POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Elettronica

Tesi di Laurea Magistrale

Development of a SMARC module for an ADAS system based on the i.MX8 processor

Relatore:
prof. Massimo Violante

Candidato:
Lorenzo Giraudi

Ottobre 2018

Acknowledgements

My thanks goes to Ideas&Motion for letting me have this experience and also to all personnel for being nice and helpful. In particular, I am grateful to M. Elia for having taught me many useful things and for having always time for my questions.

My gratitude goes to my whole family, and in particular my parents for allowing me to take this path. Also to G. Imanone, for continuous support and assistance.

Summary

The thesis work places itself as part of a European Project called Autodrive. This project includes several company of the automotive sector with the aim to advance in the Autonomous vehicle field. The thesis work has been hosted by an Italian company called Ideas&Motion, active in the automotive field by providing engineering services and consultancy. They have been assigned by one of the partners to design and provide a developing and testing platform for ADAS systems.

The processor to be used was requested to be the i.MX8 Quad Max manufactured by NXP, that features 6 Arm Cores and 2 GPUs together with an automotive grade and many modern interfaces. The developing platform has been split in two different boards: a System On Module with the iMX8 (this thesis content) and a carrier board hosting it. During the first phases of the design, it has been chosen to design a board that could be reused in other project than this one. It has been chosen to adhere to an already defined standard of System On Module such that interoperability would be easier to satisfy.

Among several possibilities, the Smart Mobile ARChitecture (SMARC) has been chosen. It is a free and open standard maintained by the Standard Group of Embedded Technologies (SGET). A complete SMARC module would be suitable for the Autodrive project. Thus, the new purpose of the project has become the design of a SMARC board. The design started by studying and understanding all the standard interfaces.

Then there has been a study and research of the other needed components, like the DRAMs, the eMMC, the voltage regulators, the Ethernet PHYs, the USB HUB and the PCIe clock generator. The general architecture has been then defined: the Module will have 6GB of LPDDR4 SDRAM, 32 GB of storage from the eMMC, 2 PCIe Gen 3.0 lanes, 2 Gigabit Ethernet ports, 5 USB ports, one DisplayPort or one HDMI, one on-module WiFi and Bluetooth adapter, one uSD socket on the module and one SD interface to the carrier. The final result consists in the schematic of the board, developed with Mentor Xpedition keeping in mind the requirements of the iMX8, SMARC standard and interface standards.

Contents

Acronyms

ADAS	Advanced driver-assistance systems
CSI	Camera Serial Interface
DSI	Display Serial Interface
ECU	Electronic Control Unit
eMMC	embedded MultiMedia Card
ESAI	Enhanced SAI
FC-PBGA	Flip Chip Plastic Ball Grid Array
GPU	graphicals processing unit
HCSL	High-speed Current Steering Logic
HDA	High Definition Audio
HDMI	High Definition Media Interface
I2C	Inter Integrated Circuit
I2S	Inter-IC Sound
LDO	Low-Dropout Regulator
LPDDR	low power double data rate
MAC	Medium Access Control
MDI	Media Dependent Interface
MIPI	Mobile Industry processor interface
OpenGL	Open Graphics Library

PCB Printed Circuit Board

PCI Peripheral Component Interconnect

PCIe PCI Express

PMIC Power Management Integrated Circuit

RGMII Reduced Gigabit Media Independent Interface

RTC Real Time Clock

SAE Society of Automotive Engineers

SAI Serail Audio Interface

SATA Serial AT Attachment

SCU System Controller Unit

SD Secure Digital

SGET Standard Group for Embedded Technologies

SMARC Smart Mobility Architecture

SNVS Secure Non Volatile Storage

SPDIF Sony/Philips Digital Interface Format

SRIS Separate Refclk Independent Spread

SSC Spread Spectrum Clock

UART Universal Asynchronous Receiver-Transmitter

USB Universal Serial Bus

USHDC Ultra Secured Digital Host Controller

VPU video processing unit

Chapter 1

Introduction

The thesis aid is to design a board for developing and testing Advanced Driving Assistance Systems (ADAS). This work is part of the European Project *"Autodrive"* which is composed of many leading companies in the automotive sector. Among them is *Ideas&Motion*, the company where the thesis work took place. The board has been split into two boards during design phase: a reusable general purpose which performs the most computation-intensive tasks, and an application-specific one that powers and provides interface to the other one, while performing fail-safe operations. This thesis focuses on the first, general purpose, board.

The Board must use the NXP processor i.MX8 Quad Max as requested by other partners in the project and the hosting company wants to design a reusable System On Module. For this last reason, it has been decided to design a board compliant with the Smart Mobility ARChitecture (SMARC) standard.

1.1 Thesis structure

The thesis is divided as follow:

- This first chapter presents general concepts about ADAS systems and autonomous levels.
- In the following chapter the main standards and interfaces studied are described briefly: the focus is often on the physical layer.
- Then, starting from the requirements, the list of the used components is presented.
- After that, a chapter covers specific aspects of the designed system and how components has been connected together.

1.2 Hosting company: Ideas&Motion

The design of the whole development platform has been assigned to Ideas&Motion, and the thesis work (regarding only the System On Module) has been performed in collaboration with their designers. It is a small Italian company that offers engineering services in the automotive sector and also produces small volumes of Electronic Control Units for niche applications, as well as Intellectual Property cores for integrated circuits. Despite being small company, they work and collaborate for many of the leader of the Automotive sector.

1.3 ADAS and Autonomous car

The automotive industry is currently investing in the field of Advanced Driving Assistance Systems. These are systems that aid the human driver with information about the environment or even takes actions on part of the vehicle, like the throttle or brakes. These systems are already commercially available in newest cars, like Front Collision Warning or Adaptive Cruise Control (ACC).

The market is going toward the self-driving car through developement of systems that enable the vehicle to work with partial human supervision in particular conditions. A completely autonomous system is an achievement still to reach.

The Society of Automotive Engineers (SAE Internationl) has defined six levels of automation of a systems. They go from 0, absence of any automation, to 5, that is a completely reliable system in any driving circumstance that does not need the human assistance.

- level 0: the vehicle may inform the driver, but all actions are made by the human. In this category falls every assistance that warn the driver without taking any action.

- level 1: the system may perform some actions on the either the steering wheel or the brake. This is just an assistance and the driver is driving most of the time. An example is the Adaptive Cruise Control, where the car manage the speed while measuring the distance from the next vehicle.

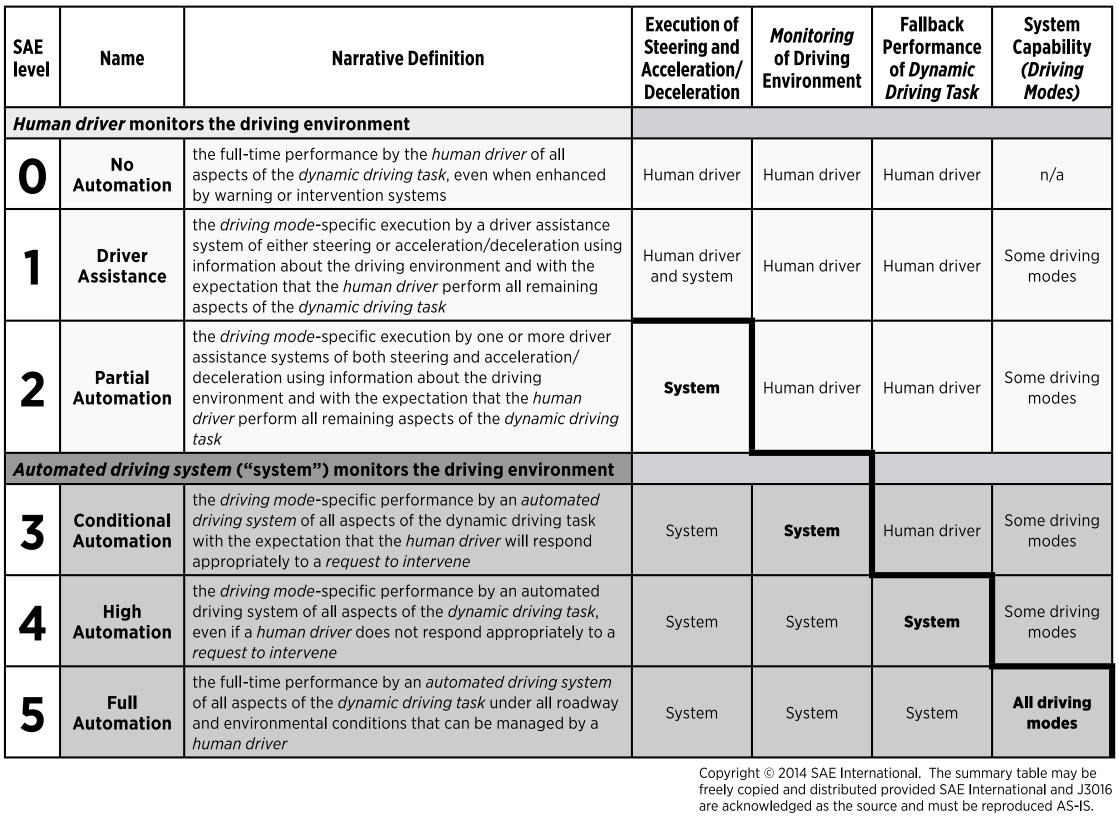
- level 2: the system can maneuver the steering wheel and brakes in particular conditions, while the driver must be vigilant and monitor the environment.

- level 3: the system is able to operate reliably in certain conditions while monitoring the driving environment for possible dangers. The driver may be noticed to take action within a certain time, but he does not have to be focused on the environment the whole time.

- level 4: the system has control and could operate without the driver except from some particular situations, where the intervention is needed. The driver may still control the vehicle.

- level 5: full automation, where the system is 100% reliable and the steering wheel and pedals could be removed, since the vehicle is autonomous in every condition.

Nowadays the level 3 and 4 are being studied and experimented. This kind of application is increasing the demand in automotive systems for high-speed interfaces, able to transmit large amount of data from cameras and radar to the ECU. In fact, autonomous cars need several camera to get data from the environment. In fact, the number of sensor inside a vehicle is increasing, resulting in a need for a higher bandwidth.



SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	**No Automation**	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	**Driver Assistance**	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	**Partial Automation**	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	**Conditional Automation**	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	**High Automation**	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	**Full Automation**	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figure 1.1: Autonomous levels. Image taken from [19]

1.4 Autodrive: European Project

Autodrive is a broad european project with the aim to gather companies and universities with the purpose to develop autonomous driving systems. It has the long term goal to develop level 5 systems like an autonomous shuttle bus. In the middle there are several shorter term goals that will bring to that final system.

The thesis work takes part of the project and its purpose is to design a developing platform for another company part of the Autodrive project. The system will be placed in a testing vehicle with all needed cameras and radars for the development of several kinds of ADAS systems. So it needs to interface with all sensors, be able to exchange a large quantity of data, process it and then safely take action. The system will be the base for developing algorithms and the application software.

Chapter 2

Protocols and Standards

Here is presented a quick overview of the main standard interfaces present in the project along with other standards. Because of the type of project, the main focus will be on the physical layer of these interfaces.

2.1 SMARC

In the early stages of the design, the system has been chosen to be compliant with SMARC form factor standard. From the specification [23]:

> SMARC (Smart Mobility Architecture) is a computer Module standard maintained by the SGeT (Standardization Group for Embedded Technologies). SMARC Modules are small form factor (82mm x 50mm and 82mm x 80mm), low power (typically <6W) computer Modules that are used on a Carrier board that utilizes a 314 pin 0.5mm pitch right-angle memory socket style connector to host the Module

In other words, this defines a System on Module (SoM) which can act as a stand alone PC, that can be inserted on a carrier board with the appropriate connector. The standard defines the dimensions of the PCB together with the interfaces to the carrier board.

These modules can be bought and inserted on a application-specific board designed for the particular project. The standard allows interchangeability between modules.

2.1.1 Specifications

The SGeT defines several aspects of the module. Some of them are completely mandatory, other are optional and model-dependent.

The most important definition is probably about the form factor and connector: the board must be 82 mmx50 mm and must be compatible with MXM3 connector. This is a standard socket typically used for Graphic Processing Unit (GPU) cards, but the SMARC defines a different pin assignment.

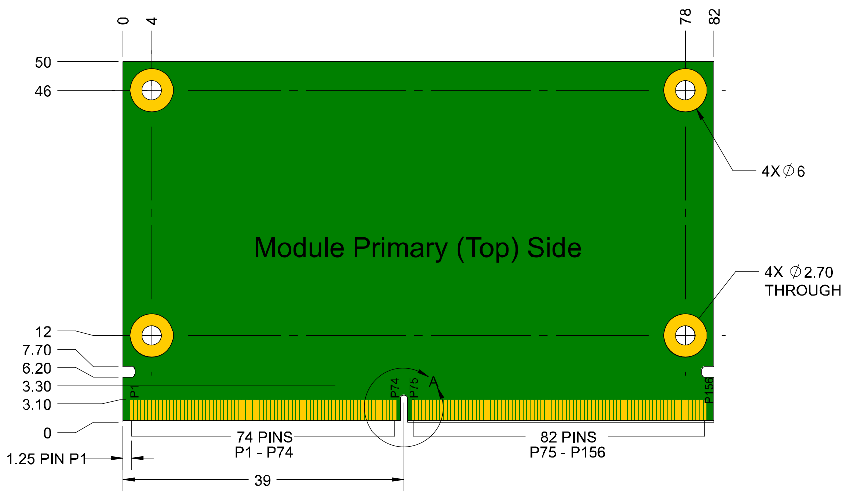


Figure 2.1: Drawing of the SMARC form factor

The possible interfaces are already all defined on specific pins. Not all channels have to be implemented, some of them are optional. Here a list of them is presented:

- LVDS;
- HDMI/Display Port;
- 2x MIPI CSI;
- SDIO;
- SPI;
- I2S/HDA;
- I2C;
- UART;
- CAN;
- 5xUSB (of which 2 can be USB 3.0)
- PCIe;

- Gigabit Ethernet;
- SATA.

It is not possible to assign a new purpose to unused pins, so they are left unconnected: violating the standard means to not guarantee compatibility. This is a quite complete set of the commonly used interface, including some high-speed one like PCIe and Ethernet. The exact pin assignment can be read in the specification document [23].

The communication between the module and the carrier along with the power pass over the edge connector, which on the module side consists in only gold-plated contacts directly on the PCB. The module is inserted in the socket parallel to the carrier, resulting in a low-profile solution.

The power is delivered at a voltage between 3.0 V and 5.25 V, letting the module to regulate the power based on its own needs. The module should tolerate the whole range, but it is allowed to be functional only in a subset of that range.

2.2 PCIe

PCI Express is an evolution of the old Peripheral Component Interconnect bus. It abandons the parallel structure in favor of a point-to-point high speed serial connection. Its generality made it popular in a variety of fields such as consumer, server and industrial applications.

2.2.1 Structure

As already stated, the protocol consists in a point-to-point serial connection. This removes the constraints and complexities about time arrival of single lines within the bus, also reducing the number of pins needed. The link may consist of multiple lanes, which is a couple of simplex differential pairs. Thanks to its low voltage levels, it reaches higher speeds while keeping a low power profile. If larger bandwidth is needed, more lanes can be used in a link.

A PCIe Switch can be used to split a link into multiple ones. The resulting network is hierarchical, where the Root Complex is the node that communicates with the central CPU and all other nodes are Endpoints.

On the software side PCIe is very similar to its predecessors PCI, in fact it maintains software compatibility with it.

The lowest bit rate is 2.5 Gb/s, while Generation 3.0 introduce 8.0 Gb/s per lane. As already said, using multiple lanes largely increases the bandwidth.

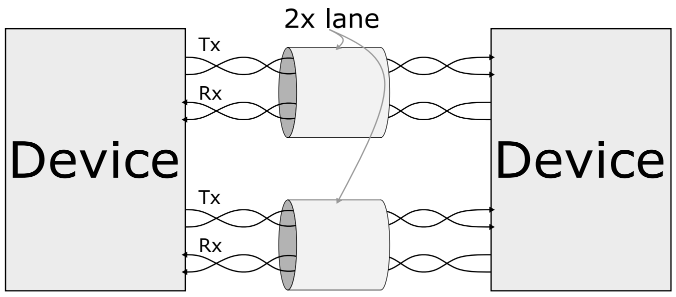


Figure 2.2: Structure of a 2xlane link.

2.2.2 Clock structure

The PCIe exploit a 8b/10b coding. This increases the number of bit transmitted, but ensures a higher number of transitions so allowing the receiver to recover the clock embedded in the data. When working at 8.0 Gb/s the encoding scheme is 128b/130b. This reduce the encoding overhead.

The PCIe needs a 100 MHz clock as reference and allows three topologies:

Common clock where the same clock is fed to both transmitter and receiver, which has an easier job to recover the clock from the data. This configuration is the most common, but can be difficult to distribute the clock over long distances.

Data clock where the clock is fed only to the transmitter, while the receiver has to rely solely on the data transitions. Thus, the receiver is more complex.

Separate clock where both the transmitter and receiver have their own reference clock. This avoid the distribution of the clock, but imposes more stringent constraints on the precision of the clock frequency and jitter.

2.2.3 Spread Spectrum Clock

The PCIe also support Spread Spectrum Clock (SSC). This technique consists in introducing a wanted jitter on the clock: this leads to spectrum more distributed and with a lower peak. This is a benefit from the Electromagnetic Interference point of view: in fact, the circuit will radiates on a wider spectrum, but the peak will be much lower.[10, 21]

SSC can be easily implemented in Common Clock and Data clock configurations, because the jitter is present at both transmitter and receiver; while in Separate clock configuration the implementation is not trivial. In this case it is called Separate Refclk Independent Spread (SRIS): two clocks are used, each with

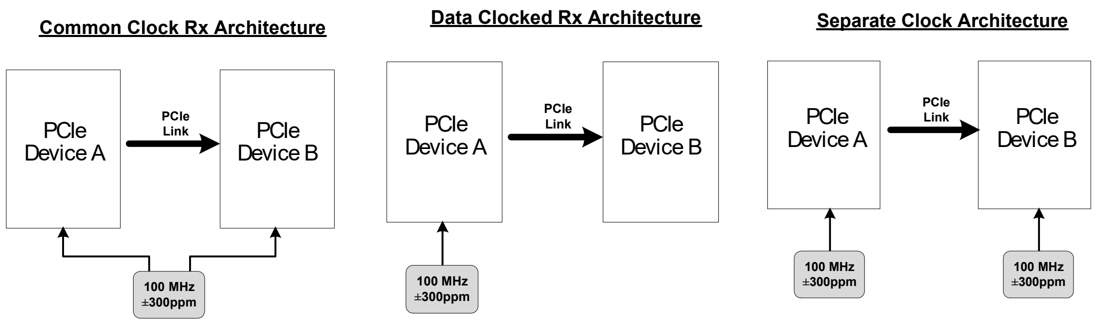

Common Clock Rx Architecture
Data Clocked Rx Architecture
Separate Clock Architecture
PCIe Device A
PCIe Device B
PCIe Link
100 MHz ±300ppm

Figure 2.3: Three clock topologies of PCIe

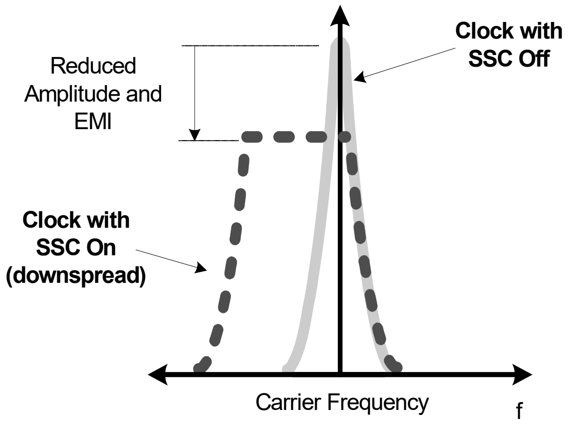


Figure 2.4: Spectral content of SSC

his own SSC. This further complicates the receiver, that should be able to recover that data despite the different frequencies.

2.2.4 Physical Layer

The serial nature of PCIe simplifies layout work: in fact, every lane can be be routed independently, without matching the length of all of them. Every lane is a pair with differential impedance of 100 Ω. Capacitors must be placed in series with Tx lanes, in order to get an AC coupled signal. Another useful feature is the Polarity inversion of the lines: it is possible to swap the two signals within a differential pair, thus avoiding vias during routing phase. Before starting communication, the PCIe controller perfrom a link training, so configuring the link width, the line polarity and performing lane-to-lane de-skew.

2.3 SATA

The SATA (Serial AT Attachment) interface is a bus used to interface a computer to a storage device. At the physical level is a point to point full duplex connection, with 100 Ω differential lines. It needs AC coupling capacitors on signals. The encoding is 8b/10b, that is 8 bits are mapped to 10 bits. Also in this case there is the possibility to use Spread Spectrum Clock.

During power on, Out Of Band signaling is used to configure the link. In this situation the differential line is not driven differentiallly but they are driven with the same voltage.

2.4 USB

Universal Serial Bus is a well known interface used in consumer electronics for a wide variety of applications, typically for PC-peripheral communication. It allows to access multiple devices using hubs, creating a tiered-star topology: one single host connected to several hubs and devices. At most there can be 5 hubs between the Host and the device, thus resulting in 7 tiers (where Tier 1 is the Host, Tier 7 is the device).

2.4.1 Physical Layer

Up to version 3.0, the interface relied solely on a half-duplex differential pair (composed by signals often called D+ and D-). All communications are managed by the host computer, and only one device at a time can communicate.

It should have a differential impedance of 90 Ω. The clock is embedded in the data, featuring bit stuffing to ensure enough transitions. In the cable are also carried GND and a 5 V VBUS, that can feed up to 500 mA to the devices.

The 3.0 standard changed the number of wires, introducing two differential pair, dedicated to transmitting and receiving thus reaching full-duplex operation (Tx+, Tx-, Rx+, Rx-). Since they are not directly retro-compatible with previous versions, inside the cable are also present older signals (D+ and D -) that are also used for configuration.

The several versions defines speed grades, shown in Tab. 2.1.

2.5 LPDDR4

Mobile devices has to cope with a trade-off between power savings and high performances. Together with the higher and higher media-orientation of new devices, there is the need for large, fast and low power memory. Because of that, JEDEC

Table 2.1: USB speeds

Name	Speed [Mbit/s]	Version
Half Speed	1.5	1.0
Full Speed	12	1.0
High Speed	480	2.0
Super Speed	5000	3.0

has defined a new branch of memory standards in order to satisfy this demand: the Low Power Double Data Rate SDRAM.

It specifies a class of devices with standard performances, behavior and interface.

2.5.1 Architecture

One of the first differences with previous kinds of memory is the architecture of the channels. LPDDR3 in fact had one command bus and one 32 bit data bus. The new structure consist in splitting the data bus in two 16x buses and duplicating command buses, so that there are two channels, each one with their own data and command. Therefore the die is split into two almost identical replicas of the memory. [9]

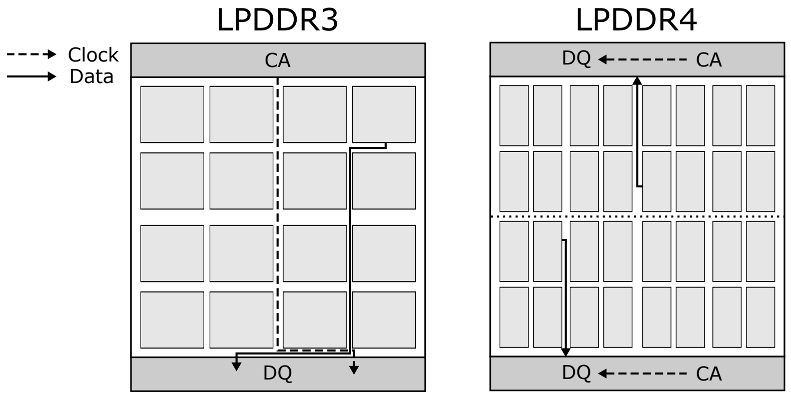


Figure 2.5: Comparison of the architecture between LPDDR3 and LPDDR4. CA is the command and address bus, while DQS is the data strobe.

This choice brings several improvements. First of all, the word and bit line are shorter since they do not have to cover the whole die: thus the amount of charge need to change voltage is smaller. Because of this, the performance are improved while keeping low power consumption. Moreover having the strobe (DQS) and clock pins on the same side reduces the power consuption: in fact, DQS is generated starting from the clock signal.

The two channels can be turned off independently, so giving the possibility to save power if one of them is not used for long periods.

DMI signal (Data Mask inversion) was previously used only for data mask, but now it has a double role.

Data mask is a useful feature to cope with the burst oriented nature of the device. It allows the controller to tell the RAM to discard some of the bytes sent during the write burst. Writing not contiguous bytes create problem about parity bits used for Error Correction Code: they are calculated over the entire range present in memory, including the bits not overwritten. In previous version, this task was done by reading the range, modifying the wanted bits, recalculating the ECC and lastly writing the whole of it. In LPDDR4, the memory itself recalculate the new ECC even though some bytes have to be excluded, thus saving transitions on bus.

Bus inversion allows to send an inverted replica of the data, and DMI signal is used to inform the receiver that the inversion has been made. If the number of bits equal to 1 is greater than 4, the RAM or the controller will send the inverted data. This minimizes the number of 1s on the lines, thus reducing the static power dissipated. In fact, the On Die Termination present on DQ pins is connected to VSS, so current will flow through it only when the voltage is HIGH.

Like modern memories, this device features link training: it consists in a phase where the controller and the memory collaborates in order to find the optimal configurations for reference voltage, delays, signal strength and terminations. A new feature regards the frequency set points. In fact, the LPDDR4 have a duplicated set of configuration registers that can be used to store trained parameter of a higher frequency configuration. This allow to change configuration run-time without passing in an untrained state and possibly loosing the link. LPDDR4 are designed for battery powered systems, so the possibility to switch between a low-power and a high-performance configuration is quite important.

2.6 HDMI

HDMI is a stanrdard interface used to send uncompressed video and audio data from a device to another. They are respectively called Source and Sink. An example is a PC as Source and a display as Sink.

At the core, it consists in 4 differential pair compliant with TMDS (Transition Minimized Differential Signaling), but inside the cable there are also auxiliary interfaces.

2.6.1 Architecture

Among the secondary interfaces there is an I2C interface called DDC (Digital Data Control) used by the source to read configuration data from a ROM mounted on the Sink. For instance, the memory stores the supported frequencies, resolutions and data formats.

Another optional part of the interface is the Consumer Electronic Control (CEC), a one-wire bidirectional connection between devices. This allows the user to control through the remote all the devices connected with HDMI cables.

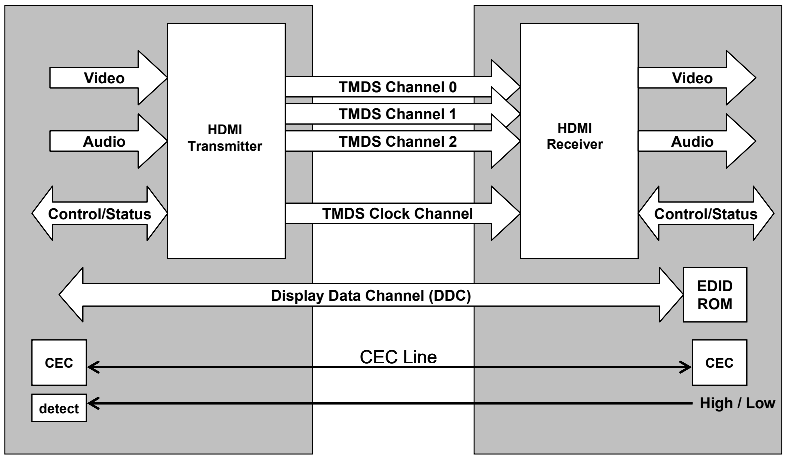


Figure 2.6: HDMI structure

The video and audio data is sent through the 4 differential pairs. One of them is used as pixel clock, providing a timing reference. During one pixel clock cycle, 10 bits are sent on each one of the three remaining pairs, thus sending 30 bits. As subsequently explained, data is actually encoded from 8 bits to 10, so the actual number of bits is 24.

The Source also provides a 5 V power line, from which the Sink can absorb at most 500 mA.

2.6.2 TMDS

TMDS stands for Transition Minimized Differential Signal, and it is the physical layer upon which HDMI is based. It is composed by a low voltage differential pair. The driver is a current steering device, while the receiver has terminations to 3.3V, as shown in Fig. 2.7.

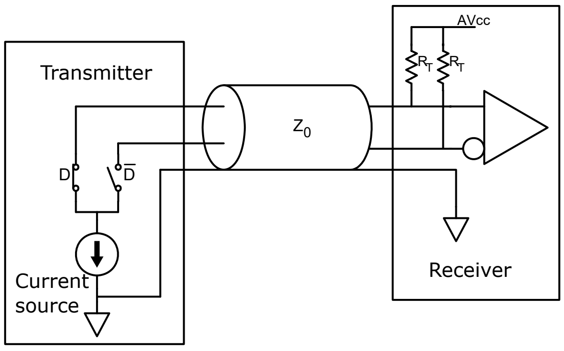


Figure 2.7: TMDS architecture

The data is not sent as is, but it is encoded with a particular 8b/10b scheme(q is the encoded symbol, while D is the original byte):

- the LSB $D[0]$ is not modified;
- the following fields is the result of the XOR or XNOR with the previous bit of that byte, in other words $q[i] = D[i]$ XOR $D[i-1]$. The operation is decided in order to minimize transitions, and the 9th bit signals which operation has been made;
- based on the previous bytes sent, the driver can decide to invert the whole symbol $q[0:8]$ and use $q[9]$ to signal the inversion.

This type of encoding accomplishes two purposes: the first one is reducing the total number of transitions on the line, and it is done through the application of XOR/XNOR functions. Less switching means less eletrco-magnetic radiation. The other purpose is to try to balance the long term number of 1s and 0s sent, i.e. to reduce the *running disparity*. Through the inversion of the bus, the scheme ensure a low running disparity. Having signals with almost no DC component reduce the Inter Symbolic Interference since, otherwise, during a long series of equal bits the lanes would get charged/discharged. [6]

2.6.3 Resolutions and frequencies

Depending on the link, several bit clocks are supported. This depends on both Sink and Source, as well as the cable used. Given a particular bandwidth, several formats and resolutions can be supported.

There are several factors that affect the needed bandwidth. The first one is resolution: this is the total number of pixels that make up a frame of the video. The second one is the color depth: for each pixel the color is encoded into a fixed number of bits, that can vary from 24 to 36 bits. Moreover, the color scheme has to be chosen: RGB (Red, Green, Blue) and YC_BC_R (Luminance, Change of blue, Change of red) are supported. The last factor is the frame rate, that consists in how many frames are sent every second.

So the total number of bits sent every second is:

$$resolution \cdot colorDepth \cdot frameRate$$

This workload is shared between the 3 data TMDS channels. For instance, with a resolution of 1280x720, a color depth of 24 bits and a frame rate of 30 Hz, the total necessary bandwith is $(1280 \cdot 720) \cdot 24 \cdot 30 = 663\,\mathrm{MHz}$.

2.7 Ethernet

Ethernet is a collection of IEEE standards for Local Area Network. It defines several the lowest layers of the OSI (Open Model Interface) stack, from the Physical to the Data Link layer.

2.7.1 Physical layers

The Ethernet can be transported over several kind of interconnections, from copper cables to optical fiber. The most common ones in consumer applications are 10BASE-T, 100BASE-T and 1000BASE-T, where the first digits tell the overall speed in Megabit/s. They are typically coupled with the RJ-45 standard plug. They all use 4 AC coupled twisted pair.

In the automotive sector 100BASE-T1 has become a popular solution. It can send 100 Mb/s over a single unshielded twisted pair. It keeps low costs while tolerating harsh environment and coping with stringent Electromagnetic Compatibility constraints. It employs PAM3, that is three voltage levels are allowed. In automotive applications 100BASE-T1 can be used to deliver many kinds of data across the vehicle.

2.7.2 MDI and MII

An Ethernet transceiver is composed by a MAC (Medium Access Control) and a PHY (Physical Medium Transceiver). The first one is the controller that manage the second layer of the OSI stack so controlling the access to the shared medium, while the second one is the transceiver that translate the data to the right voltage levels and the applying the needed encoding.

The two of them communicates with an 8 bit bi-directional bus called MII: Media Independent Interface. As the name says, it allows the exchange of data between MAC and PHY regardless of the physical layer used.

The PHY generates a set of signals called MDI, that is a general name indicating the communication over the cable. These signals are sent to a magnetic transformer that electrically insulates the cable from the board. This brings robustness from insulation, together with the common mode rejection needed for differential signaling.

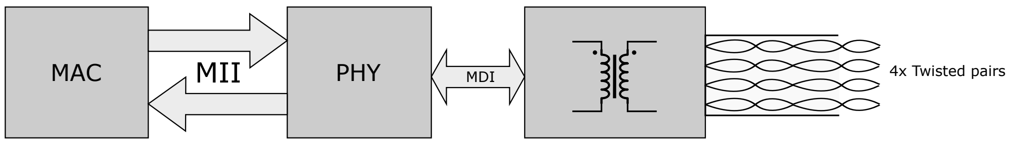


Figure 2.8: Block diagram of an Ethernet interface

MII family

The MII is composed of 8 data signals: 4 for transmission and 4 for receiving, at a frequency of 25 MHz, resulting in a total bandwith of 100Mb/s. Another 8 signal are used for control, so the interface comprehend 16 bits.

A new version of MII has been introduced in order to reduce the pin count of MACs. It is called Reduced Media Independent Interface (RMII) and has only 2 couples of data signal and 3 control signals. Since the number of lines has been halved, the signaling frequency has been doubled up to 50 MHz in order to maintain the same performances.

The need for Gigabit Ethernet has brought to the born of another two interfaces: Gigabit Media Independent Interface (GMII) and Reduced Gigabit Media Independent Interface (RGMII). The first one uses 8 transmitting and 8 receiving signals at the frequency of 125 MHz. The second one instead uses 4 transmitting and 4 receiving lanes, while working at the same frequency: this can be obtained thanks to Double Data Rate signaling (thus sending a new data on every rising and falling edge of the clock). [3]

2.7.3 MDIO/MDC

There is another interface between the MAC and PHY called MDIO/MDC. It is used to write and read all configuration and status register of the transceiver. It is a two wire interface able to connect up to 32 devices, each one with a unique address. It is composed by `MDC` that is the clock and `MDIO` that is the bidirectional data signal. The data rate is 2.5 MHz. The standard defines a 1.5 kΩ pull-up on `MDIO`, and the drivers must be open drain ouptuts.

2.8 SD Card

Secure Digital Card are non volatile memory cards typically used in portable system as removable storage.

The standard defines the physical form factor as well as the interface bus. The card exposes 9 contacts, that can be used in two way: SPI interface or SD interface.

