sTGC Trigger Data Serializer

J. Chapman, T. Dai, L. Guan, Z. Sang, J. Wang, B. Zhou, J. Zhu

University of Michigan, Ann Arbor, MI, 48109

1. Introduction

The sTGC trigger data serializer (TDS) ASIC chip is responsible for the preparation of trigger data for both pads and strips with the additional task of serializing data for transmission to the circuits on the rim of the nSW detector. The chip is designed to have two different configurations, one to deal with sTGC pads (denoted as “pad-TDS” here) and the other one to deal with sTGC strips (denoted as “strip-TDS” here). Physical pads are staggered by 1/2 pad in both directions in order to get even smaller area logical pads to reduce the amount of strip data to be transmitted on each bunch crossing, while strips are used to determine the sTGC segment direction with an angular resolution less than 1 mrad. Strip pulse heights are captured by the VMM front-end ASD chip and sent to the strip-TDS chip. Meanwhile discriminated pad signals are assigned to their bunch crossing, and the binary pattern of hit pads is serialized by the pad-TDS and sent to the pad tower electronics on the rim of the nSW. Three-out-of-four coincidences are made in pad towers in each sTGC quadruplet; these are then combined to choose a band of strips in each strip-TDS chip to be readout. The strip-TDS chip formats and serializes the pulse heights of the selected strips and the BCID and pad road information to the router board located on the rim of the nSW detector. Each router board receives the output from up to nine TDS chips and sends the data to USA15 via optical fibers. Hit centroids are found for each of the eight layers and sTGC track segments are built using centroids found from eight layers. The layout of the sTGC trigger logic is shown in Fig. 1.

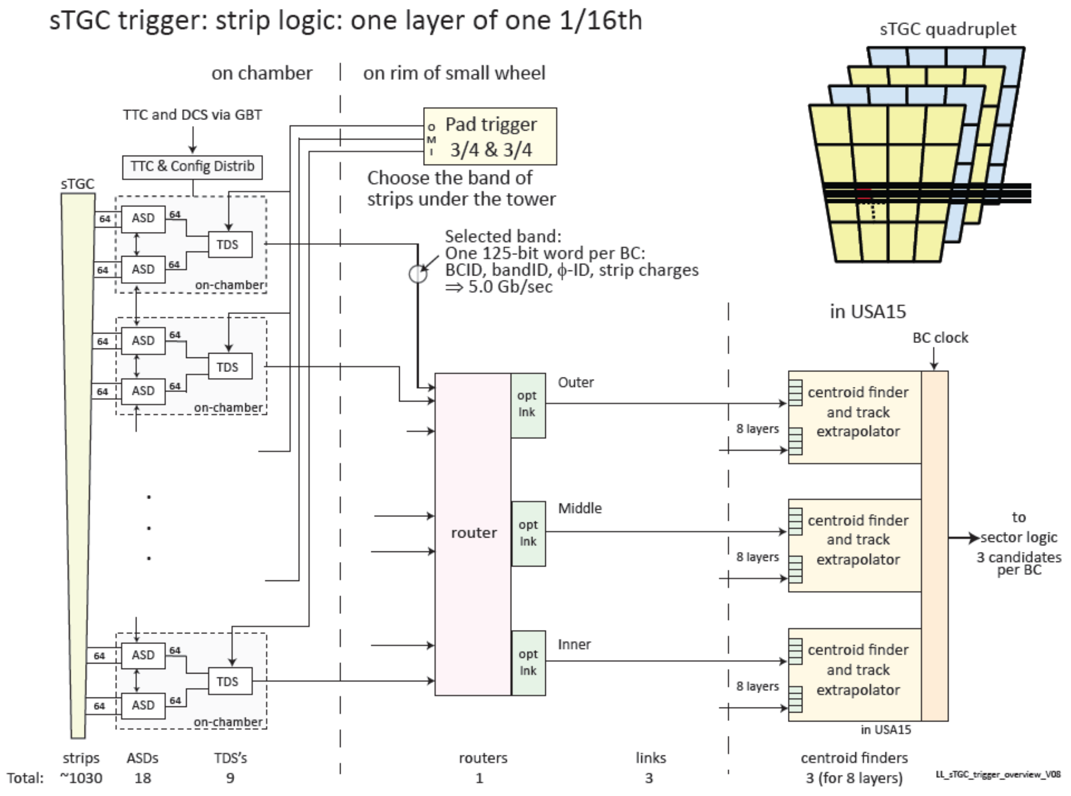
sTGC trigger: strip logic: one layer of one 1/16th
on chamber
on rim of small wheel
sTGC quadruplet
TTC and DCS via GBT
TTC & Config Distrib
Pad trigger 3/4 & 3/4
OMI
Choose the band of strips under the tower
sTGC
ASD
TDS
64
on-chamber
Selected band: One 125-bit word per BC: BCID, bandID, ϕ-ID, strip charges ⇒ 5.0 Gb/sec
in USA15
BC clock
router
opt lnk
Outer
Middle
Inner
8 layers
centroid finder and track extrapolator
to sector logic 3 candidates per BC
in USA15
strips
ASDs
TDS's
routers
links
centroid finders
Total:
~1030
18
9
1
3
3 (for 8 layers)
LL_sTGC_trigger_overview_V08

Figure 1. Layout of the sTGC trigger logic.

