HCAL µTCA Trigger and Readout Module (uHTR)

06/04/14

Data Formats on Links

DRAM_FIFO Format for 1.6 Gbps Data

For the 1.6 Gbps data, the 32-bits of data on each Front-to-Back link has the following format:

- [7:0] DATA_e (MSB always zero for 1.6 Gbps)
- [9:8] CAPID_e
- [10] CAPID_ERROR_BIT_e, filled in by ZS algorithm in the back FPGA
- [11] OTHER_ERROR_BIT_e
- [12] GBT_ERROR_BIT_e
- [13] IS_ZERO_e, filled in by ZS algorithm in the back FPGA
- [15:14] RESERVED (0)
- [23:16] DATA_o
- [25:24] CAPID_o
- [26] CAPID_ERROR_BIT_o
- [27] OTHER_ERROR_BIT_o
- [28] GBT_ERROR_BIT_o
- [29] IS_ZERO_o
- [30] -> RESERVED (0)
- [31] FIRST_CHANNEL_OF_EVENT

The subscripts “e” and “o” refer to the even and odd samples in a sequence, starting from zero. Thus, “o”=”e”+1.

DRAM_FIFO Format for 4.8 Gbps Data

For the 4.8 Gbps data, the 32-bits of data on each Front-to-Back link has the following format:

- [7:0] DATA
- [13:8] RISING_EDGE_TDC
- [17:14] FALLING_EDGE_TDC
- [19:18] CAPID
- [20] CAPID_ERROR

- [21] LINK_ERROR
- [23:22] TDC_ERROR
- [29:24] Reserved (0)
- [30] FIRST_SAMPLE_OF_CHANNEL
- [31] FIRST_CHANNEL_IN_EVENT

Notes

uHTR Data Format

Each FED data payload will contain a number of uHTR blocks. These blocks are best understood as sequences of 16-bit words and consist of a header followed by channel data blocks followed by a trailer. Two versions of the payload have been present in the history of the uHTR firmware.

1. The first version is identical to the VME HTR data format, though with a reduced header size (16 => 12 bytes) and the trailer significantly modified.
2. The second version is based on the 64-bit header concept for production AMC13 firmware, recast into 16-bit format for convenience of understanding.

Header v1

Index	**15**							**8**	**7**							**0**
0	Reserved for DCC								EvN[7:0]							
1	EvN[23:8]															
2	1	Space reserved for flags (used in DQM, debugging)														
3	OrN[4:0]					ModuleId[10:0]										
4	FormatVer[3:0] = 0x8				BcN[11:0]											
5	Reserv(0)	N(Payload Words)														

Header v2

Index	**15**							**8**	**7**							**0**
0	Data_Length64[15:0]															
1	BcN[11:0]														Data_Length64[19:16]	
2	EvN[15:0]															
3	(Filled in by AMC13)								EvN[23:16]							
4	Reserved (0)				SlotId[3:0]				CrateId[7:0]							
5	OrN[15:0]															
6	PayloadFormat=1				EventType				Firmware Flavor[7:0]							

Index	15							8	7							0
7	FirmwareVersion[19:16],FirmwareVersion[13:8],FirmwareVersion[5:0]															

I

In the AMC13 specification, word 4 is called "BoardId[15:0]" and is copied to the overall AMC13 header.

Channel Data

The channel data comes in several flavors, depending on whether the channel has a single or dual TDC functionality.

Flavor=0 (or 1)

HB/HE QIE/TDC data, one 6-bit TDC hit-per-channel

It is suggested that flavor=1 could be used for channel data which would have been zero-suppressed but has been passed through under "Mark-and-Pass" behavior.