The SD interface uses 4 bidirectional signals for passing data together with two pins for timing and for commands: they are called CLK and CMD. The Host may also start the communication using the SPI interface instead of the SD bus. This is seleted by driving low `DAT3` at power on.

Table 2.2: Signals of an SD card. I: input, O: output, S: supply.

Pin #	SD Mode			SPI mode		
	Name	Type	Description	Name	Type	Description
1	CD/DAT3	I/O	Card Detect/Data [Bit 3]	$\overline{\text{CS}}$	I	Chip Select
2	CMD	I/O	Command/Response	DI	I	Data In
3	VSS1	S	Ground	VSS	S	Ground
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS2	S	Ground	VSS2	S	Ground
7	DAT0	I/O	Data [Bit 0]	DO	O	Data Out
8	DAT1	I/O	Data [Bit 1]	RSV		
9	DAT2	I/O	Data [Bit 2]	RSV		

2.8.1 SD Family

Starting from base specifications, the standard has been extend to comprehend larger capacities. So, today there are SDHC (SD High capacity) and SDXC (SD Extended Capacity). Also, there are label for speed performances, like UHS (Ultra High Speed).

Starting from UHS-I, the card has the possibility to switch to a 1.8V signaling level in order to increase performances. So, after an initial negotiation with the host at 3.3V, both the devices switch their voltage levels to 1.8V, while the SD card is still powered by 3.3V.

Table 2.3: Capacity and speed of several versions

SD version	Capacity
SDSC	128MB - 2GB
SDHC	4GB - 32GB
SDXC	64GB - 2TB

Speed Class	Speed
Standard	25MB/s
UHS-I	104MB/s
UHS-II	312MB/s

2.9 MIPI CSI

The Camera Serial Interface is especially aided to the transmission of raw data from a camera to a microprocessor. It is maintained by Mobile Industry Processor Interface and has become quite popular in mobile systems. The communication can rely on several Physical layers also defined by MIPI, that differs on the topology of the connection: the number of lines, the presence of clock signals and the type of signal (single ended or differential).

Like for HDMI, for each pixel several bits have to be sent, depending on the color scheme. Also RAW data can be sent, before applying Bayer Filter. [1]

2.9.1 D-PHY

One of the most popular physical layer is the D-PHY. It relies on a scalable number of differential lanes (1, 2 or 4). The data is source synchronous, meaning that on another differential pair the clock is sent from the source. Since we are talking about a camera, the clock and the data are sent from the camera itself toward the processor.

One peculiar feature of this link is that it has two possible state: the first one is the High-Speed state, where the lines are driven by differential drivers with an amplitude of 200 mV. The other state is Low-Power, where instead the the differential pair is driven as two independent single ended signal, with a voltage level of 1.2 V and they are not terminated. This state is used for control data since it can send only 10 Mb/s, while the main data is sent in High-Speed state reaching a speed of 1 Gb/s per line. Since in some case this physical layer could have bidirectional communication, a Contention Detector is used to detect collisions while sending data in low power state. [2]

2.9.2 MIPI DSI

The MIPI Alliance defines another standard called Display Serial Interface. It serves the purpose of sending data from the processor to a display. While CSI and

DSI may share the same physical layer, they differ in the protocol one, being two independent standards.

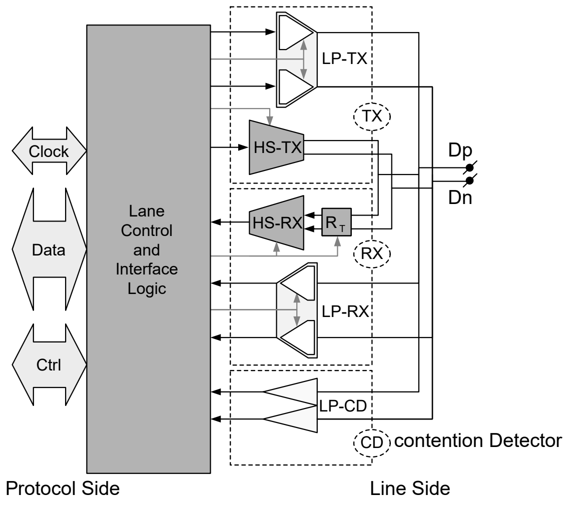


Figure 2.9: D-PHY transceiver structure

2.10 I2S

The *Inter-IC Sound* bus is a serial bus interconnection intended for sending and receiving audio data in a digital format. The basic mono-directional architecture is composed of 3 wires: the `SCK` is the bit-clock, `WS` stands for word select and `SD` for serial data. The `WS` level indicates which audio channel is being transmitted: when low, the data is related to left channel; when high the data is related to right channel. The data is clocked by the `SCK` signal and it changes on the falling edge.

This bus is intended for always transmitting data at a constant rate, between integrated circuits mounted on the same board. For instance, it can be used to pass data from a codec to an audio Digital-Analog Converter.

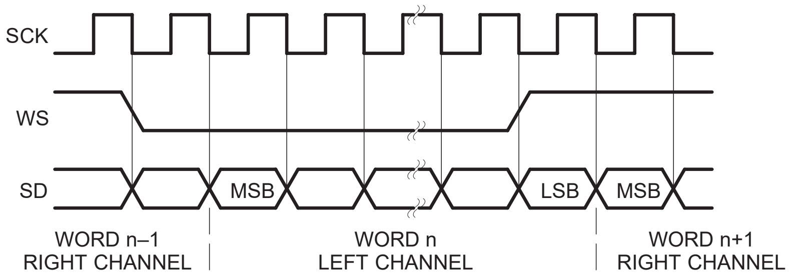


Figure 2.10: I2S timing diagram.

2.10.1 SAI and ESAI

The Synchronous Audio Interface is a peripheral of the iMX8 able to transmit audio data in several formats and frames. Among the others, there is I2S. This peripherals has 6 total pins: 3 for transmitting data and 3 for receiving. The three signals are `SYNC` (frame sync), `BCLK` (bit clock) and `DAT` (data). Those pins can be driven in order to support I2S, where `SYNC` is connected to `WS`. Moreover, there is the possibility to have receiver and transmitter working synchronously, i.e. using only one pair of control signals. This can be done if the input and output data have the same sampling frequency.

The iMX8 also has two *enhanced SAI*: it basically a SAI but with six total data signals. Those can be configured as both input and output.

Chapter 3

Components choice

This chapter focus is about the early design choices and a brief description of the most important components chosen is provided.

3.1 Requirement analysis

The Autodrive project require the development of a prototyping platform for ADAS Systems. It must be able to acquire data from several cameras, be able to communicate on a CAN network and it must have high-speed interfaces like Ethernet. The proposed structure is based on the separation of the system in two regions: the first one should get and process the data, while the second one should be the fail-safe part. This region is devoted to decision taking and should have a high safety level.

This work is focused on the data processing part. The iMX8 has been suggested as microcontroller, due to his large number of interfaces, powerful cores and GPU together with its automotive grade.

Moreover, it has been chosen to design this part of the system like a separate physical board. In the early stages of the project, it has been chosen to design a board compliant with SMARC standard. The other board, devoted to safe decision making, would host the SMARC module and has been design by another designer of the company Ideas&Motion.

Since a complete SMARC module based one the iMX8 would satisfy the requirement imposed, the new focus of the design has become the fulfillment of the standard. In fact, many interfaces would probably be unused in the Autodrive project, but they have been designed in order to get a complete System On Module. This is due to the whish of the company to get a module which could be reused in other projects with similar requirements.

In conclusion, the aim has been to fill as many interfaces of the SMARC module as possible, trying to use as much as possible of the iMX8 peripherals.

3.2 iMX8 Processor

At the core of the project is the NXP processor called iMX8. It has various multimedia capabilities that along with automotive grade features makes it suitable for infotainment applications. Beside that, it can also be used in industrial environment or in a single-board computer.[12, 14]

3.2.1 Overview

A general overview of the processor is given here:

- six ARM core architecture;
- two Graphical Processing Units (GPU) compatible with OpenGL library API;
- a Video Processing Unit (VPU) for video encoding and decoding;
- two controller for LPDDR4 SDRAM;
- SDIO 3.0 and eMMC 5.0 for storage interface;
- Quad-SPI;
- Display controller able to drive 1x4K or 4x Full HD displays;
- MIPI DSI, HDMI(Tx)/eDP and LVDS channel used to interface with displays;
- HDMI(Rx) and MIPI CSI in order to receive data from cameras;
- PCIe gen. 3.0;
- SATA 3.0;
- USB 3.0 and 2.0;
- two GigaBit Ethernet MAC RGMII;
- UART and I2C;
- SPDIF, SAI and ESAI for audio signals;
- CAN interfaces;

It is clear how the peripherals are mostly media-oriented. Also, many of the interfaces includes high frequency signals (greater than 1 GHz).

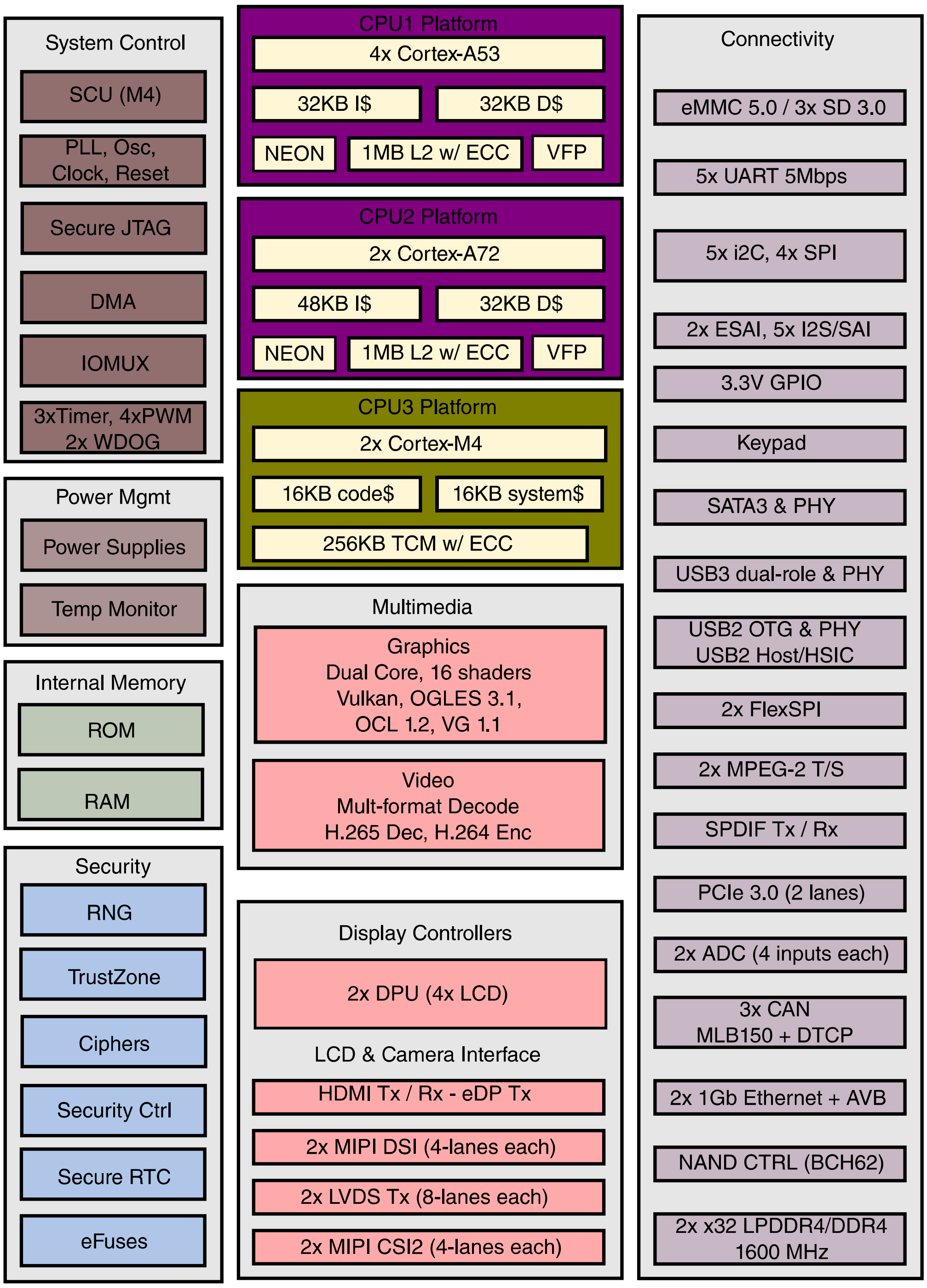


Figure 3.1: Simplified block diagram taken from iMX8 datasheet [12]

3.2.2 ARM core

The processor is based on a multi-core architecture. All cores belongs to ARM-Cortex family. [13] In particular:

- 2x Cortex-A72 cores, used for compute-intensive tasks;
- 4x Cortex-A54 cores, used during low-power states;
- 2x Cortex-M4 cores, used to manage low-power general-purpose peripherals.
- a System Controller, providing an abstraction to the hardware peripherals, along with power and boot management.

These features allow the iMX8 to run a complete operating system, like Linux, Android or FreeRTOS.

Clusters

A set of homogeneous cores is called cluster. In this case there are two clusters: one composed by A72 and one by A54.

Each cluster can perform individual voltage and frequency scaling, while all the clusters can work together as a single multi-core processor. A L2 cache is shared among the cores in a cluster. Processes can be scheduled across cores of different types.

Many other features are omitted, since they fall out of the scope of the thesis.

3.2.3 Package

The package is labeled as FC-PBGA that means "Flip Chip Plastic Ball Grid Array". This means the die is flipped and soldered with his active surface on a plastic substrate. This is an alternative to the typical wire-bonding technique. It brings several advantages: the first is the possibility to place contact on the whole surface of the die, in contrast to wire-bonding that needs the contacts on the edges: this allows a higher density of contacts. Moreover, the lack of a long wire reduces the parasitic inductance, resulting in better high-frequency performances. Finally, the passive side can be used to transport heat to a heat-spreader, thus leading to a low junction to ambient thermal resistance.[11]

A plastic substrate is used to route signals from the die to the package contacts, connecting them through internal metal connections.

The contacts themselves are actually just solder bumps placed on the whole surface of the substrate (this is the same concept seen before, between die and substrate). During the soldering process, the board is heated, so that bumps melt and create connections with the underlying pad, placed on the PCB.

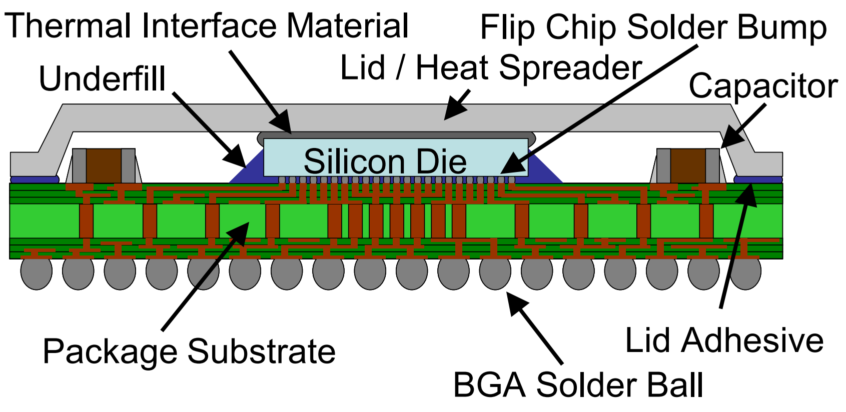


Figure 3.2: Section of FCPBGA. Image taken from NXP document.[11]

Collapsing and non collapsing connections

The contact between the bump and the pad can be classified into two groups.

The first one, called *non collapsing*, is the result of a Solder Mask Defined Pad, where the metal pad is larger than the soldermask opening. In this case the solder is touching only the top surface.

In the second case, the solder can create a bonding also with the side of the metal pad, since the solder mask opening is larger than the pad (non Solder Mask Defined). A section of the two can be seen in Fig. 3.3.

These two types have some pros and cons. The **collapsing contact** is typically used with pin pitch greater or equal to 0.65 mm and bring a larger contact area with the pad. Moreover, there is more space to route line between pads. On the other hand, tests show that this type of contact are mechanically weaker than the other type: in fact, the copper pad may detach from the substrate in case of shock. The **non-collapsing contact** instead has less contact area, while it is more mechanically robust since the solder mask overlap increases the pad adhesion to the PCB. This type is suitable for component with smaller pin-pitch where via in pad may be needed: in fact the larger copper area can suit more easily the via.

3.3 USB Hub

The iMX8 has one USB 2.0 OTG interface and a USB 3.0 interface (which comprehends also a 2.0 bus); it also has another 2.0 controller connected to a HSIC (High Speed Inter Chip) interface: a single low voltage differential line intended for on-board communication. The SMARC module instead has 6 interfaces, two of which that are capable of 3.0 operations.

In order to have as many buses as possible, a USB Hub has been chosen to be

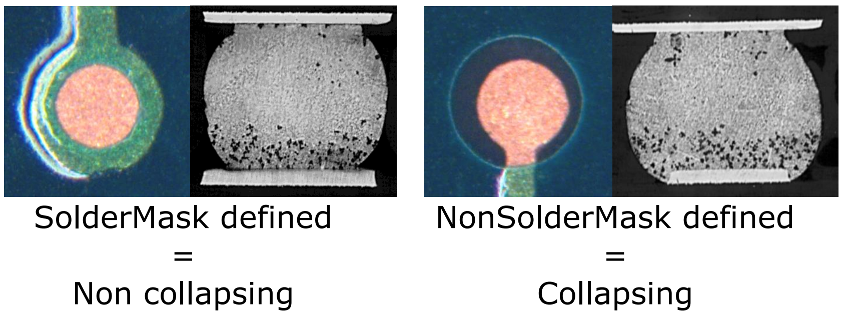


Figure 3.3: Section of bump contact, comparing collapsing and non-collapsing.

Table 3.1: SMARC USB capabilities

Name	USB 2.0	USB 3.0	OTG
USB0	x		x
USB1	x		
USB2	x	x	
USB3	x	x	x
USB4	x		
USB5	x		

inserted. A Hub is an integrated circuit that split a USB port in more ports, or, in other words, convoys many ports into a single one allowing to connect many devices to a single port. For the project the USB5744 has been chosen.

3.3.1 USB Hub: USB5744

The USB5744 by Microchip is a low power USB 3.1 HUB. It has one upstream port (the one connected to the USB controller) and 4 downstream ports, all with 3.1 capabilities. All ports include a USB 2.0 bus for retro-compatibility. It is available in an industrial grade version, able to work from −40 °C to 85 °C. Its configuration register can be written through SMB or SPI. SMB is not directly supported by iMX8, but I2C and SMB are compatible in some conditions. It requires both a 3.3V and a 1.2V power.

For each port there is a pin called PRT_CTLx that has a double role: it can be driven low by the Hub itself in order to disable the VBUS power of this port, but it is also an input, because if it is externally driven low, the Hub understands it as an over-current condition. This functionality can be easily coupled with the main part of switches dedicated to USB power. In fact they typically have an input signal called EN (enable) and an open drain output signal called $\overline{\text{OCS}}$ (over current). They could be connected together and to PRT_CTLx from the Hub. See Fig. 3.4.

A part from a 25 MHz external crystals and its load capacitors, it does not need any external component.

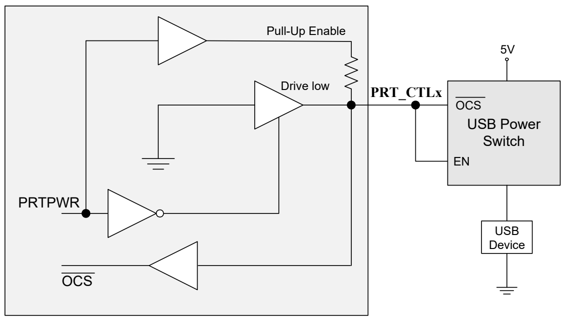


Figure 3.4: Behavior of the PRT_CTLx pin of USB5744

This component has been chosen because of the 3.0 compatibility along with the temperature range.

3.4 Power regulators

The SMARC labels 10 pins as VDD_IN, that is the input power to the module. It ranges from 3 V to 5.25 V depending on the carrier board. It has been decided to work at 5 V because it provides the highest power with a fixed maximum current of 0.5 A per pin. Thus the total current that the module can absorb is 5 A.

The iMX8 itself requires several low voltages to power the ARM cores, the GPU and some other voltages. A part from peripherals with fixed power of 1.8 V or 3.3 V, some part of the module need a dynamic scaling of the voltage depending on the current workload. For instance, `VDD_A72` are the pins powering the two A72 cores; it may have three possible values: 0.9 V, 1 V and 1.1 V. The higher voltages allows a higher clock speed.

Moreover a particular power up sequence has to be respected to avoid damages. All this reasons has led to consider an integrated circuit that NXP sells as specific to power the iMX8 family: the PF8100.

3.4.1 PMIC: PF8100

This component is labeled as Power Management Integrated Circuit, since it is not just a regulator.

It has seven high efficiency buck regulator (with integrated switches) and four linear regulators. The bucks provide up to 2.5 A and the LDO up to 400 mA. It is highly configurable and its configuration registers can be copied into a One Time Programmable memory, that can be loaded at power on. The PMIC can talk to the iMX8 through a I2C interface. Since the configurations are many, NXP sells version that are already programmed to work with a particular microcontroller. In order to power one iMX8 Quad Max you need two different PF8100, each one pre-programmed differently.

In Fig. 3.5 you can see the suggested connection. Some of the buck has been connected as bi-phase buck in order to double the output current.

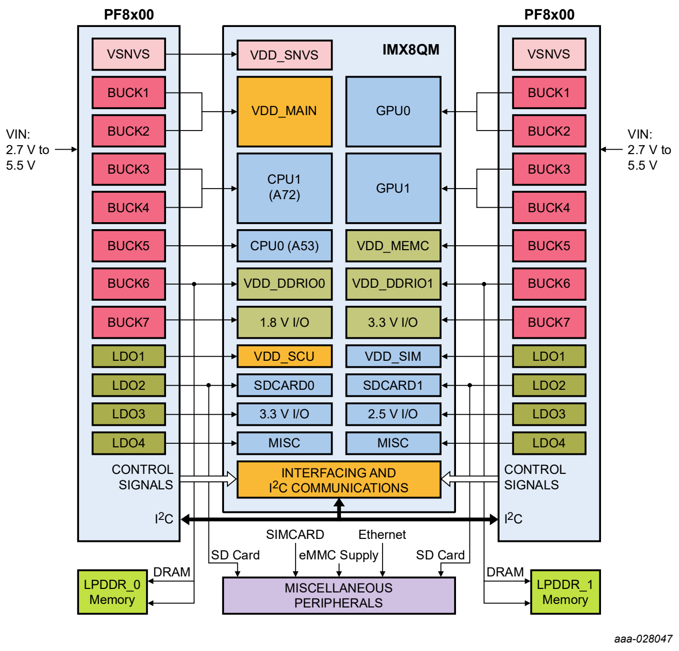


Figure 3.5: Suggested connection of iMX8 and PF8200. Image taken from PF8200 datasheet.

It has many other features as monitoring under voltage and over voltage of the outputs, configurable power on sequencing, a fail-safe state to avoid cyclic failures and other minor features. It features also a power good signal and a reset signal to the micro-controller. The reset can be asserted triggering a watchdog event through an input pin: the input pin WDI triggers a watchdog event, which resets all configuration registers of PMIC and asserts the POR (power on reset) that resets the iMX8. The PMIC has a counter for watchdog events. If it gets over a certain thresholds, the PMIC moves to a fail-safe state and avoid further power-on/reset cycles.

Since this is a complex topic and a mistake could bring permanent damage to the system, it has been chosen to rely on the pre-programmed versions of the device. This reduces the amount of work during the design, and also it removes complexity from the board, because the OTP memory does not need to be programmed on board.

This device can be sold with code PF8200 which has a ASIL B safety level: this means that is safer with respect to failures; in fact it has an improved capability of self-testing.

3.5 Ethernet

The iMX8 Quad Max include 2 Gigabit Ethernet MACs with a RGMII interface. The SMARC standard requires to pass on the connector the MDI signals (the one generated by the PHY), and it provides two Ethernet interfaces.

It is clear that PHYs IC are needed on the module, in order to translate RGMII to MDI. It must be compatible with GigaBit Ethernet and must have RGMII interface.

3.5.1 PHY: KSZ9031RNX

Amongst the commercial products, the KSZ9031RNX has been chosen. It provides the required speed and RGMII interface, supporting 10BASE-T,100BASE-T and 1000BASE-T. It is manufactured by Microchip which also offers an automotive alternative version of the chip. It has a 7x7mm QFN package with 48 pins and an exposed pad for thermal dissipation.

It needs both 3.3 V and 1.2 V to be powered. It provides a pin that can be connected to the gate of an external PMOS in order to form an LDO to regulate 1.2 V from 3.3 V. Also, an external crystal is needed for 25 MHz clock. On the other hand, termination resistor for the MDI differential pairs are embedded on the die.

Several strapping pins are present in order to set the MDIO address, enable all or part of its capabilities and set the LED pins behavior. Strapping pins are read at power on: the PHY detects if there are pull up or pull down resistor connected; after that, the pins are used as normal with their original function.

3.6 Main memory

The iMX8 has two memory controller to communicate with external RAM through DDR4 or LPDDR4. On an evaluation board of the iMX8 by NXP, two of the same kind of LPDDR4 memory were used, with a capacity of 3 GB each. The certainty that this component is compatible with the iMX8, together with the possibility to buy an automotive version of the memory, has lead to the choice of using this component. A total capacity 6 GB should be enough for a wide spectrum of applications.

3.6.1 LPDDR4: MT53B768M32D4

This LPDDR4 DRAM is manufactured by Micron. It is composed of four identical dies with a capacity of 6 Gb each, with a total of 24 Gb, i.e. 3 GB. The device support a clock speed of 1600 MHz and has an automotive grade, so it can work up to 125 °C. As it is possible to see in Fig. 3.6, the dies are coupled in 2 pairs. The two dies inside a pair share the interface pin, and only one of them has On Die Terminations. A part from these specific parameters, the component does not differ from the LPDDR4 standard.

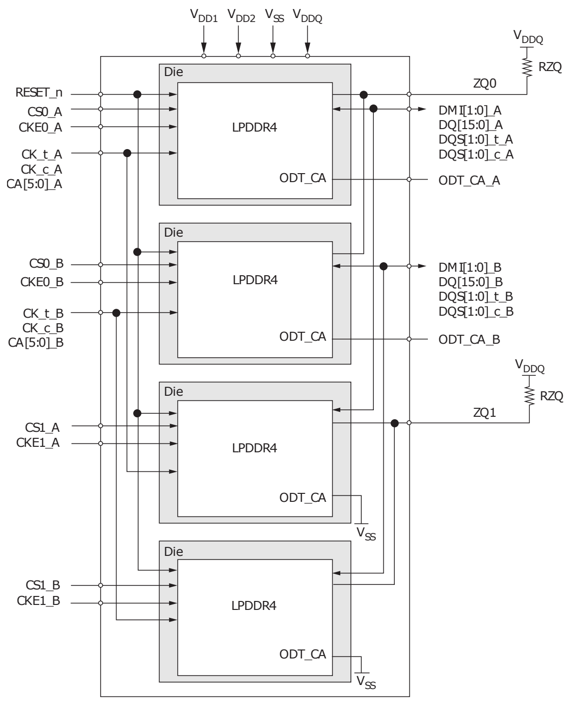


Figure 3.6: Internal architecture of the DRAM. Image taken from component's datasheet.

3.7 Storage

The iMX8 has 2 SD card and one eMMC (embedded MultiMedia Card) interfaces. An eMMC is a cheap NAND flash card, in a package directly soldered to the board. It is widely used as integrated storage into costumer electronic systems.

3.7.1 eMMC: MTFC32GAK

This is a 5.0 eMMC compatible card that can be powered at 3.3 V. It has a capacity of 32 GB. It is packaged in three version of BGA: 100 balls, 153 balls and 169 balls. The eMMC interface consists in a 8 bit bi-directional data lines. In addition there are CLK and CMD pins used for control and commands. Commands are sent serially on CMD from Host to device, as well as the response from device to Host. The data instead can be sent in both direction on DAT0-DAT7. Multiple block can be read or written in a burst, as Fig. 3.8. The interface is very similar to the SD one, but it has 8 bit interface.

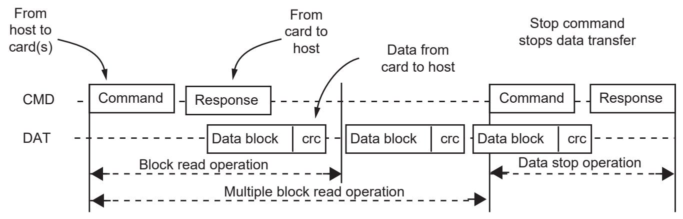


Figure 3.7: Behavior of a eMMC bus. Image taken from JEDEC specifications.[8]

There are two possible speed states: HS200 and HS400. In the first one, one bit per each line is sent at every CLK cycle. The clock has a frequency of 200 MHz, resulting in 200 MB/s. The HS400 singaling instead adopts Double Data Rate, so while keeping 200 MHz clock, it reaches 400 MB/s. In this state, a Strobe is sent from the device to the host, easing the Host sampling of received data.

3.8 NOR Flash: MT35XU512ABA

The iMX8 can also boot from a NOR Flash with an SPI interface, with a capacity of 512 Mb. The chosen device is a Micron product. It can be interfaced both with octal SPI (8 half-duplex lines) and DDR SPI. It must be powered at 1.8 V.

3.9 PCIe Clock

As already seen in Section 2.2.1, an important part of the PCIe is the clock generation and distribution. Every PCIe link needs a reference clock of 100 MHz; in addition, the Root complex (i.e. the iMX8) also needs a reference clock. This clock is switching at a frequency high enough to have the need to avoid stubs and impedance mismatches: thus it is a bad idea connecting several devices to the same clock. Because of this, three clock signals are needed (2 links, 1 root complex).

Many PCIe specific clock generators have multiple outputs in order to satisfy this kind of needs.

Two possible components are considered, and later the second one has been chosen.

3.9.1 PL607041

This clock generator is manufactured by Microchip. It has 4 HCSL output. This acronym stands for High Speed Current Steering Logic, which is a differential signaling where the two drivers are open emitter output. There is the need for a series resistor plus a terminator resistor to ground on both signals in order to work. At every time one of the two pins is sourcing a current of 14 mA which flows through the terminations. The lines need to have an impedance of 50 Ω and a differential impedance of 100 Ω.

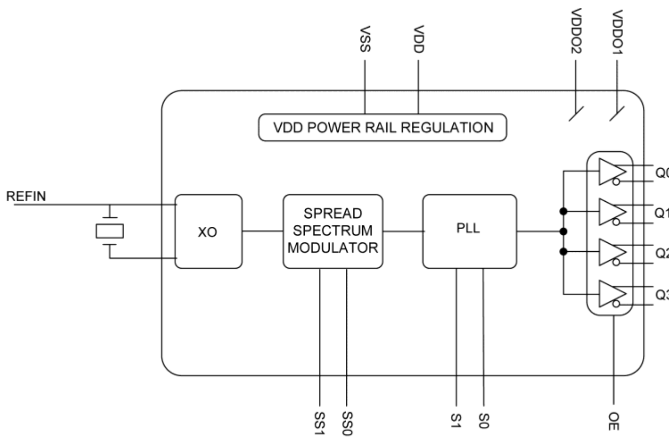


Figure 3.8: Structure of the device. Image taken from component datasheet

The component needs an external 25 MHz crystal as starting frequency that gets multiplied by an internal PLL. Furthermore, it features a Spread Spectrum clock in order to reduce the electromagnetic radiations. There are configuration pins to select the amount of spread introduced: it can be -0.25% or -0.50%. The selection can only be made by placing pull-up or pull-down on the configuration pins.

3.9.2 DSC557-04

This generator is also manufactured by Microchip. The first main difference with the previous component is the absence of an external crystal. In fact this device has internal MEMS oscillator inside the package, reducing the needed area for external components. The output signal levels are HCSL like the other component.

There are two Output Enable signals: `OE1` and `OE2`. The first one enables the first clock output, the second one enables the other two output. This is due to the fact that CLK1 and CLK2 are derived from the same oscillator and they are synchronous: so one pin turns them off at the same time.

This component has been chosen because it has the exact number of needed output, as well as having a reduced total footprint due to the absence of an external crystal. Moreover, having two OE pins, there is the possibility to disable the outputs in a finer way, thus saving more power.

The only drawback of this component is the absence of the Spread Spectrum Clocks. Since SSC is an aid to be compliant during Electro-Magnetic Emission tests, the hosting company decided to try this first version of the board without it.

3.10 Wireless communication: ATWILC3000

Since it does not complicate the project and it does provides useful features, an ATWILC3000 module has been added. This module provides a Bluetooth 4.0 and a WiFi IEEE 802.11 b/g/n interfaces. The antennas are embedded on the module, so no additional components are needed. It can be powered at both 1.8 V and 3.3 V.

The WiFi module can communicate through a SPI or SDIO interface. Instead the Bluetooth submodule interface with a UART with handshake signals (Clear To Send, Request To Send).

3.11 Auxiliary connector

Not all interfaces of the iMX8 can pass on the SMARC-defined connector. In order to avoid waste of resources, an additional connector has been added on the bottom side of the module. It has been chosen in order to have a mating height (the height when female and male are matched) equal to the height of the MXM3 connector where the module is inserted. In this way, if the male connector is present on the carrier at the right height, both connector will match during insertion.

This connector belongs to the FX8C by Hirose. The interfaces passing on this custom connector, i.e. not defined by SGET, are the least important: the module has an already complete set of features when only the standard SMARC connector is used.

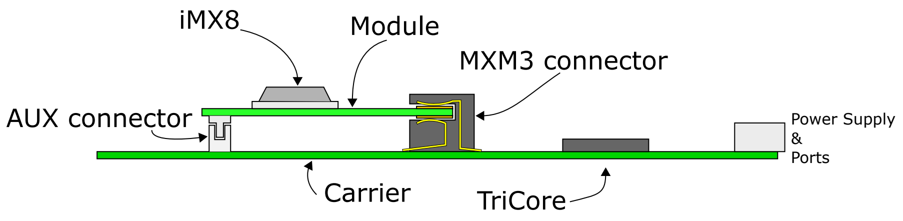


Figure 3.9: Summarizing top level diagram

Chapter 4

Module Architecture

A high-level description of the final structure of the system is here provided. The explanation is aided by blocks diagram illustrating the general concept.

4.1 Top level

As already said, the system will follow the SMARC standard. This means that there are 314 contacts that consist in the main interface of the system. Other secondary signals are connected to the auxiliary connector. The iMX8 needs the PMIC for regulations, two LPDDR4 DRAM, the eMMC and a NOR-Flash. In addition, a uSD socket is placed directly on the module. For communication, there are two Ethernet PHYs, one USB Hub, one PCIe clock generator.