For each pad, we need to send out 1-bit to indicate whether the pad is fired or not; while for each strip, we need to send out 6-bit to indicate the total charge deposited on that strip. At the same time, we also send out the BCID, roadID and phiID for the pad road. Each pad road will cover an area of at most 14 strips, and we will send out the charge information on all 15 strips (with edge effect considered) underneath the road even though most of them will have zero charge.

The TDS chip will be implemented using the IBM 130 nm CMOS process and produced together with the VMM chips so that the fabrication of both chips can share wafers and masks. The technical challenging for the design of the TDS chip are: (1) high speed (need to serialize the strip data to the router board with an output rate of 4.8 Gbps); (2) low latency (need to have a latency of 100 - 125 ns after it receives the road information from the pad tower electronics); (3) large number of I/O pins (need to deal with 128 VMM channels with differential outputs and in total about 400 pins are expected); (4) low power consumption (need to have less than 1 watt power consumption per chip).

The connections, inputs and outputs for pad-TDS and strip-TDS can be summarized in Fig. 2.

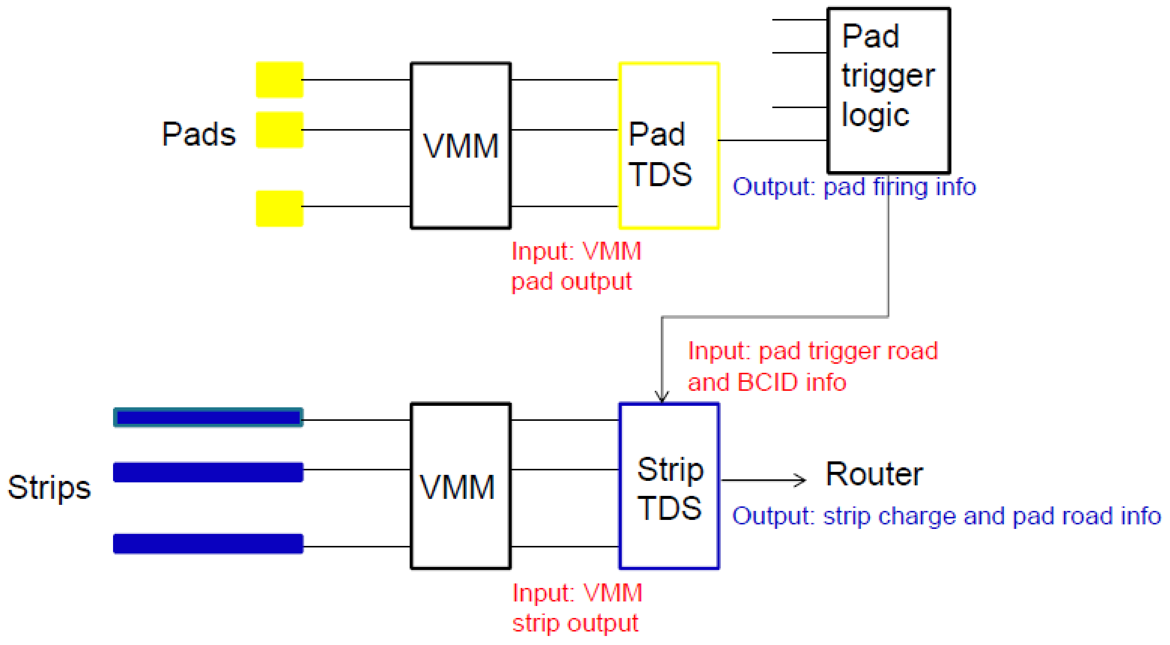

Figure 2: Inputs and outputs for pad-TDS and strip-TDS.

2. Strip TDS ASIC

2.1 Design

The architecture of the strip-TDS chip is shown in Fig. 3. The design can be divided into three major parts: VMM interface, Preprocessor, and Serialization. The VMM interface mainly reads in the VMM output data and stores it in buffers awaiting track road information from the pad trigger logic. The Preprocessor mainly performs the BCID and pad trigger matching and select strips within the pad road for transmission to the router. The Serialization part mainly prepares the trigger data and serializes it for transmission to the router.

For each strip, the VMM ASD chip provides a 6-bit measurement of the charge deposited that is sent serially to the strip-TDS. These 6-bits of charge follow the single bit that defines a strip with data. The signal from each VMM output channel is deserialized first and the pulse height information is stored in a ring register together with BCID. Time variations due to different strip and cable lengths will be compensated with programmable delay circuitry at each TDS with a precision of 3.125 to 6.25 ns. Time variations of the VMM bit clock will be compensated with programmable delay circuitry with a precision of 0.78 ns. Since the VMM timing window for the detector signal is likely to be 30 ns, two BCID counters are used with a relative time delay of 5 ns between their incoming clocks (the delay time is adjustable). The results from these two counters are compared, and the BCID from the first counter is used while a BCID flag is set to 0 or 1 depending on whether both counters provide the same BCID or not. The ring register stores the charge information for up to four hits per channel while waiting for pad trigger data from the pad tower electronics. The pad trigger decoder circuity decodes pad trigger data received and obtains BCID, band-ID, phi-ID and the starting strip address.