Index	15							8	7							0
n	1	Flavor[2:0]=0			ErrF[1:0]		CapId0[1:0]		ChannelId[7:0]							
n+1	0	SOI	TDCData[5:0] (TS 0)						QIEData[7:0] (TS 0)							
...	0	SOI	TDCData[5:0] (TS 1)						QIEData[7:0] (TS 1)							

- CapId0[1:0]: capacitor id for the first sample. The CapIds are assumed to rotate (0->1->2->3) through the rest of the samples, unless otherwise indicated by the error field
- ErrF[1:0]: Encodes errors observed for this channel (0=none, 1=capid misrotation, 2=link error)
- SOI: high for the "sample-of-interest", zero for all other samples. Indicates same information as previous "presamples" field

Flavor=2 (or 3)

HF Data with rising and falling edge TDC values (generally fewer TS/readout than HB/HE). In this case, two 16-bit words are required for each time sample.

It is suggested that flavor=3 could be used for channel data which would have been zero-suppressed but has been passed through under "Mark-and-Pass" behavior.

Index	15							8	7							0
n	1	Flavor[2:0]=2			ErrF[1:0]		CapId0[1:0]		ChannelId[7:0]							
n+1	0	0	SOI	OK	Resv (0)				QIEData[7:0] (TS 0)							
n+2	0	1	Resv (0)		TDCFall[5:0] (TS 0)					TDCRise[5:0] (TS 0)						
n+3	0	0	SOI	OK	Resv (0)				QIEData[7:0] (TS 1)							
n+4	0	1	Resv (0)		TDCFall[5:0] (TS 1)					TDCRise[5:0] (TS 1)						

Flavor=4

Trigger Data

Index	**15**							**8**	**7**							**0**
n	1	Flavor[2:0]=4			ErrF[1:0]		Reserved(0)		ChannelId[7:0]							
n+1	0	SOI	OK	TPGData[12:0] (TS 0)												
...	0	SOI	OK	TPGData[12:0] (TS 1)												

Flavor=5

Pre-upgrade HTR data (7-bit QIE)

In this case, the HTR header and trailer may be identical to the 2011 HTR data format. Only even numbers of samples are allowed – expected number is 6-10 samples generally.

Index	**15**							**8**	**7**							**0**
n	1	Flavor[2:0]=5			ErrF[1:0]		CapId0[1:0]		ChannelId[7:0]							
n+1	0	QIE Sample 1							0	QIE Sample 0						
n+2	0	QIE Sample 3							0	QIE Sample 2						
n+3	0	QIE Sample 5							0	QIE Sample 4						

Flavor=7

Technical data. This flavor is used to encode technical, changing data packets which hold information such as link status information, etc. This information for use by DQM only, and would not be unpacked into “detector-geometry” oriented information.

One class of technical data (TechnicalDataType=0) is simply padding words as needed to achieve 64-bit alignment for the overall uHTR payload.

Index	**15**							**8**	**7**							**0**
n	1	Flavor[2:0]=7			TechnicalDataType[3:0]				TechnicalChannelId[7:0]							
n+1	0	Technical Payload (depends on DataType)														
...	0	Technical Payload (depends on DataType)														

Trailer

The trailer is made shorter than the current HTR trailer to reduce the overall data volume slightly. The header+trailer combination is 128 bits (two 64-bit words), so alignment need only be considered within the data portion itself.

Index	15							8	7							0
0	CRC[15:0]															
1	EvN[7:0]								Zeroes (possibly overwritten by DTC)							

Mezzanine Cards

Formally, a mezzanine cards should be parallel to the base board it is attached to. This is not the case for the sub-assemblies on the uHTR, which are perpendicular to the base board. However, the term “power module” is highly confusing in the µTCA context, as the primary low voltage supply in a µTCA crate is through a unit called the power module. To limit confusion, we have decided to call the sub-assemblies “mezzanines”.

General Characteristics

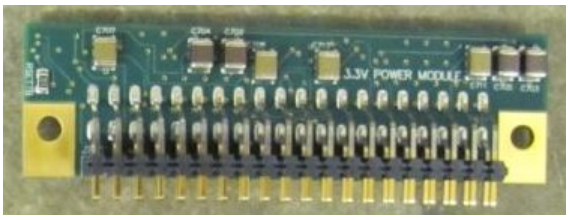
Except for the MMC, the rest of the mezzanines contain an EEPROM (either 24AA02E48T or 25AA02E48T). This 2kbit (256 byte) EEPROM is pre-programmed with a unique MAC address at the factory. The top half of the EEPROM is made non-reprogrammable, so the MAC address is protected, the bottom half is writable. During commissioning of each mezzanine, additional information is programmed into the EEPROM. The EEPROM should not then be modified during operations.

