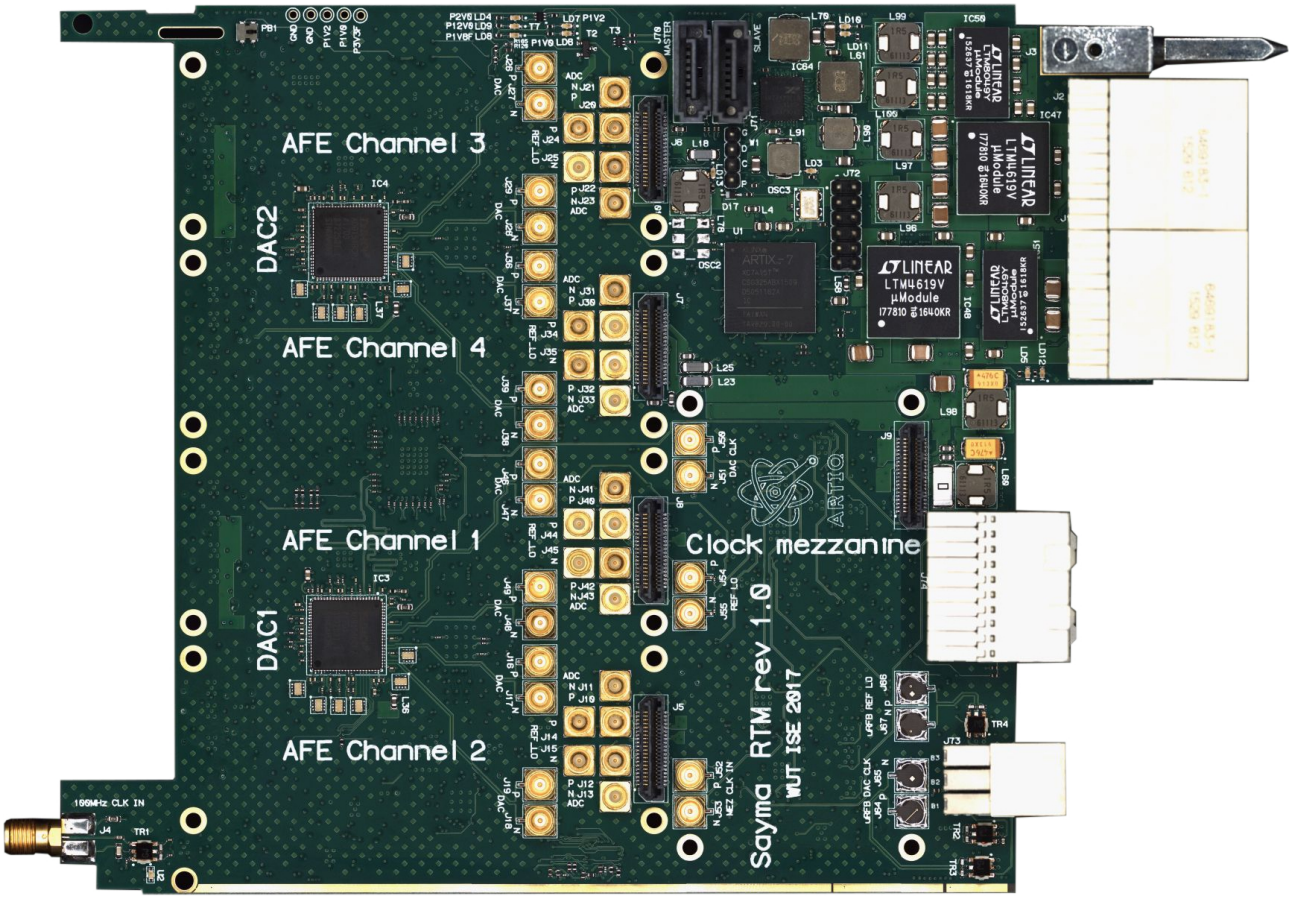
DRAFT

SAYMA RTM
specification



v1.0(02.2018)

Document version:	Preliminary
Issue Date:	February 7, 2018
Written by:	Filip Świtakowski
Approved by:	————————
Document title:	SAYMA RTM- specification

Todo list

Contents

1 Glossary

AFE Analogue front-end.

AMC Module or Modul An AMC Module is a mezzanine or modular add-on card that extends the functionality of a Carrier Board. The term is also used to generically refer to the different varieties of Multi-Width and Multi-Height Modules.

BaseMod Base-band input/output mezzanine.

COTS Commercial off-the-shelf. Product which is designed and can be easily purchased.

EEM Eurocard Extension Module is a Sinara standard for low-cost, low-bandwidth peripherals that are controlled by ARTIQ DRTIO.

Fat Pipes Ports 4 though 11 of the AMC Connector constitute the Fat Pipes Region. This Region of Ports is intended for the assignment of multiple Lane interfaces, also called "fat pipes". Fat Pipe 1 [Ports 4-7], Fat Pipe [Ports 8-11].

FMC FPGA Mezzanine Card

HEPP High Energy Physics. ???

Hot Swap To remove a component (e.g., an AMC Module) from a system (e.g., an AMC Carrier AdvancedTCA Board) and plug in a new one while the power is still on and the system is still operating.

IPMB ntelligent Platform Management Bus. The lowest level hardware management bus as described in the Intelligent Platform Management Bus Communications Protocol Specification.

Management Power or MP The 3.3V power for a Module's Management function, individually provided to each Slot by the Carrier

MGT Multi-Gigabit Transceiver.

MixMod An up-converting mezzanine, using an analogue IQ mixer to mix the input and output RF signals with a LO supplied by Sayma.

MMC Module Management Controller. The MMC is the required intelligent controller that manages the Module and is interfaced to the Carrier via IPMB-Local.

RFBP RF Backplane.

RTM Rear Transition Module.

Sayma Smart Arbitrary Waveform Generator, providing 8 channels of 1.2 GSPS 16-bit DACs (2.4 GHz DAC clock) and 125 MSPS 16-bit ADCs. It consists of an AMC, providing the high-speed digital logic, and a RTM, holding the data converters and analog components.

Sianra Open-source hardware ecosystem originally designed for use in quantum physics experiments running the ARTIQ control software. It is licensed under CERN OHL v1.2.

uTCA Micro Telecommunications Computing Architecture. MicroTCA is a modular, open standard for building high performance computer systems in a small form factor.

2 Overview

MicroTCA (uTCA) is Sinara's preferred form-factor for hardware with high-speed data converters requiring deterministic phase control, such as the *Sayma* 2.4 GSPS smart arbitrary waveform generator (SAWG).

uTCA is a modular, open standard originally developed by the telecommunications industry. It allows a single rack master – the Micro TCA Carrier Hub (MCH) – to control multiple slave boards, known as Advanced Mezzanine Cards (AMCs) via a high-speed digital backplane. uTCA chassis and backplanes are available commercially of the shelf (COTS).

We make use of the most recent extension to the uTCA standard, uTCA.4. Originating in the high-energy and particle physics (HEPP) community, uTCA.4 introduces rear-transition modules (RTMs) along with a second backplane for low-noise RF signals (RFBP). Each RTM connects to an AMC (one RTM per AMC). Typically, the AMCs hold FPGAs and other high-speed digital hardware, communicating with the MCH via gigabit serial links over the AMC backplane. The RTMs hold data converters and other low-noise analog components, controlled by the corresponding AMC. The RFBP provides low-noise clocks and local oscillators (LOs). The RTMs and RFBP are screened from the AMCs to minimise interference from the high-speed digital logic.

Figure 1: Micro TCA chassis with 3 Sayma AMC modules inserted

(above) Micro TCA chassis with 3 Sayma AMC modules inserted.

Micro TCA chassis with 4 RTM modules inserted. One of them with 4 BaseMod AFE mezzanines installed.