Given the size of logical pads, we will readout data from 15 strips starting from the starting strip. BCID and band-ID matching tasks are then performed and only matched fired strips within the pad trigger road are selected. Thirty-two 4-to-1 priority sequencers are used to read the data into FIFOs. Input channels to each sequencer are arranged so that adjacent fired strips from a single muon hit will be processed by several sequencers in parallel, and each sequencer only needs to deal with up to one fired channel. Data from all selected strips within the trigger road are serialized and send to the router board via twinax cables.

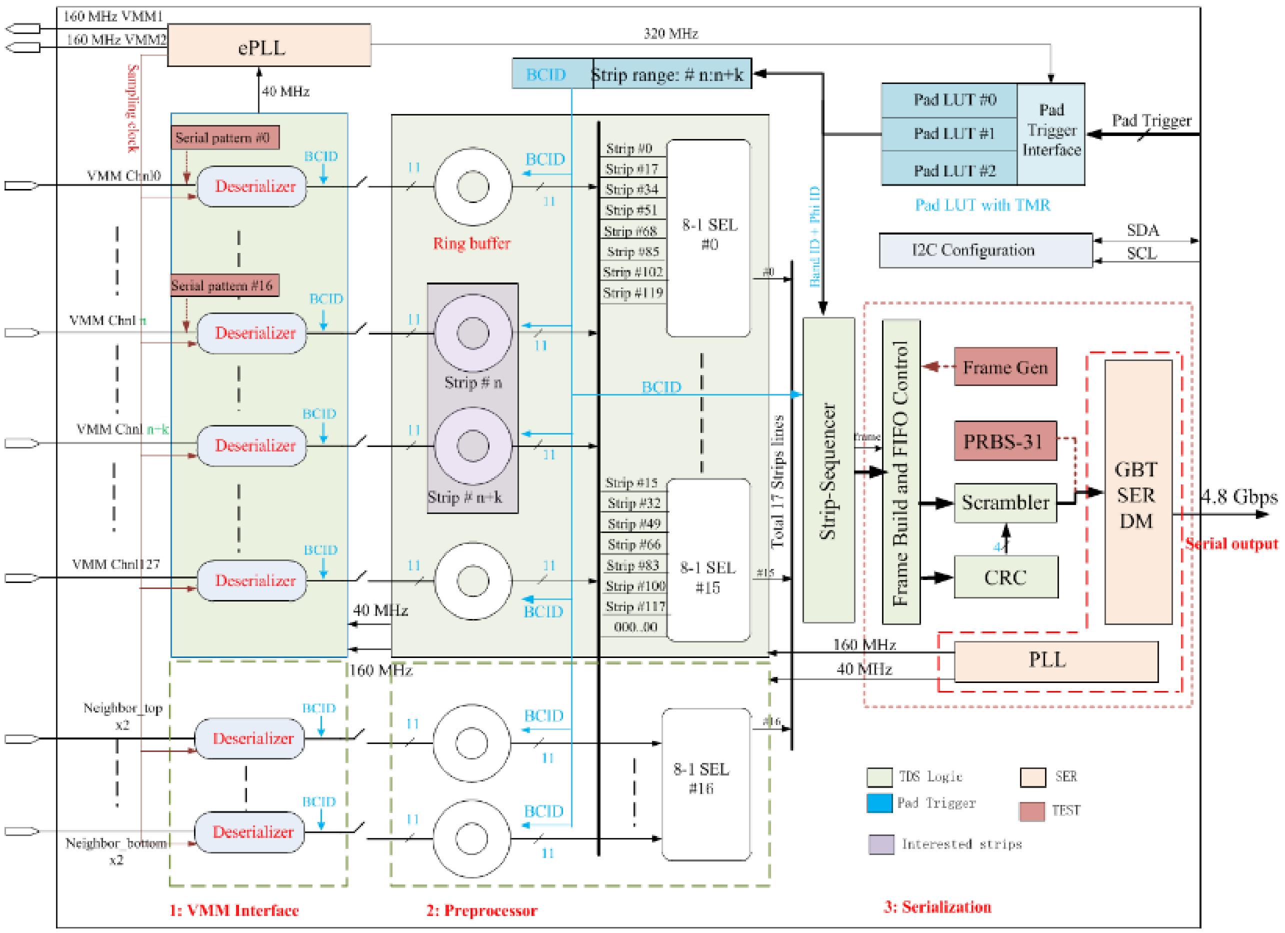

Figure 3: Layout of the strip-TDS chip.

2.2 Serializer-Core

The circuit for the final serialization needs to handle an output rate of 4.8 Gbps and has to be layout by hand, while the logic circuits before that can be implemented using the Verilog HDL code and later converted to silicon using the standard digital cells of the IBM 130nm CMOS process. The serializer-core circuit is a crucial component of the TDS chip and is modified from the existing CERN GBTx serializer. The layout of the CERN design is shown in Fig. 4a. It consists of a 120-bit input register, three 40-bit shift registers, a Phase-Lock Loop (PLL) with a 120-bit feedback divider, and a 3:1 high speed multiplexer. The serializer operation is based on the division of the 120-bit frame into three 40-bit words which are serialized at 1.6 Gbps and then time division multiplexed to form the final 4.8 Gbps serial bit stream. Triple Modular Redundancy (TMR) is used in the feed-back divider of the PLL to mitigate SEUs. To reduce the overall latency and to be consistent with the 160 MHz clock used for the whole chip, we modified the existing serializer to loading 30 bits at a rate of 160 MHz instead of loading 120 bits at a rate of 40 MHz. In order to have the fabrication together with the VMM chip, we also modified the metalization technology from LM62 to DM323. The layout of the modified serializer is shown in Fig. 4b. The serializer-core chip has been produced and tested. All preliminary results indicate that it meets our expectations. The eye diagrams for the serializer-core chip are shown in Fig. 5 at 4.5 Gbps and 5.4 Gbps. The latency is found to be less than 7 ns including all trace differences on the test board and FPGA buffer delay. The simulation indicates the latency for the serializer part is 4 ns.

