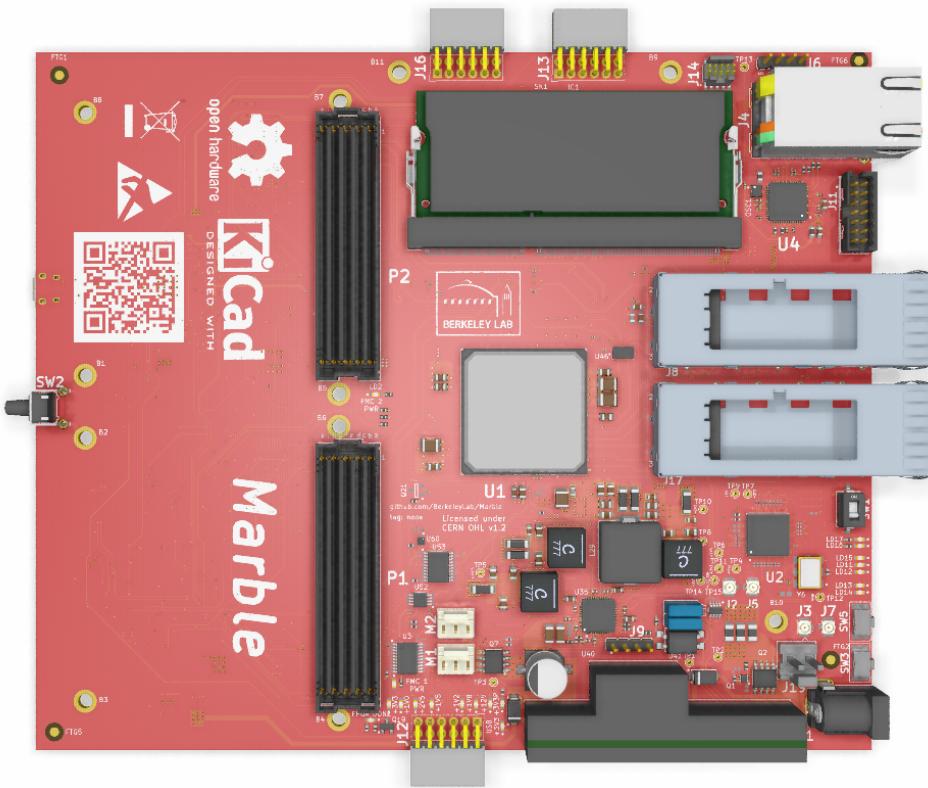


# Marble – User Guide

v1.0 2021

May 19, 2021



## Introduction

Marble is a fully open source dual FMC carrier board designed for the Accelerator Technologies Group of the Engineering Division of Lawrence Berkeley National Laboratory. This document presents the technical documentation of the Marble module divided into individual functional sections. Design files are made in KiCad and are licensed under the CERN OHL v. 1.2.

# Contents

|  |           |
|--|-----------|
| <b>1 Overview</b>                          | <b>3</b>  |
| 1.1 FPGA reset . . . . .                   | 7         |
| 1.2 JTAG . . . . .                         | 7         |
| 1.2.1 External JTAG . . . . .              | 7         |
| 1.2.2 Internal USB-JTAG . . . . .          | 7         |
| 1.2.3 Self-JTAG . . . . .                  | 8         |
| 1.3 LEDs . . . . .                         | 8         |
| 1.4 FPGA Programming . . . . .             | 8         |
| 1.4.1 Internal JTAG . . . . .              | 8         |
| 1.4.2 External JTAG . . . . .              | 9         |
| <b>2 MMC</b>                               | <b>9</b>  |
| 2.1 Programing . . . . .                   | 10        |
| 2.2 LEDs . . . . .                         | 12        |
| 2.3 I2C Tree . . . . .                     | 12        |
| <b>3 SO-DIMM</b>                           | <b>13</b> |
| <b>4 GTH Routing</b>                       | <b>13</b> |
| <b>5 Clocking</b>                          | <b>14</b> |
| 5.1 White Rabbit clock generator . . . . . | 15        |
| <b>6 Pmod connectors</b>                   | <b>15</b> |
| <b>7 QSFP</b>                              | <b>16</b> |
| <b>8 Ethernet</b>                          | <b>17</b> |
| <b>9 FMC</b>                               | <b>19</b> |
| <b>10 USB-UART</b>                         | <b>20</b> |
| 10.1 FPGA UART . . . . .                   | 20        |
| 10.2 MMC UART . . . . .                    | 20        |
| <b>11 Power</b>                            | <b>20</b> |
| 11.1 Fan controller . . . . .              | 22        |
| <b>12 Mechanical dimensions</b>            | <b>24</b> |
| <b>13 Appendix</b>                         | <b>27</b> |
| 13.1 Power supply test points . . . . .    | 27        |

# 1 Overview

Design files are open source and can be downloaded from GitHub:  
<https://github.com/BerkeleyLab/Marble>

Marble is a dual FMC carrier module based on an Kintex-7 FPGA. The block diagram of the module is shown in figure 2.

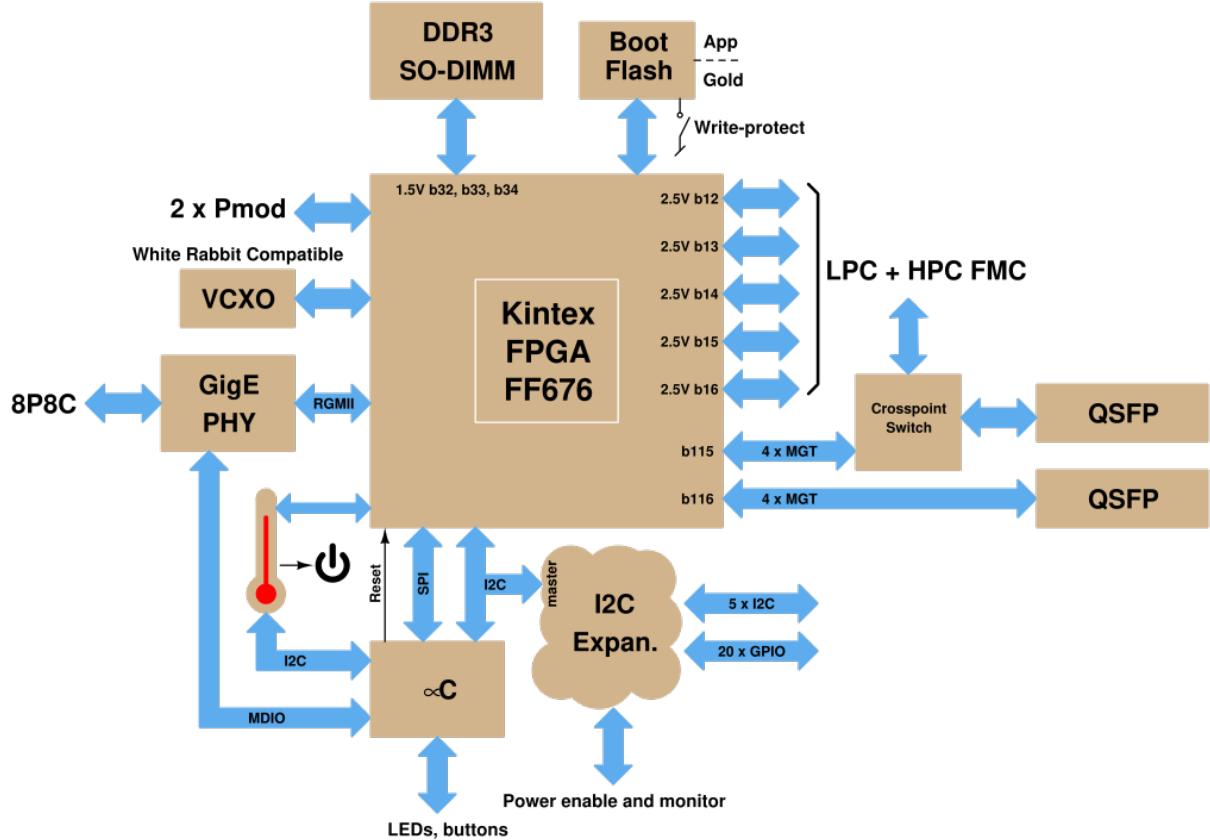


Figure 1: Marble block diagram.

 Square callout references a component on the back side of the board  Round callout references a component on the front side of the board

