DMMC-STAMP

User Manual

Version 3.8

MicroTCA TECHNOLOGY LAB

DESY.

Table of Contents

Figures

Tables

Revision History

Revision	Date	Description of Change
1.0	09/09/2019	Initial Release
1.1	14/11/2019	Add block diagrams, pin descriptions and clarifications
2.0	18/05/2021	Various improvements
2.1	26/06/2021	Minor corrections based on review
2.2	19/10/2021	Add footprint, SDK and design template information
2.3	03/08/2022	Correct SDK information table, update dimension drawing
3.0	28.09.2022	Refurbish the manual, add standard firmware information
3.1	14.04.2023	Add SWD pin numbers; improvements of pin-out table (B3, I3, C3, N2, N3, N4, D3); Add compact pinout table
3.1.1	23.12.2023	Modify imprint, add copyright, switch to „Only Office“.
3.2	17.09.2024	Add information about I2C busses and update block diagram.
3.2.1	16.01.2025	Minor fonts touch-up
3.3	22.04.2025	Correct CPLD_IOx table, add stencil information, minor corrections
3.4	22.09.2025	Add info anbout outlet temp sensor and its I2C address. Remove one stencil recommendation (manufacturer feedback). Add Information on DMMC- STAMP Mailbox. Update imprint about AI tools usage.
3.5	22.09.2025	Add information about integrated pull-up and pull-down resistors. Add „References“ chapter.
3.6	5.10.2025	Add information about Interlock implementation; update DMMC-BoB image and text
3.7	10.11.2025	Document Out-of-Crate Mode, add SDK-related note on FMC pins , document CPLD pins and add abbreviated schematifcs
3.7.1	27.11.2025	Minor touch-up of graphics and images without information update.
3.8	11.12.2025	Minor text touch-up, add power routing information (general and RTM), add mailbox address, and power supply details (general and RTM).

1. Introduction

The DESY Module Management Controller System on Module (DMMC-STAMP SoM) provides a full management solution for operating the targeted Advanced Mezzanine Card (AMC) in a MicroTCA-based ecosystem. The module itself is a small, stamp-sized (25.5 × 29.5 × 2.3 mm) component that can be mounted on the top or bottom side of any AMC PCB. In addition to the management features required by the MicroTCA specification, the DMMC-STAMP can operate the AMC power system, manage up to two FPGAs/SoCs, and host up to two slots for FPGA Mezzanine Cards (FMCs) and one Rear Transition Module (RTM).

Figure 1: DMMC-STAMP System on a Module (SoM)

1.1 Block Diagram

The DMMC-STAMP System on a Module (SoM) block diagram is shown in Figure 2.

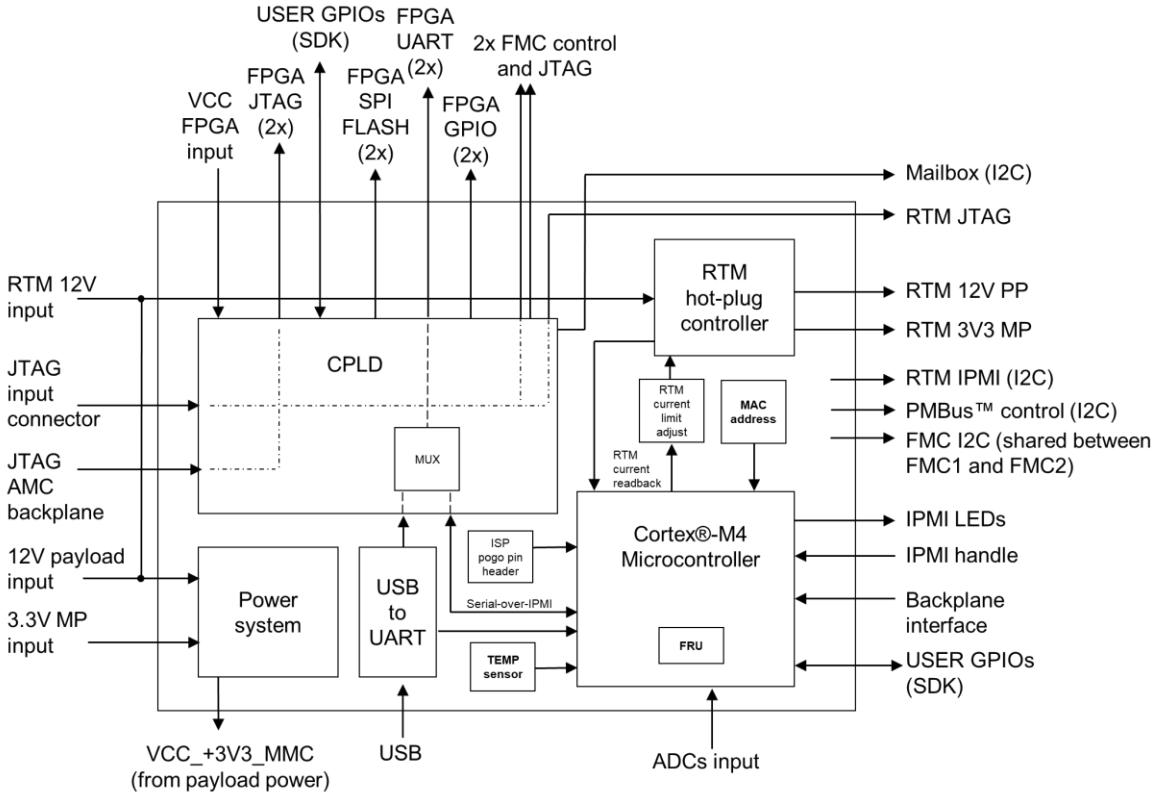


Figure 2: DMMC-STAMP Block Diagram

1.2 Module Features

The DMMC-STAMP SoM features are listed here:

- Full IPMI handling (MCH communication, LEDs, AMC/RTM power, FRU read/write)
- On-board 12-bit temperature sensor
- On-board 1kBit EEPROM (memory for non-volatile settings)
- Hard-coded 48-bit unique ID that can be used as MAC Address by FPGA (SDK required)
- 8 analog inputs (0-1V) for voltage measurement; management power (3.3V MP) and payload power (12V PP) are measured on the DMMC-STAMP by using additional ADC channels
- Board failure handling (over-current and over-temperature detection)
- Support of up to two FPGA/SoC interfaces (custom interface voltage: 1.2V - 3.3V), including IPMI firmware update of associated SPI FLASH memories and 8 GPIO pins per interface
- Control of up to two FMC modules (optional)
- Re-configurable JTAG chain management
- RTM management including hot-swap controller with adjustable payload current limit
- 2-channel USB virtual COM port for MMC and FPGAs/SoCs and remote serial-over-IPMI mirroring
- Includes pre-programmed default MMC firmware that allows the user design to run inside MicroTCA
- A Software Development Kit (SDK) is optionally available to further customize DMMC-STAMP behavior, for example to support custom sensors or enable advanced interaction between the payload electronics (e.g., SoC) and MicroTCA management.

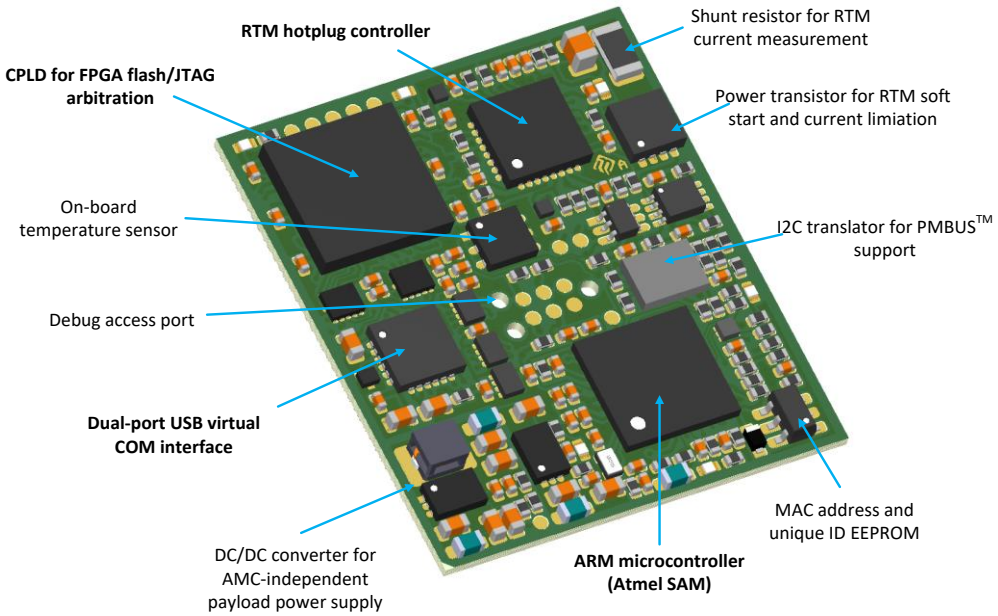


Figure 3: DMMC-STAMP (Revision A) components

1.3 Operating Conditions

The DMMC-STAMP SoM operating conditions are listed here:

Environmental	
Operating Temperature	0 - 80°C
Minimum Airflow	None
Electrical	
Input Power (Payload)	12V nominal, 10V - 14V range
Input Power (Management)	3.3V +/- 10%
Output Voltage (VCC_+3V3_MMC)	3.3V +/- 10%
Maximum Output Current (VCC_+3V3_MMC)	250mA
Maximum voltage on all Digital input Pins	VCC +0.3V
Maximum voltage on FPGA supply pins	3.3V
Maximum voltage on all analog input pins	1.0V
Maximum sink/source current on all MCU Pins	3.2 mA
Maximum sink/source current on red/green/blue LED	24mA
Maximum output current on all CPLD Pins	8 mA
RTM output current limit VCC_RTM+3V3_MP	65mA (nominal)
RTM output current limit range VCC_RTM+12V (user setting)	1.62A (19.4W) - 4.60A (55W)

Table 1: DMMC-STAMP operating conditions

2. Management in MicroTCA®

MicroTCA® is a modular and open specification for electronics that provides advanced management features and modern communication interfaces. A characteristic feature of MicroTCA® is the out-of-band management, which significantly simplifies the maintenance and the collection of status information in large installations.

Important building blocks in a MicroTCA® based environment are the Advanced Mezzanine Cards® (AMC), as defined in the PICMG AMC.0 specification. AMCs usually contain FPGAs, SoCs, DSPs or other processing components to perform a desired functionality. Apart from this main “payload” and according to the specifications, a Module Management Controller (MMC) is required, to provide the AMC board management capabilities. Shown in Figure 4 (from PICMG AMC.0, copyright PICMG, used with permission) are the mandatory management components that have to be present on every AMC module.

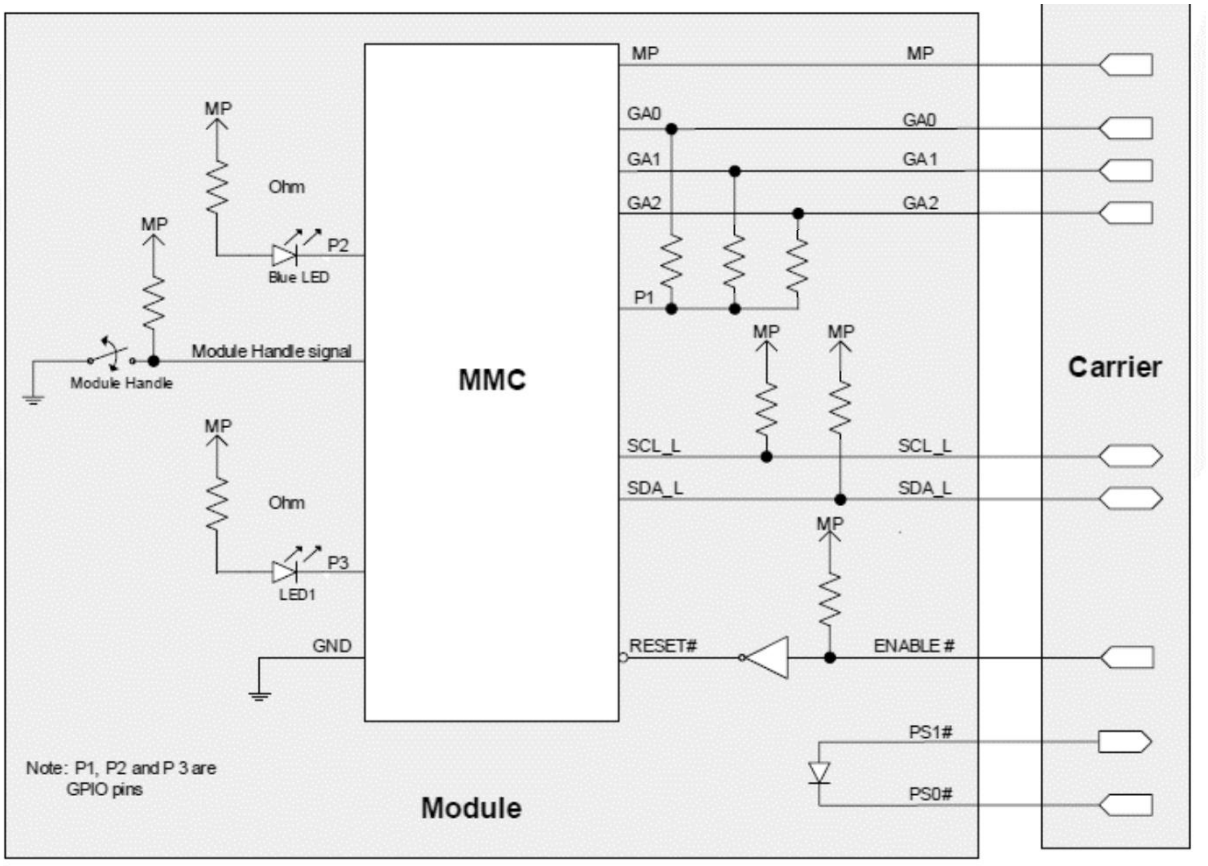


Figure 4: AMC module management hardware components

All hardware components shown above (except of the LEDs and Module Handle) are integrated on the DMMC-STAMP SoM. As a reference for a custom AMC hardware development based on the DMMC-STAMP SoM the DESY MicroTCA Technology Lab provides an AMC design template for the Altium Designer® software. For more details see 6.1.

3. Module Component Description

This chapter provides a detailed functional description of the DMMC-STAMP SoM components and features. All components described here are already shown in Figure 2 and Figure 3.

3.1 Microcontroller

The DMMC-STAMP SoM is populated with an ATSAM4LS8C Cortex-M4 microcontroller. Configured with the pre-programmed DMMC-STAMP firmware it provides the following features:

- Low-power system, running at 36 MHz and 3.3V management power
- Full IPMI handling
- Fail-safe in-system update mechanism

  The integrated bootloader allows in-system firmware update (HPM) initiated by the user. The bootloader will take over all IPMI communication until a valid application has been programmed and loaded.
- Embedded FRU storage in internal FLASH memory
- 8 ADC channels with 12 bits resolution (0 - 1V input range) used for carrier power management

3.2 CPLD

The DMMC-STAMP SoM is populated with a Lattice MachXO2 CPLD that handles the payload FPGA related functionality. Configured with the pre-programmed DMMC-STAMP firmware it provides the following features:

- Full isolation between management and payload power domains
- Complete FPGA/SoC handling (PROGRAM, INIT, DONE, …)
- Two FPGA/SoC power domains with two user-defined voltage levels (i.e. fully independent power rails for the two FPGA/SoC interfaces)
- All pins of the FPGAs/SoCs can run at an arbitrary voltage (e.g. 1,2V, 1.8V, 3.3V), which is injected from the outside (CPLD bank voltage)
- UART interface to the user FPGAs/SoCs
- Control of the FPGA/SoC SPI FLASH memories (including HPM update)
- Control of the JTAG chain (including JTAG arbitration)

During the SoM power-up sequence the CPLD gets in-system programmed by the pre-programmed DMMC-STAMP firmware. The CPLD configuration file is part of every MMC firmware update file and the CPLD update itself is fully transparent to the user. The DMMC-STAMP firmware automatically ensures that the CPLD configuration matches the firmware version and re-programs it, if necessary.

3.3 USB-to-UART Bridge

The DMMC-STAMP SoM is populated with a Silicon Labs CP2105 dual channel USB-to-UART bridge. This interface has following properties:

- Channel 1: Command Line Interface (CLI) debug access to the Cortex-M4 microcontroller
- Channel 2: Command Line Interface (CLI) debug access to the USER FPGAs/SoCs

For managing up to two FPGAs/SoCs at USB-to-UART channel 2 the interfaces get multiplexed inside the DMMC-STAMP CPLD. In the pre-programmed DMMC-STAMP firmware the FPGA/SoC selection can be changed by using the USB CLI interface to the microcontroller (channel 1).

3.4 RTM Hot-Plug Controller

The RTM power is handled by a dedicated controller on the DMMC-STAMP. The features of this chip are listed here:

- RTM power handling (12V power is activated in M4 state)
- RTM inrush current limiting
- Customizable maximum power limit (MCU-controlled I2C potentiometer)
- RTM over-current protection

3.5 Power System

The Power System section deploys all required voltages to the DMMC-STAMP components. Furthermore, there is a 3.3V power output to the target AMC which gets available as soon as the payload power on the DMMC-STAMP is activated. Usually this 3.3V rail will be the first power domain (derived from payload power) that gets available on the target AMC. The 3.3V power domain can be used to run user-defined management electronics (e.g. LDO bias voltage or PMBUS controller power).

Additional information:

- The DMMC-STAMP MCU runs on the 3.3V management power (no other voltages needed).
- The DMMC-STAMP pre-programmed firmware will initiate all steps to the activate the 12V payload power.
- All additional voltages (derived from payload power) which are needed for operating the DMMC-STAMP are generated on-board (e.g. the CPLD core voltage is generated on-board).
- The DMMC-STAMP CPLD runs in the payload power domain.
- Voltages needed for operating FPGAs/SoCs on the target AMC are provided from external power input pins.
- VCC+3V3: 3.3V is generated from payload power by a dedicated DC/DC converter followed by an LDO stage. The user can draw up to 250mA current from this rail.

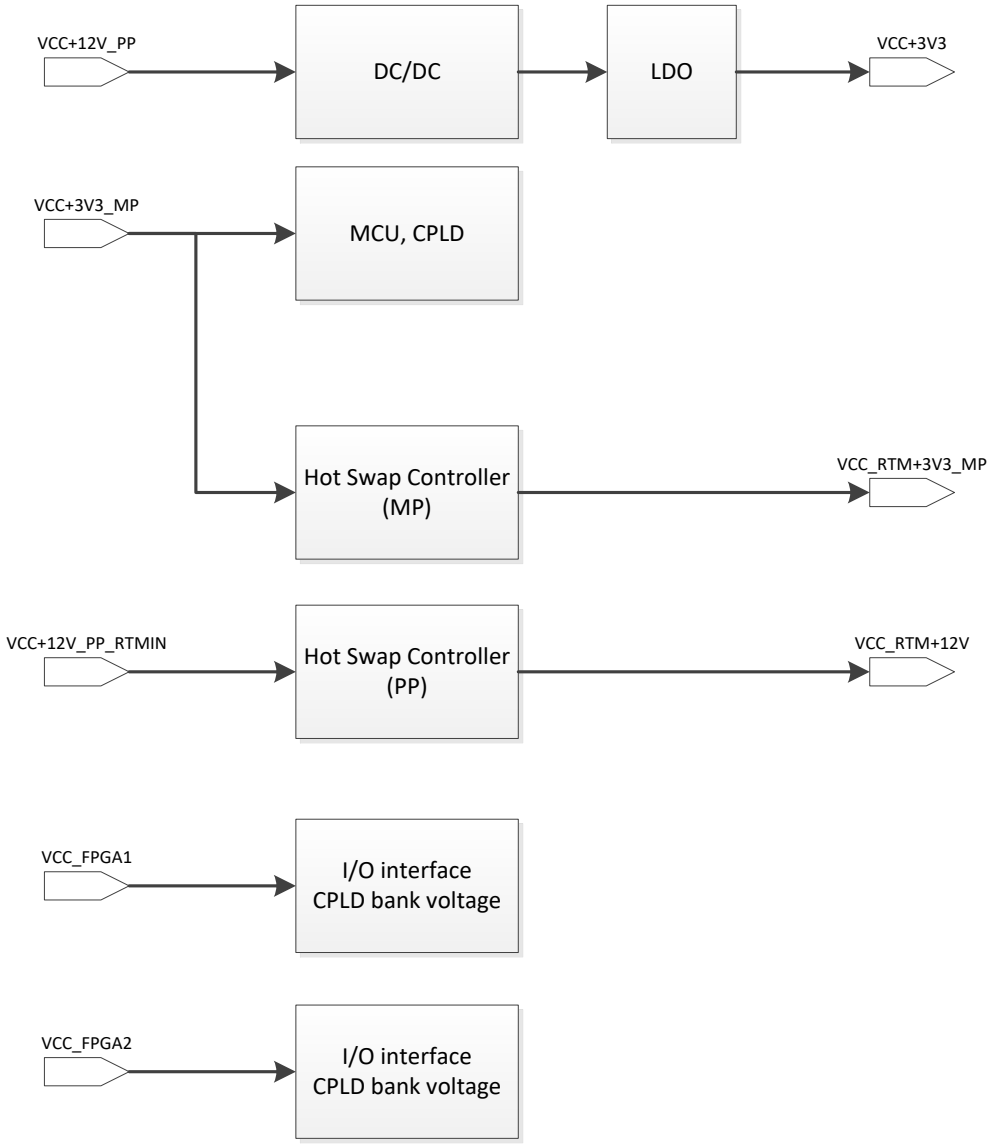


Figure 5: DMMC-STAMP power tree

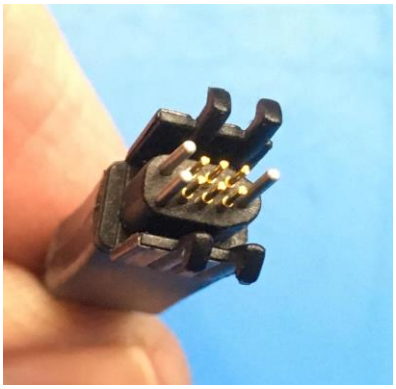
If AMC boards are to be equipped with a DMMC-STAMP whose firmware is intended to be modified using the SDK, the custom DMMC-STAMP firmware is typically designed to manage the entire power tree. This can be implemented in a simple setup using an MCU GPIO pin as a global power enable/disable to initiate the power-on/ power-off sequence, or in a more advanced configuration using a PMBus™-controlled power tree. In all cases, both power activation and deactivation should be controlled through the DMMC-STAMP to ensure a coordinated and controlled shutdown while the board is still powered. If the power system is not managed by the DMMC-STAMP, power-down will usually occur abruptly, and there may not be enough time to safely shut down all power rails, potentially violating the sequencing requirements of the user's payload components.

