SSD documentation

DAQ and Trigger specifications

Version Number : 2

27-june-02

Authors : S.Bouvier, C.Renard, F.Lefevre

1 Introduction to the SSD Readout system

The readout board is a kind of node in the SSD readout system. This board interfaces the slow control, the trigger, the DAQ system and the SSD barrel :

When the Trigger card sends a Trigger command, the Readout board freezes the data in the Front-End.
It reads all the Front-End channels and sends the data via optical fibre to the Daq Receiver board.
When the Slow-Control needs to access the Front-End boards, it configures the Readout board so that no trigger can be sent.

Description of the readout board

The main part of the readout board is in a programmable device (Altera acex 1K100 FPGA). This FPGA decodes the commands issued by the Trigger, plays the readout sequence of the front-end electronics, formats the data before sending them to the DAQ and interfaces with the slow-control and with the SSD test-bench.

The signals coming from the trigger are trigger word (four bits) and RHIC_strobe (10MHz). They are in differential PECL and are converted to TTL on board. The RHIC_strobe is phase filtered by a "roboclock" circuit. This circuit also delivers a 50MHz clock (trg_50_clk), phase locked to the RHIC_strobe. These two clocks are used to read and to decode the trigger commands.

Two handshake signals are sent back to the trigger (status_back and busy_back). Trigger people won't use these two signals described in their trigger documentation.
The onboard 60MHz-quartz oscillator (tx_clk) is phase filtered by a "roboclock" circuit. This circuit also delivers a 30MHz clock (fe_clk), phase locked to tx_clk. Fe_clk is used to sequence the readout of the front-end data (10 bit words). These data are in BLVDS and are converted to TTL onboard. Tx_clk is used to send to the DAQ the header and the formatted data (20 bit words). A Giga-link circuit serialises these 20bit-60MHz-words that are, then, converted to light signals and sent to the DAQ inside an optical fibre.

The readout board receives a 7 Volts – 6 Amperes power supply. It creates 3.3 Volts and 2.5 Volts onboard, each of them protected by a 2.5 Amperes polyfuse. Onboard 5 Volts is protected by a 4 Amperes polyfuse. The board provides the ADC board chain with a 5 Volts power supply, protected by a 2.5 Amperes polyfuse.

S4 switch permits to select if the readout board is connected to N or P side of the front-end. P14 to P17 pads permit to define the readout board serial number. It is a 4 bit binary word.

