Gamma Ray Energy Tracking
In-Beam Nuclear Array

GRETINA Digitizer Specification

Document # GRT-3-060815-0

March 25, 2008

Dionisio Doering, John Joseph, Harold Yaver, Sergio Zimmermann

Accepted and Approved by:

Sergio Zimmermann _____Date_____

I-Yang Lee _____Date_____

GRETINA Digitizer Specification
Gamma Ray Energy Tracking In-Beam Nuclear Array (GRETINA)

Revision History

Revision No.	Pages Affected	Effective Date
0	First Release	9/6/2006
1	10, 11, 12, 13, 26, 27, 28, 29, 33, 34, 35	
2	5, 17, 19, 27	3/25/2008

1 INTRODUCTION

1.1 OVERALL DESCRIPTION

The GRETINA Digitizer module is a combination of digitizer and digital signal processor. It accepts 10 inputs directly from the detector module pre-amplifiers and digitizes at a nominal frequency of 100 MHz with 14 bits ADC precision. The ADC counts interface to the FPGA, which digitally processes the data.

1.2 PRECEDENCE

The design of the GRETINA Digitizer electronics board must meet the Digitizer specific requirements defined in the GRETINA Electronics Requirements Document.

2 DIGITIZER MODULES

2.1 GRETINA DIGITIZER TOP LEVEL BLOCK DIAGRAM

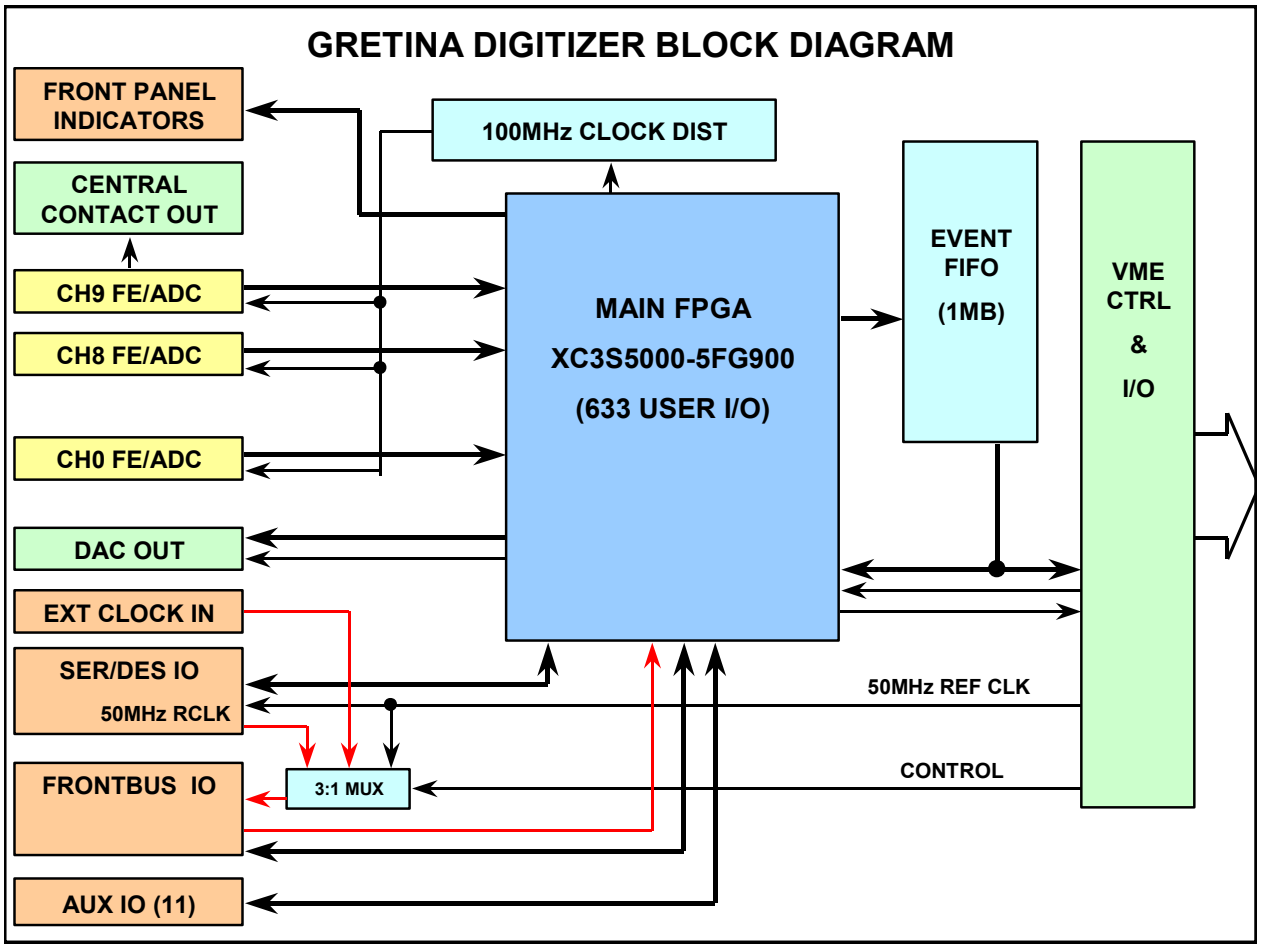


Figure 1: GRETINA Digitizer Block Diagram

2.2 FUNCTIONAL BLOCKS

The LBNL GRETINA Digitizer Electronics board will include all of the functions and blocks described in the following sections and listed below.

VMEBUS INTERFACE
Interface to the Host Computer System

FRONT END PREAMP AND DIGITIZER
Interface to the FE Pre Amplifiers of the Crystal Detectors

EVENT FIFO
256K x 32 FIFO used to buffer Event Data while the VME Host is fetching hit data

MAIN FPGA
Real-Time FPGA Signal Processor

DAC OUTPUT
General Purpose Current or Voltage output

FRONTBUS INTERFACE
Interface used by all Digitizers in one crate to communicate and share vital information

SERIALIZER/DESERIALIZER
Interface to the Trigger and Timing System

AUXILIARY INPUT INTERFACE
16 General Purpose TTL Inputs to the MAIN FPGA

100MHz CLOCK DISTRIBUTION
Onboard Clock Management

FRONT PANEL INDICATORS
User defined Front Panel Monitoring

VOLTAGE MONITORS
Monitoring of voltage level of all key power supply voltages

TEMPERATURE MONITORS
Simple over-temperature switches

2.2.1 VMEBUS INTERFACE

The GRETINA Digitizer board will communicate with a host computer using the VME64x protocol. The slave interface will support the A32/D32 address and data format for block transfers and single read/write access. Geographical addressing (GADDR) will be used to determine the base address of each board in a crate, so any Digitizer can be used in any slot without the need to set a board address on the electronics card. The block diagram of the VME Interface is shown in Figure 2.

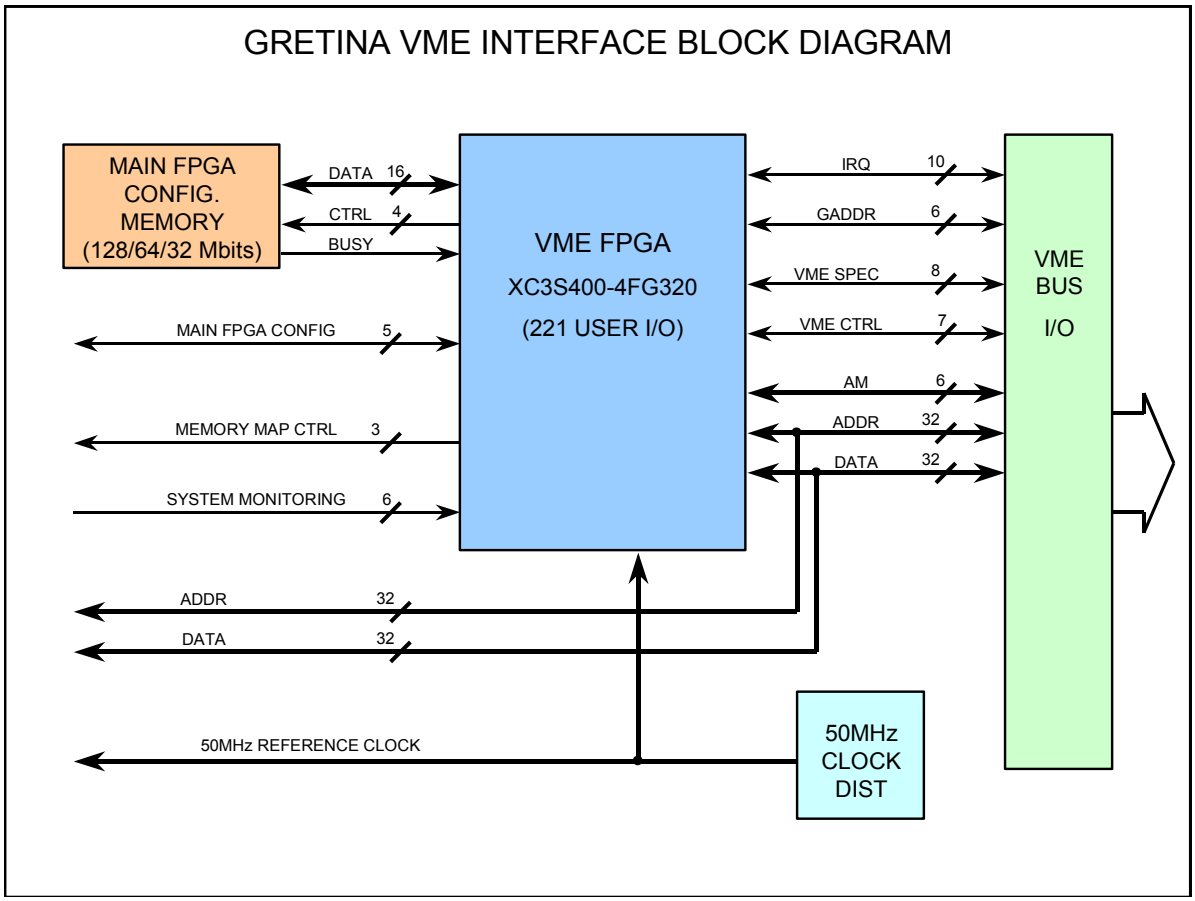


Figure 2: GRETINA VME Interface Block Digram

2.2.1.1 VME INTERFACE DESCRIPTION

The design of the VME interface on the GRETINA Digitizer will be implemented using a Xilinx Spartan3 FPGA and a small number of buffers, drivers and transceivers that are required to physically interface to the VME back plane. The VME FPGA will control the block transfer and programmed IO read/write access to the Digitizer Main FPGA, the configuration Flash memory, and the VME FPGA internal register memory map. The VME FPGA will include functional blocks that will control reset and configuration of the Main FPGA at system power-on and on command when requested by the VME host. The 3 main addressable locations on the Digitizer board are the Main FPGA, the VME Controller FPGA, and the Main FPGA Configuration Memory. Control of the on-board peripherals will be provided through the Main FPGA memory map. These peripherals include the Front End ADC circuits, the output DACs, the FrontBus interface, and the Main Readout FIFO to list a few.

The VME interface requires a stand-alone 50MHz clock that will be buffered through a 1 to 4 low skew differential clock distribution circuit. The 50MHz clock signal will be routed to the Main FPGA and to the Serializer/Deserializer (SER/DES) block to satisfy the operational requirements of each sub-system. Because the SER/DES requires a clean

stable clock, the oscillator component of the distribution shall be a 50.000MHz SMT Temperature Compensated Crystal Oscillator (TCXO) with better than +/-5.0 ppm overall frequency tolerance. A constant clock is required by the Main FPGA to allow readout of internal status registers in the case that the recovered SER/DES clock falls out of lock or is not present.

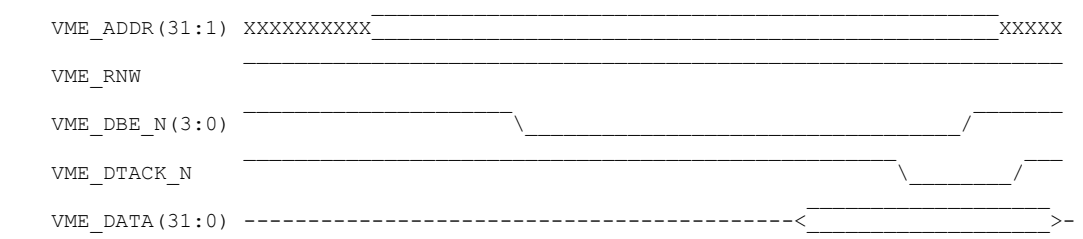
The configuration memory will be implemented with one Intel 128/64/32Mbit, 128k sector flash memory component and will be used primarily to hold configuration programs for the Main FPGA. The Main FPGA Configuration Flash will be accessible on the VME bus for Read/Write operations as required.

2.2.1.2 VME TO MAIN FPGA TIMING

The VMEBus side of the Host to Digitizer Interface is an asynchronous interface that is controlled by any Industry Standard VME Crate Master. The VME FPGA on the Digitizer is configured to resynchronize the data transfers to and from the VME Crate Master. The interface between the VME FPGA and the Main FPGA on the Digitizer board is a synchronous interface, and all control and data signals are generated and sampled on the rising edge of the on-board 50MHz clock.