4.2 Power regulation

The only power input to the system is through the 10 pins labeled as `VDD_IN` on the SMARC connector and they provide 5 A at 5 V. The two PMIC regulates all the necessary voltage required by the iMX8; they also regulates two voltages that both power the processor and are used as general purpose power for on-module component: `V_1P8` and `V_3P3` (1.8 V and 3.3 V). Some of the bucks are paired in a multi-phase configuration, allowing more current to be sourced. A summary of all the power supply generated by the PMICs is in Tab. 4.1.

The PMIC1 uses one if its LDOs to regulate `V1P8_SCU`: this voltage powers the System Controller Unit of the iMX8 and its interfaces which comprehends the lowest level of the system: the SCU manages the boot selection, the reset and exchange of info with the PMICs. Since the starting voltage level is 5 V, the total dropout is 3.2 V. Considering that declared maximum current absorbed by iMX8 from `V1P8_SCU` is 390 mA this would result in 1.248 W dissipated by the PMIC,

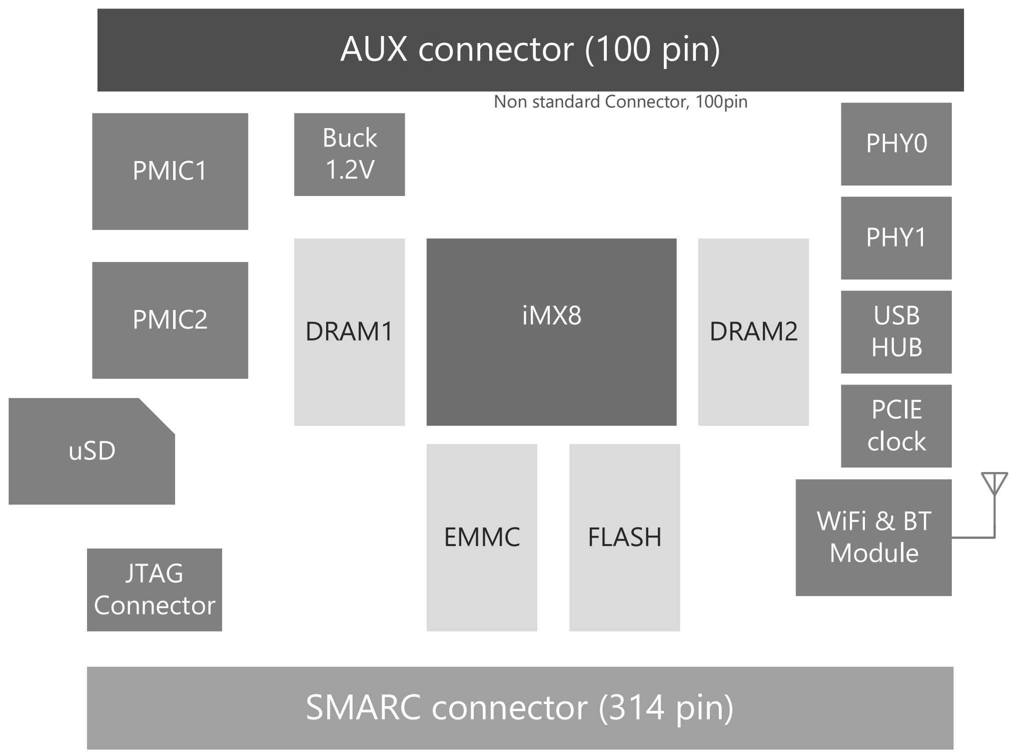


Figure 4.1: Summarizing top level diagram

with a sensible problem of self-heating. In order to attenuate this problem, a ballast diode has been placed in series before the LDO input. The chosen diode is PNS40010ER from NXP: it has a typical forward voltage of 0.89 V when T=25 °C and a forward currunt of 0.5 A. In this way, part of the voltage drops on the diode, splitting the power dissipation between the diode itself and the LDO inside the PMIC. This additional component does not reduce the efficiency, just moves away from the PMIC the power dissipation. The dissipation on the diode will be about $0.89\,\text{V} \cdot 390\,\text{mA} = 340\,\text{mW}$, thus lowering to 0.858 W the dissipation on the PMIC.

Actually `VDD_IN` passes through a high side switch before powering the PMIC. This has been done because the SMARC standard requires that the whole module is not powered up to the de-assertion of `_VIN_PWR_BAD`, which signals that `VDD_IN` in not stable yet. Therefore not even the PMIC will be powered until the voltage is at the right level.

There is one needed voltage that the PMIC cannot regulate: it is `V_1P2` and is needed by the USB Hub and the two Ethernet PHYs. A synchronous buck converter has been added in order to get `V_1P2` (1.2 V). It has been chosen with a high switching frequency, in order to have smaller external components to keep

Table 4.1: Regulator assignement to power supply names.

PMIC number	Regulator	Power Supply name	Voltage range [V]	Output current [A]
1	SW1	V_MAIN	0.95 - 1.10	5
1	SW2			
1	SW3	V_CPU1	0.85 - 1.15	5
1	SW4			
1	SW5	V_CPU0	0.85 - 1.15	2.5
1	SW6	V_DDRIO0	1.06 - 1.17	2.5
1	SW7	V_1P8	1.8	2.5
1	LDO1	V1P8_SCU	1.8	0.4
1	LDO2	V_SD1	1.8 / 3.3	0.4
1	LDO3	-	-	-
1	LDO4	-	-	-
2	SW1	V_GPU0	0.95 1.10	5
2	SW2			
2	SW3	V_GPU1	0.95 - 1.10	5
2	SW4			
2	SW5	V_MEMC	1.05 - 1.15	2.5
2	SW6	V_DDRIO1	1.06 - 1.17	2.5
2	SW7	V_3P	3.3	2.5
2	LDO1	-	-	-
2	LDO2	V_SD2	1.8 / 3.3	0.4
2	LDO3	-	-	-
2	LDO4	-	-	-

a low total footprint. The chosen converter is TPS6206 from Texas Instrument. It is an automotive regulator with a switching frequency of 3 MHz and an output current of 2 A. It is able to move to a power-save state when the load is reduced: it decreases the switching frequency and passes from Pulse Width Modulation to Pulse Frequency Modulation. The purpose is to reduce the quiescent current to keep a high efficiency.

4.2.1 Coin cell or SuperCapacitor

The SMARC defines a pin dedicated to be connected on the Carrier to a Lithium Cell or a super capacitor, with the function of a backup power for keeping the time when the system is powered off. The PMIC has a particular regulator connected to the SNVS region of the iMX8, which is the part that manages the RTC; moreover the PMIC itself can also charge up the coin cell or the SuperCapacitor from the same input pin when the main power is on. Since they have compatible features, they have been connected together. In this way, the coin cell on the carrier powers the SNVS whenever the main power is off; instead when `VDD_IN` is present, current flows from the PMIC toward the cell in order to charge it.

4.3 Boot sequence

When power is provided to the module, several events take place.

- First of all, `VDD_IN` powers the PMIC.
- The PMIC is connected in such a way that it is not enabled as soon as the input voltage is present. Instead it waits for its pin `PWRON` pin to read a high voltage. So all its output are disabled, except for `VSNVS` that is always enabled.
- The pin `VSNVS` is connected to `VDD_SNVS` of the iMX8 which powers a small part of the iMX8. This block contains `ON_OFF_BUTTON` and `PMIC_ON_REQ`. The first one is connected through a dedicated SMARC pin to a button on the carrier.
- When the button on the carrier is pressed, the signal arrives to the iMX8. Then, it asserts `PMIC_ON_REQ` so requesting the PMIC to turn on.
- At that point, the PMIC finally turns on all its regulators thus powering the iMX8.

4.4 Boot modes

Upon the release of `_POR` the System Controller Unit of the iMX8 reads a set of pins in order to decided from which peripherals it should load the program image.

The SMARC specifications defines 3 pins of the connector for boot selections, performed through switches on the carrier. For each of the 8 combination there is a boot source, that can be PCIe, USB, SD etc. Some of this can be left not implemented. The iMX8 instead has 6 bits for selecting the boot source with totally different value. This means that a simple logic net is needed to translate SMARC values to the one understandable by the iMX8.

In addition to that, a `FORCE_RECOV#` pin is routed through the SMARC connector. The standard says that it is driven low by the carrier in order to overwrite part of the boot code, reading from a USB device connected to USB0. This requirement can be satisfied by the USB Serial Download of the iMX8. It allow to

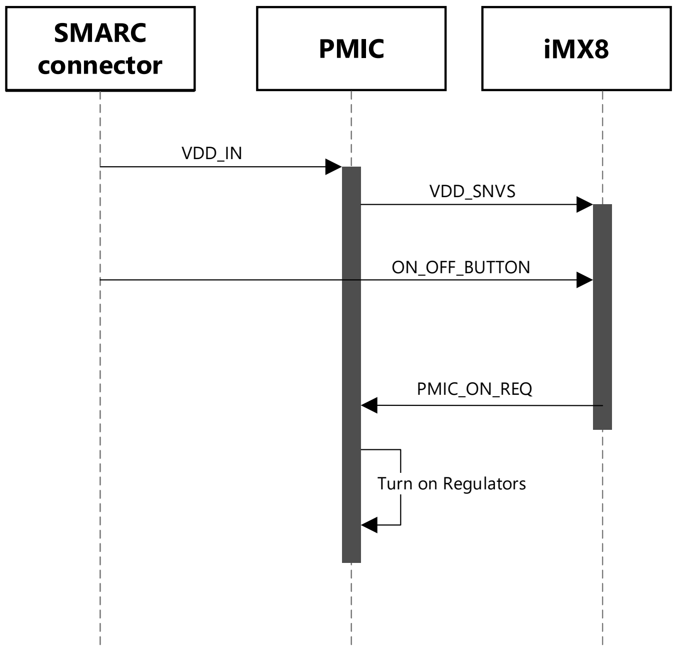


Figure 4.2: Sequence diagram of the boot process. After the turning the regulators on, the iMX8 will be fully powered.

boot from any USB interface and to download a program image to memory. This function is called in a particular configuration of the boot pins. Also, if the pin selects the fuses load option and they have not been programmed yet, the Serial Download is automatically called.

In Tab. 4.2 is shown the wanted output related to the input possible configurations and the resulting boot mode. Notice that the modes left not implemented has been considered as "don't care" configurations. Obviously the only concern was to use the minimum number of logic gate in order to save space on the PCB. Here is the found solution(where x is the input i.e. `_BOOT_SEL` and y is the output i.e. `SCU_BOOT_MODE` pins):

$$\begin{aligned} y_0 &= 0 \\ y_1 &= x_2 \cdot \overline{x_1} \\ y_2 &= y_3 \cdot \overline{x_1} \\ y_3 &= \overline{y_5} \\ y_4 &= x_1 \cdot x_0 \\ y_5 &= \overline{x_2 + x_0} \end{aligned}$$

Notice that this equation does not consider `FORCE_RECOV#`: this feature has been added starting from this logical network, that satisfies the requirement when

Table 4.2: Ideal output of the logical net

Input				Output						
FORCE_RECOV#	BOOT_SEL			BOOT_MODE						Mode
	2	1	0	5	4	3	2	1	0	
0	0	0	0	0	0	0	1	0	0	USB Recovery
0	0	0	1	0	0	0	1	0	0	USB Recovery
0	0	1	0	0	0	0	1	0	0	USB Recovery
0	0	1	1	0	0	0	1	0	0	USB Recovery
0	1	0	0	0	0	0	1	0	0	USB Recovery
0	1	0	1	0	0	0	1	0	0	USB Recovery
0	1	1	0	0	0	0	1	0	0	USB Recovery
0	1	1	1	0	0	0	1	0	0	USB Recovery
1	0	0	0	1	0	0	0	0	0	Carrier SATA
1	0	0	1	0	0	1	1	0	0	Carrier SD
1	0	1	0	X	X	X	X	X	X	Carrier ESPI (CS0#)
1	0	1	1	X	X	X	X	X	X	Carrier SPI (CS0#)
1	1	0	0	0	0	1	1	1	0	Module SD
1	1	0	1	X	X	X	X	X	X	Remote
1	1	1	0	0	0	1	0	0	0	Module eMMC
1	1	1	1	0	1	1	0	0	0	Module NOR SPI

`FORCE_RECOV#` is not asserted.

In order to add this feature, it is necessary to notice that when `FORCE_RECOV#` = 0 than `SCU_BOOT_MODE` = 000100: in other words all bits but `SCU_BOOT_MODE2` are driven to 0. Remember that the USB Serial Download is enabled when `SCU_BOOT_MODE` = 000100 but also when `SCU_BOOT_MODE` = 000000: this configuration load the fuses, but if they have not been programmed yet, the Serial download is executed instead. So a possible solution that minimizes the number of logic gates is to use `FORCE_RECOV#` to pull down all signals that must be driven low while leaving `SCU_BOOT_MODE2` as is. This solution is shown in Fig. 4.4. Diodes have been inserted in order to prevent current from flowing back to the carrier.

The actual and final output of the net is shown in Tab. 4.3. Rows in red are the one differing from the ideal output. Remember that the case where

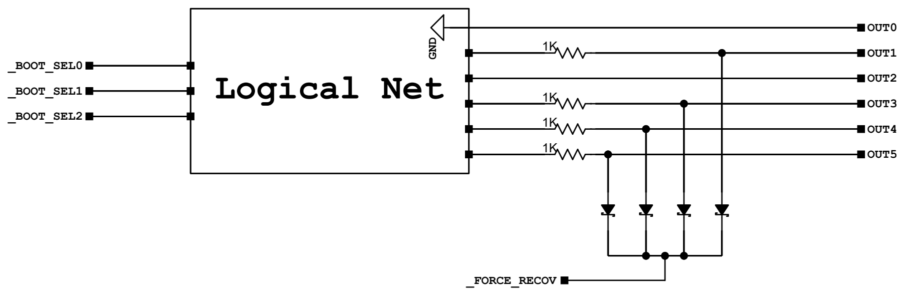


Figure 4.3: `FORCE_RECOV#` implementation. It shows how it affects the logical net output for the boot selection.

`FORCE_RECOV#` = 0 and the Fuse are loaded do not present a problem: if the they have not been blown yet, the Serial Download will be executed. In case the Fuse are already programmed, the switch on the carrier can be set in order to have the correct configuration.

4.5 Power sequencing

As already said, the two used PMIC come already pre-programmed in order to be compatible with the i.MX8, so all its power sequencing and voltage levels are already satisfied. But there are other two components that need more than one power supply and sequencing has to be considered: the Ethernet PHYs and the USB Hub. The first ones need `V_1P8` and `V_1P2` (regulated by the additional buck) while the USB Hub is powered by `V_3P3` and `V_1P2`. From the datasheets it possible to read that the PHYs need `V_1P8` before `V_1P2`, while the USB need `V_1P2` and then `V_3P3`. Since `V_1P8` and `V_3P3` are output at the same time by the two PMICs, these two requirements are in contrast. A workaround has been adopted:

- the buck that regulates `V_1P2` is enabled only when `V_1P8` is ready, so satisfying the PHYs needs.
- `V_1P2` enables a high side switch (made with a PMOS) that connect `V_3P3` to the USB Hub. Thus, it receives `V_1P2` before `V_3P3`.

4.6 USB

A USB Hub has been added in order to increase the number of ports. The iMX8 has three USB cores:

Table 4.3: Actual output of the logical net. Entries between brackets, like "(1)" , differ from the ideal implementation. Fuse eventually load USB Recovery if not programmed.

Input				Output						
FORCE_RECOV#	BOOT_SEL			BOOT_MODE						Mode
	2	1	0	5	4	3	2	1	0	
0	0	0	0	0	0	0	(0)	0	0	(Fuse)
0	0	0	1	0	0	0	1	0	0	USB Recovery
0	0	1	0	0	0	0	(0)	0	0	(Fuse)
0	0	1	1	0	0	0	(0)	0	0	(Fuse)
0	1	0	0	0	0	0	1	0	0	USB Recovery
0	1	0	1	0	0	0	1	0	0	USB Recovery
0	1	1	0	0	0	0	(0)	0	0	(Fuse)
0	1	1	1	0	0	0	(0)	0	0	(Fuse)
1	0	0	0	1	0	0	0	0	0	Carrier SATA
1	0	0	1	0	0	1	1	0	0	Carrier SD
1	0	1	0	(1)	(0)	(0)	(0)	(0)	(0)	(Carrier SATA)
1	0	1	1	(0)	(1)	(1)	(0)	(0)	(0)	(Module NOR SPI)
1	1	0	0	0	0	1	1	1	0	Module SD
1	1	0	1	(0)	(0)	(1)	(1)	(1)	(0)	(Module SD)
1	1	1	0	0	0	1	0	0	0	Module eMMC
1	1	1	1	0	1	1	0	0	0	Module NOR SPI

- The first one is a USB 2.0 compliant controller, but it only interfaces with a physical layer called HSIC (High Speed Inter Chip), that is basically a low voltage differential pair intended for on-board communication between integrated circuits. In fact, the total length must be less than 10 cm. This has not been used in the project since no useful purpose has been found for it.
- The second is full USB 2.0 core with OTG functionalities.
- The last one is USB 3.0 compatible, so it comprehends both `Tx` and `Rx` pairs, as well as the USB High Speed pair of signals.

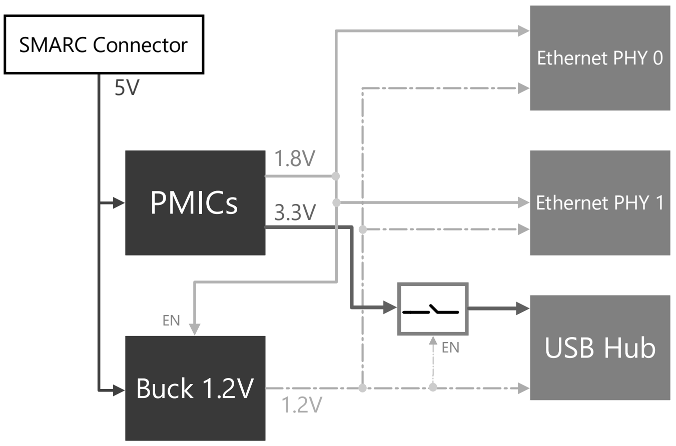


Figure 4.4: Block diagram showing the power sequencing of the 1.2V power supply. The Buck is enabled by 1.8V and 1.2V lets the 3.3V supply to reach the USB Hub.

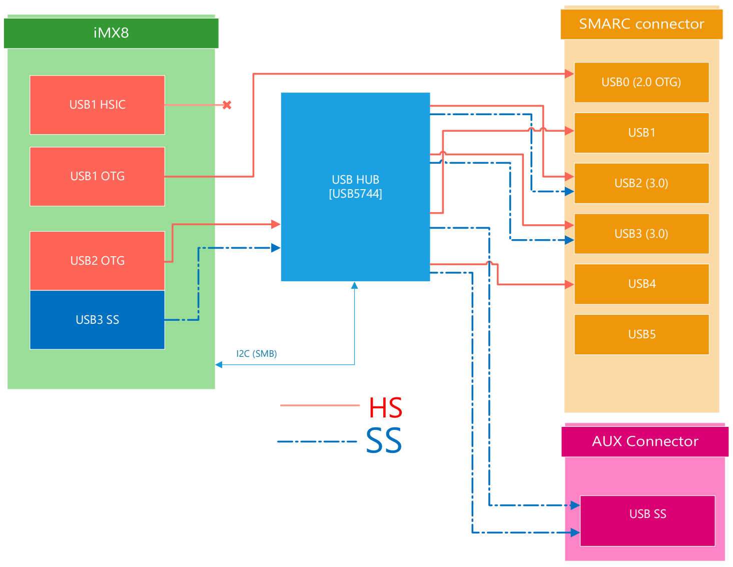


Figure 4.5: USB connection of the iMX8, Hub and connectors.

The chosen USB Hub supports USB 3.0, thus it has been connected with the USB 3.0 core of the iMX8. The OTG capable port of the processor has been directly connected to the SMARC connector. Since all the downstream port of the Hub are USB 3.0 but the SMARC ports are not, the exceeding `Tx` and `Rx` pair has been routed to the Auxiliary connector: in this way, the carrier may re-join the 2.0 and the 3.0 signals of the same port, thus having a complete 3.0 port. If only the 2.0 signals are used, the related 3.0 pairs must be left unconnected.

The Hub presents a SMB interface for configuration register access from the processor. These pins have been connected to a I2C of the iMX8 since the controller also supports SMB.

For each USB port present on the SMARC connector there is a `EN_OC` signal that has a double role: it is pulled up by the module in order to enable a switch connecting +5 V to the device; and it is pulled low by the switch to signal an over current condition. This feature is compatible with the pin of the Hub called `PRT_CTRL` that has the exact same behavior; so they can be directly connected. The same thing can not be done for the USB connected directly to the SMARC connector : in fact the processor presents one pin for enabling the power delivery and one pin for detecting the over current condition. In order to get the right behaviour, the pins has been connected together to form a single line, with the addition of a diode preventing sourcing current from the `EN` pin.

Table 4.4: USB Assignement

iMX8 peripheral	HUB port	SMARC port	Aux Connector
USB_OTG1	-	USB0	-
USB_OTG2 + USB_SS3	HS1	USB2	-
	SS1		-
	HS2	USB3	-
	SS2		-
	HS3	USB1	-
	SS3	-	HUB3
	HS4	USB4	-
	SS4	-	HUB4

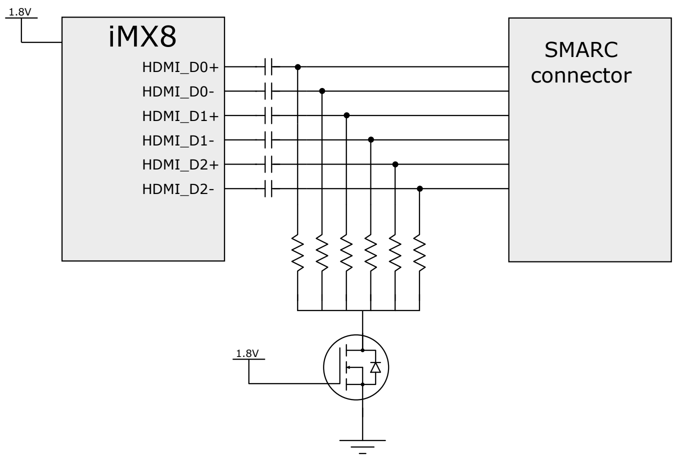
4.7 HDMI and DisplayPort

The iMX8 has multi-protocol core that can manage several protocols, one at a time. It can manage:

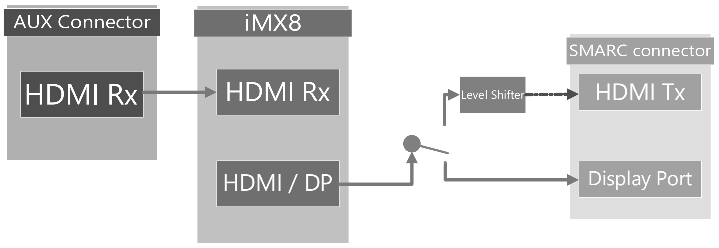
- HDMI, both version 1.4 and 2.

- DispalyPort (DP) and embedded Display Port (eDP).
- High-bandwidth Digital Content Protection system (HDCP) on both DP and HDMI.

Despite the HDMI support, the output signals are low voltage differential signals compatible with DP specification and while there is a TMDS encoder, there is no driver. Because of this, if the target is HDMI, the signal from the iMX8 must be translated to the correct levels. There are commercial chips able to translate the voltage levels, such as the SN75DP139 from Texas Instrument. NXP instead suggest to use a pull-down resistor as level shifter, to be placed on both positive and negative signal of each pair in order to scale down the voltage level. This is shown in Fig. 4.6a. The N-MOSFET disconnects the resistors when the iMX8 HDMI core is not powered.



(a) HDMI level shifter made with discrete components.



(b) HDMI/DP selection and redirection to the appropriate SMARC pins. It also shown the input HDMI to the iMX8 from the Auxiliary connector

Figure 4.6: These two diagrams shows the HDMI level shifter and the HDMI/DP selection

It has been chosen to allow the selection of HDMI over DP or viceversa by soldering or desoldering some series resistor. In this way it is possible to connect the iMX8 pins (shared between DP and HDMI) to the correct SMARC pins. Both lines require 50 Ω of characteristic impedance and 100 ohm differential. The series resistor must be placed in such a way to minimize impedance mismatches.

The DisplayPort standard defines DC blocking series capacitor on the data lanes. Those have not been placed on the module since the SMARC standard states that they must be placed on the carrier close to the connector.

4.7.1 Voltage levels

Since the SMARC suppose a level shifter, the DDC and HPD signals are defined to be 1.8 V. Instead the iMX8 pins are directly compatible with the HDMI and Display port standard. Because of this, they should be connected to 5 V signals. Directly connecting the 1.8 V signals coming from the SMARC connector may result in the processor not recognising the high logical level.

The HPD can be easly translated using a buffer, being an input signal. The used buffer is the 74LV1T34GW, that can up-translate from 1.8 V to 3.3 V, that is the standard voltage defined by DisplayPort for this signal and it is also high eneough for the HDMI.

Concerning the DDC channel, that is actually an I2C, it is necessary to keep in mind that this is a bidirectional channel. A known solution is often used to have an I2C with two different voltages: two mosfet can be placed like in Fig. 4.7. If the lines are driven low on the 1.8 V side, the V_{gs} of that mosfet become greater than the threshold voltage, conducting from drain to source and thus discharging the 3.3 V side. If the 3.3 V is driven low, the body diode start conducting and discharge the the 1.8 V side. Otherwise, the body diode is not conducting and $V_{gs} = 0$.

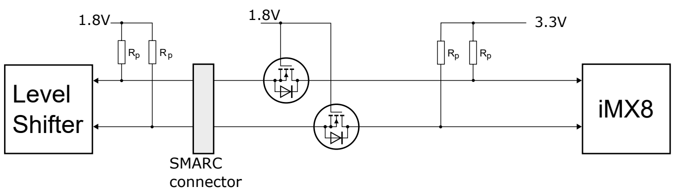


Figure 4.7: Level shifter implementation from 1.8V to 3.3V of the DDC channel.

4.7.2 HDMI Sink

The iMX8 also has an HDMI receiver, also called Sink (the same behavior of a monitor). Such interface is not present on the SMARC connector, so it has to be connected to the Auxiliary connector. It receives the +5 V passed on the cable, and use it to assert `HPD` in order to signal the connection to the Source. It is also used for the pullup on the `SCL` signal of the DDC channel for Sink, as defined by the HDMI standard.

4.8 PCIe

The iMX8 has two PCIe controller and 3 PHYs: the last one can be used by the SATA controller. The SMARC defines 4 lanes that can be used independently or can be grouped in 2x or 4x links. Only two lanes has been filled in this project.

It has been chosen to use a common clock configuration, where several copies of the same clock are sent both to the Root complex and to the devices. Two more clocks are needed for the devices. The DSC557-04 has been used, connected as shown in Fig. 4.8. The two PCIe controller and the SATA controller all use the same reference clock at 100 MHz. Being the clock a HCSL signal, series resistor and pull-downs are needed on the output of the component.

In addition to the data signals, some more control signal are present: `PERST#` and `CLKREQ#`. The first one works like a reset for the device and is generated by the Root complex. The second one is an open drain signal that may be asserted either by the Root complex or by the device: it is used to enable the clock generator. While the first one passes on the SMARC connector, `CLKREQ#` doesn't. Instead the `WAKE#` can be passed to the module and it has a similar effect of `CLKREQ#`: it turns on both the clock generation and power regulation.

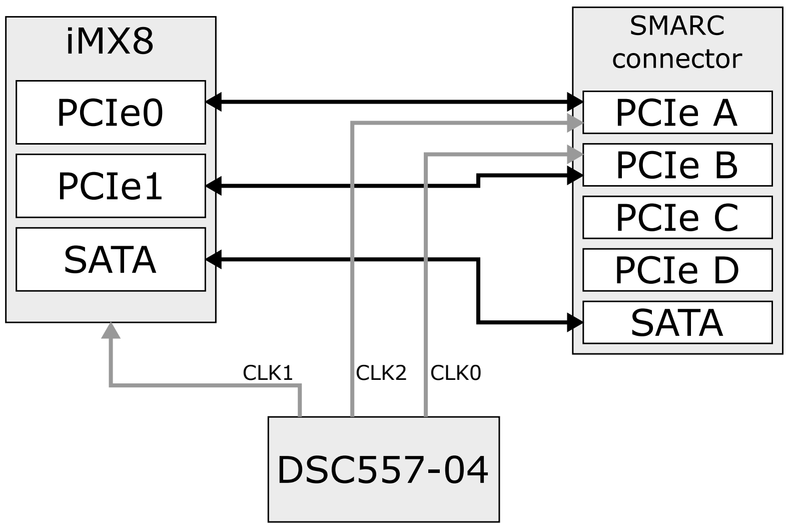


Figure 4.8: PCIe connection and clock routing. The fourth clock generator output is not used.

The signals `CLKREQ` has been connected only between the iMX8 and the clock generator. Remember that `OE1` turns on only `CLK0`, while `OE2` turns on both `CLK1` and `CLK2`. The connection has been performed such that when only PCIeA of the SMARC is used, the clock is available only to the processor and the PCIeA port; if PCIeB is used, the clock is fed to the processor, PCIeA and PCIeB. This has been done with the connection shown in Fig. 4.9.

When `_PCIE0_CLKREQ` is asserted, the lower MOSFET is turned off and and the resistor pulls up `OE2`: the result of this is enabling `CLK1` and `CLK2`, connected respectively to the iMX8 and PCIeA port. Meanwhile `_PCIE1_CLKREQ` has a high level, thus the diode is not conducting and the upper MOSFET is driving `OE1` to 0V. If instead `_PCIE1_CLKREQ` is asserted, the diode start conducting thus driving low also `_PCIE0_CLKREQ`. As a consequence, both `OE1` and `OE2` are enabled.

Actually the pull up resistor are not mounted because the DSC557 already has internal pull-up resistors on `OEx`, so if they are left floating the input as read as high level.

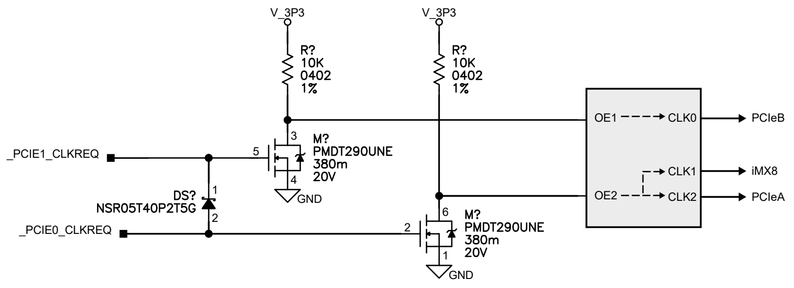


Figure 4.9: PCIe connection and clock routing. The fourth clock generator output is not used.

4.9 Ethernet

The iMX8 features two Ethernet MACs and two are the SMARC defined Ethernet ports. The SMARC requires that the signals going to the carrier have to be Media Dependent Interface, so two PHYs has to be placed on the module and the chosen components is the KSZ9031RNX.

For each PHY there is one RGMII bus and its MDIO/MDC interface. These second one is a two wire management interface needed for the configuration of the PHY itself. While it can be routed as a I2C bus, so in a multi-slave topology, the iMX8 has one MDIO bus for each MAC. Thus each PHY has its independent interface, resulting in a point-to-point connection.

Each PHY generates four differential pairs that make up the Media Dependent Interface. The SMARC says that the needed magnetics and transformers must be placed on the carrier module, close to the connector RJ45.

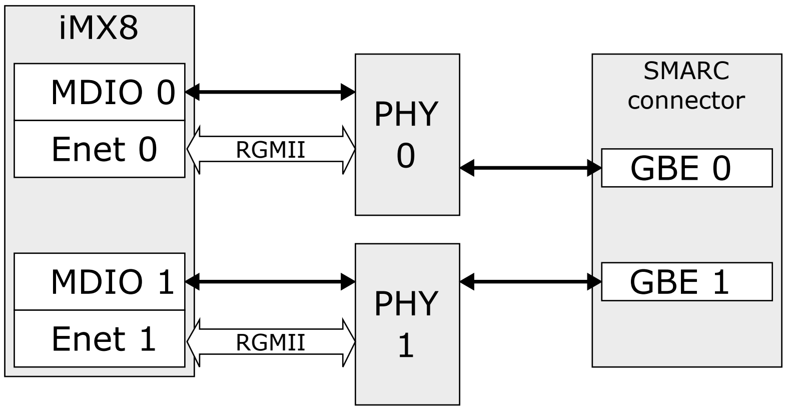


Figure 4.10: Block diagram of the Ethernet connection. It shows how two PHYs are present, each one with a RGMII and MDIO interface.

The PHYs has several strapping pin used to enable or disable some if its features (like the different data rates). It also has two pins for turning on and off LEDs signaling the activity and the speed of the link. Such pins can be connected to LEDs placed on the carrier board, passing through the SMARC connector. They are configured to be connected to the cathode of a diode and absorb current.

The PHY needs several power supplies: 1.2 V, 1.8 V and 3.3 V. The voltage level for the control pins is 1.8 V, to make it compatible with the iMX8. The LED pins are required by the SMARC to be 3.3 V tolerant. Because of this, a buffer is inserted in order to make it tolerant to the right voltage level.

4.10 RAM

The iMX8 has two memory controller for DDR4 or LPDDR4 DRAM. An automotive grade LPDRR4 from Micron has been chosen. Two of this kind will be present. As already seen, each chip has 32 single ended data signals, and inside the package there are four equal dies.

The RAMs are powered by both `V_1P8` and `V_DDRIOx` where `x` can be 0 or 1. The second supply,`V_DDRIOx`, also power part of the iMX8: so the two devices has the input/output interface driver both powered by the same voltage. The iMX8 has a set of pins that have a double role: they behave as control signal for DDR4 or for LPDDR4, which differs in the number of command signals.

The layout of a DDR interface can be quite complex, due to a relatively high number of pins that must have matched length and have to sustain high frequencies (1600 MHz). In order to simplify the design, it is possible to swap bits within a byte, as well as swapping the bytes themselves: in this way it should be possible to avoid vias and change of plane. Because of this, the connections of data signals between iMX8 and RAM might change during the layout/routing phase.

4.11 uSD socket

An automotive uSD socket has been added to the module. It has double contacts for each pad, resulting a more reliable and vibration-tolerant contact. Moreover, it has an internal switch that get closed when the card is inserted: if properly connected it drives low `_CD`, signaling the iMX8 that a card has been detected.