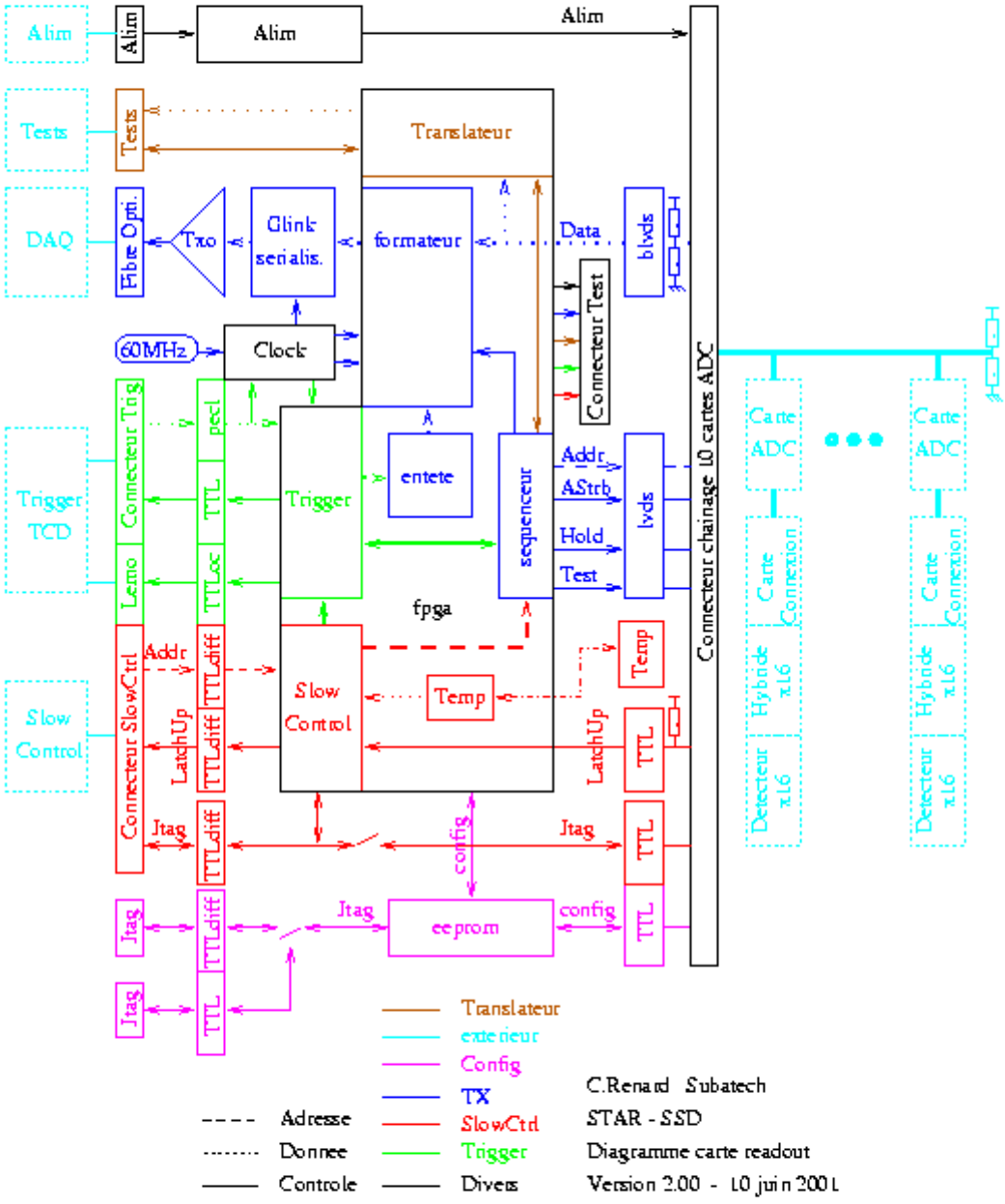

Fig.1: Schematic view of the readout board

2 Trigger

The SSD request for the time of the level 0 trigger : less than 2us

2.1 Definition of SSD trigger words

#	*name*	*STAR use*	*SSD use*
0	Idle	(followed by a null Token) idle	Idle
1	Clear	resets the RHIC_strobe counter	resets the RHIC_strobe counter

2	Master reset		executed as idle (to be defined)
3	Reserved		executed as idle
4	Trigger0	physics readout	physics readout
5	Trigger1	reserved (same as Trigger0)	pedestal readout
6	Trigger2	reserved (same as Trigger0)	executed as Trigger0
7	Trigger3	same as Trigger0 + copy into local buffer	executed as Trigger0
8	Pulser0	electrical calibration pulse	calibration pulse.
9	Pulser1	transparent mode readout	transparent mode
A	Pulser2	reserved	executed as Pulser0
B	Pulser3	read buffer stored by Trigger3	executed as Pulser0
C	Config	reads Front-End configuration	executed as Trigger0
D	Abort	aborts and clears the event identified by the Token	aborts the current event.
E	L1accept	the event identified by the Token is accepted by L1	executed as idle
F	L2accept	the event identified by the Token is accepted by L2	Allows readout to send event_end word

2.2 Description of SSD specific operation

Pedestal readout Trigger (Trigger1):

Readout when no collision occurs.
Pedestal might have to be calculated once a day. Only practising with the complete SSD will permit to determine the pedestal calculation frequency need.
Pedestal should request less than 1000 triggers.