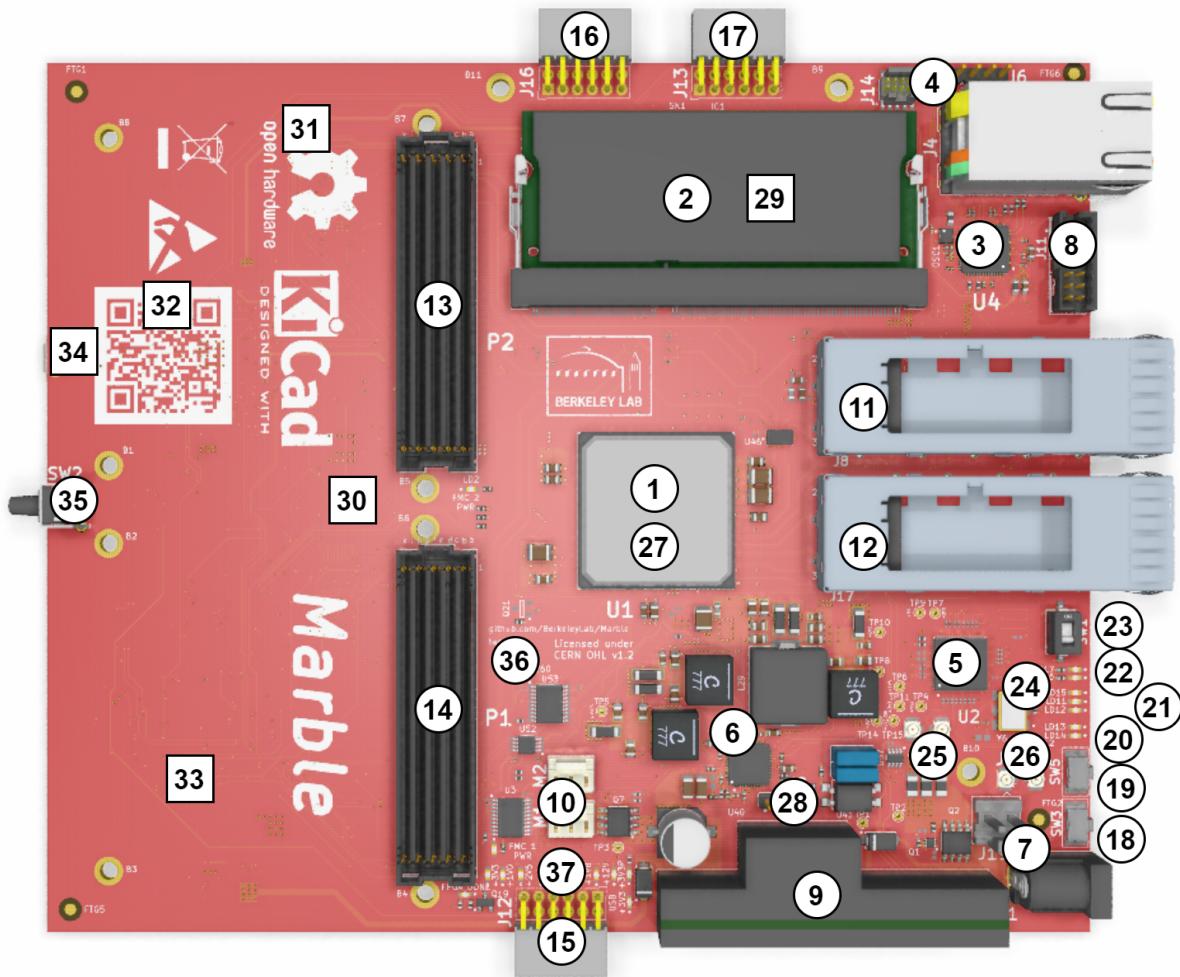


Figure 2: Marble board components.

| Callout | Reference Designator | Component Description                  | Notes                        | Schematic Page |
|---------|----------------------|--|------------------------------|----------------|
| 1       | U1                   | FPGA Xilinx Kintex-7                   | XC7K160T-2FFG676C            |                |
| 2       | SK1                  | DD3 SO-DIMM memory module              | VR7PU286458FBAMJT            |                |
| 3       | U4                   | 10/100/1000 Ethernet PHY               | 88E1512-A0-NNP2I000          |                |
| 4       | J14, J6              | Microcontroller programming connectors |                              |                |
| 5       | U2                   | 8x8 Clock Crosspoint Switch            | ADN4600ACPZ                  |                |
| 6       | U35                  | Power Management                       | XRP7724ILBTR-F               |                |
| 7       | J1, J19              | 12V power input                        | FC68148(DC-10A),<br>641119-2 |                |
| 8       | J11                  | FPGA JTAG connector                    | 87831-1420                   |                |
| 9       | U40                  | PoE module                             | AG5300                       |                |

|    |                 |   |                            |
|----|-----------------|---|----------------------------|
| 10 | M1, M2          | 3-pin PC fan connectors                                 | SWR25X-NRTC-S03-ST-BA      |
| 11 | J8              | QSFP connector  | QSFP8-038-01-L-D-RA1       |
| 12 | J17             | QSFP connector  | QSFP8-038-01-L-D-RA1       |
| 13 | P2              | FMC HPC connector                                       | ASP-134486-01              |
| 14 | P1              | FMC HPC connector                                       | ASP-134486-01              |
| 15 | J12             | PMOD connected to FPGA                                  | PPTC062LJBN-RC             |
| 16 | J16             | PMOD connected to microcontroller                       | PPTC062LJBN-RC             |
| 17 | J13             | PMOD connected to FPGA                                  | PPTC062LJBN-RC             |
| 18 | SW3             | User button connected to microcontroller                | KSS241GLFS                 |
| 19 | SW5             | FPGA reset button                                       | KSS241GLFS                 |
| 20 | LD13, LD14      | User LEDs connected to shared I2C bus                   | KPH-1608CGCK               |
| 21 | LD11-12, LD15   | User LEDs connected to microcontroller                  | KPH-1608CGCK               |
| 22 | LD16, LD17      | User LEDs connected to FPGA                             | KPH-1608CGCK               |
| 23 | SW1             | Memory write protection switch                          | A6SN-1101                  |
| 24 | Y6              | 10-280MHz Clock generator                               | SI570                      |
| 25 | J2, J5          | External clock source input                             | U.FL                       |
| 26 | J3, J7          | External clock source input                             | U.FL                       |
| 27 | U44-U50         | GTX Transceivers Mux                                    | PI3DBS12212AZBSEX          |
| 28 | J9              | Power Management programming header                     | 0.1 inch male 4-pin header |
| 29 | Y1-Y3, U20      | Internal 125 MHz & 20MHz clock sources                  | CDCM61004RHBT              |
| 30 | U30             | FPGA SPI flash memory                                   | S25FL128SAGMFIR01          |
| 31 | U5              | I2C multiplexer   | TCA9548ARGER               |
| 32 | U23             | Double USB - UART bridge, USB-JTAG bridge for FPGA      | FT4232H-56Q                |
| 33 | U54             | Housekeeping Microcontroller                            | STM32F207VCTx              |
| 34 | J10             | Micro USB connector for U23                             | 10103594-0001LF            |
| 35 | SW2             | Not populated user button (physically connected to SW3) | SKHHLQA010                 |
| 36 | U60             | Over-voltage and Under-voltage Reset IC                 | TPS3703A7330DSERQ1         |
| 37 | LD4-10, LD18-19 | Power rails indicator LEDs                              | KPH-1608CGCK               |

Table 1

The board has the following functionalities and features:

1. Xilinx Kintex-7 FPGA XC7K160T-2FFG676C
2. Supports FPGA golden image
3. Housekeeping microcontroller (Module Management Controller) with UART console
4. DDR3 204-SODIMM memory module connector. The board supports up to 4 GB memory.
5. Two FMC HPC connectors, but not all signals are connected to the FPGA.
6. 1Gb Ethernet with PoE
7. Built-in clock generator that supports White Rabbit synchronization
8. Various input clock configurations
9. Two QSFP cages that support data transfer up to 40 Gb/s each
10. Built-in USB JTAG which works with OpenOCD

| FPGA Bank  | Bank Power Supply | Description                          |
|------------|-------------------|--------------------------------------|
| Bank 12 HR | +2.5V             | FMC2 LA 00-16 (plus SPI, Self JTAG)  |
| Bank 13 HR | +2.5V             | FMC2 HA                              |
| Bank 14 HR | +2.5V             | FMC2 LA 17-33 (plus config, Pmod)    |
| Bank 15 HR | +2.5V             | FMC1 LA 00-16 (plus I2C, UART, Pmod) |
| Bank 16 HR | +2.5V             | FMC1 LA 17-33 (plus RGMII)           |
| Bank 32 HP | +1.5V             | DDR3                                 |
| Bank 33 HP | +1.5V             | DDR3 (plus Pmod, White Rabbit)       |
| Bank 34 HP | +1.5V             | DDR3                                 |

Table 2

The S25FL128SAGMFIR01 configuration memory is connected to the FPGA chip. By default, the FPGA chip loads the configuration from the flash memory after correct power cycle. The module is equipped with a switch (fig 3) that blocks the programming of the configuration memory. When the switch is in the ON position, Write Protection is enabled. WP signal status can be read via I2C IO expander.

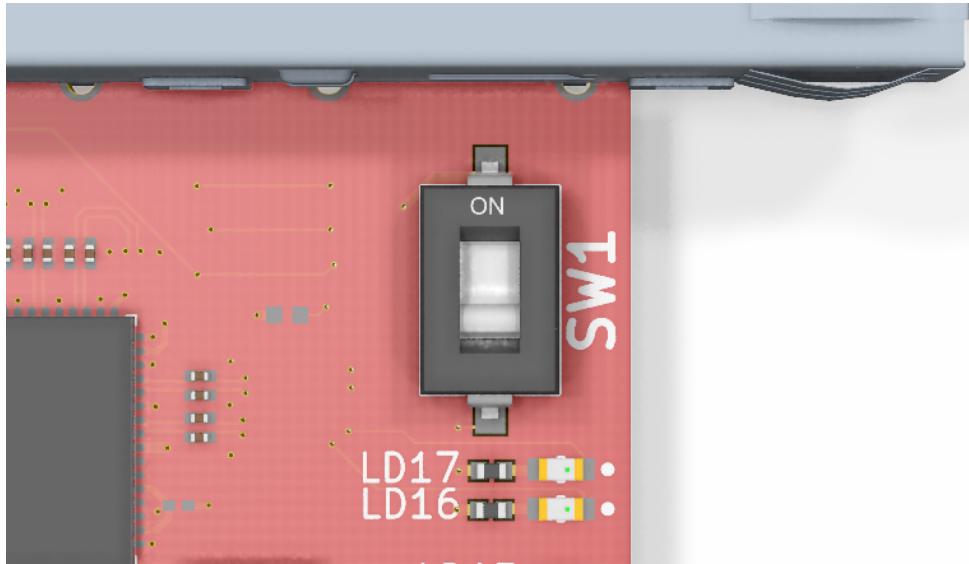


Figure 3: Memory write protection switch.

