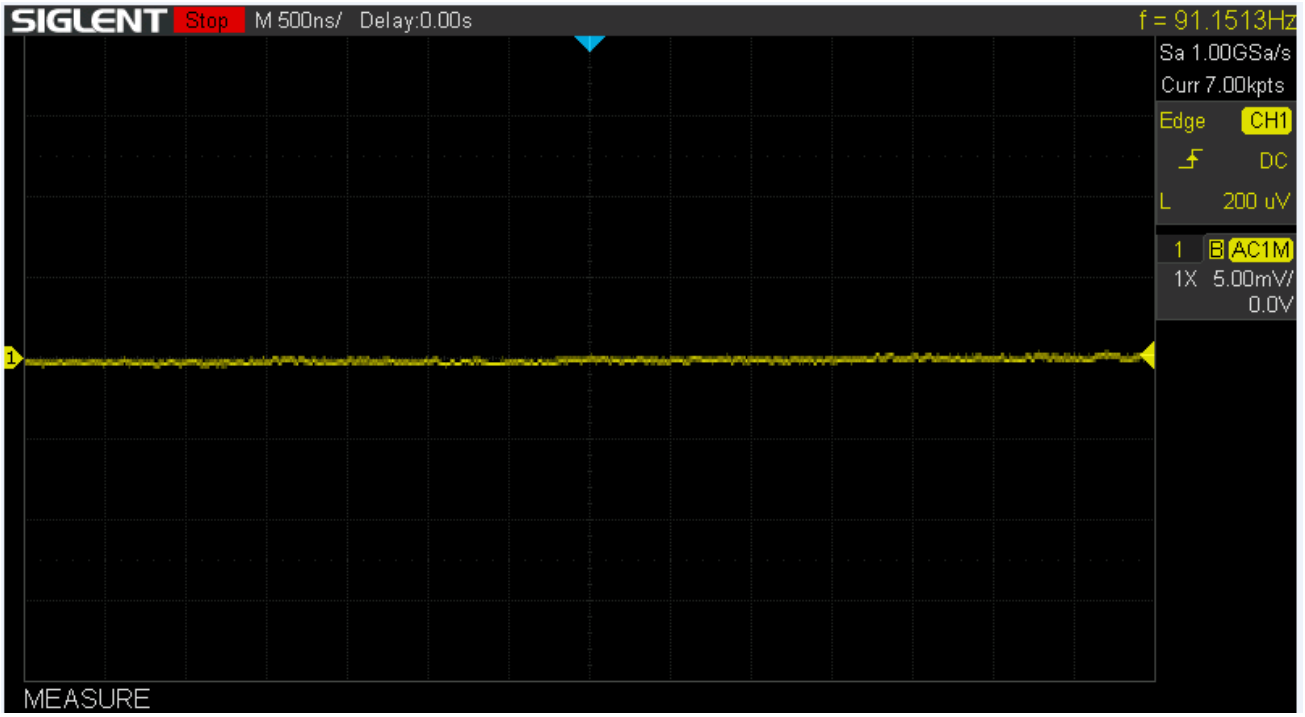
Close to the RAM:

DDR0V75


SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 9.05042kHz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
MEASURE

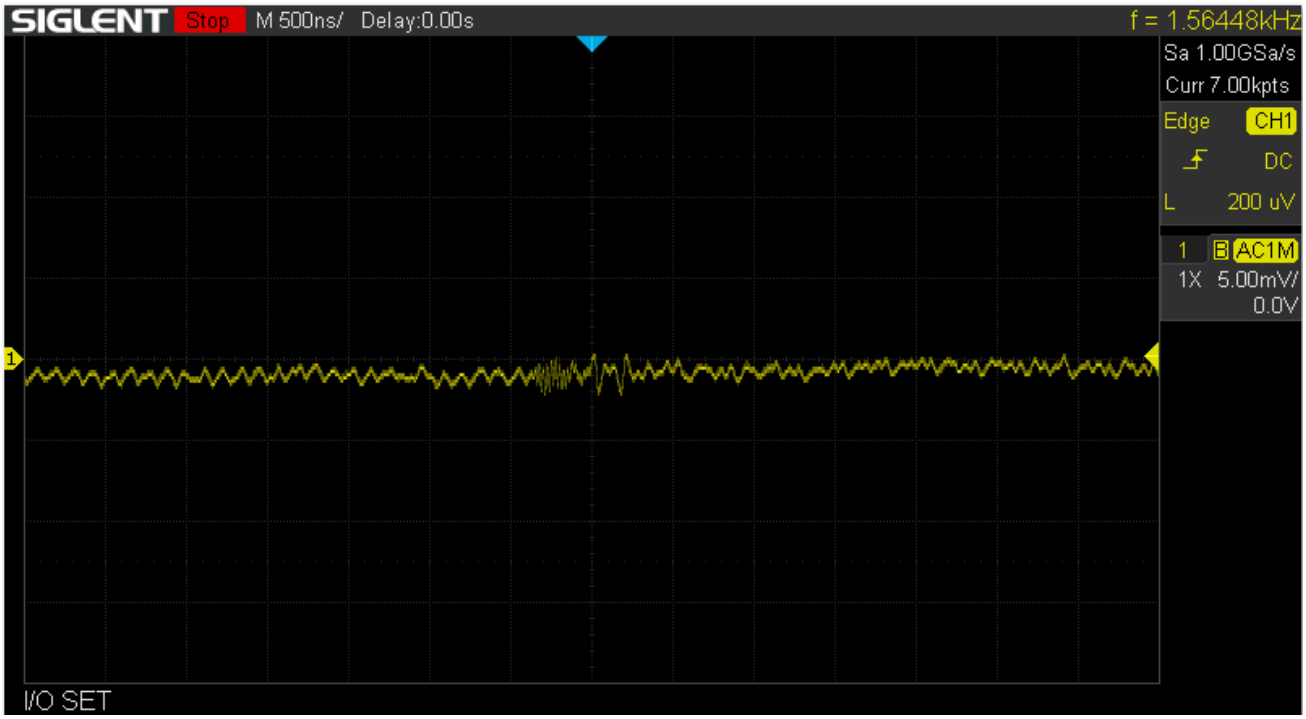
SFP

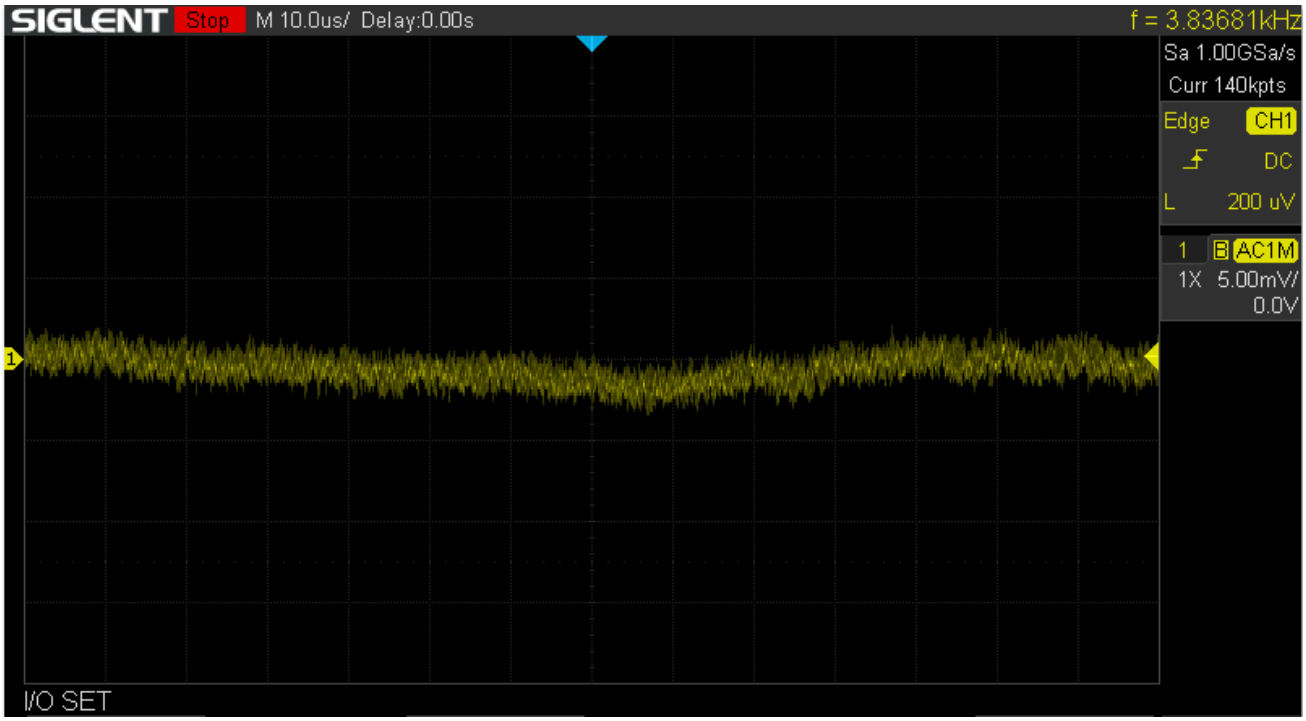
These measurement is taken at the 3V3 rail, after the chokes for the SFPs:



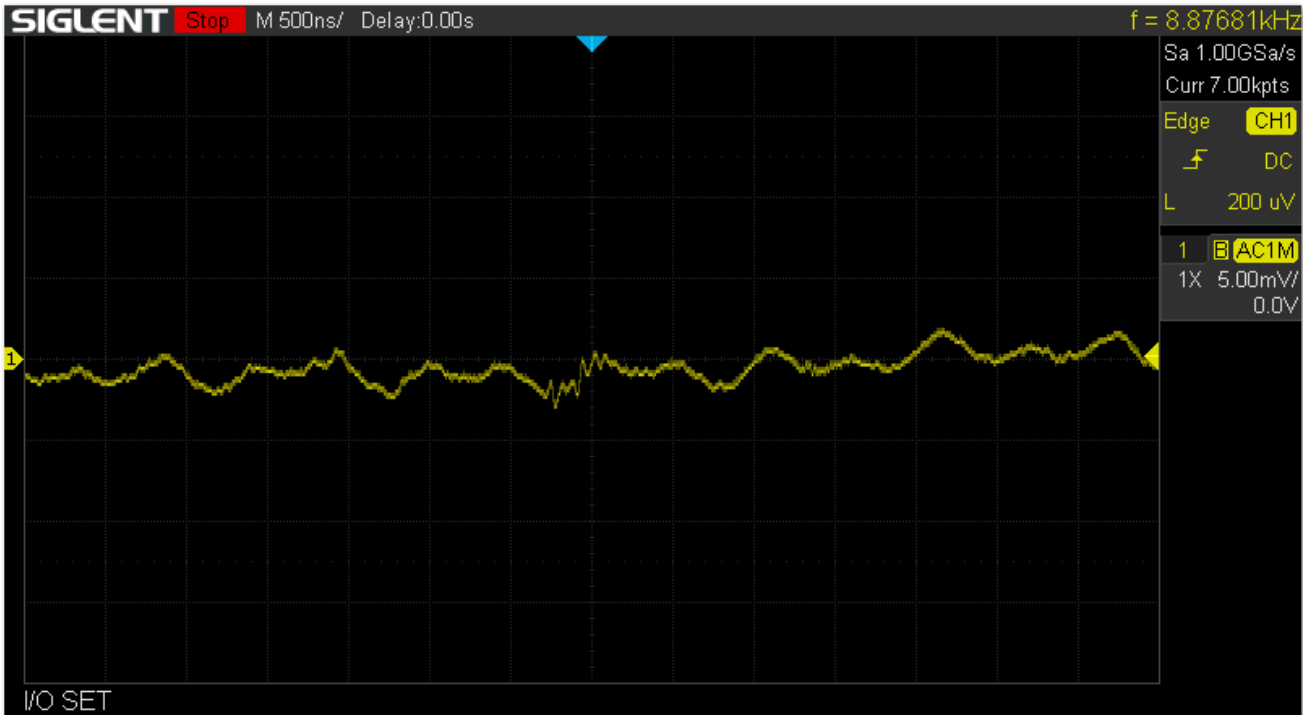
Stressed measurements

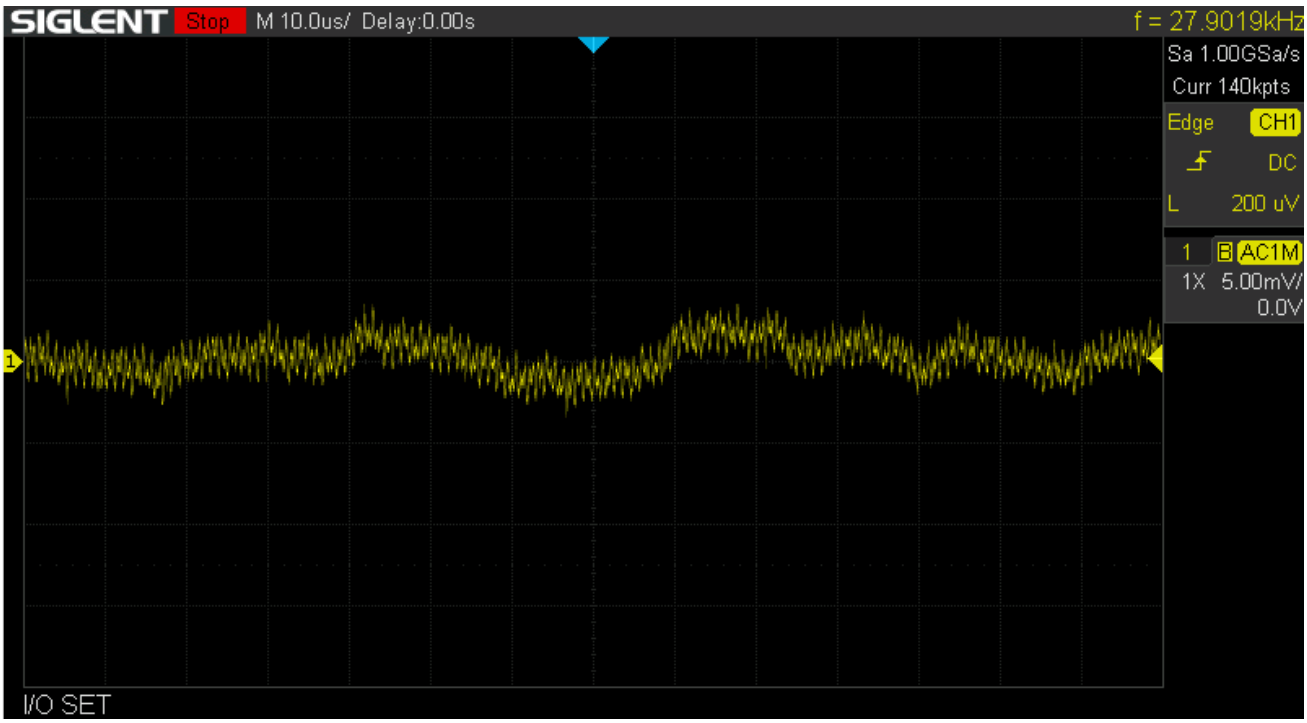
VCCINT



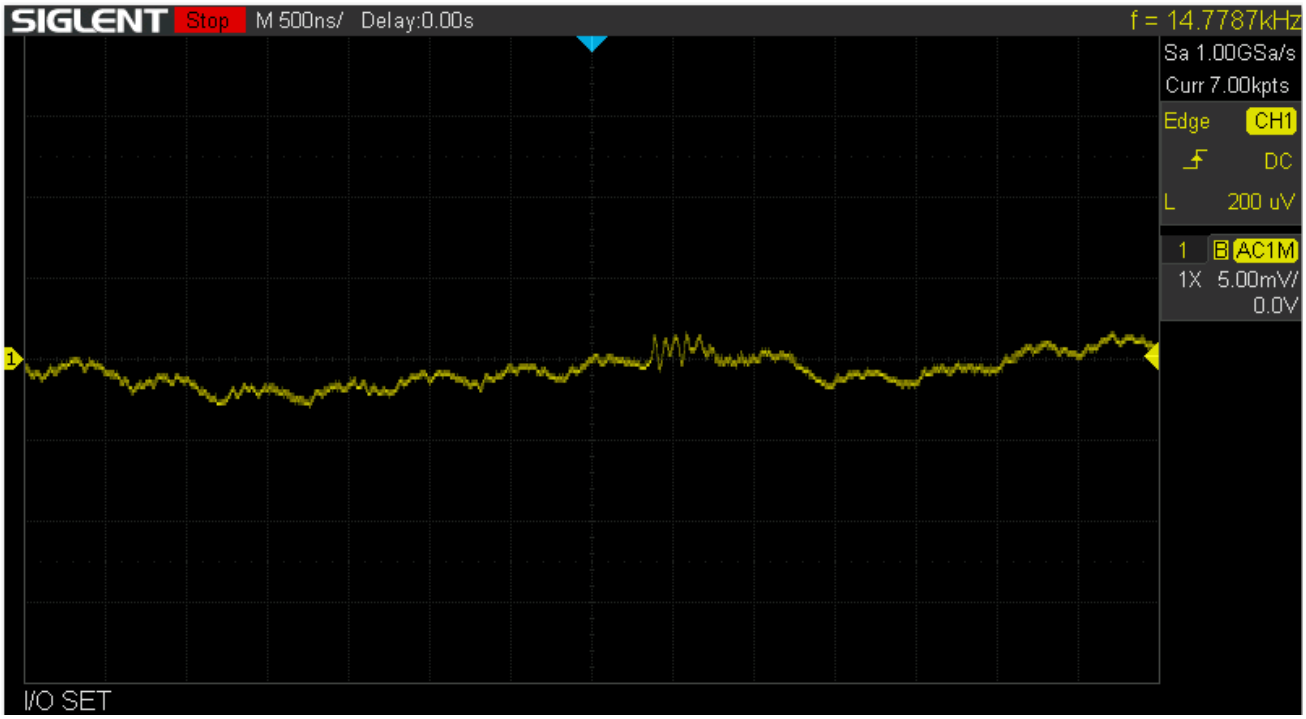


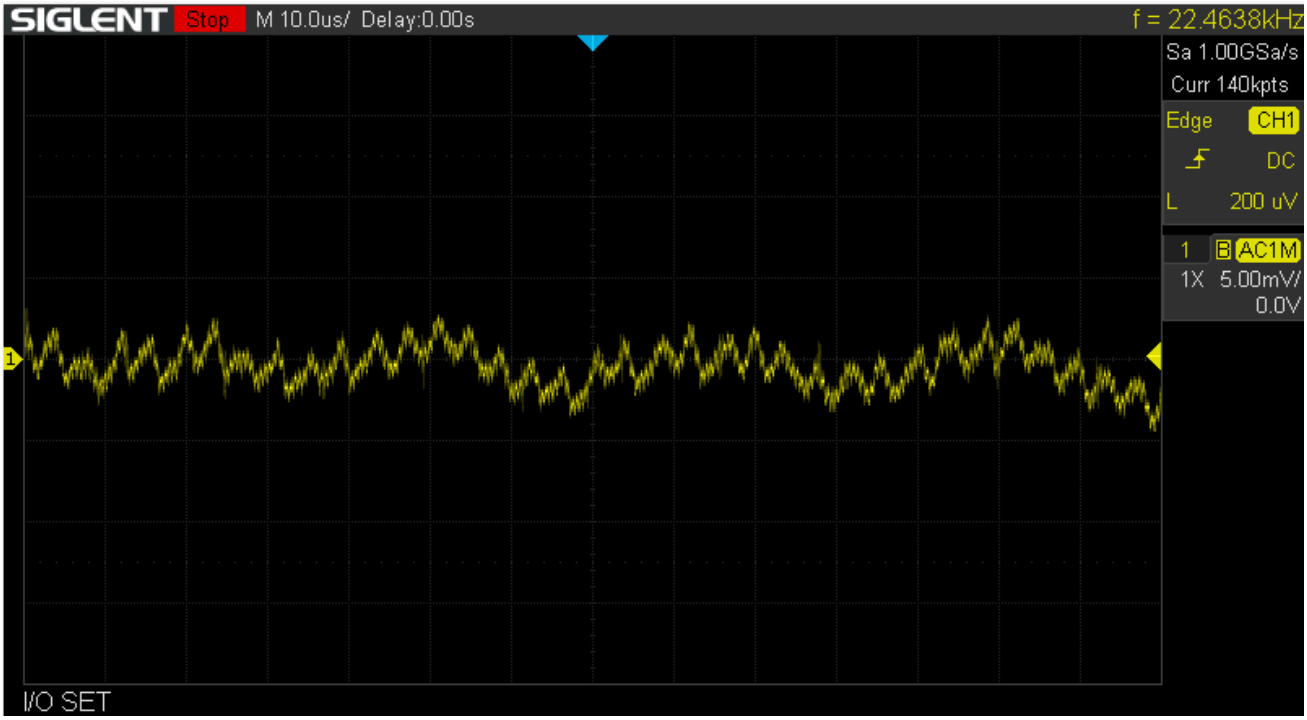
3V3


SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 8.87681kHz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
I/O SET

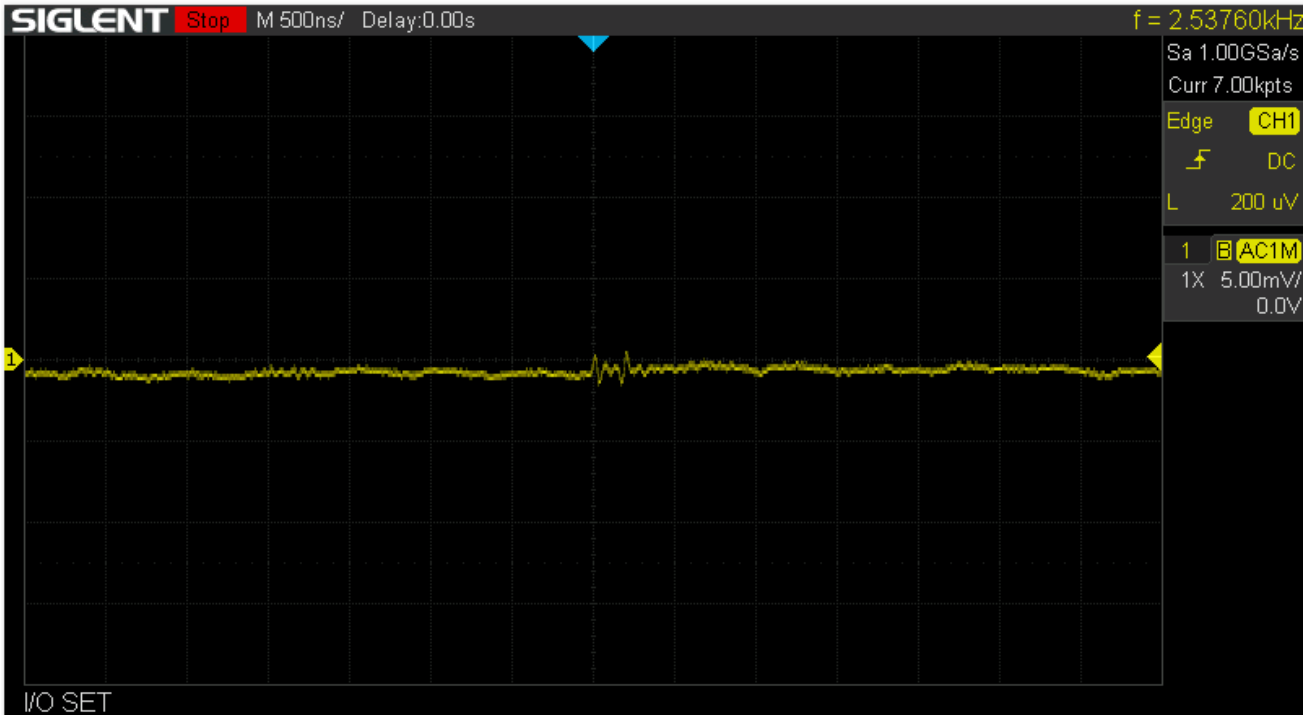

SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 27.9019kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
I/O SET