The uSD card is always powered by `V_3P3`, even if the signaling passes to 1.8V. A reset circuit allows to manage the power to the card: when `_RST` is asserted, the power going to the card is disconnected and the card itself is also discharged through a MOSFET. When instead the card has to be operational, a P-MOS deliver the power to the card. Since the `_RST` signal has the wrong polarity for this behavior, an inverter has been used. Moreover, a resistor is needed to limit the current while discharging: in fact the capacitance of the uSD card can be sensible(max 5 µF) and it adds to bypass capacitors in parallel to it.

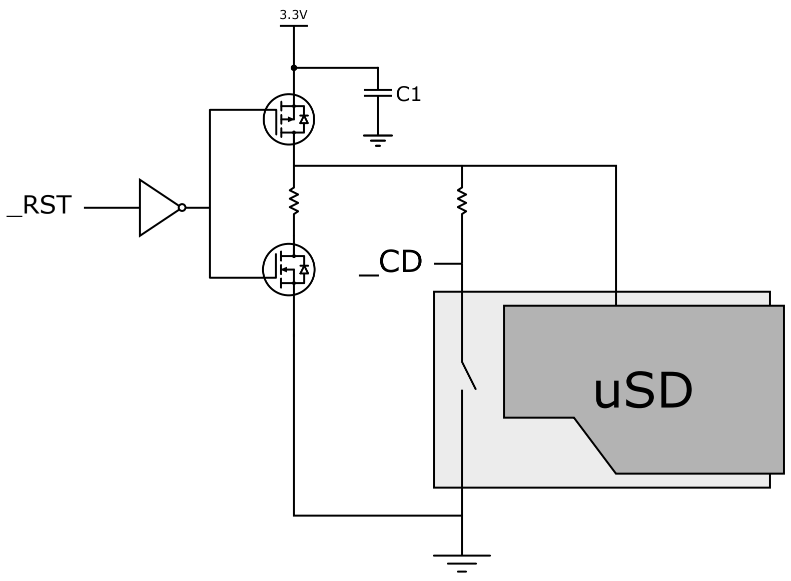


Figure 4.11: Diagram of the uSD socket. It is shown the internal switch for card detection and its relative pull-up, together with the P-MOS and N-MOS for charging and discharging the uSD card.

The capacitor C1 has been connected to the 3.3 V supply. This is has been done in order to have a reservoir of charge close to the MOSFET. In fact, when the p-MOS starts conducting, a peak of in-rush current can be present, due to the large load capacitance of about 10 µF. In this way, the charge needed is taken from C1 that is always connected to the power supply. This should prevent high peaks of absorption from the 3.3 V that may result in a temporary voltage drop.

4.12 Reset Behaviour

The reset structure has to be considered carefully. In fact there are several reset sources:

- The PMIC asserts the reset signals until all voltage levels are at their correct value.
- The iMX8 has to be able to reset itself together with all the other components.
- The SMARC defines a `RST_IN` driven by the carrier that should reset the module
- A JTAG connector is present on the module for software development and debugging: this interface comprehends a reset signal.

A general reset signal is present and it is called `_RST`. It can be driven low by the PMIC, trough its pin `RESETBMCU`. It asserts this signal during power-on, as well as when a watchdog event is triggered. This happen if the iMX8 fails to refresh a countdown timer within a certain time. The processor can also trigger the watchdog event asserting `SCU_WDOG_OUT`. In this way the iMX8 is able to reset itself in case of failure.

The SMARC signal `_RST_IN` and the `TRST` from JTAG connector as connected with a diode to `_RST`. In this way, when one of them is driven low, the corresponding diode starts conducting thus discharging the line and asserting `_RST`. The effect of this is the resetting of the Ethernet PHYs, the NOR Flash and the iMX8. The PHYs can also be reset independently using a GPIO of the processor.

The USB Hub instead is only reset by a GPIO, since it has its own internal timer for power-on reset. On the other hand, the NOR Flash is reset only by the `_POR` since the processor can reset it through SPI commands.

4.13 JTAG

The JTAG interface (defined by the Joint Test Action Group) is an interface for testing and debugging a whole system after manufacturing. Born for boundary-scan of the ICs in order to test connections to the PCB and package, has become the interface to use during software development. In fact, a simple 4 pin interface

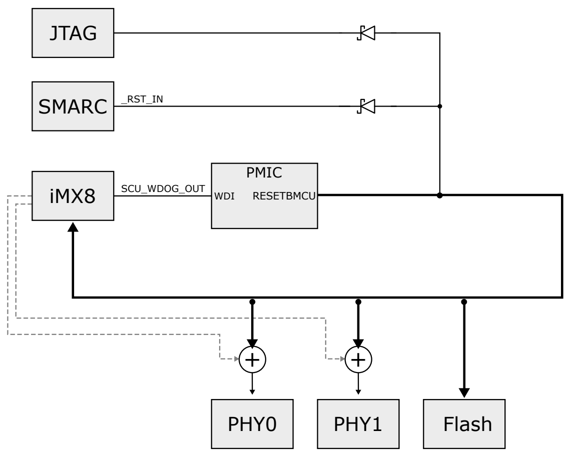


Figure 4.12: Reset circuit.

gives access to all internal registers and also to boundary scan registers. These are register that over-rides the normal funcionality of an IC pin in order to test PCB level connections: they can read or write a state to those pins. The 4-pin bus can be connected in a daisy chain configuration with all JTAG capable devices. In our case, only the iMX8 has a JTAG interface.

This protocol allows the software designer to debug and trace execution of the firmware, as well as programming the Flash memory with the program image.

On the module a 10 pin header has been placed and connected to the processor only. The reset signal, as already said, has been added to all the other reset sources.

4.14 MIPI CSI

As already seen, the CSI interface is dedicated to raw image from a camera. The data rate is scalable with the number of lanes. The SMARC defines two CSI interfaces: the first one `CSI0` has 2 lanes, while `CSI1` has four. It also defines a priority: `CSI1` cannot be connected while leaving `CSI0` unconnected.

The iMX8 has two interfaces with four lanes each. The chosen solution is to connect 2 lanes of a bus to `CSI0` and all 4 lanes of the other to `CSI1`. This solution leaves 2 unconnected lanes of the iMX8. It has been chosen then to route this 2 lanes to the auxiliary connector, and if needed, re-join the two lanes on the carrier, in order to get 4x interface. This of course complicates the layout design: data lanes must be matched with the clock signal within 0.15 ns.

This has been done because cameras interfaces are a key feature for the Autodrive project. In a project where such a bandwith is not needed, it is still possible to easily use a 2x and 4x interfaces.

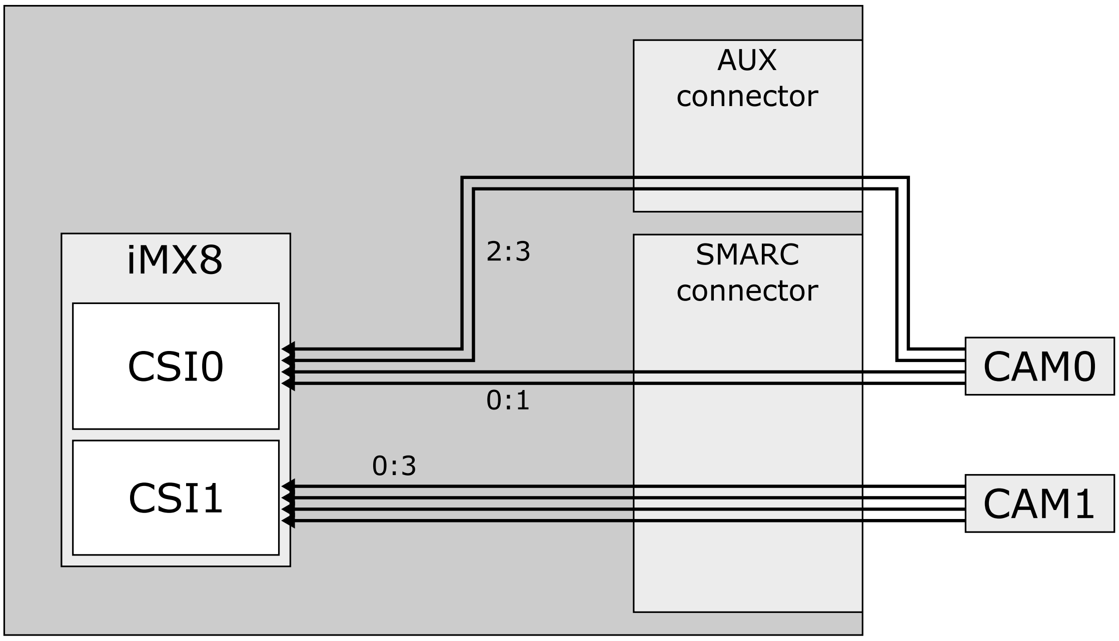


Figure 4.13: CSI connection diagram. `CSI0` is split between the SMARC connector and auxiliary one.

4.15 Wifi-Bluetooth Module

The ATWILC3000 has a pin called `VBAT` that must be tied to 3.3 V and another supply pin called `VDDIO`. This second one can be tied to 1.8 V,2.5 V or 3.3 V. The first one has been chosen because it allows communication with the microcontroller.

The component has a UART interface dedicated to the Bluetooth and an SDIO/SPI interface for WiFi. The Bluetooth sub-module present a four-signal UART, complete of handshake signals. Since the iMX8 has only two available UART port with `_CTS` and `RTS` and they are already connected to the SMARC connector, a simple two wire interface has been used. The module datasheet states that the handshake signals are optional.

Also due to availability problems, SPI has been chosen over SDIO for WiFi communication. In fact, the three available SD controller of the iMX8 are already used.

The ATWILC3000 also needed an RTC clock at 32.768 kHz, that is provided by a pin of the iMX8.

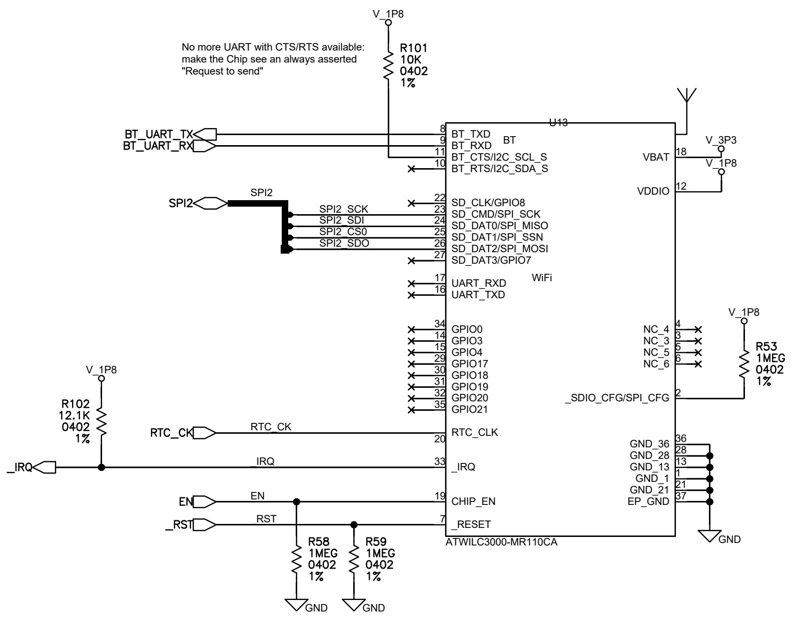


Figure 4.14: Schematic of the ATWILC3000 connection.

4.16 LVDS

The SMARC has two LVDS (Low Voltage Differential Signaling) channel for driving LCD displays, while the iMX8 potentially has 2 channel of eight bits each. Since both the SMARC channels are muxed with a DSI interface, it has been chosen to implement one DSI and one LVDS channel. This interface does not require external components. Some additional GPIOs are present for enabling the supply to the display and its back-light. Moreover, the back-light intensity can be controlled through a PWM signal generated by the iMX8. One I2C bus is shared between the LVDS and DSI for configuration of the displays.

The remaining LVDS channels of the iMX8 are connected to the Auxiliary connector, in case they might be needed in a different project.

Chapter 5

Conclusions

Starting from the non-strict requirements, a general purpose board has been developed. The standard SMARC has been chosen to get interoperability with other modules. The design started from the understanding of the i.MX8 capabilities and its interfaces. A study of all standard interfaces was needed. Then component has been looked for and chosen to match both the processor and the SMARC standard. The project has been developed up to the schematic level.

5.1 Future implementations

The project will pass to the layout and routing phase. Particular attention will be needed for the reduced board size and the high-speed interfaces. Having to deal with frequencies above 1 GHz, connections has to be considered as transmission lines. Moreover, a simulator is needed for trouble shooting the connections during design phase.

The first step is the definition of a stack-up, that is the definition of all layers composing the PCB. It must take in consideration the needed impedances: the dielectric thickness heavily influence the trace width given a target impedance. Moreover, ground and power planes will be defined during this phase. Since all high-speed traces need a return path, all traces are routed close to a ground plane, resulting in a microstrip or stripline structures. Because of this, several ground planes will be inserted, interleaved with signal ones. The SMARC requires that the total PCB height must be 1.2 mm so to be inserted in the MXM3 connector.

Moreover, the fine pitch BGA of the iMX8 impose a minimum number of layer for the break-out of the signals. As a rule of thumb, the number of needed layer can be estimated considering the number of signals to be routed out of the BGA and the number of available "channels", which is the space between balls where a trace can be routed. The formula is $layers = \frac{signals}{channels}$. Considering the iMX8 package, the number of channels is 26 per side, so $26 \cdot 4 = 104$ and the number of signal is $total - powerPin = 1313 - 719 = 594$. So, substituting in the first formula:

$$layers = \frac{signals}{channels} = \frac{594}{104} \simeq 5.71 \simeq 6$$

Therefore, a stackup where 6 signal layers are present may be a good starting point.

Moreover, the required impedances has to be considered: the main part of differential pairs need an impedance of $50\,\Omega$ single ended and $100\,\Omega$ differential, but also $85\,\Omega$ and $90\,\Omega$ are needed. Another aspect to consider of the stackup is the inter-plane capacitance that results from two parallel planes separted by a dielectric. This phenomenon can be exploited to get a low inductance capacitance between a ground and a power planes adjacent to each other. This capacitance adds to the bypass capacitors, thus reducing the noise on the power supply.

The routing phase has to be done respecting some indications. The return path of signals is an important concept to consider: it is the path that takes the current flowing in the opposite direction of the signal propagation. The path taken influence the area of the loop of current: with a solid reference plane, the current would take the least impedance path. But if some split in the plane are present, the current may be forced to take different path, increasing the loop area and than the Electromagnetic Emissions. [5] Moreover, the trace-width has to be kept constant in order to minimize impedance mismatches that would results in reflection and performance degradation.

In order to keep the common mode rejection benefits of a differential signaling, the two signals has to have an exact phase difference of 180°, and this may not be true in case of length mismatches, trace width differences or local dielectric differences. An example of this last cause is the Fiber Weave Effect, consisting in dielectric constant differences based on the presence or absence of glass fiber in the layer between the signal trace and the reference plane. If the two traces have different impedances, the phase velocity will not be the same. This effect can be avoided by using a a material with tighter glass fiber weave.

All this considerations have to be kept in mind while trying to satisfy all the timing specifications of the standard interfaces.

Appendix A

Schematic

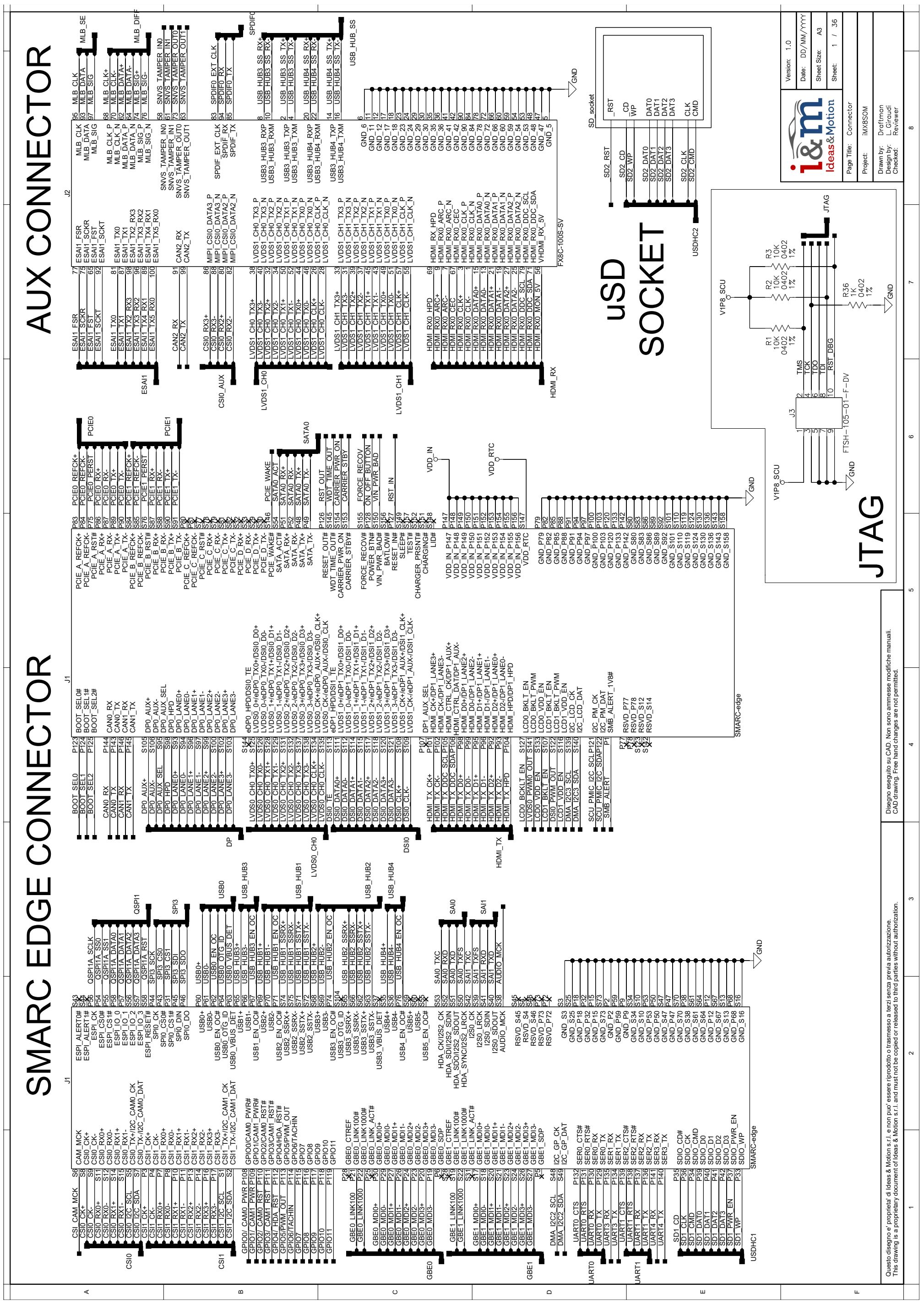


Figure A.1: Connectors

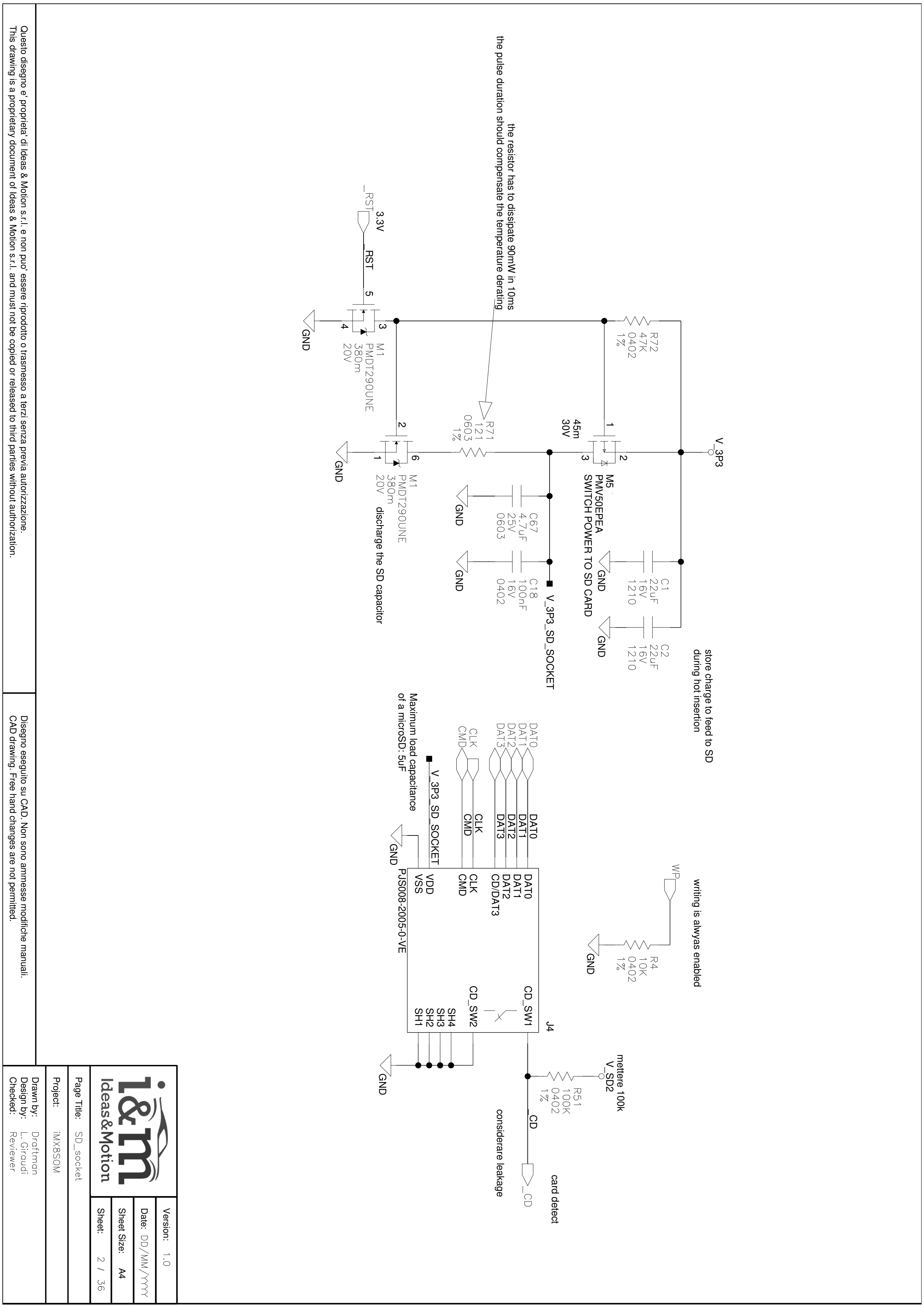

the resistor has to dissipate 90mW in 10ms
the pulse duration should compensate the temperature derating
3.3V
_RST
_RST
GND
M1 PMDT290UNE 380m 20V
R72 47K 0402 1%
R71 121 0603 1%
M5 PMV50EPEA
45m 30V
SWITCH POWER TO SD CARD
V_3P3
M1 PMDT290UNE 380m 20V
discharge the SD capacitor
C67 4.7uF 25V 0603
C18 100nF 16V 0402
V_3P3_SD_SOCKET
C1 22uF 16V 1210
C2 22uF 16V 1210
store charge to feed to SD during hot insertion
DAT0
DAT1
DAT2
DAT3
CLK
CMD
Maximum load capacitance of a microSD: 5uF
V_3P3_SD_SOCKET
DAT0
DAT1
DAT2
CD/DAT3
CLK
CMD
VDD
VSS
PJS008-2005-0-VE
J4
CD_SW1
CD_SW2
SH4
SH3
SH2
SH1
WP
writing is alwyas enabled
R4 10K 0402 1%
mettere 100k
V_SD2
R51 100K 0402 1%
_CD
considerare leakage
_CD
card detect
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
I&M Ideas&Motion
Page Title: SD_socket
Project: iMX8SOM
Drawn by: Draftman
Design by: L.Giraudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A4
Sheet: 2 / 36

Figure A.2: uSD Socket

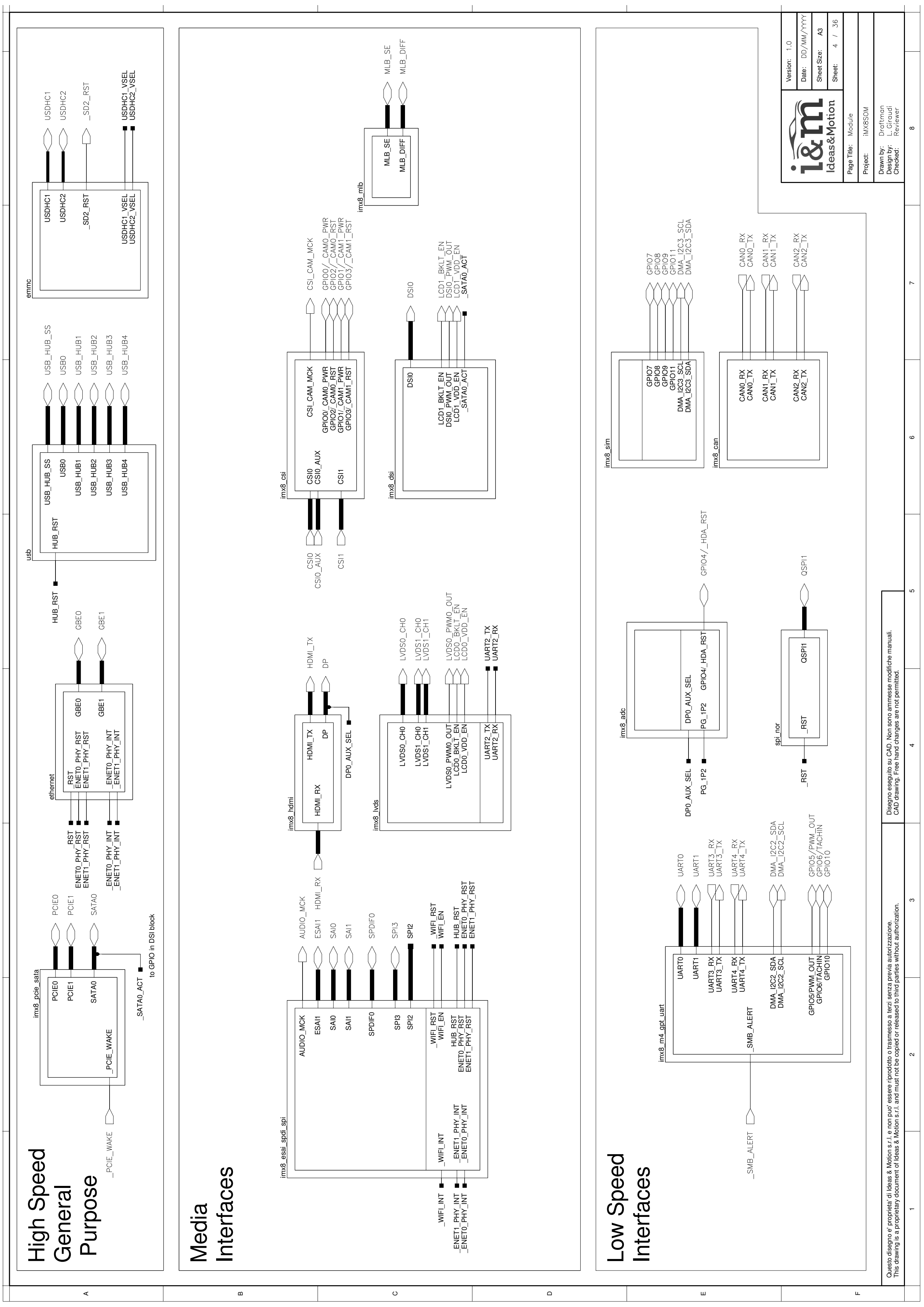


Figure A.3: First part of Top Level

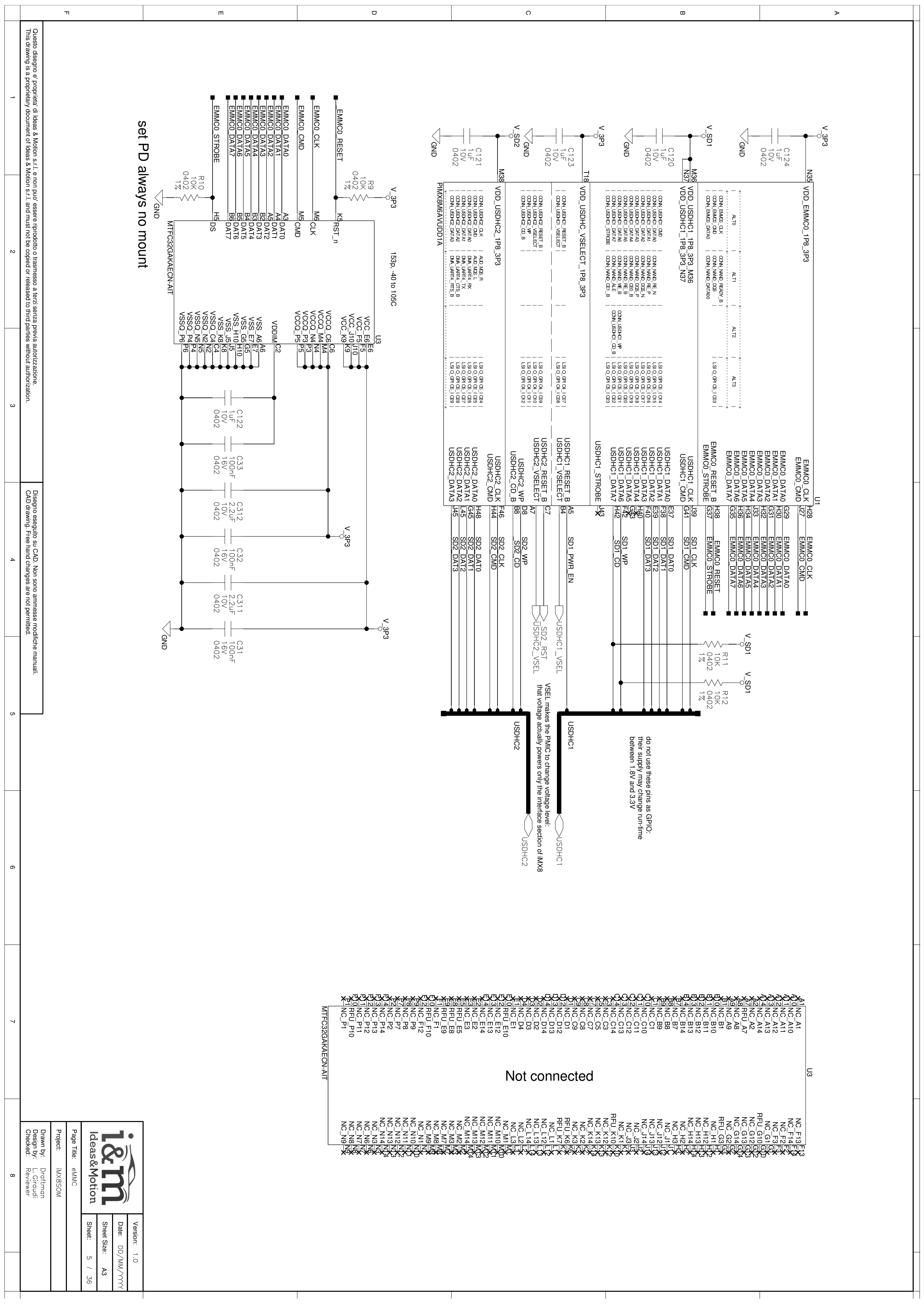

U1
PIMX8M6AVUDD1A
VDD_EMMC0_1P8_3P3
VDD_USDHC1_1P8_3P3_M36
VDD_USDHC1_1P8_3P3_N37
VDD_USDHC_VSELECT_1P8_3P3
VDD_USDHC2_1P8_3P3
USDHC1
USDHC2
do not use these pins as GPIO: their supply may change run-time between 1.8V and 3.3V
VSEL makes the PMIC to change voltage level: that voltage actually powers only the interface section of IMX8
U3
MTFC32GAKAECN-AIT
153p, -40 to 105C
set PD always no mount
Not connected
V_3P3
V_SD1
V_SD2
GND
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
i&m Ideas&Motion
Page Title: eMMC
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giraudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 5 / 36