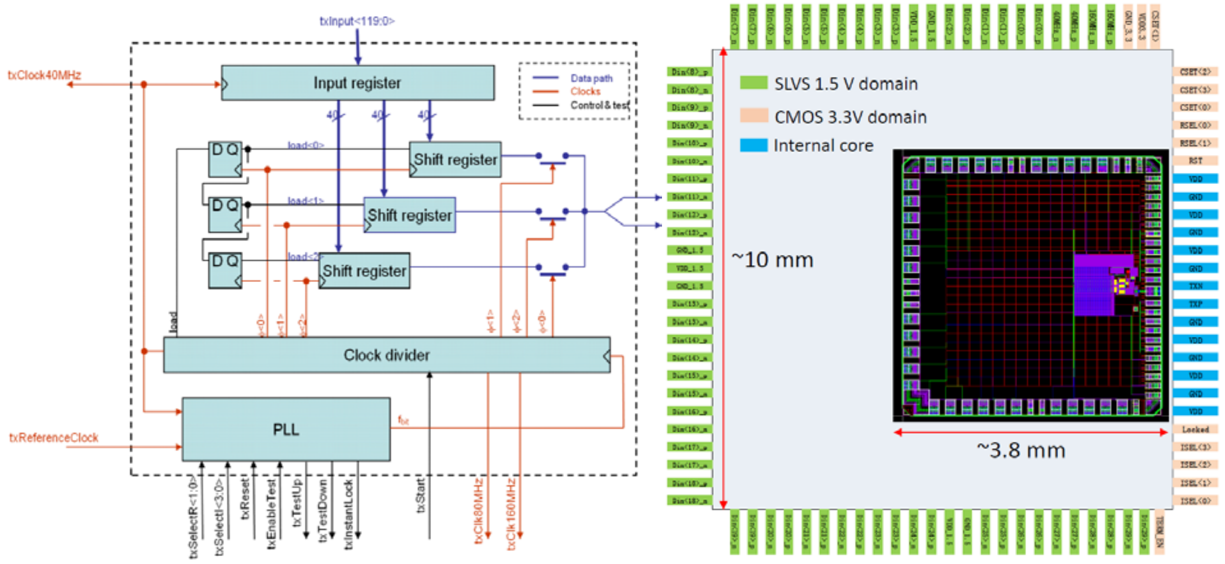

Figure 4: (a) Layout of the CERN GBTx serializer. (b) Layout of the serializer-core chip.

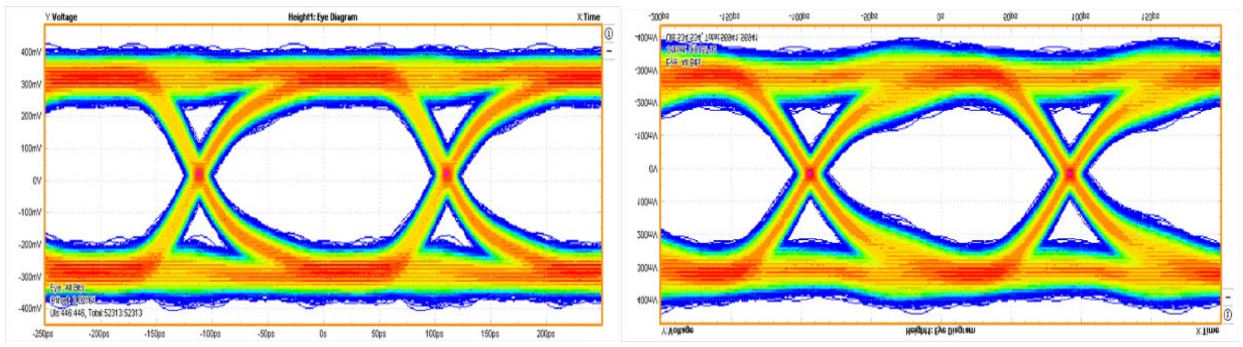

Figure 5: Eye diagrams of the serializer-core chip at 4.5 Gbps (Left) and 5.4 Gbps (Right).

2.3 Interface to the VMM chip

Each strip-TDS reads in 128 VMM channels plus two neighboring channels from each of the two adjacent TDSs in order to remove the TDS edge effect. The SLVS driver developed by CERN is used to handle the VMM output signal. The SLVS driver has an embedded 100-Ohm termination resistor, and can accept a wide common-mode voltage from 0.2 V to 1.2 V with a typical swing of ± 200 mV. The 6-bit charge from VMM is sampled with dual edges of a 160 MHz clock. The 160 MHz clock is provided by the ePLL circuit inside the strip-TDS, this will guarantee that the clock used by the TDS to sample the VMM outputs and the clock used by the VMM to send the output have the same phase. The 160 MHz clock can also be adjusted to different phases with a programmable phase shift at a step of 400 ps.