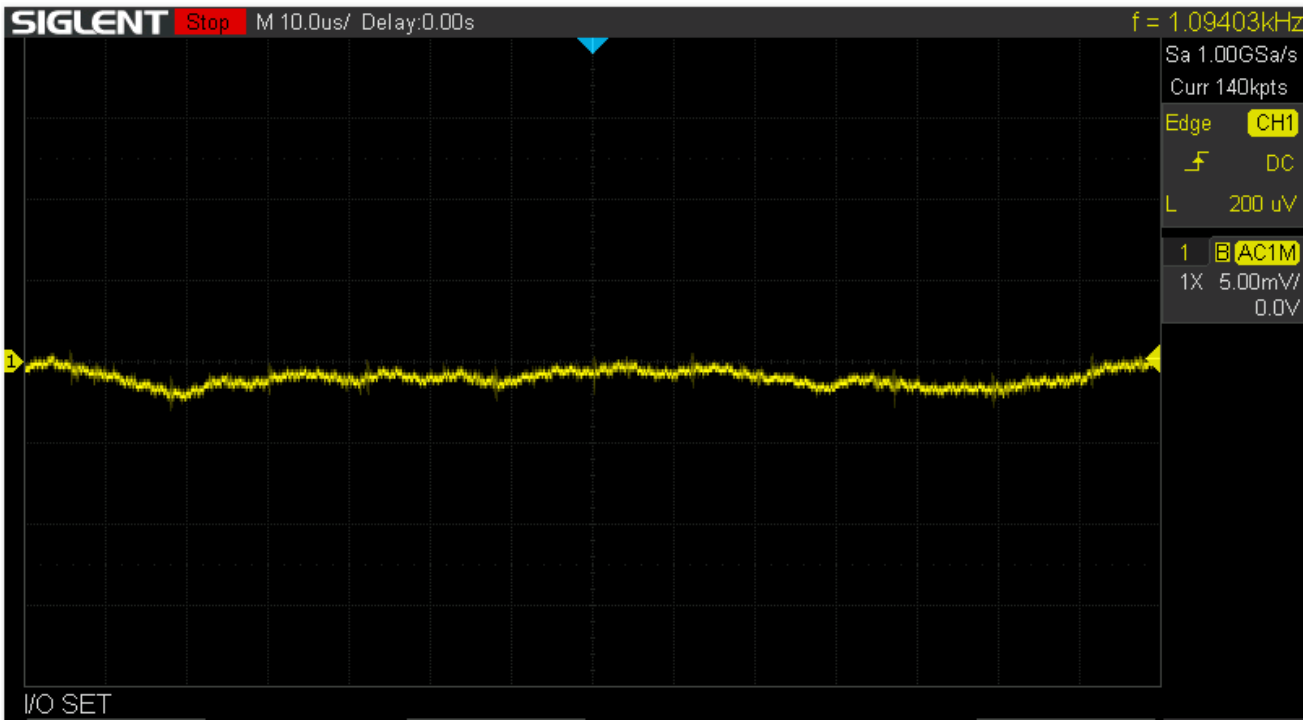
VCCAUX


SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 14.7787kHz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
I/O SET

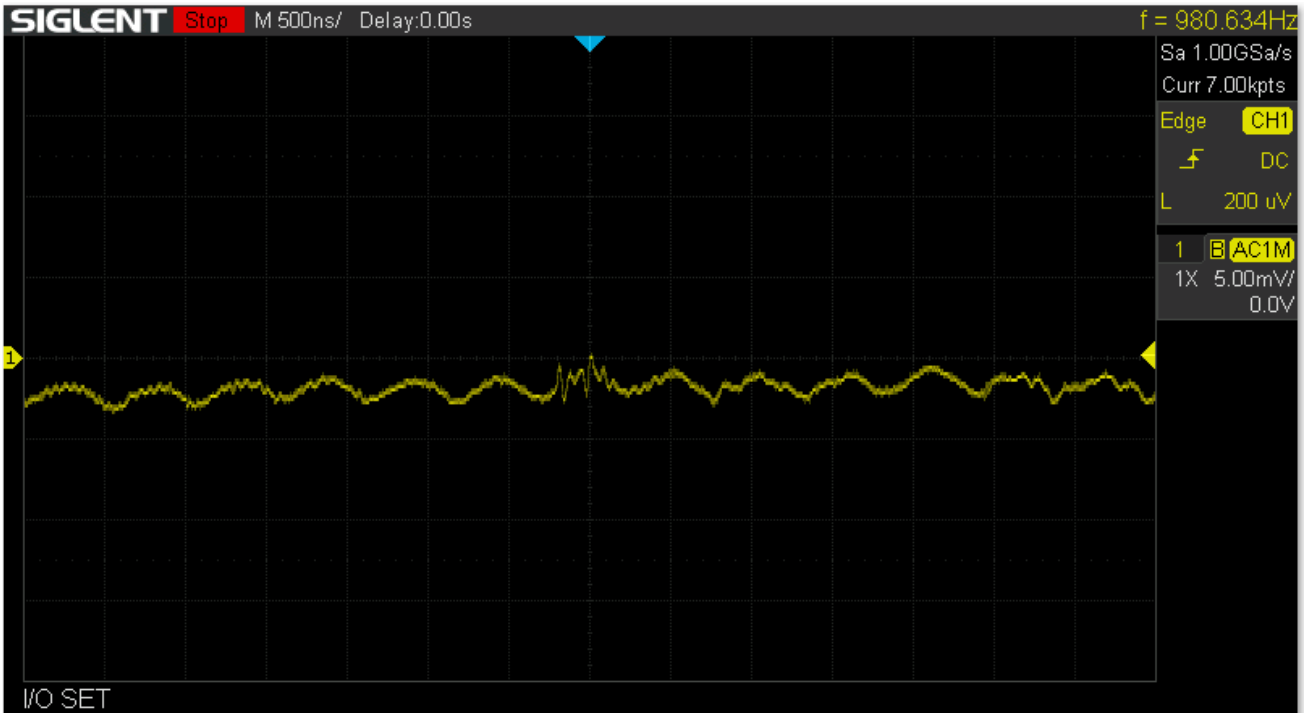

SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 22.4638kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
I/O SET

MGTAVCC


SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 2.53760kHz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
I/O SET

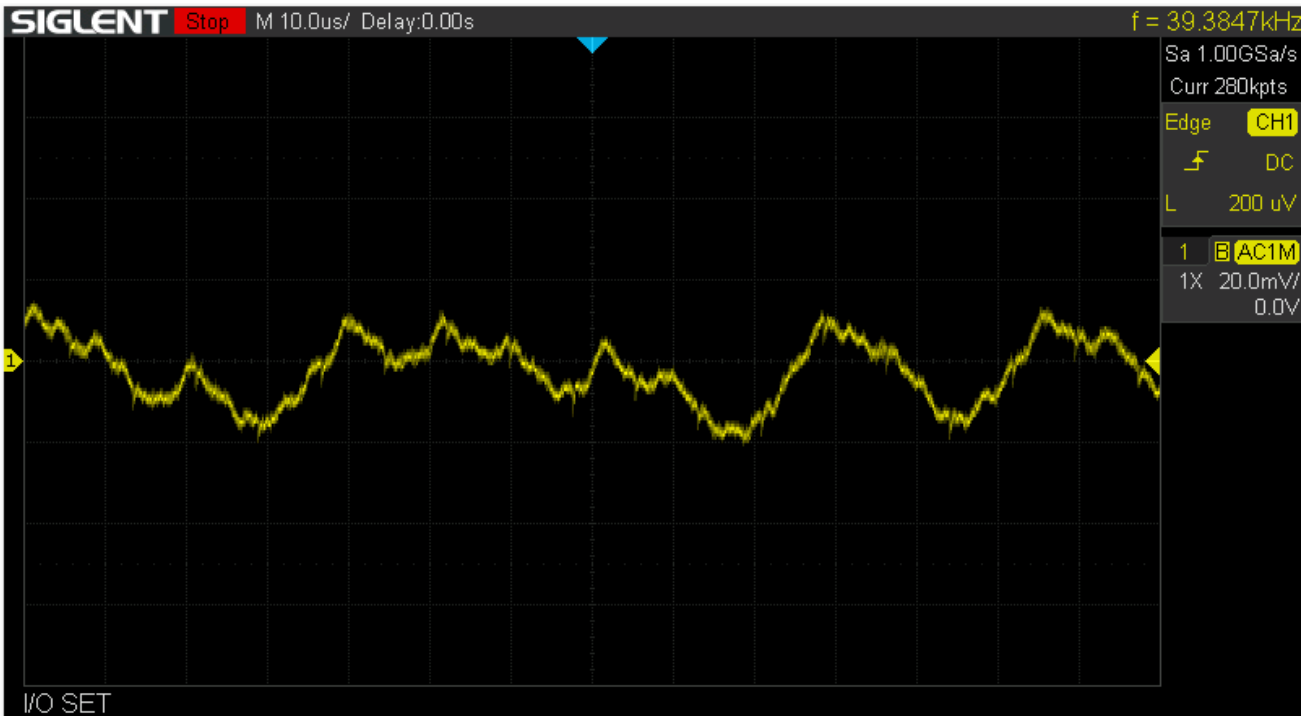

SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 1.09403kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L 200 uV
1 B AC1M
1X 5.00mV/
0.0V
I/O SET

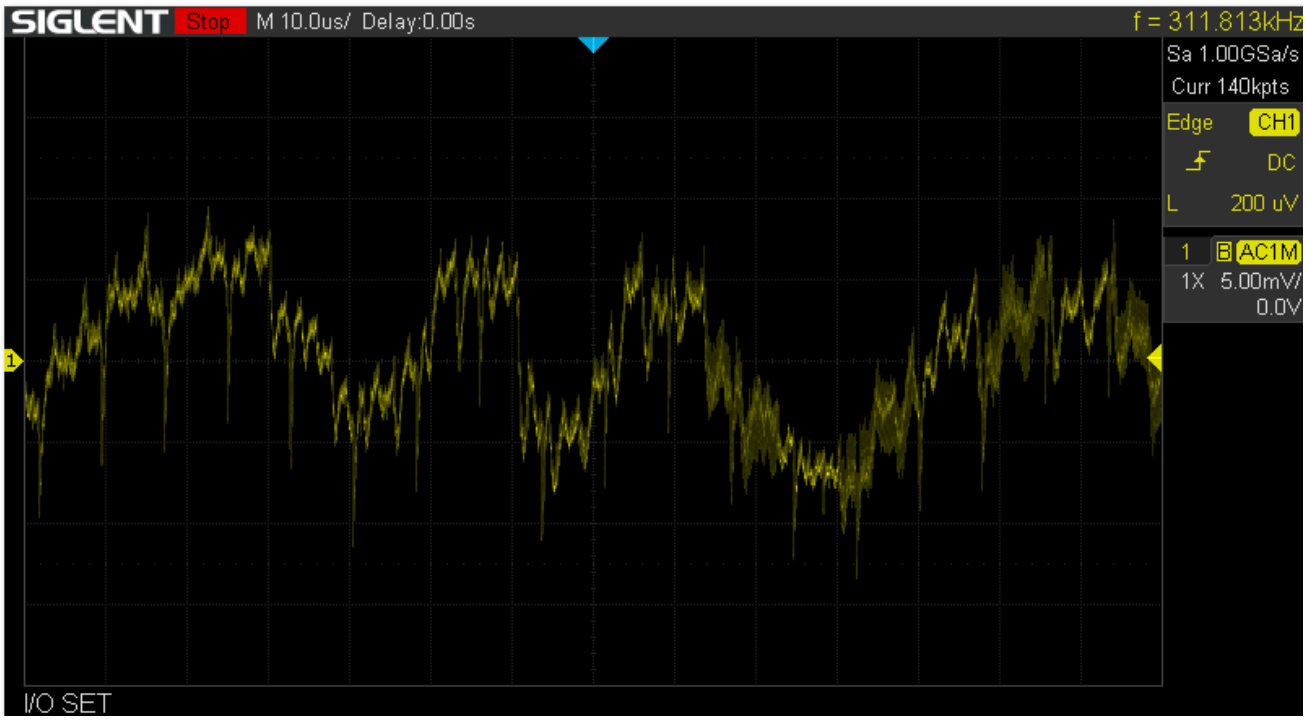
MGTAVTT


SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 980.634Hz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L
200 uV
1
B AC1M
1X 5.00mV/
0.0V
I/O SET