## 1.1 FPGA reset

During power system startup, the U60 chip keeps the PROGRAM\_B signal low. When a valid supply voltage is detected on the last power-up sequence, U60 changes state on the PROGRAM\_B signal which causes the configuration file to be loaded from flash memory.

A local operator can reset the FPGA manually using the SW5 button. Additionally, the MMC can reset the FPGA.

## 1.2 JTAG

There are 3 sources of JTAG for FPGA:

1. external JTAG (highest priority)
2. internal USB-JTAG (middle priority)
3. self JTAG (lowest priority)

### 1.2.1 External JTAG

When the external JTAG is connected to J11, GNDDetect signal from the connector switches the multiplexer to pass JTAG signals from the connector to the FPGA. After unplugging cable, the GNDDetect signal is not present and the multiplexer connects internal USB-JTAG to FPGA.

When external JTAG is connected, any other JTAG sources are not available.

### 1.2.2 Internal USB-JTAG

Internal USB-JTAG is done by using the first data channel of FT4232, which can work as a JTAG. When the micro USB cable is connected, +5V from USB bus switches the multiplexer to pass data from FT4232 to FPGA.

### 1.2.3 Self-JTAG

Internal self-JTAG can be used only when USB cable and external JTAG are not connected. In this configuration FPGA JTAG signals are connected to FPGA Bank 12:

| Signal name | Self JTAG signal | FPGA pin |
|-------------|------------------|----------|
| JTAG TDI    | Self_FPGA_TDI    | IO_L10P  |
| JTAG TCK    | Self_FPGA_TCK    | IO_L10N  |
| JTAG TMS    | Self_FPGA_TMS    | IO_L20P  |
| JTAG TDO    | Self_FPGA_TDO    | IO_L20N  |

Table 3

## 1.3 LEDs

Two general purpose LEDs are connected to the FPGA chip:

1. LD16 - connected to pin IO\_L18P\_33
2. LD17 - connected to pin IO\_25\_33

## 1.4 FPGA Programming

Vivado reference design with a constraint file can be found here:

| <https://github.com/BerkeleyLab/bedrock>

See its projects/test\_marble\_family directory.

### 1.4.1 Internal JTAG

Download the latest version of the FPGA testing code from GitHub:

| <https://github.com/BerkeleyLab/Bedrock>

| Before testing the FPGA, it is recommended to set up the current limit to 2A on the lab power supply.

Program the FPGA using the following steps:

1. Plug micro USB cable
2. Go to the folder **Bedrock/projects/test\_marble\_family/**
3. Open command terminal and run command:

```
$ mutill usb
```

4. After the successful programming, LEDs LD16 and LD17 should blink alternately.

### 1.4.2 External JTAG

Programming FPGA using Vivado and Digilent JTAG HS3 connected to J11:

1. Run Vivado
2. Go to `Flow > Open Hardware Manager` and then `Tools > Auto Connect`
3. Click `Tools > Program Device > xc7k160t_0` to open the programming window.
4. Choose the *bitstream file* and click `Program`
5. After the successful programming, LEDs LD16 and LD17 should blink alternately.

## 2 MMC

Module Management Controller (MMC) is based on STM32F207 microcontroller and provides housekeeping functions such as:

1. Simple UART console over USB-UART bridge to control all functions
2. Monitoring voltage, current consumption and warning signals on power rails
3. Temperature monitoring at several locations
4. Controlling and monitoring fans
5. Configuring clock multiplexer
6. Configuring MGT switches
7. Resetting FPGA and controlling booting
8. Programming Power Management Controller
9. Controlling FMC power delivery and the presence of the cards
10. Ensuring communication over MDIO with Ethernet PHY
11. Ensuring communication over SPI with FPGA

Most of the above mentioned functionality is available from the UART console level. Using a simple terminal user can control MMC functions by sending appropriate commands by selecting an item from menu (Fig. 4).

Build based on git commit GIT\_REV  
Menu:  
0) Loopback  
1) MDIO/PHY  
2) I2C monitor  
3) Status counters  
4) GPIO control  
5) Reset FPGA  
6) Push IP&MAC  
7) MAX6639  
8) LM75\_0  
9) LM75\_1  
a) I2C scan all ports  
b) Config ADN4600  
c) INA219 Main power supply  
d) MGT MUX - switch to QSFP 2  
e) PM bus display  
f) XRP7724 flash  
g) XRP7724 go  
h) XRP7724 hex input  
i) timer check/cal  
j) save IP to EEPROM  
k) read IP from EEPROM  
Single-character actions, ? for menu

Figure 4: UART console menu

## 2.1 Programming

MMC programing can be done by using external tools such as STM Nucleo-SWD programmer, SEGGER J-LINK Mini (Fig. 5, Fig. 6).

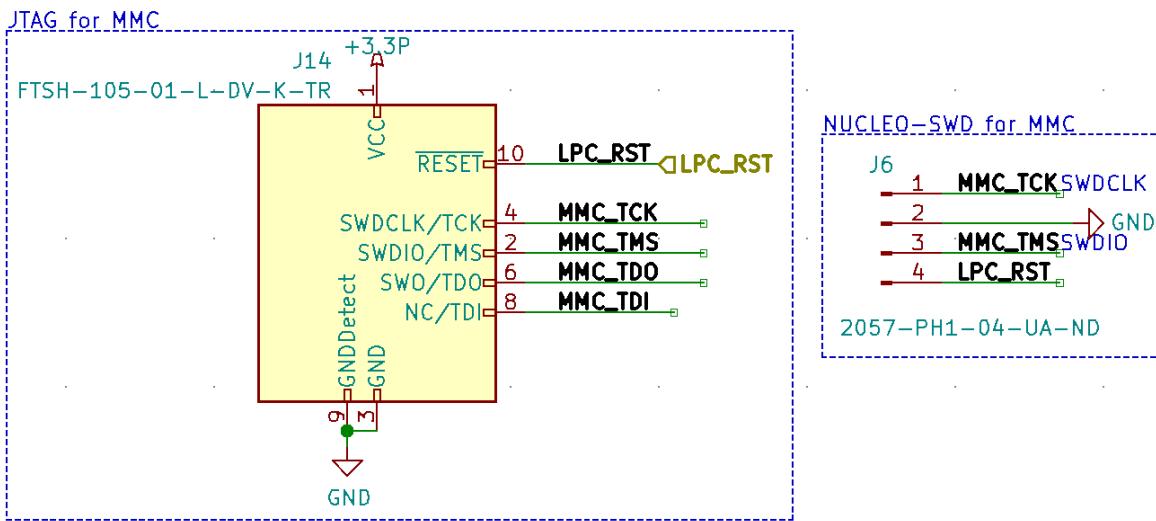


Figure 5: MMC JTAG and SWD interfaces

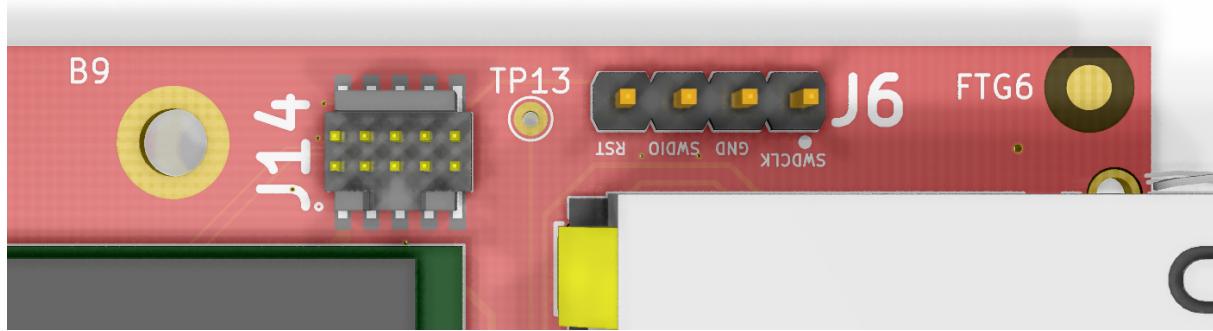


Figure 6: MMC JTAG and SWD connectors

Download the latest version of the microcontroller testing code from GitHub:

[https://gitlab.lbl.gov/spaiagua/marble\\_mmc/-/tree/unified\\_marble](https://gitlab.lbl.gov/spaiagua/marble_mmc/-/tree/unified_marble)

A recent version of OpenOCD (v0.10.0 or later) is required.

1. Connect JTAG module to **J14**
2. Connect the micro USB cable and using the serial terminal, connect to the last serial port for the new listed in the operating system. Use 115200 baudrate.
3. Power up the board.
4. Program the microcontroller using the following commands:
  - (a) Go to the main folder of the downloaded repository.
  - (b) Open command terminal and run command:

```
$ make marble_download
```

5. After successful programming, a menu in the serial terminal should appear and LEDs (LD15, LD11, LD12) should blink in the "snake" pattern.