TDS will assign BCID based on the first edge from the VMM for the VMM strip signal peaking time. There will be a 12-bit BCID phase offset for each TDS that can be configured using the I2C interface. Since the signal routing lengths for all strips routed to the same strip-TDS have small variations, we assume no need to adjust the signal arrival time for different strips on one strip-TDS chip.

The matching window of each TDS is programmable, from 25 ns to 50 ns with a step of 6.25 ns, to compensate for the signal processing delays to and through the VMM chip. In case of the matching window is larger than 25 ns, a BCID extension flag will be assigned. For example, when using a 31.25 ns matching window, a VMM hit in the first 6.25 ns of the second BC will be assigned the current BC, but will also be given an extension flag indicating that it might also belong to the following BCID. There is a configuration bit that can decide if we want to send out hits with the current BCID together with the extension bit or only send out hits for the current BCID. This latter choice implies that hits in the extension time receive BCID+1 counter value.

For each of the first 15 strips, a VMM 6-bit charge serial pattern can be generated for self-testing purpose.

2.4 Interface to the Pad-trigger logic circuit

The pad-trigger logic circuit located on the rim receives the pad-TDS data and perform algorithm to find the region of interest. The resulting pad road information is sent to the strip-TDS at the rising edge of the global 40 MHz clock. The area covered by each road involves about 12 - 15 strips, and we allow the strips inside one pad road to be readout by two strip-TDSs. There are about 90 roads per sector and thus 7 bits are needed to indicate the global road ID. The pad-trigger logic circuit thus needs to send out a 8th bit to indicate which TDS to be readout. In total the pad-trigger logic needs to use 8 bits to strip-TDS for the road ID information. Pad road information from the pad trigger logic is delivered to each strip-TDS via two lines operating at 640 Mbps each. We assume 25 ns is used to send out the data, and thus 32 bits in total for these two lines. The pad road information includes 5 bits phi-ID, 8 bits band-ID, and 12 bits BCID. The arrangement of the 16 bit info for each line is listed below:

Line 0	“10” header	12 bits trigger BCID	2 spare bits
Line 1	“10” header	5 bits phi ID + 8 bits band ID	1 spare bit

In case there is no valid trigger info, the pad trigger logic will send out 0x8000 (hex) for each line. It will also be useful for the pad trigger logic first send out a few 0x800 for each line after powering up. This will help TDS to maintain the data synchronization. We assume the connection between TDS and the pad trigger logic for both lines are DC-coupled via LVDS. There are no needs for scrambler and descrambler.

Each TDS will have a LUT that contains the starting channel for at most 8 roads. The address for the element of each LUT is the 8 bits band-ID and the content is the 7 bits starting strip address. The size of the LUT is 15 bits width x 8 bits depth. TMR is used to reduce SEU. With the starting strip address, TDS will search for a range of 17 strips (including four neighboring channels), and match the pad trigger BCID with the BCID of all 17 buffered hits. If the matching window is set to 25 ns, exact BCID matching is performed; otherwise, the BCID flag will be taken in account. For example, if the pad trigger BCID is k, and one strip has BCID k+1 and a valid BCID flag, this case will also be considered as a valid match and the data from this strip also needs to be readout.

2.5 Interface to the router

Depending on the firing status of the 17 strips, either the first 15 strips or the last 15 strips will be selected for output. The total output data have a size of 120 bits. With header information include, the overall data that need to be transmitted are 150 bits. The output data includes BCID, band-ID, phi-ID, CRC and data from 15 strips. The total number of useful data bits is 120. We use 1010 for the overall header for both NULL and non-NULL packets, and use 10 to indicate non-NULL packet and 01 to indicate NULL packet. The total number of bits we need to send out is thus 150 and will be sent out with 5 packets with 30 bits per packet. The overall transmission time is 31.25 ns. The output data format is shown in Fig. 6(a). All bits except headers are scrambled (instead of 8b/10b conversion) to keep DC balance. We use a multiplicative (self-synchronizing) scrambler defined by a polynomial 1+X^39+X^58, the IEEE standard for 10 G Ethernet. The algorithm used for the scrambler is shown in Fig. 6(b). A 4-bit CRC with polynomial 0x9: 1 + x + x^4 is used. Even though 4 CRC bits are insufficient to correct errors for a total of 116 bits and will only be used to indicate a transmission error has occurred.