3.6 Debug Access Port

Integrated into the target AMC and during operation, the DMMC-STAMP gets programmed and updated by using the Intelligent Platform Management Interface (IPMI) provided by the MicroTCA® infrastructure. For debug and development purposes, the DMMC-STAMP PCB is equipped with landing pads for a Tag-Connect™ cable which provide access to the Serial Wire Debug (SWD) interface of the Microcontroller Unit (MCU).

Debug connector Pinout:

5: GND	3: RST_B	1: VCC+3V3_MP
6: N.C.	4: SWD_CLK	2: SWD_DIO



The picture shows the TC2030 with „legs“. They need to be removed to mate to the DMMC-STAMP

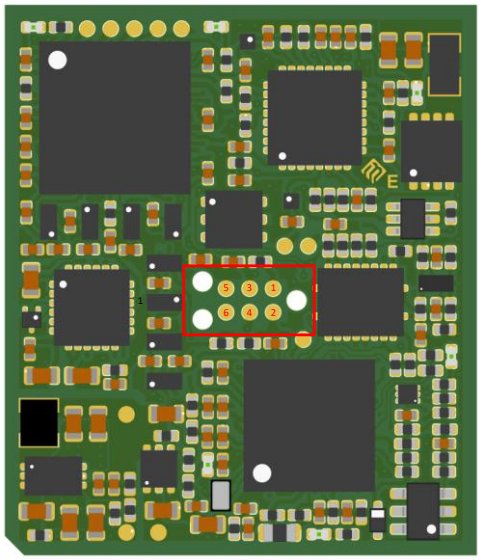


Figure 6: MCU Debug (Tag-Connect pads)

Additional information:

- The DMMC-STAMP SoM is designed to match with the TC2030-CTX-NL connector given here:
  https://www.tag-connect.com/product/tc2030-ctx-nl-6-pin-no-legs-cable-with-10-pin-micro-connector-for-cortex-processors
- The connector needs retainer clips:
  https://www.tag-connect.com/product/tc2030-retaining-clip-board-3-pack
- Alternative source from Digi-Key:
  https://www.digikey.de/product-detail/de/tag-connect-llc/TC2030-CTX/TC2030-CTX-ND/5023324
  https://www.digikey.de/product-detail/de/tag-connect-llc/TC2030-CLIP/TC2030-CLIP-ND/2605371

  These links point to the “legged” version of the cable. The legs can be simply removed to manually create no-legged version of the cable.
- The one end of the cable needs to be plugged into the ATMEL-ICE ARM programmer. The other end of the cable needs to connect to the SWD pads of the DMMC-STAMP.

3.7 On-Board LEDs

Described in Figure 7 are the DMMC-STAMP on-board LEDs:

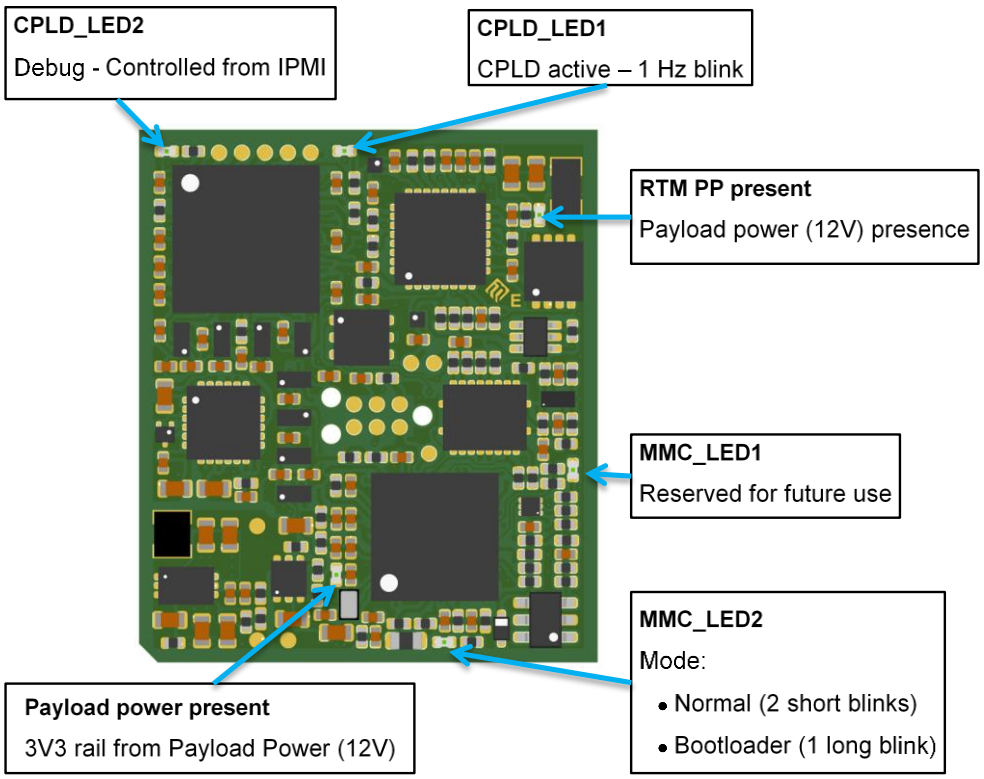


Figure 7: DMMC-STAMP On-Board LEDs

4. LGA Footprint

This chapter describes the Land Grid Array (LGA) footprint of the DMMC-STAMP. All drawings below show the pin assignment as seen from the top side of the module (view through the PCB).

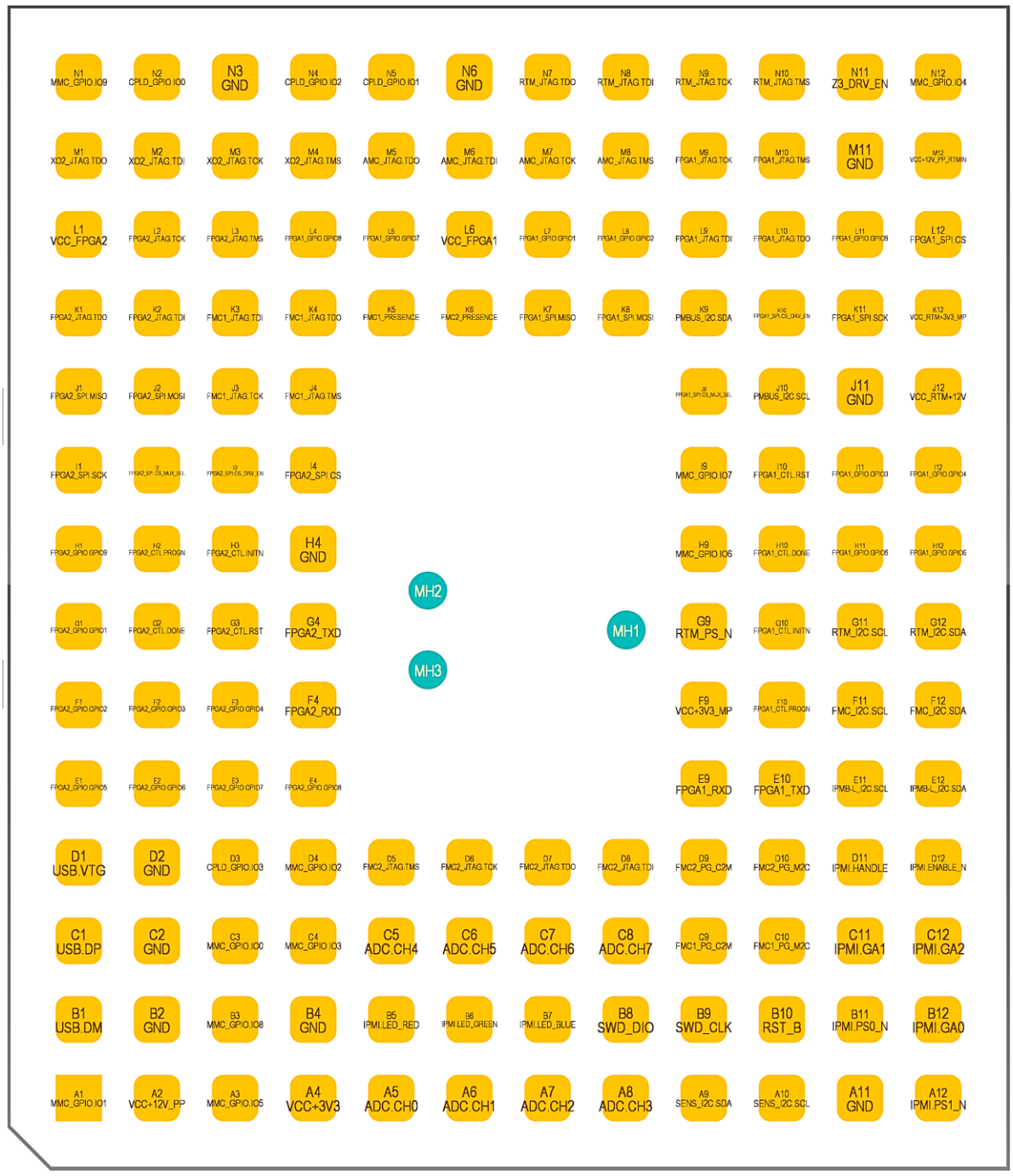


Figure 8: DMMC-STAMP Pad Naming

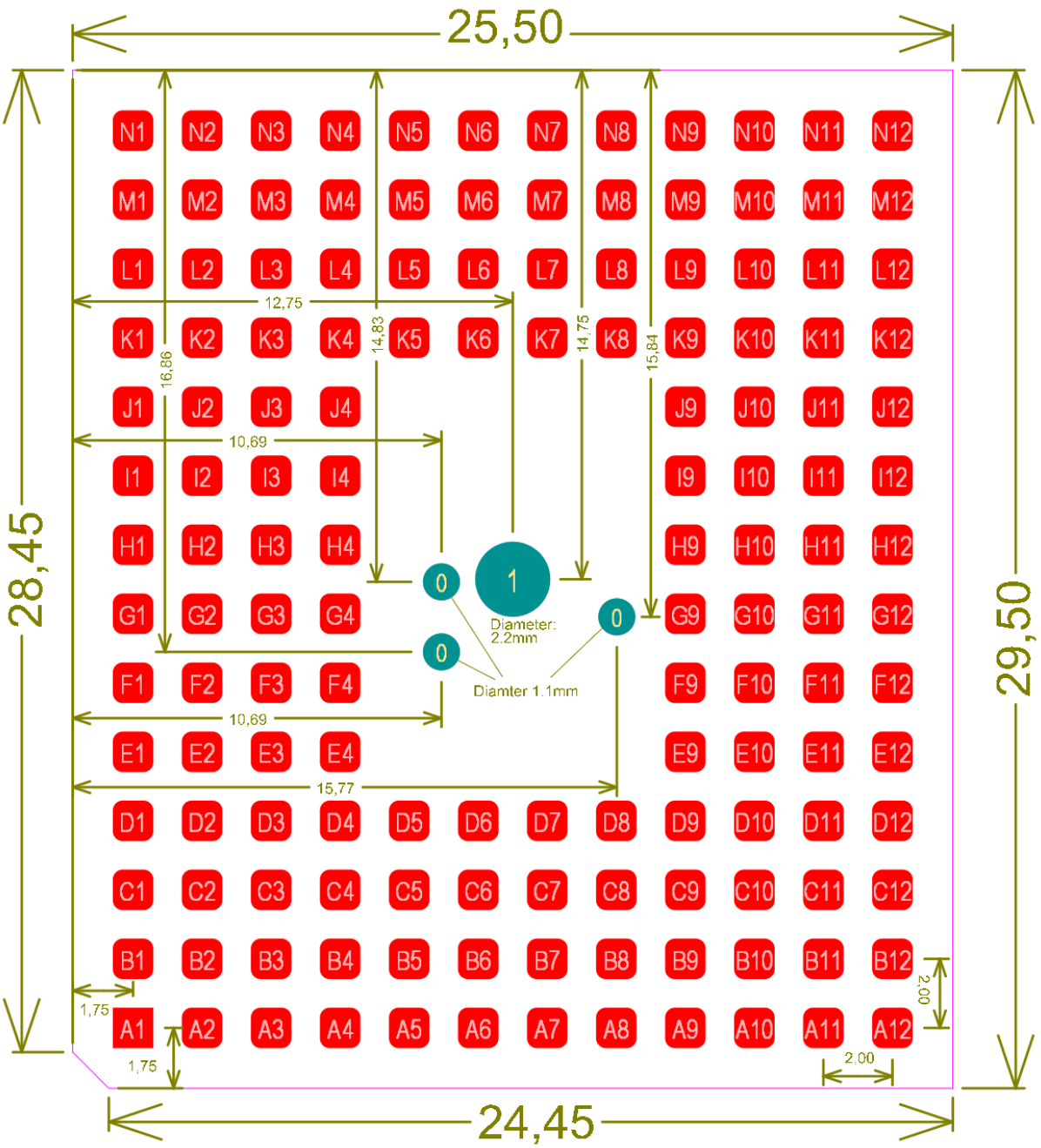


Figure 9: DMMC-STAMP Dimensions

The hole marked with “1” is used for checking the alignment of the DMMC-STAMP SoM. The LGA contains a cross in the center, which shall be located in the middle of the hole within a correct assembly. Holes marked with “0” are the guide holes used by the Tag-Connect™ connector which can be plugged to the DMMC-STAMP SoM for debugging and development purposes. If used, the alignment pins will protrude the board and the retaining clip can be mounted below the user board.

The outer dimensions of the DMMC-STAMP SoM are given with a tolerance of +/- 0,5 mm.

4.1 Layout Information

- A component clearance of 2 mm around the DMMC-STAMP SoM is recommended.
- It is strongly recommended to add the Tag-Connect™ alignment holes.
- Adding the center hole for checking the alignment is optional.
- All pads shall have a 2 mm pitch.
- The pad geometry (except pad A1) shall be 1.2 mm x 1.2 mm with rounded corners.
- The pad A1 geometry shall be 1.2 mm x 1.2 mm with a rectangular shape.
- Solder paste coverage of the complete pad is recommended.
- Non-solder mask defined pads with 0.075 mm mask expansion is recommended.

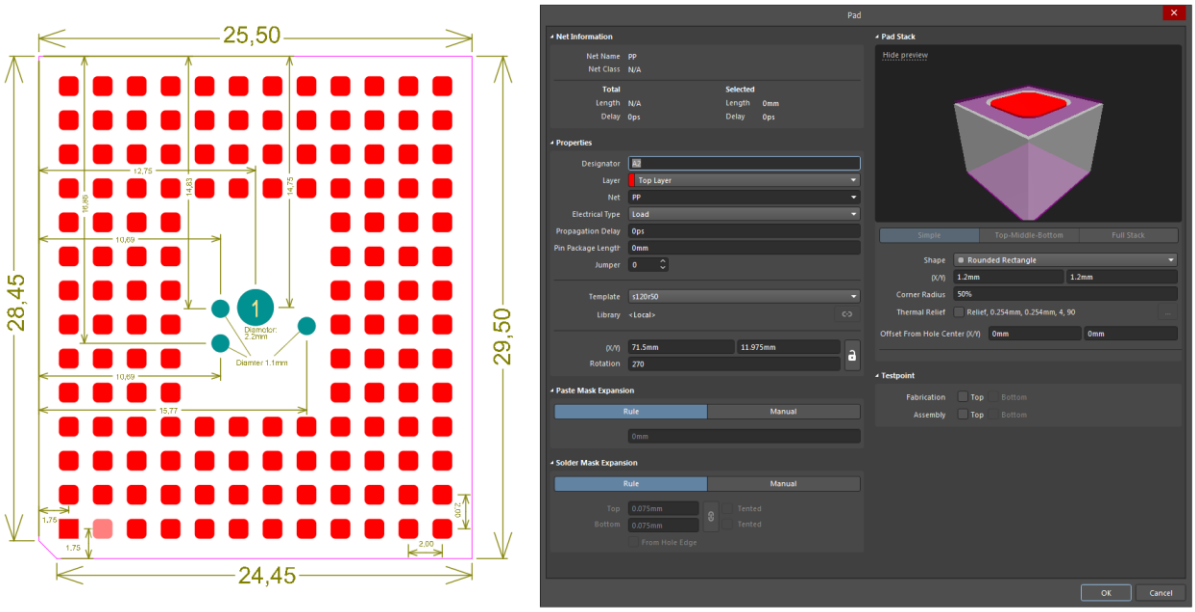


Figure 10: DMMC-STAMP Pad Geometry (except pad A1)

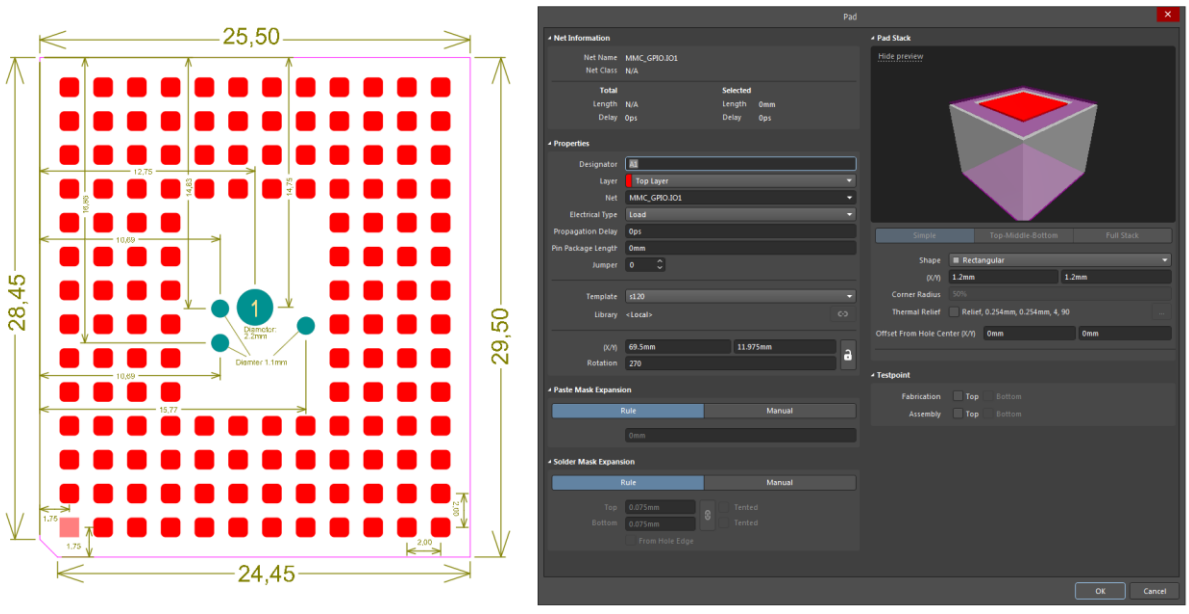


Figure 11: DMMC-STAMP Pad A1 Geometry

4.2 Stencil information

On AMC boards, the MMC stamp is one of the less critical parts in terms of stencil requirements. Usually, other parts define the stencil thickness and the stamp is used with the same stencil thickness.

DESY is aware of two different stencil designs used by its manufacturers and partners:

1. Standard 100µm stencil: A 100µm thick stencil with reduced solder mask opening is used - solder mask opening is reduced to 90% to have an effective opening of 1.08x1.08mm with a radius of 270µm.
2. 200um/100um step stencil: A milled paste stencil at 0.200µm thickness is used. The LGA stamp module is soldered at full 200µm thickness and the rest of the stencil is milled down to 100µm thickness.

These are two three examples that have been used successfully. Other stencil types and thicknesses may also give good results.

5. Module Interface Description

This chapter provides a detailed description of the DMMC-STAMP SoM interfaces. CPLD pins (customizable via DMMC-STAMP SDK) have green background.

5.1 MicroTCA® and AMC Pins

The MTCA and the AMC interfaces provide all signals that are to be connected to the backplane and to the IPMI-related user interface (LEDs, handle). All items listed below are mandatory (except of the JTAG signals). The JTAG interface provides access to the DMMC-STAMP SoM CPLD. Equipped with the appropriate configuration logic the CPLD provides access to the complete user JTAG chain of the target AMC which can include FPGAs/SoCs or devices on the RTM and FMC modules.

Shown in Figure 12 and Figure 13 are exceptions from the schematics of AMC template (see 6.1) which provides more detailed information on how the mandatory MicroTCA® components should be used.