Calibration Trigger (Pulser0):

Before a calibration trigger could be sent, the Slow-Control may have to change some calibration DAC values. If latch-up occurred, the front-end electronics configuration has to be restored. As the trigger and the slow-control cannot access the front-end electronics at the same time, the readout board contains a one bit register, set by the slow-control. This bit is sent to the Trigger system via the Status_back signal. When Status_back is active, the trigger system stops emitting commands to the readout Board. Once the front-end electronics configuration and internal calibration generator values are set, the slow-control resets the Status_back bit in the readout board and the trigger can go on, sending commands.
Calibration might have to be done once an hour. Only practising with the complete SSD will permit to determine the calibration frequency need.
Calibration should request less than 4x100 triggers.

Transparent mode (Pulser1):

Debug readout of a Front-End channel.
Only reads the 2 ADC cards whose address are defined in the Slow-Control's transparent mode register.

Handshake signals:

These 2 bits are open collector (active low). Other sub-detectors do not use them.
Trigger people will not implement them.

Busy_back:
- set when the Readout executes a trigger command.
- Only "reset", "abort" and "L2accept" commands are decoded when this bit is active.
- This bit is also available on a Lemo connector for test purpose.

Status_back:
- modified by Slow-Control. Set when Slow-Control accesses the Front-End.
- No command can be sent when this bit is active.
- This bit is also available on a Lemo connector for test purpose.

Notice : Even if these two signals won't be used for the moment, they will still be available on the boards for further use if necessary. In fact, **the busy_back will be generated internally on TCD board**, and all the pedestal and calibration trigger will be generated during a period dedicated to this purpose, so the SSD will be managed by online control.

3 DAQ

3.1 Use of BNL DAQ system

The STAR ASIC was originally developed to read time multiplexed detectors like Time Projection Chamber or Silicon Drift Detector. Contrary to those detectors, the SSD is not made of time multiplexed elements, but with a total of 491520 channels, the SSD has a rather high number of cells. In order to keep the number of STAR ASICs low for the SSD, the time multiplexing feature of the ASIC will be used to multiplex SSD channels. Thus the proposed design requires only 4 of those cards.

In order to have a balanced load on the ASICs, it is proposed to logically bind one end of the ladder (or one ADC) to one STAR ASIC letting unused 8 chips on the TPC receiver board.

3.2 Data processing

Used in TPC mode, the STAR ASIC is intended to be connected to 64 detector channel, each channels is read from 4 to 1024 times per event. For the SSD, 192 time steps are required.

As pedestal subtraction values depend both on the detector number and the current time bin, this computation can be done with the STAR ASIC for the data coming from the SSD.

Contrary to the pedestal subtraction, the look up translation function that compresses 10 bit ADC data to an 8 bit value depends only on the detector number. The ASIC can handle a maximum of 256 such translation function. It is then proposed to use one translation for all channels.

Few words about DATA mapping
Few words about compression function.

3.3 Structure of the data

Description of the header sent to DAQ

The header words are separated by 1 idle word.