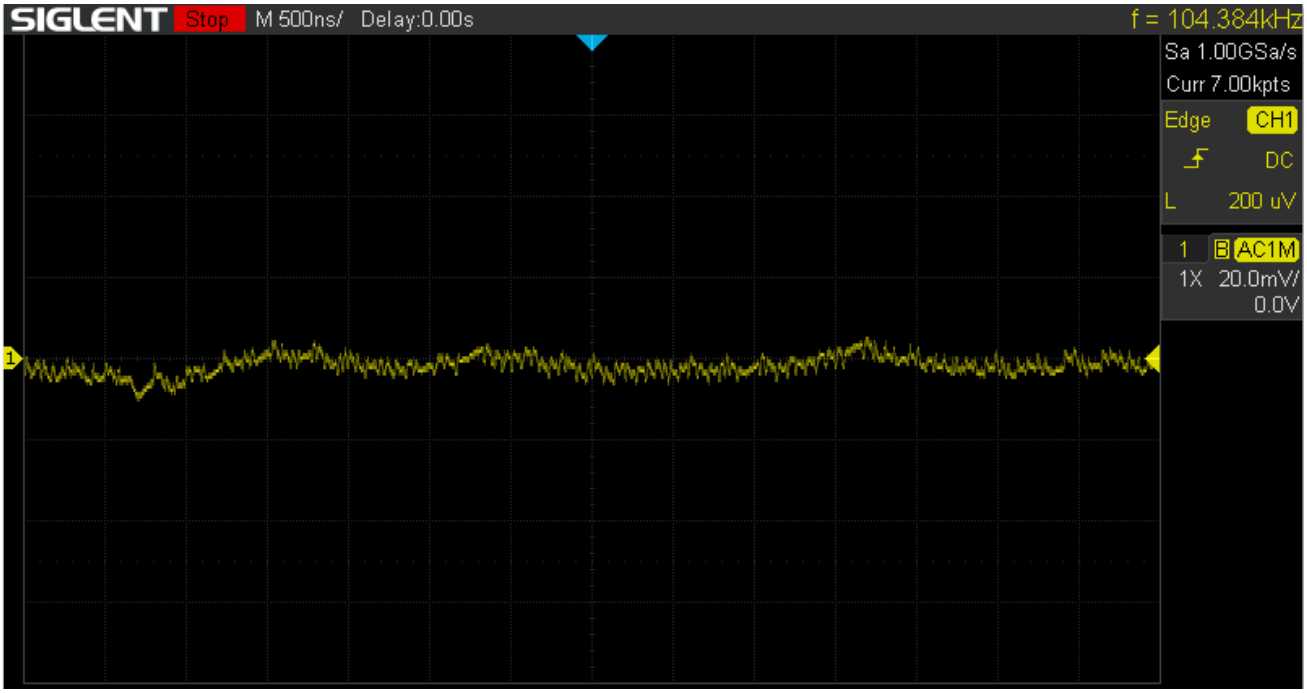

SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 3.68377kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L
200 uV
1
B AC1M
1X 5.00mV/
0.0V
I/O SET

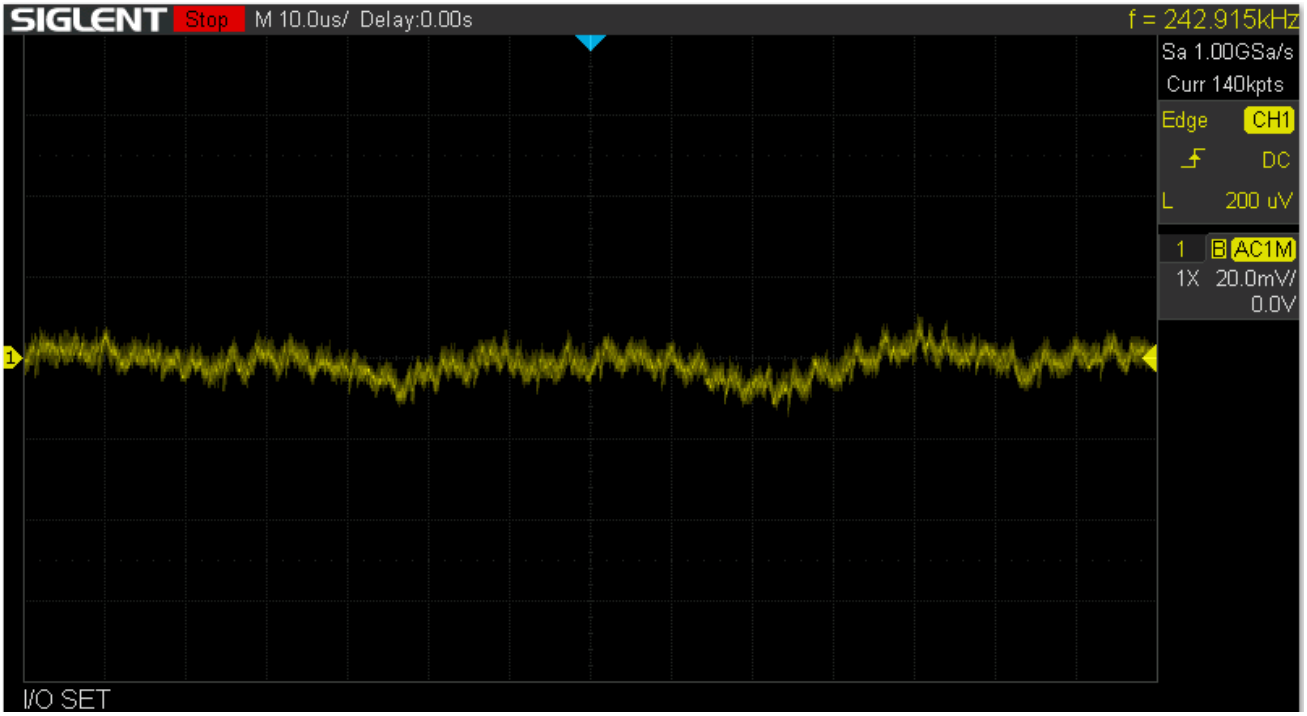
DDR1V5


SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 39.3847kHz
Sa 1.00GSa/s
Curr 280kpts
Edge
CH1
DC
L
200 uV
1
B
AC1M
1X
20.0mV/
0.0V
I/O SET

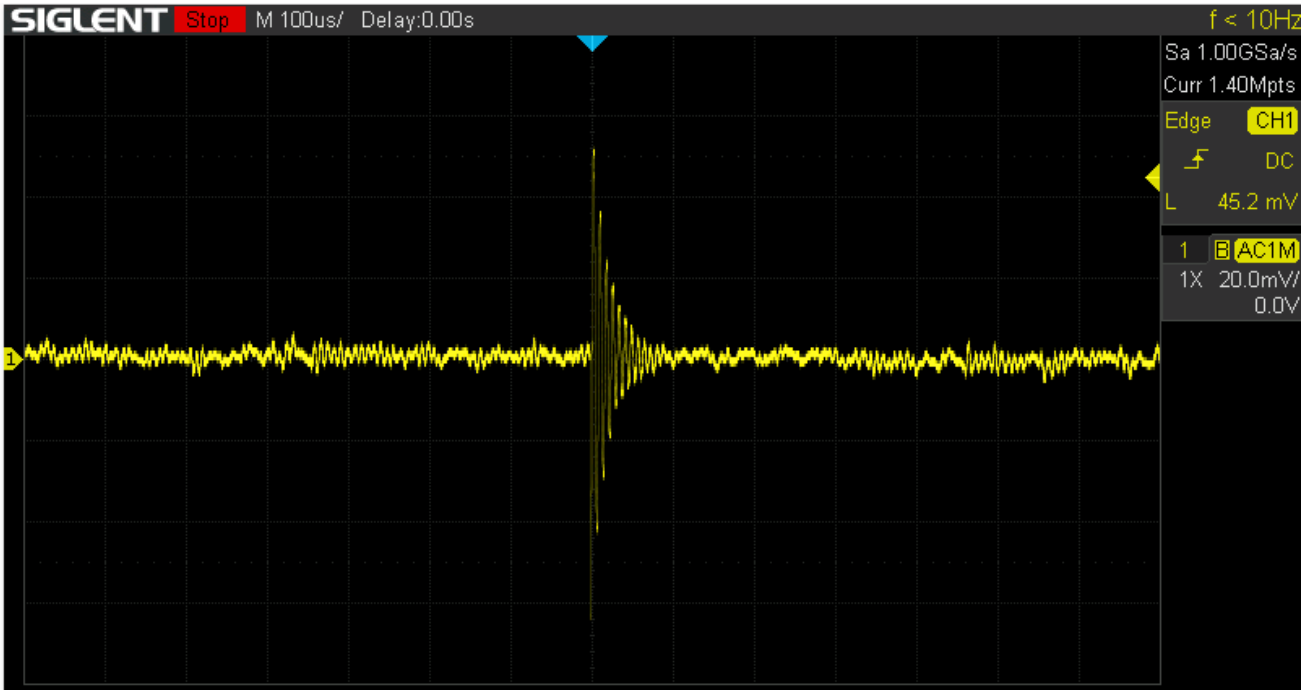

SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 311.813kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L
200 uV
1
B
AC1M
1X
5.00mV/
0.0V
I/O SET

DDR0V75


SIGLENT
Stop
M 500ns/ Delay:0.00s
f = 104.384kHz
Sa 1.00GSa/s
Curr 7.00kpts
Edge
CH1
DC
L
200 uV
1
B AC1M
1X 20.0mV/
0.0V


SIGLENT
Stop
M 10.0us/ Delay:0.00s
f = 242.915kHz
Sa 1.00GSa/s
Curr 140kpts
Edge
CH1
DC
L
200 uV
1
B AC1M
1X 20.0mV/
0.0V
I/O SET

SFP



In general SFP rail is low-noise even stressed, however these glitches appear with a frequency <10 Hz, and are particularly interesting. Since the SFP rails are connected to the power rails with chokes, this pulse did not appear in the power rails.

7.8 Eyescans

The eyescans were performed on a large range of data rates. The GTP transceivers on the ZYNQ have a maximum data rate specification of 6.6 Gbps. However in this system, they did show good performance up to considerably higher overclocked data rates. Eyescans were taken in this data rates as well, to aid data integrity analysis. The data rate range is from 1.25 Gbps to 10 Gbps. Up to ~9 Gbps, BER lower than 10^{13} was demonstrated. The horizontal units range from -0.5 to +0.5 Unit Intervals of the MGT clock cycle, and the vertical scale is at Voltage Codes.