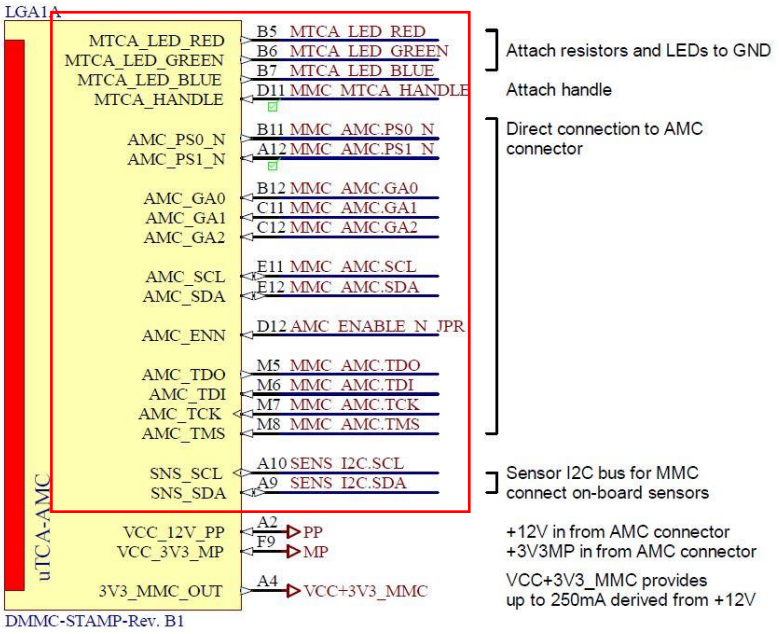


Figure 12: AMC connector pins

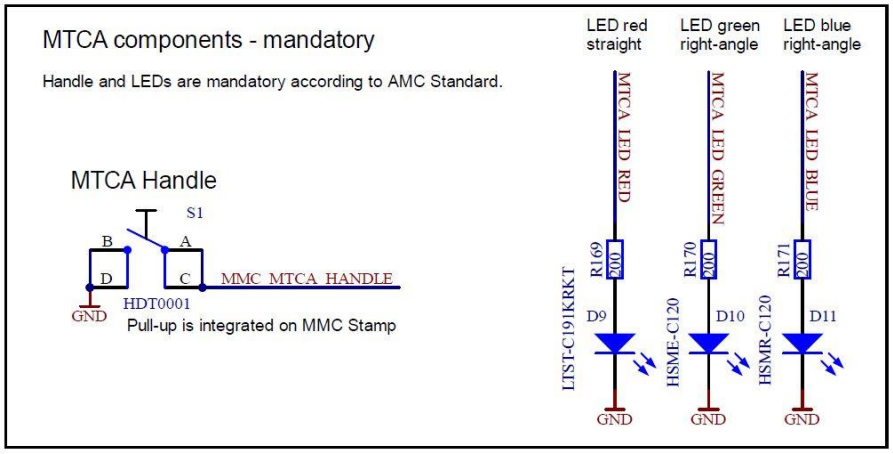


Figure 13: MicroTCA® mandatory components

Pin Name	Location	Direction	Description
AMC_PS1_N	A12	IN	Module presence detect, as defined in chapter “3.2.2 PS0# and PS1#” in the PICMG AMC.0 specification. The DMMC-STAMP SoM contains a Schottky diode between PS1# and PS#0.
AMC_PS0_N	B11	OUT	Module presence detect, as defined in chapter “3.2.2 PS0# and PS1#” in the PICMG AMC.0 specification.
AMC_EN_N	D12	IN	MMC enable, as defined in chapter “3.2.3 ENABLE#” in the PICMG AMC.0 specification. This pin serves as a microcontroller reset and has to be pulled low to activate the microcontroller (this is normally done by MCH after presence detection).
AMC_GA2	C12	IN	Geographic address (bit 2), as defined in chapter “3.2.1 Geographic Address [2..0] (GA[2..0])” in the PICMG AMC.0 specification. On DMMC-STAMP, there is a 3.3kOhm pull-up resistor on this pin.
AMC_GA1	C11	IN	Geographic address (bit 1), as defined in chapter “3.2.1 Geographic Address [2..0] (GA[2..0])” in the PICMG AMC.0 specification. On DMMC-STAMP, there is a 3.3kOhm pull-up resistor on this pin.
AMC_GA0	B12	IN	Geographic address (bit 0), as defined in chapter “3.2.1 Geographic Address [2..0] (GA[2..0])” in the PICMG AMC.0 specification. On DMMC-STAMP, there is a 3.3kOhm pull-up resistor on this pin.
AMC_SDA	E12	IN/OUT	Serial data signal for IPMB, as defined in chapter “3.2.4 IPMB-L” in the PICMG AMC.0 specification. On DMMC-STAMP, there is a 3.3kOhm pull-up resistor on this pin.
AMC_SCL	E11	IN	Serial clock signal for IPMB, as defined in chapter “3.2.4 IPMB-L” in the PICMG AMC.0 specification. On DMMC-STAMP, there is a 3.3kOhm pull-up resistor on this pin.

Table 2: MCH/Backplane interface pin description

Pin Name	Location	Direction	Description
MTCA_HANDLE	D11	IN	Hot-plug handle, shall be low when handle is inserted. 10kOhm pull-up (to 3.3V MP) is integrated on the DMMC-STAMP SoM.
MTCA_LED_BLUE	B7	OUT	Blue LED on the front-panel, use appropriate current-limiting resistors (output is at 3.3V; 24mA max.). According to the AMC.0 specification it shall be located on the bottom side
MTCA_LED_GREEN	B6	OUT	LED 2 (green) on the front-panel, use appropriate current-limiting resistors (output is at 3.3V; 24mA max.). According to the AMC.0 specification it shall be located on the top side.
MTCA_LED_RED	B5	OUT	LED 1 (red) on the front-panel, use appropriate current-limiting resistors (output is at 3.3V; 24mA max.). According to the AMC.0 specification it shall be located on the top side.

Table 3: Handle and LED pin description

Shown in Table 4 are the on-board sensor interface signals. The DMMC-STAMP SoM provides a local I2C bus for the management related sensors. Typically, another two MAX6626 temperature sensor are connected on the target AMC to measure the inlet (address 0x49) and outlet (address 0x4A) temperature and to report the value to the MCH. MAX6626 I2C temperature sensor (address 0x48), the parameter EEPROM (address 0x51) and the RTM I2C current selection potentiometer (address 0x2E) are already connected to the bus on the DMMC-STAMP SoM. These addresses must not be used by external devices. When using the DMMC-SDK, more sensors such as FPGA/SoC temperature sensors can be connected here.

Pin Name	Location	Direction	Description
SNS_SCL	A10	OUT	Sensor bus management serial clock. On DMMC-STAMP, there is a 10kOhm pull-up resistor on this pin.
SNS_SDA	A9	I/O	Sensor bus management serial data. On DMMC-STAMP, there is a 10kOhm pull-up resistor on this pin.

Table 4: Sensor interface pin description

Shown in Table 5 are the JTAG interface signals connected to the backplane. The voltage level for these pins is 3.3V derived from VCC+3V3.

Pin Name	Location	Direction	Description
AMC_TMS	M8	IN	JTAG mode select in - connect directly to AMC TMS (pin 166)
AMC_TCK	M7	IN	JTAG clock input - connect directly to AMC TCK (pin 165)
AMC_TDI	M6	IN	JTAG data input - connect directly to AMC TDI (pin 169)
AMC_TDO	M5	OUT	JTAG data output - connect directly to AMC TDO (pin 168)

Table 5: AMC JTAG pin description (CPLD pins indicated in green)

5.2 Power Pins

The DMMC-STAMP SoM accepts management power (VCC+3V3_MP) and payload power (VCC+12V_PP) as input. From the payload power the Power System (see 3.5) generates a dedicated 3.3V output (VCC+3V3_MMC) that can be used on the target AMC. The 12V input to the RTM Hot-Plug Controller (see 3.4) is sourced from a dedicated power input pin (VCC_12VPP_IN). RTM management- and payload power are driven by separate pins (VCC_RTM+3V3_MP, VCC_RTM+12V). In addition, for operating the FPGA/SoC interfaces at the CPLD (see 3.2) one bank voltage per FPGA/SoC interface (VCC_FPGA1, VCC FPGA2) has to be supplied to the DMMC-STAMP SoM.

Shown in Figure 14 are exceptions from the schematics of AMC template (see 6.1.1). The 3V3 management- and 12V payload power is connected from the backplane (AMC connector) to LGA1A (pins A2 and F9 of the DMMC-STAMP SoM).

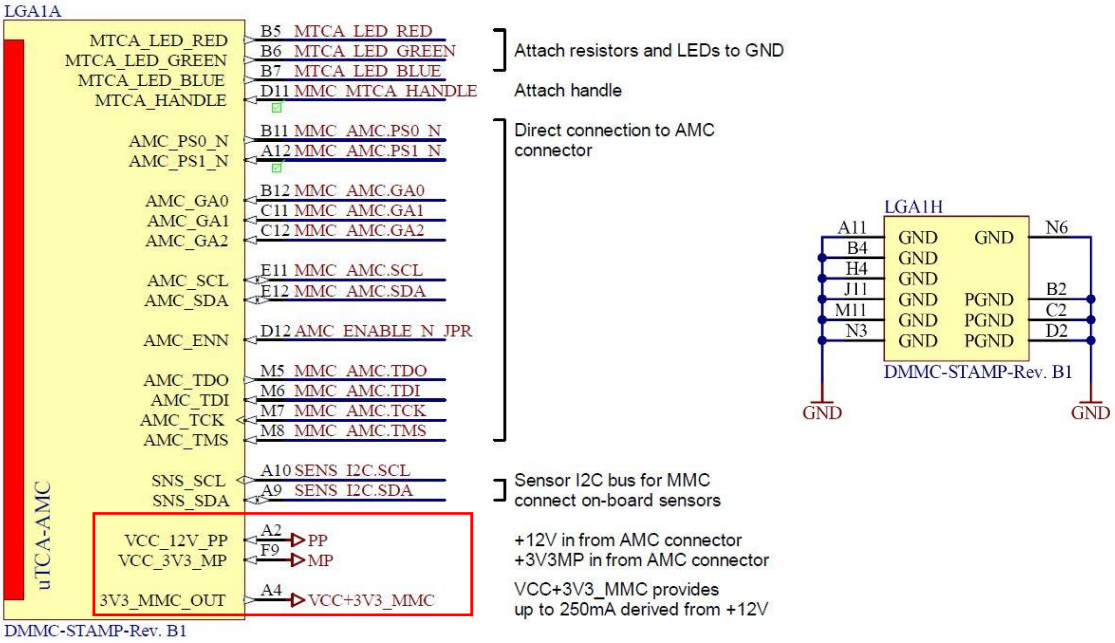


Figure 14: DMMC-STAMP SoM power input/output pins

Pin Name	Location	Direction	Description
VCC+3V3_MMC	A4	POWER OUT	Generated on the DMMC-STAMP SoM via DC/DC converter from payload power. external current draw: 250mA max
VCC+3V3_MP	F9	POWER IN	Management power - connect to AMC connector
VCC+12V_PP	A2	POWER IN	Payload power - connect to AMC connector
VCC_FPGA1	L6	POWER IN	See 5.4 FPGA/SoC Pins
VCC_FPGA2	L1	POWER IN	
VCC_12VPP_IN	M12	POWER IN	See 5.5 RTM Pins
VCC_3V3MP_RTM	K12	POWER OUT	
VCC_12V_RTM	J12	POWER OUT	
GND	A11, B4, H4, J11, M11, N3, N6	POWER GND	Ground
PGND	B2, C2, D2	POWER GND	Power ground of the DC/DC converter, to be connected to GND on the target AMC under the DMMC-STAMP SoM, directly (PGND and GND shall be connected to the same GND plane under the module)

Table 6: Power Pin Description

5.2.1 Global Power and Ground Routing Considerations

All power and ground nets should be routed with the lowest possible impedance. Multiple parallel type VII vias (resin-filled and copper-capped) placed directly inside the DMMC-STAMP pads are preferred. Power nets should be implemented as power polygons, and ground vias should connect directly to the solid internal ground planes of the PCB.

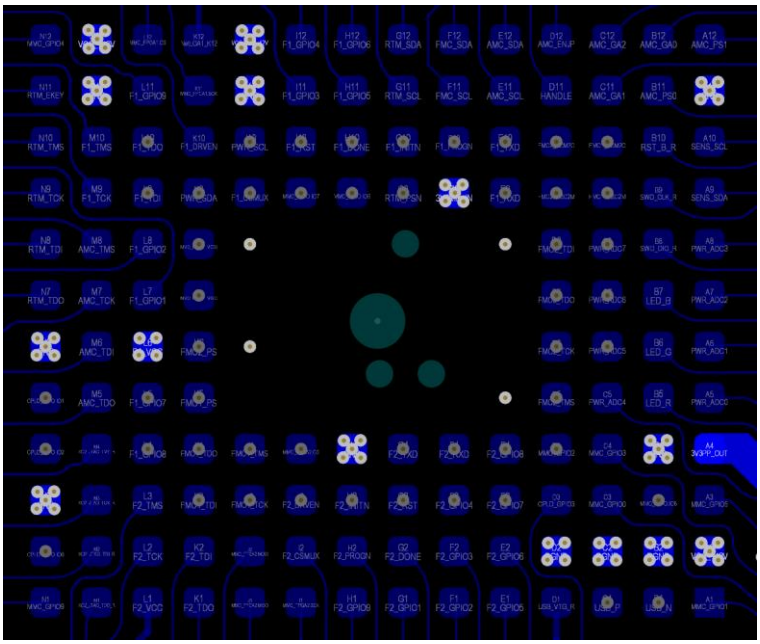


Figure 15: DMMC-STAMP Power and Ground connection using Type VII vias

When Type VII vias are not available (this is usually the case on cost-sensitive boards), large-diameter vias should be used -preferably several in parallel- together with wide traces to connect to the DMMC-STAMP power and ground pads. Inside the board, use solid power polygons to keep impedance low. Power polygon layer transitions should always be done with multiple vias in parallel. Ground pads should connect directly into a large ground plane located immediately beneath the DMMC-STAMP.

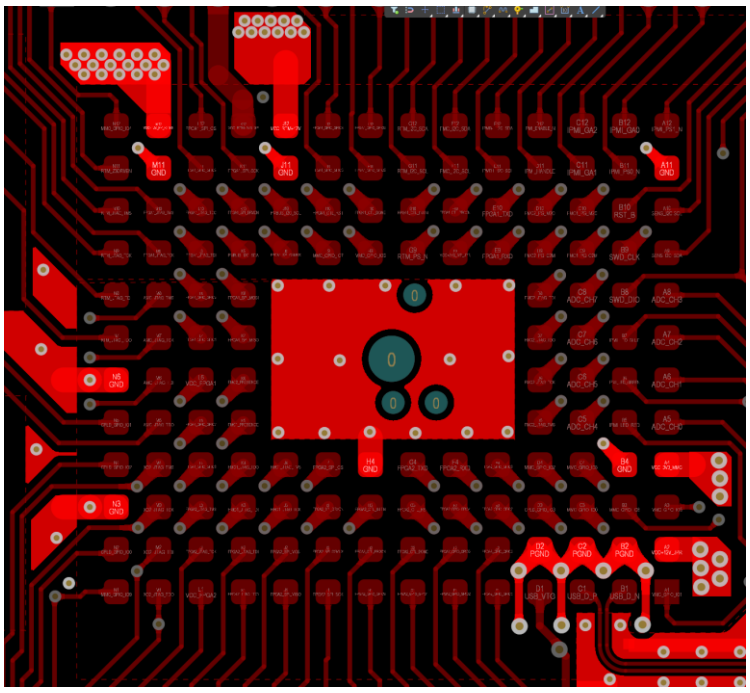


Figure 16: DMMC-STAMP Power and Ground connection using standard vias

5.3 Target AMC Power Management Pins

The DMMC-STAMP SoM provides the required infrastructure for a target AMC power management. With the pre-programmed firmware, the target AMC power management is limited to the report of the ADC voltages. The control of devices on the I2C bus is not foreseen and has to be implemented within a customized firmware using the DMMC-SDK (see 6.2).

Listed below are some additional information for the target AMC power supervision:

- The MCU can read voltages via 8 ADC inputs.
- All ADC inputs have a resolution of 12 bits and allow input voltages from 0 to 1 V.
- The MCU can interface PMBUS™ DC/DC converters or PMBUS™ power managers via a dedicated I2C bus (SDK required). On this bus address 0x70 is reserved for an I2C multiplexer located on the DMMC-STAMP.
- The MCU provides 10 GPIO pins with a 3.3 V level, derived from management power (SDK required to implement user functions, such as power enables).
- The MCU provides 4 CPLD GPIO pins with 3.3 V level, derived from payload power (SDK required to implement user functions, such as power enables).

Shown in Figure 17 are exceptions from the schematics of AMC template (see 6.1) showing the interface described above.

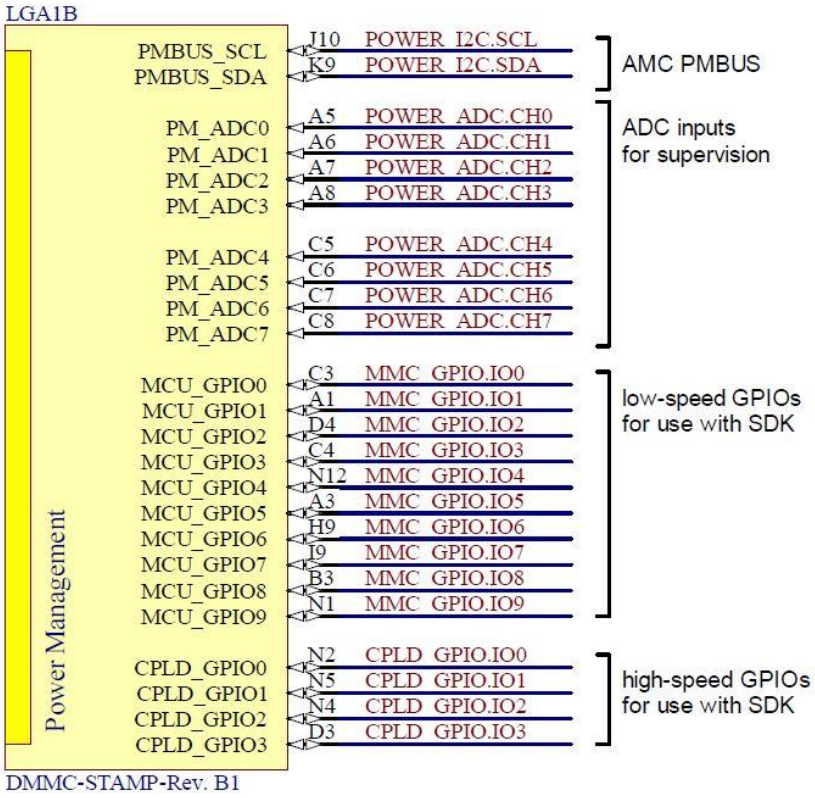


Figure 17:DMMC-STAMP power management pins

Pin Name	Location	Direction	Description
PMBUS_SDA	K9	I/O	PMBUS™ management serial clock. On DMMC-STAMP, there is a 5kOhm-equivalent pull-up resistor on this pin.
PMBUS_SCL	J10	OUT	PMBUS™ management serial data. On DMMC-STAMP, there is a 5kOhm-equivalent pull-up resistor on this pin.

Table 7: Target AMC PMBUS™ Pin Description

Pin Name	Location	Direction	Description
PM_ADC0	A5	IN	ADC inputs, channel 0
PM_ADC1	A6	IN	ADC inputs, channel 1
PM_ADC2	A7	IN	ADC inputs, channel 2
PM_ADC3	A8	IN	ADC inputs, channel 3
PM_ADC4	C5	IN	ADC inputs, channel 4
PM_ADC5	C6	IN	ADC inputs, channel 5
PM_ADC6	C7	IN	ADC inputs, channel 6
PM_ADC7	C8	IN	ADC inputs, channel 7

Table 8: ADC pin description for target AMC power monitoring

Table 9 shows the low-speed GPIOs controlled by the MCU which are operating in the management power domain (always available when management power is present).

Pin Name	Location	Direction	Description
MCU_GPIO0	C3	IN	Custom user function in MCU (SDK required)
MCU_GPIO1	A1	IN	Custom user function in MCU (SDK required)
MCU_GPIO2	D4	IN	Custom user function in MCU (SDK required)
MCU_GPIO3	C4	IN	Custom user function in MCU (SDK required)
MCU_GPIO4	N12	IN	Custom user function in MCU (SDK required)
MCU_GPIO5	A3	IN	Custom user function in MCU (SDK required)
MCU_GPIO6	H9	IN	Custom user function in MCU (SDK required)
MCU_GPIO7	I9	IN	Custom user function in MCU (SDK required)
MCU_GPIO8	B3	IN	Custom user function in MCU (SDK required)
MCU_GPIO9	N1	IN	Custom user function in MCU (SDK required)

Table 9: Low-speed MCU GPIO pin description

Table 10 shows the high-speed GPIOs controlled by the CPLD which are operating in the payload power domain with fixed 3.3V levels.

Pin Name	Location	Direction	Description
CPLD_GPIO0	N2	I/O	Custom user function in CPLD (SDK required)
CPLD_GPIO1	N5	I/O	Custom user function in CPLD (SDK required)
CPLD_GPIO2	N4	I/O	Custom user function in CPLD (SDK required)
CPLD_GPIO3	D3	I/O	Custom user function in CPLD (SDK required)

Table 10: High-speed CPLD GPIO pin description

5.4 FPGA/SoC Pins

The DMMC-STAMP SoM is able to interface up to two FPGAs/SoC via two separated LGA interface “banks”. Both “banks” are identical in terms of pin functionality where the pins itself operate at the voltage level defined by the input of the VCC_FPGA[1/2] pin.

Given below are some more detailed information about the FPGA interface pin functionality:

- UART (RXD/TXD): provides a virtual COM port interface to the FPGAs/SoCs that is accessible via the DMMC-STAMP USB-to-UART interface (see 3.3). There is only one FPGA/SoC UART interface which can be directed to the USB at one time (multiplexed within the CPLD). When the DMMC-SDK is used, one or both of these ports can be exposed over the network by using mmcterm (see 6.4).
- Configuration control signals: PROG, INIT, DONE, RESET
- JTAG interface that can be connected to the DMMC-STAMP SoM JTAG connector (multiplexed within the CPLD).
- SPI FLASH configuration interface which allows an in-system programming of the FPGAs/SoCs FLASH via HPM. This is very useful to allow the user to re-program the on-board FPGAs/SoCs without any additional software or hardware. A redundant SPI memory is supported as well (used to store “golden” image).
- GPIO interface: General-Purpose IO pins that can be used for a custom firmware. Users of the DMMC-STAMP SoM can use the DMMC-SDK to add their own functionality to these pins.
- Input voltage: VCC_FPGA[1/2] is an input pin and defines the output level of all signals of the FPGA/SoC interface “bank”, i.e. if the FPGA[1/2] configuration interface (typically in Bank 0) is operated at a 1.8V level, this voltage has to be fed to the VCC_FPGA[1/2] pin.