Word name	Description of SSD use	Comparison with SVT
trg cmd	1 digit (comes from Trigger tree)	
daq cmd	1 digit (comes from Trigger tree)	
trg tock_H	1 digit (comes from Trigger tree)	
trg tock_M trg tock_L	2 digits (come from Trigger tree)	
bunch crossing count	1 byte ()	
detector ID	0x00012	*0x00002 for SVT*
readout unit ID	5 bit position on TPC wheel	*5 bit position on TPC wheel for SVT*
spare 1/5	0x00000	*0x00000 for SVT*
spare 2/5	0x00000	*0x00000 for SVT*
spare 3/5	0x00000	*0x00000 for SVT*
spare 4/5	0x00000	*0x00000 for SVT*
spare 5/5	0x00000	*0x00000 for SVT*
tagword	(0x000de)	
tagword	(0x000ad)	
tagword	(0x000fa)	
tagword	(0x000ce)	
det. specific 1/48	1 byte: Readout FPGA version	*sca_offset_count for SVT*
det. specific 2/48	1 byte: Readout FPGA day	*phase_counter for SVT*
det. specific 3/48	1 byte: Readout FPGA month	*0x00000 for SVT*
det. specific 4/48	1 byte: Readout FPGA year x100	*0x00000 for SVT*
det. specific 5/48	1 byte: Readout FPGA year	*0x00000 for SVT*
det. specific 6/48	5 bits: tdo_sel; pm_enable; pc_enable; P/N_side; board_number	*0x00000 for SVT*
det. specific 7/48	LatchUp; status_back; trg_phy; trg_pied; trg_pm; trg_calib; trg_transp; trg_pc	*0x00000 for SVT*
det. specific 8/48	1 byte: latchup number	*0x00000 for SVT*
det. specific 9/48	1 byte: Hold delay used	*0x00000 for SVT*
det. specific 10/48	1 digit: transp addr 2; 1 digit: transp addr 1	*0x00000 for SVT*
det. specific 11/48	2 bits: ladder 10 pedestal 2 bits: ladder 9 pedestal	*0x00000 for SVT*
det. specific 12/48	2 bits: ladder 8 pedestal 2 bits: ladder 7 pedestal 2 bits: ladder 6 pedestal 2 bits: ladder 5 pedestal	*0x00000 for SVT*
det. specific 13/48	2 bits: ladder 4 pedestal 2 bits: ladder 3 pedestal 2 bits: ladder 2 pedestal 2 bits: ladder 1 pedestal	*0x00000 for SVT*
det. specific 14/48	configuration number high	*0x00000 for SVT*
det. specific 15/48	configuration number low	*0x00000 for SVT*
det. specific 16/48	0x00000	*0x00000 for SVT*

det. specific 17/48	0x00000	*0x00000 for SVT*
det. specific 18/48	0x00000	*0x00000 for SVT*
det. specific 19/48	0x00000	*0x00000 for SVT*
det. specific 20/48	0x00000	*0x00000 for SVT*
det. specific 21/48	0x00000	*0x00000 for SVT*
det. specific 22/48	0x00000	*0x00000 for SVT*
det. specific 23/48	0x00000	*0x00000 for SVT*
det. specific 24/48	0x00000	*0x00000 for SVT*
det. specific 25/48	0x00000	*0x00000 for SVT*
det. specific 26/48	0x00000	*0x00000 for SVT*
det. specific 27/48	0x00000	*0x00000 for SVT*
det. specific 28/48	0x00000	*0x00000 for SVT*
det. specific 29/48	0x00000	*0x00000 for SVT*
det. specific 30/48	0x00000	*0x00000 for SVT*
det. specific 31/48	0x00000	*0x00000 for SVT*
det. specific 32/48	0x00000	*0x00000 for SVT*
det. specific 33/48	0x00000	*0x00000 for SVT*
det. specific 34/48	0x00000	*0x00000 for SVT*
det. specific 35/48	0x00000	*0x00000 for SVT*
det. specific 36/48	0x00000	*0x00000 for SVT*
det. specific 37/48	0x00000	*0x00000 for SVT*
det. specific 38/48	0x00000	*0x00000 for SVT*
det. specific 39/48	0x00000	*0x00000 for SVT*
det. specific 40/48	0x00000	*0x00000 for SVT*
det. specific 41/48	0x00000	*0x00000 for SVT*
det. specific 42/48	0x00000	*0x00000 for SVT*
det. specific 43/48	0x00000	*0x00000 for SVT*
det. specific 44/48	0x00000	*0x00000 for SVT*
det. specific 45/48	0x00000	*0x00000 for SVT*
det. specific 46/48	0x00000	*0x00000 for SVT*
det. specific 47/48	0x00000	*0x00000 for SVT*
det. specific 48/48	0x00000	*0x00000 for SVT*