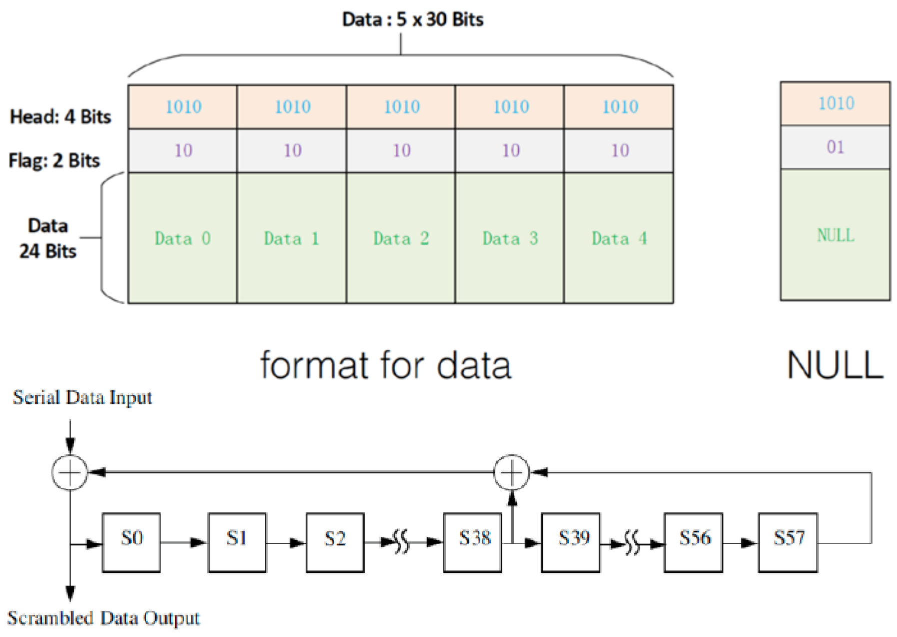

Figure 6: (a) Output format for the strip-TDS data. Five packets with 30 bits per packet are used to send out non-NULL data. (b) Algorithm used for the scrambler.

The simulated overall latency (time difference when the starting strip address has been decoded and when the first bit is sent out by the serializer) is found to be about 50 ns. We expect the final chip to have a slightly longer latency.

To test the serial link performance, the TDS can be configured to output PRBS-31. This is useful for eye-diagram observation and bit error evaluation of the link.

3. Pad TDS ASIC

The architecture of the pad-TDS chip is shown in Fig. 7. For each pad, the information from the VMM is nearly identical with that for strips with the difference being the presence of 6 additional bits of charge for the strips. A similar deserializer circuit is designed to obtain the 1-bit yes/no information from each pad. These information will then be scrambled and serialized to the pad trigger logic.

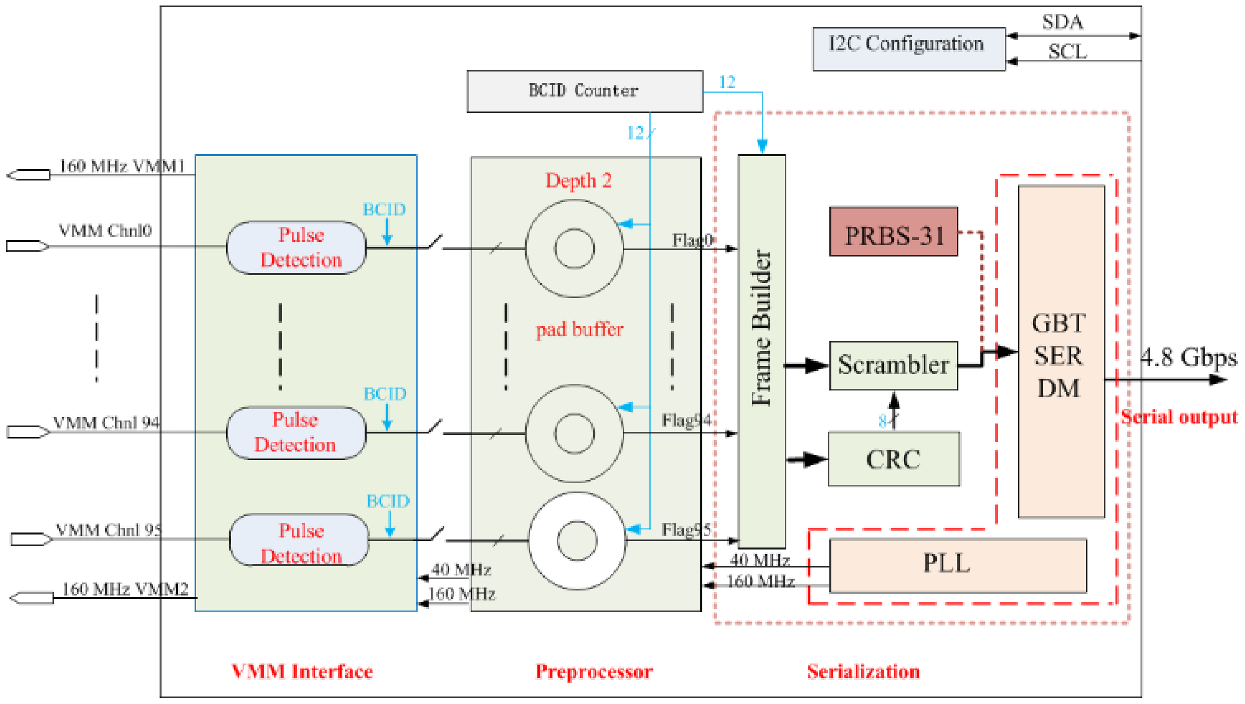

Figure 7: Layout of the pad-TDS chip.

The pad road data sent by the pad tower electronics to the strip-TDS are transmitted by four serial lines using 160 MHz clock and data bits sent at both edges of each clock. The address for the starting strip is assumed to be sent also. If the starting strip address is not provided, we need to add the circuilts to provide a look up table to convert the road ID to the starting strip address in the strip-TDS.