5.4.1 MMC Stamp Mailbox

The Mailbox is an optional I²C-based communication interface between the DMMC-STAMP and the FPGA, primarily intended for SoCs running an embedded OS. It is available only to customers who have purchased the DMMC-STAMP SDK. The Mailbox is enabled by compiling user-specific CPLD code that implements the required functionality. This user-specific CPLD image is usually integrated into the MMC firmware while compiling the SDK code. This ensures that CPLD updates are automatically deployed with the MMC firmware. CPLD firmware updates are fully managed by the MMC firmware and run automatically without any user-intervention when a CPLD contains an outdated image.

Using the Mailbox, the FPGA can communicate directly with the DMMC-STAMP, enabling controlled or graceful OS shutdown sequences.

Purpose and functionality

- System information exchange:

  The MMC-STAMP emulates a standard AT24Cxx I²C EEPROM interface inside its CPLD. The hard-coded address is 0x2A by default. This allows the FPGA to access relevant data such as MAC address, MTCA slot number, temperature values, and other operational parameters. During operating system boot, existing drivers that support reading from I²C EEPROMs can use the Mailbox without modification, treating it like a standard EEPROM device. After boot, a Yocto-based daemon takes over and manages continuous communication with the MMC-STAMP.

- Controlled shutdown handling:

  The Mailbox enables the MMC to signal the FPGA and the operating system when a shutdown or reboot is required, for example when the handle is pulled or the MCH initiates a reset. This mechanism ensures the operating system can shut down properly. Without it, unexpected power loss may occur, leading to boot image corruption and unstable system behavior.

- System integration:

  By providing a structured communication channel, the Mailbox allows close coordination between hardware management functions on the MMC and application logic implemented on the FPGA. In the absence of the Mailbox, the FPGA operates without access to management data and without the ability to handle controlled power transitions.

Implementation

The MMC Mailbox is implemented inside the STAMP CPLD. On the hardware side, the user must connect two FPGAx_GPIOy pins on the STAMP to serve as the I²C interface. These pins can be chosen arbitrarily, provided that the correct voltage level is ensured. External pull-up resistors, typically 10 kΩ, must be added to complete the I²C bus connection.

On the FPGA side, two pins of the Processing System (PS) should be chosen, so that the Mailbox functionality is available before the Programmable Logic (PL) has been initialized. Ideally, I²C-capable PS pins are selected. However, this is not mandatory: a software-based I²C implementation (soft-I²C) will also work on any PS I/O pin, which is fully supported by Yocto. Care must be taken to ensure that the FPGAx_GPIOy pins on the STAMP and the chosen PS bank operate at the same voltage level.

Shown in Figure 18 are exceptions from the schematics of AMC template (see 6.1) showing the FPGA/SoC interface described above.

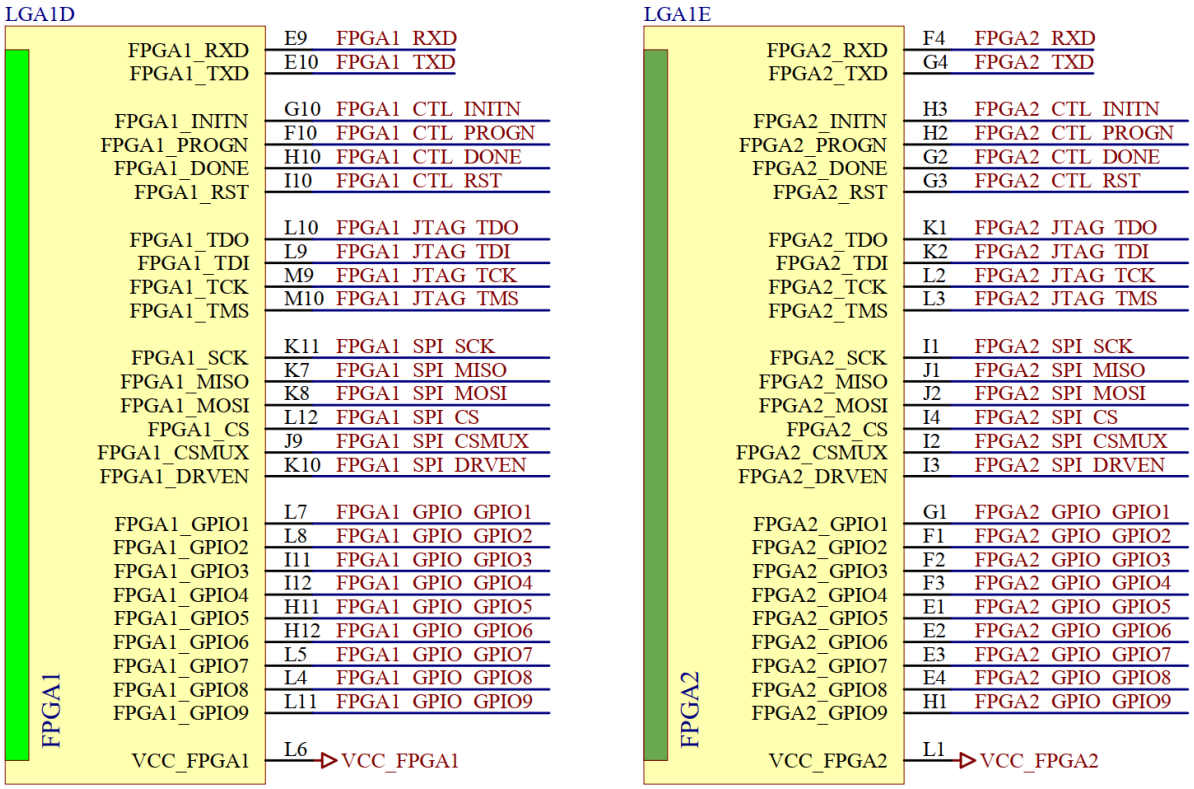


Figure 18: DMMC-STAMP FPGA/SoC interface pins

Pin Name	Location	Direction	Description
FPGA[1/2]_RXD	E9/F4	OUT	FPGA[1/2] UART receive
FPGA[1/2]_TXD	E10/G4	IN	FPGA[1/2] UART transmit

Table 11: FPGA/SoC UART pin description

Table 12 shows the FPGA/SoC JTAG signals which can be forwarded either from a JTAG connector on the target AMC or from the AMC backplane connector. These signals are multiplexed within the DMMC-STAMP SoM CPLD.

Pin Name	Location	Direction	Description
FPGA[1/2]_JTAG.TMS	M10/L3	OUT	JTAG Mode Select - connect to TMS pin of FPGA[1/2]
FPGA[1/2]_JTAG.TCK	M9/L2	OUT	JTAG Clock Output - connect to TCK pin of FPGA[1/2]
FPGA[1/2]_JTAG.TDI	L9/K2	OUT	JTAG Data Input - connect to TDI pin of FPGA[1/2]
FPGA[1/2]_JTAG.TDO	L10/K1	IN	JTAG DATA Output - connect to TDO pin of FPGA[1/2]

Table 12: FPGA/SoC JTAG pin description

Pin Name	Location	Direction	Description
FPGA[1/2]_RST	I10/G3	OUT	User firmware reset (active high), connect to any GPIO pin of the FPGA/SoC, a user firmware reset can be initiated with this pin
FPGA[1/2]_DONE	H10/G2	IN	A high signal on the DONE pin indicates completion of the configuration sequence. Connect to DONE on FPGA[1/2]. (a pull-up is needed according to the specific FPGA/SoC, see FPGA/SoC vendor configuration guide)
FPGA[1/2]_INITN	G10/H3	IN	Active-Low FPGA/SoC initialization pin or configuration error signal. Connect to INIT_B on FPGA[1/2] (a pull-up is needed according to the specific FPGA/SoC, see FPGA/SoC vendor configuration guide)
FPGA[1/2]_PROGN	F10/H2	OUT	Active-Low reset to configuration logic. Connect to PROGRAM_B on FPGA[1/2]

Table 13: FPGA/SoC Control pin description

Table 14 shows the fast DMMC-STAMP CPLD GPIO pins for FPGA/SoC control signals. DESY typically connects Bank 0 mode pins and additional FPGA/SoC control pins here (such as PS_POR_B, PS_SRST, PS_ERROR_OUT, PS_ERROR_STATUS, POR_OVERRIDE) and the MMC Mailbox. To implement a custom CPLD configuration image, the DMMC-SDK is required.

Pin Name	Location	Direction	Description
FPGA[1/2]_GPIO.GPIO9	L11/H1	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO8	L4/E4	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO7	L5/E3	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO6	H12/E2	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO5	H11/E1	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO4	I12/F3	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO3	I11/F2	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO2	L8/F1	I/O	Custom user function in CPLD (SDK required)
FPGA[1/2]_GPIO.GPIO1	L7/G1	I/O	Custom user function in CPLD (SDK required)

Table 14: FPGA/SoC GPIO pin description

Table 15 shows the FPGA/SoC SPI Flash configuration control pins. For operating a SPI Flash configuration memory on the target AMC two external components are needed (see below).

Pin Name	Location	Direction	Description
FPGA[1/2]_SPI.CS	L12/I4	OUT	MOSI for FLASH access (connect to buffer)
FPGA[1/2]_SPI.DRVEN	K10/I3	OUT	Driver activate signal (active_low). When the DMMC-STAMP wishes to access the SPI FLASH memories, the DRVEN signal is driven low.
FPGA[1/2]_SPI.CSMUX	J9/I2	OUT	Chip select multiplexer control (selects primary=high or redundant=low memory)
FPGA[1/2]_SPI.SCK	K11/I1	OUT	SCK for FLASH access (connect to buffer)
FPGA[1/2]_SPI.MOSI	K8/J2	OUT	MOSI for FLASH access (connect to buffer)
FPGA[1/2]_SPI.MISO	K7/J1	IN	MISO for FLASH access (connect to buffer)

Table 15: FPGA/SoC SPI Flash control pin description

5.4.2 Flash Access Scheme

In previous DESY MMC designs, the FLASH interface was routed to the MMC CPLD, directly. This simple scheme is not possible anymore. It is assumed, that the DMMC-STAMP SoM is placed far away from the FLASH memories which means that routing fast (100 MHz+) FLASH traces over long lines is not possible. To address this issue, the DMMC-STAMP SoM is separated via a discrete buffer and a discrete demultiplexer chip. This means that the traces can be kept short between the FLASH and the additional ICs in between so that signal integrity is not severely influenced (see Figure 19 to Figure 21).

Selection of primary/redundant FLASH memory:

- A 1-of-2 demultiplexer chip (e.g. 74LVC1G18) is used for decoding the FPGA/SoC chip select signal and for routing this signal to one of the two SPI memories. The chip should be placed next to the SPI memories, so that the CS path has minimal additional length.
- The DMMC-STAMP SoM controls the “select” input of the demultiplexer via the CSMUX output so that the primary/secondary memory can be selected via MMC CLI.
- If the 74LVC1G18 is used, pull-up resistors are needed on the CS lines at the FLASH memory.

Access to the SPI FLASH memories

- In order to access the FLASH memories, the DMMC-STAMP SoM pulls the FPGA/SoC into reset (PROG_N). The FPGA tri-states all configuration lines including the FLASH interface.
- When the FPGA/SoC is de-activated, the FLASH memories are reset, to terminate any special modes (e.g. x4 mode) that have been used for configuration and allow access in standard SPI mode.
- With the FPGA in reset, the external buffer is activated by pulling the DRVEN signal low. In this mode, the DMMC-STAMP SoM can access the SPI memories exclusively.
- All SPI pins that are unused in standard operation mode (HOLD/DQ3, WP/DQ2 etc.) need proper pull-ups or pull-downs so that these pins do not float during DMMC-STAMP SoM FLASH access.

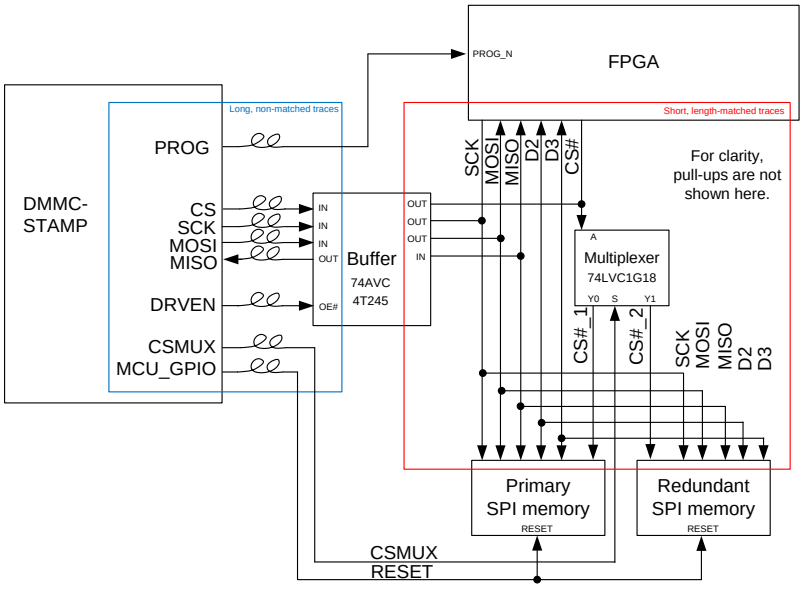


Figure 19: SPI Flash configuration memory connection scheme

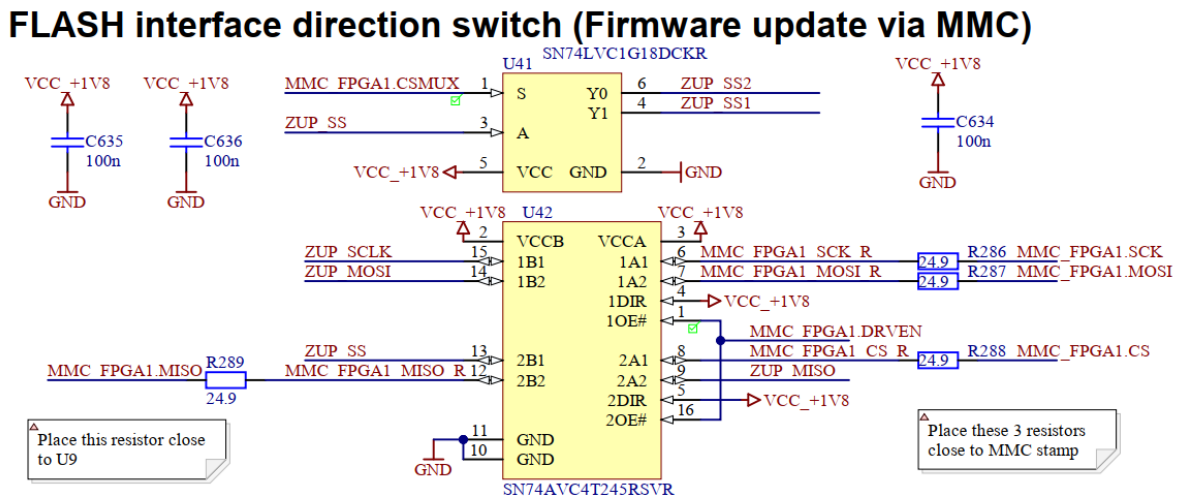


Figure 20: SPI Flash multiplexer implementation

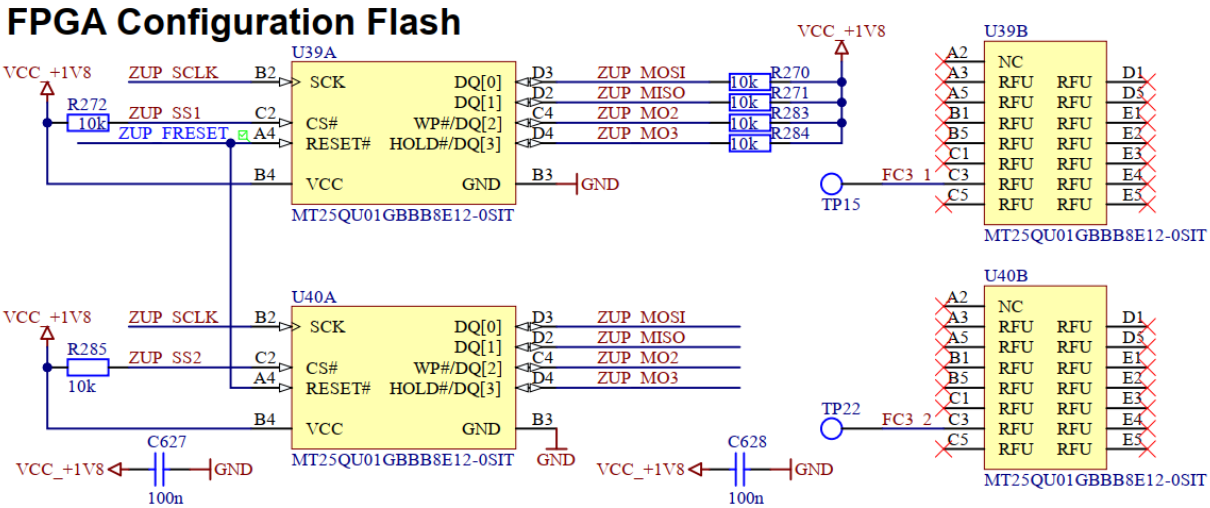


Figure 21: SPI Flash implementation

5.5 RTM Pins

The DMMC-STAMP SoM can manage one MicroTCA.4 compliant Rear Transition Module (RTM). It provides power control, including inrush-current limiting and a programmable overcurrent detection. Shown in Figure 22 is an exception from the schematics of AMC template (see 6.1) showing the RTM interface described here.

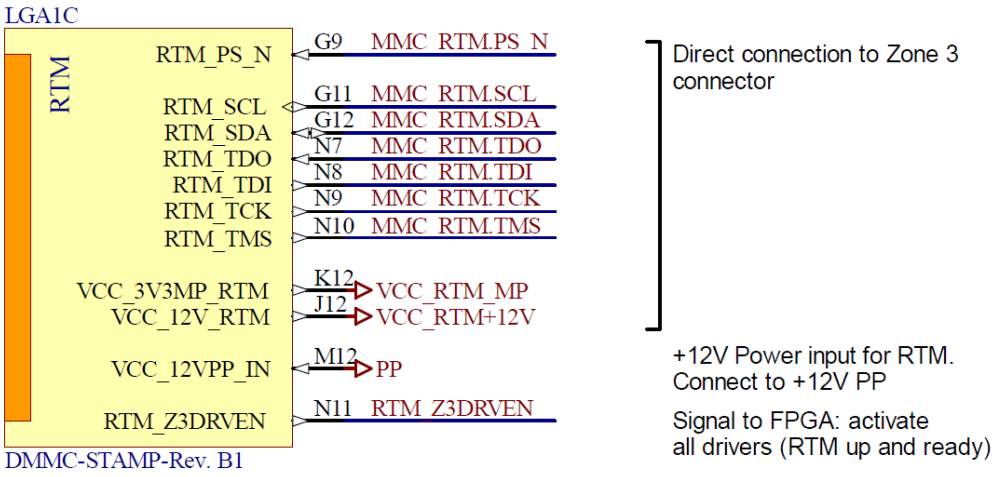


Figure 22: RTM interface pins

Pin Name	Location	Direction	Description
VCC+12VPP_IN	M12	POWER IN	Payload power input for RTM - connect to 12V payload power with a low-impedance power polygon to prevent voltage drop.
VCC_3V3MP_RTM	K12	OUT	Management power for RTM - connect to the Zone 3 connector
VCC_12V_RTM	J12	OUT	Payload power for RTM - connect to the Zone 3 connector

Table 16: RTM power pin description

Table 17 shows the RTM management signals. These signals are translated to the 3V3MP_RTM power domain. Pull-ups are integrated on the DMMC-STAMP SoM. All I2C signals are tri-stated when the RTM is not inserted.

Pin Name	Location	Direction	Description
RTM_SCL	G11	OUT	RTM management serial clock. On DMMC-STAMP, there is an 5kOhm-equivalent pull-up resistor on this pin. Do not connect any additional pull-up resistors to this pin.
RTM_SDA	G12	I/O	RTM management serial data. On DMMC-STAMP, there is an 5kOhm-equivalent pull-up resistor on this pin. Do not connect any additional pull-up resistors to this pin.
RTM_PS_N	G9	IN	RTM presence detect. Route to the PS# pin on the Zone 3 connector. On DMMC-STAMP, there is a 10kOhm pull-up resistor on this pin.

Table 17: RTM management signal pin description

Table 18 shows the FPGA/SoC tri-state control signal which will be used as a drive-enable flag for the Zone3 data interface signals (to/from the FPGA/SoC). The firmware on the target AMC FPGA/SoC has to disable all drivers to the RTM if this signal is low. Using this signal, RTM hot-plug support is achieved since the Zone 3 pins can be set to tri-state during RTM insertion or removal.

Attention:

The voltage level on this pin is 3.3V. Voltage translation (e.g. with a resistive divider) to the FPGA/SoC bank voltage is usually required. Alternatively, this information can be output on any FPGA[1/2]_GPIO pin since these signals are already translated to FPGA/SoC bank voltage (DMMC-SDK required).

Pin Name	Location	Direction	Description
RTM_Z3DRVEN	N11	OUT	This pin goes high when the RTM is in M4 state and is low in all other states

Table 18: FPGA/SoC tri-state control pin description

Table 19 shows the RTM JTAG access signal which operate in the +3.3V payload power domain. All JTAG signals are tri-stated when the RTM is not inserted.

Pin Name	Location	Direction	Description
RTM_TMS	N10	OUT	JTAG Mode Select - connect to TMS pin on RTM connector
RTM_TCK	N9	OUT	JTAG Clock Output - connect to TCK pin on RTM connector
RTM_TDI	N8	OUT	TDI - connect to TDI pin of Z3 RTM connector (directly routed to TDI pin of FPGA on RTM)
RTM_TDO	N7	IN	TDO - connect to TDO pin of Z3 RTM connector (directly routed to TDO pin of FPGA on RTM)

Table 19: RTM JTAG pin description

5.5.1 RTM start-up power limit control

To implement RTM current limiting, DMMC-STAMP uses a power distribution switch designed to control inrush and actively limit load current. It monitors the voltage across an external sense resistor and adjusts the gate of an external MOSFET to regulate and cap current. When the load current rises and the sensed voltage reaches the internal threshold, the controller begins throttling the MOSFET, which behaves like a variable resistor during this phase, continuously modulating it to maintain the current at the programmed limit. This mechanism regulates inrush, preventing excessive surges when charging RTM input capacitors, and it also activates during RTM failures when current demands exceed the MicroTCA specification. During this limiting phase, which lasts approximately 300 ms, the MOSFET operates in its linear region. During that phase, the MOSFET is operated near its Safe Operating Area limit. If the excessive current persists beyond this period, the RTM is shut down and a fault is indicated.