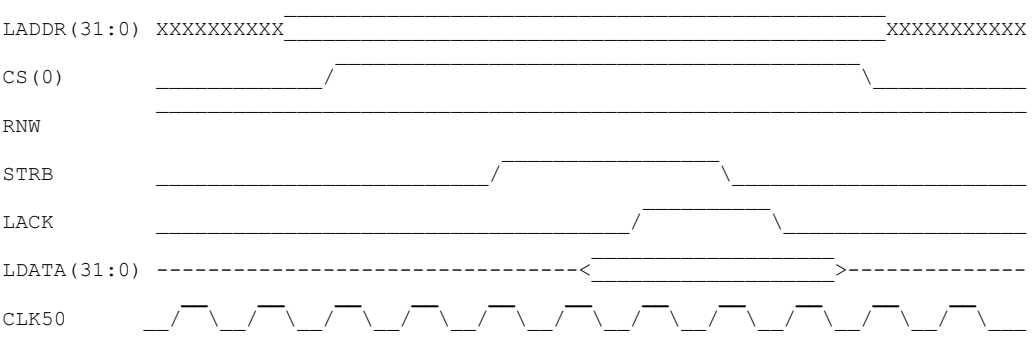
The following timing diagrams show representations of programmed I/O (single data transfer) access only. Block transfer is supported for all registers in the MAIN and VME FPGA. The Block Transfer timing for transfers from the MAIN FPGA to the VME FPGA is not shown because the VME FPGA will handle all the control protocol as long as the MAIN FPGA asserts LACK for every valid DATA STROBE/CYCLE. Block Transfer will be allowed for read access to the MAIN FPGA configuration Flash memory. Block Transfer for write access to the MAIN FPGA configuration Flash memory may not be supported.

Local VME Read Cycle



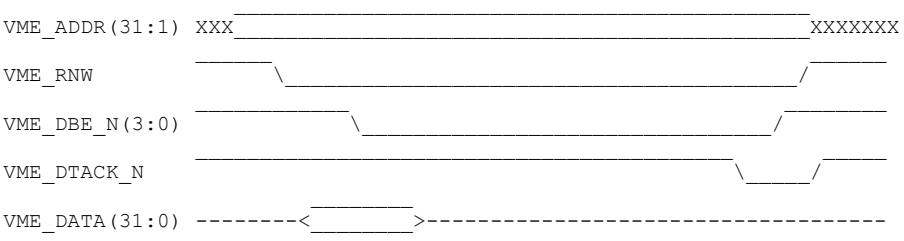
The VME FPGA generates VME_DTACK_N for completed valid cycle.

VME FPGA to MAIN FPGA Read Timing



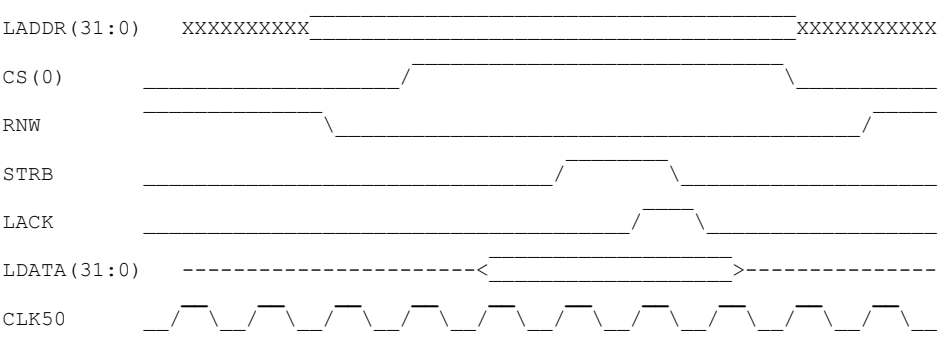
The MAIN FPGA generates LACK when LDATA is valid on the data bus.

Local VME Write Cycle



The VME FPGA generates VME_DTACK_N for completed valid cycle.

VME to MAIN Write Timing



The MAIN FPGA generates LACK when LDATA is latched in the device.

2.2.1.3 GRETINA DIGITIZER MEMORY MAP

The Geographical Address bits on the VME back plane determine the Base Address of the Digitizer boards. The GA bits are compared to bits 24 to 20 of the VME address word.

Table 1 – GRETINA Digitizer Memory Map

GRETINA Digitizer: Top Level Memory Map (CL Proposal)			
(31:25)	(24)	(23:20)	(19:0) Start Address/Description - D32 unless noted
OS Dep	GA4	GA3:GA0	
0x0	0	SLOT	0x00000: Main FPGA Registers
0x0	0	SLOT	0x00900: VME FPGA Registers
0x0	0	SLOT	0x00980: Flash Address
0x0	0	SLOT	0x00984: Flash Data with Address Auto-Increment
0x0	0	SLOT	0x00988: Flash Data
0x0	0	SLOT	0x0098C: Flash Commands (16 bit)
0x0	0	SLOT	0x01000: Read Out FIFO Access
0x0	0	SLOT	0x80FFC: Top of Read Out FIFO

2.2.1.4 VME FPGA MEMORY MAP

Description	Address	Access	Width
DIGITIZER MAIN FPGA Configuration Control Register	**BASE_ADDRESS + 0x0900**	**RW**	**32**
All bits in the Register are Self-Clearing, unless noted otherwise.		**Bit Value**	
		1	**0**
Bit 0: Configure Main Digitizer FPGA If this bit is set, the Main Digitizer FPGA will be erased and programmed with a Configuration Bit Stream that is stored in the on board flash memory		Configure Main FPGA	Idle
Bit 1: Configure Main Digitizer FPGA If this bit is set, the Main Digitizer FPGA will be erased and programmed with a Configuration Bit Stream that is stored in the on board flash memory		Configure Diags FPGA	Idle
Bit 2: TBD			
Bit 3: TBD			
Bit 4: Reset Main Digitizer FPGA If this bit is set, a reset pulse is sent to the Main Digitizer FPGA		Reset	Idle
Bit 5: Reset Flash Memory If this bit is set, a reset pulse is sent the Flash Memory		Reset	Idle
Bit 6: TBD			
Bit 7: TBD			
Bit 8: Configuration Override If this bit is set, the FPGA will not send configuration data to the Main FPGA when a configuration command is issued		Override	Idle
Bits[31:6]: TBD			

Description	Address	Access	Width
DIGITIZER MAIN FPGA Configuration Status Register	**BASE_ADDRESS + 0x0904**	**R**	**32**
Status of the MAIN FPGA Configuration		**Bit Value**	
		1	**0**
Bit 0: MAIN FPGA DONE Bit Status		DONE	CNFG
Bit 1: MAIN FPGA INIT_N Bit Status		ERASE	OK
Bit 2: MAIN FPGA PROG_N Bit Status		CNFG	OK
Bit 3: MAIN FPGA HALT Configuration Status		HALT	OK
Bit 4: MAIN FPGA RESET Status		RESET	OK
Bit 5: MAIN FPGA Configuration Enable Status		Config	OK
Bit 6: MAIN FPGA Configuration Done Status		Done	OK
Bit 7: Flash Memory Status		Busy	Ready
Bit 8: Main FPGA DMM LACK		ACK	IDLE
Bit 9: VME FPGA DCM PS Done		Done	
Bit 10: VME FPGA DCM Locked		Lock	Unlock
Bits[31:11]: TBD			

Description	Address	Access	Width
DIGITIZER Voltage and Temperature Status Register	**BASE_ADDRESS + 0x0908**	**R**	**32**
Status of the Digitizer Voltage and Temperature		**Bit Value**	
		1	**0**
Bit 0: All Power Supplies Status		FAULT	OK
Bit 1: Over Voltage Status		FAULT	OK
Bit 2: Under Voltage Status		FAULT	OK
Bit 3: Temperature Sensor 0 Status		FAULT	OK
Bit 4: Temperature Sensor 1 Status		FAULT	OK
Bit 5: Temperature Sensor 2 Status		FAULT	OK
Bits[31:6]: Not Used			

Description	Address	Access	Width
VME GP CONTROL Register	**BASE_ADDRESS + 0x0910**	**RW**	**32**
General purpose VME Control setting		**Bit Value**	
		1	**0**
Bit [0]: Clock Select 0		EXT/REF	SD CLK
Bit [1]: Clock Select 1		EXT CLK	REF CLK
Bit [3:2]: TBD		EXT CLK	REF CLK
Bit [4]: Set FLASH in PROG/ERASE MODE		PROG/ERASE	READ ONLY
Bits [31:2]: TBD		VALUE	

Description	Address	Access	Width
VME TIMEOUT VALUE Register	**BASE_ADDRESS + 0x0914**	**RW**	**32**
VME timeout counter value, 20ns increments		**Bit Value**	
		1	**0**
Bits [31:0]: VME TimeOut Value		VALUE	

Description	Address	Access	Width
VME FPGA VERSION/REVISION Status Register	**BASE_ADDRESS + 0x0920**	**R**	**32**
		Bit Value	
		1	**0**
Bits [15: 0]: Serial Number (11:0)		VALUE	
Bits [23:16]: BOARD Revision Number		VALUE	
Bits [31:24]: VHDL Version Number		VALUE	

Description	Address	Access	Width
VME FPGA SAND BOX REGISTER BLOCK	**BASE_ADDRESS + 0x0930 to 0x093C**	**RW**	**32**
4 Registers to use for Test and Debug		**Bit Value**	
		1	**0**
Bits [31: 0]: Registers for general purpose Read/Write access		VALUE	

2.2.1.5 FLASH MEMORY FOR MAIN FPGA CONFIGURATION

The Main FPGA configuration data is stored in a 128Mbit Intel Flash memory IC that can be accessed over the VME bus. The data bus on the device is 16 bits wide, so the VME FPGA translates the D32 access to the proper byte locations automatically. The Flash memory has 128 128k Bytes blocks and supports 32 Bytes of block transfer programming in an internal write buffer. Because the Flash commands require single access 16 bit writes, the commands occupy a different address map range than the data locations. The Main FPGA configuration data starts at address 0x0, or block 0 of the flash.

The flash access is performed using a Host-Port-Interface type of architecture that is controlled using 4 registers in the VME Interface FPGA. The Flash Address Register is used to store the memory location that is required for each specific operation. If the Flash is accessed using the Flash Data Register with Address Auto-Increment, then the

address held in the value of the flash address will be incremented by 4 at the end of each cycle. Data access through the standard Flash Data Register leaves the address value static. Command access through the Flash Command Register writes 16 bit commands to the address location in the address register.

At power up, the flash is in "Read Array" mode, but after and Erase or command, a "Read Array" command must be sent to reset the internal state machine. The "Read Array" command is a write of data 0xFFh to the Flash Command Register in the VME FPGA. A Flash memory reset will also put the device back in "Read Array" mode.

To perform a block-erase operation, a Block Erase command sequence must be issued at the desired block address. The Block Erase command sequence requires two cycles, and is shown in the table below:

Command	VME Address	VME Data
Write – Sector Address	0x0980	Base Address of sector to erase
Write - Setup Block Erase	0x098C	0x0020h
Write – Confirm Block Erase	0x098C	0x00D0h
Read - Status Signal	0x0904	Bit(7)

If Bit(7) = 0, then Block Erase is complete

Buffered programming operations simultaneous program multiple words into the flash memory array, significantly reducing effective word-write times. User-data is first written to a write buffer, and then programmed into the flash memory array in buffer-size increments. Optimal performance and power consumption is realized only by aligning the start address on 32-word boundaries. Crossing a 32-word boundary during a buffered programming operation can cause programming time to double.

To perform a buffered programming operation, first issue the Buffered Program setup command at the desired starting address. Status Register Polling SR7 determines write-buffer availability (0 = not available, 1 = available). If the write buffer is not available, re-issue the setup command and check SR7; repeat until SR7 = 1. The Status Signal output can also be checked to indicate if the Flash is busy or ready. Next, issue the word count at the desired starting address. The word count represents the total number of words to be written into the write buffer, minus one. This value can range from 0x01h (2 bytes) to a maximum of 0x1Fh (32 bytes). If the part is configured in 16 bit mode, the maximum value is 0xFh. Following the word count, the write buffer is filled with user-data. User-data is programmed into the flash array at the address issued when filling the write buffer. After all user-data is written into the write buffer, issue the confirm command. If a command other than the confirm command is issued to the device, a command sequence error occurs and the operation aborts. After issuing the confirm command, write-buffer contents are programmed into the flash memory array. The Status Signal indicates a busy status (Bit7 = 1) during array programming.

Issuing another setup command, and repeating the buffered programming bus-cycle sequence can initiate additional buffered programming operations. However, any errors in the Status Register must first be cleared before another buffered programming operation can be initiated. (See Ref [4] for more specification details).