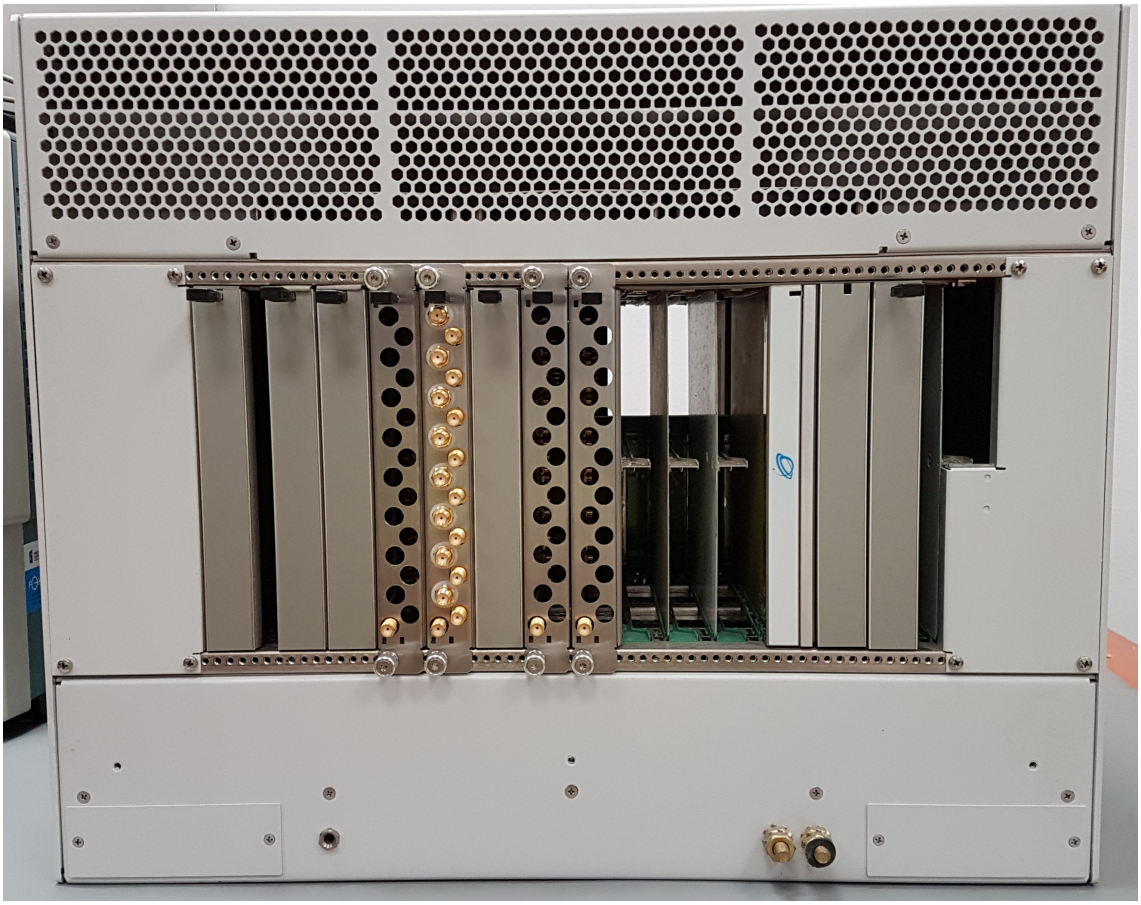


Figure 2: Micro TCA chassis with 4 RTM modules inserted. One of them has 4 BaseMod AFE mezzanines installed.

3 uTCA.4 RF Backplane

RF BP datasheet RF BP measurements

4 uTCA in Sinara

Metlino has been developed as an MCH optimised for use in Sinara. It can either be the ARTIQ master or a slave, connected to the master via DRTIO.

uTCA hardware interfaces with the extension modules either directly, using a VHDCI carrier, or indirectly, using a Kasli DRTIO slave.

To do: * Some images to illustrate what uTCA systems look like * Explain how Baikal etc fit in * Add BP schematics that show what the connectivity is * Any more useful information?

5 uTCA parts and suppliers

Add parts and suppliers from the issues list...

6 Schematic / Layout Viewer

Mentor has a free tool called visECAD Viewer.

7 Project description

The Sayma RTM module extends Sayma AMC board connectivity by DACs and ADCs modules.

jakiś opis Cite high speed ADC and DAC ICs by name and summarize high-level specifications – that's what this board is about!

8 Functional specifications

Programmable resources:

- Xilinx Artix-7 XC7A15T-1CSG325

Memory:

- EEPROM with MAC and unique ID

Connectivity:

- 4x mezzaninne connector LSS-120-01-L-DV-A
- 40x SMP connector for ADC/DAC
- Stand-alone 12V power connector
- RTM connector with 16 GTP pair routed to it.
- GTP on RTM connector connected to:
  - DAC x16 [Tx]
  - ADC x8 [Rx]
  - FPGA MGT 2x2 [Tx + Rx]
  - SATA x2
- uRFB connector

Supply:

- Monitoring of voltage and Power supply for FPGA and P3V3

Clocking:

- UFL CLK input
- SMA CLK output
- Si5324 Clock recovery

Other:

- Temperature, voltage and current monitoring for critical power buses

9 Product view

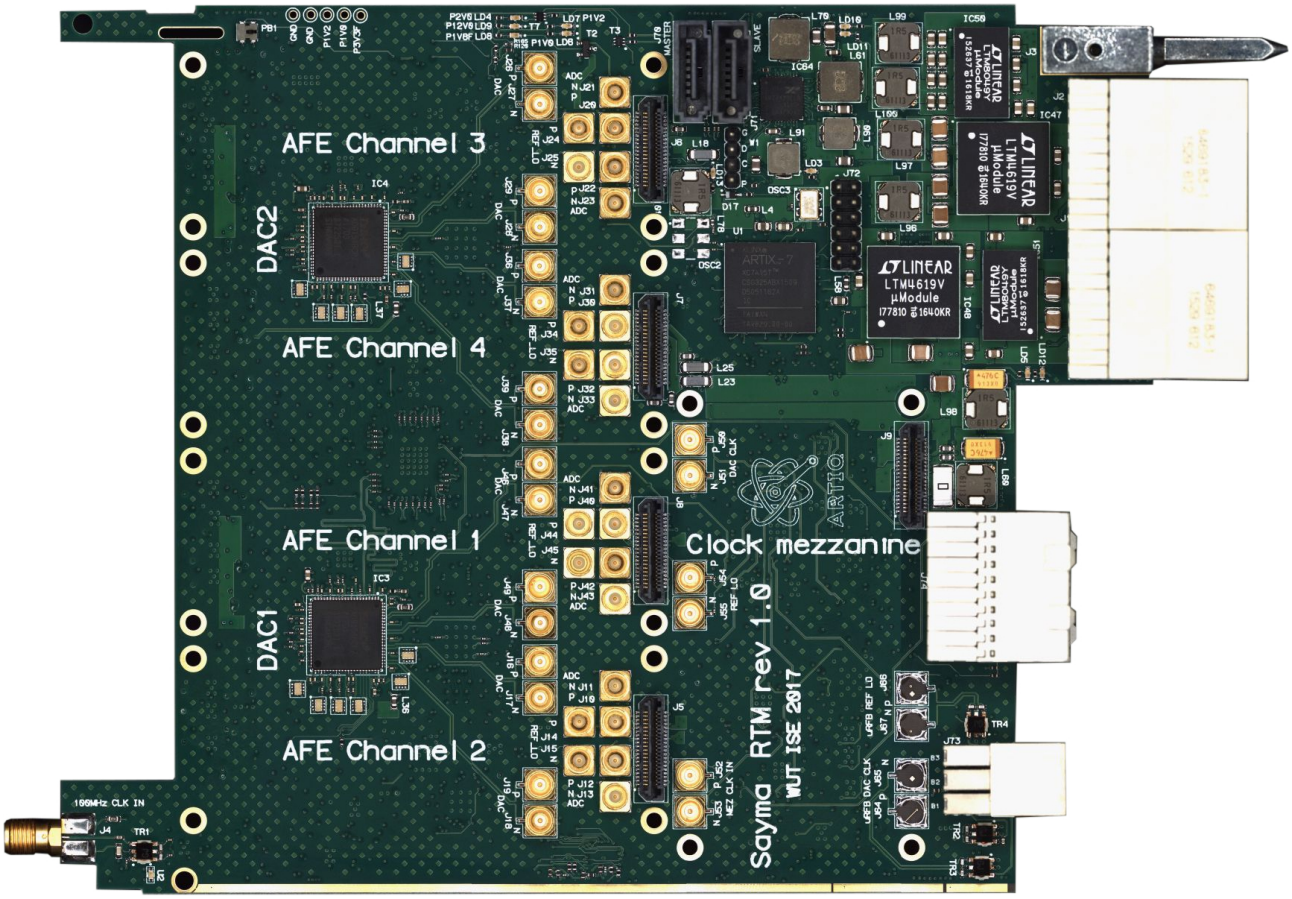


Figure 3: Top view

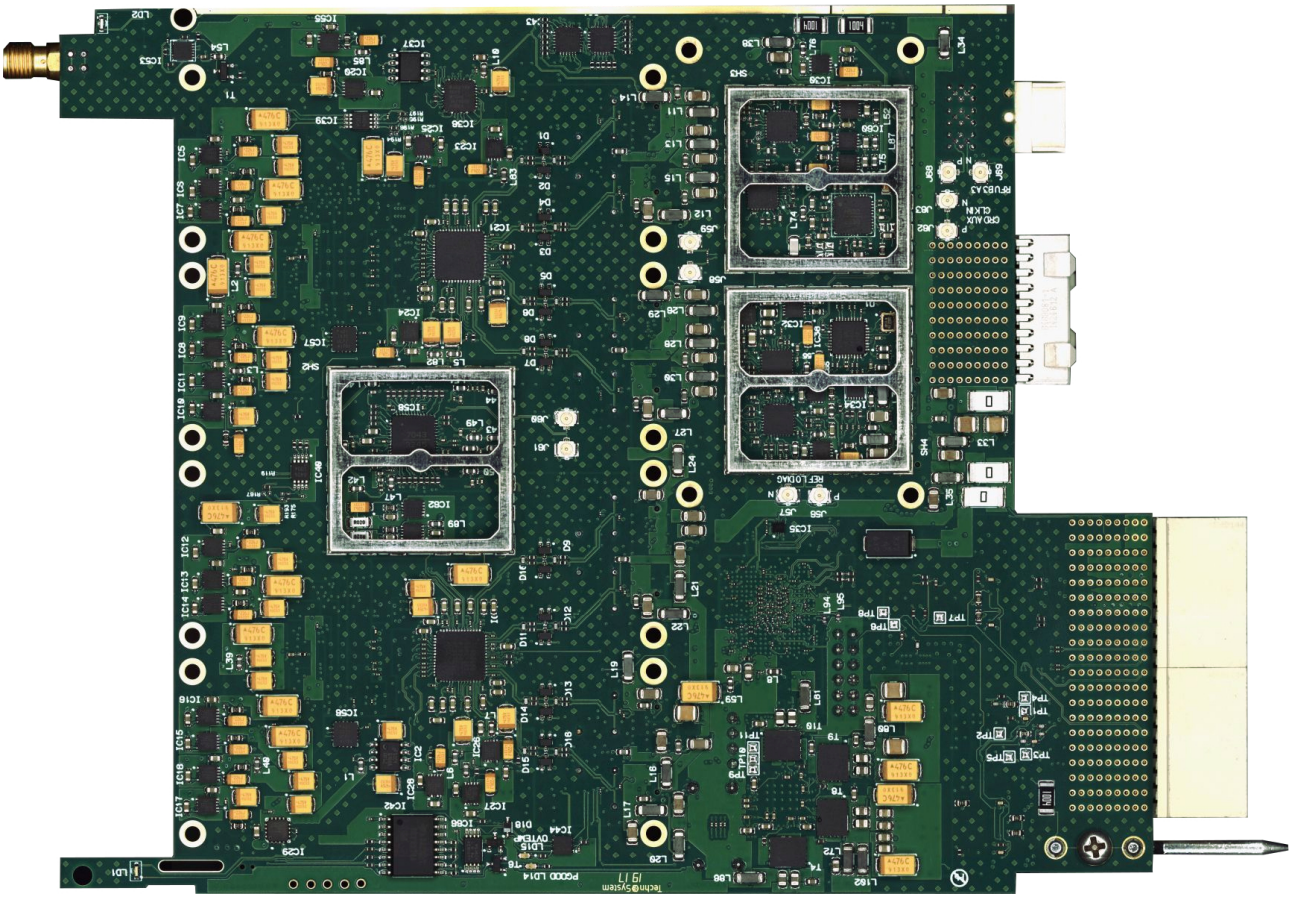


Figure 4: Bottom view

10 Routing

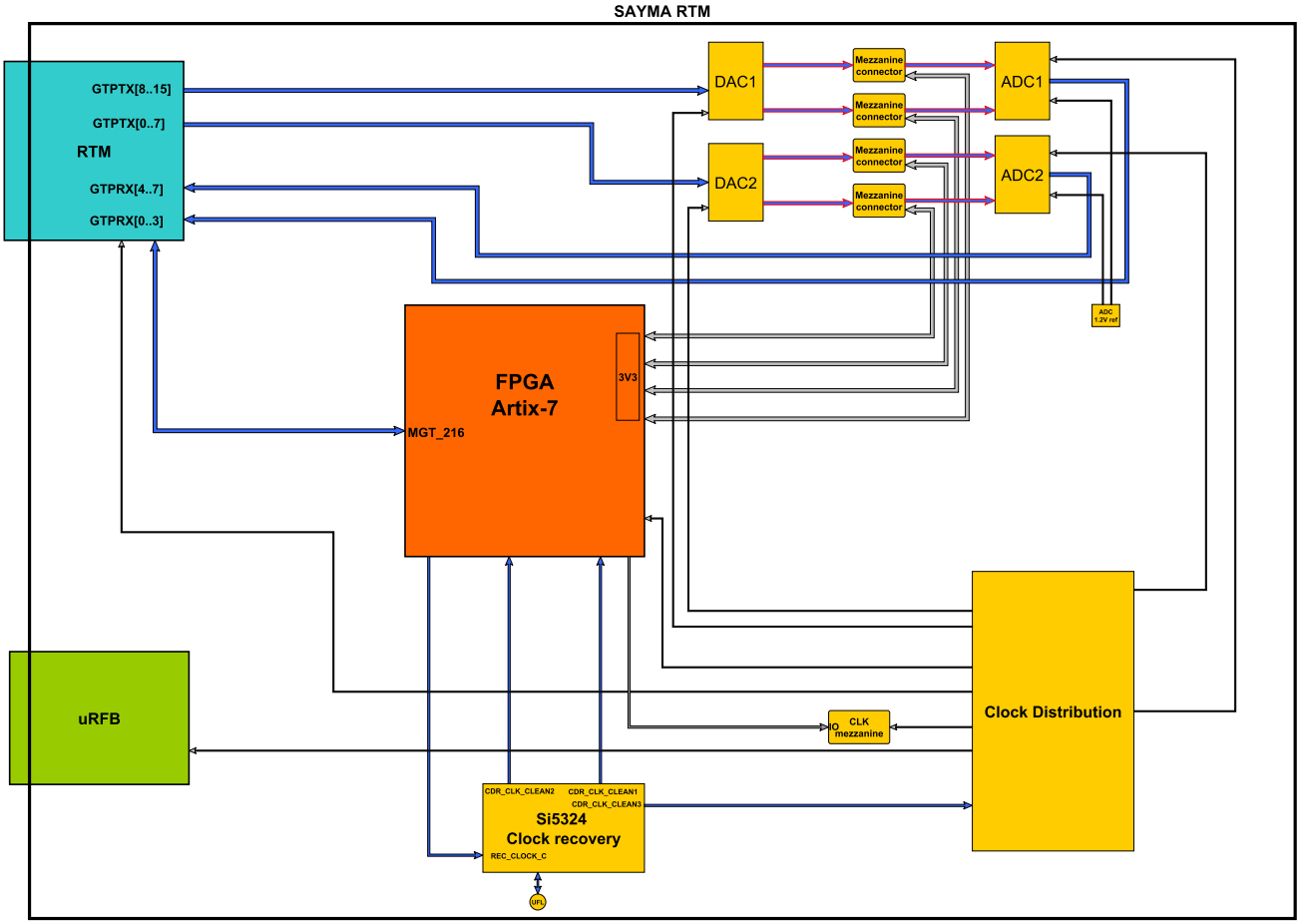


Figure 5: Block Scheme

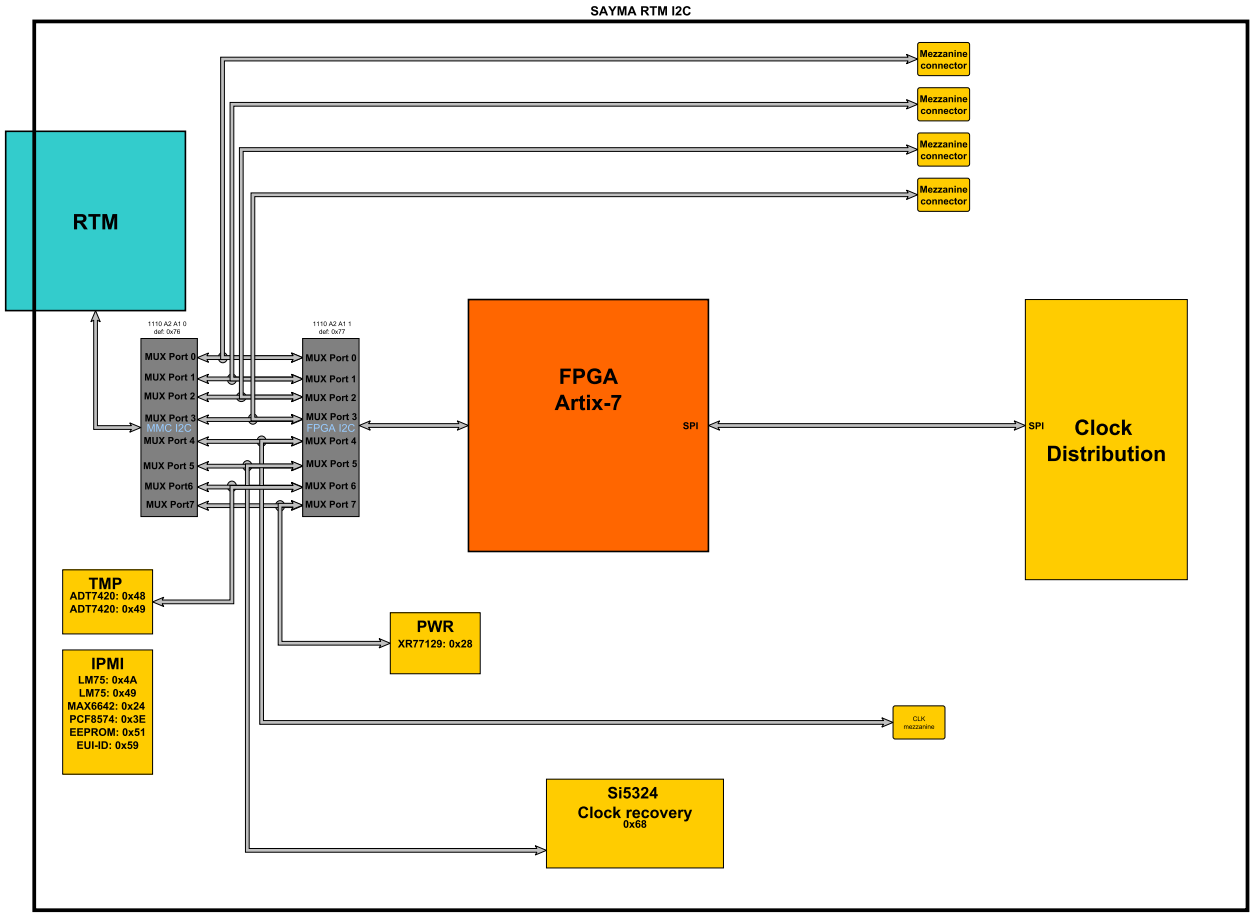


Figure 6: I2C

11 RF Front-End Mezzanines

Mezzanines providing analogue front-ends (AFEs) for the ADCs and DACs on Sayma.

12 Available AFEs

12.1 TestMod

Simple mezzanine designed for thermal and connectivity testing, and to serve as a template for other mezzanine designs.

To do: add image and description of functionality.

Design files are here, the schematic is here.

12.2 BaseMod

BaseMod is a base-band input/output mezzanine. Design files are here, the schematic is here.

NB: 12/2017 Allaki was renamed BaseMod.

To do: add image

12.2.1 Outputs

BaseMod provides two independent RF outputs, featuring:

- **Bandwidth**: 10MHz - 4GHz (upper frequency is limited by several different components)
- **Max output power**: ?dBm (limited by ?).
- **Output filters**: either 3 Mini-Circuits FV1206 series filters, or a user-definable discrete 9-pole discrete-element filter using 0402 components.
- **Low phase noise amplifier**: Mini-Circuits ERA-4XSM+); 14.2dBm gain at 1GHz.
- **Digitally programmable attenuator**: HMC542BLP4E; 0dB to 31.dB in 0.5dB steps; controllable in real-time.
- **Fast, high-isolation RF switch**: HMC349LP4C; 67dB isolation at 1GHz; controllable in with real-time control.
- **Power detector**: AD8363ACPZ on switch "off" port for monitoring and power levelling.
- **Optional isolation of output grounds**: to avoid ground loops, achieved by fitting capacitors and washers.

12.2.2 Inputs

BaseMod provides two independent inputs, each of which can be configured (component placement) as:

1. Direct feed to ADC via ADA4927-1 buffer for maximum bandwidth
2. Low-noise programmable gain instrumentation amplifier (AD8253) front-end

- **Bandwidth**: DC-300kHz
- **Input ranges**: ±0.1V, ±1V, ±10V
- **Fully differential inputs**: 100k between each input signal and ground and the circuit ground
- **Filters**: Common-mode and differential mode filtering of RF interference for optimum DC precision
- **Input protection**: diodes between each input and the supply rails for maximum ruggedness
- Supports both high-speed input directly coupled into a high-speed pre-amp, and low-frequency inputs using a variable-gain instrumentation amplifier (choice by component selection). Pull details from #81
- Instrumentation amp: gain, filters, etc.

12.3 MixMod

MixMod is an up-converting mezzanine, using an analogue IQ mixer to mix the input and output RF signals with a LO supplied by Sayma.

The LO provided by Sayma should be a 3V3 PECL square-wave.

12.3.1 Outputs

MixMod provides a single RF output between 2.5GHz and 3.5GHz, produced by mixing two DAC channels with a LO supplied by Sayma. Other than the IQ mixer, the output signal-chain is identical to BaseMod.

12.3.2 Inputs

MixMod's two inputs can either be operated in baseband or downconversion mode (selectable by component choice). In baseband mode, the inputs function identically to BaseMod's. In downconversion mode, a single SMA input feeds the RF port on an IQ mixer to produce a pair of baseband signals, which then feed the two signal chains.

13 General Specification

13.1 Mechanical

- Board size