SFP 1 (X0Y1(TX)/X0Y2(RX))

1.25 Gbps

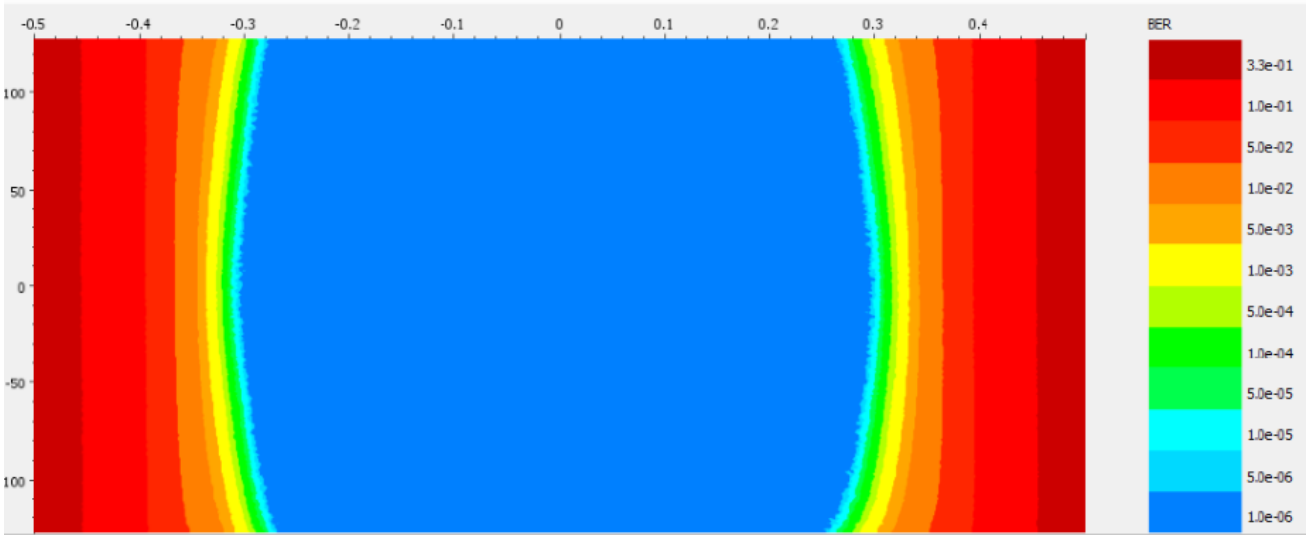


Figure 111. SFP1 @ 1.25 Gbps, eyescan dwell BER = 10^{-5}

2.5 Gbps

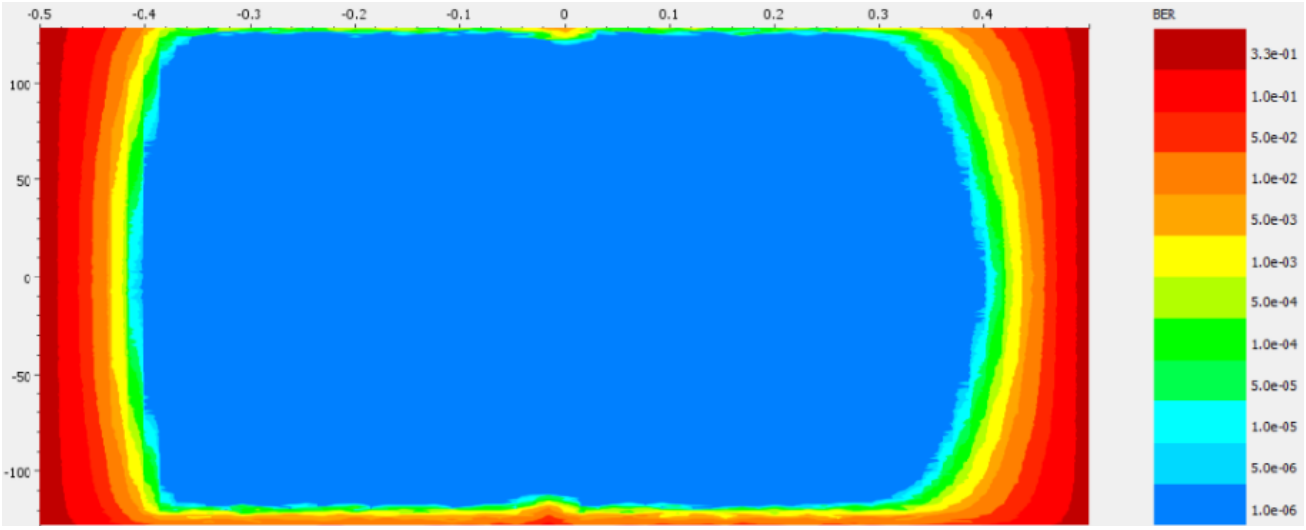


Figure 112. SFP1 @ 2.5 Gbps, eyescan dwell BER = 10^{-6}

3.906 Gbps

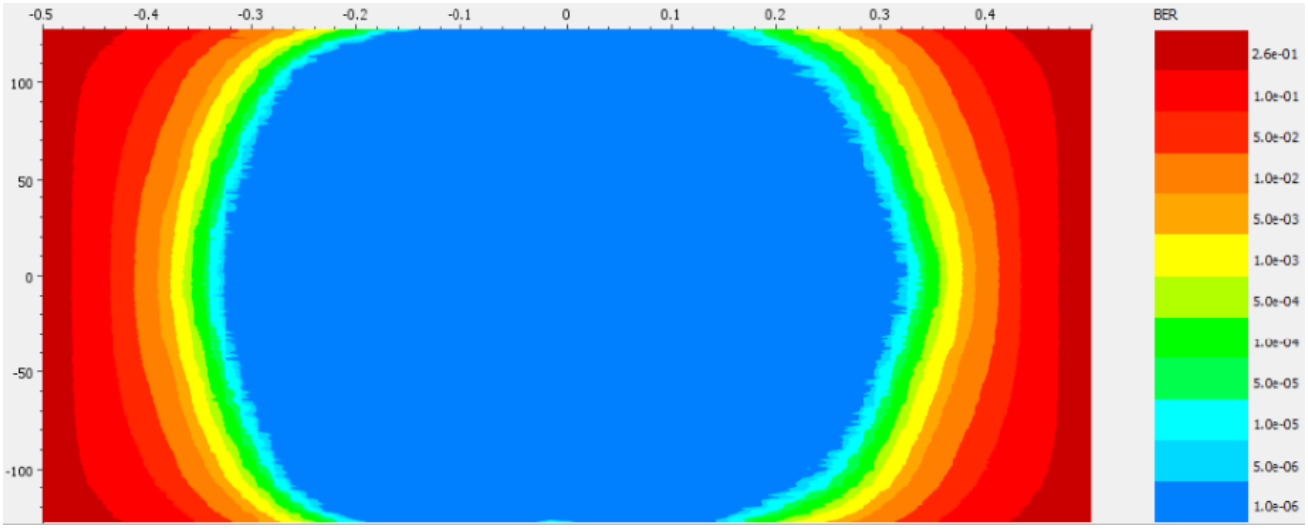


Figure 113. SFP1 @ 3.906 Gbps, eyescan dwell BER = 10^{-6}

5 Gbps

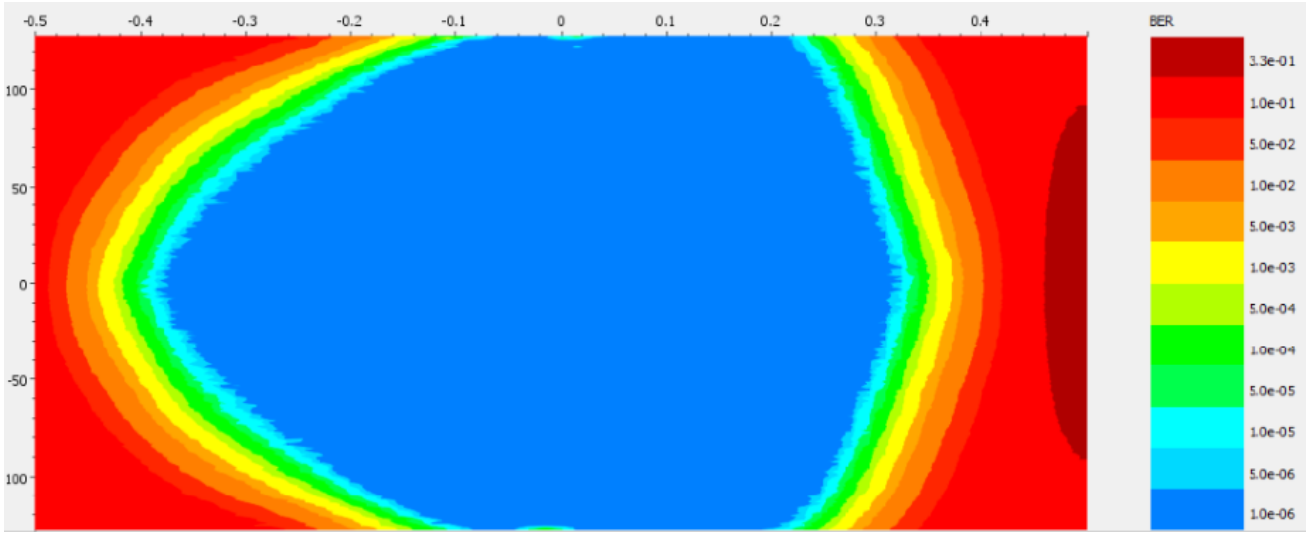


Figure 114. SFP1 @ 5 Gbps, eyescan dwell BER = 10^{-6}