The diagram for the TDS configuration and monitoring is shown in Fig. 6. For each chip, we configure BCID counter offset, BCID matching option (only send out charge information for strips belong to the current BCID, or also send out charge information for strips that belong to the current BCID plus one), delay time between two BCID counters, and serializer running modes etc. For each TDS input channel, we configure programmable delay time adjustment, and provide channel enable or disable signal etc. TMR is used for all these configuration bits to avoid SEUs.

Each pad-TDS handles a total of 96 pads with SLVS input and can accept the VMM's "LVDS" like input. Each pad-TDS has 12 bits of BCID offset and a configurable phase of a local 40 MHz BCID clock. The phase shift precision is 6.25 ns. Each pad-TDS divides the 96 channels into 6 groups with 16 pads each. Each group can be configured to four different phases of the 40 MHz clock. This is mainly used to compensate different cable length from the pad detector to the VMM. Pad-TDS checks the presentence of a leading edge of the VMM pulse for each channel in each BC. A channel is marked as "yes" as long as the leading edge of a pulse appears in the current BC, otherwise, a "no" will be recorded. For the example shown below, channels 0 and 1 are fired while the rest channels are not. The "yes/no" flags of all channels will be sent to the pad logic every BC. Pad-TDS uses the same serializer as used by the strip-TDS. For each BC, we can arrange all data including headers and CRC into 120 bits. The format is listed below:

header	Pad-TDS status	BCID	CRC
"1010"	96 bits "yes/no"	12 bits	8 bits

The data except for the header bits are scrambled to keep DC-balanced. The same scrambler polynomial is used as the one used for the strip-TDS. Due to available space, 8 CRC bits are used. The CRC is done with a polynomial 0x97: 1+x+x^2+x^3+x^4+x^8. This algorithm has a HD of 4 for 112 bits of data and is capable to recognize all combinations of 1 bit, 2 bits, 3 bits errors, and almost all the combinations of 4 bits errors.

4. Configuration and monitoring

The TDS chips will be configured by the GBT-SCA chips using the I^{2}C protocol, and a few important values will be stored in the diagnostic registers and can be readout for debugging and monitoring purposes. According to the requirement, we placed 240 bits parameter registers for configuration and 40 bits diagnostic registers inner TDS chip. Figure 8 shows the block diagram of I^{2}C configuration part. Triple modular redundancy is used here.

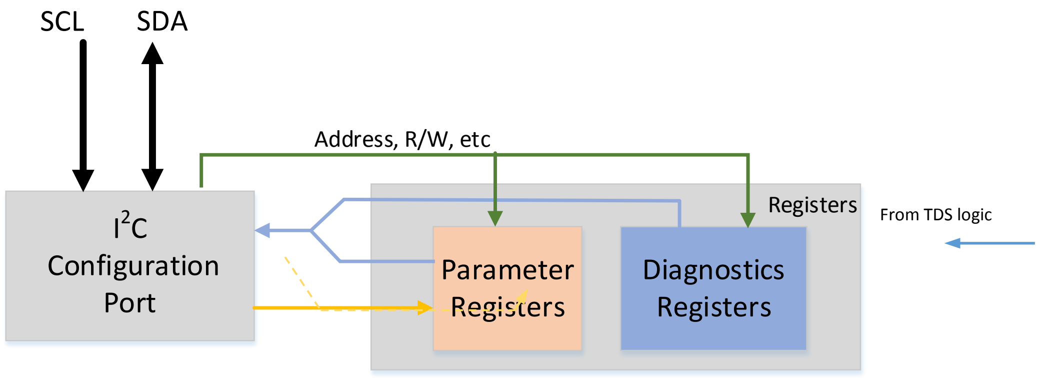

Figure 8: Block diagram of I2C configuration.

I^{2}C is a standard data transmission protocol used in many field. We adopt 10-bits addressing mode and single-byte read/write operation supported by the GBT-SCA chip. Only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors are used in I^{2}C. The TDS chip, as a slave node, will never control SCL, but will put some data on SDA when the chip is selected in read operation. The operating speed depends on the GBT-SCA chip and varies from 100kbps to 1Mbps.

Since only four or five TDS chips will be placed on each front-end board, we arrange 3 bits among the 10-bits address as the chip ID, which allows at maximum eight devices share the same I^{2}C bus. The rest 7 bits are used as inner memory address, which has 35 defined space from 0x0000000 to 0x0100010. The details can be found in Tab. 1. The registers with address from 0x0000000 to 0x0011101 are used as parameter registers that are read/write accessible. The registers with address from 0x0011110 to 0x0100010 are used as diagnostic registers that are read-only via I^{2}C. The diagnostic registers are written by inner TDS logic in general mode. The I^{2}C stream format is defined in Fig. 9. It mainly contains three bytes and some acknowledge bits (ACK). The first two bytes are sent by master (GBT-SCA chip here). The last byte, also called data byte could be sent either by master in write operation or by slave (TDS chip here) in read operation.

Start	1	1	1	1	0	ChipID	ChipID	R/W	ACK	ChipID	A6	A5	A4	A3	A2	A1	A0	ACK	D7	D6	D5	D4	D3	D2	D1	D0	ACK	Stop

Figure 9: I2C stream format.