Command	Address	Data
Write – Program start address	0x0980	Start address
Write - Setup Write Buffer	0x098C	0x00E8h
Read - Status Signal	0x0904	Bit(7)
Write – Word Count	0x098C	0x0001h to 0x000Fh
Write – Load Buffer	0x0984	Array Data
Write – Confirm Write	0x098C	0x00D0h
Read - Status Signal	0x0904	Bit (7)

If Bit(7) = 0, then Buffer Write is complete

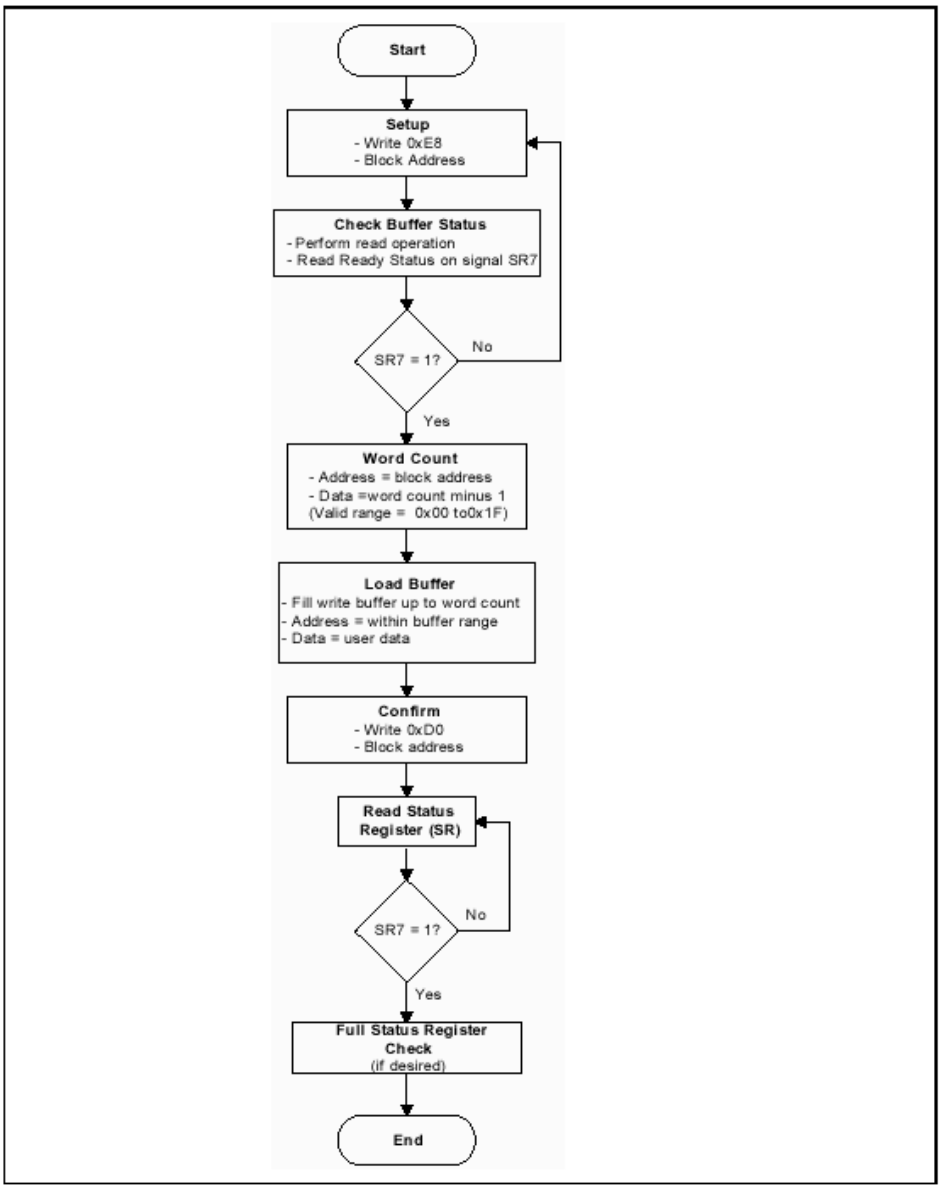


Figure 3: Flash Write to Buffer Flow Chart

2.2.1.6 VMEBUS J1/P1 CONNECTOR PIN ASSIGNMENTS

A 160-pin connector will be used for P1/J1 to allow use of the Geographical addressing pins, the 3.3V power supply, and extra Ground pins available in the standard VME64x crate. The VME Interface on the Digitizer is a simple Slave type interface and will not use any of the Bus Grant or Bus Request capabilities available on the P1 connector. The VMEBus SYSCLK pin will not be connected on the Digitizer board because the on-board system require clocks with stability in the range of +/5 ppm. All signals shaded in red in Table 2 are not connected to the Digitizer VME Interface.

Table 2 – VME P1 Signals

VME I/O: J1/P1 Pin Assignments (VME64x)					
Pin	**Row Z**	**Row A**	**Row B**	**Row C**	**Row D**
1	MPR	D00	BBSYn	D08	VPC
2	GND	D01	BCLRn	D09	GND
3	MCLK	D02	ACFAILn	D10	+V1
4	GND	D03	BG0INn	D11	+V2
5	MSD	D04	BG0OUTn	D12	RsvU
6	GND	D05	BG1INn	D13	-V1
7	MMD	D06	BG1OUTn	D14	-V2
8	GND	D07	BG2INn	D15	RsvU
9	MCTL	GND	BG2OUTn	GND	GAPn
10	GND	SYSCLK	BG3INn	SYSFAILn	GA0n
11	RESPn	GND	BG3OUTn	BERRn	GA1n
12	GND	DS1n	BR0n	SYSRESETn	+3.3V
13	RsvBus	DS0n	BR1n	LWORDn	GA2n
14	GND	WRITEn	BR2	AM5	+3.3V
15	RsvBus	GND	BR3n	A23	GA3n
16	GND	DTACKn	AM0	A22	+3.3V
17	RsvBus	GND	AM1	A21	GA4n
18	GND	Asn	AM2	A20	+3.3V
19	RsvBus	GND	AM3	A19	RsvBus
20	GND	IACKn	GND	A18	+3.3V
21	RsvBus	IACKINn	SERA	A17	RsvBus
22	GND	IACKOUTn	SERB	A16	+3.3V
23	RsvBus	AM4	GND	A15	RsvBus
24	GND	A01	IRQ7n	A14	+3.3V
25	RsvBus	A02	IRQ6n	A13	RsvBus
26	GND	A03	IRQ5n	A12	+3.3V
27	RsvBus	A04	IRQ4n	A11	LI/In
28	GND	A05	IRQ3n	A10	+3.3V
29	RsvBus	A06	IRQ2n	A09	LI/On
30	GND	A07	IRQ1n	A08	+3.3V
31	RsvBus	-12 VDC	+5 VSTDBY	+12 VDC	GND
32	GND	+5 VDC	+5 VDC	+5 VDC	VPC

2.2.1.7 VMEBUS J2/P2 CONNECTOR PIN ASSIGNMENTS

A 96 pin connector will be used for P2/J2 because there are no proposed connections to Row D and Row Z.

Table 3 - VME P2 Signals

VME I/O: P2/J2 Pin Assignments (VME64x)					
Pin	**Row Z**	**Row A**	**Row B**	**Row C**	**Row D**
1	UsrDef	UsrDef	+5 VDC	UsrDef	GND
2	GND	UsrDef	GND	UsrDef	Aux Bus
3	UsrDef	UsrDef	RETRYn	UsrDef	Aux Bus
4	GND	UsrDef	A24	UsrDef	Aux Bus
5	UsrDef	UsrDef	A25	UsrDef	Aux Bus
6	GND	UsrDef	A26	UsrDef	Aux Bus
7	UsrDef	UsrDef	A27	UsrDef	Aux Bus
8	GND	UsrDef	A28	UsrDef	Aux Bus
9	UsrDef	UsrDef	A29	UsrDef	Aux Bus
10	GND	UsrDef	A30	UsrDef	GND
11	UsrDef	UsrDef	A31	UsrDef	Aux Bus
12	GND	UsrDef	GND	UsrDef	Aux Bus
13	UsrDef	UsrDef	+5 VDC	UsrDef	Aux Bus
14	GND	UsrDef	D16	UsrDef	Aux Bus
15	UsrDef	UsrDef	D17	UsrDef	Aux Bus
16	GND	UsrDef	D18	UsrDef	Aux Bus
17	UsrDef	UsrDef	D19	UsrDef	Aux Bus
18	GND	UsrDef	D20	UsrDef	Aux Bus
19	UsrDef	UsrDef	D21	UsrDef	GND
20	GND	UsrDef	D22	UsrDef	Aux Bus
21	UsrDef	UsrDef	D23	UsrDef	Aux Bus
22	GND	UsrDef	GND	UsrDef	Aux Bus
23	UsrDef	UsrDef	D24	UsrDef	Aux Bus
24	GND	UsrDef	D25	UsrDef	Aux Bus
25	UsrDef	UsrDef	D26	UsrDef	GND
26	GND	UsrDef	D27	UsrDef	Aux Bus
27	UsrDef	UsrDef	D28	UsrDef	Aux Bus
28	GND	UsrDef	D29	UsrDef	Aux Bus
29	UsrDef	UsrDef	D30	UsrDef	UsrDef
30	GND	UsrDef	D31	UsrDef	UsrDef
31	UsrDef	UsrDef	GND	UsrDef	GND
32	GND	UsrDef	+5 VDC	UsrDef	VPC

2.2.2 FRONT END PREAMP AND DIGITIZER

Each GRETINA Digitizer electronics board has 10 front-end input channels. The data signals from the detector amplifiers are transmitted to the digitizer front-end inputs on differential wire pairs in a mass terminated cable assembly. The receptacle connector on the Digitizer is a 100-pin double density D type with 75-mil contact spacing, manufactured by ITT/Cannon, part number 2DD100SBRP. The received input signal is buffered with a true differential input amplifier before the signal is digitized using a 14-bit 100Msps Analog to Digital Converter. The digitized data of all channels is read out simultaneously by a controller/processor FPGA at 200Mbytes/sec/channel.

2.2.2.1 ANALOG INPUT

The analog inputs to the digitizer front-end amplifiers are true differential, high impedance type. A block diagram is shown in Figure 4.

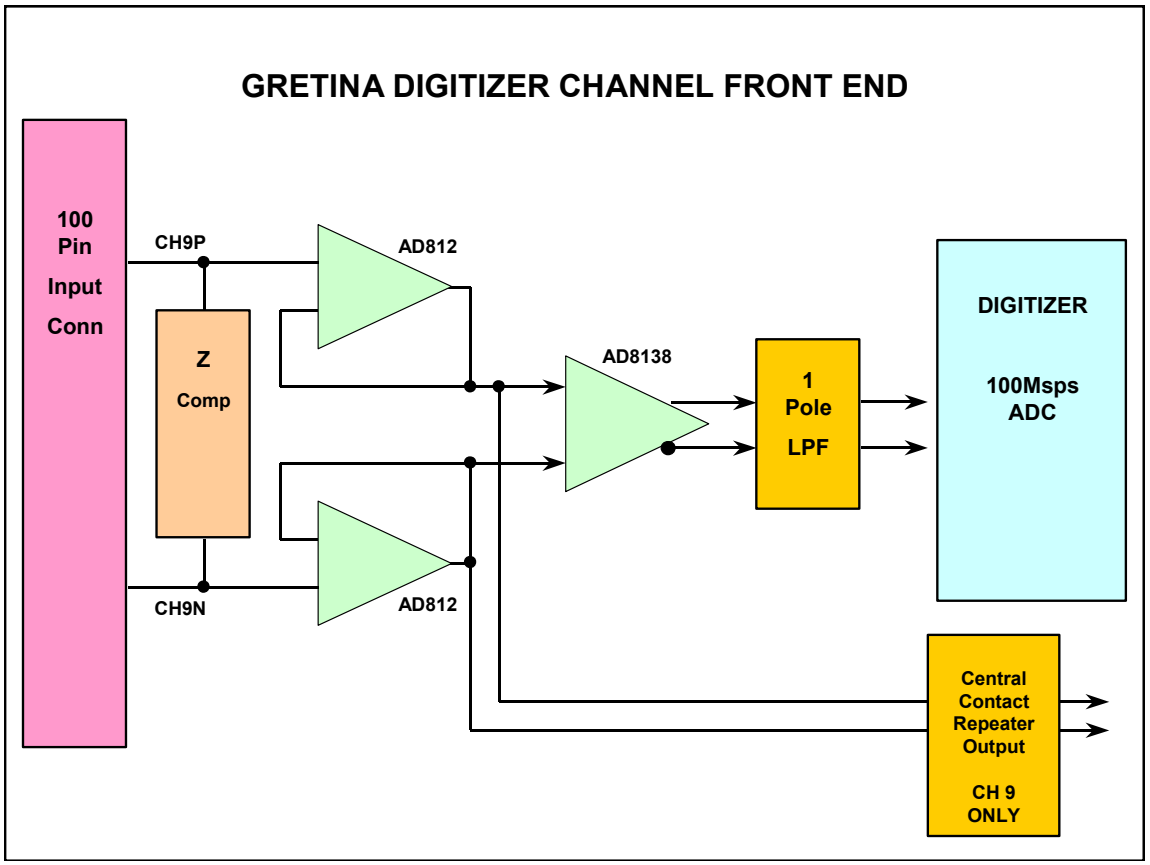


Figure 4: Digitizer Front End

The pin assignments for the analog input signals are defined in Table 4. The labeling for the signals is as follows:

CH0P: Channel 0, positive polarity input signal
CH0N: Channel 0, negative polarity input signal
SHIELD0: Internal cable shield for Channel 0 wire pair

CH1P: Channel 1, positive polarity input signal
CH1N: Channel 1, negative polarity input signal
SHIELD1: Internal cable shield for Channel 1 wire pair

……

CH8P: Channel 8, positive polarity input signal
CH8N: Channel 8, negative polarity input signal
SHIELD8: Internal cable shield for Channel 8 wire pair

CH9P: Channel 9, positive polarity input signal
CH9N: Channel 9, negative polarity input signal
SHIELD9: Internal cable shield for Channel 9 wire pair

SP1: Auxiliary wire pair connections and shield
SP2: Auxiliary wire pair connections and shield

GND: Board Ground

Table 4 - Digitizer Input Pin Assignments

Digitizer Input Interface (100pin ITT Cannon Double Density D Connector, 2DD100SBRP) Mating Connector: 2DD100P							
Pin	**Description**	**Pin**	**Description**	**Pin**	**Description**	**Pin**	**Description**
1	**SHIELD9**	**27**	**SHIELD9**				
2	**CH9N**	**28**	**CH9P**	**52**	SP2	**76**	SP2
3	GND	**29**	GND	**53**	GND	**77**	SP2
4	GND	**30**	GND	**54**	**SHIELD8**	**78**	**SHIELD8**
5	GND	**31**	GND	**55**	**CH8P**	**79**	**CH8N**
6	**SHIELD7**	**32**	**SHIELD7**	**56**	GND	**80**	GND
7	**CH7N**	**33**	**CH7P**	**57**	GND	**81**	GND
8	GND	**34**	GND	**58**	GND	**82**	GND
9	GND	**35**	GND	**59**	**SHIELD6**	**83**	**SHIELD6**
10	GND	**36**	GND	**60**	**CH6P**	**84**	**CH6N**
11	**SHIELD5**	**37**	**SHIELD5**	**61**	GND	**85**	GND
12	**CH5N**	**38**	**CH5P**	**62**	GND	**86**	GND
13	GND	**39**	GND	**63**	GND	**87**	GND
14	GND	**40**	GND	**64**	**SHIELD4**	**88**	**SHIELD4**
15	GND	**41**	GND	**65**	**CH4P**	**89**	**CH4N**
16	**SHIELD3**	**42**	**SHIELD3**	**66**	GND	**90**	GND
17	**CH3N**	**43**	**CH3P**	**67**	GND	**91**	GND
18	GND	**44**	GND	**68**	GND	**92**	GND
19	GND	**45**	GND	**69**	**SHIELD2**	**93**	**SHIELD2**
20	GND	**46**	GND	**70**	**CH2P**	**94**	**CH2N**
21	**SHIELD1**	**47**	**SHIELD1**	**71**	GND	**95**	GND
22	**CH1N**	**48**	**CH1P**	**72**	GND	**96**	GND
23	GND	**49**	GND	**73**	GND	**97**	GND
24	GND	**50**	GND	**74**	**SHIELD0**	**98**	**SHIELD0**
25	SP1	**51**	SP1	**75**	**CH0P**	**99**	**CH0N**
26	SP1					**100**	GND

2.2.2.2 FRONT END PREAMP

High impedance input buffers are used to allow the cable termination resistors to dominate the impedance matching of the transmission line and to improve isolation while allowing signal balance on the wire pair. The full-scale amplitude at the input of the front-end amplifier is 10MeV for the central contact signal and 30MeV for all of the segment signals.