During startup, DMMC-STAMP temporarily raises the current limit moderately above 30 W to accommodate higher inrush demands. Once startup is complete, the limit returns to just above 30 W for normal operation. This approach allows safe powering of a constant 30 W load while handling inrush, ensuring stable startup and compliance with MicroTCA specifications.

Users can configure the current limit through the SDK. When exceeding the specified MicroTCA envelope (i.e., setting the limit above 30W), it is critical that any custom settings respect the MOSFET's

SOA; exceeding it may damage the device, for example by forcing it into a constant-on state, potentially compromising the system.

5.5.2 RTM-specific Power Routing Considerations

According to the MTCA standard, an RTM can draw up to 30 W, corresponding to 2.5 A at 12 V. The DMMC-STAMP temporarily increases this limit to about 50 W during RTM activation to allow faster charging of the RTM input capacitors, which corresponds to roughly 4 A. Many RTMs are sensitive to drops on their supply voltage. Because the MicroTCA power supply operates in a 10-14 V range, and additional losses occur on the AMC backplane and AMC connector, it is important to avoid introducing further voltage drop around the DMMC-STAMP. The resistance of the 12 V path feeding the DMMC-STAMP should therefore be kept as low as possible. Track width and layer transitions require particular attention. We recommend placing five vias directly into the 12 V power pads of the RTM power supply using Type VII vias. Power for the RTM should be routed exclusively through robust power power polygons, and any layer transitions should use multiple parallel vias to minimize resistance.

The net VCC+12VPP_IN (RTM input power) should be implemented as a large passive internal power polygon. The length and thickness of the power polygon on the VCC_12V_RTM net (power output towards RTM) should be chosen to keep the path voltage drop low, ideally around 0.1 V or better.

Management power consumption of the AMC and RTM are generally low. Because of that, the management rail receives no priority within the AMC backplane. To prevent additional voltage drop, a 1 mm wide trace should be routed from the MMC-STAMP to the Zone 3 connector.

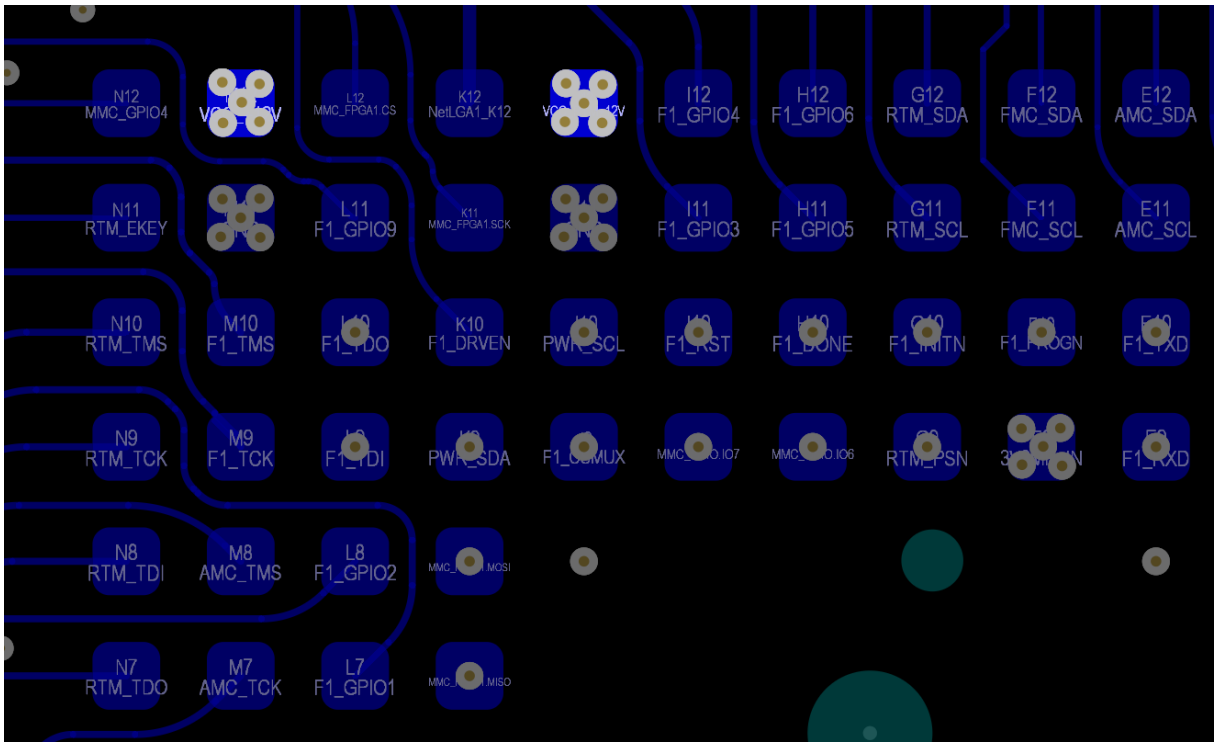


Figure 23: RTM power routing example using Type VII vias

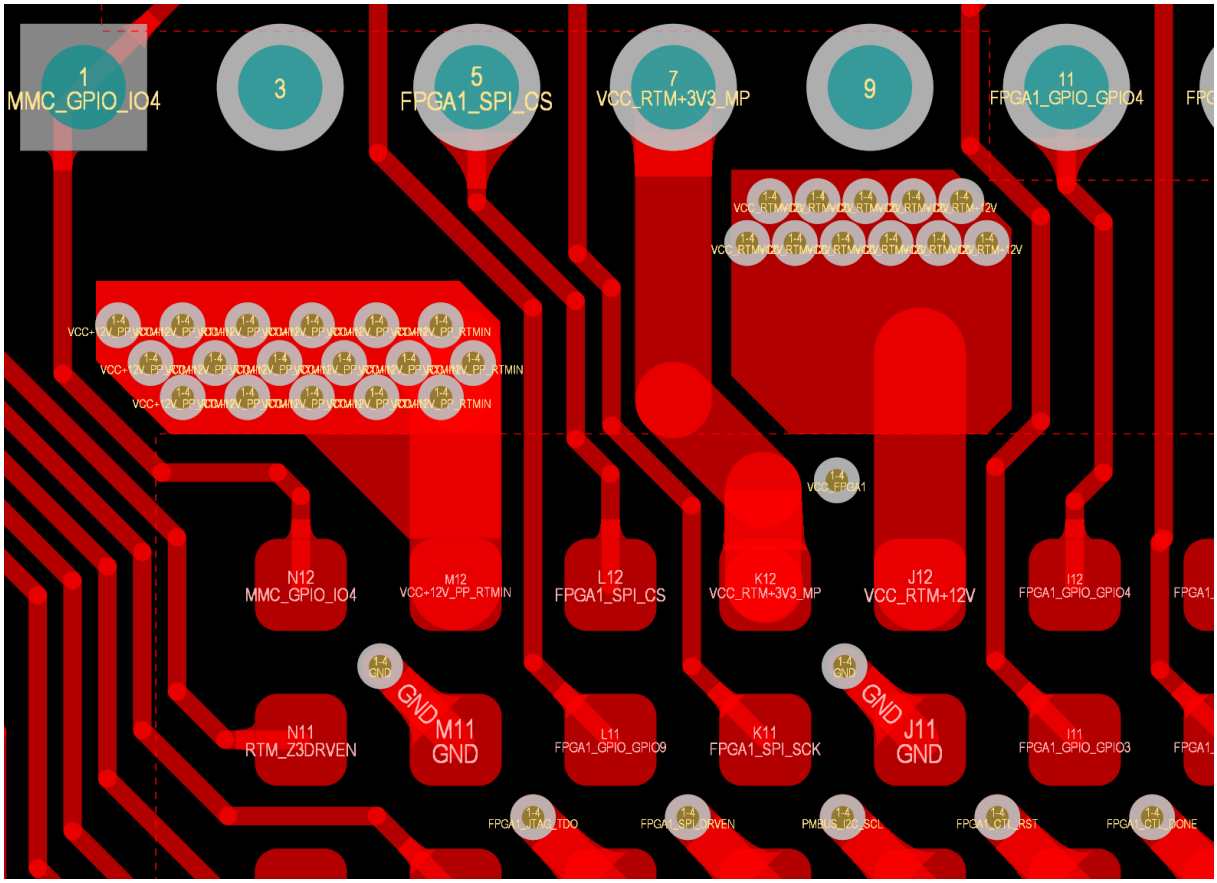


Figure 24: RTM power routing example using standard vias

5.6 FMC Pins

When the DMMC-SDK is used, the DMMC-STAMP SoM can manage up to two FPGA Mezzanine Card (FMC) modules. All pins have 3.3V level according to the FMC specification. The I2C bus is shared between both modules. Make sure that the GA pins of the FMC connectors differ from each other - only in this case the addresses of the FMC modules are unique and no address conflicts occur (see FMC specification). The address 0x70 is reserved on this bus as this is the address of the I2C multiplexer located on the DMMC-STAMP SoM.

All pins (except I2C) are routed to the DMMC-STAMP CPLD. Although the CPLD FMC Pins can be re-assigned using the SDK, these pins are not indeneded for custom functions (i.e. not related to FMC) since the FMC detection and activation logic may interfere with the desired user functions.

Please note that level translation is necessary in case levels different than 3.3V are desired.

Shown in is an exception from the schematics of AMC template (see 6.1) showing the FMC interface described here.

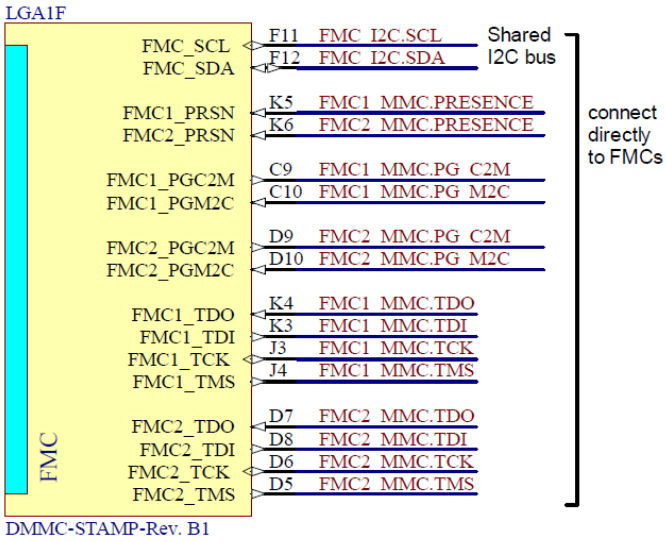


Figure 25: FMC interface pins

Pin Name	Location	Direction	Description
FMC_SCL	F11	OUT	FMC management serial clock (shared between FMC1 and FMC2). Ensure different I2C addresses of the FMC modules by assigning the GA[0..1] pins of the FMC connector different for both modules. On DMMC-STAMP, there is an 5kOhm-equivalent pull-up resistor on this pin.
FMC_SDA	F12	I/O	FMC management serial data (shared between FMC1 and FMC2). On DMMC-STAMP, there is a 10kOhm pull-up resistor on this pin. On DMMC-STAMP, there is an 5kOhm-equivalent pull-up resistor on this pin.
FMC[1/2]_PRSN	K5/K6	IN	FMC presence detection. Connect to presence pin of FMC connector. MCU-internal weak pull-up is included in the DMMC-STAMP (see ATSAM4L spec. —> References)
FMC[1/2]_PGC2M	C9/D9	OUT	Power-good signal that is asserted by DMMC-STAMP
FMC[1/2]_PGM2C	C10/D10	IN	Power-good signal that is asserted by FMC module CPLD-internal weak pull-up is included in the DMMC-STAMP (see MachXO2 spec. —> References)

Table 20: FMC management pin description

Pin Name	Location	Direction	Description
FMC[1/2]_TMS	J4/D5	OUT	JTAG Mode Select - connect to TMS pin on FMC[1/2]
FMC[1/2]_TCK	J3/D6	OUT	JTAG Clock Output - connect to TCK pin on FMC[1/2]
FMC[1/2]_TDI	K3/D8	OUT	JTAG Data Input - connect to TDI pin on FMC[1/2]
FMC[1/2]_TDO	K4/D7	IN	JTAG Data Output - connect to TDO pin on FMC[1/2]

Table 21: FMC JTAG pin description

5.7 Debug Pins

The DMMC-STAMP SoM is equipped with a pre-programmed firmware image for the on-board CPLD and MCU providing access to the debug interfaces described here. It is up to the user to decide which debug interface to route on the target AMC (see Figure 26):

- USB: Provides CLI access to MMC and user FPGAs. Highly recommended, no diagnostic output or user configuration is possible without USB connector.
- JTAG: provides access to the FPGAs/SoC and the DMMC-STAMP SoM CPLD, fallback programming option for CPLD
- Cortex-M4 MCU debug access, recommended when using the DMMC-SDK to program user-defined functions into the DMMC-STAMP

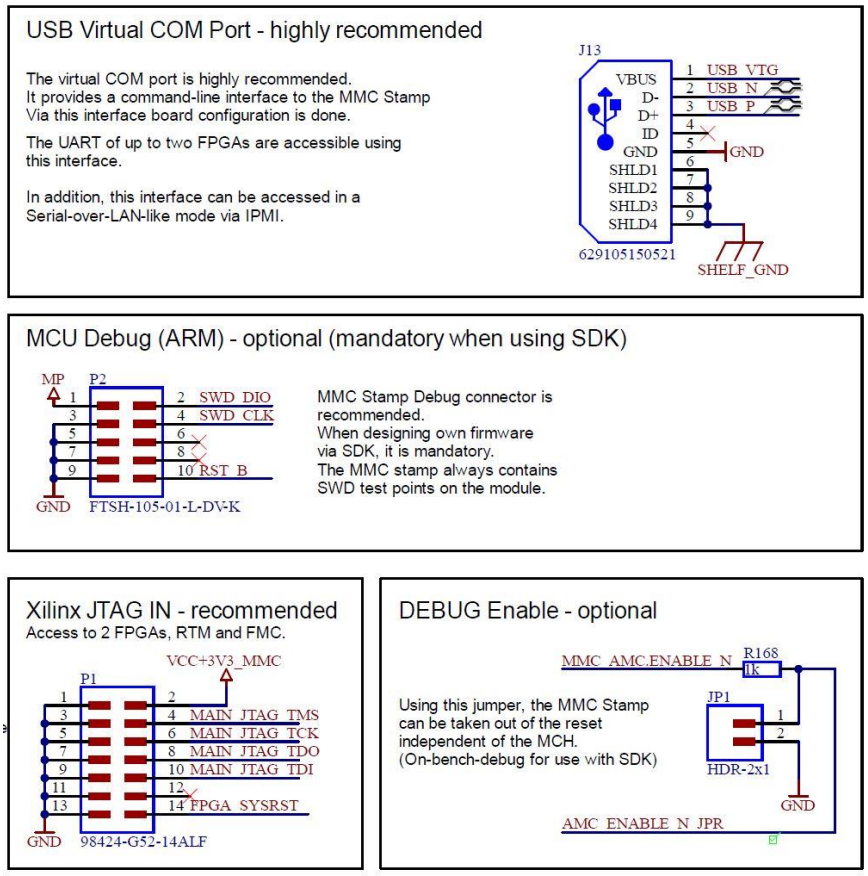


Figure 26: DMMC-STAMP debug interfaces

Shown in Table 22 are the central JTAG connector pins. The voltage level for these pins is 3.3V which is derived from VCC+3V3.

Pin Name	Location	Direction	Description
CON_TMS	M4	OUT	JTAG Mode Select IN - Connect directly to TMS on on-board connector.
CON_TCK	M3	OUT	JTAG Clock Input - Connect directly to TCK on on-board connector.
CON_TDI	M2	OUT	JTAG Data Input - Connect directly to TDI on on-board connector.
CON_TDO	M1	IN	JTAG DATA Output - Connect directly to TDO on on-board connector.

Table 22: Central JTAG connector pin description

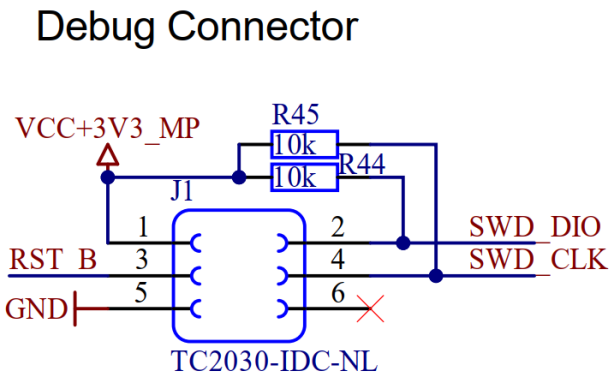


Figure 27: DMMC-STAMP debug interfaces

Pin Name	Location	Direction	Description
USB_VTG	D1	POWER INPUT	+5V from USB connector. This voltage is used to power the CP2105 USB-to-UART bridge (host-powered). The DMMC-STAMP translates all UART signals to the MP (MCU) and PP (CPLD) domain. Connect directly to USB connector.
USB_DP	C1	IN/OUT	USB Data+. Connect directly to USB connector.
USB_DM	B1	IN/OUT	USB Data-. Connect directly to USB connector.

Table 23: USB connector pin description

Pin Name	Location	Direction	Description
SWD_DIO	B8	IN/OUT	Serial Wire Debug DATA in/out (parallel to TC2030 connector). On DMMC-STAMP, there is an 10kOhm pull-up resistor on this pin.
SWD_CLK	B9	IN	Serial Wire Debug CLOCK (parallel to TC2030 connector). On DMMC-STAMP, there is an 10kOhm pull-up resistor on this pin.
RST_B	B10	IN	Microcontroller Reset IO (parallel to TC2030 connector). This line is gated by the IPMI Reset. To release the microcontroller from Reset, the MCH has to drive AMC_EN_N low. For on-desk operation (without MCH) the AMC_EN_N has to be grounded (e.g. with a Jumper) to allow normal operation of the Microcontroller.

Table 24: SWD pin description

5.8 Interlock forwarding to RTM

Interlocks are safety mechanisms designed to prevent hazardous conditions. Since they are crucial for machine safety, the distribution of safety information must function under all circumstances. In particular, it must be independent of the state of certain logic components (e.g., the FPGA or its firmware).

The following chapter describes how this functionality is implemented in the DESY installations. This implementation is highly specific and not part of the MicroTCA standard. However users may wish to implement similar safety or multi-point board-to-board communication functionality, the principle is documented here. All DMMC-STAMP-based AMCs support Interlock functionality described here. For customers who wish to design their own custom AMCs, the DMMC-STAMP SDK is required to implement similar functionality on custom boards.

5.8.1 M-VLDS Reception from Backplane

The standard PICMG backplane defines a region of pins used for Multipoint-LVDS connections - specifically, Ports 17 to 20. This group enables fast, low-latency communication between AMCs and is highly suited for one-to-many trigger signal infrastructure.

Typically, for each M-LVDS line, exactly one AMC is designated as the sender, and all other AMCs act as listeners. Multi-sender scenarios (open-drain-like configurations) are also possible but require special precautions to prevent short circuits between multiple drivers.

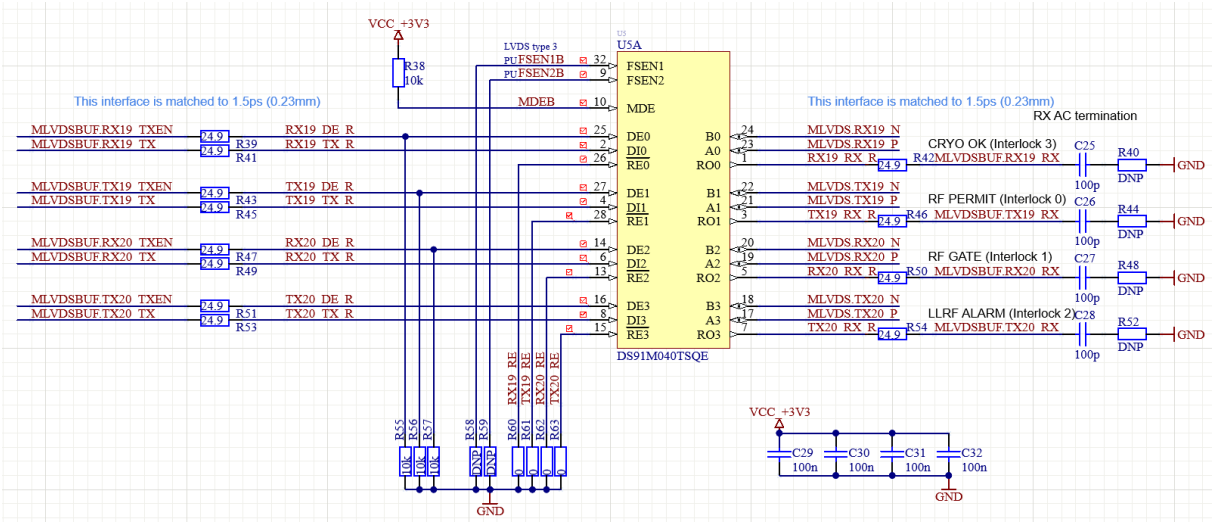


Figure 28: Typical M-LVDS Implementation

Using DS01M040 transceivers, every card listens to the M-LVDS bus. The single AMC designated as the sender can pull the “DEx” line high to transmit on that lane to all other AMCs (listeners). For each lane only one AMC can be a sender at a given time.

5.8.2 DESY-specific Interlock generation and forwarting to RTM

At DESY, the interlock information is generated (only) by the Machine Protection AMC (sender) and distributed to all AMCs (reveivers) via M-LVDS signals. These signals can be decoded by all FPGAs on all AMCs simultaneously with minimal latency.