Figure A.4: eMMC and SD interface

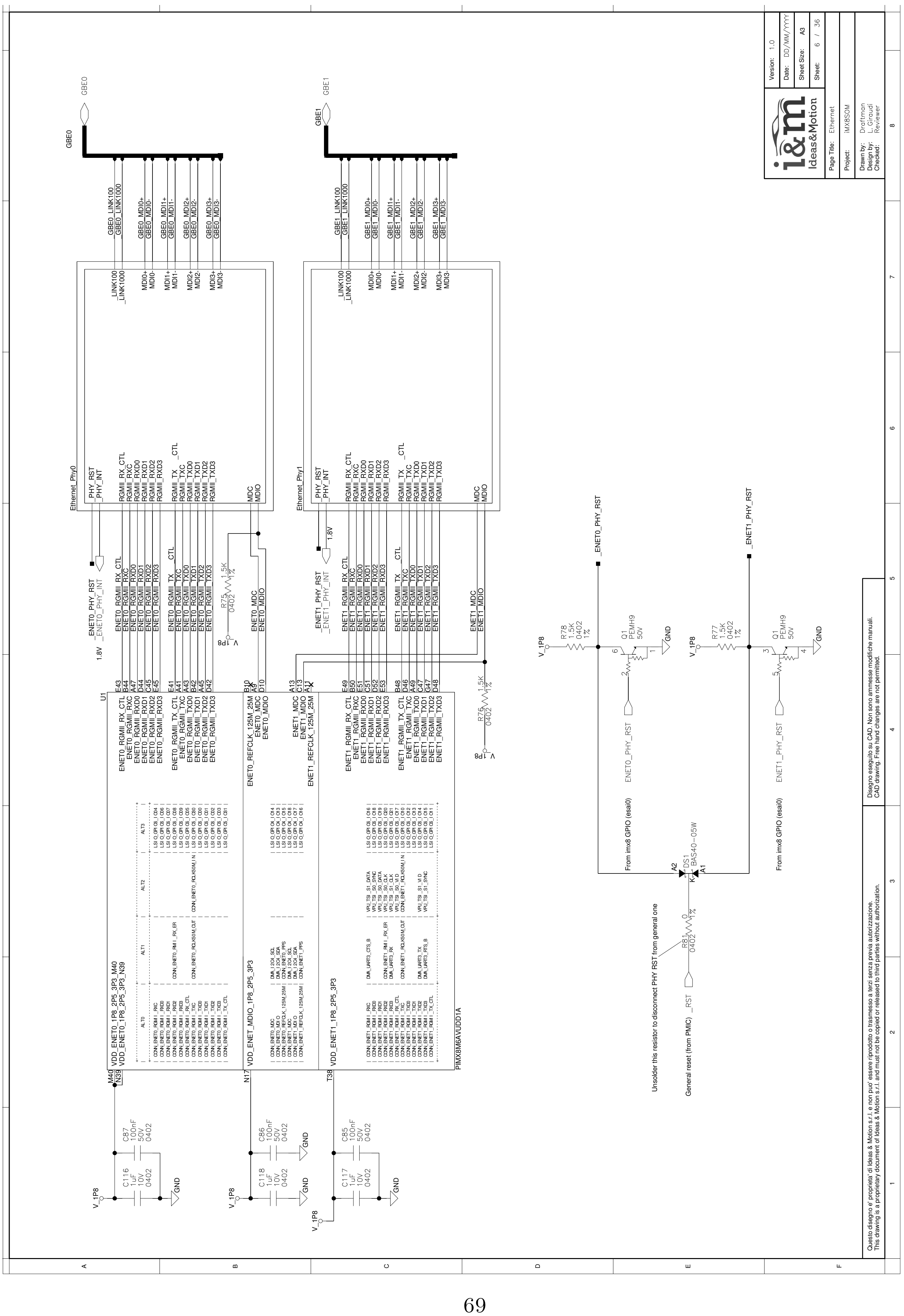
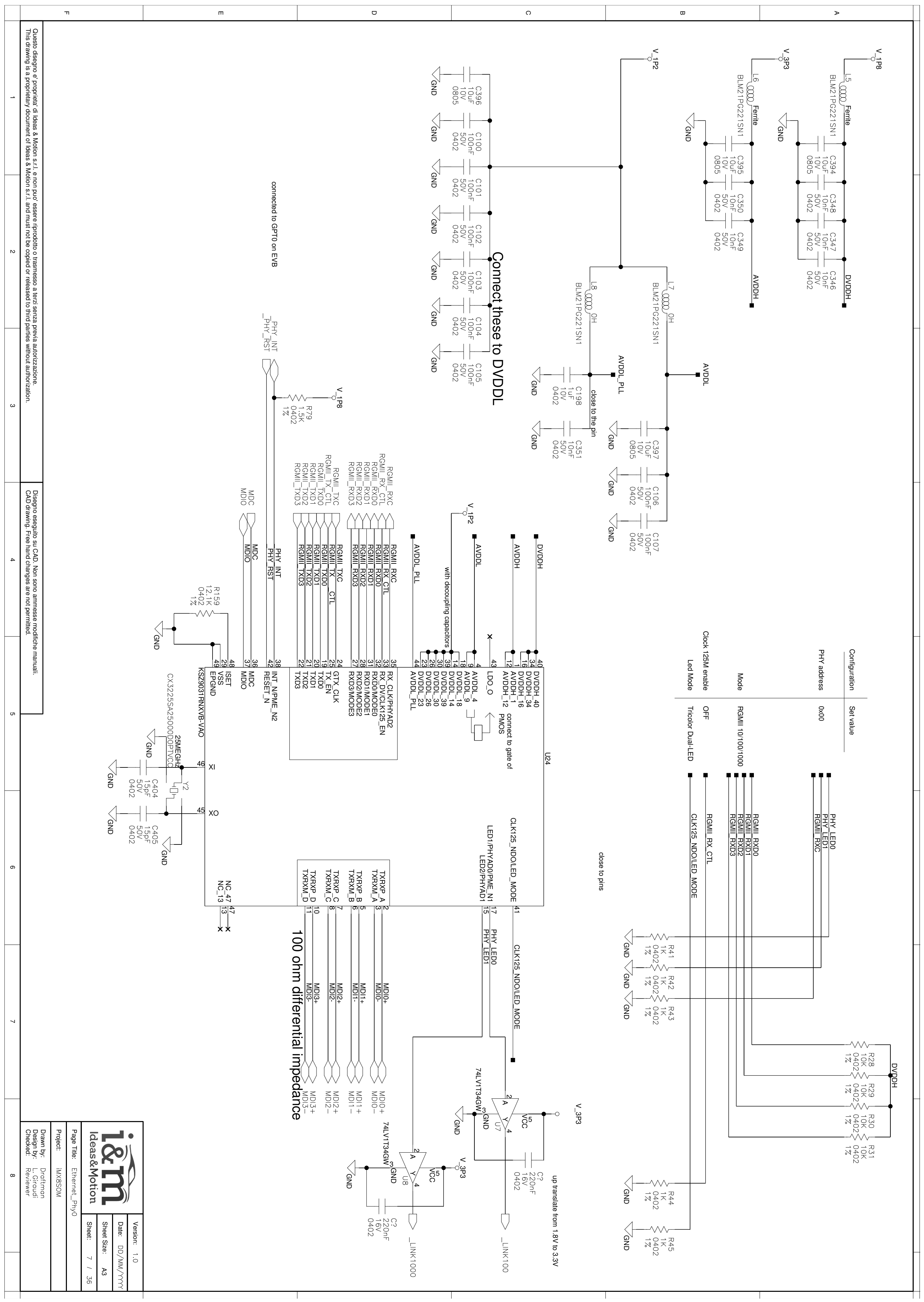

GBE0
GBE1
Ethernet_Phy0
Ethernet_Phy1
_ENET0_PHY_RST
_ENET1_PHY_RST
ENET0_PHY_RST
ENET1_PHY_RST
From imx8 GPIO (esai0)
General reset (from PMIC) _RST
Unsolder this resistor to disconnect PHY RST from general one
PIMX8M6AVUDD1A
U1
Q1 PEMH9 50V
DS1 BAS40-05W
GND
V_1P8
Page Title: Ethernet
Project: iMX8SOM
Version: 1.0
Sheet Size: A3
Sheet: 6 / 36
Ideas&Motion
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.

Figure A.5: Ethernet block of iMX8



Configuration
Set value
PHY address
0x00
Mode
RGMII 10/100/1000
Clock 125M enable
OFF
Led Mode
Tricolor Dual-LED
Connect these to DVDDL
close to the pin
close to pins
connected to GPIT0 on EVB
with decoupling capacitors
connect to gate of PMOS
100 ohm differential impedance
up translate from 1.8V to 3.3V
KSZ9031RNXVB-VAO
U24
U7
U8
74LV1T34GW
CX3225SA25000D0PTVCC
25MEGHZ
Y2
V_1P8
V_3P3
V_1P2
DVDDH
AVDDH
AVDDL
AVDDL_PLL
DVDDL
GND
_LINK100
_LINK1000
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
i&m Ideas&Motion
Page Title: Ethernet_Phy0
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giroudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 7 / 36

Figure A.6: Ethernet PHY

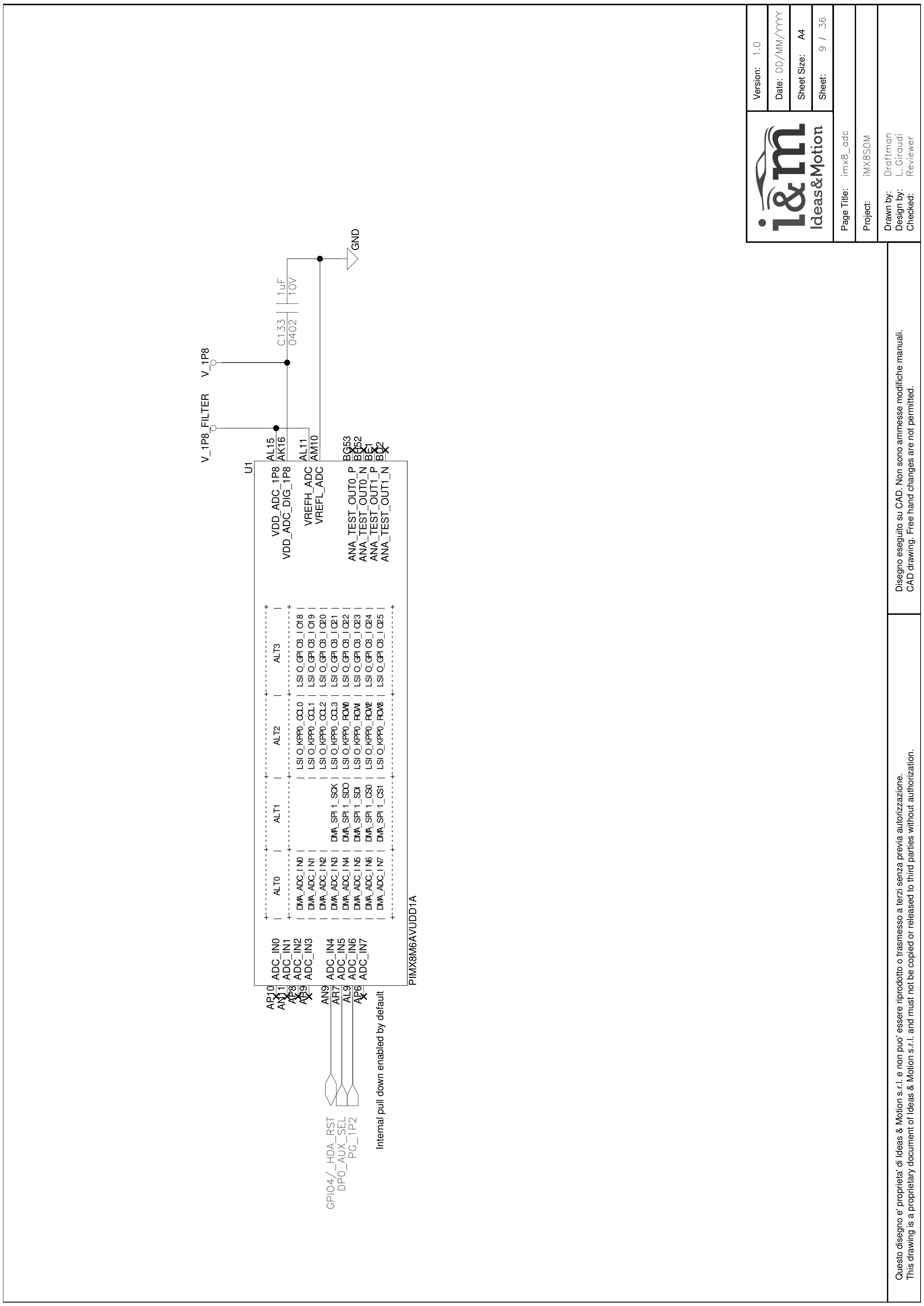


Figure A.7: ADC Block of iMX8

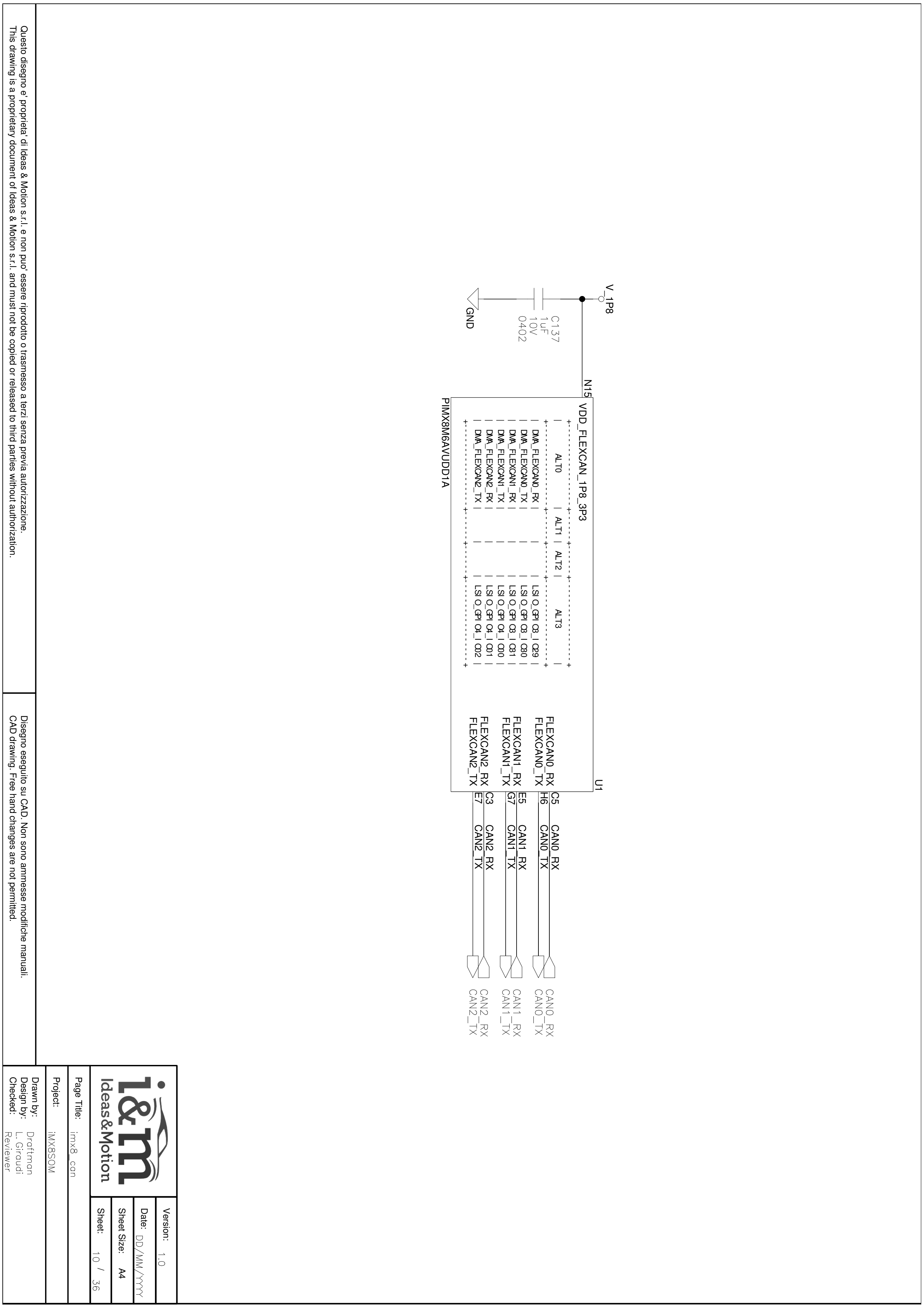


Figure A.8: Can Block of iMX8

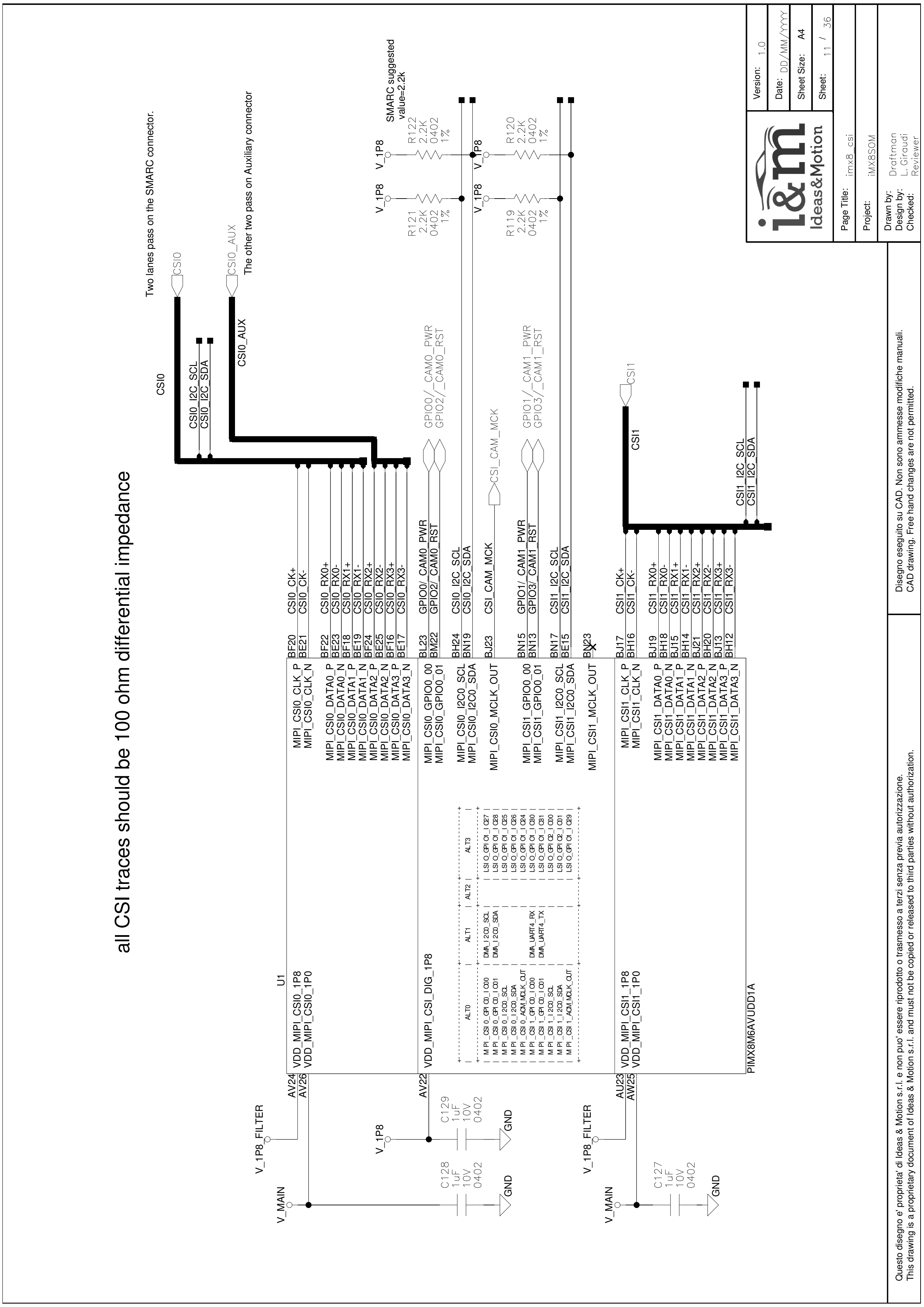


Figure A.9: CSI block of iMX8

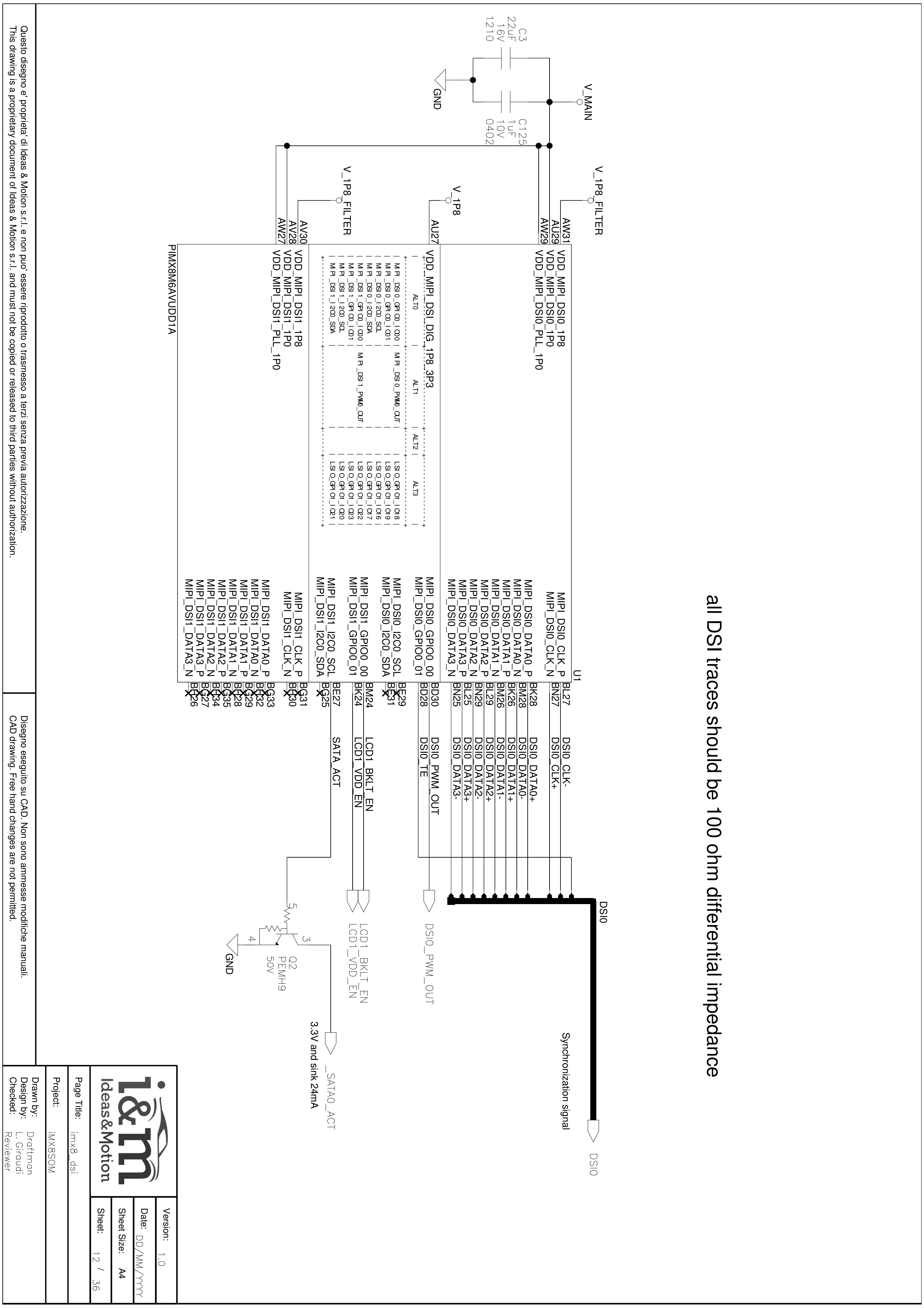


Figure A.10: DSI block of iMX8

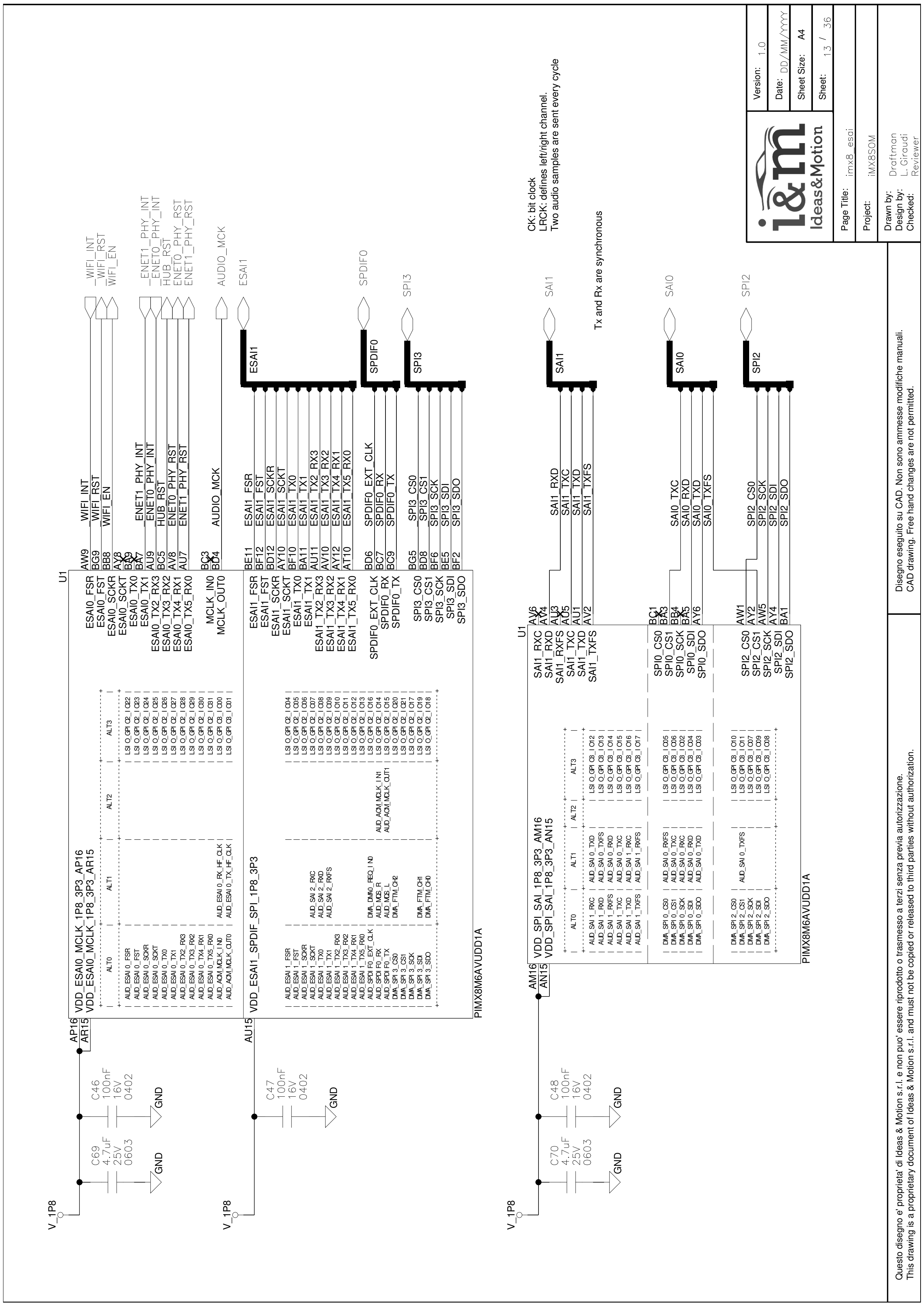

CK: bit clock
LRCK: defines left/right channel.
Two audio samples are sent every cycle
Tx and Rx are synchronous
WIFI_INT
WIFI_RST
WIFI_EN
ENET1_PHY_INT
ENET0_PHY_INT
HUB_RST
ENET0_PHY_RST
ENET1_PHY_RST
AUDIO_MCK
ESAI1
SPDIF0
SPI3
SAI1
SAI0
SPI2
U1
PIMX8M6AVUDD1A
VDD_ESAI0_MCLK_1P8_3P3_AP16
VDD_ESAI0_MCLK_1P8_3P3_AR15
VDD_ESAI1_SPDIF_SPI_1P8_3P3
VDD_SPI_SAI_1P8_3P3_AM16
VDD_SPI_SAI_1P8_3P3_AN15
V_1P8
C46 100nF 16V 0402
C47 100nF 16V 0402
C48 100nF 16V 0402
C69 4.7uF 25V 0603
C70 4.7uF 25V 0603
GND
Page Title: imx8_esai
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giraudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A4
Sheet: 13 / 36
Ideas&Motion
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.

Figure A.11: SAI/ESAI/SPI block of iMX8

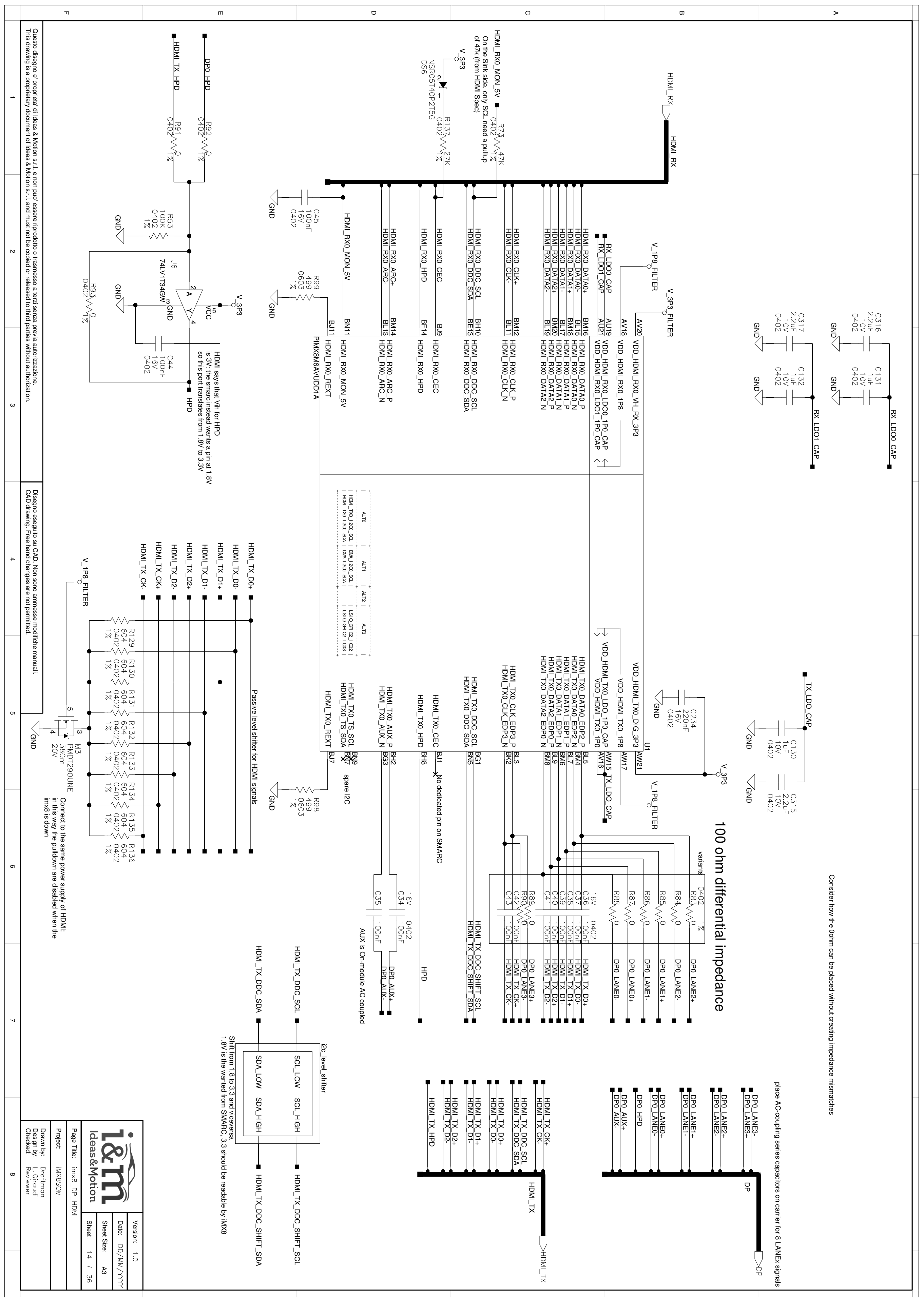

HDMI_RX
HDMI_TX
DP
100 ohm differential impedance
Passive level shifter for HDMI signals
i2c_level_shifter
Shift from 1.8 to 3.3 and viceversa
1.8V is the wanted from SMARC, 3.3 should be readable by iMX8
Consider how the 0ohm can be placed without creating impedance mismatches
place AC-coupling series capacitors on carrier for 8 LANEx signals
AUX is On-module AC coupled
No dedicated pin on SMARC
spare I2C
PIMX8M6AVUDD1A
74LV1T34GW
PMDT290UNE
i&m Ideas&Motion
Page Title: imx8_DP_HDMI
Project: iMX8SOM
Version: 1.0
Sheet Size: A3
Sheet: 14 / 36

Figure A.12: HDMI/DP block of iMX8 and level shifter

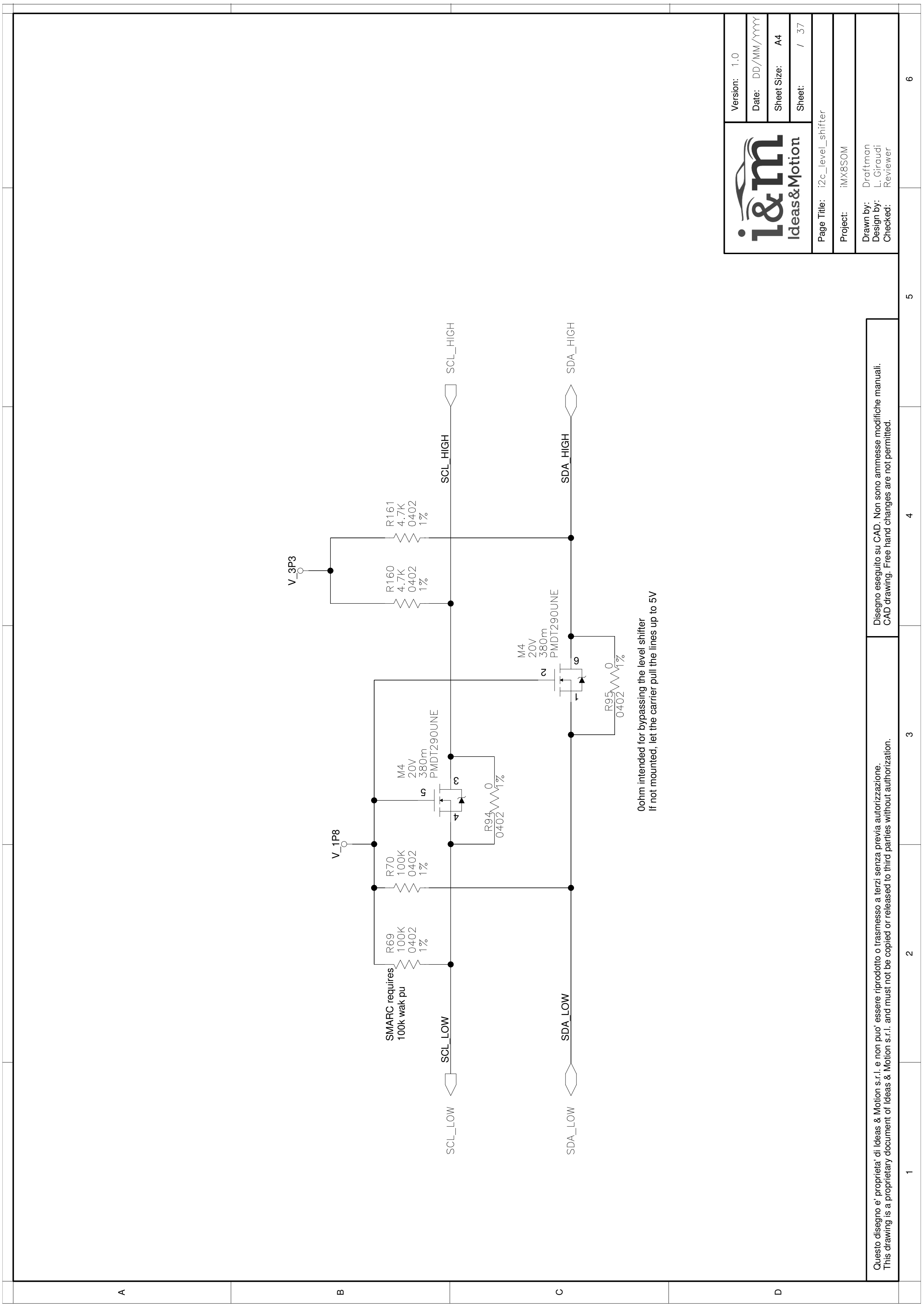

SCL_LOW
SDA_LOW
SCL_HIGH
SDA_HIGH
V_1P8
V_3P3
SMARC requires 100k wak pu
R69 100K 0402 1%
R70 100K 0402 1%
R160 4.7K 0402 1%
R161 4.7K 0402 1%
M4 20V 380m PMDT290UNE
R94 0402 0 1%
R95 0402 0 1%
0ohm intended for bypassing the level shifter
If not mounted, let the carrier pull the lines up to 5V
i&m Ideas&Motion
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A4
Sheet: / 37
Page Title: i2c_level_shifter
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giraudi
Checked: Reviewer
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.