2.2.2.3 CENTRAL CONTACT REPEATER OUTPUT

The central contact of the detector shall be connected to the Channel 9 wire pair of the Master Digitizer that is connected to a differential repeater amplifier as shown in Figure 4. The repeater amplifier is used to buffer the Central Contact signal for connection to a front panel LEMO connector. This allows the Central Contact signal to be viewed externally during run-time.

2.2.2.4 DIGITIZER CIRCUIT

The Digitizer circuit will utilize a 100Msps Analog Devices AD6645ASQ-105 ADC for each input channel. There is a one-pole anti-aliasing filter at the input of the ADC that can be loaded if required. The Main FPGA reads out each digitizer channel ADC simultaneously and writes the data to an internal circular buffer in real time.

2.2.3 EVENT FIFO

The main FPGA writes full events into the event FIFO after receiving a valid command from the trigger system. The event FIFO is implemented using two 256Kx18, 7ns FIFO memory chips. Write control is managed solely by the main FPGA while read control is managed by the main FPGA through the VME interface. The Event FIFO shall be write and read accessible by the VME host to allow functional testing of the FIFO components.

2.2.4 MAIN FPGA

Figure 1 shows the block diagram of the Digitizer. The main FPGA is responsible for control the data acquisition. It uses a Spartan 3 FPGA XC3S5000 (5M gates, 104 Block RAMs) from Xilinx. The configuration memory has a minimum of 16 Mbits and is accessible from VME for in system reconfiguration. The Main FPGA is configured in Slave-Serial Mode as described in the Xilinx Spartan 3 data sheets.

Four Digitizers instrument one crystal. One of these modules is configured as master and is connected to the crystal's central contact. The other three modules are configured as slaves. The VHDL code for all the 4 modules will be identical which means that we are going to have one part of the code that is the master and another part that is the slave. The master code will be enabled only for the modules that are working as a master while the slave code will be enable in all the modules.

Figure 5 shows a block diagram of the main FPGA. All the FPGA logic is enabled when the module is configured as a master. However, when the module is configured as a slave, the master logic is disabled. The FrontBus logic receives commands and information from the FrontBus and either executes the command or sends back the information requested. The VME control block does the communication between the main FPGA and the VME FPGA. This means that the main FPGA access the VME bus only through the VME FPGA. The DAC control block is responsible for retrieving information from the channels, processing it, and controlling the DAC chips. VME registers will configure the processing. The channels block communicates with the FIFO interface. The FIFO interface reads data out of the 10 channels, and during readout transfers data to VME.

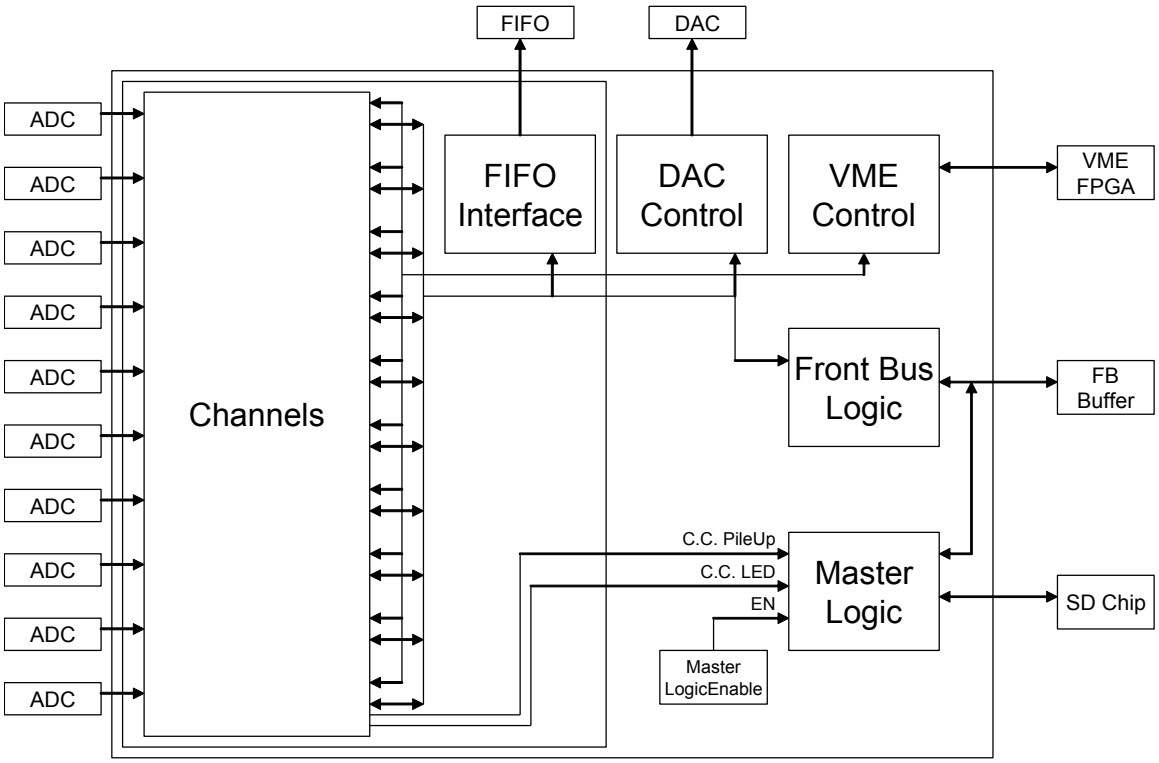


Figure 5: Main FPGA Block Diagram

The channels block implements all the following DSP algorithms:

- Leading Edge Discrimination:
  - $y_n = x_n - x_{n-k}$ (differentiation)
  - $y_n = (x_n + x_{n-2}) + x_{n-1}<<1$ (×4, Gaussian filtering)
  - Threshold comparison → Lead Edge Discriminator (LED) time
- Constant Fraction Discrimination: (See also David Radford's new CFD)
  - $y_n = x_n - x_{n-k}$ (differentiation)
  - $y_n = (x_n + x_{n-2}) + x_{n-1}<<1$ (×2, Gaussian filtering)
  - $y_n = x_{n-k} - fx_n$ (constant fraction, f is an attenuation factor)
  - Zero crossing comparison → CFD time
- Trapezoidal filter and energy determination (V.T Jordanov, G.F. Knoll, NIM A345 (1994) 337-345)
  - $y_n = y_{n-1} + ((x_n + x_{n-2m-k})) - (x_{n-m} + x_{n-m-k}))$
  - Maximum tracking → energy
- Pole-Zero correction
  - $y_n = x_n + I_n / t$ (where t is the pre-amplifier time constant)
  - $I_n = I_{n-1} + x_n$

The master logic block will be enabled by the master logic enable signal. This signal will be sent by the VME FPGA and will indicate to the main FPGA if it is a master or slave module, which will be determined by a VME register.

The master logic uses the front panel bus to control the four modules connected to a crystal. Also, it interfaces with the Serializer/Deserializer (SER/DES), which communicates with the Trigger Timing & Control Link (TTCL). Figure 6 shows the block diagram of the FrontBus master logic.

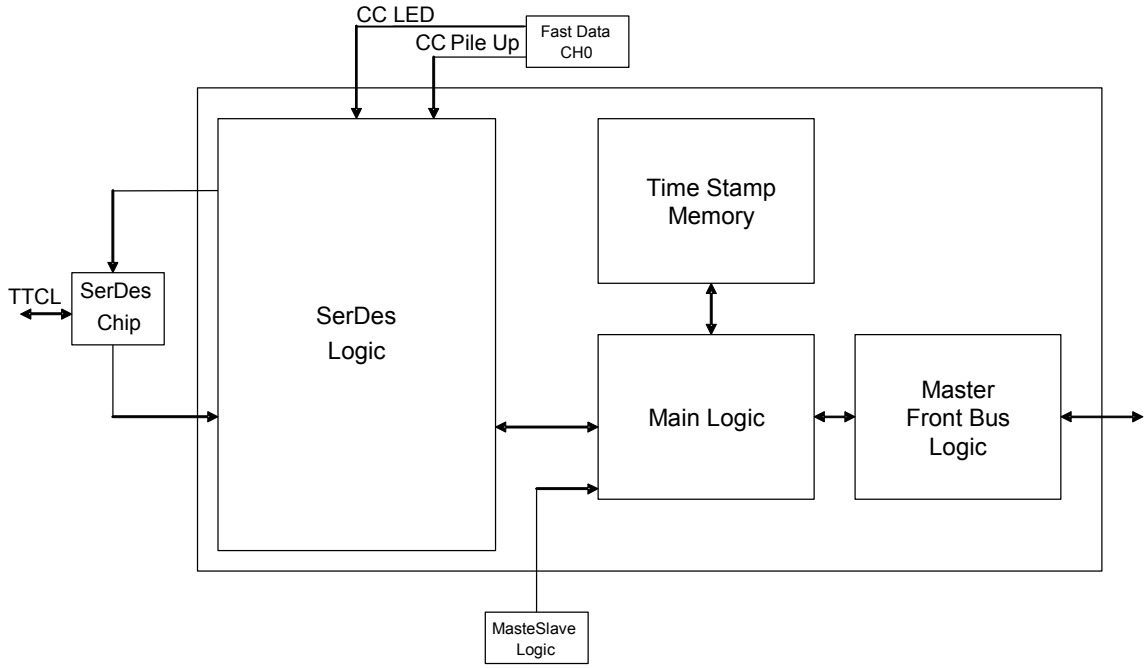


Figure 6: Block Diagram of the FrontBus Master Logic

The Main Logic block controls all the other functional blocks in this algorithm and interprets the TTCL commands. The SER/DES logic is responsible for reading and validating the data package received from the TTCL. This data package is described in reference [3]. Once it validates the data package, it passes it on to the Main Logic block so that the required commands can be executed. The Main Logic block is also responsible for sending the slow and fast hit description data back to the TTCL. This other data package structure is described in reference [1]. The fast hit description data can be the Central Contact Leading Edge Discriminator (CC LED) and/or the Central Contact Pile Up signals that are connected directly to this Main Logic block. The Time Stamp memory (TS Memory) is a set of registers that store all the time stamps that were sent to the TTCL and still point to valid raw data inside the circular buffer.

2.2.4.1 MODES OF OPERATION

UNDER DISCUSSION … THE CURRENT PROPOSED ETC. The main FPGA will operate in four different modes (see Table 5). We will now describe these modes.

Table 5 – Modes of Operation

Mode	Board type
00	Internal
01	External
10	Internal with validation
11	TTCL

2.2.4.1.1 INTERNAL MODE

This mode can be used with one single board or one full crate without the TTCL communication. In the internal mode all channels are independent. When the logic detects a leading edge (through the LED algorithm) this will start timers, which will latch the results of the DSP algorithms (energy, CFD, etc) into the header memory. Later, the data from the circular buffer and header is transferred to the pre-buffer memory (for more information on the pre-buffer memory, refer to Section 2.2.4.5). In this mode, all the events will be readout and we don't expect any validation.

2.2.4.1.2 EXTERNAL MODE

This mode can also be used with one single board or one full crate without the TTCL communication. One external trigger signal on the auxiliary inputs is used to start timers, which will latch the results of the DSP algorithms (energy, CFD, etc) into the header memory. This external signal is propagated to all channels so all of them are going to store information even if they didn't have a valid LED. Later, the data from the circular buffer and the header memory is transferred to the pre buffer memory (for more information on the pre-buffer memory, refer to Section 2.2.4.5). In this mode, all the triggered events will be readout and we don't expect any additional validation. The timing specification for this mode is yet to be determined.

2.2.4.1.3 INTERNAL WITH VALIDATION MODE

This mode is very similar to the internal mode. However, the data from the circular buffer and header is not automatically transferred to the pre-buffer memory. It waits for an external validation signal, provided through an auxiliary input. The timing specification for this mode is yet to be determined.

2.2.4.1.4 TRIGGER TIMING & CONTROL LINK MODE

While operating in TTCL Mode the digitizers associated with one crystal will be connected to the TTCL through the SER/DES of the master digitizer.

An event will start when the LED associated the crystal central contact (CC) fires. The CC will always be connected to channel 1 of the master digitizer module. The master digitizer informs all crystal channels (through the FB_LED line) and the TTCL (through the fast data link) that the LED was detected. When the LED signal is detected, the results of the DSP algorithms of all of the enabled channels (energy, CFD, etc) are

latched into the header memory. The master digitizer assigns to this event a specific header memory position. Therefore, all channels will store the DSP algorithm information in the same header memory position. In parallel, the master module will collect information to assemble the trigger buffer. It will calculate the low-resolution energy of the CC and collect the channel hit pattern. When the TTCL demands slow data from the digitizer boards, the master module will then send the information that is in the trigger buffer to the TTCL. The master digitizer will save the time stamp and the header memory position. The modules will then be ready to handle another event or to receive a readout command.