This is the DESY-specific Backplane assignment:

Lane	Assignment	Forwarding to RTM via DMMC-STAMP
RX 17	TIMING CLOCK (trigger & data)	no
TX 17	TIMING DATA	no
RX 18	LLRF TRIG (in LLRF crates) or BEAM INFO (in master timing and MPS crates only)	no
TX 18	Spare (could be used for RESET in crates without an RF backplane)	no
RX 19	CRYO OK (interlock & alarm)	YES
TX 19	RF PERMIT	YES
RX 20	RF GATE	YES
TX 20	LLRF ALARM	no

Table 25: DESY-Specific M-LVDS assignment

Sending and receiving data from or by the AMC works completely independently of the DMMC-STAMP. However, since these signals are critical for machine protection, some of them (CRYO OK, RF PERMIT, RF GATE) are also relevant for the RTM. Because this is safety-critical, the ability to forward these signals to the RTM must not depend on the state of the FPGA on the AMC. Instead, the distribution must be implemented in hardware that operates entirely independently of the FPGA.

This forwarding, along with its configuration, is handled within the DMMC-STAMP. Eventually the interlock will be available on the OUTx lanes of the Zone 3 connector.

For this purpose, the three relevant lines - RX19, TX19 and RX20 - are “copied” to the DMMC-STAMP and fed into its CPLD, allowing them to be decoded independently.

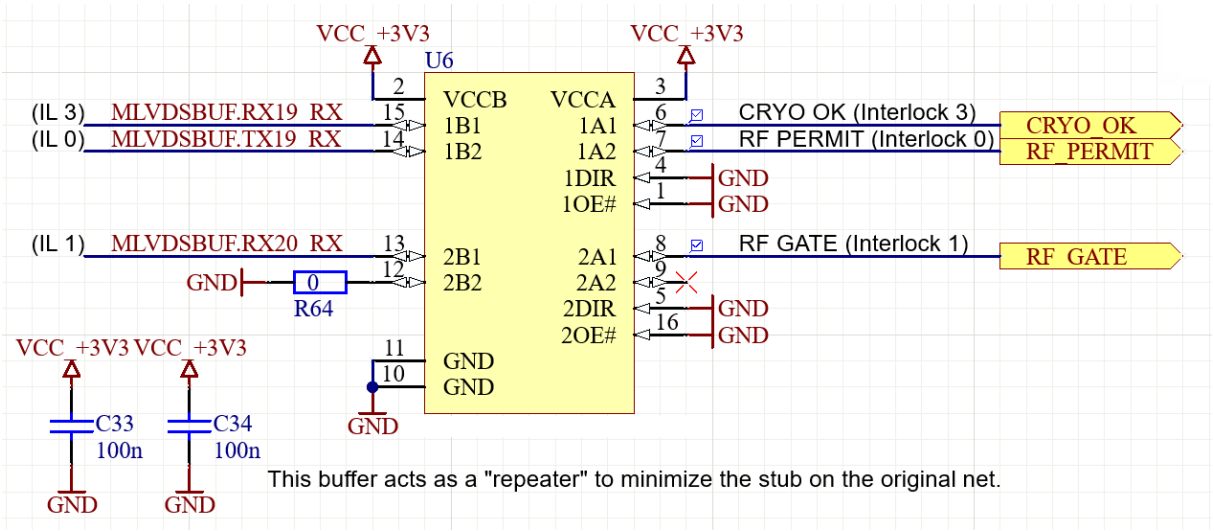


Figure 29: Typical „M-LVDS-copy-to-RTM“ implementation

The buffer chip (such as the 74AVC4T245, 74LVC245A, 74AVC1T45, or a similar device) must be placed so that the stub – ideally only one via - on the receiver line (left side) toward the FPGA is as short as possible, ensuring that fast signal reception by the FPGA is not impaired. The routing must be designed to support data rates of 125 MHz / 250 Mbit/s or even more, depending on the system architecture.

The left side is routed toward the STAMP. Here, longer trace lengths (typically in the range of 100–200 mm) are common and acceptable, provided that proper length matching is applied.

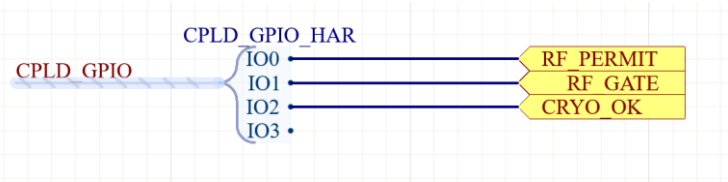


Figure 30: Outputs of M-LVDS receivers connection to DMMC-STAMP

The ideal connection point for the buffer outputs are the CPLD_GPIOx pins, as they operate at a fixed 3.3V levels and are therefore typically unused. Alternatively, these signals can be routed to FPGAx_GPIOy pins; however, in that case, voltage translation to the appropriate bank voltage (i.e., to the VCC_FPGAx level of the same bank) is required.

The DMMC-STAMP contains configurable interlock logic to process the interlock signals and configure their forwarding to the RTM. After the interlock information has been processed by the DMMC-STAMP, it is forwarded to the OUTx lanes on the Zone 3 connector. It is important to note that this M-LVDS signal reception and forwarding to the RTM are performed by dedicated hardware, meaning that the function operates independently of the FPGA state - for example, even during an FPGA firmware update.

The forwarding to the Zone 3 pins can occur on all CPLD pins of the STAMP. Again, the FPGAx_GPIOy pins are well suited for this purpose. Since Zone 3 supports LVDS only, translation to LVDS must be implemented (e.g., via U65, U68, U69). The LVDS buffers originate from relatively old technology; therefore, these buffers (such as NBA3N011S or DS90LV011A) are available only with 3.3V LVCMOS interfaces on the single-ended side. Consequently, translation from the single-ended DMMC-STAMP signal levels to 3.3 V must often occur.

Typically, the single-ended-to-LVDS buffers are connected to FPGAx_GPIOy pins, which usually requires voltage translation. It is essential to match the voltage of the buffer (U68 in the image below, using components such as 74AVC4T245 or 74AVC1T45) to the voltage level of the corresponding CPLD bank (VCC_FPGAx). This is achieved by U67 in the schematic.

Another important consideration is the requirement to tri-state all outputs of the Zone 3 interface when the RTM is in the M4 state. The only way to achieve this is by disabling power to the LVDS buffers, which is accomplished by generating a dedicated voltage that can be enabled or disabled via the Z3DRVEN pin. Note that VR1 powers the LVDS drivers as well as the right side of buffer U67. This allows the Zone-3 interface to be fully off when the RTM is in the M1 or M7 state.

The tri-state requirement is mandatory and must not be ignored, as it prevents potential damage to the AMC or RTM during hot-plugging of RTMs.

The following picture illustrates a discrete LVDS driver stage. Please note that the DMMC-STAMP signals OUTx_BUF operate at 3.3 V and therefore undergo translation via U67 to enable tri-stating of the LVDS outputs.

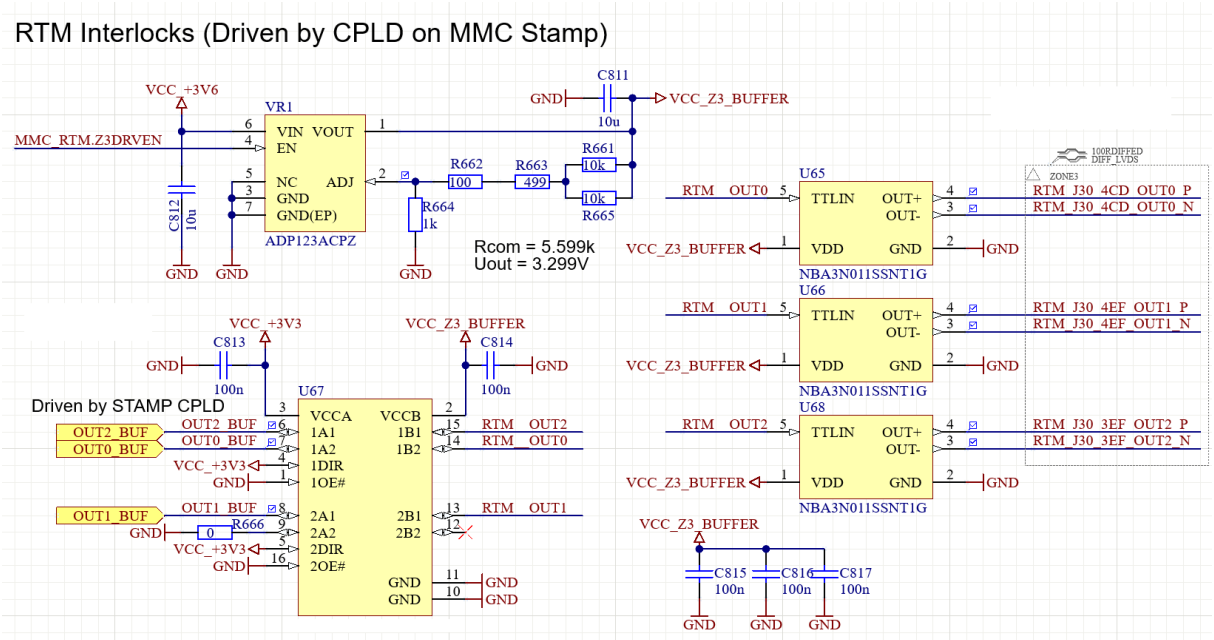


Figure 31: Zone3 OUTx Implemenatation example 1

The following figure illustrates an alternative approach using a tri-state-capable LVDS driver, the SN65LVDS047. In this example, the DMMC-STAMP output signals operate at 1.8 V; therefore, a level translation to 3.3 V is required for compatibility with the SN65LVDS047. Disabling the Zone 3 outputs is accomplished by connecting the DMMC-STAMP's Z3DRVEN signal to the driver's enable pin.

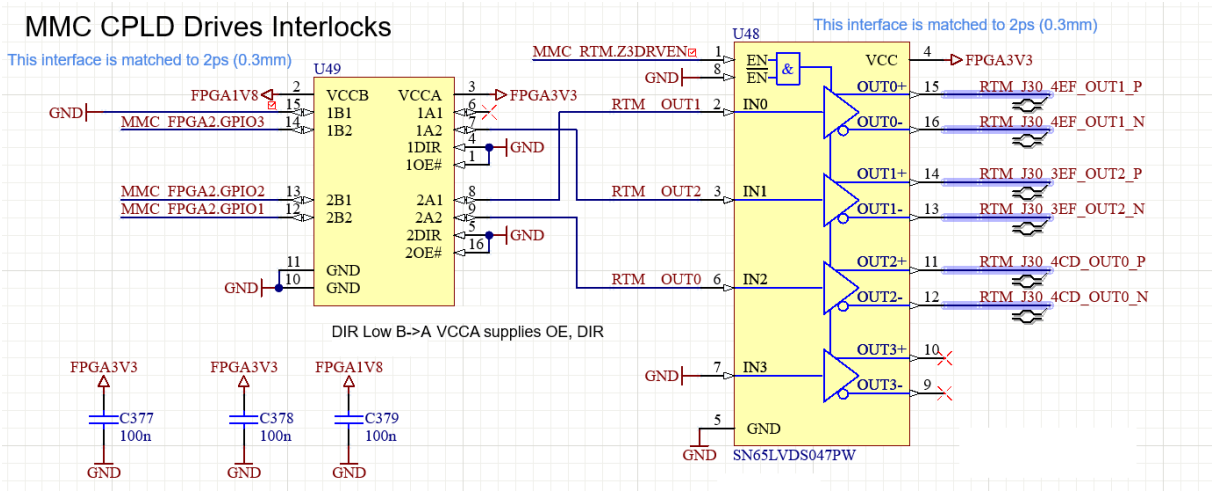


Figure 32: Zone3 OUTx Implemenatation example 2

5.8.3 Interlock Forwarding Configuration inside DMMC-STAMP

The DMMC-STAMP receives three signals and one additional signal from the FPGA to emulate a NAND gate, which drives the OUTx lanes of the Zone 3 connector. Alternatively, each OUTx lane can be driven directly by the FPGA if required.

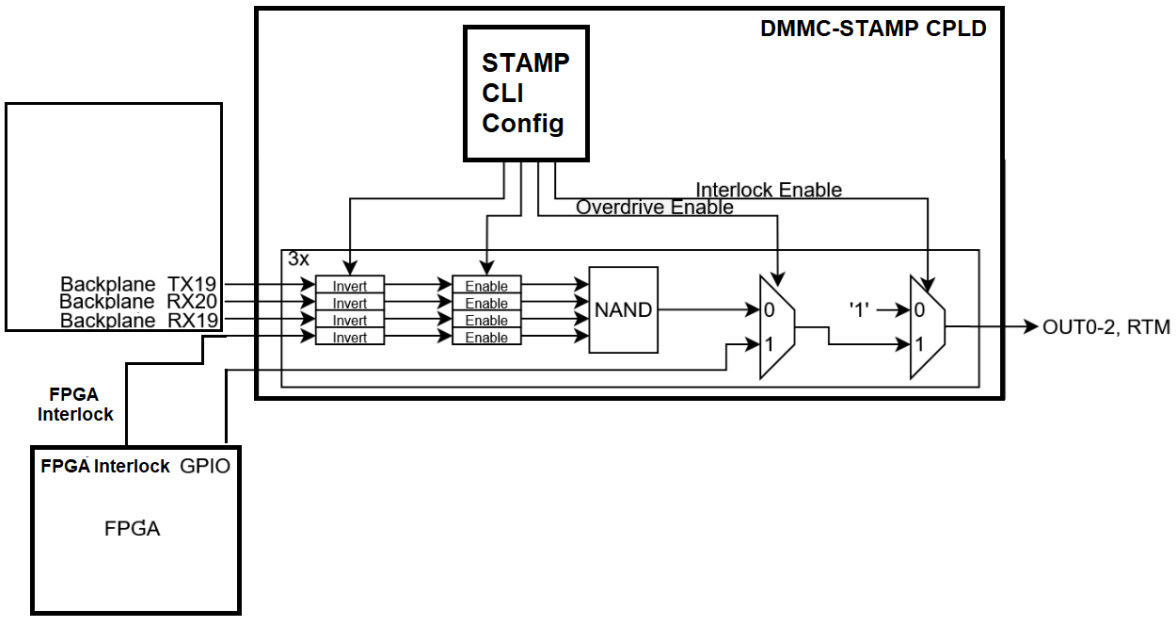


Figure 33: Interlock Logic on DMMC-STAMP

The CLI of the DMMC-STAMP allows configuration of each Zone 3 OUTx lane to operate in one of three modes:

- Static Mode: Each OUTx lane can be set to a fixed logic level - either high or low.

- NAND Mode: Each OUTx lane can function as the output of a NAND gate. The gate has four configurable input sources: backplane TX19 (RF Permit), backplane RX20 (RF Gate), backplane RX19 (Cryo OK), and FPGA Interlock. Each of these four inputs can be individually enabled or disabled as relevant to the NAND logic. The output becomes valid only when all selected (i.e., relevant) inputs to the NAND gate are valid.
- Override Mode: In this mode, the FPGA has direct control over the logic level of each OUTx pin on the Zone 3 connector.

5.9 Pre-Installed Firmware

With delivery the DMMC-STAMP SoM is fully tested and pre-programmed with a default MMC firmware. The feature set of this default firmware is basically comprehensive enough to operate an AMC according to the PICMG MicroTCA® specification, i.e. handling of the mandatory components as well as power management is included. Furthermore, there are some DMMC-STAMP SoM hardware related add-on functionalities that are integrated into the default firmware.

Given with this chapter are some more detailed information about the functionality of main building blocks of the pre-programmed default firmware.

5.10 Full IPMI Stack

The DMMC-STAMP SoM default firmware contains a full Intelligent Platform Management Interface (IPMI) Stack to make the target AMC fully operational within in MicroTCA® ecosystem. In more detail the MMC features the handling of IPMI messages, Sensor Data Records (SDR), a System Event Generator and Field Replaceable Unit (FRU). IPMI Messages are send by using the Intelligent Platform Management Bus (IPMB) protocol.

5.10.1 Sensor Data Records (SDR)

According to the AMC specification the Module Management Controller (MMC) has to contain information about the on-board sensors (e.g. for voltages or temperature) and the position of the Hot Swap handle. Sensors can also specify thresholds where different actions must be taken, e.g. for maximum allowable temperature of the components of the boards. This sensor information is stored in the Sensor Data Records (SDR) of the MMC firmware as described in the IPMI specification.

Shown below is the SDR listing from the DMMC-STAMP pre-installed default firmware, as obtained from ipmitool. When developing an AMC, the user can customize the SDR of the MMC according to the sensors placed on the board.

```
$ ipmitool -I lan -H <hostname> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> \
                                                          sdr list all
STAMP             | Dynamic MC @ 7Eh  | ok
AMC Hot Swap      | 0x00              | ok
801F127C4431      | 0x00              | ok
STAMP Temp        | 22.50 degrees C   | ok
AMC MP 3V3        | 3.22 Volts        | ok
AMC PP 12V        | 12.50 Volts       | ok
I_RTM MP 3V3      | 0 Amps            | ok
I_RTM PP 12V      | 0 Amps            | ok
CPLD Done         | 0x01              | ok
RTM MP 3V3 PG     | 0x00              | ok
RTM PP 12V PG     | 0x00              | ok
```

```
RTM Fault         | 0x00                | ok
```

5.10.2 System Event Generator

The DMMC-STAMP SoM pre-installed firmware contains a System Event Generator (as a part of the System Even Log, SEL) which is capable of generating events according to the IPMI specification. These events are used to e.g. notify the shelf manager about the Hot Plug handle begin inserted and extracted or in the case sensors (e.g. voltage and temperature) exceed pre-defined limits.

Users of the DMMC-STAMP do not need to change either the Event Generator implementation or any related setting. The sensor threshold limits are specified in the SDR (see above).

5.10.3 Field Replaceable Unit (FRU)

The Field Replaceable Unit (FRU) is stored in the User Page of DMMC-STAMP SoM MCU Flash. The structure of the FRU data is defined in the IPMI specification (IPMI – Platform Management FRU information Storage Definition). In a MicroTCA® based environment the FRU contains additional records defined by the PICMG (PICMG AMC.0, PICMG MTCA.0 and PICMG MTCA.4).

Table 26 shows the structure of the FRU data. The DMMC-STAMP SoM pre-programmed default firmware includes a default FRU and users have to customize and extend the content e.g. by using frugy (see 0).

Field	Description
Common Header	Indicates that DMMC-STAMP provides Board and Product Info and MultiRecords.
Board Area	Provides information about the board - the user of the DMMC-STAMP should change this field. Manufacturing date and FRU File ID can also be stored here.
Product Area	Provides information about the product, similar as for the *Board Area*. It is assumed that most boards are also products
MultiRecord Area	Contains: PICMG Module Current Requirements record PICMG AdvancedMC Point-to-Point records PICMG Zone 3 Interface Compatibility record PICMG Clock Configuration record

Table 26: FRU date structure

The FRU data can be obtained from the MMC CLI or by using IPMI tool:

```
$ ipmitool -I lan -H <hostname> -P "" -B 0 -b 7 -T 0x82 -t <fru-id> \
                                                              fru print 0
 Board Mfg Date        : Mon Jan  1 01:00:00 1996
 Board Mfg             : DESY
 Board Product         : DMMC-STAMP-Breakout Rev.B
 Board Serial          : 0000
 Board Part Number     : 0000
```

Product Manufacturer : DESY

```
 Product Name          : DMMC-STAMP-Breakout Rev.B
 Product Part Number   : 0000
 Product Version       : 0000
 Product Serial        : 0000
 Product Asset Tag     : none
```

5.11 Command Line Interface

The DMMC-STAMP SoM MMC default firmware features a command line interface (CLI) for configuration access. A connection to the CLI can be established either by using a “serial over IPMB” connection with mmcterm (see 6.4) or by using channel 1 of the USB-to-UART bridge (see 3.3). For connecting to the virtual COM port, one has to use the following parameters:

- Baud rate: 115200
- Data bits: 8
- Stop bits: 1 (8N1)
- Parity: None
- Flow control: None

Further details on how to connect to the MMC CLI are given in the DMMC-STAMP-BoB User Manual. Shown below are the help prints of the MMC CLI which is basically a list of available commands:

```
STAMP@0x7E MMC>?
? / h / help: Show list of available commands
c ..................................................... Clear screen
r ..................................................... Reset MMC
v ..................................................... Show firmware version
xm [0..n] ............................................. Start XMODEM update
sb .................................................... Start bootloader
vb [0..6] ............................................. Get/set verbosity
tm [smart|dumb|auto] .................................. Get/set terminal mode
eefd .................................................. Set EEPROM factory
defaults
s ..................................................... Get status
lc [0..3] [on|off|blink] [on_ms] [off_ms] ........ Set LED
ser [addr] [lun] ...................................... Get/set event receiver
pu .................................................... Payload power up
pd .................................................... Payload power down
ppf [stop|retry|ignore] ............................... Get/set payload power
fail policy
pc .................................................... Toggle CPLD programming
/ JTAG forwarding mode
sj [con|bp|raw] [fpga(1|2|12)|rtm|fmc(1|2)] ...... Get/set JTAG
multiplexing
st [0..15] ............................................ Get/set RTM temp sensor
mask
rte [enable|override] ................................. Get/set RTM e-keying
policy
rto [calibrate] ....................................... Get/set I_RTM PP 12V
calibration
cfu ................................................... CPLD force update
fru [0..n] ............................................ Dump FRU information
rtp [auto|high|low] ................................... Get/set RTM Power Good
polarity
i2cd [ipmb|sens|rtm|pmbus] ............................ Detect I2C peripherals
i2cget [ipmb|sens|rtm|pmbus] [addr] [reg] [nbytes] Get I2C peripheral
register
i2cset [ipmb|sens|rtm|pmbus] [addr] [reg] [data..] Set I2C peripheral
register
sg [mmc|fpga1|fpga2|reset] [0..9] [i|o] [0|1] .... Set MMC GPIO direction &
level
```

5.12 Reconfigurable JTAG Chain

For a configurable JTAG access of the devices connected to the DMMC-STAMP SoM the pre-installed default firmware provides a kind of arbitration functionality. By using appropriate commands on the CLI users can switch between the different JTAG source and targets. The DMMC-STAMP SoM provides interfaces for all typically AMC required JTAG sources and targets which are listed below.