Figure A.13: Level shifter of DDC channel

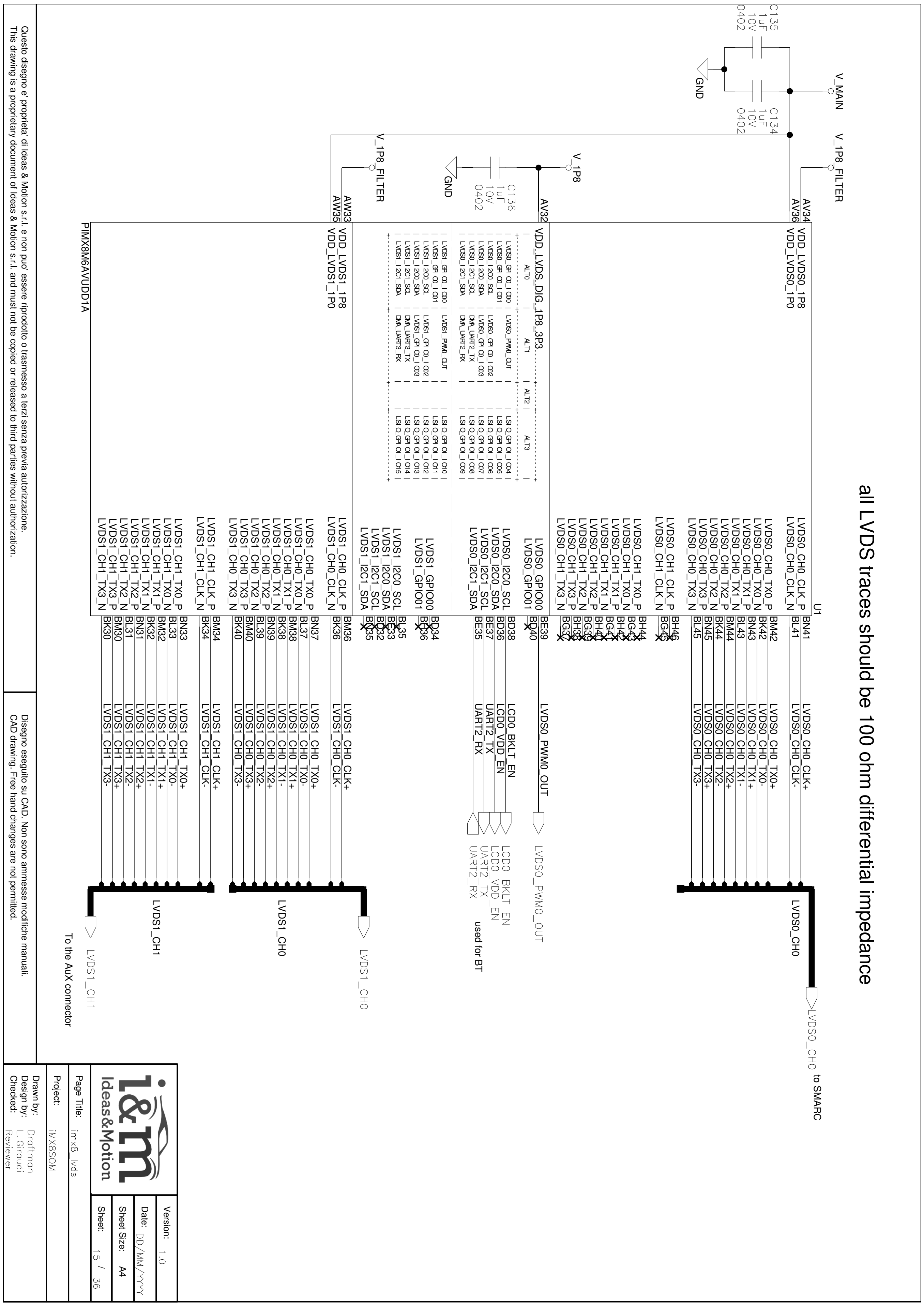


Figure A.14: LVDS block of iMX8

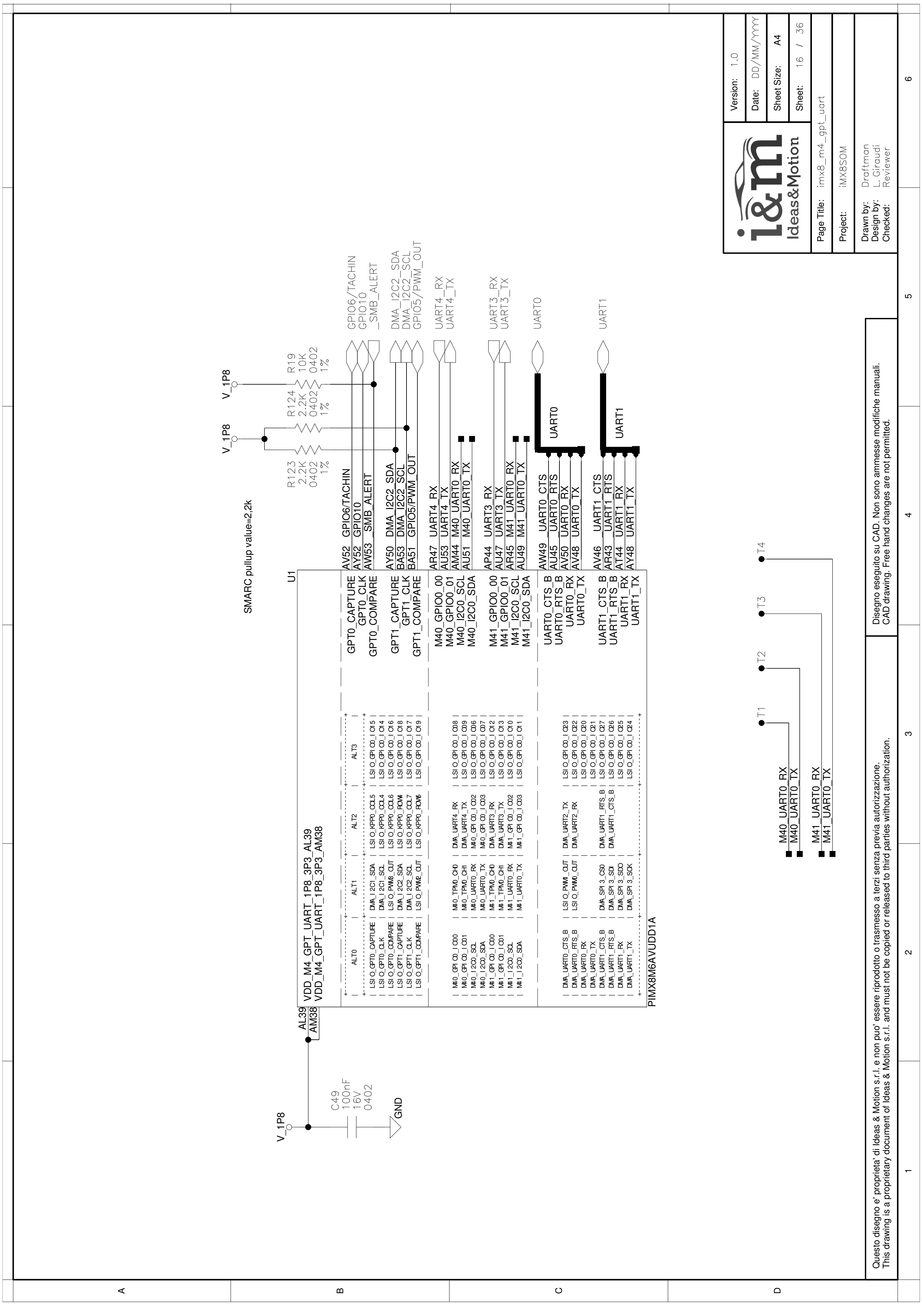


Figure A.15: M4 peripheral block of iMX8

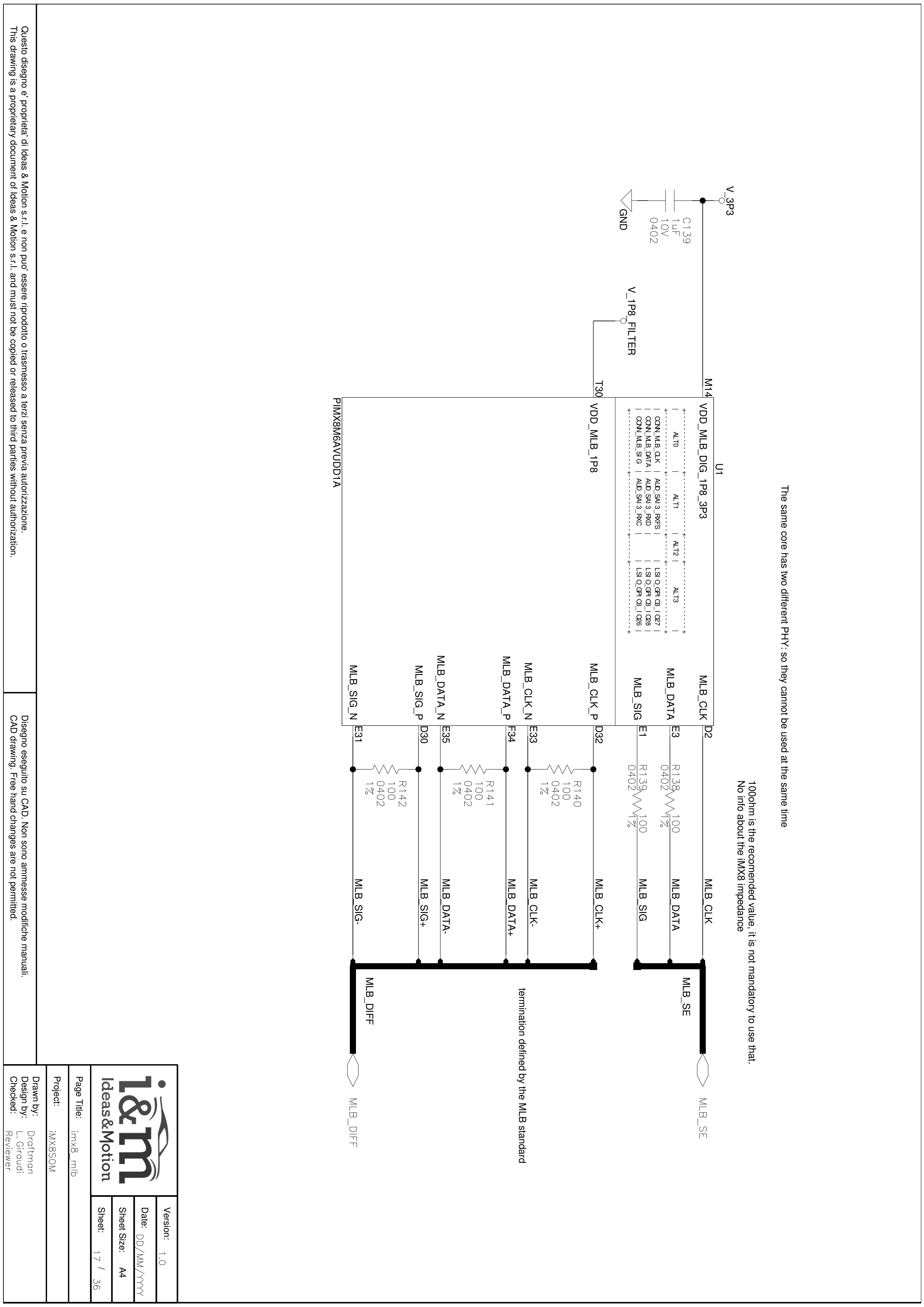


Figure A.16: MLB block of iMX8

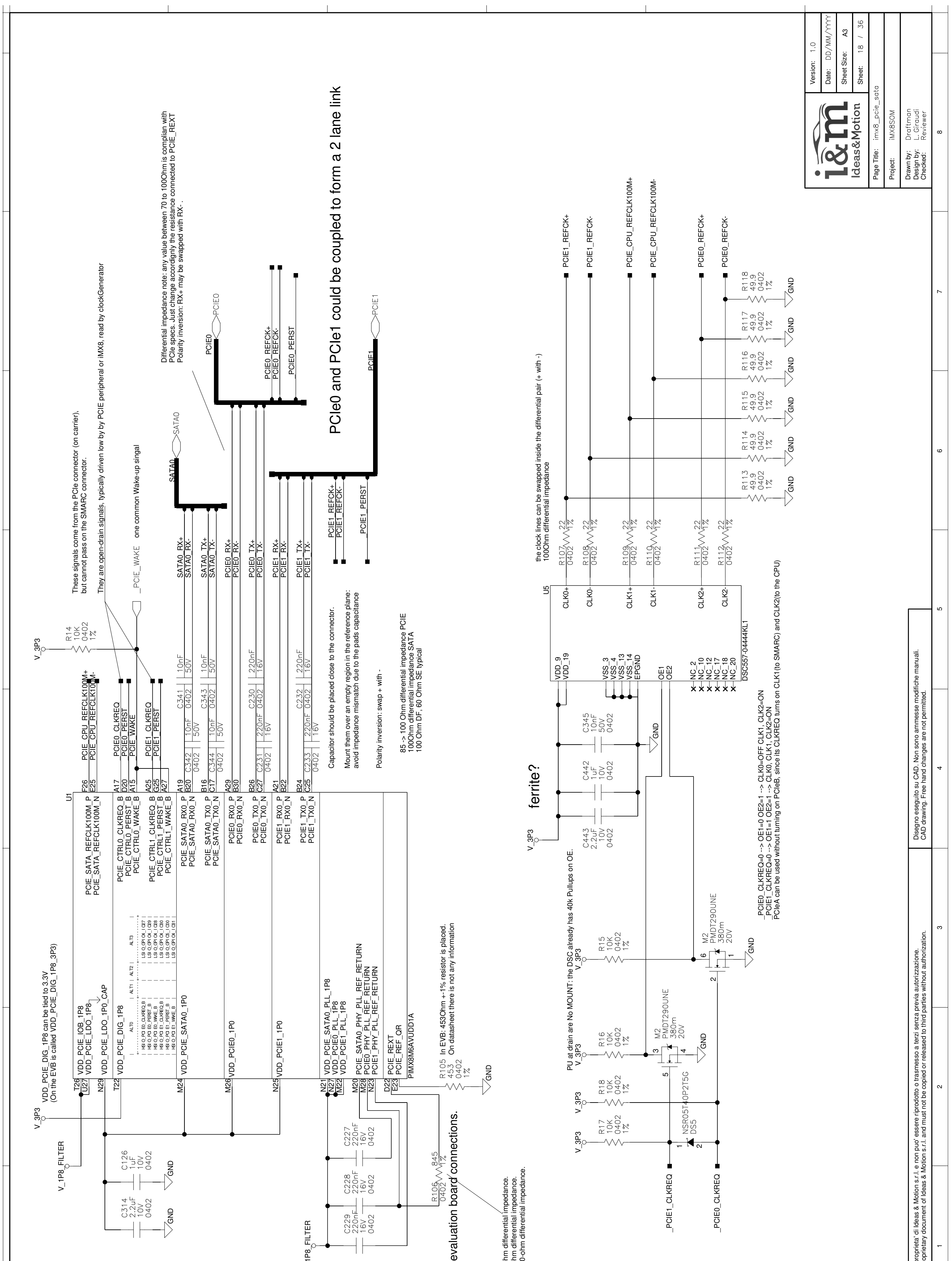

PCIe0 and PCIe1 could be coupled to form a 2 lane link
ferrite?
Followed evaluation board connections.
U1
U5
DSC557-04444KL1
PIMX8M6AVUDD1A
PCIE0_REFCK+
PCIE0_REFCK-
PCIE1_REFCK+
PCIE1_REFCK-
PCIE_CPU_REFCLK100M+
PCIE_CPU_REFCLK100M-
_PCIE1_CLKREQ
_PCIE0_CLKREQ
V_3P3
V_1P8_FILTER
GND
PCIE_REXT
845-Ohm for 85-ohm differential impedance.
909-Ohm for 90-ohm differential impedance.
1.02-kOhm for 100-ohm differential impedance.
Capacitor should be placed close to the connector.
Polarity inversion: swap + with -
85 -> 100 Ohm differential impedance PCIE
100Ohm differential impedance SATA
100 Ohm DF, 60 Ohm SE typical
the clock lines can be swapped inside the differential pair (+ with -)
100Ohm differential impedance
PU at drain are No MOUNT: the DSC already has 40k Pullups on OE.
Page Title: imx8_pcie_sata
Project: iMX8SOM
Version: 1.0
Sheet Size: A3
Sheet: 18 / 36
Ideas&Motion

Figure A.17: PCIe block of iMX8, with clock generator

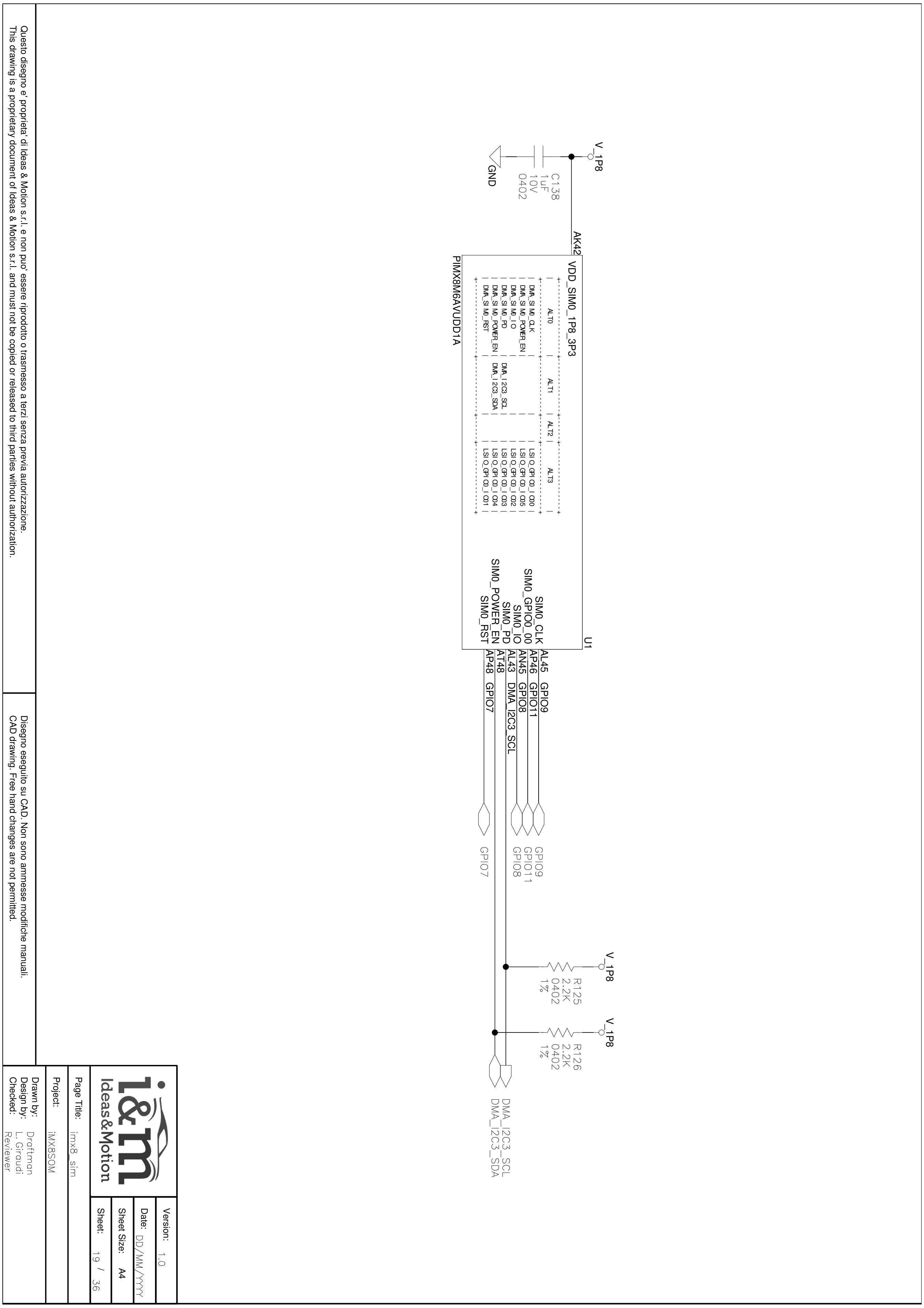


Figure A.18: SIM block of the iMX8. Used as GPIO

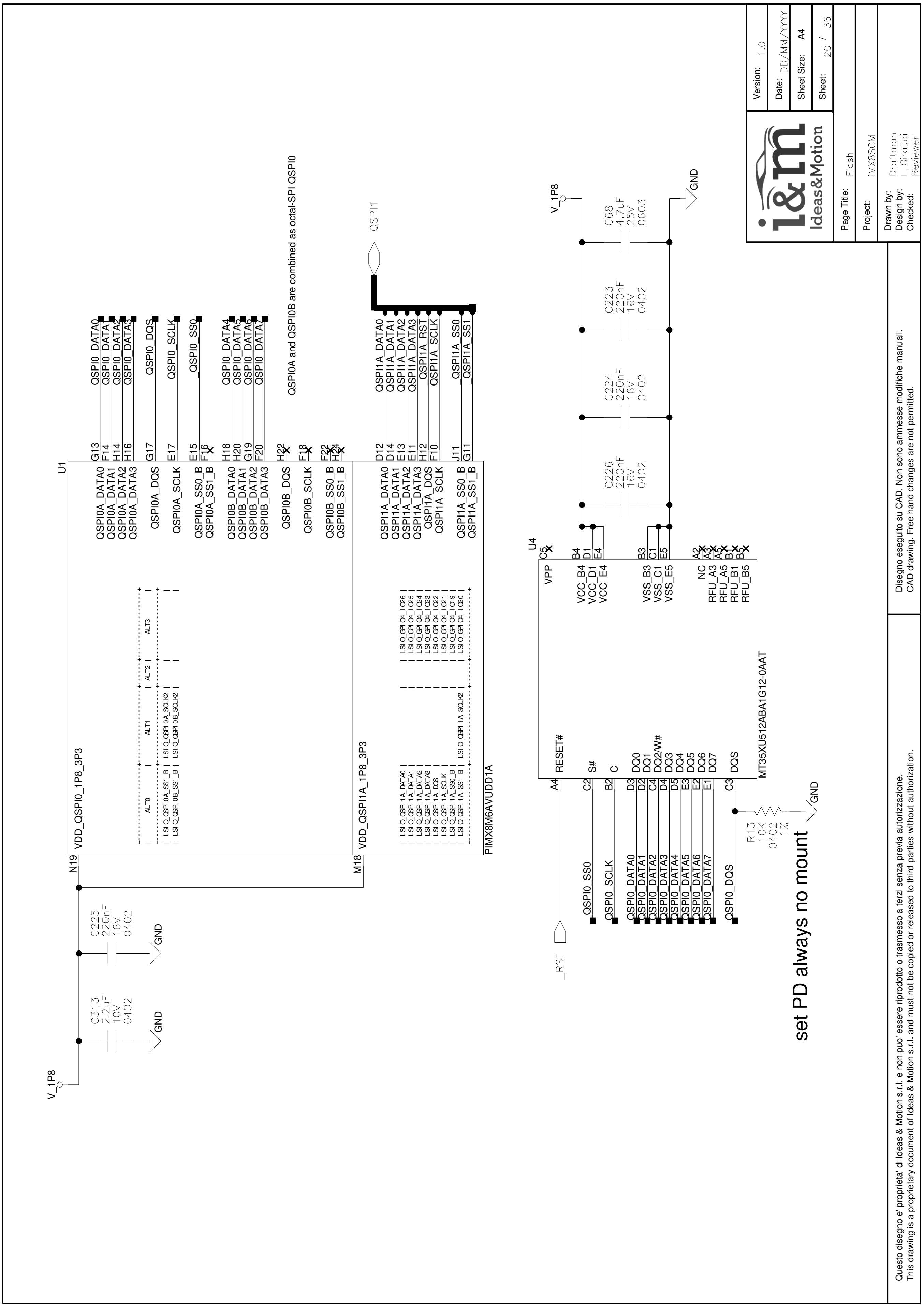


Figure A.19: QSPI block of iMX8 and NOR Flash.

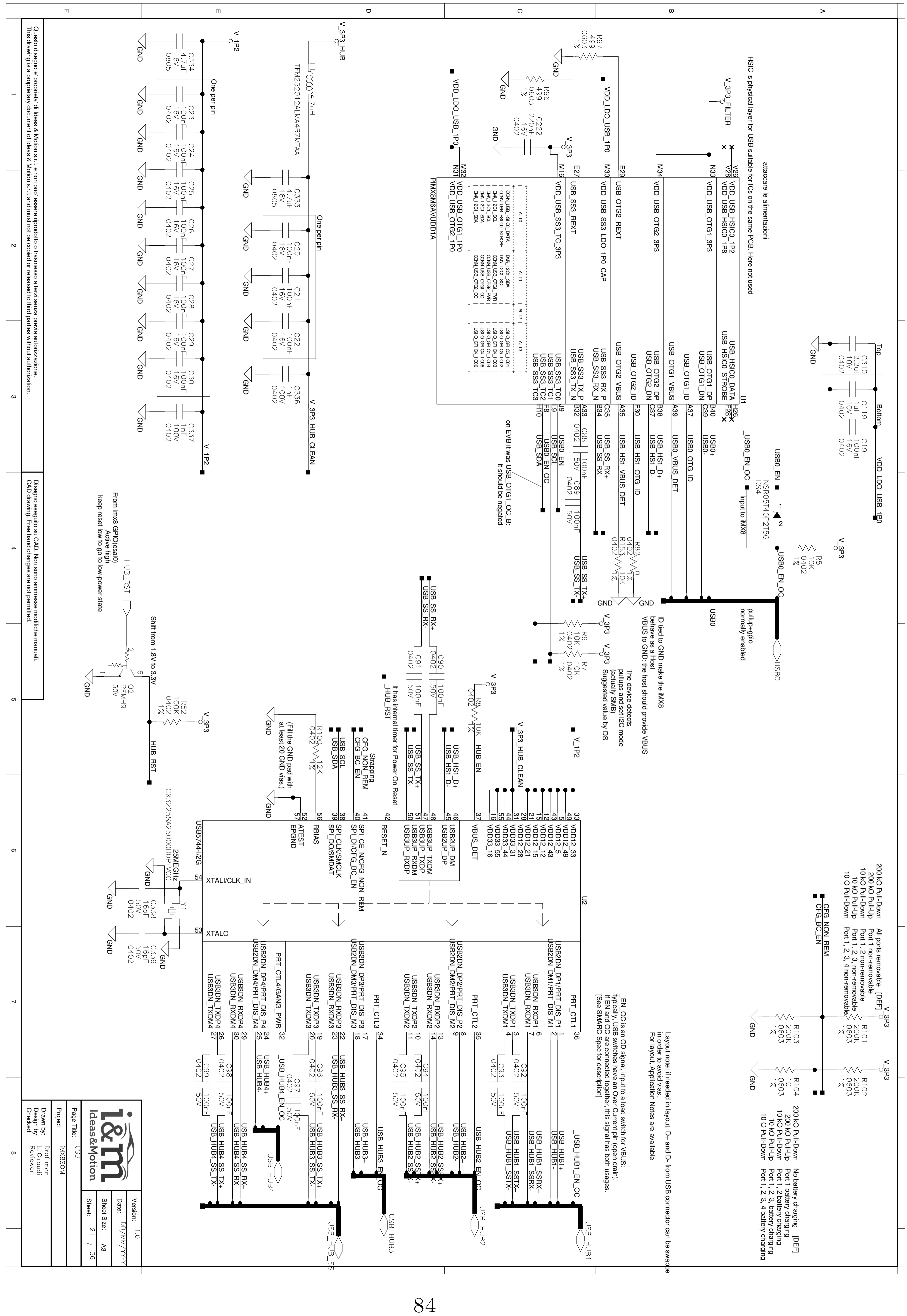

HSIC is physical layer for USB suitable for ICs on the same PCB. Here not used
attaccare le alimentazioni
V_3P3_FILTER
VDD_LDO_USB_1P0
V_3P3
U1
PIMX8M6AVUDD1A
VDD_USB_HSIC0_1P2
VDD_USB_HSIC0_1P8
VDD_USB_OTG1_3P3
VDD_USB_OTG2_3P3
USB_OTG2_REXT
VDD_USB_SS3_LDO_1P0_CAP
USB_SS3_REXT
VDD_USB_SS3_TC_3P3
VDD_USB_OTG1_1P0
VDD_USB_OTG2_1P0
USB_HSIC0_DATA
USB_HSIC0_STROBE
USB_OTG1_DP
USB_OTG1_DN
USB_OTG1_ID
USB_OTG1_VBUS
USB_OTG2_DP
USB_OTG2_DN
USB_OTG2_ID
USB_OTG2_VBUS
USB_SS3_RX_P
USB_SS3_RX_N
USB_SS3_TX_P
USB_SS3_TX_N
USB0+
USB0-
USB0_OTG_ID
USB0_VBUS_DET
USB_HS1_D+
USB_HS1_D-
USB_HS1_OTG_ID
USB_HS1_VBUS_DET
USB_SS_RX+
USB_SS_RX-
USB_SS_TX+
USB_SS_TX-
USB0_EN
USB0_EN_OC
USB0_SCL
USB0_SDA
on EVB it was USB_OTG1_OC_B: it should be negated
Input to iMX8
pullup+gpio normally enabled
ID tied to GND make the iMX8 behave as a Host
VBUS to GND: the host should provide VBUS
The device detects pullups and set I2C mode (actually SMB)
Suggested value by DS
USB0
V_3P3_HUB
TFM252012ALMA4R7MTAA
V_3P3_HUB_CLEAN
V_1P2
One per pin
GND
From imx8 GPIO(esai0)
Active high
keep reset low to go to low-power state
Shift from 1.8V to 3.3V
HUB_RST
PEMH9
It has internal timer for Power On Reset
Strapping
CFG_NON_REM
CFG_BC_EN
(Fill the GND pad with at least 20 GND vias.)
U2
USB5744-I/2G
XTALI/CLK_IN
XTALO
25MEGHz
CX3225SA25000D0PTVCC
VBUS_DET
RESET_N
RBIAS
ATEST
EPGND
200 kO Pull-Down All ports removable [DEF]
200 KO Pull-Up Port 1 non-removable
10 KO Pull-Down Port 1, 2 non-removable
10 KO Pull-Up Port 1, 2, 3, non-removable
10 O Pull-Down Port 1, 2, 3, 4 non-removable
200 KO Pull-Down No battery charging [DEF]
200 KO Pull-Up Port 1 battery charging
10 KO Pull-Down Port 1, 2 battery charging
10 KO Pull-Up Port 1, 2, 3, battery charging
10 O Pull-Down Port 1, 2, 3, 4 battery charging
Layout note: if needed in layout, D+ and D- from USB connector can be swapped in order to avoid vias.
For layout, Application Notes are available
_EN_OC is an OD signal, input to a load switch for VBUS: typically, USB switches have an Over Current pin (open drain).
If EN and OC are connected together, this signal has both usages.
[See SMARC Spec for description]
USB_HUB1
USB_HUB2
USB_HUB3
USB_HUB4
USB_HUB_SS
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
i&m Ideas&Motion
Page Title: USB
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giroudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 21 / 36

Figure A.20: USB block of iMX8 and Hub

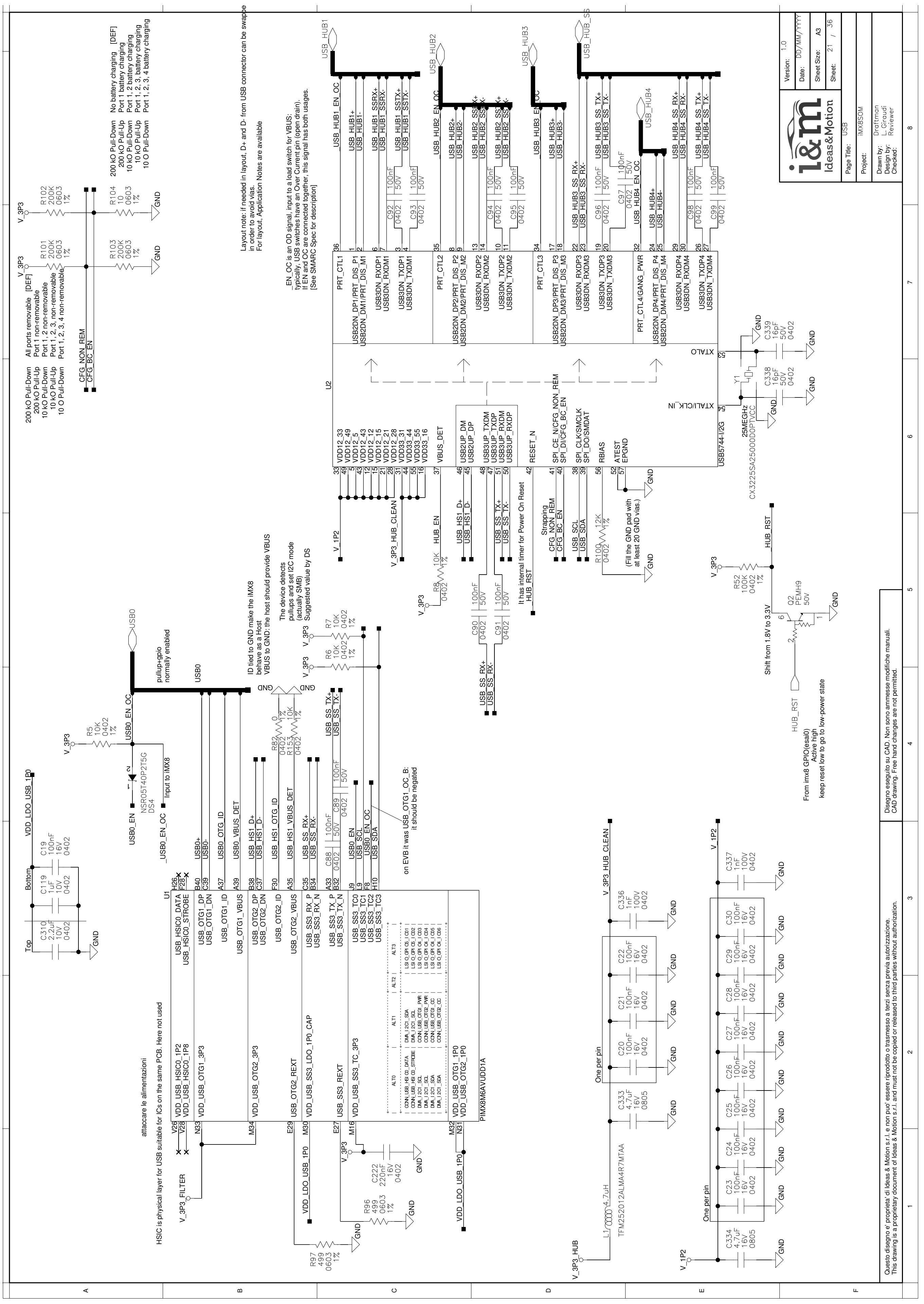

U1
PIMX8M6AVUDD1A
U2
USB5744-I/2G
USB0
USB_HUB1
USB_HUB2
USB_HUB3
USB_HUB4
USB_SS
V_3P3
V_1P2
V_3P3_HUB
V_3P3_HUB_CLEAN
VDD_LDO_USB_1P0
GND
HUB_RST
USB0_EN
USB0_EN_OC
CFG_NON_REM
CFG_BC_EN
25MEGHz
CX3225SA25000D0PTVCC
TFM252012ALM4R7MTAA
NSR05T40P2T5G
PEMH9
Shift from 1.8V to 3.3V
From imx8 GPIO(esai0) Active high keep reset low to go to low-power state
One per pin
HSIC is physical layer for USB suitable for ICs on the same PCB. Here not used
ID tied to GND make the iMX8 behave as a Host
VBUS to GND: the host should provide VBUS
Layout note: if needed in layout, D+ and D- from USB connector can be swapped in order to avoid vias. For layout, Application Notes are available
_EN_OC is an OD signal, input to a load switch for VBUS: typically, USB switches have an Over Current pin (open drain). If EN and OC are connected together, this signal has both usages. [See SMARC Spec for description]
It has internal timer for Power On Reset
(Fill the GND pad with at least 20 GND vias.)
on EVB it was USB_OTG1_OC_B: it should be negated
Diesgno eseguito su CAD. Non sono ammesse modifiche manuali. CAD drawing. Free hand changes are not permitted.
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione. This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
i&m Ideas&Motion
Page Title: USB
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giraudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 21 / 36