## 2.2 LEDs

Three general purpose LEDs are connected to the MMC chip:

1. LED11 - connected to pin PE1
2. LED12 - connected to pin PE2
3. LED15 - connected to pin PE0

## 2.3 I2C Tree

Marble is equipped with two I2C buses (block diagram is shown in fig. 7):

1. I2C\_PM - supports devices for power management, temperature measurement and fan control
2. I2C\_FPGA - This bus is shared between the MMC and the FPGA chip. Through an I2C switch, it is connected to:
  - (a) FMC 1 and FMC 2
  - (b) Clock multiplexer
  - (c) SO-DIMM module
  - (d) QSFP 1 and QSFP 2
  - (e) Current measurement devices, Si570 and IO expanders

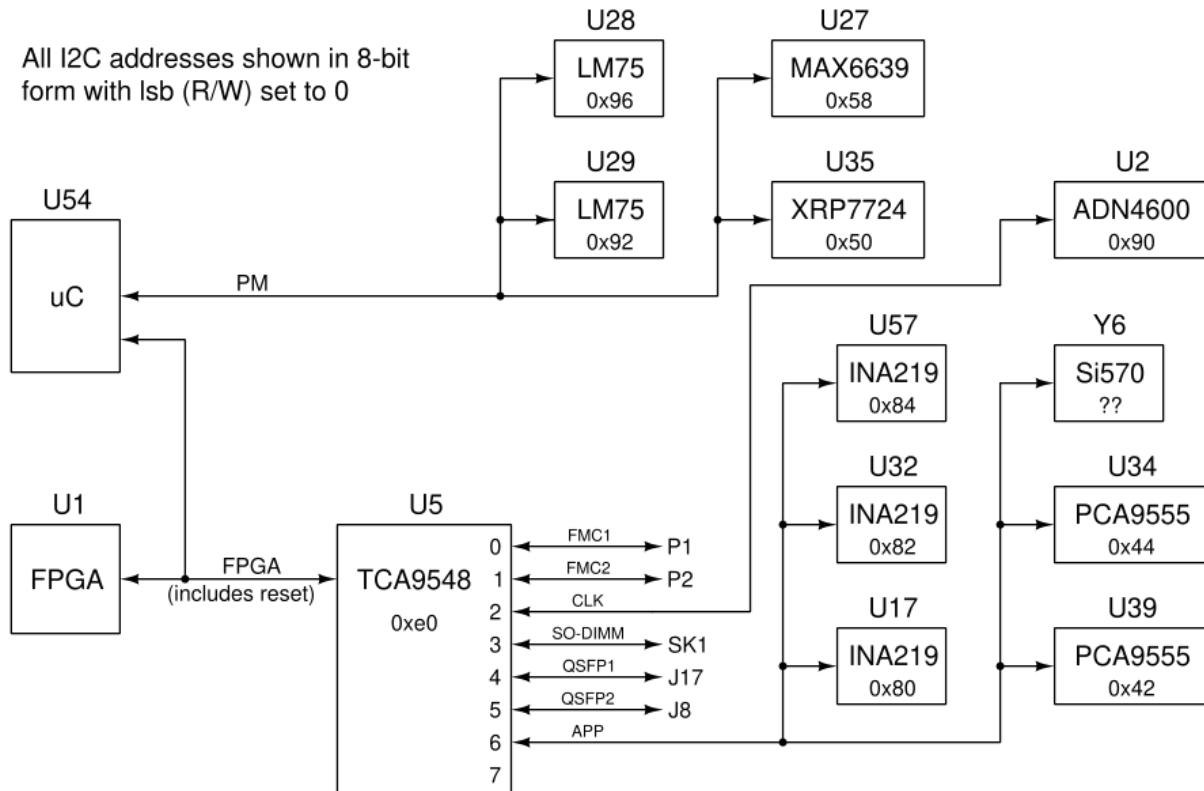


Figure 7: I2C map

### 3 SO-DIMM

The size of the DDR3 memory can be determined by the user by assembling the appropriate SO-DIMM module to the board. The use of a 204 pin SO-DIMM connector allows up to 4 GB of RAM to be connected to the FPGA. An I2C interface is provided to the memory module so that additional information about the module can be read. The default power supply for memory and HP banks is set to 1.5V and can only be changed by changing resistors. ECC is not supported by the module.

A reference design with a memory controller and a constraint file can be found here:

| <https://github.com/BerkeleyLab/bedrock>

See its projects/test\_marble\_family directory.

### 4 GTH Routing

Gigabit transceivers routing can be configured by the microcontroller. Transceivers from Bank 116 are permanently connected to module QSFP1. Transceivers from Bank 115 can be routed in 3 ways:

1. 4 transceivers connected to QSFP2
2. 4 transceivers connected to FMC P2
3. 2 transceivers connected to FMC P2 and 2 transceivers connected to FMC P1

Transceiver routing configuration can be set by the microcontroller. Three signals control the multiplexers which provide high quality signal switching. Gigabit multiplexer switching can be done from the UART console. By default, the multiplexers are set to route signals to the second QSFP connector. In the table 4, the MUXx columns correspond to the controlling logical state.

| MUX3 | MUX2 | MUX1 | MGT4       | MGT5       | MGT6       | MGT7      |
|------|------|------|------------|------------|------------|-----------|
| 0    | 0    | 0    | FCM2-DP0   | FMC2-DP1   | FMC2-DP2   | FMC1-DP1  |
| 0    | 0    | 1    | FCM2-DP0   | FMC2-DP1   | FMC1-DP0   | FMC1-DP1  |
| 0    | 1    | 0    | FCM2-DP0   | FMC2-DP1   | FMC2-DP2   | FMC2-DP3  |
| 0    | 1    | 1    | FCM2-DP0   | FMC2-DP1   | FMC1-DP0   | FMC2-DP3  |
| 1    | X    | X    | QSFP2:3/10 | QSFP2:1/12 | QSFP2:2/11 | QSFP2:4/9 |

Table 4: GTH transceivers routing table

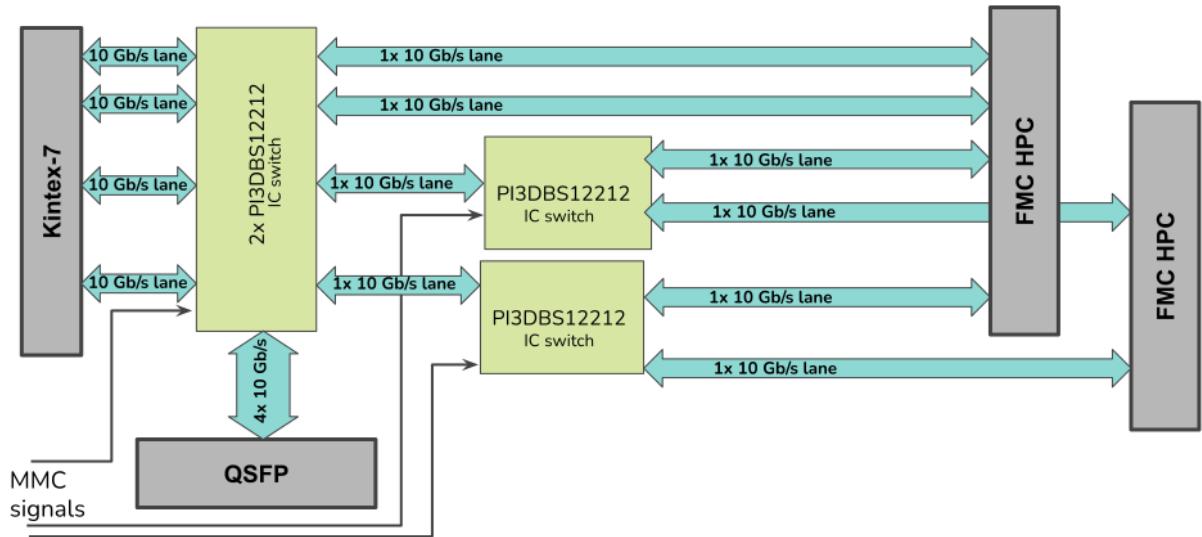


Figure 8: Transceivers routing block diagram.

Use UART console to change the transceiver routing style.

## 5 Clocking

This section describes how and where clock signals are routed. There are 7 on-board clock sources:

| Clock Name                               | Reference | Description  |
|--|-----------|--|
| System Clock                             | Y1, Y2    | Y1 and Y2 can be soldered interchangeably and provide a reference clock source for the U20 chip  |
|  | U20       | U20 is an ultra-low jitter clock generator that provides a 125 MHz clock for the FPGA bank 33 that can operate as a DDR3 reference clock. Additionally, U20 generates 125 MHz clock which is connected to the clock multiplexer. |
| Additional Clock                         | Y3        | VCXO - provides 20 MHz clock for FPGA bank 33  |
| User Clock                               | Y6        | Si570 - provides variable clock frequencies for 10 MHz to 280 MHz; connected to the clock multiplexer  |
| User U.FL Clock 1<br>(differential pair) | J2, J5    | External clock source input  |
| User U.FLClock 2<br>(differential pair)  | J3, J7    | External clock source input  |
| FMC1 M2C                                 | P1        | GBTCLK 0 & 1   |
| FMC2 M2C                                 | P2        | GBTCLK 0 & 1   |

Table 5

Marble supports external clock sources from FMC and from U.FL connectors. Thanks to the clock multiplexer, any clock input can be connected to any FPGA MGT clock input. Clock routing and Si570 frequency can be changed over I2C from the house-keeping microcontroller or the FPGA.