JTAG Sources:

- AMC board JTAG connector
- AMC connector (backplane)

JTAG Targets:

- FPGA1
- FPGA2
- RTM
- FMC1
- FMC2
- DMMC-STAMP SoM CPLD (on-module, expert mode)

One advantage of realizing the JTAG switching inside by using the DMMC-STAMP SoM is that the JTAG chain can be configured on-the-fly. This allows to remove components that are e.g. not supported by a programming tool out of the chain. For example, this could be an RTM FPGA which is not supported by the programming tool used for the AMC FPGA. Since the JTAG member interface voltages are supplied to the DMMC-STAMP SoM via dedicated pins, the DMMC-STAMP SoM will also take over level shifting that is usually needed when JTAG chaining devices with different voltage levels.

Figure 34 gives an overview of the possible JTAG chain on an AMC managed by the DMMC-STAMP SoM:

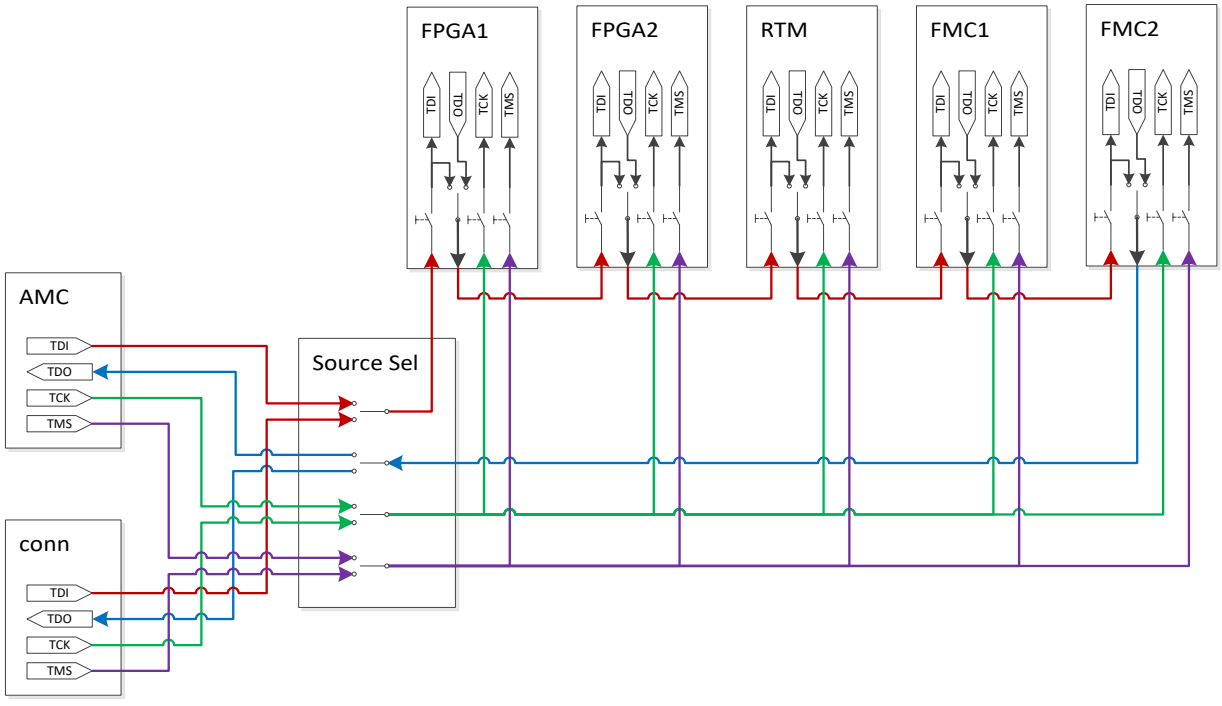


Figure 34: Reconfigurable JTAG chain (Source Sel = DMMC-STMAP SoM)

5.13 Out-of-Crate Mode

The DMMC-STAMP module supports an Out-of-Crate Mode, which allows the AMC and connected RTMs to operate independently of a MicroTCA environment: particularly without an MCH. This mode is intended for laboratory use (e.g., bring-up, hardware debugging) and low-level firmware development.

In a standard configuration, the DMMC-STAMP functions within a MicroTCA crate under the supervision of the MCH, where power sequencing, hot-swap control, and communication are handled via the backplane. However, it is often desirable to operate the module in a bench setup, without a crate or MCH. The Out-of-Crate Mode enables this by allowing the DMMC-STAMP to power up and function even when the usual management and backplane signals are absent. The DMMC-STAMP firmware automatically detects the absence of a crate environment by checking one of the presence pins at power-up: On the DMMC-STAMP, the PS0# pin (AMC detect) has a pull-up resistor. When the AMC is in a crate, this pin is forced to GND. This means a high level on PS0# will trigger out-of-crate detection, while a low level will enable normal (in-crate) operation.

For benchtop operation, both 3.3V MP and 12V PP should be activated simultaneously by an external power supply. A current limit of 150 mA for MP protects the AMC. PP should also be current-limited. Once the power system is up, a PP current limit of 2.5 A is a typical value (board- and design-dependent). Operating with 3.3V MP only allows some management debug but is usually not a suitable mode of operation for users. When in Out-of-Crate Mode, the firmware no longer waits for management commands from the MCH, and the board can be powered up. The handle (used as power switch) serves as an indicator for powering the board up or down. In this mode, connected RTMs will also be powered up or down accordingly.

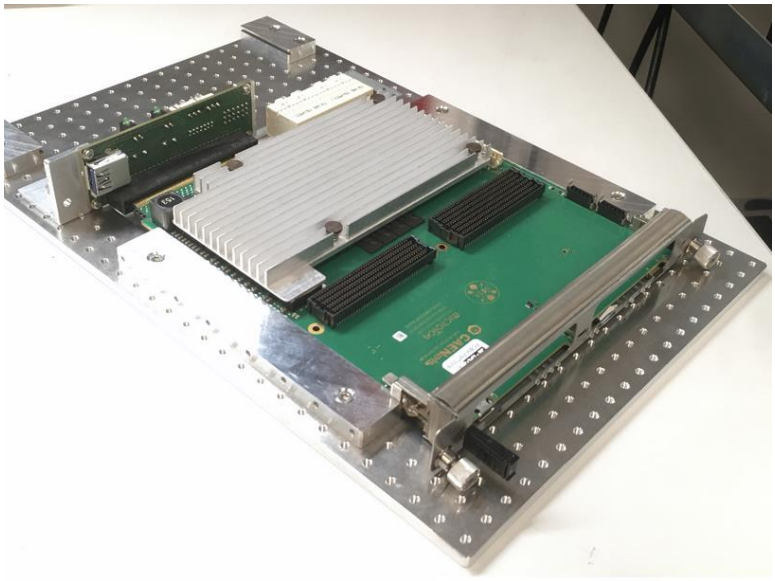


Figure 35: Bring-up mechanics and bring-up adapter (backplane dummy)

Prerequisites:

- A safe mechanical platform that prevents tilting of the board inside the AMC connector under power (doing so will result in fatal damage of the DMMC-STAMP and/or payload)
- External 12V and 3.3V supply
- The AMC_EN_N line of the Backplane is pulled low externally (e.g. via Jumper)
- Forced air cooling (e.g. using a fan) to prevent overheating of the payload

6. Development Tools

This chapter gives some information about useful development tools for the DMMC-STAMP SoM. The resources described here will be provided by the DESY MicroTCA Technology Lab as Open Source on GitHub, free of additional charge (on request) or on a commercial base.

6.1 Hardware Design Templates

As a starting point for a custom hardware development the DESY MicroTCA Technology Lab provides Altium Designer® templates for AMCs and RTMs. These templates can be seen as “empty” boards which are fully MicroTCA® standard compliant. This means they include all components that are mandatory required by the MicroTCA® specification.

Both, the AMC and the RTM design template will be provided as a ZIP archived Altium Designer® template on request.

6.1.1 AMC Design Template

The AMC design template is based on the DMMC-STAMP SoM. All mandatory, highly recommended and useful components required by the MicroTCA® specification and described in the chapters above are already included with the schematics. In combination with the pre-programmed DMMC-STAMP SoM the AMC resulting out of this template implements all mandatory MicroTCA® management operations (e.g. hot-plug, LEDs, sensors) as well as power delivery. Shown in Figure 36 is an exception from the schematics of the AMC design template.

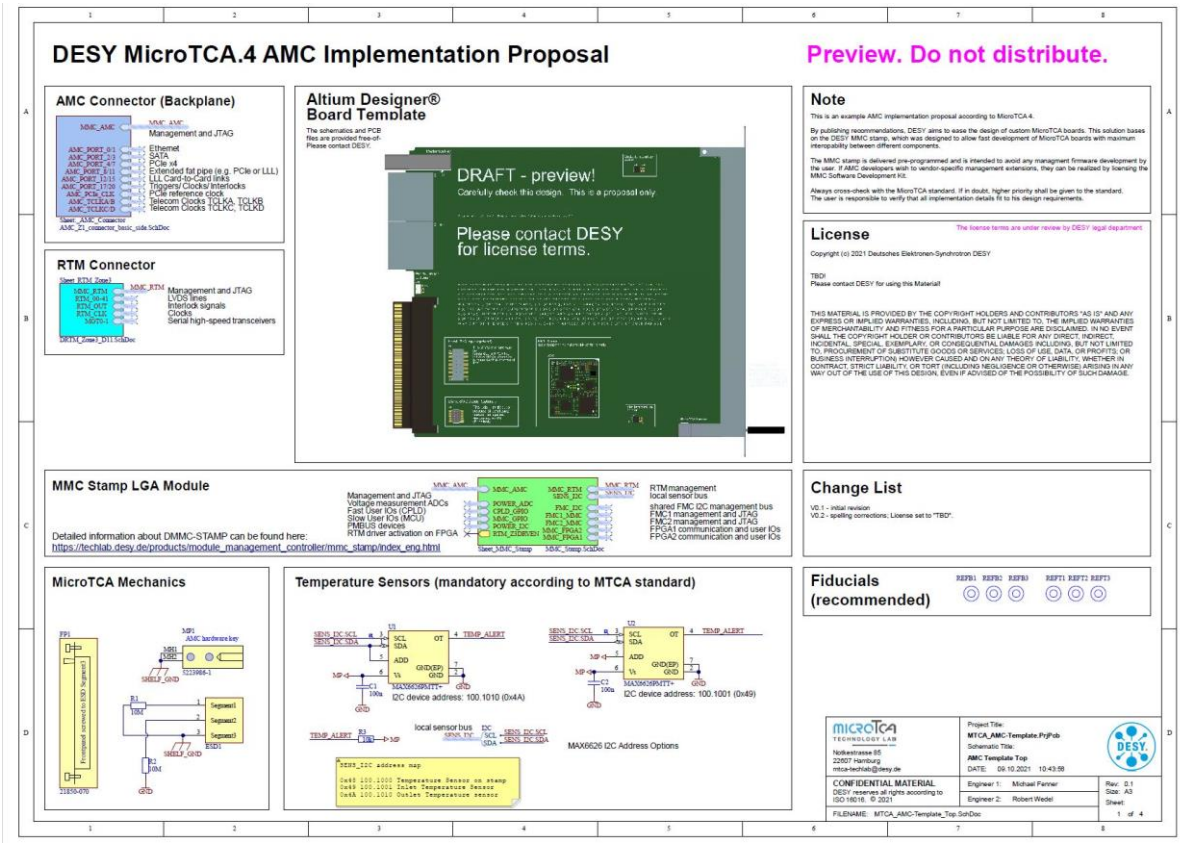


Figure 36: AMC design template (exception)

6.1.2 RTM Design Template

The RTM design template includes all mandatory, useful and highly recommended components required by the MicroTCA® specification (e.g. FRU, mechanical key, IO expander) with the schematics. The MicroTCA® specification leaves some freedom for the AMC-RTM interface implementation which is usually vendor specific. For increasing the interoperability DESY has introduced different Zone 3 which are provided as a recommendation (see link) Shown in Figure 36 is an exception from the schematics of the RTM design template.

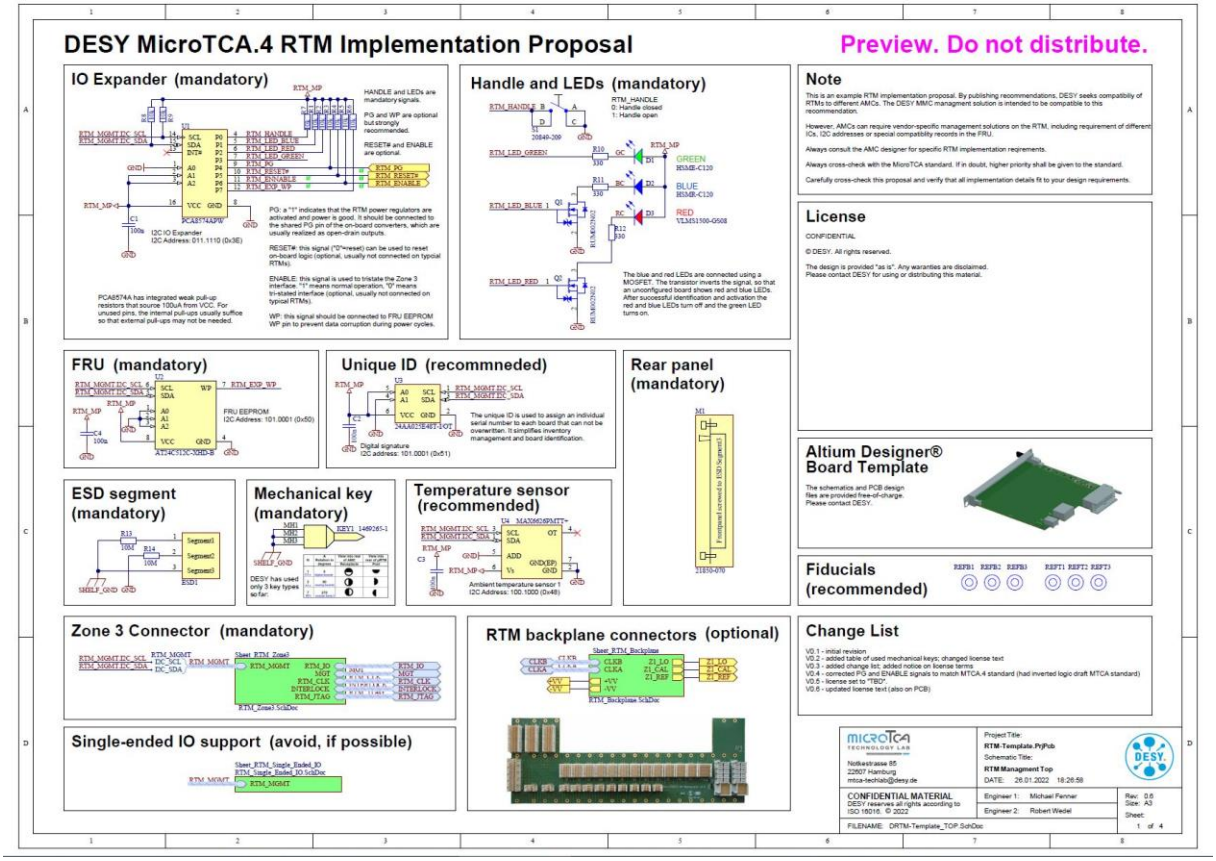


Figure 37: RTM design template (exception)

6.2 DMMC-SDK (Software Development Kit)

The pre-programmed DMMC-STAMP SoM firmware provides all the functionality which is mandatorily required by the MicroTCA® specification for operating an AMC. Additional features to implement the board specific AMC functionality (e.g. hosting FMC cards, implementing specific sensors, using custom IPMI commands) need adjustments and extensions of the pre-installed firmware.

To implement an AMC hardware-specific firmware for the DMMC-STAMP SoM while minimizing the need for a generic firmware programming the MicroTCA Technology Lab provides the DMMC-Software-Development-Kit (DMMC-SDK) on a commercial base. Table 27 provides a list of features with a separation between pre-installed firmware (generic) and DMMC-SDK (board specific).

Feature	pre-installed MMC firmware	DMMC-SDK
Full IPMI handling (MCH communication, LEDs, power, FRU)	yes	yes
Event handling (e.g. over-temperature)	yes	yes
Custom FRU read/write (AMC/RTM)	yes	yes
Serial-over-IPMI (remote access of MMC console)	yes	yes
USB virtual COM port for MMC and FPGAs/SoCs	yes	yes
Basic FPGA/SoC control (PROG, INIT, DONE, RESET)	yes	yes
Re-configurable JTAG chain management	yes	yes
RTM control (power, over-current)	yes	yes
Basic sensor control (temperature, voltages, FPGA done)	yes	yes
DMMC-STAMP SoM in-system-update via HPM.1	yes	yes
Serial-over-IPMI (remote access of FPGA/SoC console)	no	yes
Control of up to two FMC modules	no	yes
Custom IPMI commands	no	yes
Support of additional sensors	no	yes
User FPGA/SoC in-system-update via HPM.1	no	yes
DC/DC converter control	no	yes
Complete AMC power management via PMBUS™	no	yes
MTCA system information forwarding to user-FPGA/SoC (MMC Mailbox)	no	yes
User-specific DMMC-STAMP SoM GPIO pin control	no	yes
M-LVDS trigger or Interlock forwarding to RTM	no	yes

Table 27: Differences DMMC-STAMP SoM pre-installed firmware and DMMC-SDK

The DMMC-SDK grants access to the DESY MMC software library that provides a common code base shared between different MMC implementations. Included with the software library is a various set of device and interface drivers (e.g. sensors, PMBUS™), functionality needed for implementing custom MMC and IPMI commands as well as useful features to be used with modern SoC based board architectures. The code base has been maintained for more than 10 years and is in a reliable 24/7 operation at several facilities at DESY and around the world.

The features of the DMMC-SDK are listed here:

- Straight-forward development: CMake based, VS Code® integrated and CI build via Docker®
- High-level API for AMC board-specific MMC developments
- Clear separation between generic MMC library and board specific implementation
- Support for additional, board-specific IPMI sensors
- In-system firmware upgrade of payload FPGAs/SoCs using the PICMG HPM.1 specification
- Implementation of custom IPMI commands
- Implementation of user-specific DMMC-STAMP SoM GPIO pin control
- MTCA system information forwarding to FPGAs/SoCs

Shown in Figure 38 is the simplified structure of the DMMC-SDK.

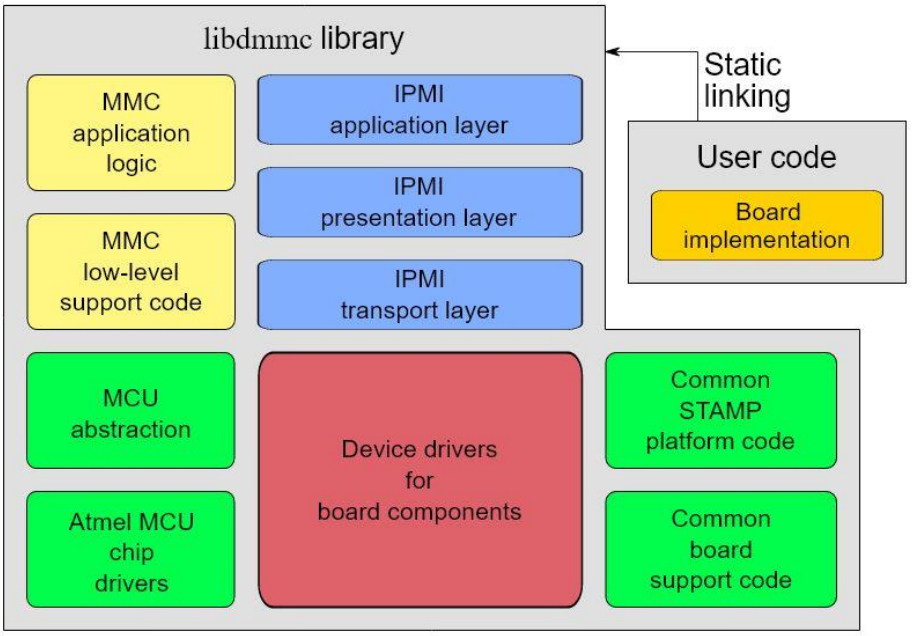


Figure 38: DMMC-SDK code structure (simplified)

Included with the DMMC-SDK (pre-compiled binaries) are some board specific example implementations for the DMMC-STAMP SoM firmware (source code):

- dmmc-stamp (pre-installed firmware of the DMMC-STAMP SoM)
- dmmc-stamp-breakout (MMC firmware of the DMMC-STAMP-BoB)
- damc-fmc2zup (MMC firmware of the DAMC-FMC2ZUP FMC+ Carrier)

These example projects can be used as reference for an own hardware specific AMC MMC implementation.

6.3 DMMC-STAMP Breakout Board

Evaluating the functionality of the DMMC-STAMP SoM, its pre-installed firmware, or debugging a custom development project based on the DMMC-SDK can be carried out conveniently using the DMMC-STAMP Breakout Board (DMMC-STAMP-BoB, see Figure 39). The DMMC-STAMP-BoB features an AMC form factor and is designed to simplify hardware testing, firmware evaluation, and software development in both laboratory and prototyping environments. The board provides access to all LGA module pins through standard 2.54 mm pin headers, enabling direct connection for signal measurement, debugging, and integration with external hardware. This makes it an ideal platform for developers who need to evaluate interfaces, or perform early-stage validation of custom designs. In addition to the pin access, the DMMC-STAMP-BoB incorporates several key components that enhance its functionality. These include a PMBUS manager for power management and monitoring, a DC/DC converter for power delivery, a JTAG output for debugging and firmware flashing of “target boards”, and an example FLASH memory peripheral for testing IPMI firmware update functions. Together, these features provide a comprehensive environment for developing, testing, and validating custom applications using the DMMC-SDK.

By combining robust hardware access with integrated peripheral components, the DMMC-STAMP-BoB significantly reduces setup time and complexity, allowing developers to focus on system optimization and feature development rather than hardware configuration.

Figure 39 : DMMC-STAMP-Breakout-Board Rev. C

The DMMC-STAMP-BoB can be provided by DESY on request.