6.25 Gbps

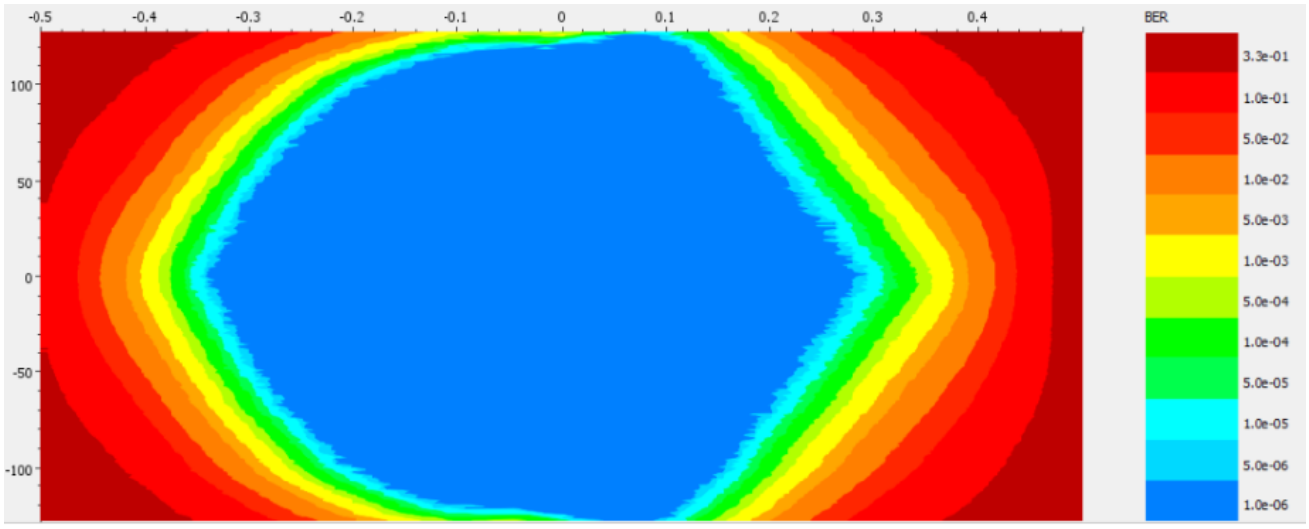


Figure 115. SFP1 @ 6.25 Gbps, eyescan dwell BER = 10^{-6}

7.5 Gbps

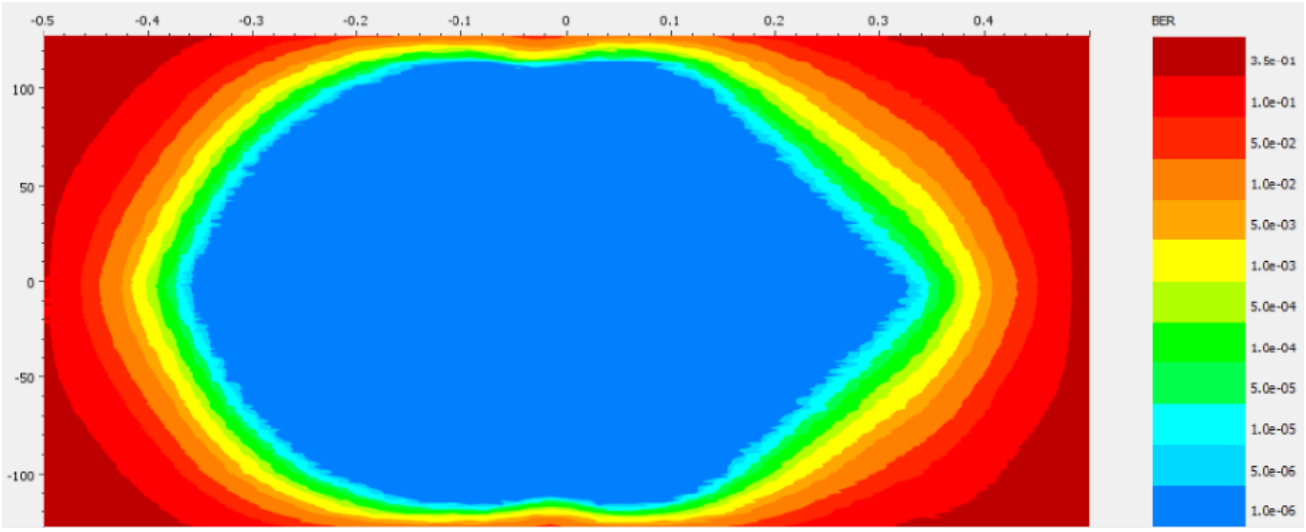


Figure 116. SFP1 @ 7.5 Gbps, eyescan dwell BER = 10^{-6}

9.375 Gbps

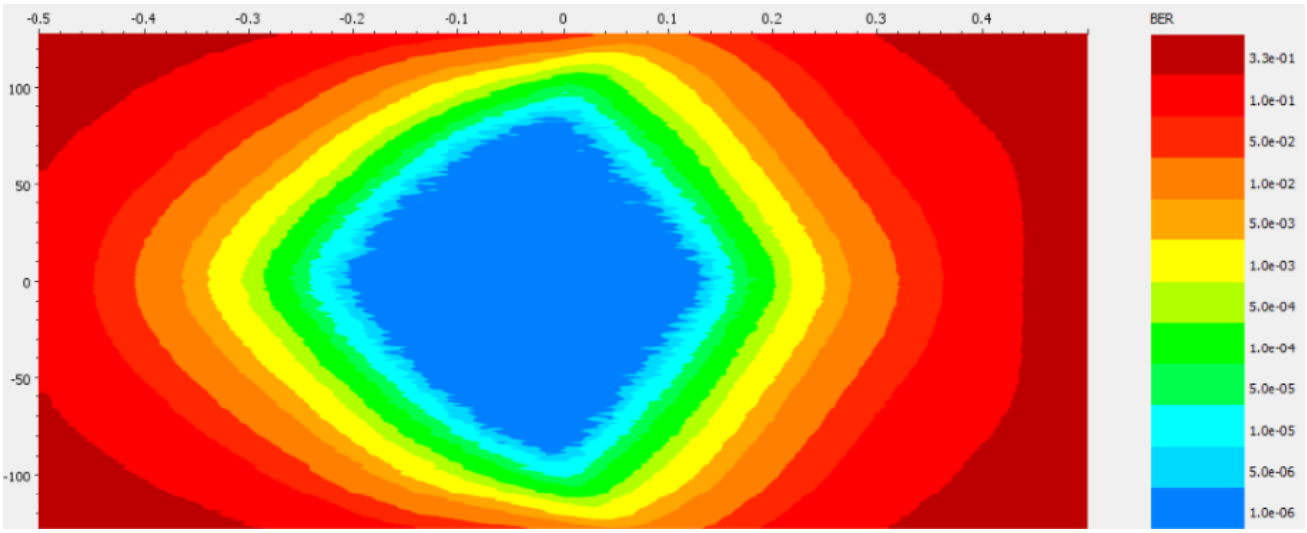


Figure 117. SFP1 @ 9.375 Gbps, eyescan dwell BER = 10^{-6}

10 Gbps

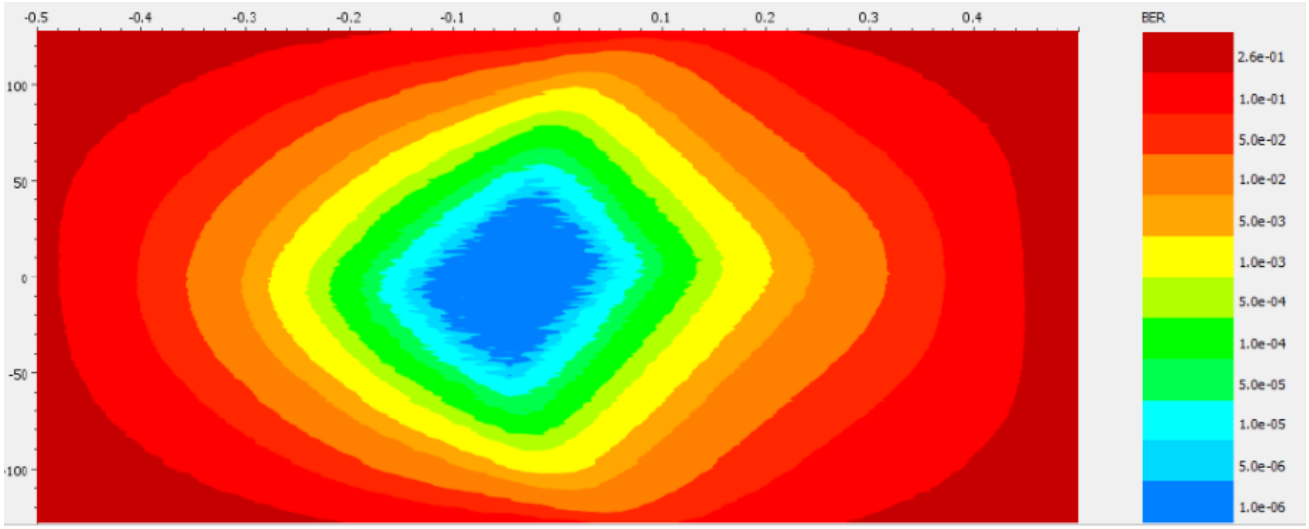


Figure 118. SFP1 @ 10 Gbps, eyescan dwell BER = 10^{-6}

SFP 2 (X0Y2(TX)/X0Y1(RX))