Once the TTCL makes a decision that an event needs to be readout it will send to the digitizer boards a readout command together with the time stamp of the event. The master digitizer module will then compare this time stamp with all the time stamps that it has stored and that are still valid (i.e., for which the raw data are still available inside the circular buffer). If the master digitizer finds a match it will send a readout command to all channels informing them which header memory position needs to be readout. The channels will then transfer the information from the header memory and the raw data to the pre-buffer memory. Later the data is transfer to the external FIFO. If the event is not readout before the circular buffer is overwritten, that particular event is flushed from the header memory.

2.2.4.2 CHECKING SYNCHRONISM

Every channel has its own timer that generates the time stamp. It is therefore necessary to verify that all channels are in sync. Since the sync command will be sent every 2us and we don't want to overload the FrontBus communication, we will divide this task into two parts.

Every time the master digitizer module receives the sync command it will check the TTCL time stamp against the Central Contact time stamp. If they match then no error flag will be set. This will be done every 2us.

At a slower rate, the master digitizer module will check the central contact time stamp against each channel. This will be done by sending a command to latch the current status of all channels as a broadcast command through the FrontBus. Then the master digitizer will broadcast the CC TS and each channel will compare the CC TS with its latched TS. If there is a mismatch the channel will raise an error flag.

The master digitizer will then send a command to get the status of each board. If there is any error it will send an error message to the TTCL. No action besides sending this information to the TTCL will be done. If a TS reset is required, the TTCL will send a specific command asking for that. Otherwise this flag will remain set.

2.2.4.3 INITIALIZATION

2.2.4.3.1 SYSTEM INITIALIZATION

All Digitizers in a crate will behave as slave boards after start up or reset. The VME host sets a register in the Main FPGA that will determine if a board is a Master or Slave Digitizer. After the VME host determines which board is a Master, the Master logic will take control of the crate.

2.2.4.3.2 TRIGGER TIMING & CONTROL LINK INITIALIZATION

During initialization and reset the SER/DES link will be down for some period of time while it locks to the Master Trigger TX signal. After acquiring lock, the system will start receiving clock signals with a 20 ns period. The TTCL will keep the SYNC signal active until its lock signal goes high, meaning that the Master Digitizer Board will start receiving nine words with MSB equal to '0' and LSB equal to '1'.

Once the main FPGA has received a valid clock from the Master Trigger SER/DES and the system reset goes low, the Master Digitizer Board will start sending data to the TTCL. Initially it will send sync data packets that are used to cause the SER/DES chip to acquire lock. After the TTCL SER/DES locks to the Digitizer SER/DES, then it will start sending real data to the Master Digitizer Board.

The SER/DES block of the Master Digitizer Board will transition from Sync Mode to Operating Mode once it receives the first data packet from the Master Trigger board that is different from the sync data. Once the Master Digitizer is out of Sync Mode, different commands will be sent to the Main Logic block of the Master Digitizer (Figure 6) every 5 clocks.

If the lock signal goes down or a reset occurs then the SER/DES Block of the Master Digitizer Board will go back to the sync mode and use the same procedure in order to start receiving commands again.

2.2.4.4 CLOCK MANAGEMENT

The main FPGA has three clock inputs. One of them comes from the VME FPGA and is always present. The second one is the system clock source that is supplied by the TTCL link, the FrontBus interface, the 50MHz VME reference clock or the external clock Input on the front panel Auxiliary I/O bus. The VME FPGA controls the system clock multiplexer with 2 bits in an internal register that is on the VME Bus.

When a board is operating in stand-alone mode, it may receive the system clock from the internal 50MHz clock distribution or from the Auxiliary I/O external clock input. When 4 boards are operating as an independent crate, the slave boards will receive their clock through the FrontBus and the Main Digitizer Board will receive its clock internal 50MHz clock distribution or from the Auxiliary I/O external clock input.

When working on the TTCL mode the Main Digitizer Board will receive its clock from the recovered clock of the TTCL link and will spread it through the FrontBus to all the slave boards.

2.2.4.5 FIFO AND OUTPUT DATA FORMAT

Each ADC channel has a pre-buffer memory. The data are transferred from the pre-buffers to the FIFO for VME readout. The structure of the data package can be seen on Figure 7. If the module is a master, additional information, the master header specific to the master functions, is stored in the FIFO for readout.

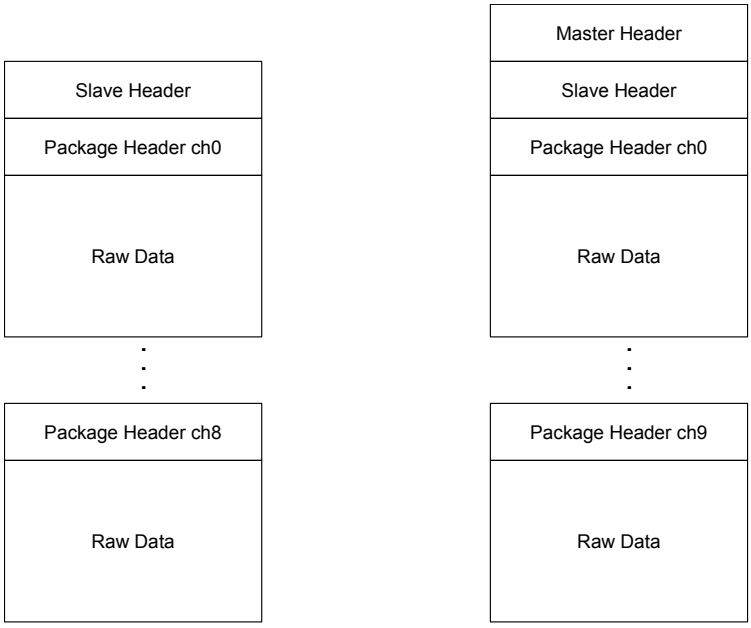


Figure 7: (a) Data package from a slaver board. (b) Data package from a master board.

The Package header contains the information that belongs to the channels. Its structure is shown in Figure 8.

0 1 2 3	4 5 6 7 8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
CH_ID	UserDef	Packet length (header included) GA
LED / external timestamp bits 0 - 15*		LED / external timestamp bits 16 – 31*
LED / external timestamp bits 32 - 48*		Energy bits 0 – 15
Energy bit 16 – 24	x x T S E C P	CFD Timestamp bits 0 – 15
CFD Timestamp bits 16 - 31		CFD Timestamp bits 32 – 47
CFD point 1 bits 0 - 15		CFD point 1 bits 16 – 31
CFD point 2 bits 0 - 15		CFD point 2 bits 16 – 31
Raw data point 0 (Sign extended)		Raw data point 1 (Sign extended)
Raw data point 2 (Sign extended)		Raw data point 3 (Sign extended)
. . .		. . .

Figure 8: Data Package.

*The TS is updated at 20ns.
Notes:
GA is geographical address (5 bits)
S is for the sign of the LED crossing. (1 is negative).
E is for external trigger flag (the timestamp is external and LED and CFD are not valid).
C is for CFD valid indicating that a CFD crossing occurred.
P is for pileup flag indicating that the energy is corrupted.
T is for data that was generated by a timeout signal during the TTCL mode.
CH_ID identify which channel the data package belongs to.
User def are 12 bits for user defined information set by a register accessed by VME.

The slave header contains information from the event that is common to all the channels on that specific board (Figure 9). All Digitizers in a crate produce the Slave Header. The Master Digitizer will add a top-level header to the data packet that is used to identify the start of a new event from a crystal (Figure 10).

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Board ID	Header length
FB_LED bits 0 – 15	FB_LED bits16 – 31
FB_LED bits 32 – 48	HM number
TBD	TBD

Figure 9: Slave Header.

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Board ID	Header length
Trigger_TS bits 0 – 15	Trigger _TS bits 16 – 31
Trigger _TS bits 32 – 48	HM number
Low Resolution Energy	TBD

Figure 10: Master Header.

2.2.4.6 MAIN FPGA REGISTERS

The value xFFFFFFFF will be returned when reading registers that are not defined,
When reading registers that were not defined they will return the value xFFFFFFFF.

Description	Address	Access	Width
Board ID	**0x00**	**R**	**32**
Value is the board serial number that is given by the switch configuration.		**Bit Value**	
		1	**0**
Bits[11: 0]: Board Id (Pins 8 down to 4 show Geographical Addressing, all other pins are set to zero)		Value	
Bits[15:12]: TBD			
Bits[31:16]: FIRMWARE VERSION			

Description	Address	Access	Width
Programming Done	**0x04**	**R**	**32**
This gives the programming status of the VHDL module as follows.		**Bit Value**	
		1	**0**
Bits[9: 0]: Programming status of the channels			
Bit 10: Programming status of the DAC module			
Bit 11: Programming status of the front bus (slave) module			
Bit 12: Programming status of the master logic module			
Bit 13: Programming status of the Debug module			
Bit 14: Programming status of the FIFO module			
Bit 15: Programming status of the Self Trigger module			
Bits[19:16]: TBD			
Bit 20: FIFO 0 EF flag			
Bit 21: FIFO 1 EF flag			
Bit 22: FIFO 0 PAE flag			
Bit 23: FIFO 0 HF flag			
Bit 24: FIFO 0 PAF flag			
Bit 25: FIFO 0 FF flag			
Bit 26: FIFO 1 FF flag			
Bits[31:27]: TBD			

Description	Address	Access	Width
External Window	**0x08**	**RW**	**32**
External validation window length in clock cycles. Value at reset is 0x0190 or 4us.		**Bit Value**	
		1	**0**
Bits[10: 0]: External window length		Value	
Bits[31:11]: TBD			

Description	Address	Access	Width
Pileup Window	**0x0C**	**RW**	**32**
Pileup window length. Value at reset 0x0400 (10us).		**Bit Value**	
		1	**0**
Bits[10: 0]: Pileup window length		Value	
Bits[31:11]: TBD			

Description	Address	Access	Width
Noise Window	**0x10**	**RW**	**32**
Noise window length. Value at reset 0x0040 (640ns).		**Bit Value**	
		1	**0**
Bits[6: 0]: Noise window length		Value	
Bits[31:11]: TBD			

Description	Address	Access	Width
External trigger sliding length	**0x14**	**RW**	**32**
Length before we read the energy when we operate in external trigger. Value at reset 0x0190 (4.0us)		**Bit Value**	
		1	**0**
Bits[10: 0]: External trigger sliding length		Value	
Bits[31:11]: TBD			

Description	Address	Access	Width
Collection time	**0x18**	**RW**	**32**
Collection time maximum length (length of the flat top in the trapezoid, typically set to 0x0020 or 320ns). Value at reset 0x01C2 (4.5us)		**Bit Value**	
		1	**0**
Bits[8: 0]: Collection time		Value	
Bits[31:9]: TBD			

Description	Address	Access	Width
Integration time	**0x1C**	**RW**	**32**
Integration time length (length of one side of the trapezoid). Value at reset 0x01C2 (4.5us)		**Bit Value**	
		1	**0**
Bits[8: 0]: Integration time		Value	
Bits[31:9]: TBD			

Description	Address	Access	Width
Hardaware status	**0x20**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0]: DCM Front Bus Lock signal		**Locked**	
Bits[1]: DCM VME CLK Lock signal		**Locked**	
Bits[2]: SD Lock signal		**Locked**	
Bits[15: 3]: TBD		Value	
Bits[31:16]: TBD			

Description	Address	Access	Width
Data package user defined data	**0x24**	**RW**	**32**
The data written here will show up at the data package header		**Bit Value**	
		1	**0**
Bits[11: 0]: User defined data		Value	
Bits[31:12]: TBD			

Description	Address	Access	Width
Control/Status channel #Number	*	RW	32
This register address is the only one that is read/write.		Bit Value	
		1	0
Bit 0: START/STOP. This bit must be set to one for the channel to operate. Value at reset is 0. *Read/Write.*			
Bit 1: Debug Mode. This bit when set to 1 puts the channel in debug mode operation using internal data. Value at reset 0. (*Read/Write)*			
Bit 2: Pile-up drop-out enable. When this bit is set to one, the channel drops any event that occurs in a pileup window else it just notifies pileup through a flag in the event header. Value at reset 1. *Read/Write.*			
Bits [4:3]: Trigger mode. We have the following configuration: 00: Internal mode 01: External mode 10: Internal with Validation mode 11: TTCS Decision mode Value a reset: 00. *Read/Write.*			
Bit 5: CFD Tap delay setup. If set to 0, the tap delay is filling-up and no event can be acquired. *Read-only.*			
Bit 6: Tap delay 1 setup. If set to 0, the tap delay is filling-up and no event can be acquired. *Read-only.*			
Bit 7: Tap delay 2 setup. If set to 0, the tap delay is filling-up and no event can be acquired. *Read-only.*			
Bit 8: Tap delay 3 setup. If set to 0, the tap delay is filling-up and no event can be acquired. *Read-only.*			
Bit 9: Tap delay 4 setup. If set to 0, the tap delay is filling-up and no event can be acquired. *Read-only.*			
Bits [11:10]: Polarity validation. Value at reset 11. 01: only positive trigger considered 10: only negative trigger considered 11: both triggers considered 00: no trigger considered (the LED still fires externally though, as opposed to START/STOP) *Read/Write.*			
Bit 12: Select CFD. (not implemented)			
Bit 13: Select Pole-zero (not implemented)			
Bit 14: TBD			
Bit 15: Pre-buffer Ready flag			
Bits[31:16]: TBD			

- There is one of this registers for each channels and their address are specified by the next table

Description	Address	Access	Width
Control/Status channel 0	**0x40**	**RW**	**32**
Control/Status channel 1	**0x44**	**RW**	**32**
Control/Status channel 2	**0x48**	**RW**	**32**
Control/Status channel 3	**0x4C**	**RW**	**32**
Control/Status channel 4	**0x50**	**RW**	**32**
Control/Status channel 5	**0x54**	**RW**	**32**
Control/Status channel 6	**0x58**	**RW**	**32**
Control/Status channel 7	**0x5C**	**RW**	**32**
Control/Status channel 8	**0x60**	**RW**	**32**
Control/Status channel 9	**0x68**	**RW**	**32**

Description	Address	Access	Width
LED Threshold	*	RW	32
This is the Leading Edge Discriminator (LED) threshold. Only bits 17 to 0 are used. This is an unsigned value and the three lower bits are extra precision bit. (The dot is between bit 3 and 2). The value at reset is 0x1FFFF (full range so that no event should trigger). It is internally converted to a signed value depending on how the sign of the current sample is. *Write-only.*		Bit Value	
		1	0
Bits[16: 0]: LED Threshold		Value	
Bits[31:18]: TBD			

* There is one of this registers for each channels and their address are specified by the next table.