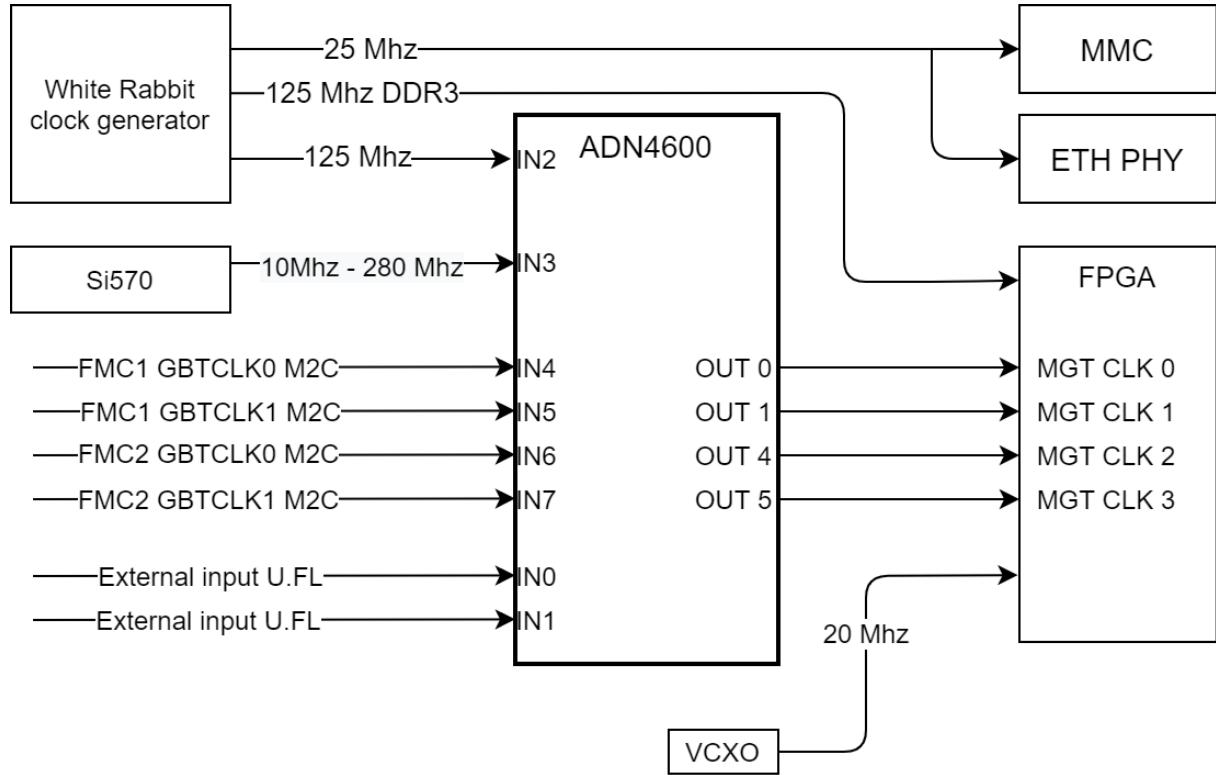


Figure 9: Marble clocking scheme

## 5.1 White Rabbit clock generator

The clock chips available on the board allow implementation of the White Rabbit protocol. White Rabbit provides sub-nanosecond accuracy and picoseconds precision of synchronization for large distributed systems.

## 6 Pmod connectors

Marble has 3 Pmod connectors that support 3.3V logic level:

1. J12 and J13 are connected to FPGA
2. J6 is connected the MMC

|           | Pmod 1 (J12)<br>FPGA | Pmod 2 (J13)<br>FPGA | Pmod (J16)<br>MMC |
|-----------|----------------------|----------------------|-------------------|
| Pmodx_C_0 | IO_L6N_14            | IO_L7P_33            | PB9 (SEL)         |
| Pmodx_C_1 | IO_L7N_14            | IO_L2N_33            | PC3 (MOSI)        |
| Pmodx_C_2 | IO_25_14             | IO_L4N_33            | PC2 (MISO)        |
| Pmodx_C_3 | IO_L7P_14            | IO_L7N_33            | PB10 (SCK)        |
| Pmodx_C_4 | IO_0_14              | IO_L2P_33            | PB14 (EINT1)      |
| Pmodx_C_5 | IO_L5N_15            | IO_L8P_33            | PB15              |
| Pmodx_C_6 | IO_L4P_15            | IO_L4P_33            | PD6 (UART4 RX)    |
| Pmodx_C_7 | IO_L5P_15            | IO_L3N_33            | PD5 (UART4 TX)    |

Table 6: Pmod connectors pins assignment

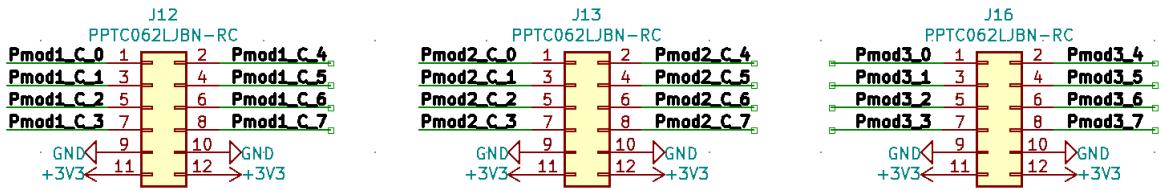


Figure 10: Pmod pinout

## 7 QSFP

Marble is equipped with 2 QSFP cages (11). Each can support 40 Gb/s of data transfer:

1. QSFP number 1 - high-speed lines are directly connected to the FPGA bank 116
2. QSFP number 2 - high-speed lines are connected to the FPGA bank 115 through the gigabit multiplexer.

QSFPs control signals are connected to the I2C IO expander (U34)(12) which is accessible from FPGA or MMC. The connection of the differential signals to the FPGA is shown in the tables (7)(8). QSFP2 is not directly connected to the FPGA chip. It is connected via a gigabit multiplexer which is controlled from the MMC. By default, the multiplexer is configured to connect the QSFP2 to the FPGA. In order to choose a different configuration or restore the default, select the appropriate option from the MMC console.

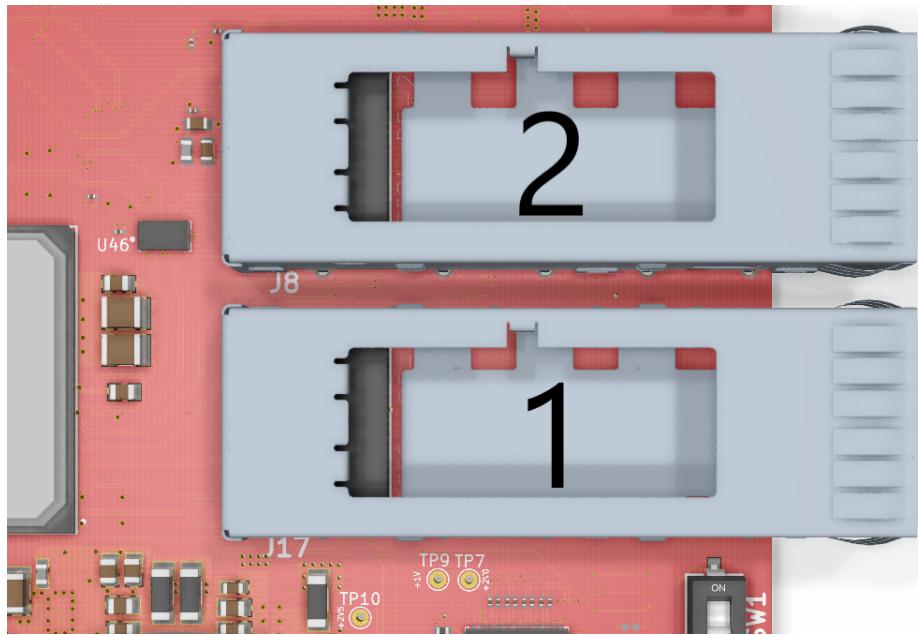


Figure 11: QSFP cages

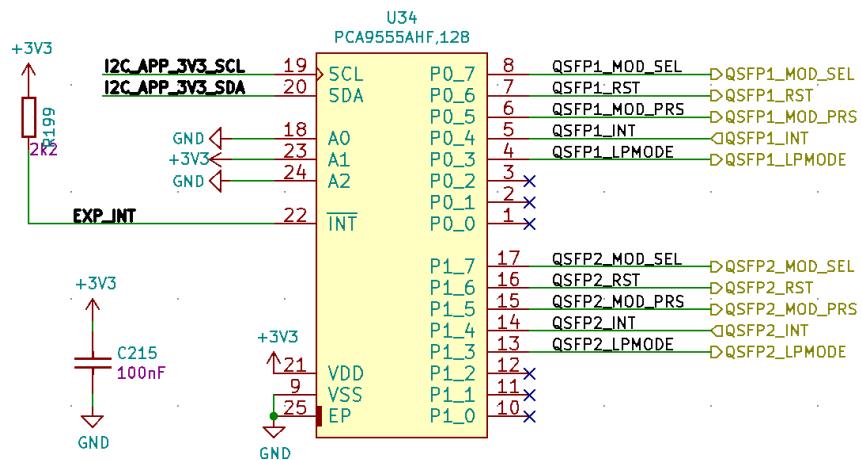


Figure 12: I2C IO expander which controls QSFPs signals

## 8 Ethernet

Marble is equipped with Ethernet PHY (88E1512) which supports 10/100/1000BASE-T. PHY is connected to the FPGA bank 16 via RGMII interface:

The Ethernet PHY can be monitored over MDIO by the MMC. IP address and MAC address are stored in the MMC's internal EEPROM memory (EEPROM functionality is emulated in the internal flash). To read the internal PHY configuration registers, select the appropriate option from the MMC console. By default, the MDIO address is set to 1 but

| FPGA Pin     | QSFP Signal  |
|--------------|--------------|
| MGTXTXP0_116 | QSFP1_TX_3_P |
| MGTXTXN0_116 | QSFP1_TX_3_N |
| MGTXRXP0_116 | QSFP1_RX_3_P |
| MGTXRXN0_116 | QSFP1_RX_3_N |
| MGTXTXP1_116 | QSFP1_TX_1_P |
| MGTXTXN1_116 | QSFP1_TX_1_N |
| MGTXRXP1_116 | QSFP1_RX_1_P |
| MGTXRXN1_116 | QSFP1_RX_1_N |
| MGTXTXP2_116 | QSFP1_TX_2_P |
| MGTXTXN2_116 | QSFP1_TX_2_N |
| MGTXRXP2_116 | QSFP1_RX_2_P |
| MGTXRXN2_116 | QSFP1_RX_2_N |
| MGTXTXP3_116 | QSFP1_TX_4_P |
| MGTXTXN3_116 | QSFP1_TX_4_N |
| MGTXRXP3_116 | QSFP1_RX_4_P |
| MGTXRXN3_116 | QSFP1_RX_4_N |