- Mounting holes
- SMA locations and pns
- Connectors

13.2 Electrical

- Suggest ADL5375 IQ modulator. Good intrinsic carrier/sideband rejection, relatively low temp coefficients, sufficient IF bandwidth, good I/Q linearity. This is the chip used in the NIST Magtrap drive system.
- Signal levels etc

14 Clock distribution mezzanine

14.1 Overview

Clock mezzanines generate high-quality RF/microwave signals for use as data converter clocks and local oscillators (LOs) by phase-locking low-noise VCOs to a supplied reference source. They mount on a suitable carrier, such as `Baikal`, which supplies global signals to all RTMs in a uTCA.4 rack, or to an individual RTM, such as `Sayma`, for local frequency generation.

14.2 Features and specification

- Reference input from carrier PCB via 2xSMPs, typically 100MHz AC-coupled differential 3V3 PECL square-wave.
- Two independent phase-locked loops (PLLs), one typically used as one data converter clock (CLK) and one as a high-frequency reference/local oscillator (REF_LO). Outputs are AC coupled 3V3 PECL square-waves provided as differential signals over 2xSMPs.
- PLL lock indicators accessible via TTLs
- PLL multiplication factors (output frequencies) accessible from carrier via I2C
- Auxiliary CLK input from MMCX connector on top of PCB with isolated ground. Typical input is 50Ohm single-ended, +10dBm. Switching between on-board PLL and auxiliary input using integrated ultra-low noise clock mux controllable from carrier.
- Digital/power: copy from AFE specification, and specify pins for PLL locked indicators (high-locked) and mux

14.3 Mezzanines

14.3.1 Template mezzanine

Used for thermal and electrical testing of carriers, such as Sayma and Baikal, and as a template for designing clock mezzanines.

The design files are located in ARTIQ_ALTIUM/PCB_mezzanine_clock_template, the schematic is here.

14.3.2 Low phase noise clock mezzanine

An ultra-low noise, dual-output fixed-frequency signal generator.

The design files are located in ARTIQ_ALTIUM/PCB_mezzanine_clock, the schematic is here.

Specification:

- PLL: HMC440
- VCOs: Crystek CVCO55CC family of narrow-band VCOs

- Output range for 100MHz input: 400MHz to 3.2GHz, limited by available VCOs and HMC440 multiplication factor
- PLL multiplication factors (output frequencies) fixed by component section, but readable from carrier via I2C
- To do: measure long-term phase stability

15 Clock distribution

15.1 Crate clock distribution

The crate distributes a 100MHz clock on a RTM RF backplane. This clock is typically externally supplied from a high quality source, but it is desirable to include a 100 MHz oscillator on the MCH RTM and on the Sayma RTM the for turnkey/standalone operation (with limited timing performance).

In a multi-crate system, all crates need to receive the same 100MHz clock to support sample-accurate operation.

15.2 RTIO

Sayma and Metlino shall include a general purpose XO of e.g. 125MHz, connected to a general purpose FPGA clock input pin. This is a simple addition that make the boards a bit friendlier to developers. It also allows for debugging and bootstrapping of the clocks during development: This XO becomes necessary if we use a transceiver PLL chip that needs to be configured before it outputs a clock.

15.2.1 Metlino

In root mode, the Metlino receives the 100MHz clock and turns it into a 200MHz RTIO clock that it uses as reference clock for its DRTIO transmitters.