6.4 MMC Terminal (mmcterm)

The MMC Terminal (mmcterm) is a Python based console for the custom “serial over IPMB” protocol used by the DMMC-STAMP SoM. For DMMC-STAMP SoM based AMCs that are part of a running MicroTCA® system, mmcterm can be used for accessing the MMC console from a host computer connected to the same network (without establishing a serial connection). Depending on the AMCs payload and equipped with an appropriate custom MMC firmware it is also possible to access the UART of an embedded processing system (e.g. Xilinx Zynq PS) by specifying the console channel.

Given below is the help print of mmcterm. For downloading the tool, itself visit the MicroTCA Technology Lab on GitHub.

```
mmcterm—help
usage: mmcterm [-h] [-v] [-c CHANNEL] [-l] [-d] [-i] mch_addr mmc_addr
DESY MMC Serial over IPMB console
positional arguments:
  mch_addr              IP address or hostname of MCH
  mmc_addr              IPMB-L address of MMC

optional arguments:
h, --help            show this help message and exit
v, --version         show program’s version number and exit
c CHANNEL, --channel CHANNEL
console channel (default 0)
l, --list            list available channels
d, --debug           pyipmi debug mode
i, --ipmitool        make pyipmi use ipmitool instead of native rmcp
```

6.5 FRU Generator YAML (frugy)

The FRU generator YAML (frugy) is a Python based tool that generates a binary FRU image out of an appropriate YAML configuration file. This binary can be downloaded directly into an on-board FRU EEPROM by using ipmitool. Frugy is compliant to the IPMI Field Replaceable Unit (FRU) standard. The DMMC-STAMP SoM supports writing the FRU EEPROM of AMCs, RTMs and FMCs by using ipmitool.

For debug and evaluation purposes frugy also supports the other way around. A binary FRU image downloaded via ipmitool from an on-board EEPROM can be parsed into a YAML configuration file or dumped to the stdout.

Given below is the help print of frugy. For downloading the tool, itself visit the MicroTCA Technology Lab on GitHub.

```
$ frugy -help
usage: frugy [-h] [--version] [-o OUTPUT] [-w] [-r] [-d] [-e EEPROM_SIZE]
[-s SET] [-t] [-b] [-c] [-l [LIST]] [-v VERBOSITY] [srcfile]

FRU Generator YAML

positional arguments:
srcfile               Source file for reading

optional arguments:
h, --help             show this help message and exit
version               show program's version number and exit
o OUTPUT, --output OUTPUT
output file (derived from input file if not set)
w, --write            FRU write mode (convert YAML to FRU image), default
r, --read             FRU read mode (convert FRU image to YAML)
d, --dump             dump FRU information to stdout (same as -r -o -)
e EEPROM_SIZE, --eeprom-size EEPROM_SIZE
pad FRU image to match EEPROM size in bytes (only valid in write mode)
s SET, --set SET      set FRU record field to a value (only valid in write
mode)
t, --timestamp        set BoardInfo.mfg_date_time timestamp to current UTC
time (only valid in write mode)
b, --broken           enable workaround to parse Opal Kelly EEPROMs
c, --ignore-checksum-errors
ignore checksum errors when parsing a FRU image
l [LIST], --list [LIST]
list supported FRU records or schema of specified record
v VERBOSITY, --verbosity VERBOSITY
set verbosity (0=quiet, 1=info, 2=debug)
```

Figure 40 shows the FRU configuration file of a DAMC-FMC2ZUP AMC as example.

```yaml
# created with frugy 0.2.3 from "damc-fmc2zup.bin"

BoardInfo:
  manufacturer: DESY/CAEN ELS
  product_name: DAMC-FMC2ZUP-11EG
  serial_number: 21Y01W0000
  part_number: DAMCFMC2ZUP1
  fru_file_id: fru_damc-fmc2zup.bin

ProductInfo:
  manufacturer: DESY/CAEN ELS
  product_name: DAMC-FMC2ZUP-11EG
  part_number: DAMCFMC2ZUP1
  version: revB
  serial_number: 21Y01W0000
  asset_tag: none
  fru_file_id: fru_damc-fmc2zup.bin

MultirecordArea:
- type: ModuleCurrentRequirements
  current_draw: 6.5

- type: PointToPointConnectivity
  guids:
  - 2d776f4c-616c-6574-6e63-79206c696e6b
  record_type: amc_module
  channel_descriptors:
  - [4, 5, 6, 7]
  - [8, 9, 10, 11]
  - [0]
  - [1]
  - [2]
  - [3]
  - [12]
  - [13]
  - [14]
  - [15]
  link_descriptors:
```

Figure 40: DAMC-FMC2ZUP YAML configuration file (abbreviated)

6.6 HPM File Generator (bin2hpm)

The HPM file generator bin2hpm is a Python-based tool that creates HPM.1 (Hardware Platform Management IPM Controller Firmware Upgrade Specification of the PICMG) firmware update files out of input binaries. HPM files can be downloaded to the DMMC-STAMP SoM by using the ipmitool command “hpm upgrade”. In MicroTCA® this method allows to update various components (MMC, CPLDs, FPGAs) in the field without using proprietary software or hardware tools like JTAG programmers and without having direct access to the hardware.

The bin2hpm tool is intended to be used with DMMC-STAMP SoM based AMCs, such as the DAMC-FMC2ZUP or the DAMC-FMC1Z7IO. The features of the bin2hpm tool are listed here:

- converts binary- to firmware update files according to the HPM.1 specification
- creates a sequence of two HPM.1 actions: prepare component and upload image
- embeds metadata according to HPM.1 spec:
    - IANA manufacturer / product ID
    - target device / component

    - version information
    - auxiliary metadata
- parses Xilinx bitstream file (optional)
- performs RLE compression (optional / DESY MMC proprietary)

By default, the Xilinx bitfile mode is determined from the input file name. If the input file ends on ".bit" the bitfile mode is selected. The bitfile mode can be also forced (-b) or suppressed (-n) independent of file name.

All MMC firmware updates supplied by DESY are based on HPM.1 file. In addition, bin2hpm can be used by the AMC board designers to package and deploy custom FPGA/SoC images files for their users.

Given below is the help print of the bin2hpm tool. For downloading the tool tool itself visit the MicroTCA Technology Lab on GitHub.

```
$ bin2hpm—help
usage: bin2hpm [-h] [--version] [-o OUTFILE] [-v FILE_VERSION] [-a
AUXILLARY] [-c COMPONENT] [-d DEVICE] [-m MANUFACTURER] [-p PRODUCT] [-r]
[-s DESCRIPTION] [-q] [-b | -n] infile
HPM.1 update file converter
positional arguments:
infile                Input file
optional arguments:
h, --help            show this help message and exit
version             show program's version number and exit
o OUTFILE, --outfile OUTFILE
output file (derived from input file if not set)
v FILE_VERSION, --file-version FILE_VERSION
file version information (format major.minor, e.g. 1.2)
a AUXILLARY, --auxillary AUXILLARY
additional metadata, hex format, 4 bytes
c COMPONENT, --component COMPONENT
HPM component ID (default 1)
d DEVICE, --device DEVICE
HPM device ID (hex, default 0)
m MANUFACTURER, --manufacturer MANUFACTURER
IANA manufacturer ID (hex, 6 bytes max)
p PRODUCT, --product PRODUCT
IANA product ID (hex, 4 bytes max)
r, --compress        Enable RLE compression (requires DESY MMC)
s DESCRIPTION, --description DESCRIPTION
Additional description string (max. 21 chars)
q, --quiet           Quiet mode
b, --bitfile         Force bitfile mode
n, --binfile         Force binfile mode
```

6.7 LGA Pinout

Pin Number	Pin Name
A1	MCU_GPIO1
A10	SNS_SCL
A11	GND
A12	AMC_PS1_N
A2	VCC+12V_PP
A3	MCU_GPIO5
A4	VCC+3V3_MMC
A5	PM_ADC0
A6	PM_ADC1
A7	PM_ADC2
A8	PM_ADC3
A9	SNS_SDA
B1	USB_DM
B10	RST_B
B11	AMC_PS0_N
B12	AMC_GA0
B2	PGND
B3	MCU_GPIO8
B4	GND
B5	MTCA_LED_RED
B6	MTCA_LED_GREEN
B7	MTCA_LED_BLUE
B8	SWD_DIO
B9	SWD_CLK
C1	USB_DP
C10	FMC1_PGM2C
C11	AMC_GA1
C12	AMC_GA2
C2	PGND
C3	MCU_GPIO0
C4	MCU_GPIO3
C5	PM_ADC4
C6	PM_ADC5
C7	PM_ADC6
C8	PM_ADC7
C9	FMC1_PGC2M

Pin Number	Pin Name
D1	USB_VTG
D10	FMC2_PGM2C
D11	MTCA_HANDLE
D12	AMC_EN_N
D2	PGND
D3	CPLD_GPIO3
D4	MCU_GPIO2
D5	FMC2_TMS
D6	FMC2_TCK
D7	FMC2_TDO
D8	FMC2_TDI
D9	FMC2_PGC2M
E1	FPGA2.GPIO5
E10	FPGA1_TXD
E11	AMC_SCL
E12	AMC_SDA
E2	FPGA2.GPIO6
E3	FPGA2.GPIO7
E4	FPGA2.GPIO8
E9	FPGA1_RXD
F1	FPGA2.GPIO2
F10	FPGA1_PROGN
F11	FMC_SCL
F12	FMC_SDA
F2	FPGA2.GPIO3
F3	FPGA2.GPIO4
F4	FPGA2_RXD
F9	VCC+3V3_MP
G1	FPGA2.GPIO1
G10	FPGA1_INITN
G11	RTM_SCL
G12	RTM_SDA
G2	FPGA2_DONE
G3	FPGA2_RST
G4	FPGA2_TXD
G9	RTM_PS_N

Pin Number	Pin Name
H1	FPGA2.GPIO9
H10	FPGA1_DONE
H11	FPGA1.GPIO5
H12	FPGA1.GPIO6
H2	FPGA2_PROGN
H3	FPGA2_INITN
H4	GND
H9	MCU_GPIO6
I1	FPGA2.SCK
I10	FPGA1_RST
I11	FPGA1.GPIO3
I12	FPGA1.GPIO4
I2	FPGA2.CSMUX
I3	FPGA2.DRVEN
I4	FPGA2.CS
I9	MCU_GPIO7
J1	FPGA2.MISO
J10	PMBUS_SCL
J11	GND
J12	VCC_12V_RTM
J2	FPGA2.MOSI
J3	FMC1_TCK
J4	FMC1_TMS
J9	FPGA1.CSMUX
K1	FPGA2_JTAG.TDO
K10	FPGA1.DRVEN
K11	FPGA1.SCK
K12	VCC_3V3MP_RTM
K2	FPGA2_JTAG.TDI
K3	FMC1_TDI
K4	FMC1_TDO
K5	FMC1_PRSN
K6	FMC2_PRSN
K7	FPGA1.MISO
K8	FPGA1.MOSI
K9	PMBUS_SDA

Pin Number	Pin Name
L1	VCC_FPGA2
L10	FPGA1_JTAG.TDO
L11	FPGA1.GPIO9
L12	FPGA1.CS
L2	FPGA2_JTAG.TCK
L3	FPGA2_JTAG.TMS
L4	FPGA1.GPIO8
L5	FPGA1.GPIO7
L6	VCC_FPGA1
L7	FPGA1.GPIO1
L8	FPGA1.GPIO2
L9	FPGA1_JTAG.TDI
M1	CON_ TDO
M10	FPGA1_JTAG.TMS
M11	GND
M12	VCC_12VPP_IN
M2	CON_ TDI
M3	CON_ TCK
M4	CON_TMS
M5	AMC_TDO
M6	AMC_TDI
M7	AMC_TCK
M8	AMC_TMS
M9	FPGA1_JTAG.TCK
N1	MCU_GPIO0
N10	RTM_TMS
N11	RTM_Z3DRVEN
N12	MCU_GPIO4
N2	CPLD_GPIO0
N3	GND
N4	CPLD_GPIO2
N5	CPLD_GPIO1
N6	GND
N7	RTM_TDO
N8	RTM_TDI
N9	RTM_TCK

6.8 Abbreviated Schematics

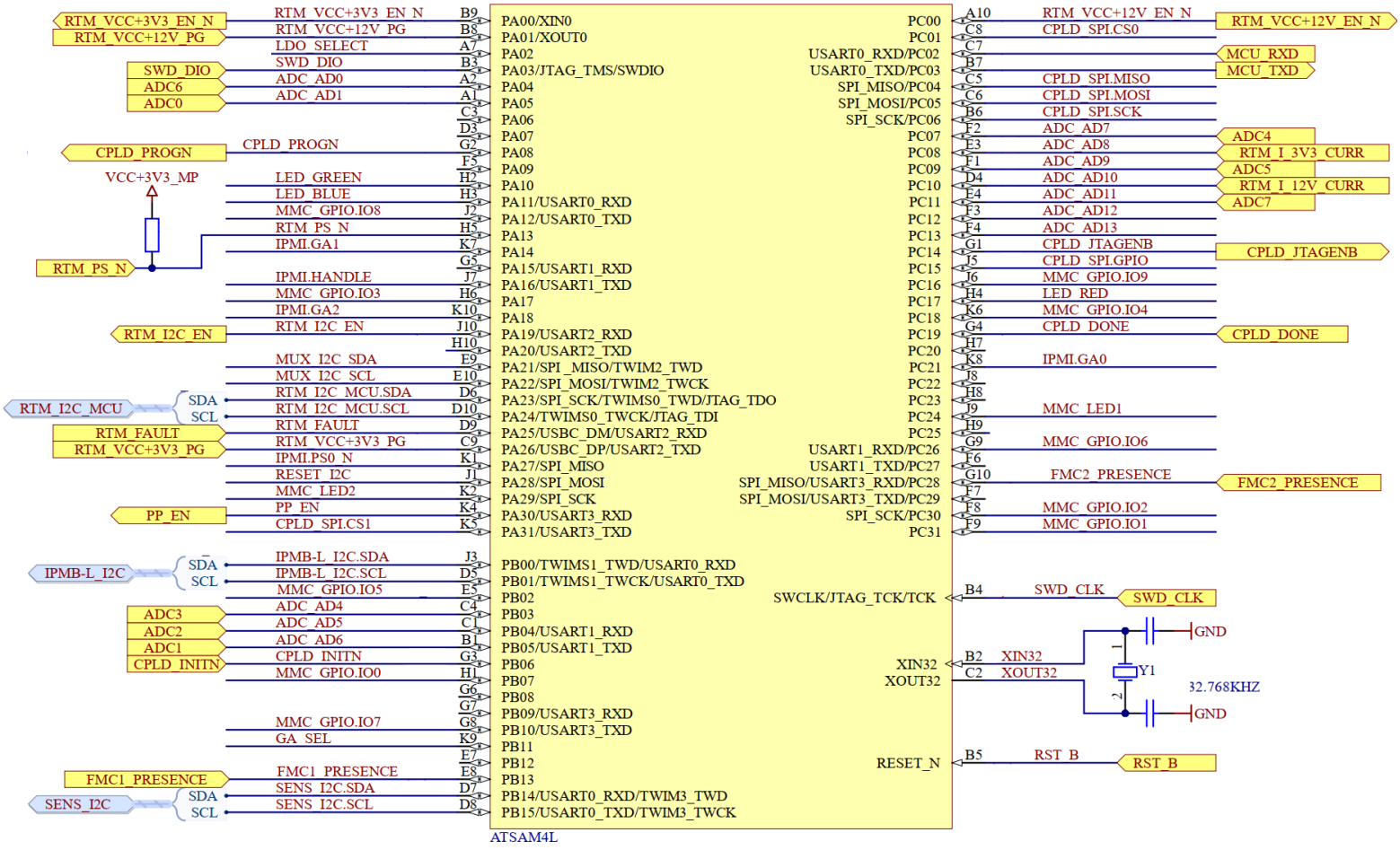


Figure 41: MCU pin-out

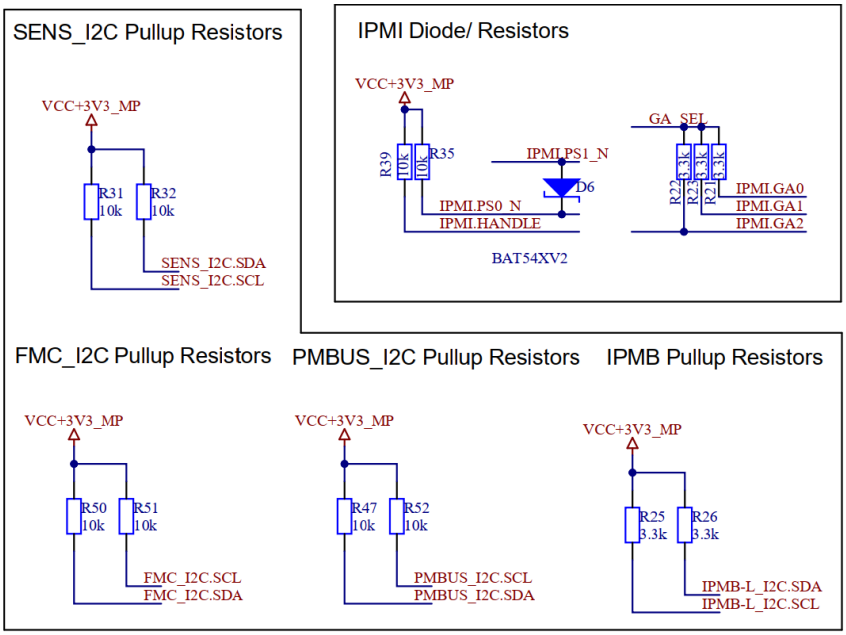


Figure 42: DMMC-STAMP pull-ups

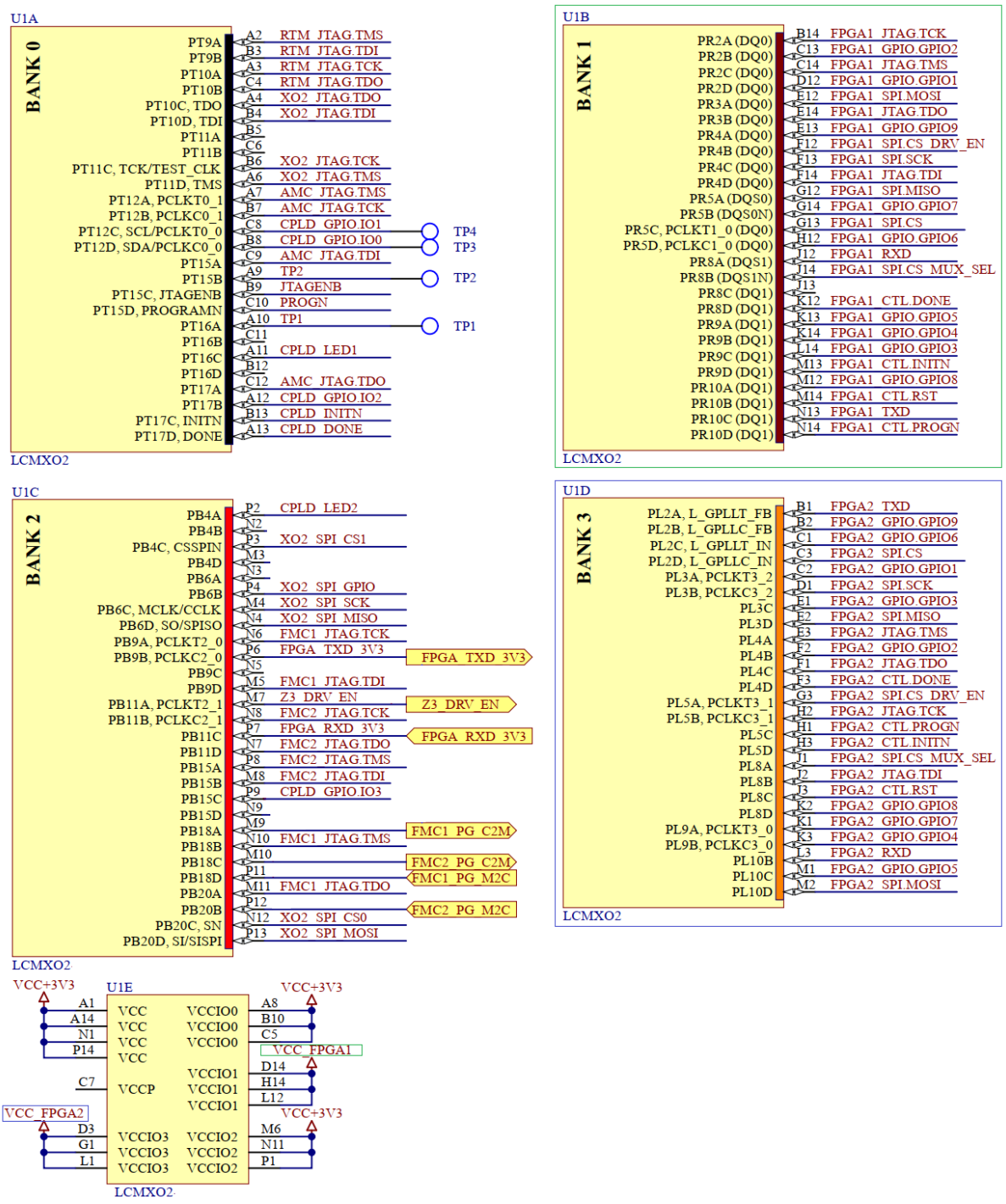


Figure 43: CPLD pin-out

7. References

Document	Details
MicroTCA 4.0 Specification	Advanced Mezzanine Card Base Specification
MicroTCA 4.1 Specification	MicroTCA Enhancements for Rear I/O and Precision Timing
FMC Specification	ANSI/VITA 57.1 FPGA Mezzanine Card (FMC) Standard
ATSAM4L Datasheet	https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/Atmel-42023-ARM-Microcontroller-ATSAM4L-Low-Power-LCD_Datasheet.pdf
MachXO2 Datasheet	https://www.latticesemi.com/~/media/LatticeSemi/Documents/DataSheets/MachXO23/MachXO2FamilyDataSheet.pdf

Imprint

Deutsches Elektronen-Synchrotron DESY
Notkestraße 85
D-22607 Hamburg

Contact

support-mtca@desy.de

AI tools

AI tools have been used for writing and correction assistance.