The structure of this data is as follows:

```
struct EEPROM_data {
  uint8_t data_format_version; // this is version 1
  uint8_t mezz_type_code;      // see codes in the sections below
  uint8_t mezz_subtype_code;   // see codes in the sections below
  uint8_t mezz_type[16];       // String version of mezzanine type+sub type
                               // (zero –terminated)
  uint8_t serial_number[2];    // construction serial number ([0]+[1]*256)
  uint8_t manu_date[11];       // DAY-MON-YEAR (zero-terminated)
  uint8_t manu_site[16];       // Site of manufacture (zero-terminated)
  uint8_t manu_tester[16];     // Name of tester (zero-terminated)
  uint8_t test_release[8];     // Release version of test code used
  uint8_t notes[56];           // String notes field in this version,
                               //  available for future use if needed
};
```

This structure is aligned at address zero of the EEPROM.

Power Mezzanine



The primary power mezzanines are designed to convert up to 20W of power from the bulk 12V input power, either as 20A of 1V or 6A of 3.3V power. Each mezzanine contains two LTM4601AV µmodule power converters, one operating as a slave to the other. The LTM4601AV supports voltage-margining, which raises or lowers the voltage around the nominal voltage. This variation is useful to test aging effects and identify possible weak components or boards. On the power mezzanines, the voltage margin is set as ±5%.

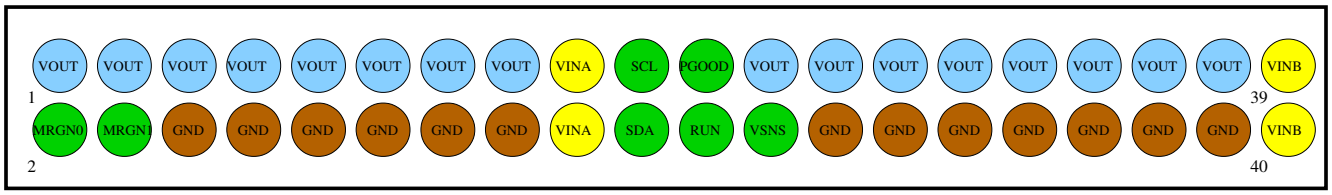
The output voltage is set by a feedback resistor, allowing the same mezzanine to provide any of many voltages by an appropriate resistor setting. Two particular values are used in the uHTR:

- 3.3V power mezzanine
  - R_{SET} = 6.62 k
  - `mezz_type_code`=1
  - `mezz_subtype_code`=3
  - `mezz_type` = “PM3.3”
- 1V power mezzanine
  - R_{SET} = 45 k
  - `mezz_type_code`=1
  - `mezz_subtype_code`=1
  - `mezz_type` = “PM1.0”

The power mezzanine supports an I2C bus which contains the following devices:

- Address 1010000 : ID EEPROM
- Address 0100000 : PCA9534BS3 8-bit GPIO device which allows control over the margining functionality (bits 0 and 1), reading the margin bits (bits 2 and 3), and reading the power-good signal (bit 7)
- Address 1101000 : MCP3426A0-E/MC two-channel ADC. Channel 1 is attached to a TMP36GRT temperature sensor and channel 2 to the output voltage of the µmodules.

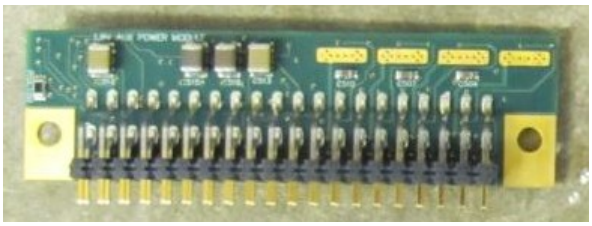
The mezzanine connector is a 40-pin, dual-row 0.1” rectangular connector.