In satellite mode, the Metlino recovers the RTIO clock from the fiber.

The following clock resources should be available on Metlino to support this operation:

- Si5324 for 100->200MHz in root mode, and CDR jitter filtering in satellite mode.
- Si5324 free-running based on local XO for providing a CDR reference in satellite mode.

The Si5324 shall have its two clock outputs connected to a transceiver clock input (so that we can transmit back synchronously and at fixed latency) and to a general purpose clock input on the FPGA. Transceiver-fabric clock routing inside the FPGA is of poor quality, so we want to mitigate that.

The Metlino will be double-width and connected to its RTM to receive the 100MHz RTM clock (required in root mode).

15.2.2 Sayma

Sayma cards recover their RTIO clock from the backplane's transceiver link or - if they are stand-alone – from their SFP/SATA DRTIO transciever link. This requires the same hardware as the Metlino in root mode: Si5324 connected in the same way.

15.3 DRTIO

DRTIO (distributed real-time input/output) achieves three distinct things over a single high speed serial link:

- It transfers the RTIO clock
- It transfers the RTIO time. This means that it will designate a specific RTIO clock cycle as timestamp zero.
- It transfers data. Data consists of RTIO events (outputs or inputs) and low bandwidth non-realtime auxiliary traffic.

Note that the RTIO time (clock plus the cycle counter) is the primary and authoritative source of time in the ARTIQ tree. The RTIO clock is however not an extremely low noise clock that could serve as the sample clock in data conversion or as a base clock for picosecond level timestamping. Having another "better" clock do these tasks is not trivial since the alignment between such a sample clock and the RTIO clock is unknown. When data is transferred between the two clock domains it is undefined which RTIO cycle corresponds to which sample clock cycle.

15.4 JESD204 synchronization procedure

While JESD204B subclass 1 provides "fixed latency" for the data transfer between a converter (ADC or DAC) and the FPGA, this is fundamentally insufficient for DRTIO. We need more than just fixed latency. A JESD204B link has two deviceclocks: one for the converter and one for the FPGA. The SYSREF signal is used to designate which cycle of the faster of the two deviceclocks corresponds to the beginning of a cycle in the slower deviceclock. The slower deviceclock and SYSREF have an a priori unknown phase with respect to the RTIO clock.

Timestamping a certain sample to a specific RTIO cycle requires two things in addition to JESD204B subclass 1 deterministic latency:

- Reproducible alignment of the sample clock with the RTIO clock. This is guaranteed by fixed latencies in the DRTIO branch of the clocking (master oscillator -> MCH RTM -> Metlino -> AMC backplane DRTIO link -> Sayma AMC) and in the sample branch (master oscillator -> MCH RTM -> RF backplane -> Sayma RTM -> PLL -> clock distribution -> DAC/ADC). This also requires the backplane clock and the sample clock to be integer multiples of the RTIO clock.
- Reproducible alignment of SYSREF and the slower FPGA deviceclock to the RTIO clock. This is done actively.

The FPGA shall align SYSREF with designated RTIO clock edges. The alignment should be better than a DAC clock cycle and reproducible across reboots.

The FPGA first roughly aligns SYSREF within one cycle before a desired RTIO clock edge by asserting the synchronization signal of the clock chip, which resets its dividers. This alignment is optional and may have an uncertainty of several DAC clock cycles. It is only used to decrease the required scan range of the delay elements used in the next steps.

The FPGA then analyzes SYSREF by repeatedly sampling it with the RTIO clock while scanning a calibrated I/O input delay. This measures the SYSREF phase with a high precision.

The delay scan mechanism is limited by the resolution and stability of the scan element. The resolution must be significantly smaller than a DAC clock period. There are three delay elements available to perform the scan:

- IDELAYE3 in the FPGA. Uncertainty about PVT effects.
- Digital delay in the clock distribution chip. Infinite delay, low noise.
- Analog delay in the clock distribution chip (HMC704X only, not AD9516-1). Very fine and well calibrated, but too noisy to be used on a sample clock.

We plan to use the latter two elements for the scan.

The FPGA then rounds the phase to an integer multiple of sample clock cycles using previously stored fractional delay data (delay <- round(measured - fractional)) and stores the new fractional delay (fractional <- measured - delay). It now programs the digital phase shifters of the slower clocks (FPGA deviceclock and SYSREF) with the negative of the rounded delay value.

This technique can be implemented on the AD9154 FMC cards, using the digital delay of the AD9516-1 and IDELAYE3.

15.5 Sayma RTM clock chip connections

The HMC7044 has 14 outputs. We should use them for:

- DAC1 deviceclock
- DAC1 SYSREF
- DAC2 deviceclock
- DAC2 SYSREF
- ADC1 deviceclock
- ADC1 SYSREF
- ADC2 deviceclock
- ADC2 SYSREF
- FPGA SYSREF [with fine delay]
- FPGA MGT reference clock for DAC
- FPGA MGT reference clock for ADC
- additional outputs to FPGA, usable e.g. if we have problems with the recovered RTIO clock.

15.6 Clock constraints

15.6.1 Constraints

- t_RTIO = n * 1ns
  - period of the coarse RTIO clock
  - n integer to avoid rounding errors and beating between RTIO clock and user habit
  - n not necessarily a power of two
  - the same throughout the ARTIQ tree to avoid beating of channels
- t_DRTIO_link = n * 10 * t_RTIO with n being 1, 2, 4, 8
  - line period of the DRTIO link
  - due to 8b10b and parallel bus width
  - n not a power of two could work but looks impractical.
  - does not need to be the same n for each link
  - AMC backplane links can probably not to 10 GHz line rate but 5 GHz, fibers (SFP+) can
- t_SAWG_DATA = t_RTIO/

```
f_DAC/f_SAWG: {1, 2, 4, 8}
f_SAWG/f_RTIO: {1, 2, 4, 8}
f_RTIO/f_DRTIO: {10, 20, 40}
f_JESD_P/f_RTIO: {1, 2}
f_JESD/f_JESD_P: {40}
```

(GHz)	f_DAC	f_SAWG	f_JESD_P	f_JESD	f_RTIO	f_DRTIO
A	2.4	0.6	0.15	6	0.15	3
B	2	1	0.25	10	0.125	5
C	0.3	0.3	0.15	6	0.15	3

16 Clock distribution

16.1 Crate clock distribution

The crate distributes a 100MHz clock on a RTM RF backplane. This clock is typically externally supplied from a high quality source, but it is desirable to include a 100 MHz oscillator on the MCH RTM and on the Sayma RTM the for turnkey/standalone operation (with limited timing performance).

In a multi-crate system, all crates need to receive the same 100MHz clock to support sample-accurate operation.

16.2 RTIO

Sayma and Metlino shall include a general purpose XO of e.g. 125MHz, connected to a general purpose FPGA clock input pin. This is a simple addition that make the boards a bit friendlier to developers. It also allows for debugging and bootstrapping of the clocks during development: This XO becomes necessary if we use a transceiver PLL chip that needs to be configured before it outputs a clock.

16.2.1 Metlino

In root mode, the Metlino receives the 100MHz clock and turns it into a 200MHz RTIO clock that it uses as reference clock for its DRTIO transmitters.

In satellite mode, the Metlino recovers the RTIO clock from the fiber.

The following clock resources should be available on Metlino to support this operation:

- Si5324 for 100->200MHz in root mode, and CDR jitter filtering in satellite mode.
- Si5324 free-running based on local XO for providing a CDR reference in satellite mode.

The Si5324 shall have its two clock outputs connected to a transceiver clock input (so that we can transmit back synchronously and at fixed latency) and to a general purpose clock input on the FPGA. Transceiver-fabric clock routing inside the FPGA is of poor quality, so we want to mitigate that.

The Metlino will be double-width and connected to its RTM to receive the 100MHz RTM clock (required in root mode).

16.2.2 Sayma

Sayma cards recover their RTIO clock from the backplane's transceiver link or - if they are stand-alone – from their SFP/SATA DRTIO transciever link. This requires the same hardware as the Metlino in root mode: Si5324 connected in the same way.

16.3 DRTIO

DRTIO (distributed real-time input/output) achieves three distinct things over a single high speed serial link:

- It transfers the RTIO clock
- It transfers the RTIO time. This means that it will designate a specific RTIO clock cycle as timestamp zero.
- It transfers data. Data consists of RTIO events (outputs or inputs) and low bandwidth non-realtime auxiliary traffic.