Figure A.21: USB block of iMX8 and Hub

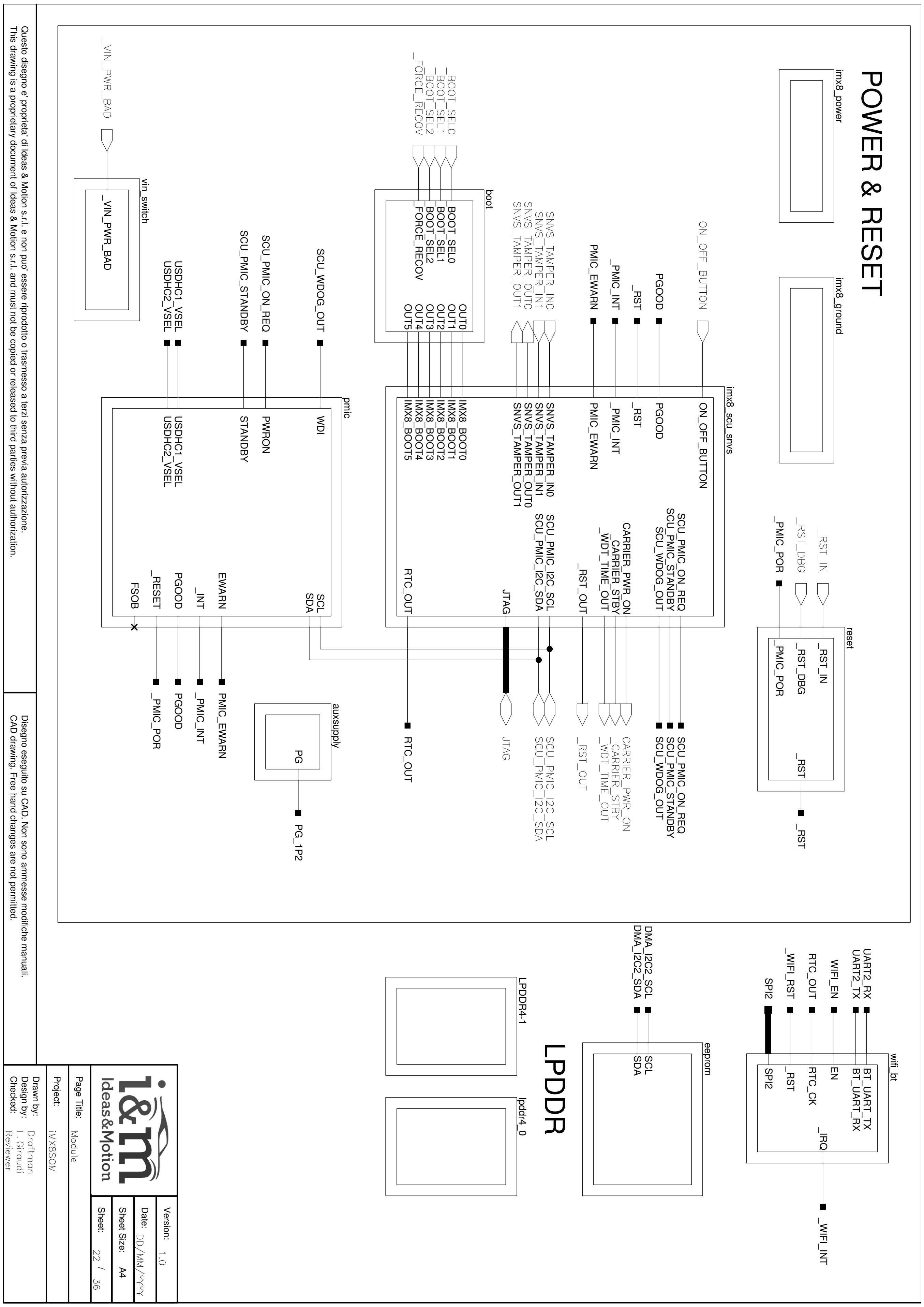


Figure A.22: Second part of Top Level

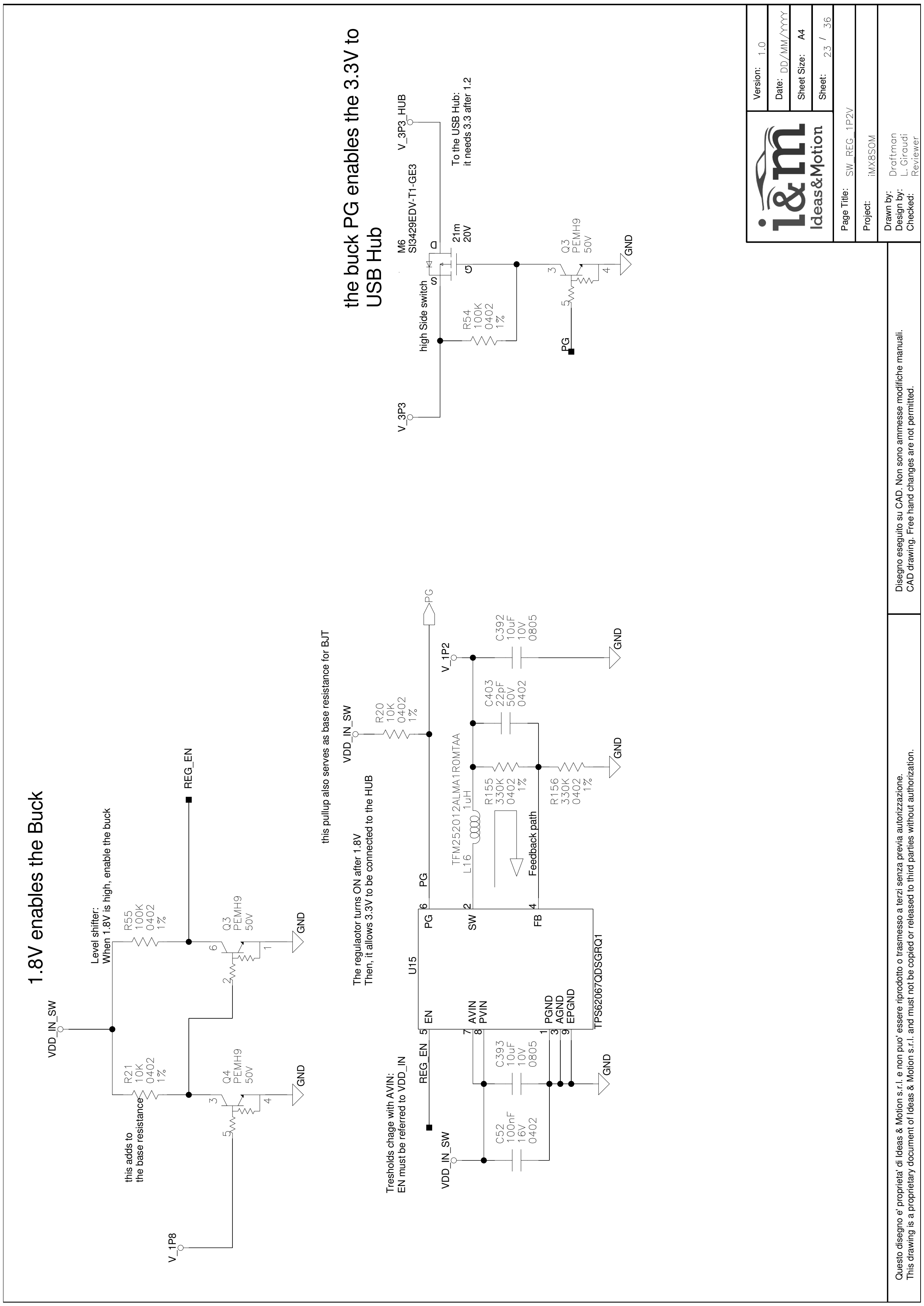
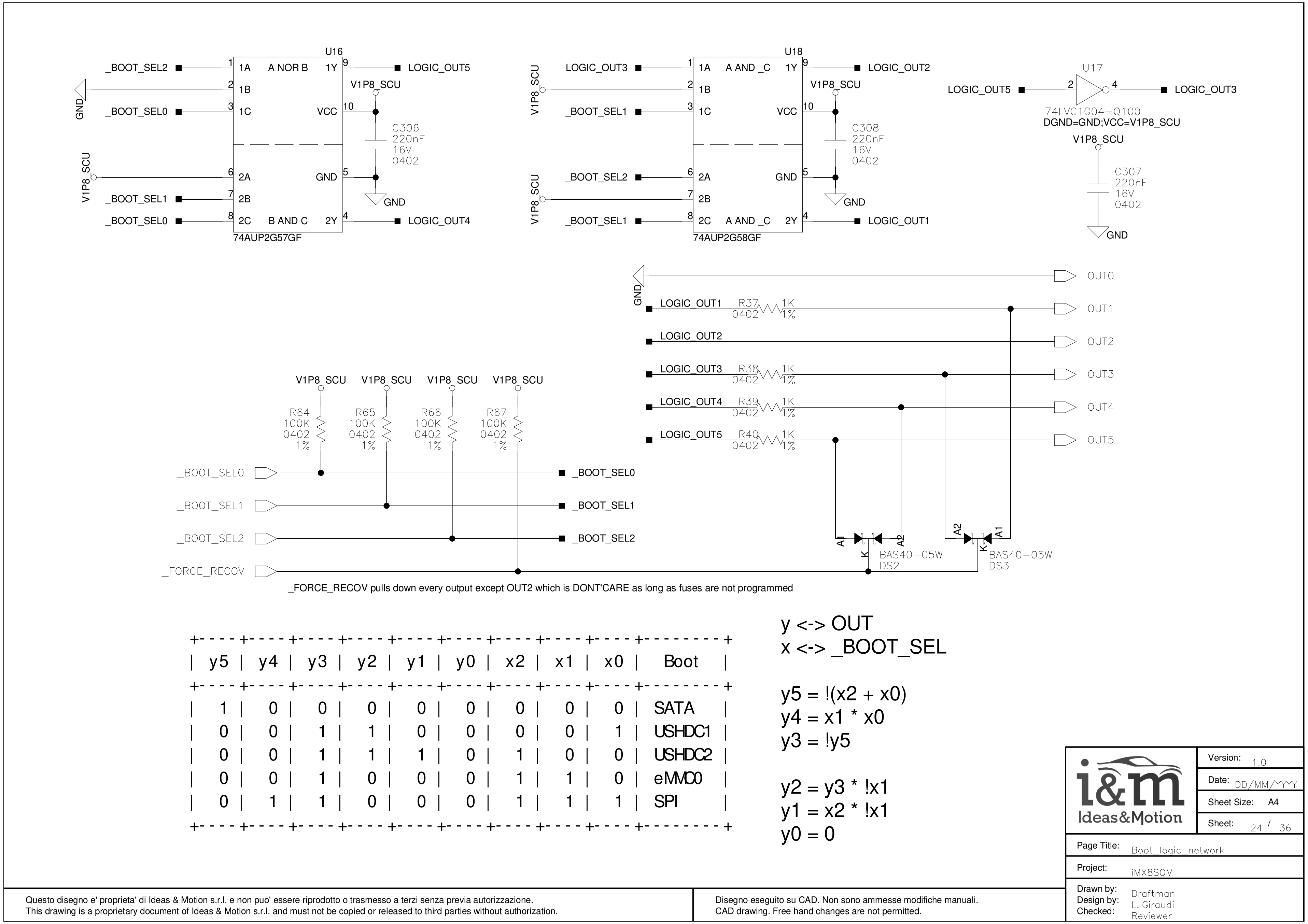


Figure A.23: Buck generating 1.2V



_FORCE_RECOV pulls down every output except OUT2 which is DONT'CARE as long as fuses are not programmed

y5	y4	y3	y2	y1	y0	x2	x1	x0	Boot
1	0	0	0	0	0	0	0	0	SATA
0	0	1	1	0	0	0	0	1	USHDC1
0	0	1	1	1	0	1	0	0	USHDC2
0	0	1	0	0	0	1	1	0	eMMC0
0	1	1	0	0	0	1	1	1	SPI

y <-> OUT
x <-> _BOOT_SEL

y5 = !(x2 + x0)
y4 = x1 * x0
y3 = !y5

y2 = y3 * !x1
y1 = x2 * !x1
y0 = 0

Version:	1.0
Date:	DD/MM/YYYY
Sheet Size:	A4
Sheet:	24 / 36
Page Title:	Boot_logic_network
Project:	iMX8SOM
Drawn by:	Draftman
Design by:	L. Giraudi
Checked:	Reviewer

Figure A.24: Boot selection logical net

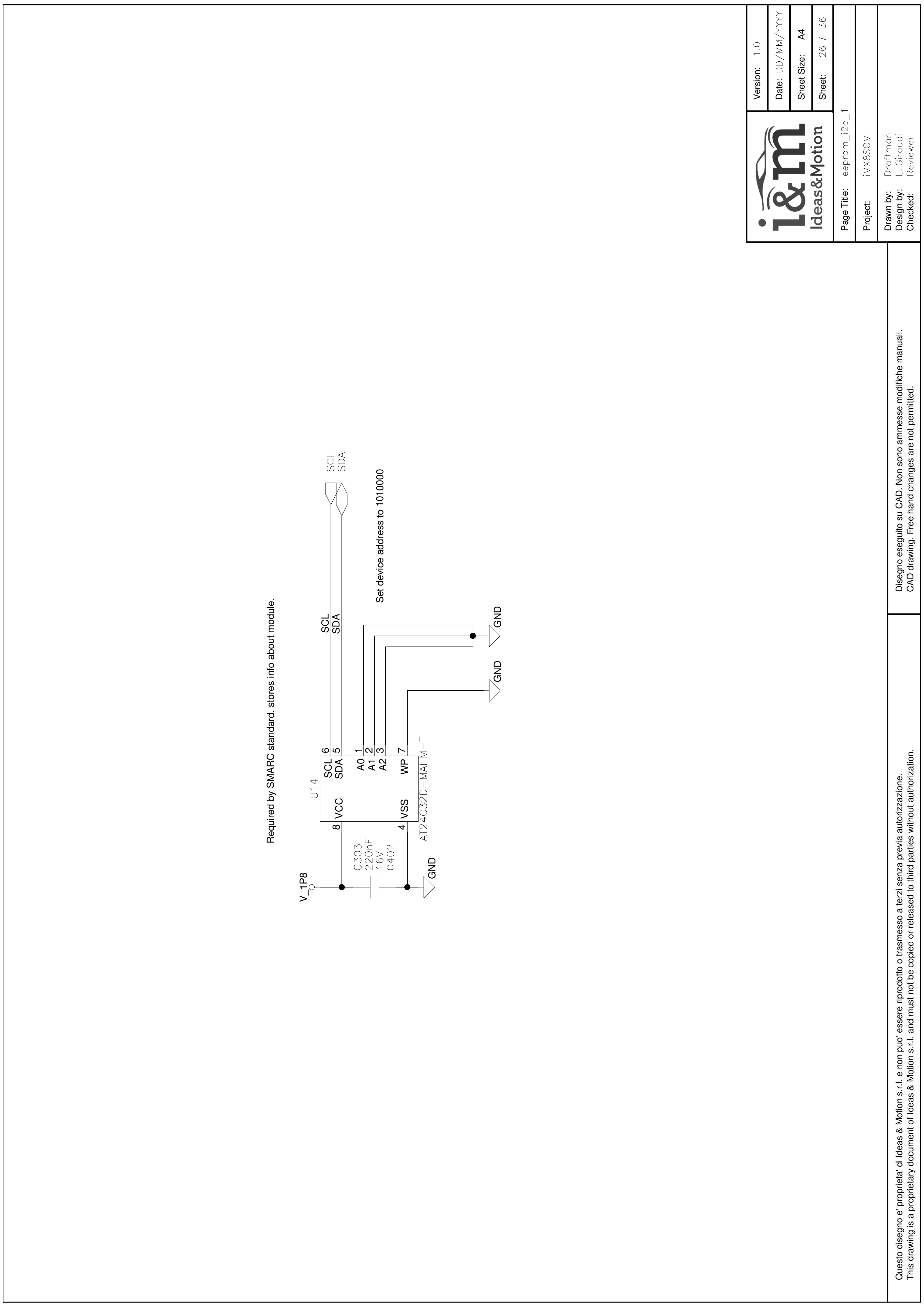

Required by SMARC standard, stores info about module.
V_1P8
C303
220nF
16V
0402
GND
U14
8 VCC
4 VSS
SCL 6
SDA 5
A0 1
A1 2
A2 3
WP 7
AT24C32D-MAHM-T
SCL
SDA
GND
GND
Set device address to 1010000
SCL
SDA
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A4
Sheet: 26 / 36
i&m Ideas&Motion
Page Title: eeprom_i2c_1
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giraudi
Checked: Reviewer
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.

Figure A.25: I2C EEPROM

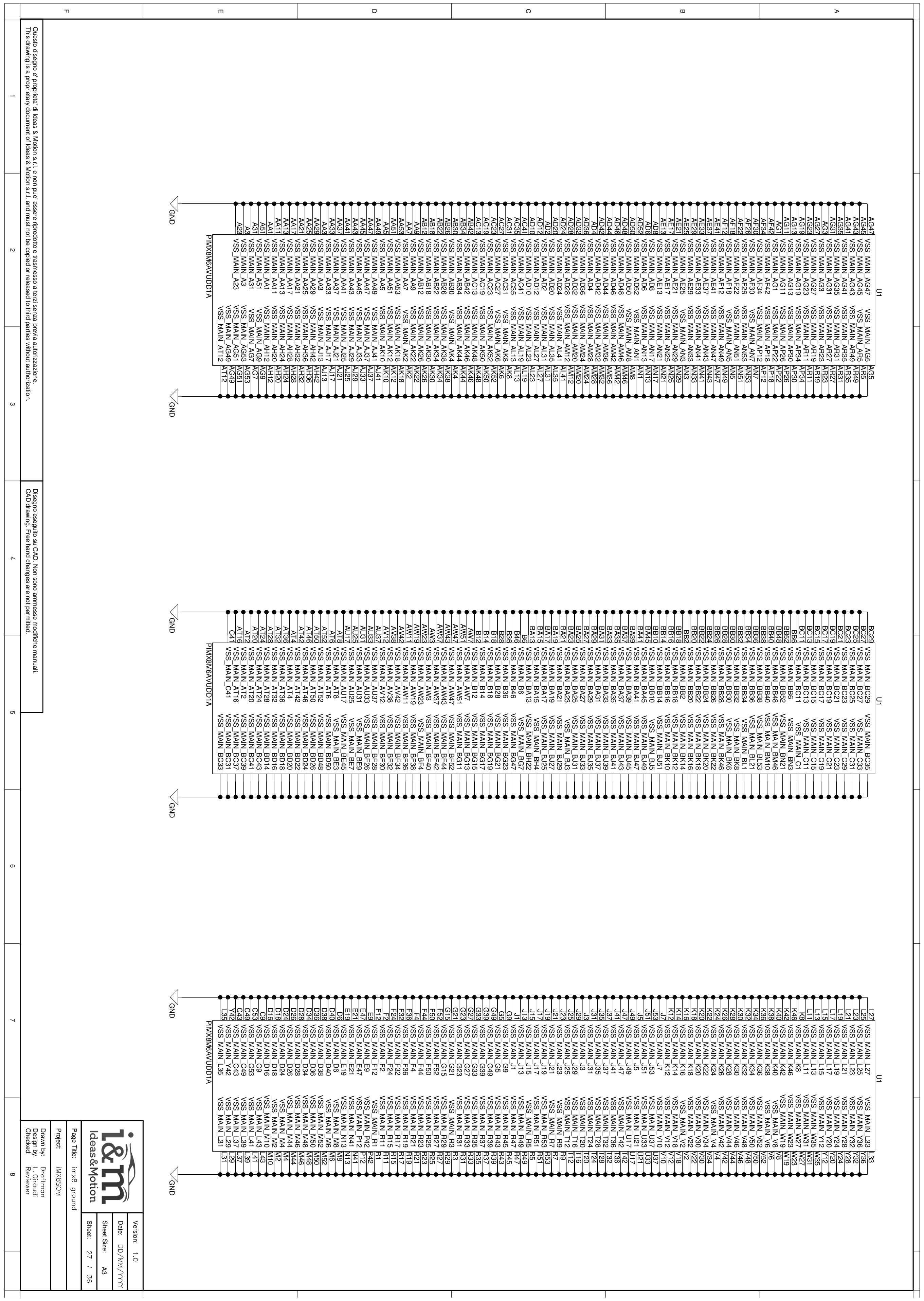

U1
PIMX8M6AVUDD1A
GND
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
i&m Ideas&Motion
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 27 / 36
Page Title: imx8_ground
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giroudi
Checked: Reviewer

Figure A.26: Ground block iMX8

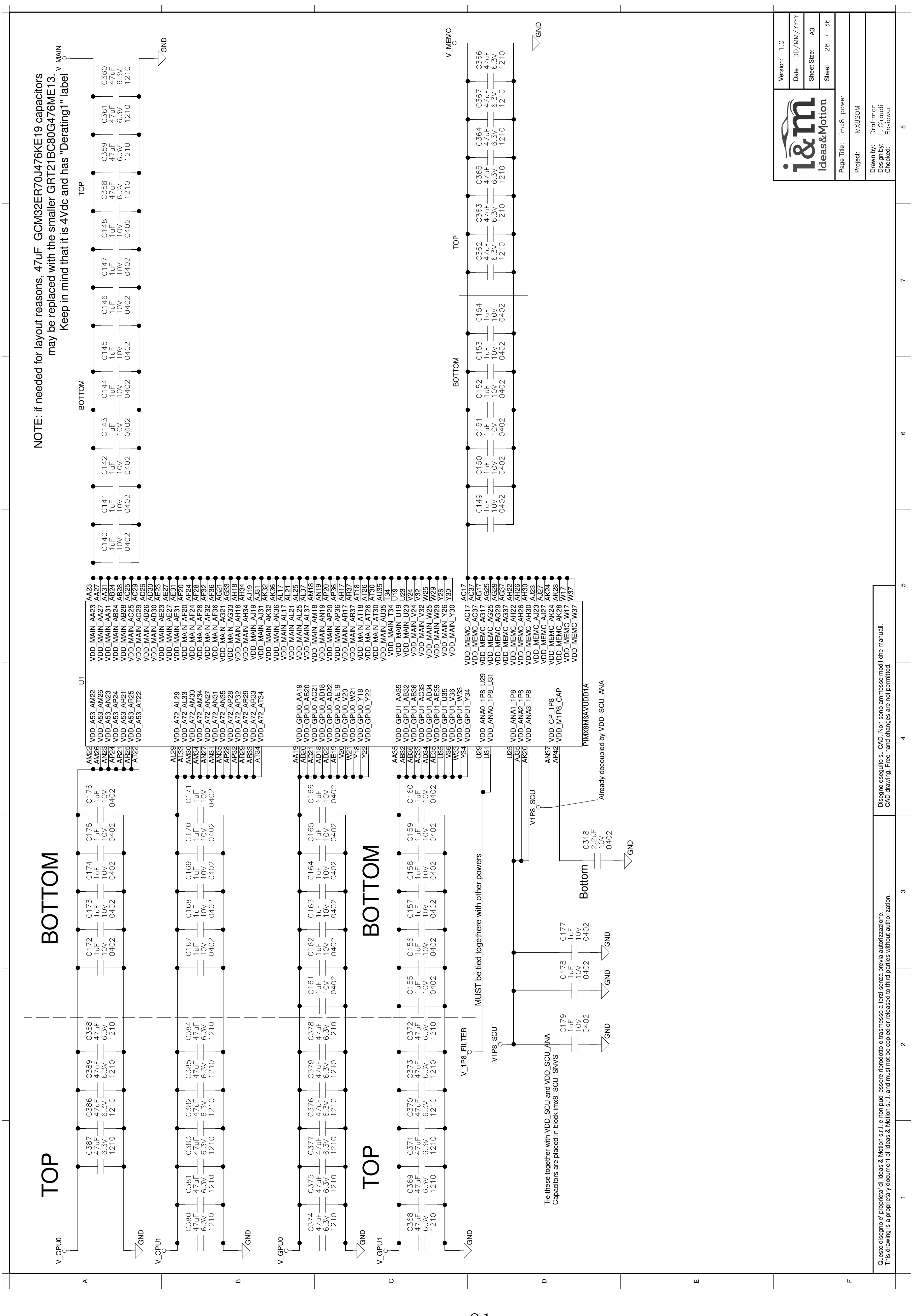

NOTE: if needed for layout reasons, 47uF GCM32ER70J476KE19 capacitors
may be replaced with the smaller GRT21BC80G476ME13.
Keep in mind that it is 4Vdc and has "Derating1" label
V_MAIN
GND
TOP
BOTTOM
V_MEMC
V_CPU0
V_CPU1
V_GPU0
V_GPU1
U1
PIMX8M6AVUDD1A
MUST be tied togethere with other powers
V_1P8_FILTER
V1P8_SCU
Already decoupled by VDD_SCU_ANA
Tie these together with VDD_SCU and VDD_SCU_ANA
Capacitors are placed in block imx8_SCU_SNVS
Bottom
C318 2.2uF 10V 0402
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
i&m Ideas&Motion
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 28 / 36
Page Title: imx8_power
Project: iMX8SOM
Drawn by: Draftman
Design by: L.Giraudi
Checked: Reviewer

Figure A.27: Main power block of iMX8

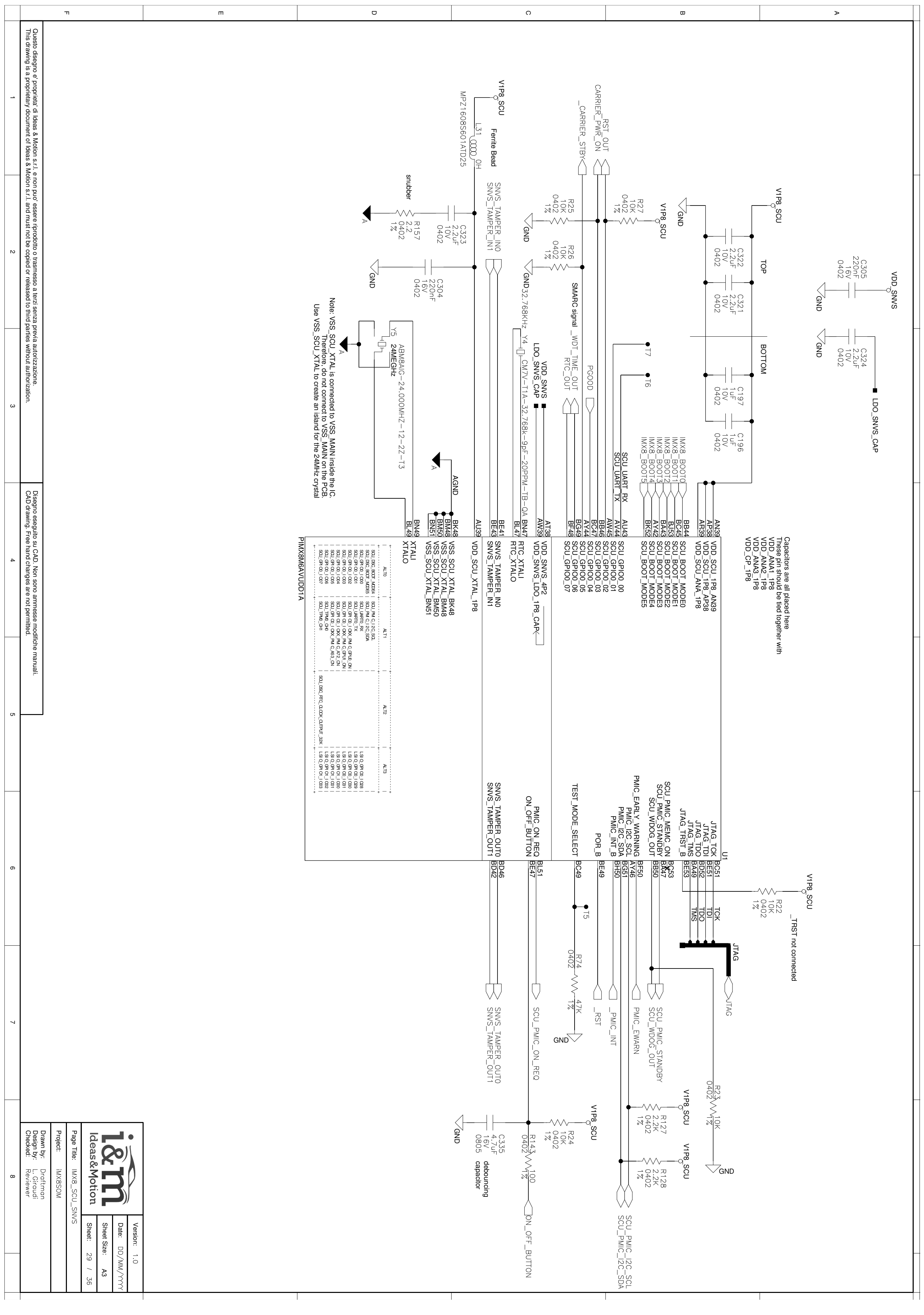


Figure A.28: SCU/SNVS block of iMX8

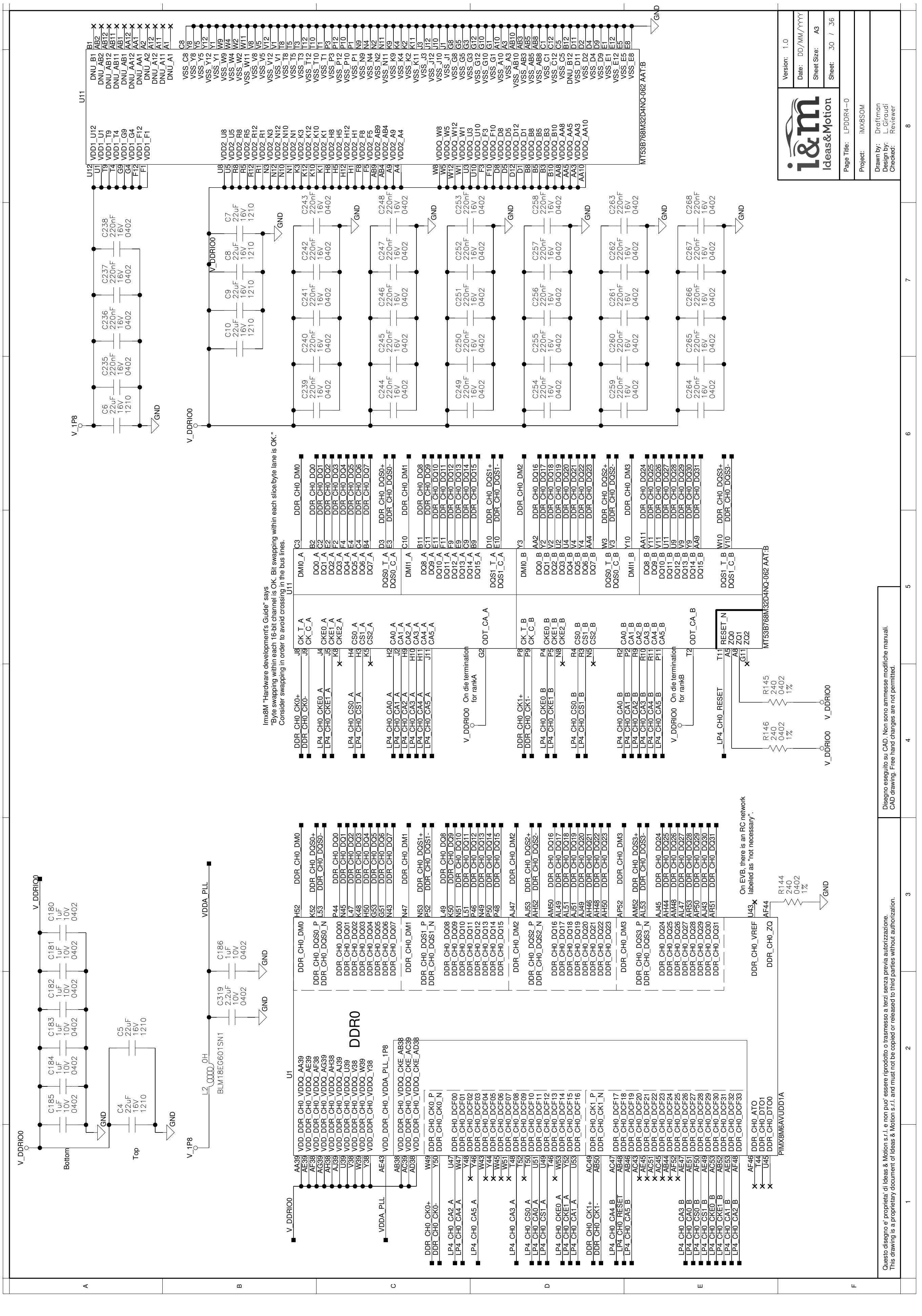

DDR0
U1
U11
PIMX8M6AVUDD1A
MT53B768M32D4NQ-062 AAT:B
Imx8M "Hardware development's Guide" says
"Byte swapping within each 16-bit channel is OK. Bit swapping within each slice/byte lane is OK."
Consider swapping in order to avoid crossing in the bus lines.
On die termination for rankA
On die termination for rankB
On EVB, there is an RC network labeled as "not necessary".
V_DDRIO0
V_1P8
VDDA_PLL
GND
Bottom
Top
BLM18EG601SN1
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
Ideas&Motion
Page Title: LPDDR4-0
Project: iMX8SOM
Drawn by: Draftman
Design by: L.Giroudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 30 / 36