Pins	Function
VOUT	Output voltage
VINA, VINB	Input (12V) power to the master (A) and slave (B) micromodules
SDA/SCL	(I/O) I2C control lines
MRGN0, MRGN1	(Input) Set MRGN0=3V, MRGN1=GND for margin down Set MRGN0=GND, MRGN1=3V for margin up Any other values, provide nominal voltage
PGOOD	(Output) Open-drain, active-high signal that the power is good. On-mezzanine pull-up to 5V through a 100k resistor.
RUN	(Input) Active high (2V to 5V) control to turn on the power converters. Pulled high on the mezzanine through a 100k resistor
VSNS	(Input) Sense input to be tied to the voltage plane on the carrier board

The testing program for the power mezzanine produces (what).

Auxiliary Power Mezzanine



The auxiliary power mezzanines are designed to provide a range of separate voltages at a lower total current than the power mezzanine. Each mezzanine contains one LTM4601AV µmodule power converter and eight linear regulators operating in pairs. The LTM4601AV supports voltage-margining, which raises or lowers the voltage around the nominal voltage, set to ±5% on the LTM4601AV. The linear regulator voltages are set proportionally to the primary regulator, so they follow the margin variations.

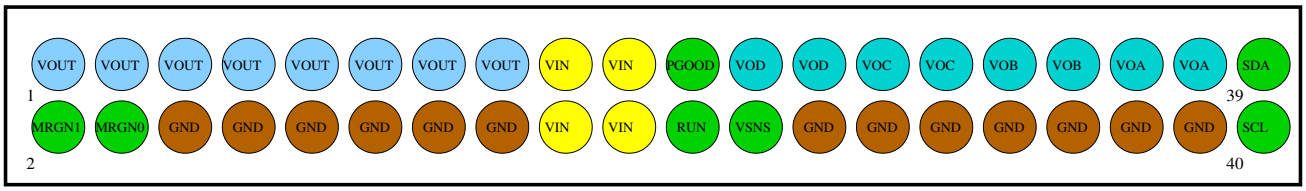
There are two APMs used on the uHTR, one where the µmodule provides 2.5V power and one where it provides either 1.8V or 1.5V. In both cases, the linear regulators provide two independent sources of 1V and two of 1.2V power. These lower voltage, lower current supplies are used for supplying the PLL and gigabit transceiver portions of the FPGAs. The 1.8V or 1.5V APM is used for the front FPGA, where a larger current is required – the use of the lower voltage as the supply to the linear regulators saves power and heating of the linear regulators. The 2.5V APM primary output is used in several places on the board and the linear outputs power the back FPGA, which has fewer GTX blocks.

- 2.5V auxillary power mezzanine
  - `mezz_type_code`=2
  - `mezz_subtype_code`=2
  - `mezz_type` = “APM2.5”
- 1.8V auxillary power mezzanine
  - `mezz_type_code`=2
  - `mezz_subtype_code`=1
  - `mezz_type` = “APM1.8”
- 1.5V auxillary power mezzanine
  - `mezz_type_code`=2
  - `mezz_subtype_code`=5
  - `mezz_type` = “APM1.5”

The power mezzanine supports an I2C bus which contains the following devices:

- Address 1010000 : ID EEPROM
- Address 0100000 : PCA9534BS3 8-bit GPIO device which allows control over the margining functionality (bits 0 and 1), reading the margin bits (bits 2 and 3), and reading the power-good signal (bit 7)
- Address 1101000 : MCP3426A0-E/MC two-channel ADC. Channel 1 is attached to a TMP36GRT temperature sensor and channel 2 to the output voltage of the µmodule.

The mezzanine connector is a 40-pin, dual-row 0.1” rectangular connector.