Note that the RTIO time (clock plus the cycle counter) is the primary and authoritative source of time in the ARTIQ tree. The RTIO clock is however not an extremely low noise clock that could serve as the sample clock in data conversion or as a base clock for picosecond level timestamping. Having another "better" clock do these tasks is not trivial since the alignment between such a sample clock and the RTIO clock is unknown. When data is transferred between the two clock domains it is undefined which RTIO cycle corresponds to which sample clock cycle.

16.4 JESD204 synchronization procedure

While JESD204B subclass 1 provides "fixed latency" for the data transfer between a converter (ADC or DAC) and the FPGA, this is fundamentally insufficient for DRTIO. We need more than just fixed latency. A JESD204B link has two deviceclocks: one for the converter and one for the FPGA. The SYSREF signal is used to designate which cycle of the faster of the two deviceclocks corresponds to the beginning of a cycle in the slower deviceclock. The slower deviceclock and SYSREF have an a priori unknown phase with respect to the RTIO clock.

Timestamping a certain sample to a specific RTIO cycle requires two things in addition to JESD204B subclass 1 deterministic latency:

- Reproducible alignment of the sample clock with the RTIO clock. This is guaranteed by fixed latencies in the DRTIO branch of the clocking (master oscillator -> MCH RTM -> Metlino -> AMC backplane DRTIO link -> Sayma AMC) and in the sample branch (master oscillator -> MCH RTM -> RF backplane -> Sayma RTM -> PLL -> clock distribution -> DAC/ADC). This also requires the backplane clock and the sample clock to be integer multiples of the RTIO clock.
- Reproducible alignment of SYSREF and the slower FPGA deviceclock to the RTIO clock. This is done actively.

The FPGA shall align SYSREF with designated RTIO clock edges. The alignment should be better than a DAC clock cycle and reproducible across reboots.

The FPGA first roughly aligns SYSREF within one cycle before a desired RTIO clock edge by asserting the synchronization signal of the clock chip, which resets its dividers. This alignment is optional and may have an uncertainty of several DAC clock cycles. It is only used to decrease the required scan range of the delay elements used in the next steps.

The FPGA then analyzes SYSREF by repeatedly sampling it with the RTIO clock while scanning a calibrated I/O input delay. This measures the SYSREF phase with a high precision.

The delay scan mechanism is limited by the resolution and stability of the scan element. The resolution must be significantly smaller than a DAC clock period. There are three delay elements available to perform the scan:

- IDELAYE3 in the FPGA. Uncertainty about PVT effects.
- Digital delay in the clock distribution chip. Infinite delay, low noise.
- Analog delay in the clock distribution chip (HMC704X only, not AD9516-1). Very fine and well calibrated, but too noisy to be used on a sample clock.

We plan to use the latter two elements for the scan.

The FPGA then rounds the phase to an integer multiple of sample clock cycles using previously stored fractional delay data (delay <- round(measured - fractional)) and stores the new fractional delay (fractional <- measured - delay). It now programs the digital phase shifters of the slower clocks (FPGA deviceclock and SYSREF) with the negative of the rounded delay value.

This technique can be implemented on the AD9154 FMC cards, using the digital delay of the AD9516-1 and IDELAYE3.

16.5 Sayma RTM clock chip connections

The HMC7044 has 14 outputs. We should use them for:

- DAC1 deviceclock
- DAC1 SYSREF
- DAC2 deviceclock
- DAC2 SYSREF
- ADC1 deviceclock
- ADC1 SYSREF
- ADC2 deviceclock
- ADC2 SYSREF
- FPGA SYSREF [with fine delay]
- FPGA MGT reference clock for DAC
- FPGA MGT reference clock for ADC
- additional outputs to FPGA, usable e.g. if we have problems with the recovered RTIO clock.

16.6 Clock constraints

16.6.1 Constraints

- t_RTIO = n * 1ns
  - period of the coarse RTIO clock
  - n integer to avoid rounding errors and beating between RTIO clock and user habit
  - n not necessarily a power of two
  - the same throughout the ARTIQ tree to avoid beating of channels
- t_DRTIO_link = n * 10 * t_RTIO with n being 1, 2, 4, 8
  - line period of the DRTIO link
  - due to 8b10b and parallel bus width
  - n not a power of two could work but looks impractical.
  - does not need to be the same n for each link
  - AMC backplane links can probably not to 10 GHz line rate but 5 GHz, fibers (SFP+) can
- t_SAWG_DATA = t_RTIO/

```
f_DAC/f_SAWG: {1, 2, 4, 8}
f_SAWG/f_RTIO: {1, 2, 4, 8}
f_RTIO/f_DRTIO: {10, 20, 40}
f_JESD_P/f_RTIO: {1, 2}
f_JESD/f_JESD_P: {40}
```

(GHz)	f_DAC	f_SAWG	f_JESD_P	f_JESD	f_RTIO	f_DRTIO
A	2.4	0.6	0.15	6	0.15	3
B	2	1	0.25	10	0.125	5
C	0.3	0.3	0.15	6	0.15	3

17 Housekeeping Signals

coppied from AMC

17.1 sensors

Temperature:

No	Addr.	placement	Type	Accuracy
IC8	0x4B	NOR Flash	LM75	+/- 2
IC34	0x49	FPGA	LM75	+/- 2
IC35	0x4A	Under SFPs	LM75	+/- 2
IC36	0x4F	power section	LM75	+/- 2
IC37	0x24	middle of the board	MAX664A	+/- 1

All temperature sensors are tied tohether to one I2C bus - I2C_SENS.

Current:

No	Addr.	placement	Type	Accuracy
IC27	0x40	RTM_P12V0	INA219	+/- 0.2%
IC28	0x41	FMC_P12V0	INA219	+/- 0.2%

All current sensors are tied tohether to one I2C bus - PM_I2C.

17.2 Safety interlocks

TBD OVERTEMPn

18 Power

18.1 Power supply

Coppied from AMC

TBD voltage noise

The 12V power can be connected either from AMC connector or from Stand alone power supply connected to Molex Connector(39-28-1043).

GND	GND
+12	+12

Maximum board(AMC+RTM module) power consumption estimate to 3A @ 12V.

Note: *Please note that power consumption mostly depends from FPGA configuration.*

- Input voltage range: 10.8-13.2 [V]
- The board needs active cooling. Approx. 20CFM in 20 C air.

18.2 Power configuration

18.2.1 Power map

voltages and currents		
P0V9	0.9V	10A
P0V95	0.95V	31mA
P1V0	1.0V	3A
P1V2	1.2V	0.6A
P1V5	1.5V	7.5A
P1V8	1.8V	1.6A
P3V3	3.3 V	2A
P3V3MP	3.3V	0.18A
P5V0	5.0V	0.5A

Maximum RTM voltages and currents		
P12V0	12V	3A
P3V3MP_RTM	3.3V	30mA

18.2.2 Exar parameters

Exar chip has 4 configurable outputs with configirable current limits. Channels 1, 3, 4 are power un on chip enable with 10ms delay. Channel 2 is power on 'EN_PSU_CH' signal.

18.2.3 Exar configuration

Exar chips are configured via I2C bus (MUX Port 5) or directly by connecting to W1 (call-out 28) header. For proper configuration **Exar Power Architect** in version **5.2-r1** is needed.
Exar Power Archtect 5.2-r1: https://www.exar.com/content/document.ashx?id=21632
Configuration files: https://github.com/m-labs/sinara/tree/master/EXAR_config
Datasheet:https://www.exar.com/ds/xr77129_1a_120514.pdf
Quick Start Guide: https://www.exar.com/files/powerxr/PA5-QSG_110_010614.pdf

Actual voltages and current consumption, temperature can be found in Chip Dashboard. There is also oportunity to adjust settings.