Description	Address	Access	Width
LED Threshold 0	**0x80**	**RW**	**32**
LED Threshold 1	**0x84**	**RW**	**32**
LED Threshold 2	**0x88**	**RW**	**32**
LED Threshold 3	**0x8C**	**RW**	**32**
LED Threshold 4	**0x90**	**RW**	**32**
LED Threshold 5	**0x94**	**RW**	**32**
LED Threshold 6	**0x98**	**RW**	**32**
LED Threshold 7	**0x9C**	**RW**	**32**
LED Threshold 8	**0XA0**	**RW**	**32**
LED Threshold 9	**0xA4**	**RW**	**32**

Description	Address	Access	Width
CFD Parameters	*****	**RW**	**32**
Bit 12-7: CFD delay. Value at reset is 0x3F giving a delay of 630ns. For the algorithm to perform correctly this delay must be smaller then the collection time. Bit 6-5: CFD fraction. It is a value indicating what fraction is used (see table 9 for values). Value at reset is "00" giving a fraction of 0.5. Bit 4-0: CFD threshold. It is an unsigned value of the CFD threshold. The dot is after bit 0. Value at reset is 0x10000 (160kev).		**Bit Value**	
		1	**0**
Bits[4: 0]: CFD threshold		Value	
Bits[6: 5]: CFD fraction		Value	
Bits[12: 7]: CFD delay		Value	
Bits[31:13]: TBD			

- There is one of this registers for each channels and their address are specified by the next table.
- We may need some more parameters such as an integration time for the pulse going into the CFD (DCR, 8/31/2006)

Description	Address	Access	Width
CFD Parameters 0	**0XC0**	**RW**	**32**
CFD Parameters 1	**0xC4**	**RW**	**32**
CFD Parameters 2	**0xC8**	**RW**	**32**
CFD Parameters 3	**0xCC**	**RW**	**32**
CFD Parameters 4	**0xD0**	**RW**	**32**
CFD Parameters 5	**0xD4**	**RW**	**32**
CFD Parameters 6	**0xD8**	**RW**	**32**
CFD Parameters 7	**0xDC**	**RW**	**32**
CFD Parameters 8	**0xE0**	**RW**	**32**
CFD Parameters 9	**0xE4**	**RW**	**32**

Description	Address	Access	Width
Raw data sliding length	*****	**RW**	**32**
Delay before we start retrieving raw data. Only bits 10 to 0 are used. Value at reset is 0x01C2 (4.5us). *Write-only*		**Bit Value**	
		1	**0**
Bits[10: 0]: Raw data sliding length		Value	
Bits[31:11]: TBD			

- There is one of this registers for each channels and their address are specified by the next table.

Description	Address	Access	Width
Raw data sliding length 0	**0x100**	**RW**	**32**
Raw data sliding length 1	**0x104**	**RW**	**32**
Raw data sliding length 2	**0x108**	**RW**	**32**
Raw data sliding length 3	**0x10C**	**RW**	**32**
Raw data sliding length 4	**0x110**	**RW**	**32**
Raw data sliding length 5	**0x114**	**RW**	**32**
Raw data sliding length 6	**0x118**	**RW**	**32**
Raw data sliding length 7	**0x11C**	**RW**	**32**
Raw data sliding length 8	**0x120**	**RW**	**32**
Raw data sliding length 9	**0x124**	**RW**	**32**

Description	Address	Access	Width
Raw data window	*	RW	32
Length of raw data retrieved. Only bits 9 to 0 are used. Value at reset is 0x32 (500ns). *Write-only*		Bit Value	
		1	0
Bits[9: 0]: Raw data length		Value	
Bits[31:10]: TBD			

- There is one of this registers for each channels and their address are specified by the next table.

Description	Address	Access	Width
Raw data window 0	0x140	RW	32
Raw data window 1	0x144	RW	32
Raw data window 2	0x148	RW	32
Raw data window 3	0x14C	RW	32
Raw data window 4	0x150	RW	32
Raw data window 5	0x154	RW	32
Raw data window 6	0x158	RW	32
Raw data window 7	0x15C	RW	32
Raw data window 8	0x160	RW	32
Raw data window 9	0x164	RW	32

Description	Address	Access	Width
DAC	0x400	RW	32
This register defines which signal is going to be written into the dac output at every clock cycle.		Bit Value	
		1	0
Bit [3 : 0]: Mux selection : 0 to 9 the DAC outputs the MSB of the channels 0 to 0 A DAC outputs x7F A DAC outputs x4A A DAC outputs x15 A DAC outputs xEB A DAC outputs counter up A DAC outputs bits 11 : 4		Value	
Bits[11: 4]: Data to be written by the DAC			
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
Slave front bus status	0x480	RW	32
		Bit Value	
		1	0
Bits[0]: slave front bus enabled		Value	
Bits[2:1]: slave board ID (00 Master board; 01,10 and 11 slave boards)		Value	
Bits[31: 3]: TBD		Value	

Description	Address	Access	Width
Channel zero Time stamp LSB	0x484	RW	32
After a latch command the TS on channel zero will be latched in this register.		Bit Value	
		1	0
Bits[31: 0]: TS [31:0]		Value	

Description	Address	Access	Width
Channel zero Time stamp MSB	0x488	RW	32
After a latch command the TS on channel zero will be latched in this register.		Bit Value	
		1	0
Bits[15: 0]: TS [47:32]		Value	
Bits[31:16]: TBD			

Description	Address	Access	Width
Slave Front Bus Send Box 18	0x48C	RW	32
Default value xDEADF00D.		Bit Value	
		1	0
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 17	**0x490**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 16	**0x494**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 15	**0x498**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 14	**0x49C**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 13	**0x4A0**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 12	**0x4A4**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 11	**0x4A8**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 10	**0x4AC**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 9	**0x4B0**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 8	**0x4B4**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 7	**0x4B8**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 6	**0x4BC**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 5	**0x4C0**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 4	**0x4C4**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 3	**0x4C8**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 2	**0x4CC**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 1	**0x4D0**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front bus register 00	**0x4D4**	**RW**	**32**
Bits zero to nine are commands that are going to be executed by the slave front bus. They self clear after the execution of the command.		**Bit Value**	
		1	**0**
Bits[0]: Sync command		Value	
Bits[1]: Front End Calibration Inject		Value	
Bits[2]: Latch Status		Value	
Bits[3]: Front End reset		Value	
Bits[4]: Imperative sync		Value	
Bits[5]: Readout command		Value	
Bits[9: 6]: TBD		Value	
Bits[10: 19]: FB register 1		Value	
Bits[20: 29]: FB register 2		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
Slave Front bus register 01	**0x4D8**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 02	**0x4DC**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 03	**0x4E0**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			
Description	**Address**	**Access**	**Width**
Slave Front bus register 04	**0x4E4**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 05	**0x4E8**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 06	**0x4EC**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 07	**0x4F0**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 08	**0x4F4**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 09	**0x4F8**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Slave Front bus register 10	**0x4FC**	**R**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[9: 0]: FB register 0		Value	
Bits[19:10]: FB register 1			
Bits[29:20]: FB register 2			
Bits[31:30]: TBD			

Description	Address	Access	Width
Master Logic Status	**0x500**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0]: Master Logic Enabled		Value	
Bits[1]: Start Snap Shot		Value	
Bits[2]: SD_Local_LE		Value	
Bits[3]: SD_Line_LE		Value	
Bits[4]: SD_PEM(0)		Value	
Bits[5]: SD_PEM(1)		Value	
Bits[6]: FB_Debug_Command (1 data transfer only)		Value	
Bits[7]: FB_Debug_Command (continuous data transfer)		Value	
Bits[8]: Reset Serdes lost lock flag		If 1 keeps the flag on reset.	
Bits[9]: Snap Shot reset			
Bits[10]: Snap Shot read data select		0 -> SD_RX 1 -> SD_TX	
Bits[11]: Snap Shot trigger on CC_LED event		Value	
Bits[15:12]: TBD		Value	
Bits[16]: FIFO Full		Value	
Bits[17]: SD_Lock		Value	
Bits[18]: HasDataFlag		Value	
Bits[19]: FB_Busy high level logic		Value	
Bits[20]: FB_Busy low level logic		Value	
Bits[21]: Snap Shot Busy RX		Value	
Bits[22]: Snap Shot Empty RX		Value	
Bits[23]: Serdes RX SM in sync with TTCL system		Value	
Bits[24]: Serdes lost lock flag		Value	
Bits[25]: Snap Shot Busy TX		Value	
Bits[26]: Snap Shot Empty TX		Value	
Bits[31: 24]: TBD		Value	

Description	Address	Access	Width
GetSlowData_CCLED_timers	**0x504**	**RW**	**32**
Number of clock cycles that the CC_LED flag will stay high. The CC_LED flag signal is sent to the TTCL through one of the serdes auxiliary IOs.		**Bit Value**	
		1	**0**
Bits[0: 15]: time		Value	
Bits[16: 31]:		Value	

Description	Address	Access	Width
DeltaT155_DeltaT255	**0x508**	**RW**	**32**
Limits of the time windows for the comparison type x55.		**Bit Value**	
		1	**0**
Bits[0: 11]:		Value	
Bits[16: 27]:		Value	

Description	Address	Access	Width
DeltaT15A_DeltaT25A	**0x50C**	**RW**	**32**
Limits of the time windows for the comparison type x5A.		**Bit Value**	
		1	**0**
Bits[0: 11]:		Value	
Bits[16: 27]:		Value	

Description	Address	Access	Width
DeltaT1A5_DeltaT2A5	**0x510**	**RW**	**32**
Limits of the time windows for the comparison type xA5.		**Bit Value**	
		1	**0**
Bits[0: 11]:		Value	
Bits[16: 27]:		Value	

Description	Address	Access	Width
SnapShot	**0x514**	**RW**	**32**
This register is used to read all the data captured by the snap shot memory. This memory stores the data received by the SerDes.		**Bit Value**	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
XTAL ID	**0x518**	**RW**	**32**
XTAL ID used on the slow data that is sent to the TTCL.		**Bit Value**	
		1	**0**
Bits[7: 0]: XTAL ID		Value	
Bits[31: 8]: TBD		Value	

Description	Address	Access	Width
Length of Time to Get Hit Pattern	**0x51C**	**RW**	**32**
If necessary to add a delay on the CC_LED signal that goes to the slave boards this register shows how big it is. Default zero		**Bit Value**	
		1	**0**
Bits[15: 0]: Time		Value	
Bits[31: 16]: TBD		Value	

Description	Address	Access	Width
Front Bus Debug Register	**0x520**	**RW**	**32**
Data that tell what information is to be read/written from the FB.		**Bit Value**	
		1	**0**
Bits[9: 0]: Write Data			
Bits[11: 10]: 00			
Bits[19: 12]: Address		Value	
Bits[27: 20]: TBD		Value	
Bit [28]: RNW		Value	
Bits[31: 29]: TBD		Value	

For example:
If the boards are located as follow: Master board on slot 6, slave boards on slots 5, 4 and 3.
Write command through the front bus
Write on 520h 30AAh
Write on 500h 41h
Write on 500h 01h
This will write on address 4D8h on the slave board on slot 4 through the front bus.
Read command through the front bus
Write on 520h 10003000h
Write on 500h 41h
Write on 500h 01h
This will transfer the data from address 4D8h on the slave board on slot 4 through the front bus to address 558h in the master board.

Description	Address	Access	Width
Test digitizer Rx TTCL	**0x524**	**RW**	**32**
In order to test some of the functionalities of the master logic for debug purpose there is a module that generates the TTCL data package inside the SD block. This register configures that module as described below.		**Bit Value**	
		1	**0**
Bits[31]: Test Enable		Value	
Bits[30:29]: Data package selection			
Bits[28]: Send alternate data package (one time only, rising edge)			
Bits[27:26]: Alternate data package selection			
Bits[25:19]: TBD			
Bits[18]: Write enable (store user def data on the rising edge)			
Bits[17:11]: Address to write the package for the slow data			
Bits[10]: Fast Data – Central contact LED			
Bits[9]: Fast Data – Central contact pile up			
Bits[8]: Fast Data – Error flag			
Bits[7: 0]: Slow Data			

Description	Address	Access	Width
Slave Front Bus Send Box 11	**0x528**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 10	**0x52C**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 9	**0x530**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 8	**0x534**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 7	**0x538**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 6	**0x53C**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 5	**0x540**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 4	**0x544**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 3	**0x548**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 2	**0x54C**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
Slave Front Bus Send Box 1	**0x550**	**RW**	**32**
Default value xDEADF00D.		**Bit Value**	
		1	**0**
Bits[31: 0]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 0	**0x554**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0]: Sync command		Value	
Bits[1]: Front End Calibration Inject		Value	
Bits[2]: Latch Status		Value	
Bits[3]: Front End reset		Value	
Bits[4]: Imperative sync		Value	
Bits[5]: Readout command		Value	
Bits[9: 6]: TBD		Value	
Bits[10: 19]: TBD		Value	
Bits[20: 29]: TBD		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 1	**0x558**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[7: 0]: Serdes slow data HP_A		Value	
Bits[15: 8]: Serdes slow data HP_B		Value	
Bits[23: 16]: Serdes slow data HP_C		Value	
Bits[29: 24]: Serdes slow data HP_D(5:0)			
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 2	**0x55C**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[1: 0]: Serdes slow data HP_D(7:6)		Value	
Bits[9: 2]: Serdes slow data HP_E		Value	
Bits[29: 10]: TBD		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 3	**0x560**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 4	**0x564**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 5	**0x568**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 6	**0x56C**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 7	**0x570**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 8	**0x574**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 9	**0x578**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
FrontBus Registers 10	**0x57C**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0: 9]:		Value	
Bits[10: 19]:		Value	
Bits[20: 29]:		Value	
Bits[31: 30]: TBD		Value	