Table 7: QSFP1 pins connection

| FPGA Pin     | QSFP Signal  |
|--------------|--------------|
| MGTXTXP0_115 | QSFP1_TX_3_P |
| MGTXTXN0_115 | QSFP1_TX_3_N |
| MGTXRXP0_115 | QSFP1_RX_3_P |
| MGTXRXN0_115 | QSFP1_RX_3_N |
| MGTXTXP1_115 | QSFP1_TX_1_P |
| MGTXTXN1_115 | QSFP1_TX_1_N |
| MGTXRXP1_115 | QSFP1_RX_1_P |
| MGTXRXN1_115 | QSFP1_RX_1_N |
| MGTXTXP2_115 | QSFP1_TX_2_P |
| MGTXTXN2_115 | QSFP1_TX_2_N |
| MGTXRXP2_115 | QSFP1_RX_2_P |
| MGTXRXN2_115 | QSFP1_RX_2_N |
| MGTXTXP3_115 | QSFP1_TX_4_P |
| MGTXTXN3_115 | QSFP1_TX_4_N |
| MGTXRXP3_115 | QSFP1_RX_4_P |
| MGTXRXN3_115 | QSFP1_RX_4_N |

Table 8: QSFP2 pins connection

| RGMII signal | FPGA pin   |
|--------------|------------|
| RGMII_RXD0   | IO_L4N_16  |
| RGMII_RXD1   | IO_0_16    |
| RGMII_RXD2   | IO_L1P_16  |
| RGMII_RXD3   | IO_L1N_16  |
| RGMII_RX_DV  | IO_L4P_16  |
| RGMII_RX_CLK | IO_L14P_16 |
| RGMII_TXD0   | IO_L6N_16  |
| RGMII_TXD1   | IO_L6P_16  |
| RGMII_TXD2   | IO_L8N_16  |
| RGMII_TXD3   | IO_L8P_16  |
| RGMII_TX_EN  | IO_L10P_16 |
| RGMII_TX_CLK | IO_L11N_16 |
| PHY_RSTn     | IO_L10N_16 |

Table 9: RGMII pins assignment

it can be changed by changing the resistor on the module. To set address to 1, desolder resistor R80 and solder resistor R65 with value 0R.

## 9 FMC

Both FMC sockets are equipped with HPC (High Pin Count) type connector but not all signals were connected to FPGA and MMC. The connection of both FMC connectors is shown below:

### 1. FMC P1 signal connection:

- (a) LA00...LA33 – connected to FPGA banks 15 and 16.
- (b) HA00...HA23 – *not connected*.
- (c) HB00...HA21 – *not connected*.
- (d) CLK[0..1]\_M2C – connected to FPGA bank 15.
- (e) GBTCLK[0..1]\_M2C – connected to CLK MUX IN4 and IN5.
- (f) DP[0]\_M2C – connected to high speed MUX.
- (g) DP[0]\_C2M – connected to high speed MUX.
- (h) JTAG – connected to MMC.
- (i) I2C – connected to I2C bus shared with MMC and FPGA

### 2. FMC P2 signal connection:

- (a) LA00...LA33 – connected to FPGA banks 12 and 14.
- (b) HA00...HA23 – connected to FPGA bank 13.
- (c) HB00...HA21 – *not connected*.
- (d) CLK[0..1]\_M2C – connected to FPGA banks 12 and 14.

- (e) GBTCLK[0..1]\_M2C - connected to CLK MUX IN6 and IN7.
- (f) DP[0..3]\_M2C - connected to high speed MUX.
- (g) DP[0..3]\_C2M - connected to high speed MUX.
- (h) JTAG - connected to MMC.
- (i) I2C - connected to I2C bus shared with MMC and FPGA

## 10 USB-UART

By default, the USB-UART bridge (FT4232H) has 4 UART channels. Only the last 2 of these are used for communication over the serial interface. The first channel is used as a JTAG interface, while the second channel uses only the DTR signal to generate a reset signal.

### 10.1 FPGA UART

FPGA UART occupies FT4232's channel 3. UART signals are connected to bank 15 pins:

1. TX input - IO\_L1P\_15
2. RX output - IO\_0\_15

### 10.2 MMC UART

MMC UART occupies FT4232's channel 4 and is used as the MMC serial console. Terminal configuraton is 1N8 115200 baudrate. The FT4232's channel 2 DTR signal can be used to provide a reset signal for the microcontroller.

## 11 Power

Several hardware options are available to provide the nominal +12V power to the board. It can come from TE Connector (3-641119-2) - J19 (fig. 13) or by standard barrel connector (Type A: 5.5 mm OD, 2.1 mm ID) - J1. Input voltage range: 10V - 18V. Additionally, PoE can be used to power the module.

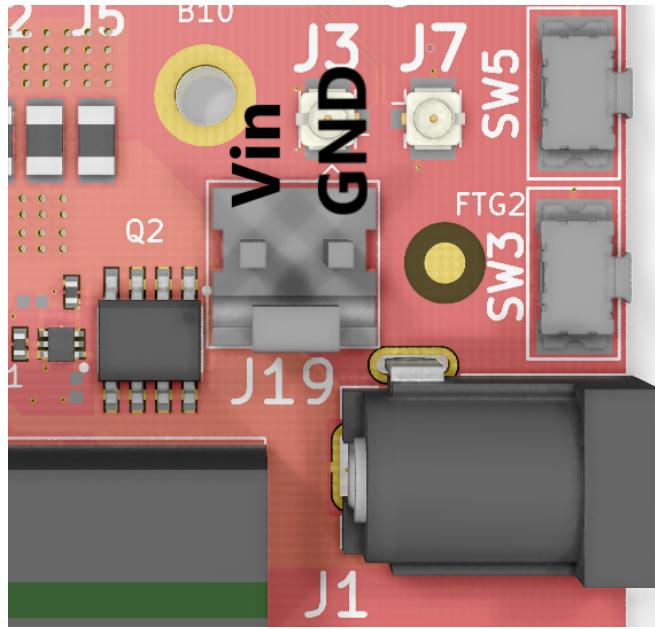


Figure 13: J19 connector with indicated Vin and GND

When the power is connected, the board starts up automatically. If there is a failure of any power rail, the LED corresponding to it will not light up.

Block diagram of Marble's power tree is shown in figure 14. The USB-UART bridge is powered from the micro USB connector via a converter (U22). The supervising microcontroller has a separate converter (U18) connected directly to the power input. The main voltages supplying the FPGA are produced by U35 – the programmable power management system. U35 turns on the individual power channels in the proper sequence and provides power to the additional converter (U58) and LDOs (U31, U36, U37, U47).

Power supply features:

1. Over-temperature protection.
2. Power rails for the FPGA can be switched off and on by the microcontroller.
3. All power rails generated by XRP7724 can be monitored by the microcontroller and they are equipped with over-current protection.
4. Current consumption, voltage and rails status can be read by microcontroller.
5. The presence of the power rails is indicated by LED diodes
6. 12V power supply for both FMCs can be controlled independently by the microcontroller. Additionally, current can be measured.