Pins	Function
VOUT	Primary output voltage (2.5V or 1.8V)
VOA	Secondary output voltage (1.2V)
VOB	Secondary output voltage (1.2V)
VOC	Secondary output voltage (1.0V)
VOD	Secondary output voltage (1.0V)
VIN	Input (12V) power
SDA/SCL	(I/O) I2C control lines
MRGN0, MRGN1	(Input) Set MRGN0=3V, MRGN1=GND for margin down Set MRGN0=GND, MRGN1=3V for margin up Any other values, provide nominal voltage
PGOOD	(Output) Open-drain, active-high signal that the power is good. On-mezzanine pull-up to 5V through a 100k resistor.
RUN	(Input) Active high (2V to 5V) control to turn on the power converters. Pulled high on the mezzanine through a 100k resistor
VSNS	(Input) Sense input for the primary output (2.5V/1.8V) to be tied to the voltage plane on the carrier board

The testing program for the power mezzanine produces (what).

FLASH Mezzanine

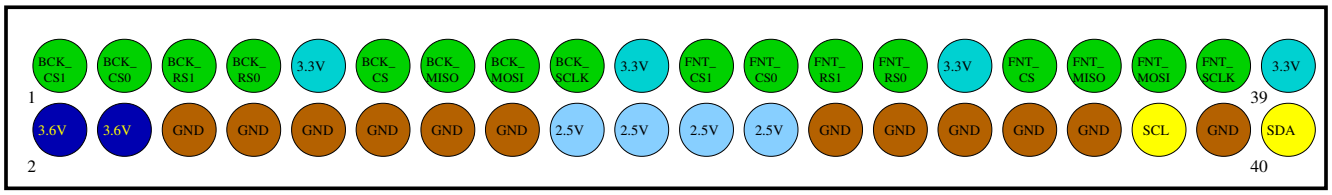
The FLASH mezzanine contains the SPI FLASH modules which contain the primary and backup FPGA configurations and which can also contain user data. The mezzanine can contain a total of eight 128Mb (M25P128) FLASH memories. The memories are accessed via SPI interfaces. There are separate SPI interfaces for the two FPGAs on the uHTR.

- FLASH Mezzanine
  - `mezz_type_code`=3
  - `mezz_subtype_code`=(# of FLASH chips for back FPGA)+8*(# of FLASH chips for the front FPGA)
  - `mezz_type` = "FLASH[(# back),(# front)]"

The power mezzanine supports an I2C bus which contains a single device:

- Address 1010000 : ID EEPROM

The mezzanine connector is a 40-pin, dual-row 0.1" rectangular connector.



Pins	Function
3.3V	3.3V power supply, used for on-board power purposes. If this is connected to the management power of the AMC card, the 3.6V pin should be supplied by payload power.
2.5V/VIO	2.5V power supply, used for I/O on and off the board (could be a different voltage in a different application)
3.6V	3.6V power which is diode-reduced to 3.3V for on-board power purposes. Should be provided from payload power if 3.3V comes from management power.
SDA/SCL	(I/O) I2C control lines
BCK_CS1	(Input, active-low) SPI select line for the second "User" FLASH memory for the back FPGA. Only one of BCK_CS, BCK_CS0, BCK_CS1 should be active at a single time.
BCK_CS0	(Input, active-low) SPI select line for the first "User" FLASH memory for the back FPGA. Only one of BCK_CS, BCK_CS0, BCK_CS1 should be active at a single time.
BCK_CS	(Input, active-low) SPI select line for the FPGA configuration memories. The active configuration memory is chosen by the RS lines. Only one of BCK_CS, BCK_CS0, BCK_CS1 should be active at a single time.
BCK_RS0, BCK_RS1	(Input) Revision-select to pick the active FPGA configuration memory for either programming or FPGA configuration. In the production version of the mezzanine, BCK_RS1 is ignored.
FNT_CS1	(Input, active-low) SPI select line for the second "User" FLASH memory for the front FPGA. Only one of FNT_CS, FNT_CS0, FNT_CS1 should be active at a single time.

FNT_CS0	(Input, active-low) SPI select line for the first “User” FLASH memory for the front FPGA. Only one of FNT_CS, FNT_CS0, FNT_CS1 should be active at a single time.
FNT_CS	(Input, active-low) SPI select line for the FPGA configuration memories. The active configuration memory is chosen by the RS lines. Only one of FNT_CS, FNT_CS0, FNT_CS1 should be active at a single time.
FNT_RS0, FNT_RS1	(Input) Revision-select to pick the active FPGA configuration memory for either programming or FPGA configuration. In the production version of the mezzanine, FNT_RS1 is ignored.