Description	Address	Access	Width
Debug data buffer address	**0x780**	**RW**	**32**
Used by debug module (see section XIII of the GRETINA VHDL modules description document).		**Bit Value**	
		1	**0**
Bits[31: 0]:		Value	

Description	Address	Access	Width
Debug data buffer data	**0x784**	**RW**	**32**
Used by debug module (see section XIII of the GRETINA VHDL modules description document).		**Bit Value**	
		1	**0**
Bits[31: 0]:		Value	

Description	Address	Access	Width
LED Flag window	**0x788**	**RW**	**32**
Number of clock cycles the LED flag is kept high on the serdes fast data as well as on the RJ45 Auxiliary signals. Range varies from 1 (10ns) to 1024 (~1us).		**Bit Value**	
		1	**0**
Bits[31: 0]:		Value	

Description	Address	Access	Width
AUX_IO_READ	**0x800**	**RW**	**32**
Data read from the Auxiliary IO		**Bit Value**	
		1	**0**
Bits[11: 0]: Auxiliary IO input value		Value	
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
AUX_IO_WRITE	**0x804**	**RW**	**32**
Data written into the auxiliary IO		**Bit Value**	
		1	**0**
Bits[11: 0]: Auxiliary IO output value		Value	
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
AUX_IO_CONFIG	**0x808**	**RW**	**32**
Enable signals for the Auxiliary IO's drivers.		**Bit Value**	
		1	**0**
Bits[0]: EN_AUX2A_TX			
Bits[1]: EN_AUX2A_RX			
Bits[2]: EN_AUX2B_TX			
Bits[3]: EN_AUX2B_RX			
Bits[4]: EN_AUX2C_TX			
Bits[5]: EN_AUX2C_RX			
Bits[6]: EN_AUX2D_TX			
Bits[7]: EN_AUX2D_RX			
Bits[8]: EN_AUX2E_TX			
Bits[9]: EN_AUX2E_RX			
Bits[10]: EN_AUX2F_TX			
Bits[11]: EN_AUX2F_RX			
Bits[15: 12]: MUX control*		0000 -> Debug mode 0001 -> Mode 1 of operation 0010 -> Mode 2 of operation others-> Mode 3 of operation	
Bits[31:12]: TBD		Value	

* Mode 1 of operation:

Auxiliary IO signal	Signal type (input/output)	Routed signal to the Aux_IO
Aux IO (0)	Output	Global Trigger signal
Aux IO (1)	Output	100MHz clock
Aux IO (2)	Input	Global Validate/External trigger
Aux IO (3)	Input	Sync signal NOT ('1' or 5V resets time stamp)
Aux IO (4)	Input	TBD
Aux IO (5)	Input	TBD
Aux IO (6)	Input	TBD
Aux IO (7)	Input	TBD
Aux IO (8)	Input	TBD
Aux IO (9)	Input	TBD
Aux IO (10)	Input	TBD

* Mode 2 of operation:

Auxiliary IO signal	Signal type (input/output)	Routed signal to the Aux_IO
Aux IO (0)	Output	Trigger channel (0)
Aux IO (1)	Output	Trigger channel (1)
Aux IO (2)	Output	Trigger channel (2)
Aux IO (3)	Output	Trigger channel (3)
Aux IO (4)	Output	Trigger channel (4)
Aux IO (5)	Output	Trigger channel (5)
Aux IO (6)	Output	Trigger channel (6)
Aux IO (7)	Output	Trigger channel (7)
Aux IO (8)	Output	Trigger channel (8)
Aux IO (9)	Output	Trigger channel (9)
Aux IO (10)	Input	TBD

* Mode 3 of operation:

Auxiliary IO signal	Signal type (input/output)	Routed signal to the Aux_IO
Aux IO (0)	Input	TBD
Aux IO (1)	Input	TBD
Aux IO (2)	Input	TBD
Aux IO (3)	Input	TBD
Aux IO (4)	Input	TBD
Aux IO (5)	Input	TBD
Aux IO (6)	Input	TBD
Aux IO (7)	Input	TBD
Aux IO (8)	Input	TBD
Aux IO (9)	Input	TBD
Aux IO (10)	Input	TBD

Description	Address	Access	Width
FB_READ	**0x820**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[11: 0]: Auxiliary IO input value		Value	
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
FB_WRITE	**0x824**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[11: 0]: Auxiliary IO output value		Value	
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
FB_CONFIG	**0x828**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[0]: FB_CLK_OUT_EN			
Bits[1]: FB_SD_DP_EN			
Bits[2]: FB_DATA_DIR			
Bits[3]: FB_DATA_EN			
Bits[4]: FB_CTRLO_DIR			
Bits[5]: FB_CTRL0_EN			
Bits[6]: FB_CTRL1_DIR			
Bits[7]: FB_CTRL1_EN			
Bits[8]: FB_DATA_TS			
Bits[9]: FB_ADDR_TS			
Bits[10]: FB_RNW_TS			
Bits[11]: FB_STRB_TS			
Bits[12]: FB_SPARE_TS			
Bits[13]: FB_LED_TS			
Bits[14]: FB_WORB_TS			
Bits[16: 15]: MUX control		00 -> Debug mode 01 -> Normal operation	
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
SD_READ	**0x840**	**RW**	**32**
Data read from the Serdes.		**Bit Value**	
		1	**0**
Bits[17: 0]: SD input value		Value	
Bits[30:18]: TBD			
Bits[31]: Lock signal		Value	

Description	Address	Access	Width
SD_WRITE	**0x844**	**RW**	**32**
Data written into the Serdes.		**Bit Value**	
		1	**0**
Bits[18: 0]: SD output value		Value	
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
SD_CONFIG	**0x848**	**RW**	**32**
Configuration signals for the SD x31 serdes disabled x22 Local loop and sync mode x02 Loop back mode Before using a loop back mode it is necessary to put the serdes in local loop and sync so the SD can lock and then move to the Loop back mode. If the DCM doesn't lock it is necessary to reset it. For reliable use it is recommended to reset the ten channel module.		**Bit Value** **1**	**0**
Bits[0]: SD_RPDWN			
Bits[1]: SD_LOCAL_LE			
Bits[2]: SD_PEM_0			
Bits[3]: SD_PEM_1			
Bits[4]: SD_TPDWN			
Bits[5]: SD_SYNC			
Bits[6]: SD_LINE_LE			
Bits[8: 7]: MUX control		00 -> Debug mode 01 -> Normal operation	
Bits[9]: DCMResetCommand : when asserted keeps the DCM for the 100MHZ clock under reset.			
Bits[10]: TenChannelResetCommand : Reset the ten channel module.			
Bits[31:11]: TBD		Value	

Description	Address	Access	Width
ADC_CONFIG	**0x84C**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[1:0]: Clock mux 00 => CLK 01 => CLK90 10 => CLK180 11 => CLK270			
Bits[2]: TBD			
Bits[11: 3]: TBD			
Bits[31:12]: TBD		Value	

Description	Address	Access	Width
Self Trigger Enable	**0x860**	**RW**	**32**
Set bit one to start the self trigger process. This bit resets itself when the number of trigger events is completed. This self trigger is valid on the external trigger mode only.		**Bit Value**	
		1	**0**
Bit 1: Self trigger start		Value	
Bits[31:2]: TBD		Value	

Description	Address	Access	Width
Self Trigger Period	**0x864**	**RW**	**32**
To get the period in time multiply this register by 10ns.		**Bit Value**	
		1	**0**
Bits[31:0]: Period between trigger events.		Value	

Description	Address	Access	Width
Self Trigger Count	**0x868**	**RW**	**32**
		Bit Value	
		1	**0**
Bits[31:0]: Number of trigger events.		Value	

2.2.5 DAC OUTPUT

The GRETINA Digitizer will utilize 1 general-purpose 100MHz Digital to Analog converter (DAC) that will have an output available at the front panel of the module. The design will be implemented with a Current output DAC to allow signal summing using a NIM module. The board will be designed to allow the option of loading the circuit as a differential voltage output DAC if required. The option must be selected at the time that the board is assembled and will not be user selectable in the system. Signals required by the DAC: Clock and 8 bit data word. (Analog Devices AD9748ACP)

2.2.6 FRONTBUS INTERFACE

The FrontBus Interface will be used by the Master Digitizer to distribute the system clock and to communicate with the three Slave Digitizers in each VME crate. The Master will set some control registers asking the boards to execute some task and will read and write from other registers in order to transfer information. The Master Digitizer is always connected to the Central Contact of the Crystal, so the time required to fetch hit patterns will not affect the ability to supply information to the “Fast Decision Trigger” algorithms.

The 50MHz System Clock is recovered from the Serial Data transmitted by the Global Trigger and Timing crate and then routed as a differential pair to all of the Digitizers over the FrontBus. The 50MHz clock is multiplied to 100MHz in the Main FPGA and then used to digitize the input data for each channel.

A block diagram of the system clock distribution circuit is shown in Figure 11.

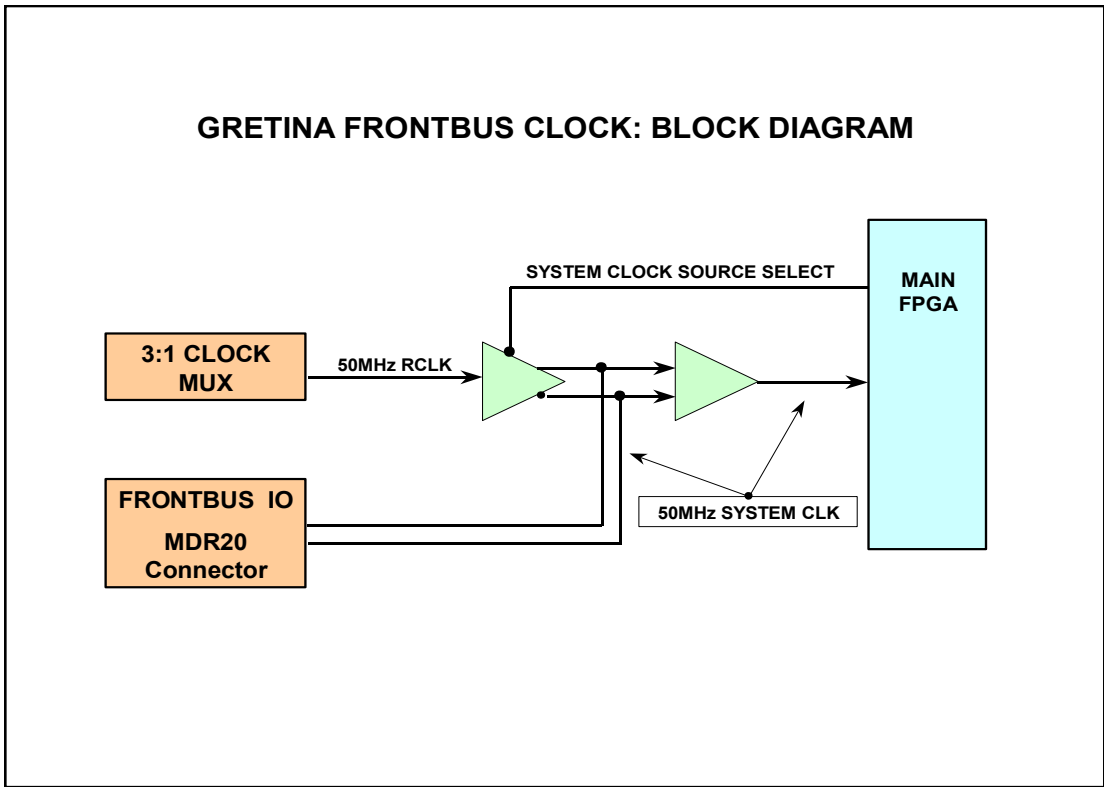


Figure 11: FrontBus Clock Distribution

Data Bus: 10 bit bi-directional parallel bus

a. Valid data latched on each clock transition
b. The data and command bus are implemented using standard LVT245 transceivers (Bus Hold and built in series termination resistors)

Address Bus: 8 bits wide

Table 6 – FrontBus Base Address, bits[7:6]

FrontBus Base Address (Bits [7:6]	Target Device for Read Access only
00	Digitizer 0 (Master)
01	Digitizer 1
10	Digitizer 2
11	Digitizer 3

For Read access, the Master activates one Digitizer per complete cycle.

Bit 5 is used to indicate a Broadcast function is active on the FrontBus.

Table 7 – FrontBus commands, Address bits[5:0]

FrontBus Command	Command Name	Command Description
100000	Write trigger data to slave.	This command performs a read out operation and will send the Header Memory index in the data.
1000001	Reset TS	All time stamps will be reset.
1000010	Send TS	Send the time stamp that will be compared with the channel TS to verify if they are in sync
1000011	Latch status	Latch the status of all channels in the same time, so it is possible to compare the TS of the channels.
1000100	Calibration Inject	

* This is an option if we want to have more then 10 bits of status for every board.

Leading Edge Detect Wired-OR signal – 1 bit
This signal is used to indicate that a detector crystal has detected a leading edge.