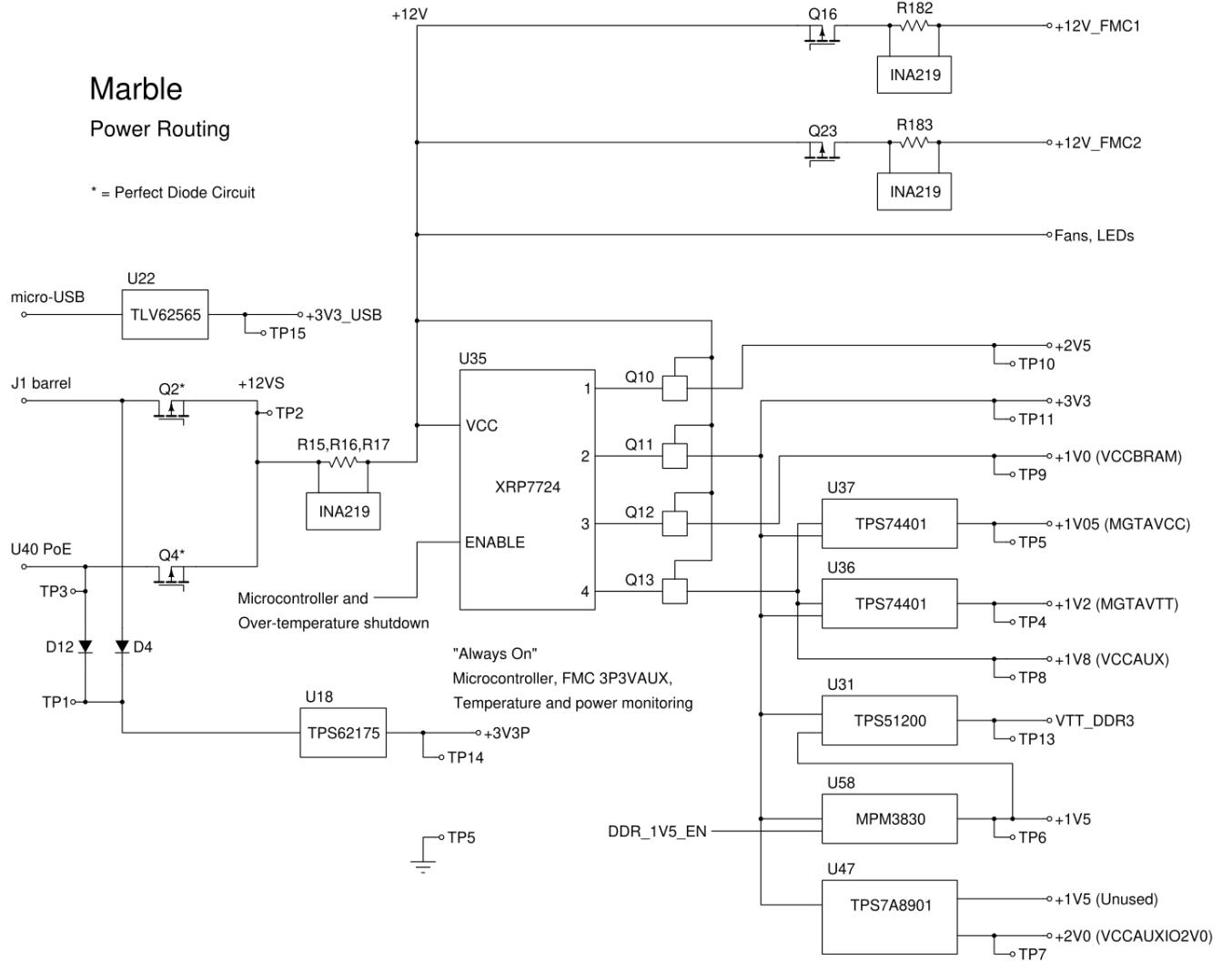


Figure 14: Marble power routing

## 11.1 Fan controller

The fan control and temperature monitoring of the FPGA chip is done with the MAX6639 chip. Selection, mounting, and configuration of the fans is outside the scope of this document. The fans can be automatically controlled by measuring the temperature on the diode inside the FPGA. If the temperature exceeds a preset alarm threshold an ALERT signal will be issued. If the temperature continues to rise and exceeds another threshold, an "OVER-TEMP" signal will be issued and the FPGA will automatically power down. Through the I<sup>2</sup>C interface it is possible to read and write all MAX6639 configuration registers. Additionally, signals from both fan tachometers are monitored.

Two additional temperature sensors based on the LM75 chip provide temperature measurements around the main power converter and under the FMC P1 card. By default they are set to shut down the main power inverter when it exceeds 75 C.

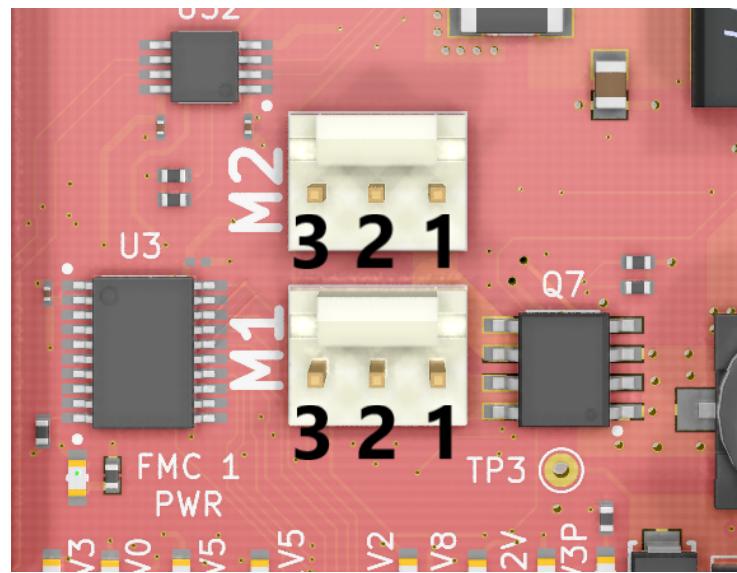


Figure 15: FANs connectors

|   | M1    | M2      |
|---|-------|---------|
| 1 | GND   | 1 GND   |
| 2 | 12V   | 2 12V   |
| 3 | Tacho | 3 Tacho |