Figure A.29: LPDDR4 RAM0

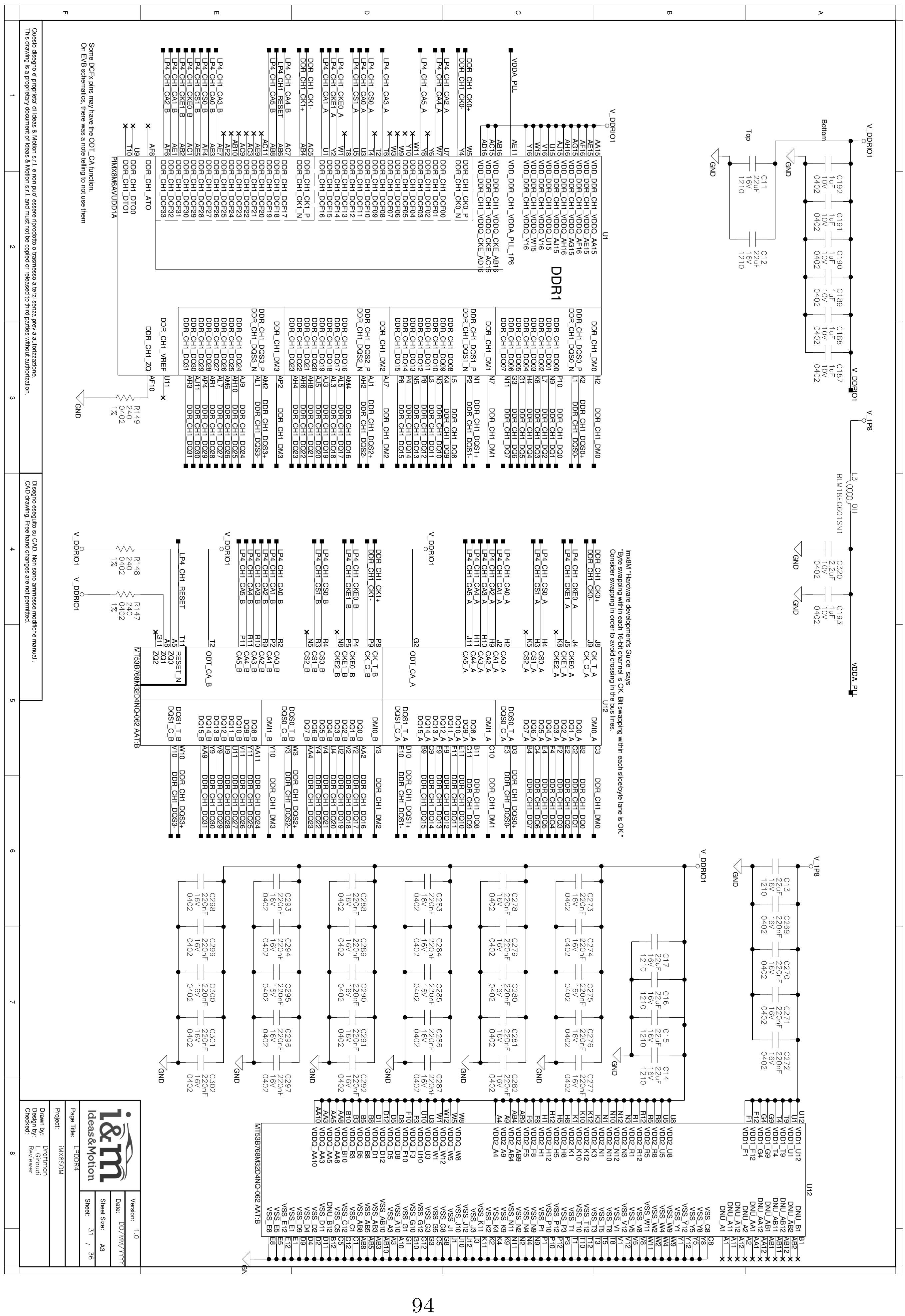
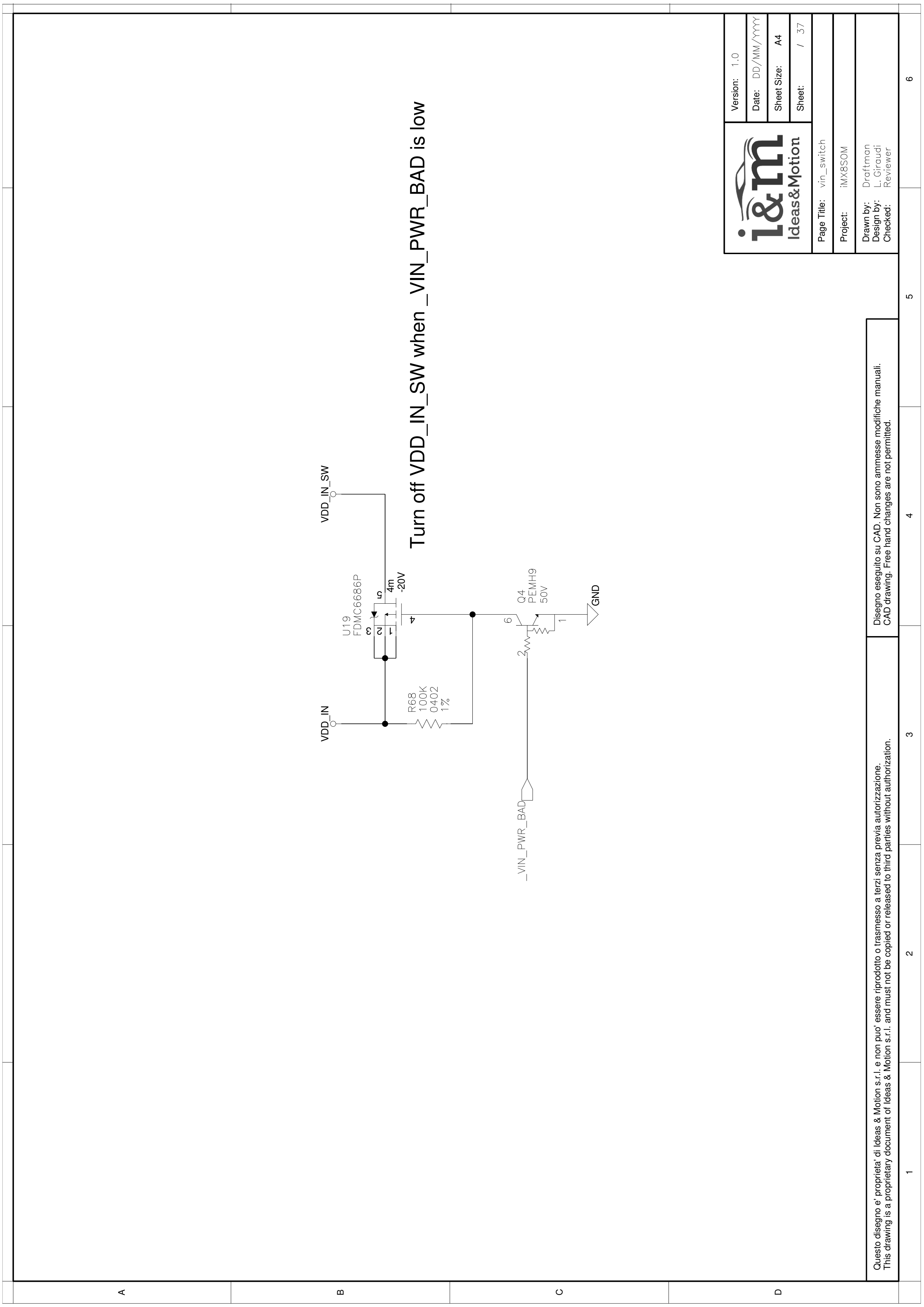

DDR1
U1
PIMX8M6AVUDD1A
U12
MT53B768M32D4NQ-062 AAT:B
V_DDRIO1
V_1P8
VDDA_PLL
L3 0000_0H BLM18EG601SN1
GND
Top
Bottom
R147 240 0402 1%
R148 240 0402 1%
R149 240 0402 1%
LP4_CH1_RESET
DDR_CH1_VREF
DDR_CH1_ZQ
DDR_CH1_ATO
DDR_CH1_DTO0
DDR_CH1_DTO1
InnxM "Hardware developments Guide" says "Byte swapping within each 16-bit channel is OK. Bit swapping within each slice/byte lane is OK." Consider swapping in order to avoid crossing in the bus lines.
Some DCFx pins may have the ODT_CA function. On EVB schematics, there was a note telling to not use them
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione. This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali. CAD drawing. Free hand changes are not permitted.
i&m Ideas&Motion
Page Title: LPDDR4
Project: IMX8SOM
Drawn by: Draftman
Design by: L. Giraudi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 31 / 36

Figure A.30: LPDDR4 RAM1



Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.

Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.

Figure A.31: VIN_PWR_BAD switch

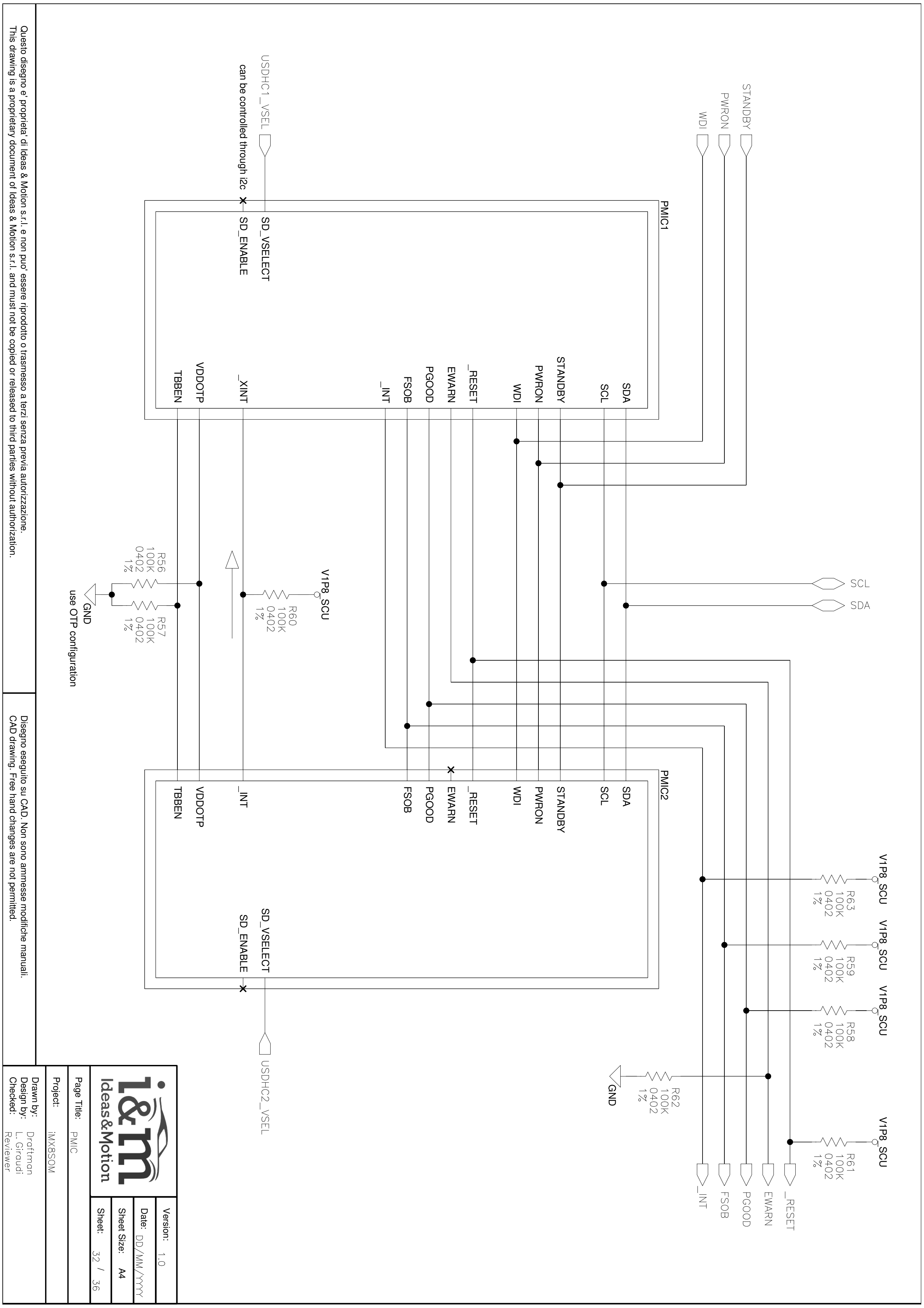


Figure A.32: PMIC Top Level

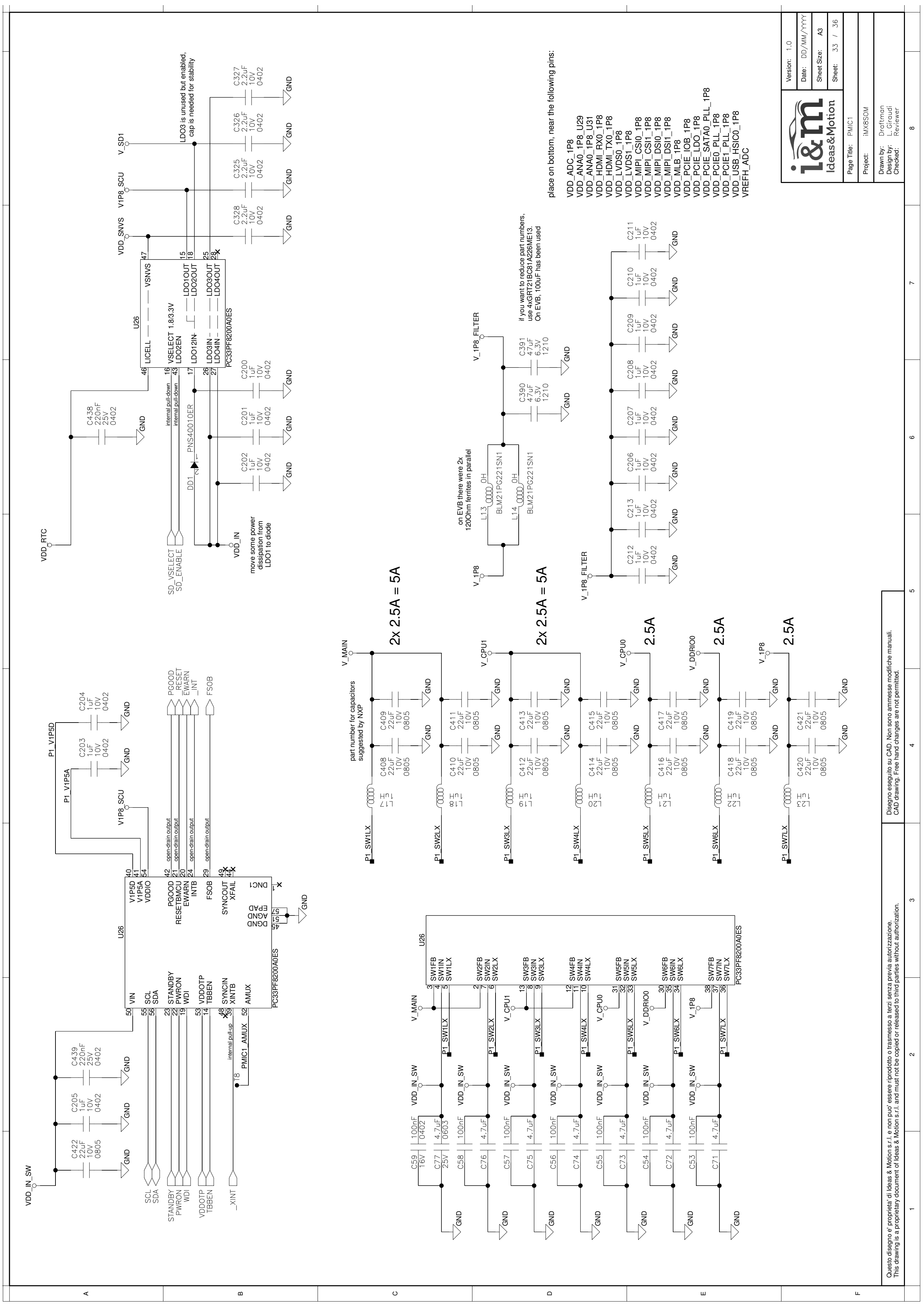

U26
PC33PF8200A0ES
LDO3 is unused but enabled, cap is needed for stability
move some power dissipation from LDO1 to diode
on EVB there were 2x 120Ohm ferrites in parallel
if you want to reduce part numbers, use 4xGRT21BC81A226ME13. On EVB, 100uF has been used
place on bottom, near the following pins:
VDD_ADC_1P8
VDD_ANA0_1P8_U29
VDD_ANA0_1P8_U31
VDD_HDMI_RX0_1P8
VDD_HDMI_TX0_1P8
VDD_LVDS0_1P8
VDD_LVDS1_1P8
VDD_MIPI_CSI0_1P8
VDD_MIPI_CSI1_1P8
VDD_MIPI_DSI0_1P8
VDD_MIPI_DSI1_1P8
VDD_MLB_1P8
VDD_PCIE_IOB_1P8
VDD_PCIE_LDO_1P8
VDD_PCIE_SATA0_PLL_1P8
VDD_PCIE0_PLL_1P8
VDD_PCIE1_PLL_1P8
VDD_USB_HSIC0_1P8
VREFH_ADC
part number for capacitors suggested by NXP
2x 2.5A = 5A
2x 2.5A = 5A
2.5A
2.5A
2.5A
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 33 / 36
i&m Ideas&Motion
Page Title: PMIC1
Project: iMX8SOM
Drawn by: Draftman
Design by: L.Giroudi
Checked: Reviewer
Disegno eseguito su CAD. Non sono ammesse modifiche manuali. CAD drawing. Free hand changes are not permitted.
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione. This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.

Figure A.33: PMIC1

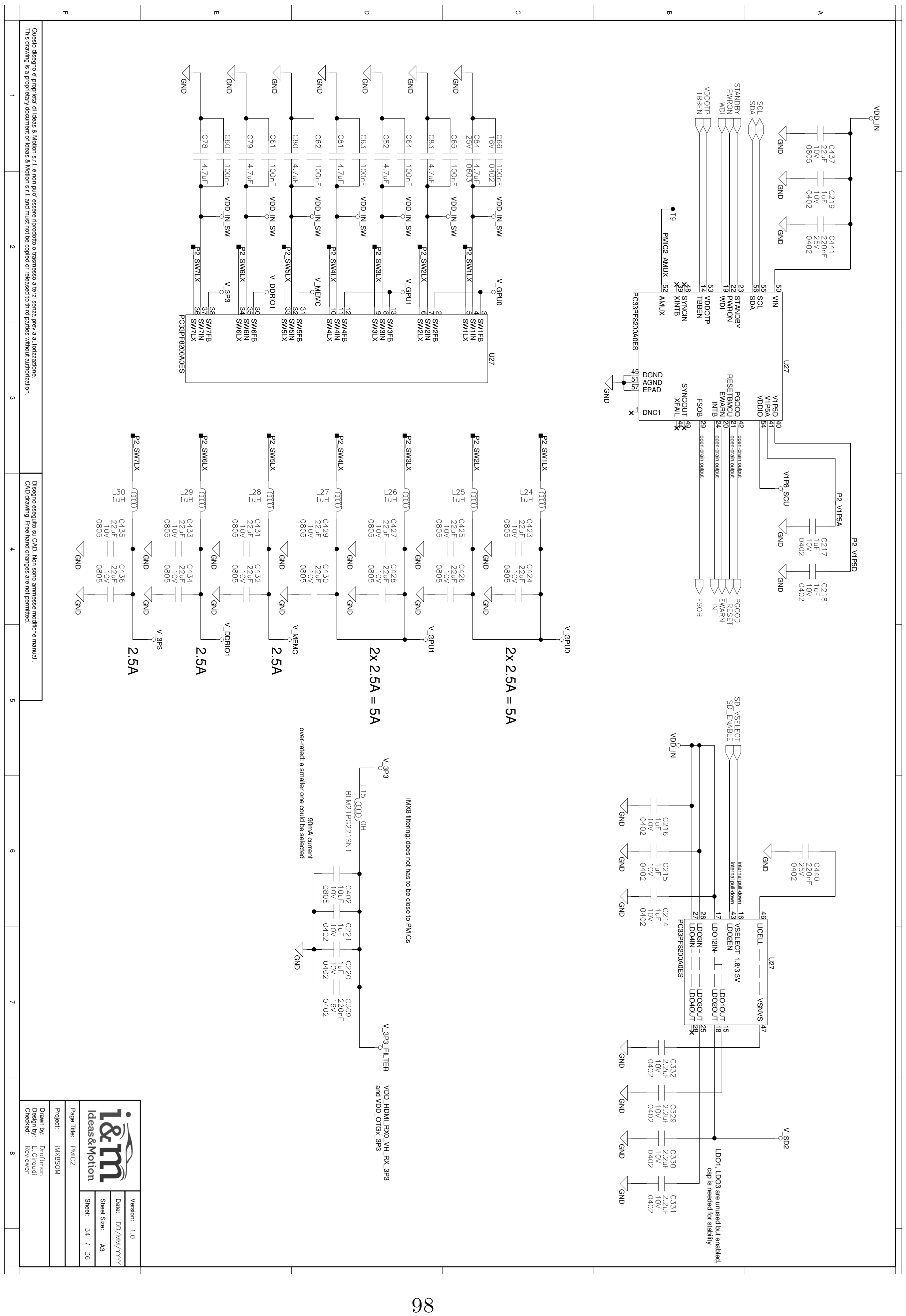

U27
PC33PF8200A0ES
VDD_IN
VDD_IN_SW
V_GPU0
V_GPU1
V_MEMC
V_DDRIO1
V_3P3
V_SD2
V_3P3_FILTER
VDD_HDMI_RX0_VH_RX_3P3 and VDD_OTGx_3P3
2x 2.5A = 5A
2x 2.5A = 5A
2.5A
2.5A
2.5A
iMX8 filtering: does not has to be close to PMICs
90mA current
over-rated: a smaller one could be selected
L15
BLM21PG221SN1
LDO1, LDO3 are unused but enabled, cap is needed for stability
PGOOD
RESET
EWARN
INT
FSOB
SCL
SDA
STANDBY
PWRON
WDI
VDDOTP
TBBEN
SD_VSELECT
SD_ENABLE
PMIC2_AMUX
P2_V1P5A
P2_V1P5D
V1P8_SCU
GND
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
i&m Ideas&Motion
Page Title: PMIC2
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Girardi
Checked: Reviewer
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A3
Sheet: 34 / 36

Figure A.34: PMIC2

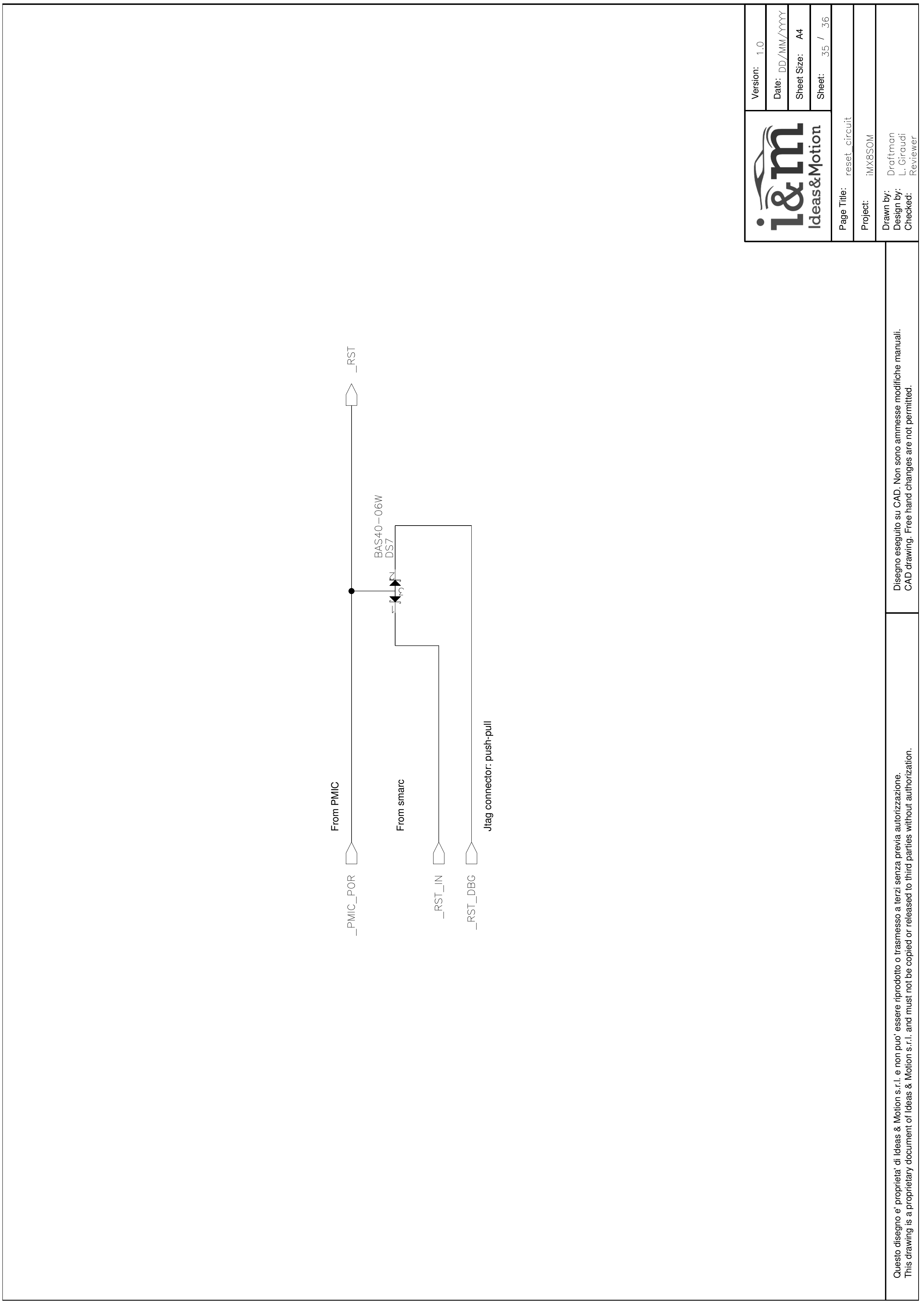

_RST
BAS40-06W
DS7
_PMIC_POR
From PMIC
_RST_IN
From smarc
_RST_DBG
Jtag connector: push-pull
Version: 1.0
Date: DD/MM/YYYY
Sheet Size: A4
Sheet: 35 / 36
i&m Ideas&Motion
Page Title: reset_circuit
Project: iMX8SOM
Drawn by: Draftman
Design by: L. Giraudi
Checked: Reviewer
Disegno eseguito su CAD. Non sono ammesse modifiche manuali.
CAD drawing. Free hand changes are not permitted.
Questo disegno e' proprieta' di Ideas & Motion s.r.l. e non puo' essere riprodotto o trasmesso a terzi senza previa autorizzazione.
This drawing is a proprietary document of Ideas & Motion s.r.l. and must not be copied or released to third parties without authorization.

Figure A.35: Reset circuit

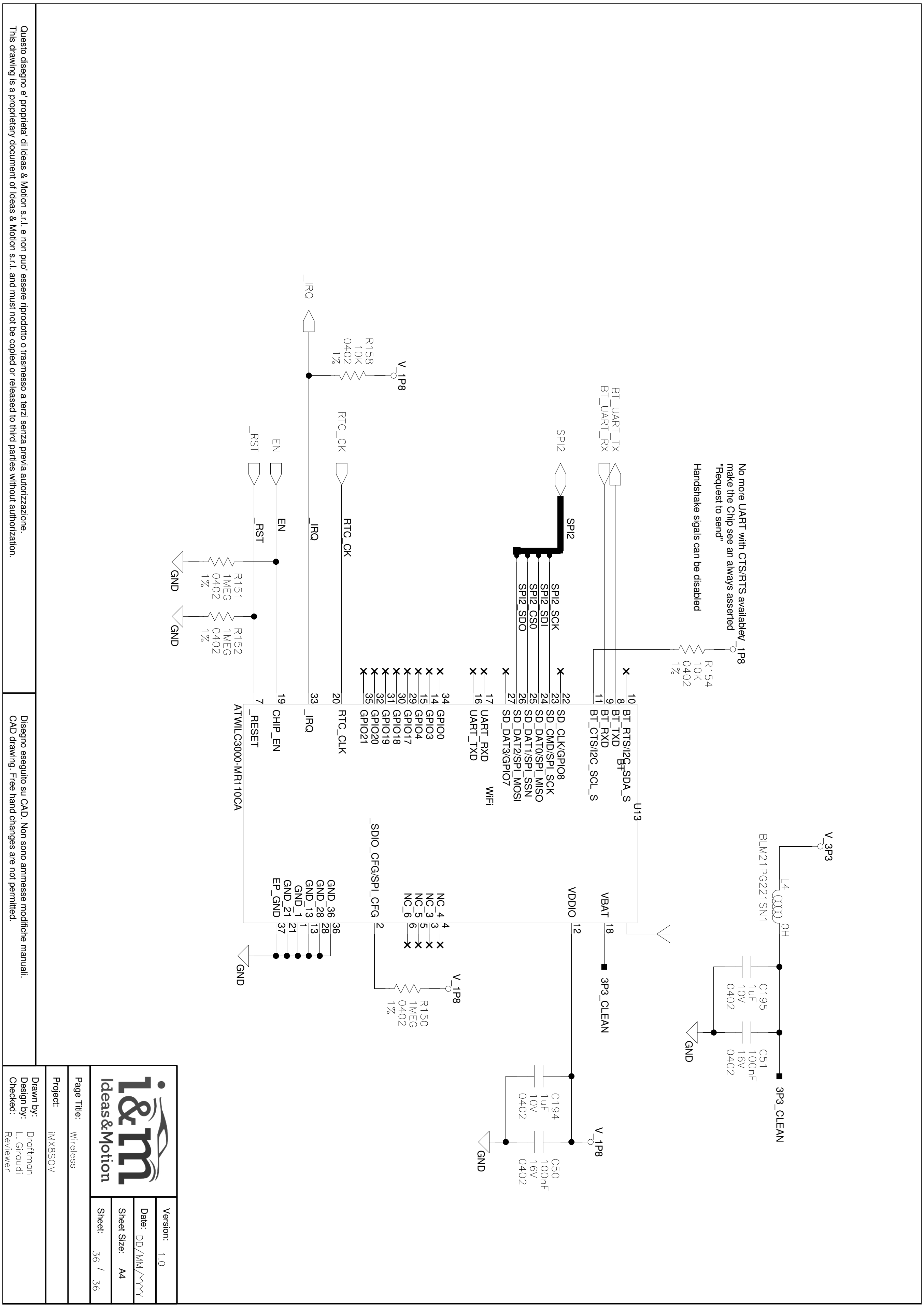


Figure A.36: Wifi and Bluetooth Adapter

Bibliography

[1] MIPI Alliance. *MIPI Alliance Specification for Camera Serial Interface 2.* 2009.

[2] MIPI Alliance, ed. *Specification for D-PHY*. Version 1.2. 2014.

[3] Marvell Broadcom Hp. *Reduced Gigabit Media Independent Interface (RGMII).* Version 1.3. 2000.

[4] Marc Greenberg. *LPDDR3 and LPDDR4: How Low-Power DRAM Can Be Used in High Bandwidth Applications*. Ed. by JEDEC. 2013. URL: https://www.jedec.org/sites/default/files/M_Greenberg_Mobile%20Forum_May_%202013_Final.pdf.

[5] Stephen H. Hall and Howard L. Heck. *Advanced Signal Integrity for High-Speed Digital Designs*. John Wiley & Sons, Inc., Mar. 2009. DOI: 10.1002/9780470423899.

[6] LLC HDMI Licensing. *High Definition Media Interface Specification v 1.4.* 2009.

[7] IPC. *IPC-7351B. Generic Requirements for Surface Mount Design and Land Pattern Standard.* 2010.

[8] JEDEC, ed. *Embedded Multi-Media Card (e•MMC) Electrical Standard (5.0).* 2013.

[9] JEDEC. *Low Power Double Data Rate 4*. 2014.

[10] Silicon Labs. *PCI E XPRESS 3.1 J ITTER R EQUIREMENTS*. 2015. URL: https://www.silabs.com/documents/public/application-notes/AN562.pdf.

[11] NXP. *Flip Chip Plastic Ball Grid Array, Application Note*. 2012. URL: https://www.nxp.com/docs/en/package-information/FC-PBGAPRES.pdf.

[12] NXP. *i.MX 8QuadMax Applications Processor Datasheet rev D.* 2016.

[13] NXP. *i.MX 8QuadMax Applications Processor Reference Manual.* 2016.

[14] NXP. *i.MX8 Applications Processors Family Fact Sheet*. Sept. 27, 2016. URL: https://www.nxp.com/docs/en/fact-sheet/IMX8FAMFS.pdf.

[15] NXP. *Level shifting techniques in I2C-bus design*. 2007.

[16] NXP. *PF8x00 Datasheet Rev. 2.0*. 2018.

[17] PCI-SIG. *PCI Express Base Specification Revision 3.0*. 2010.

[18] Donovan Porter. *100BASE-T1 Ethernet: the evolution of automotive networking*. Ed. by Texas Instrument. 2018.

[19] SAE. *J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2014.

[20] NXP Semiconductor. *MIPI–CSI2 Peripheral on i.MX6 MPUs*. 2016.

[21] ON Semiconductor. *A System Designer's Guide for Building a PCIe ® Clock Tree while Addressing Timing Challenges*. 2015. URL: `http://www.onsemi.com/pub/Collateral/AND9202-D.PDF`.

[22] SGET. *Smart Mobility ARChitecture, Design Guide. SMARC Design Guide 2.0*. Mar. 23, 2017. URL: `https://www.sget.org/standards/smarc.html`.

[23] SGET. *Smart Mobility ARChitecture, Hardware Specification*. Version 2.0. June 2, 2016. URL: `https://www.sget.org/fileadmin/user_upload/SMARC_Hardware_Specification_V200_Errata.zip`.

[24] USB-IF. *Universal Serial Bus Specification*. Version 2.0. 2000.