CPLD Mezzanine

The CPLD mezzanine is designed to host a moderate-scale Xilinx CPLD and an EEPROM. On the uHTR, the CPLD mezzanine is used to configure the Silicon Labs clock multiplier chips automatically on power-on.

To do:

1. More description
2. Pinout
3. Usage plans

JTAG Mezzanine

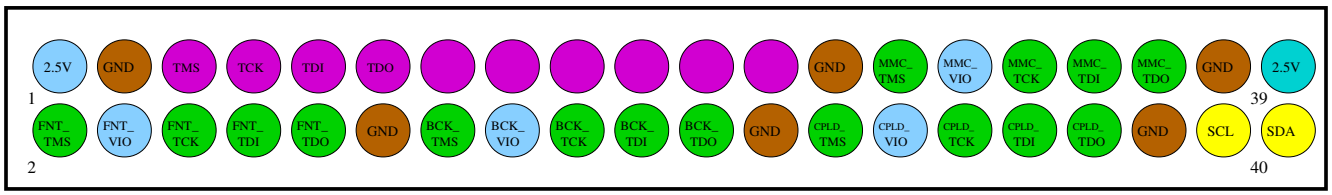
The JTAG mezzanine connects to the JTAG ports of the front and back FPGAs as well as the MMC microcontroller and the CPLD mezzanine. It also connects to the front-panel RJ45 connectors. Two versions of the JTAG mezzanine have been made. One is a simple implementation which provides connectors on the mezzanine for each of the JTAG chains. The second, which is the production version for the uHTR, provides an interface between the JTAG chains and the RJ45 connectors, allowing reprogramming of the FPGAs from the front-panel of a full uTCA crate.

- JTAG Mezzanine
  - `mezz_type_code`=4
  - `mezz_subtype_code`=1
  - `mezz_type` = “JTAG”

The power mezzanine supports an I2C bus which contains a single device:

- Address 1010000 : ID EEPROM

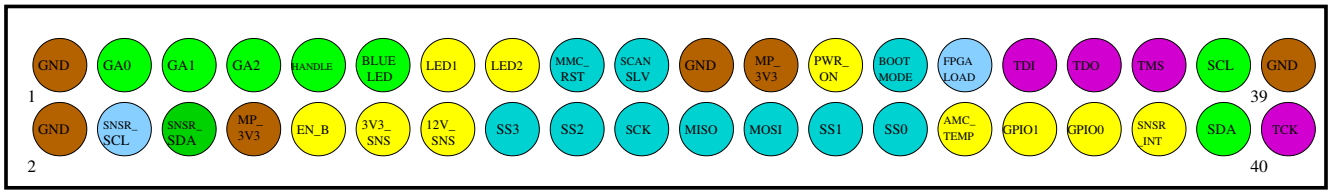
The mezzanine connector is a 40-pin, dual-row 0.1” rectangular connector.



Pins	Function
2.5V	2.5V power supply
***X*_VIO**	I/O voltage for the given JTAG chain
***X*_TMS, *X*_TCK, *X*_TDO, *X*_TDI**	JTAG signals for the given JTAG chain
SDA/SCL	(I/O) I2C control lines for the id EEPROM
TMS, TCK, TDO, TDI	Master JTAG signals (from the front panel)
Magenta Pins	Control pins connected to the front-panel JTAG pins

MMC Mezzanine

The MMC mezzanine (formally, this means “Mezzanine Management Controller Mezzanine”), provides the µTCA-standard control functionality required to identify the power requirements of the AMC and carry out the necessary house-keeping functions. The MMC design and base firmware was provided by the University of Wisconsin group. The uHTR MMC mezzanine is a re-formatting of the design into a compact board which can be easily reused on many AMC cards.



To do:

- Add pinout
- Discuss firmware