Table 10

## 12 Mechanical dimensions

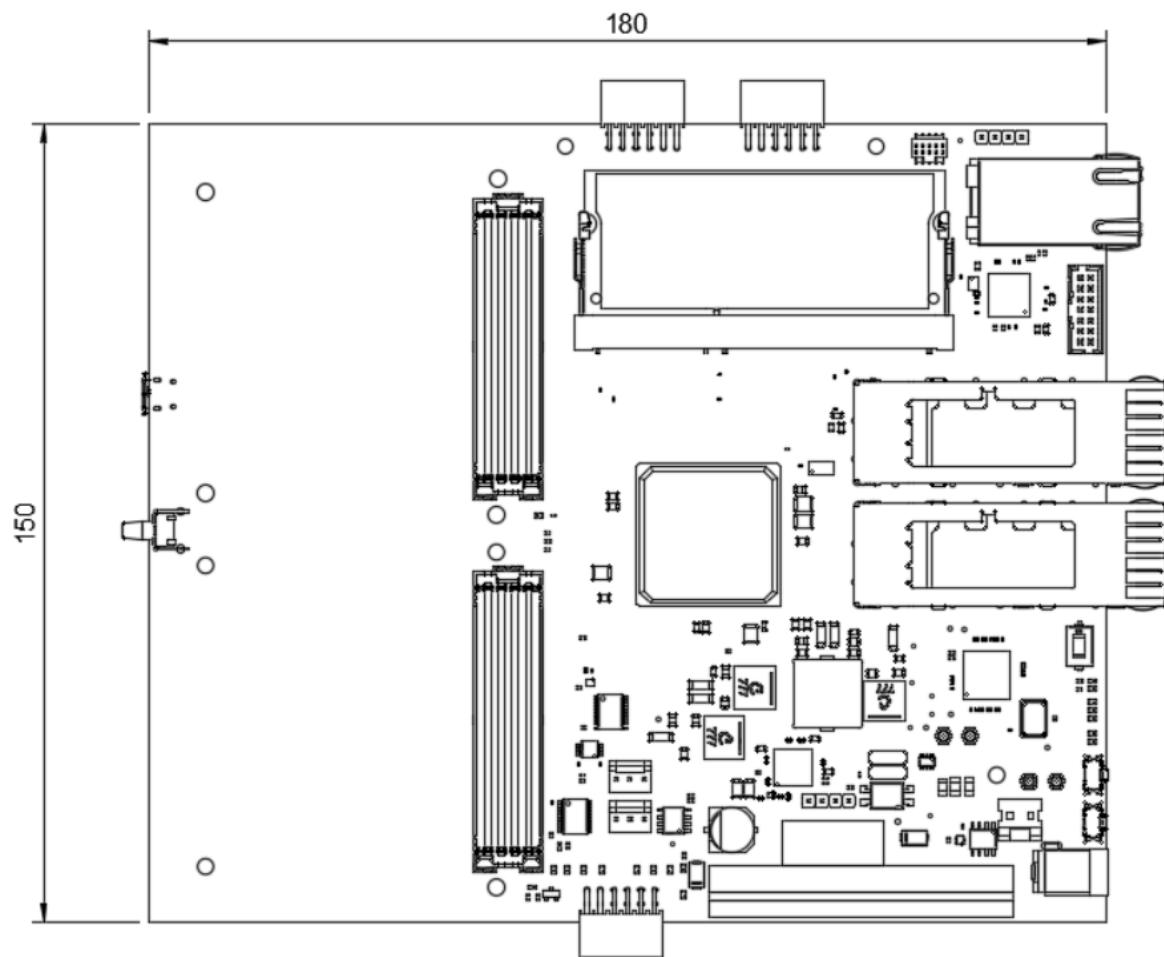


Figure 16: Mechanical dimensions

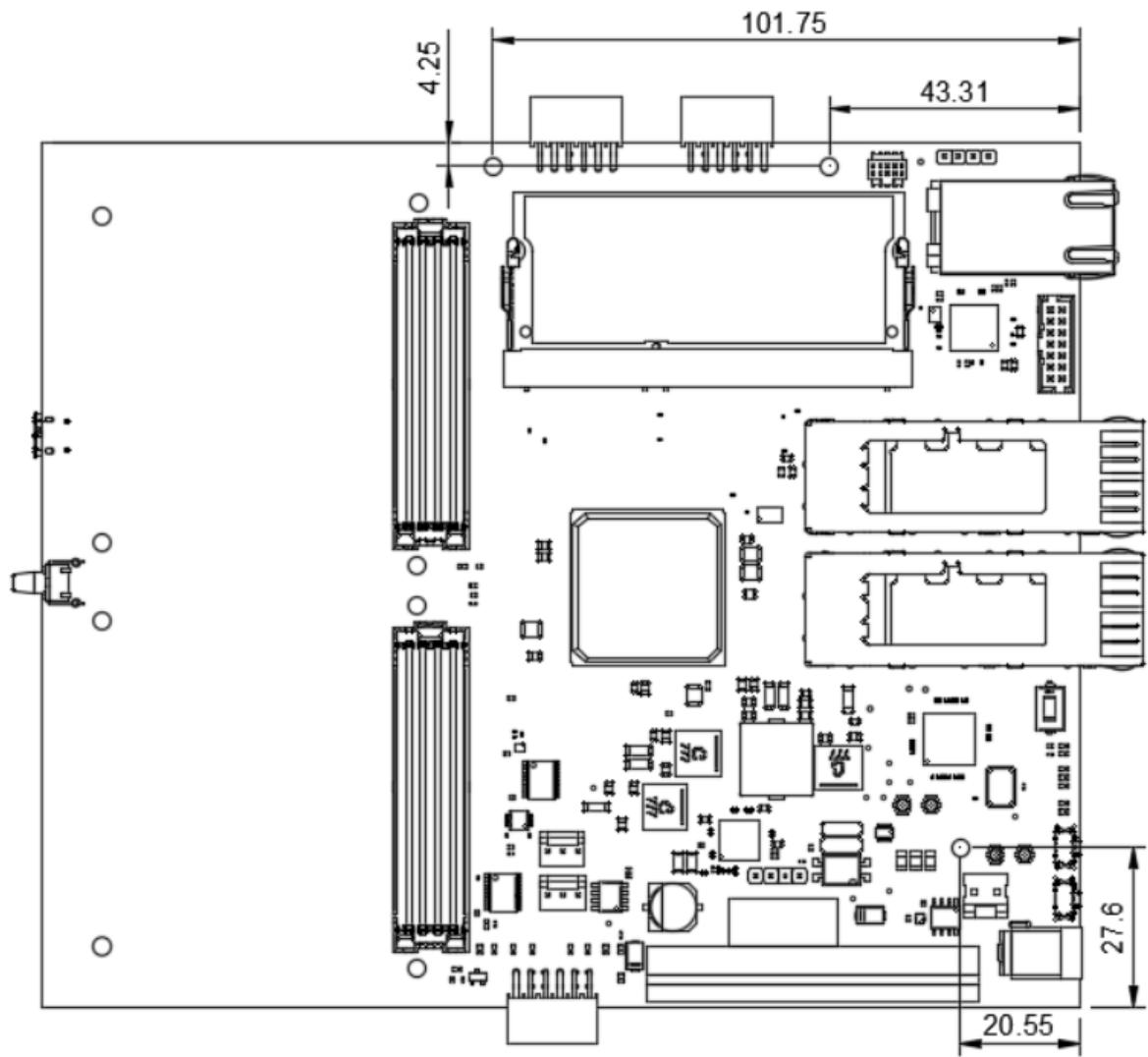


Figure 17: Mounting holes positioning

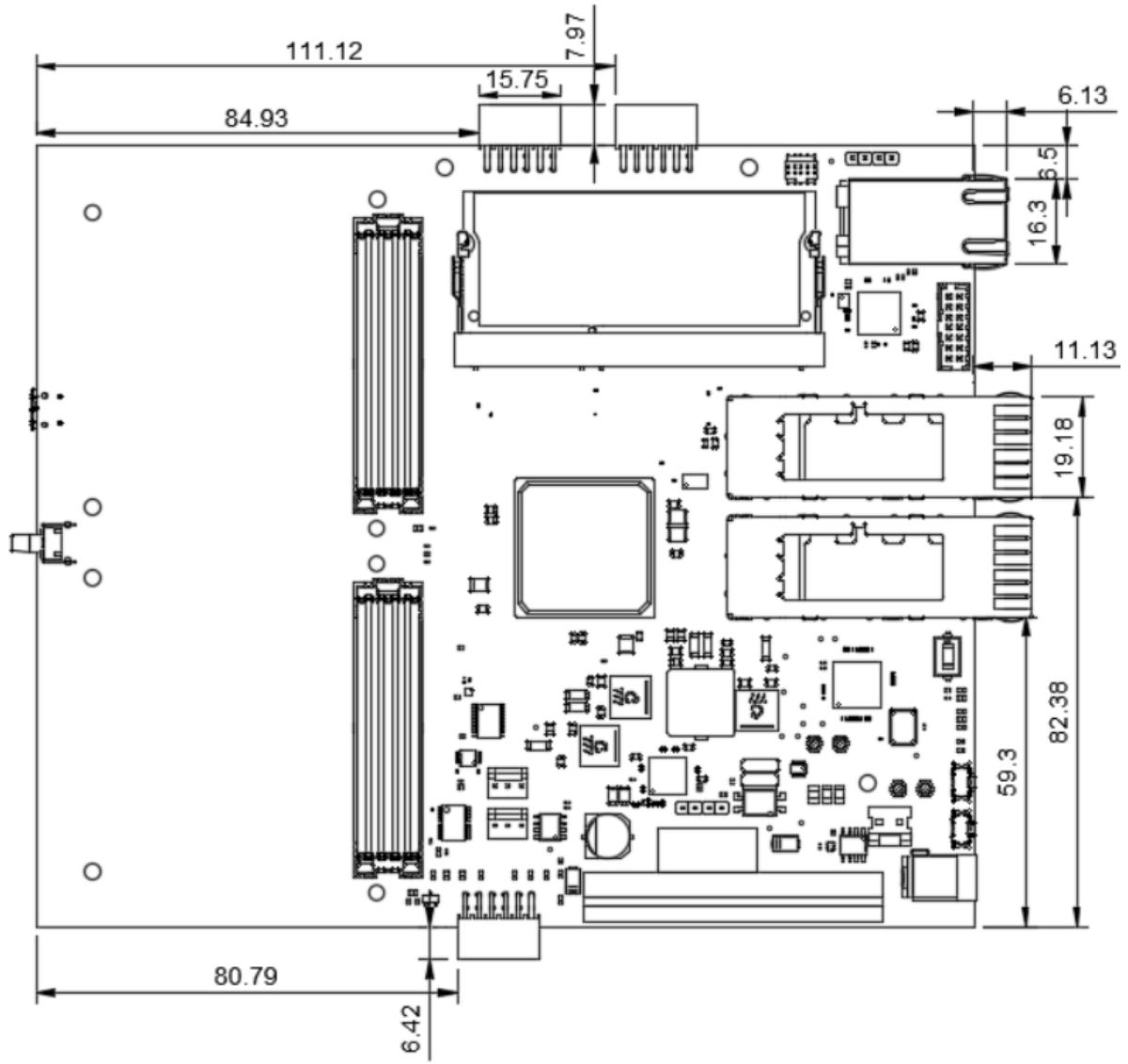


Figure 18: Protruding connectors positioning

# 13 Appendix

## 13.1 Power supply test points

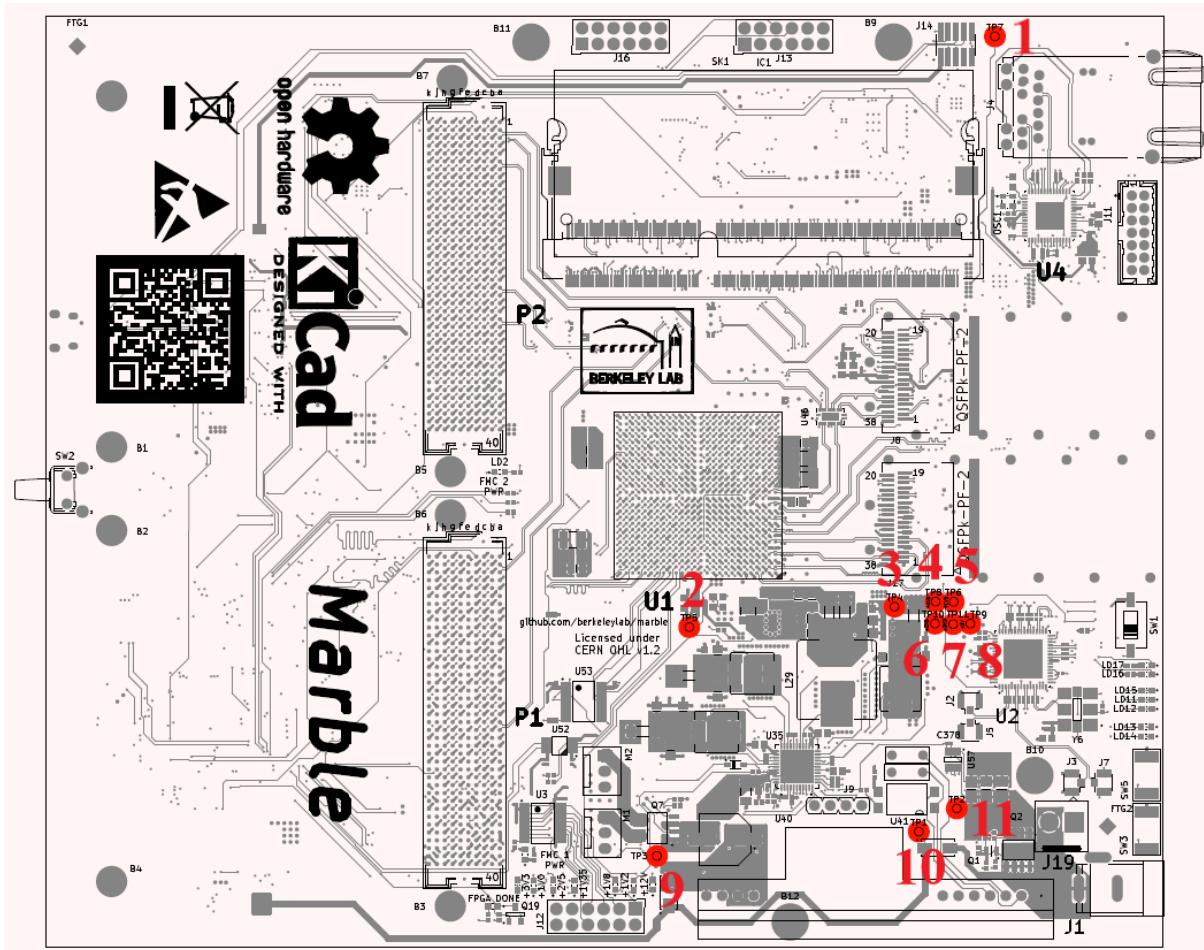


Figure 19: Test point map

- (a) Number 1 - TP13 VTT\_DDR3 (0.75V).
- (b) Number 2 - TP5 MGTAVCC\_DDR3 (1.05V).
- (c) Number 3 - PoE
- (d) Number 4 - MGTAVTT (+1.2V)
- (e) Number 5 - MGTAVCC (+1.05V)
- (f) Number 6 - +1V5
- (g) Number 7 - VCCAUXIO2V0 (+2.0V)
- (h) Number 8 - VCCAUX (+1.8V)
- (i) Number 9 - VCCBRAM (+1.0V)

- (j) Number 10 - **+2V5**
- (k) Number 11 - **+3V3**
- (l) Number 12 - **GND**
- (m) Number 13 - **VTT\_DDR3 (+0.75V)**
- (n) Number 14 - **+3.3P (always-on)**
- (o) Number 15 - **+3V3\_USB**

Test points 12-15 are not present on Marble v1.0.

## References