A Appendix

FPGA ball	FPGA signal	Signal on the board
A1	GND	GND
A2	MGTAVTT	MGTAVTT
A3	MGTPRXN1_216	RTM_FPGA_GTP_Rx1_N
A4	MGTPRXP1_216	RTM_FPGA_GTP_Rx1_P
A5	GND_3	GND
A6	MGTRREF_216	NC
A7	GND_4	GND
A8	GND_5	GND
A9	IO_L3N_T0_DQS_AD1N_15	CAL_ADC_MCLK1
A10	IO_L5N_T0_AD9N_15	CAL_ADC_CSn
A11	GND_1	GND
A12	IO_L7N_T1_AD2N_15	MEZZ3_IO3
A13	IO_L8P_T1_AD10P_15	MEZZ3_IO4
A14	IO_L8N_T1_AD10N_15	MEZZ3_IO5
A15	IO_L10N_T1_AD11N_15	MEZZ3_IO9
A16	VCCO_15	P3V3F
A17	IO_L15N_T2_DQS_ADV_B_15	HMC_SPI_SCLK
A18	GND_2	GND
B1	MGTPTXN3_216	RTM_FPGA_GTP_Tx3C_N
B2	MGTPTXP3_216	RTM_FPGA_GTP_Tx3C_P
B3	GND_7	GND
B4	MGTAVCC	MGTAVCC
B5	MGTREFCLK1N_216	CDR_CLK_CLEAN1_N
B6	MGTREFCLK1P_216	CDR_CLK_CLEAN1_P
B7	GND_8	GND
B8	GND_9	GND
B9	IO_L3P_T0_DQS_AD1P_15	CAL_ADC_MCLK2
B10	IO_L5P_T0_AD9P_15	CAL_ADC_SYNCn
B11	IO_L4N_T0_15	CAL_ADC_DIN
B12	IO_L7P_T1_AD2P_15	MEZZ3_IO2
B13	VCCO_15_1	P3V3F
B14	IO_L10P_T1_AD11P_15	MEZZ3_IO8
B15	IO_L9N_T1_DQS_AD3N_15	MEZZ3_IO7
B16	IO_L15P_T2_DQS_15	HMC_SPI_SDATA
B17	IO_L16N_T2_A27_15	USR_UART_N
B18	GND_6	GND
C1	MGTAVTT_1	MGTAVTT
C2	GND_11	GND
C3	MGTPRXN2_216	RTM_FPGA_GTP_Rx2_N
C4	MGTPRXP2_216	RTM_FPGA_GTP_Rx2_P
C5	MGTAVCC_1	MGTAVCC
C6	GND_12	GND
C7	GND_13	GND
C8	IO_L1N_T0_AD0N_15	HMC830_SPI_SEN
C9	IO_L2N_T0_AD8N_15	CAL_ADC_SCLK
C10	VCCO_15_2	P3V3F
C11	IO_L4P_T0_15	CAL_ADC_DOUT
C12	IO_L6N_T0_VREF_15	MEZZ3_IO1
C13	IO_L11N_T1_SRCC_15	MEZZ3_IO11
C14	IO_L9P_T1_DQS_AD3P_15	MEZZ3_IO6

C15	GND_10	GND
C16	IO_L16P_T2_A28_15	USR_UART_P
C17	IO_L18P_T2_A24_15	CLK_MEZZ_IO1
C18	IO_L18N_T2_A23_15	CLK_MEZZ_IO2
D1	MGTPTXN2_216	RTM_FPGA_GTP_Tx2C_N
D2	MGTPTXP2_216	RTM_FPGA_GTP_Tx2C_P
D3	GND_15	GND
D4	GND_16	GND
D5	MGTREFCLK0N_216	RTM_FPGA_GTP_CLKC_P
D6	MGTREFCLK0P_216	RTM_FPGA_GTP_CLKC_N
D7	GND_17	GND
D8	IO_L1P_T0_AD0P_15	HMC_SPI_GPIO
D9	IO_L2P_T0_AD8P_15	HMC_SPI_SDO
D10	IO_0_15	SI5324_INT_ALM
D11	IO_L6P_T0_15	MEZZ3_IO0
D12	GND_14	GND
D13	IO_L11P_T1_SRCC_15	MEZZ3_IO10
D14	IO_L12N_T1_MRCC_15	MEZZ3_IO13
D15	IO_L13N_T2_MRCC_15	MEZZ3_IO14
D16	IO_L14N_T2_SRCC_15	HMC7043_SLEN
D17	VCCO_15_3	P3V3F
D18	IO_L17N_T2_A25_15	CLK_MEZZ_IO0
E1	MGTAVTT_2	MGTAVTT
E2	GND_18	GND
E3	MGTPRXN0_216	RTM_FPGA_GTP_Rx0_N
E4	MGTPRXP0_216	RTM_FPGA_GTP_Rx0_P
E5	MGTAVCC_2	MGTAVCC
E6	GND_19	GND
E7	GND_20	GND
E8	CCLK_0	FPGA_CFG_CCLK
E9	GND_21	GND
E10	VCCO_0	P3V3F
E11	VCCBATT_0	P1V8F
E12	CFGBVS_0	NC
E13	IO_L12P_T1_MRCC_15	MEZZ3_IO12
E14	VCCO_15_4	P3V3F
E15	IO_L13P_T2_MRCC_15	NC
E16	IO_L14P_T2_SRCC_15	MEZZ3_IO15
E17	IO_L17P_T2_A26_15	SI5324_RST
E18	IO_L24N_T3_RS0_15	CLK_MEZZ_IO15
F1	MGTPTXN1_216	RTM_FPGA_GTP_Tx1C_N
F2	MGTPTXP1_216	RTM_FPGA_GTP_Tx1C_P
F3	MGTAVTT_3	MGTAVTT
F4	GND_24	GND
F5	MGTAVCC_3	MGTAVCC
F6	GND_25	GND
F7	VCCINT	P1V0
F8	TCK_0	TCK
F9	VCCINT_1	P1V0
F10	GND_22	GND
F11	VCCBRAM	VCCBRAM
F12	DONE_0	FPGA_CFG_DONE
F13	M2_0	NC

F14	IO_L22N_T3_A16_15	CLK_MEZZ_IO10
F15	IO_L21N_T3_DQS_A18_15	CLK_MEZZ_IO8
F16	GND_23	GND
F17	IO_L24P_T3_RS1_15	CLK_MEZZ_IO13
F18	IO_L19N_T3_A21_VREF_15	CLK_MEZZ_IO4
G1	GND_26	GND
G2	MGTAVTT_4	MGTAVTT
G3	MGTPRXN3_216	RTM_FPGA_GTP_Rx3_N
G4	MGTPRXP3_216	RTM_FPGA_GTP_Rx3_P
G5	GND_29	GND
G6	GND_30	GND
G7	GND_31	GND
G8	VCCINT_2	P1V0
G9	GND_32	GND
G10	VCCBRAM_1	VCCBRAM
G11	GND_27	GND
G12	VCCAUX	VCCAUX
G13	GND_28	GND
G14	IO_L22P_T3_A17_15	CLK_MEZZ_IO9
G15	IO_L21P_T3_DQS_15	CLK_MEZZ_IO7
G16	IO_L20N_T3_A19_15	CLK_MEZZ_IO6
G17	IO_L19P_T3_A22_15	CLK_MEZZ_IO3
G18	VCCO_15_5	P3V3F
H1	MGTPTXN0_216	RTM_FPGA_GTP_Tx0C_N
H2	MGTPTXP0_216	RTM_FPGA_GTP_Tx0C_P
H3	GND_35	GND
H4	GND_36	GND
H5	GND_37	GND
H6	GND_38	GND
H7	VCCINT_3	P1V0
H8	GND_39	GND
H9	VCCINT_4	P1V0
H10	GND_33	GND
H11	VCCBRAM_2	VCCBRAM
H12	GND_34	GND
H13	VCCAUX_1	VCCAUX
H14	IO_25_15	CLK_MEZZ_IO14
H15	VCCO_15_6	P3V3F
H16	IO_L20P_T3_A20_15	CLK_MEZZ_IO5
H17	IO_L23P_T3_FOE_B_15	CLK_MEZZ_IO11
H18	IO_L23N_T3_FWE_B_15	CLK_MEZZ_IO12
J1	GND_40	GND
J2	GND_44	GND
J3	GND_45	GND
J4	IO_L2N_T0_34	MEZZ4_IO2
J5	IO_L2P_T0_34	MEZZ4_IO1
J6	IO_0_34	RTM_FPGA_SCL
J7	GND_46	GND
J8	VCCINT_6	P1V0
J9	GNDADC_0	NC
J10	VCCADC_0	NC
J11	GND_41	GND
J12	VCCINT_5	P1V0