1.25 Gbps

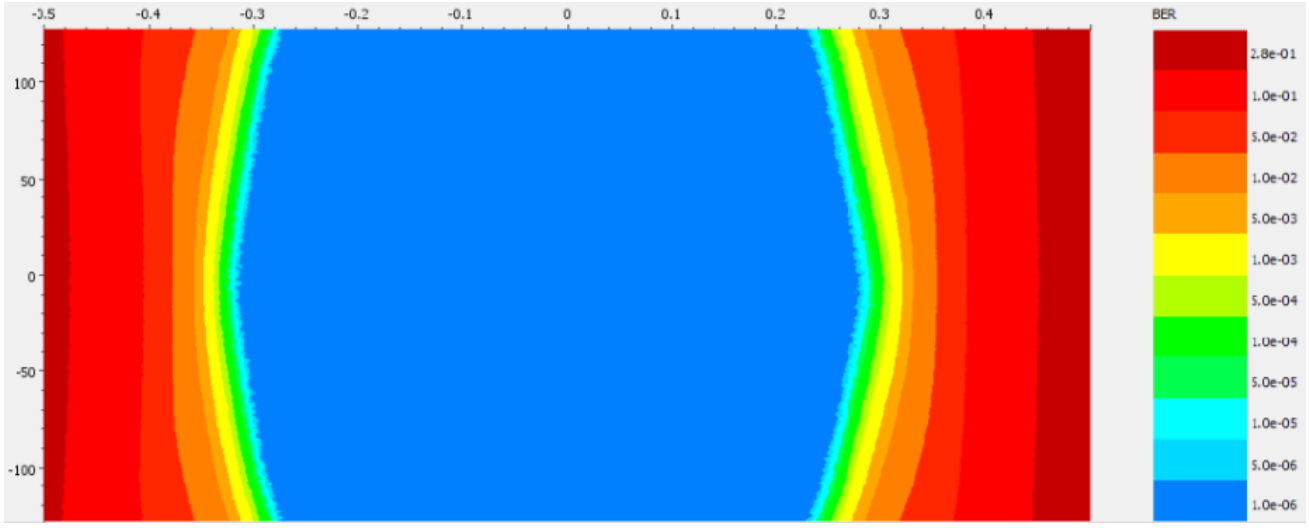


Figure 119. SFP2 @ 1.25 Gbps, eyescan dwell BER = 10^{-5}

2.5 Gbps

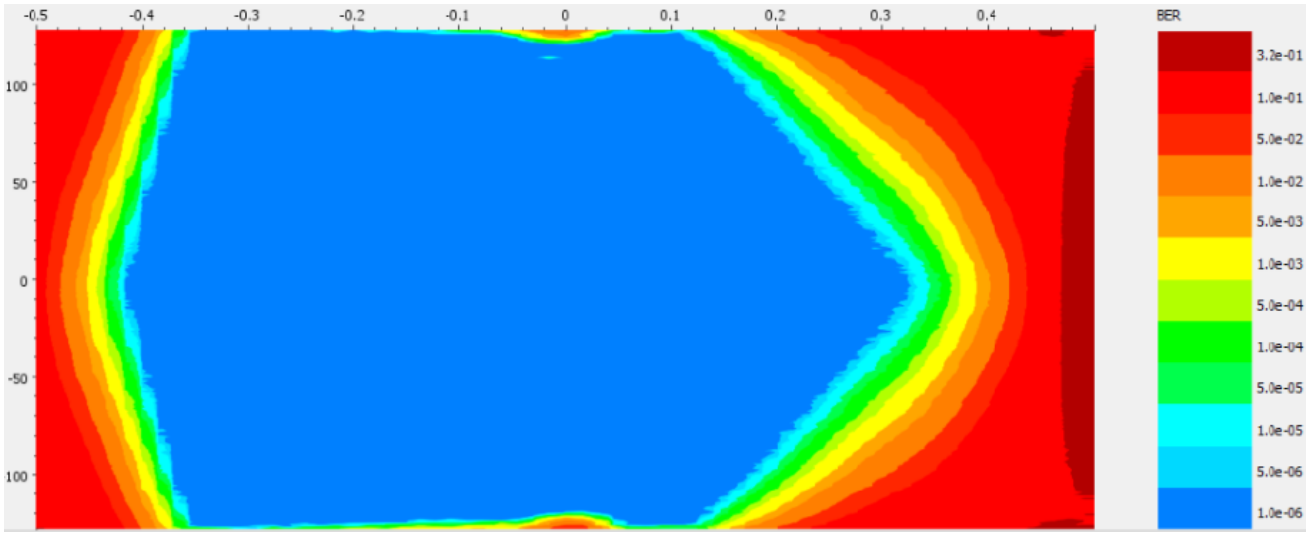


Figure 120. SFP2 @ 2.5 Gbps, eyescan dwell BER = 10^{-6}

3.906 Gbps

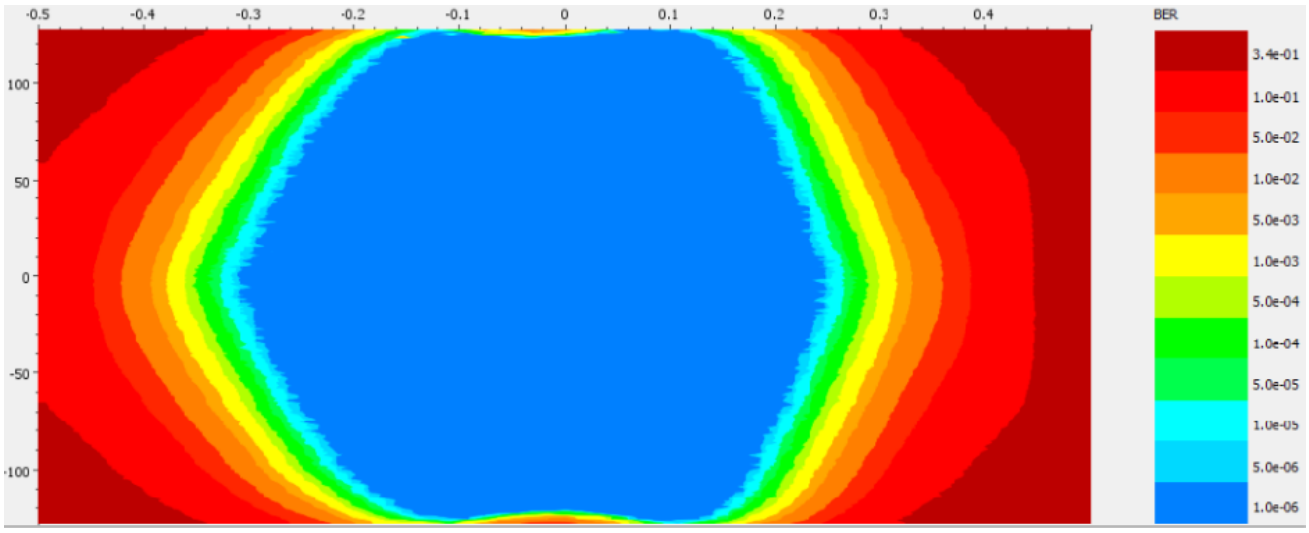


Figure 121. SFP2 @ 3.906 Gbps, eyescan dwell BER = 10^{-6}

5 Gbps

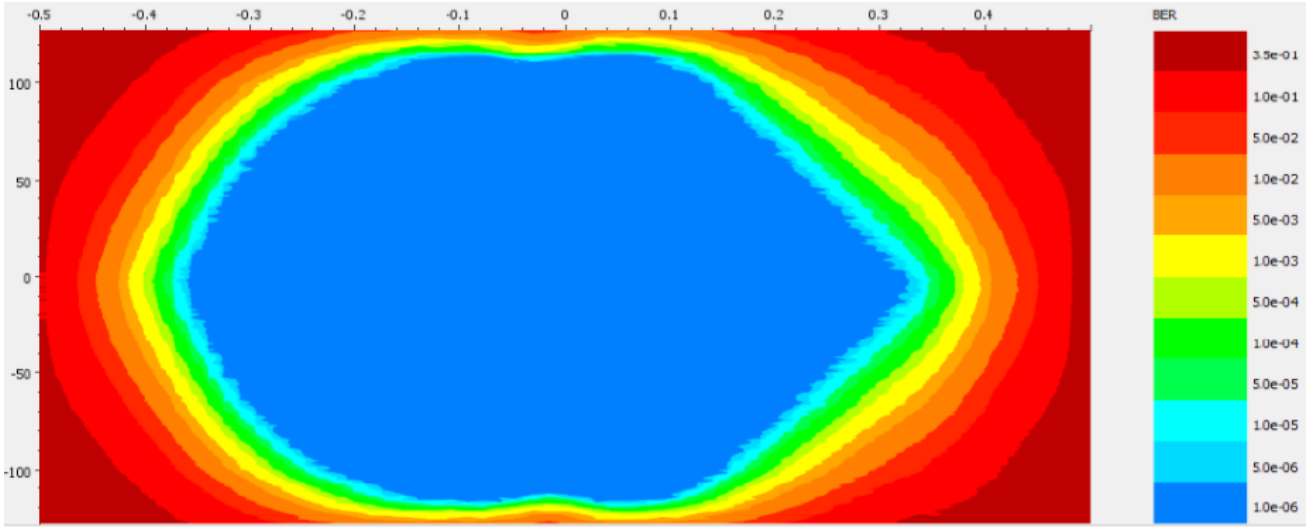


Figure 122. SFP2 @ 5 Gbps, eyescan dwell BER = 10^{-6}

6.25 Gbps

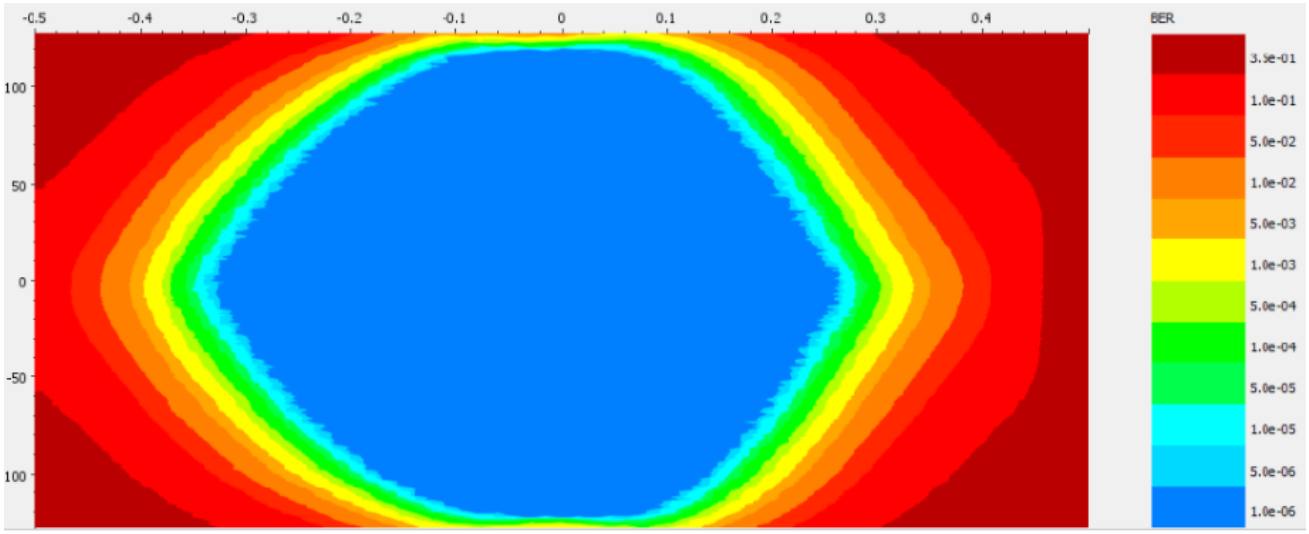


Figure 123. SFP2 @ 6.25 Gbps, eyescan dwell BER = 10^{-6}

7.5 Gbps

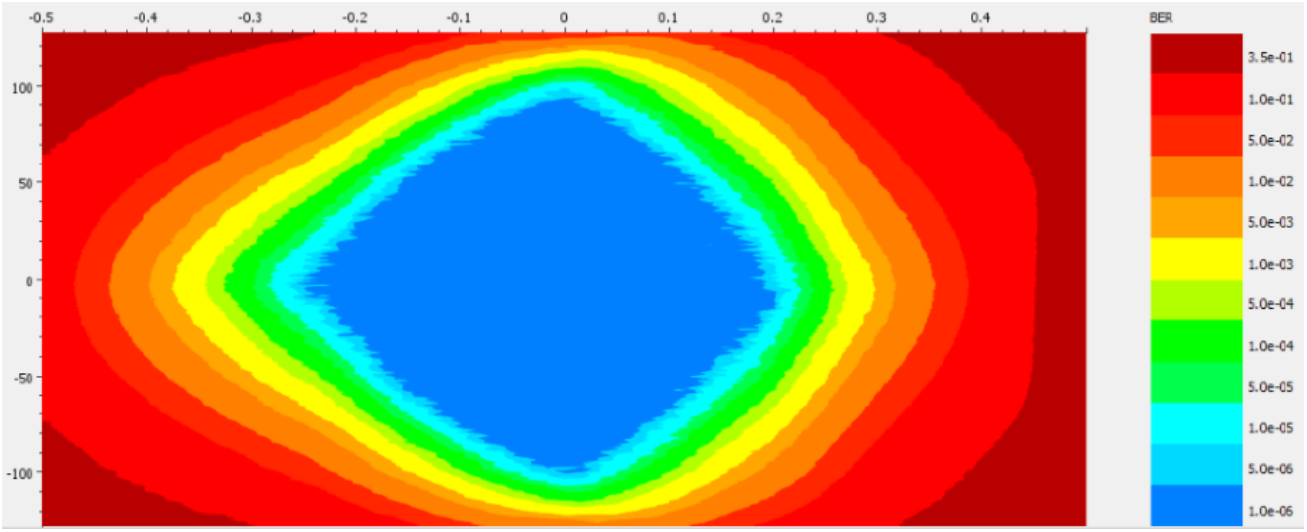


Figure 124. SFP2 @ 7.5 Gbps, eyescan dwell BER = 10^{-6}

9.375 Gbps

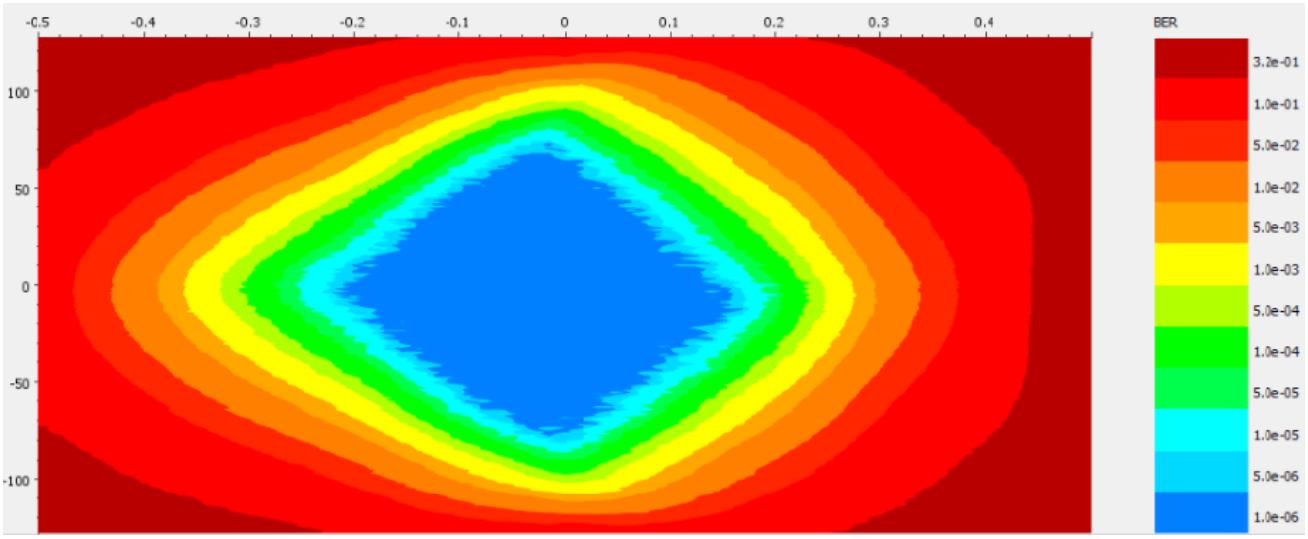


Figure 125. SFP2 @ 9.375 Gbps, eyescan dwell BER = 10^{-6}

10 Gbps

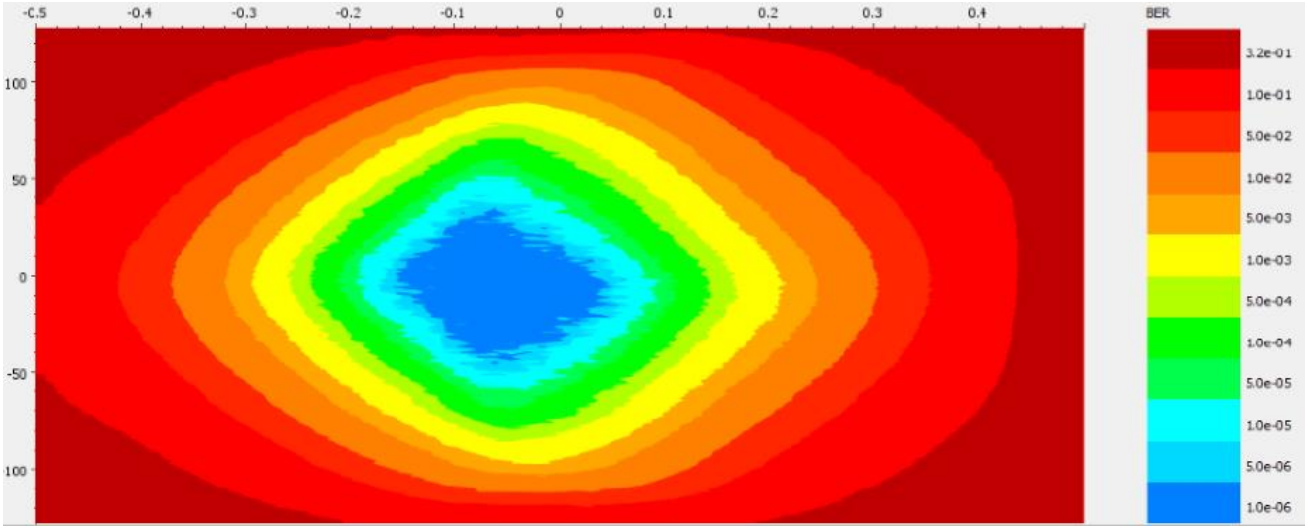


Figure 126. SFP2 @ 10 Gbps, eyescan dwell BER = 10^{-6}

SMA MGTs

Eyescans were taken on the MGTs exposed on the SMA connectors, which exhibit error-free performance as well, however since the cables available at the timing of the thesis were suboptimal, rated for below 1 GHz applications, and show very high attenuation at higher frequencies, only a few scans are included and more will be added soon.