Table 8 - **FrontBus Interface**

FrontBus Interface (40-pin MT Connector AMP)		
Pin	**NAME**	**Description**
1	**CLK_IO_P**	**FB Clock positive polarity**
2	**CLK_IO_N**	**FB Clock negative polarity**
3	**GND**	**Board Ground**
4	**GND**	**Board Ground**
5	**DATA(0)**	**FB Data Bus Bit 0**
6	**DATA(1)**	**FB Data Bus Bit 1**
7	**DATA(2)**	**FB Data Bus Bit 2**
8	**GND**	**Board Ground**
9	**DATA(3)**	**FB Data Bus Bit 3**
10	**DATA(4)**	**FB Data Bus Bit 4**
11	**DATA(5)**	**FB Data Bus Bit 5**
12	**GND**	**Board Ground**
13	**DATA(6)**	**FB Data Bus Bit 6**
14	**DATA(7)**	**FB Data Bus Bit 7**
15	**DATA(8)**	**FB Data Bus Bit 8**
16	**GND**	**Board Ground**
17	**DATA(9)**	**FB Data Bus Bit 9**
18	**ADDR(0)**	**FB Address Bit 0**
19	**ADDR(1)**	**FB Address Bit 1**
20	**GND**	**Board Ground**
21	**ADDR(2)**	**FB Address Bit 2**
22	**ADDR(3)**	**FB Address Bit 3**
23	**ADDR(4)**	**FB Address Bit 4**
24	**GND**	**Board Ground**
25	**ADDR(5)**	**FB Address Bit 5**
26	**ADDR(6)**	**FB Address Bit 6**
27	**ADDR(7)**	**FB Address Bit 7**
28	**GND**	**Board Ground**
29	**RNW**	**FB Read NOT Write Signal**
30	**STROBE**	**FB Data Strobe**
31	**SPARE**	**Spare Pin**
32	**GND**	**Board Ground**
33	**SPARE**	**Spare Pin**
34	**SPARE**	**Spare Pin**
35	**LE DETECT**	**FB Leading Edge Detect (Wired OR)**
36	**GND**	**Board Ground**
37	**SPARE W-OR**	**Spare Pin (Wired O R)**
38	**GND**	**Board Ground**
39	**SP DIFF PAIR**	**Spare Pin DIFF**
40	**SP DIFF PAIR**	**Spare Pin DIFF**

There will be two types of commands for the FrontBus, the read and write. The signal will be asserted on the falling edge of the clock and will be valid on the raising edge of the clock. The timing diagrams of these two commands are shown in Figure 12.

During a reset all the modules will start working as slave, so none of them are going to be driving the control signals. The Strobe*[1] line is the signal that tells the board that a new command needs to be executed. In order to avoid any false commands from being executed right after a reset this line will have a pull-up resistor.

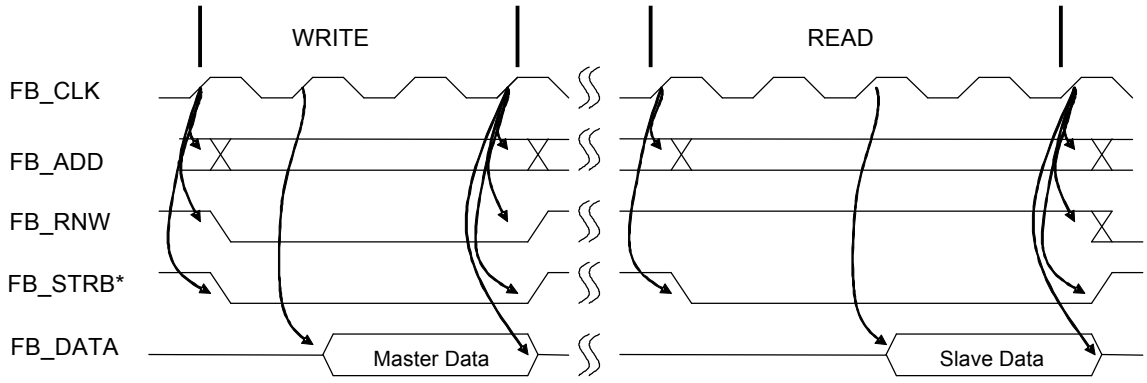


Figure 12: FrontBus Communication Timing Diagram – Master Write/Read Operations.

2.2.6.1 FRONT BUS REGISTERS

See section 2.2.4.6 for detailed descriptions and address locations.

2.2.7 SERIALIZER/DESERIALIZER INTERFACE

The Digitizer receives Trigger commands, transmits event information and receives the GRETINA System Clock through the Serializer/Deserializer (SER/DES) circuit. The 50 MHz system clock is generated by the Global Trigger and recovered by the SER/DES on each Master Digitizer. The circuit design includes the National Semiconductor DS92LV18TVV, LVDS SER/DES and a pre-emphasis IC used to boost the fidelity of the transmitted signals. A custom, shielded RJ45 to 2mm hard Metric cable will be used to connect the Digitizer SER/DES with the Global trigger SER/DES.

RJ-45 Connector Pin Assignments	
Pin Number	Description
1	Auxiliary In0 +
2	Auxiliary In0 -
3	SerDes TX Out +
4	Auxiliary In1 +
5	Auxiliary In1 -
6	SerDes TX Out -
7	SerDes RX Out +
8	SerDes RX Out -

[1] * means that the signal has its logic inverted.

More detailed documentation with regard to the SER/DES circuits can be found in these [1, 2, 3] documents:

2.2.8 AUXILIARY INPUT INTERFACE

The Auxiliary Input Interface is used to link the GRETINA Digitizers to other detectors and experiments. There are 16 configurable I/O signals on the front panel connector and one dedicated external clock input pair that can be used as a clock source for the Digitizer ADC circuits. High speed RS485 transmitters and receivers are used for the interface components and the configuration of inputs and outputs can be controlled in increments of 2 signal pairs. The available configuration I/O combinations are shown in Table 9. The connector used is a standard header with 0.1x0.1 spacing.

Table 9: Auxiliary I/O Configurations

INPUTS	OUTPUTS
11	0
10	1
8	2
6	4
4	6
2	8
1	10
0	11

Table 10: Auxiliary I/O Signal Definitions

PIN	FUNCTION	PIN	FUNCTION	PIN	FUNCTION
1	GND	2	Aux0 I/O +	3	Aux0 I/O -
4	GND	5	Aux1 I/O +	6	Aux1 I/O -
7	GND	8	Aux2 I/O +	9	Aux2 I/O -
10	GND	11	Aux3 I/O +	12	Aux3 I/O -
13	GND	14	Aux4 I/O +	15	Aux4 I/O -
16	GND	17	Aux5 I/O +	18	Aux5 I/O -
19	GND	20	Aux6 I/O +	21	Aux6 I/O -
22	GND	23	Aux7 I/O +	24	Aux7 I/O -
25	GND	26	Aux8 I/O +	27	Aux8 I/O -
28	GND	29	Aux9 I/O +	30	Aux9 I/O -
31	GND	32	Aux10 I/O +	33	Aux10 I/O -
34	GND	35	Ext Clock In +	36	Ext Clock In -

2.2.9 100MHZ CLOCK DISTRIBUTION

The MAIN FPGA will generate a 100MHz clock from the recovered 50MHz system clock and this clock will by distributed as an LVPEVL signal to all of the Analog Front Ends. The component used to distribute the 100MHz clock is a 10-port LVPECL Clock Repeater. The typical channel-to-channel output skew of the device is 100ps, and the part-to-part skew is 1.5ns.

2.2.10 FRONT PANEL INDICATORS

The Digitizer Front Panel will include 20 dual color indicators. The Main FPGA controls 13 indicators, the VME FPGA controls 4 indicators, and three power supplies are connected to the remaining 3 indicators. The indicators will visually indicate the status of the following items:

a. Status per channel: (CHAN 0, CHAN 1, …)
   i. Channel OFF – OFF
   ii. Channel ON – GREEN
   iii. Channel is ON and there is Leading Edge Discriminator – RED
   iv. Channel is ON and the data of this channel is being transferred to the Readout-FIFO – YELLOW

b. Status per board:
   i. VME access (V0) - YELLOW
   ii. VME is accessing Readout FIFO
   iii. VME is accessing MAIN FPGA
   iv. The Readout-FIFO has overflowed (i.e., the channels are writing data faster than VME can readout)
   v. The SER/DES is in lock. – GREEN
   vi. The MAIN FPGA clock circuit is in lock – GREEN

17 of the 20 indicators are controlled by either the Main or VME FPGA, are user definable, and can be modified with changes to the FPGA firmware. The 3 that are connected directly to the power supply planes, +3.3V, +5.0V, and -5.0V, cannot be changed.

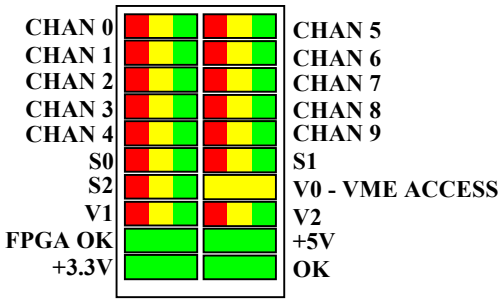


Figure 13: Diagram of Front Panel Indicators

2.2.11 VOLTAGE MONITORS

The Digitizer has a voltage monitor chip on-board to monitor key power supply voltages in real time. The monitor circuit includes an "All-OK" signal that is connected to the on-board power-on reset circuit and to a VME FPGA Status Register There is also an under-voltage signal and an over-voltage signal that are also connected to the same VME FPGA register. The monitored voltages are +3.3V, +5.0V, -5.0V, and +1.2V (Main FPGA). The circuit is powered by the +12V supply, and if the 1.2V VME core voltage is faulty than the board will not be readable on the VME bus.

2.2.12 TEMPERATURE MONITORS

There are three temperature-monitoring switches on the Digitizer that shunt closed if the temperature of a monitored location on the Digitizer board exceeds 70C. The temperature monitors are distributed around the board at locations that may experience damage if components were subjected to excessive heat. One temperature monitor is located in close proximity to the Main FPGA. These signals are all connected to a VME FPGA status register that can be monitored by the VME host.

2.2.13 POWER REQUIREMENTS

The GRETINA Digitizer Electronics board requires +5V, +3.3V, +12V, and -12V to operate correctly. The current requirements are still to be determined. All power supply inputs to the Digitizer Electronics board are fused.

3 DIGITIZER LAYOUT

3.1 COMPONENT PLACEMENT

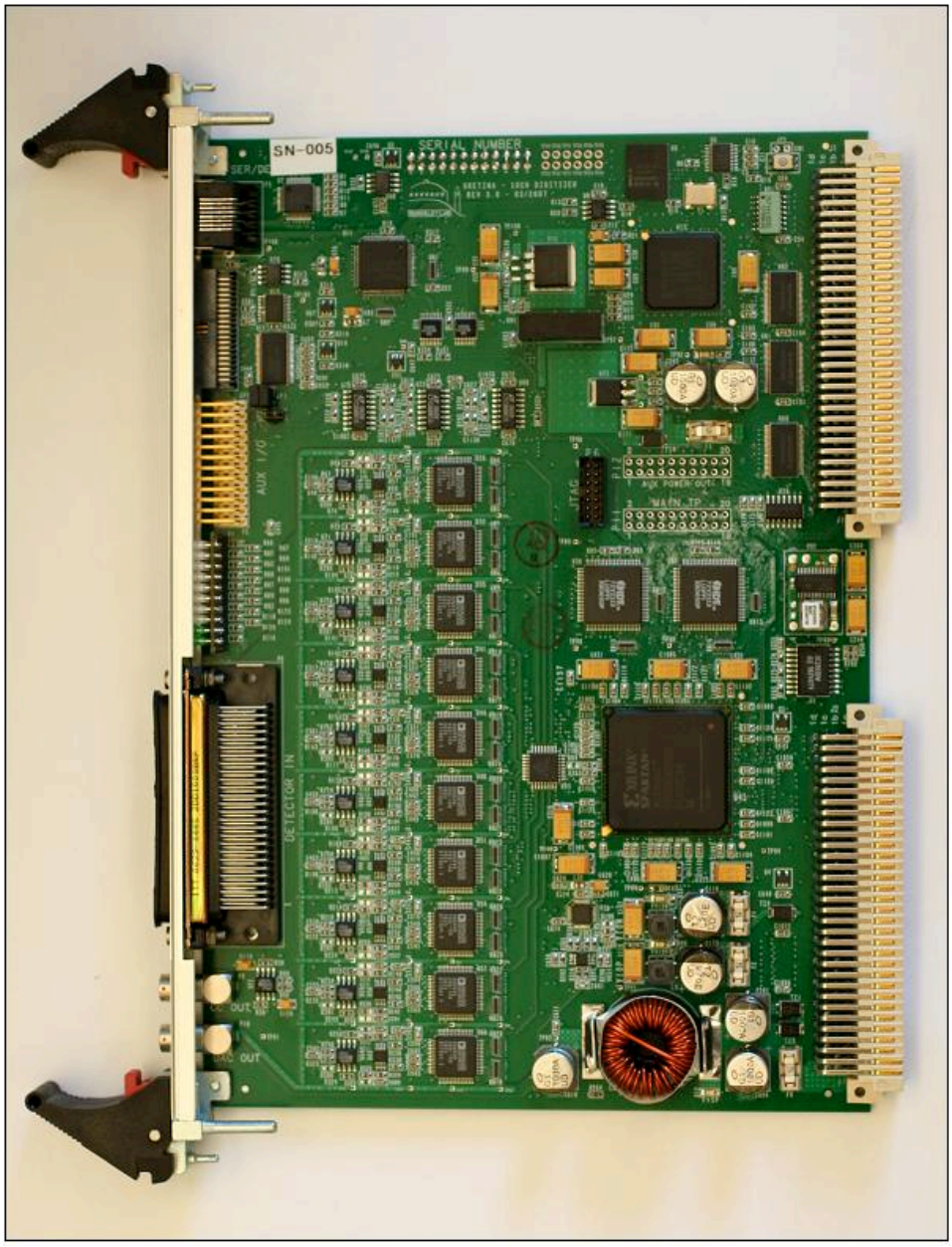


Figure 14: Rev3 DIGITIZER COMPONENT LAYOUT

4 REFERENCES

[1]Anderson, J. Link Specification: **Trigger Input Data Link**. v.1.3. GRETINA note. August, 2006.

[2] Anderson, J. Link Specification: **Trigger Module (Router & Master Trigger)**. v1.1. GRETINA note. August, 2006.

[3] Anderson, J.; Zimmermann, S. Link Specification: **Trigger Timing & Control Link**. v1.5. GRETINA note. August, 2006.

[4] Intel Technical: Datasheet, Intel ® Embedded Flash Memory (J3 v. D), Document number: 308551-003, February 2006.