J13	GND_42	GND
J14	IO_L5P_T0_D06_14	REF_CLK_SRC_SEL_1V8
J15	IO_L2P_T0_D02_14	DAC2_SPI_SDIO
J16	IO_L2N_T0_D03_14	DAC2_SPI_SDO
J17	GND_43	GND
J18	IO_L3P_T0_DQS_PUDC_B_14	DAC2_SPI_SCLK
K1	IO_L3N_T0_DQS_34	MEZZ4_IO4
K2	IO_L3P_T0_DQS_34	MEZZ4_IO3
K3	IO_L4P_T0_34	MEZZ4_IO5
K4	GND_49	GND
K5	IO_L1N_T0_34	MEZZ4_IO0
K6	IO_L1P_T0_34	RTM_FPGA_SDA
K7	VCCINT_8	P1V0
K8	GND_50	GND
K9	VREFN_0	NC
K10	VP_0	NC
K11	VCCINT_7	P1V0
K12	GND_47	GND
K13	VCCAUX_2	VCCAUX
K14	GND_48	GND
K15	IO_L5N_T0_D07_14	NC
K16	IO_L1P_T0_D00_MOSI_14	NC
K17	IO_L4P_T0_D04_14	DAC2_IRQn
K18	IO_L3N_T0_DQS_EMCCLK_14	DAC2_SPI_CSn
L1	GND_51	GND
L2	IO_L4N_T0_34	MEZZ4_IO6
L3	IO_L5N_T0_34	MEZZ4_IO8
L4	IO_L5P_T0_34	MEZZ4_IO7
L5	IO_L6P_T0_34	MEZZ4_IO9
L6	VCCO_34	P3V3F
L7	GND_54	GND
L8	VCCINT_10	P1V0
L9	VN_0	NC
L10	VREFP_0	NC
L11	GND_52	GND
L12	VCCINT_9	P1V0
L13	GND_53	GND
L14	IO_0_14	DAC2_TXEN1
L15	IO_L6P_T0_FCS_B_14	DIO7
L16	VCCO_14	P1V8F
L17	IO_L1N_T0_D01_DIN_14	DAC2_TXEN0
L18	IO_L4N_T0_D05_14	REF_LO_CLK_SEL_1V8
M1	IO_L7N_T1_34	MEZZ4_IO12
M2	IO_L7P_T1_34	MEZZ4_IO11
M3	VCCO_34_1	P3V3F
M4	IO_L10P_T1_34	MEZZ1_IO1
M5	IO_L6N_T0_VREF_34	MEZZ4_IO10
M6	IO_L8P_T1_34	MEZZ4_IO13
M7	VCCINT_12	P1V0
M8	GND_57	GND
M9	DXN_0	FPGA_DXN
M10	DXP_0	FPGA_DXP
M11	VCCINT_11	P1V0

M12	GND_55	GND
M13	VCCAUX_3	VCCAUX
M14	IO_L8P_T1_D11_14	DIO5
M15	IO_L6N_T0_D08_VREF_14	DIO6
M16	IO_L7P_T1_D09_14	REC_CLOCK_P_1
M17	IO_L7N_T1_D10_14	REC_CLOCK_N_1
M18	GND_56	GND
N1	IO_L9P_T1_DQS_34	MEZZ4_IO15
N2	IO_L11N_T1_SRCC_34	MEZZ1_IO4
N3	IO_L11P_T1_SRCC_34	MEZZ1_IO3
N4	IO_L10N_T1_34	MEZZ1_IO2
N5	GND_61	GND
N6	IO_L8N_T1_34	MEZZ4_IO14
N7	GND_62	GND
N8	VCCINT_15	P1V0
N9	GND_63	GND
N10	VCCINT_13	P1V0
N11	GND_58	GND
N12	VCCINT_14	P1V0
N13	GND_59	GND
N14	IO_L8N_T1_D12_14	DIO4
N15	GND_60	GND
N16	IO_L9P_T1_DQS_14	DIO3
N17	IO_L9N_T1_DQS_D13_14	DIO2
N18	IO_L10P_T1_D14_14	DIO1
P1	IO_L9N_T1_DQS_34	MEZZ1_IO0
P2	GND_65	GND
P3	IO_L12N_T1_MRCC_34	MEZZ1_IO6
P4	IO_L12P_T1_MRCC_34	MEZZ1_IO5
P5	IO_L19N_T3_VREF_34	MEZZ2_IO4
P6	IO_L19P_T3_34	MEZZ2_IO3
P7	VCCO_34_2	P3V3F
P8	GND_66	GND
P9	VCCINT_17	P1V0
P10	PROGRAM_B_0	FPGA_CFG_PROGRAM_B
P11	VCCINT_16	P1V0
P12	GND_64	GND
P13	VCCAUX_4	VCCAUX
P14	IO_L12P_T1_MRCC_14	RTM_MASTER_AUX_CLK_P
P15	IO_L11P_T1_SRCC_14	REF_CLK_SRC_EXT_SEL_1V8
P16	IO_L11N_T1_SRCC_14	DAC_CLK_SRC_SEL_1V8
P17	VCCO_14_1	P1V8F
P18	IO_L10N_T1_D15_14	DIO0
R1	IO_L13N_T2_MRCC_34	MEZZ1_IO8
R2	IO_L13P_T2_MRCC_34	MEZZ1_IO7
R3	IO_L14P_T2_SRCC_34	MEZZ1_IO9
R4	VCCO_34_3	P3V3F
R5	IO_L21P_T3_DQS_34	MEZZ2_IO7
R6	IO_25_34	MEZZ2_IO15
R7	IO_L22P_T3_34	MEZZ2_IO9
R8	TMS_0	TMS
R9	GND_67	GND
R10	VCCO_0_1	P3V3F

R11	M1_0	NC
R12	M0_0	NC
R13	IO_L19P_T3_A10_D26_14	DAC1_SPI_SDO
R14	VCCO_14_2	P1V8F
R15	IO_L12N_T1_MRCC_14	RTM_MASTER_AUX_CLK_N
R16	IO_L14P_T2_SRCC_14	RTM_FPGA_LVDS1_P
R17	IO_L14N_T2_SRCC_14	RTM_FPGA_LVDS1_N
R18	IO_L15P_T2_DQS_RDWR_B_14	RTM_FPGA_USR_IO_P
T1	VCCO_34_4	P3V3F
T2	IO_L14N_T2_SRCC_34	MEZZ1_IO10
T3	IO_L17N_T2_34	MEZZ2_IO0
T4	IO_L17P_T2_34	MEZZ1_IO15
T5	IO_L21N_T3_DQS_34	MEZZ2_IO8
T6	GND_69	GND
T7	IO_L22N_T3_34	MEZZ2_IO10
T8	TDO_0	TDO
T9	TDI_0	TDI
T10	INIT_B_0	FPGA_CFG_INIT_B
T11	VCCO_14_3	P1V8F
T12	IO_L22P_T3_A05_D21_14	ADC2_CSB
T13	IO_L19N_T3_A09_D25_VREF_14	DAC1_SPI_SCLK
T14	IO_L13P_T2_MRCC_14	CDR_CLK_CLEAN2_P
T15	IO_L13N_T2_MRCC_14	CDR_CLK_CLEAN2_N
T16	GND_68	GND
T17	IO_L16P_T2_CSI_B_14	RTM_FPGA_LVDS2_P
T18	IO_L15N_T2_DQS_DOUT_CSO_B_14	RTM_FPGA_USR_IO_N
U1	IO_L15N_T2_DQS_34	MEZZ1_IO12
U2	IO_L15P_T2_DQS_34	MEZZ1_IO11
U3	GND_71	GND
U4	IO_L18P_T2_34	MEZZ2_IO1
U5	IO_L20N_T3_34	MEZZ2_IO6
U6	IO_L20P_T3_34	MEZZ2_IO5
U7	IO_L23P_T3_34	MEZZ2_IO11
U8	VCCO_14_5	P1V8F
U9	IO_L24P_T3_A01_D17_14	ADC1_CSB
U10	IO_25_14	ADC1_SDIO
U11	IO_L23P_T3_A03_D19_14	ADC1_SYNC
U12	IO_L22N_T3_A04_D20_14	ADC2_SDIO
U13	GND_70	GND
U14	IO_L20P_T3_A08_D24_14	DAC1_SPI_CSn
U15	IO_L17P_T2_A14_D30_14	DAC2_RESETn
U16	IO_L17N_T2_A13_D29_14	DAC1_TXEN1
U17	IO_L16N_T2_A15_D31_14	RTM_FPGA_LVDS2_N
U18	VCCO_14_4	P1V8F
V1	GND_72	GND
V2	IO_L16N_T2_34	MEZZ1_IO14
V3	IO_L16P_T2_34	MEZZ1_IO13
V4	IO_L18N_T2_34	MEZZ2_IO2
V5	VCCO_34_5	P3V3F
V6	IO_L23N_T3_34	MEZZ2_IO12
V7	IO_L24N_T3_34	MEZZ2_IO14
V8	IO_L24P_T3_34	MEZZ2_IO13
V9	IO_L24N_T3_A00_D16_14	ADC2_PDWN

V10	GND_73	GND
V11	IO_L23N_T3_A02_D18_14	ADC1_SCLK
V12	IO_L21P_T3_DQS_14	ADC2_SYNC
V13	IO_L21N_T3_DQS_A06_D22_14	ADC2_SCLK
V14	IO_L20N_T3_A07_D23_14	DAC1_IRQn
V15	VCCO_14_6	P1V8F
V16	IO_L18P_T2_A12_D28_14	DAC1_TXEN0
V17	IO_L18N_T2_A11_D27_14	DAC1_SPI_SDIO
V18	GND_74	GND