The 3-bits chip ID will be wired out of chip and connected to logic “1” or “0” on board. The bits para[239:3] in Tab. 1, in total 237 bits could serve as a statics level for controlling the TDS. The three bits para[2:0] are reserved for test. If all of these three reserved bits are “1”, the 40-bits diagnostic registers are writing accessible via I^{2}C to verify their function. The bits Diag[39:0] are diagnostic registers that are read-only in general mode. A few important values will be loaded under the control of TDS logic part and readout via I^{2}C to achieve debugging and monitoring purposes.

5. Packaging

To deal with a large number of I/O pins and 4.8 Gbps output data rate, we will use wire-bond BGA (Ball Grid Array) packaging for all TDS chips. The total number of I/O pins is expected to be around 400. The trace length between the two fast speed TDS output pins and the BGA balls can be controlled to have a distance of at least 0.6 mm.

Address	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
0x0000000	para[7]	para[6]	para[5]	para[4]	para[3]	reserved	reserved	reserved
0x0000001	para[15]	para[14]	para[13]	para[12]	para[11]	para[10]	para[9]	para[8]
0x0000010	para[23]	para[22]	para[21]	para[20]	para[19]	para[18]	para[17]	para[16]
0x0000011	para[31]	para[30]	para[29]	para[28]	para[27]	para[26]	para[25]	para[24]
0x0000100	para[39]	para[38]	para[37]	para[36]	para[35]	para[34]	para[33]	para[32]
0x0000101	para[47]	para[46]	para[45]	para[44]	para[43]	para[42]	para[41]	para[40]
0x0000110	para[55]	para[54]	para[53]	para[52]	para[51]	para[50]	para[49]	para[48]
0x0000111	para[63]	para[62]	para[61]	para[60]	para[59]	para[58]	para[57]	para[56]
0x0001000	para[71]	para[70]	para[69]	para[68]	para[67]	para[66]	para[65]	para[64]
0x0001001	para[79]	para[78]	para[77]	para[76]	para[75]	para[74]	para[73]	para[72]
0x0001010	para[87]	para[86]	para[85]	para[84]	para[83]	para[82]	para[81]	para[80]
0x0001011	para[95]	para[94]	para[93]	para[92]	para[91]	para[90]	para[89]	para[88]
0x0001100	para[103]	para[102]	para[101]	para[100]	para[99]	para[98]	para[97]	para[96]
0x0001101	para[111]	para[110]	para[109]	para[108]	para[107]	para[106]	para[105]	para[104]
0x0001110	para[119]	para[118]	para[117]	para[116]	para[115]	para[114]	para[113]	para[112]
0x0001111	para[127]	para[126]	para[125]	para[124]	para[123]	para[122]	para[121]	para[120]
0x0010000	para[135]	para[134]	para[133]	para[132]	para[131]	para[130]	para[129]	para[128]
0x0010001	para[143]	para[142]	para[141]	para[140]	para[139]	para[138]	para[137]	para[136]
0x0010010	para[151]	para[150]	para[149]	para[148]	para[147]	para[146]	para[145]	para[144]
0x0010011	para[159]	para[158]	para[157]	para[156]	para[155]	para[154]	para[153]	para[152]
0x0010100	para[167]	para[166]	para[165]	para[164]	para[163]	para[162]	para[161]	para[160]
0x0010101	para[175]	para[174]	para[173]	para[172]	para[171]	para[170]	para[169]	para[168]
0x0010110	para[183]	para[182]	para[181]	para[180]	para[179]	para[178]	para[177]	para[176]
0x0010111	para[191]	para[190]	para[189]	para[188]	para[187]	para[186]	para[185]	para[184]
0x0011000	para[199]	para[198]	para[197]	para[196]	para[195]	para[194]	para[193]	para[192]
0x0011001	para[207]	para[206]	para[205]	para[204]	para[203]	para[202]	para[201]	para[200]
0x0011010	para[215]	para[214]	para[213]	para[212]	para[211]	para[210]	para[209]	para[208]
0x0011011	para[223]	para[222]	para[221]	para[220]	para[219]	para[218]	para[217]	para[216]
0x0011100	para[231]	para[230]	para[229]	para[228]	para[227]	para[226]	para[225]	para[224]
0x0011101	para[239]	para[238]	para[237]	para[236]	para[235]	para[234]	para[233]	para[232]
0x0011110	Diag[7]	Diag[6]	Diag[5]	Diag[4]	Diag[3]	Diag[2]	Diag[1]	Diag[0]
0x0011111	Diag[15]	Diag[14]	Diag[13]	Diag[12]	Diag[11]	Diag[10]	Diag[9]	Diag[8]
0x0100000	Diag[23]	Diag[22]	Diag[21]	Diag[20]	Diag[19]	Diag[18]	Diag[17]	Diag[16]
0x0100001	Diag[31]	Diag[30]	Diag[29]	Diag[28]	Diag[27]	Diag[26]	Diag[25]	Diag[24]
0x0100010	Diag[39]	Diag[38]	Diag[37]	Diag[36]	Diag[35]	Diag[34]	Diag[33]	Diag[32]

Table 1: Registers inner TDS chip (240-bits parameter registers and 40-bits diagnostic registers).