3.4 Description of the data sent to the DAQ board

In the optical fibre, each 20bit word contains data from two ladders, issued to two Asics.
All the data from a ladder side (N or P) is stored in the same Asics.
The DAQ reads data for 3x6 Asics.
As each Asics might need 64x256=16384 words, and a ladder side only corresponds to 16x768=12288 words, we might have to full-fill every Asics with 16384-12288=4096 zeros.
As DAQ people do not have time to test the 64x256 word mode, we must be able to send to every asic with 64x512=32768 words (same as TPC) that is 20480 zeros.
For the zero suppression to work properly, the data from the ladders is complemented when the readout board is connected to the N side of the ladders.
In transparent mode, the data from two ladders is divided each into 5 out of the 10 used Asics.

normal mode	l1	l2	l3	l4	l5	0	l6	l7	l8	l9	l10	0	0	0	0	0	0	0
	a1.1	a1.2	a1.3	a1.4	a1.5	a1.6	a2.1	a2.2	a2.3	a2.4	a2.5	a2.6	a3.1	a3.2	a3.3	a3.4	a3.5	a3.6

Transparent mode	lt1 a1.1	lt2 a1.2	lt1 a1.3	lt2 a1.4	lt1 a1.5	0 a1.6	lt2 a2.1	lt1 a2.2	lt2 a2.3	lt1 a2.4	lt2 a2.5	0 a2.6	0 a3.1	0 a3.2	0 a3.3	0 a3.4	0 a3.5	0 a3.6

Data distribution in the DAQ (we use the first 5 Asics of the 6 Asics in each of the first 2 mezzanines):
l1=ladder 1 ... l10=ladder 10, lt1 and lt2=transparent mode ladders,
a1.1=asic 1 in mezzanine 1 ... a3.6=asic 6 in mezzanine 3.

33.3ns	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*	*33.3ns*
l'1	l'2	l'3	l'4	l'5	l'6	l'7	l'8	l'9	l'10	l"1	l"2	l"3	l"4
Idle idle	idle idle	idle idle	l1 l3 a1.1 a1.3	l5 l6 a1.5 a2.1	l8 l10 a2.3 a2.5	0 0 a3.1 a3.3	0 a3.5 idle	idle idle	Idle idle	idle idle	idle idle	idle idle	l'1 l'3 a1.1 a1.3
			l2 l4 a1.2 a1.4	0 l7 a1.6 a2.2	l9 0 a2.4 a2.6	0 0 a3.2 a3.4	0 a3.6						l2 l4 a1.2 a1.4

Superposition of the readout time of the ladders and the writing time into the Asics:
l1=ladder 1 ... l10=ladder 10, a1.1= Asics 1 in mezzanine 1 ... a3.6= Asics 6 in mezzanine 3.

3.5 Readout time calculation

	clock60	*clock50*	*clock30*	*clock100*	*other*	*individual*	*cumulated*	
ns	*16.667*	*20.000*	*33.333*	*100.000*	*1.000*	*ns*	*ns*	
cmd input				1		100.0	100.0	
cmd decode		1				20.0	120.0	
clock shift					16.67	16.7	136.7	
hold wait	3		45			1550.0	1686.7	
event_start	9					150.0	1836.7	
Header	64					1066.7	2903.3	
data_start	9					150.0	3053.3	
TPC wait					100000	100000.0	103053.3	
Token in			2			66.7	103120.0	no abort
ADC pipeline			30			1000.0	104120.0	no abort
Rdo pipeline			10			333.3	104453.3	no abort
Min data	15360					256000.0	360453.3	no abort
data	230400					3840000.0	4200453.3	
asic fill to 64x256	36864					614400.0	4814853.3	to be verified
asic fill to 64x512	147456					2457600.0	7272453.3	to be verified
event_end	9					150.0	7272603.3	
Total						**7272603.3**		