SMA 1 (X0Y3(TX)/X0Y3(RX))

3.25 Gbps

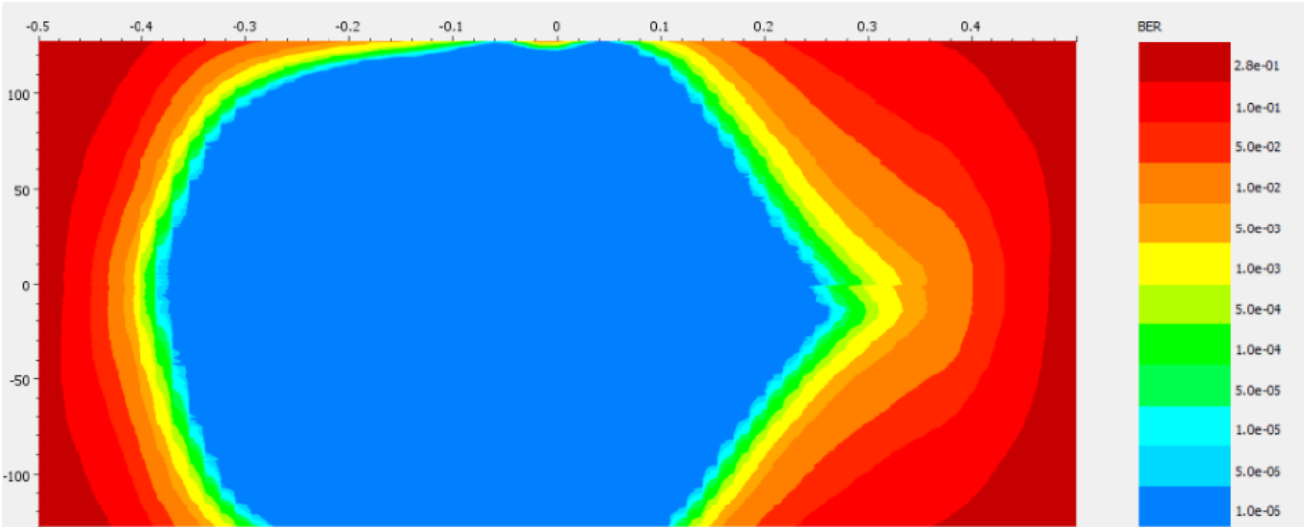
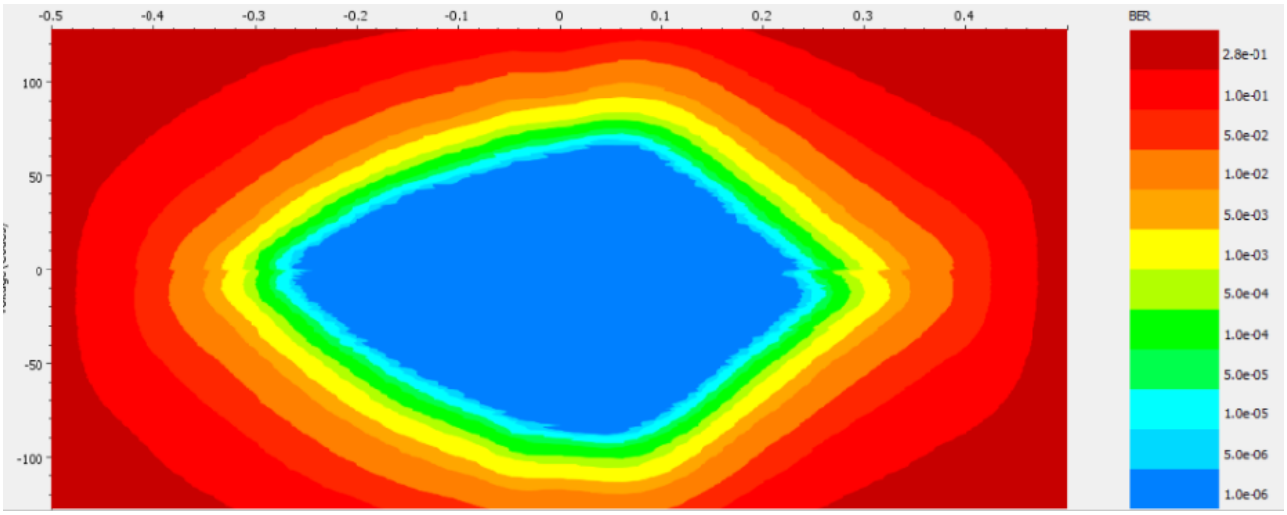


Figure 127. SMA1 @ 3.25 Gbps, eyescan dwell BER = 10^{-6}

6.25 Gbps



SMA 2 (XOYO(TX)/XOYO(RX))

3.25 Gbps

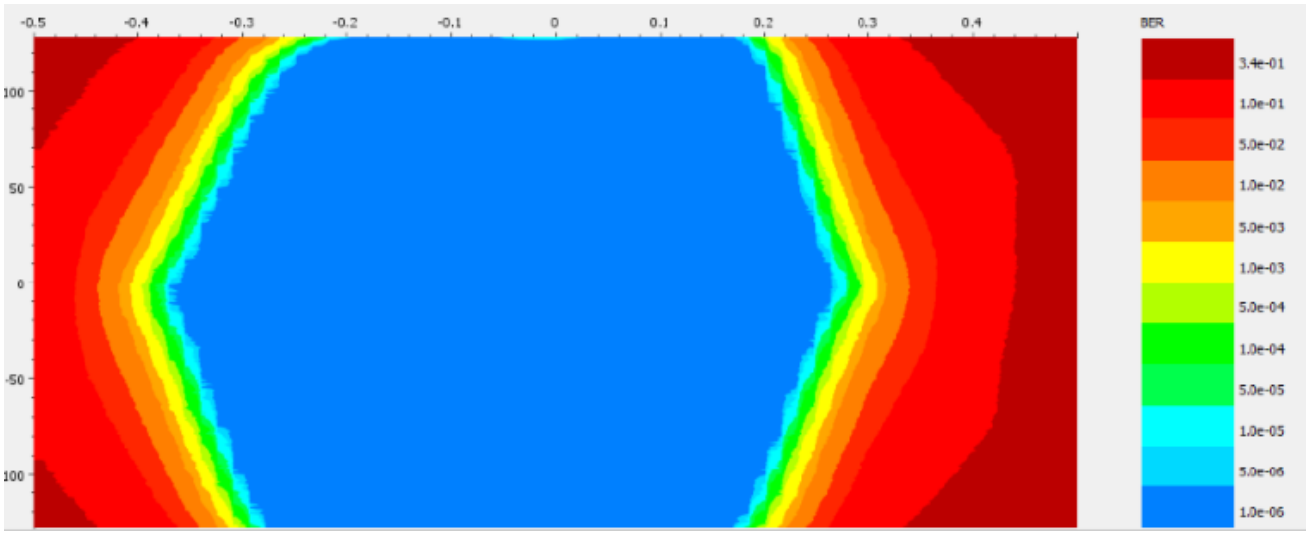


Figure 129. SMA2 @ 3.25 Gbps, eyescan dwell BER = 10^{-6}

6.25 Gbps

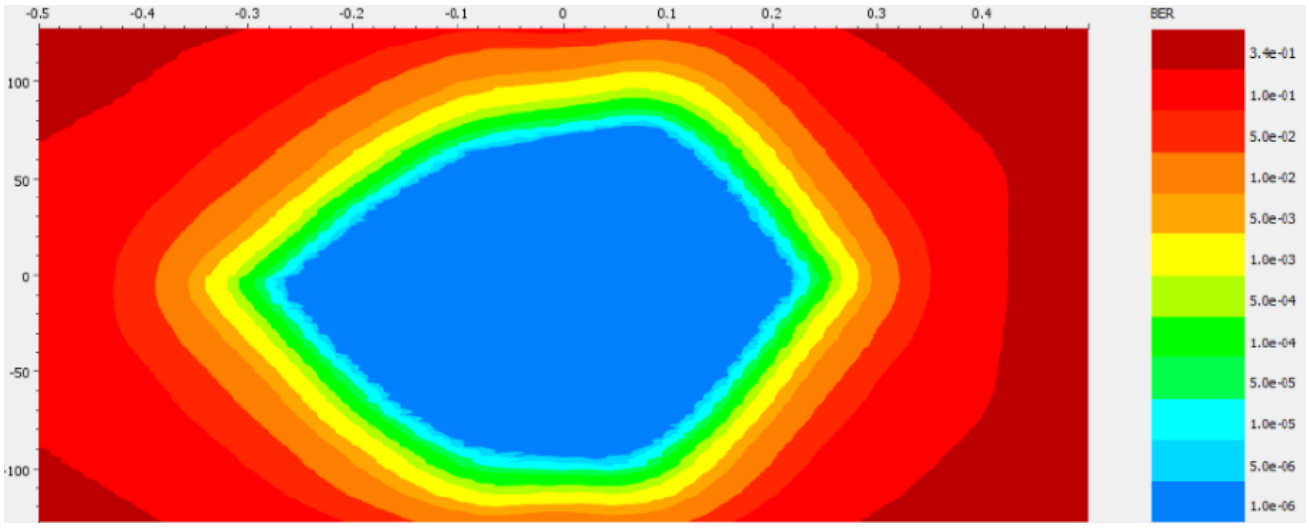


Figure 130. SMA1 @ 6.25 Gbps, eyescan dwell BER = 10^{-6}

7.9 Errata

Resistor R5, which forms a low pass filter with C8 to feed the analog power supply to the U6 regulator was registered in the BOM as 10kΩ instead of 10Ω, resulting to not enough current flowing. Removing the resistor and soldering the pads shorts to eliminate the LPF, or replacing the resistor with a 10Ω one fixes the problem.

Resistor R72 of the UART circuitry, which is characterized as DNP (Do Not Place), was placed as a result of a BOM error. De-soldering this resistor fixes this problem.

As a result of a CAD error, VCCAUX did not have its feedback connected. Soldering the feedback fixes the problem.

On the I2C mux, the SCL and SDA are crossed. De-soldering the pins and cross-soldering correctly fixes this problem.