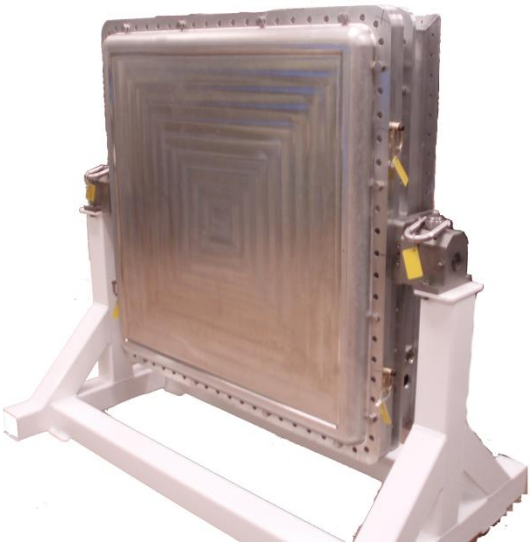
One Meter Neutron PAD Detector

User Guide



DRAFT

(WORK IN PROGRESS)

Jack Fried

10/25/2016

Contents

- System Boards and IO
- PAD BOARD
  - Pad channel locations
- IMP_ROC
  - Function overview
  - Register Map
- Rigid Flex Cable
  - Layout
  - ID selection
- Flange Transition Board
  - Function overview
  - ID Selection
- Master Timing & Control board
  - Function overview
  - Power
  - Register MAP
- Pizza Box

System Boards and IO

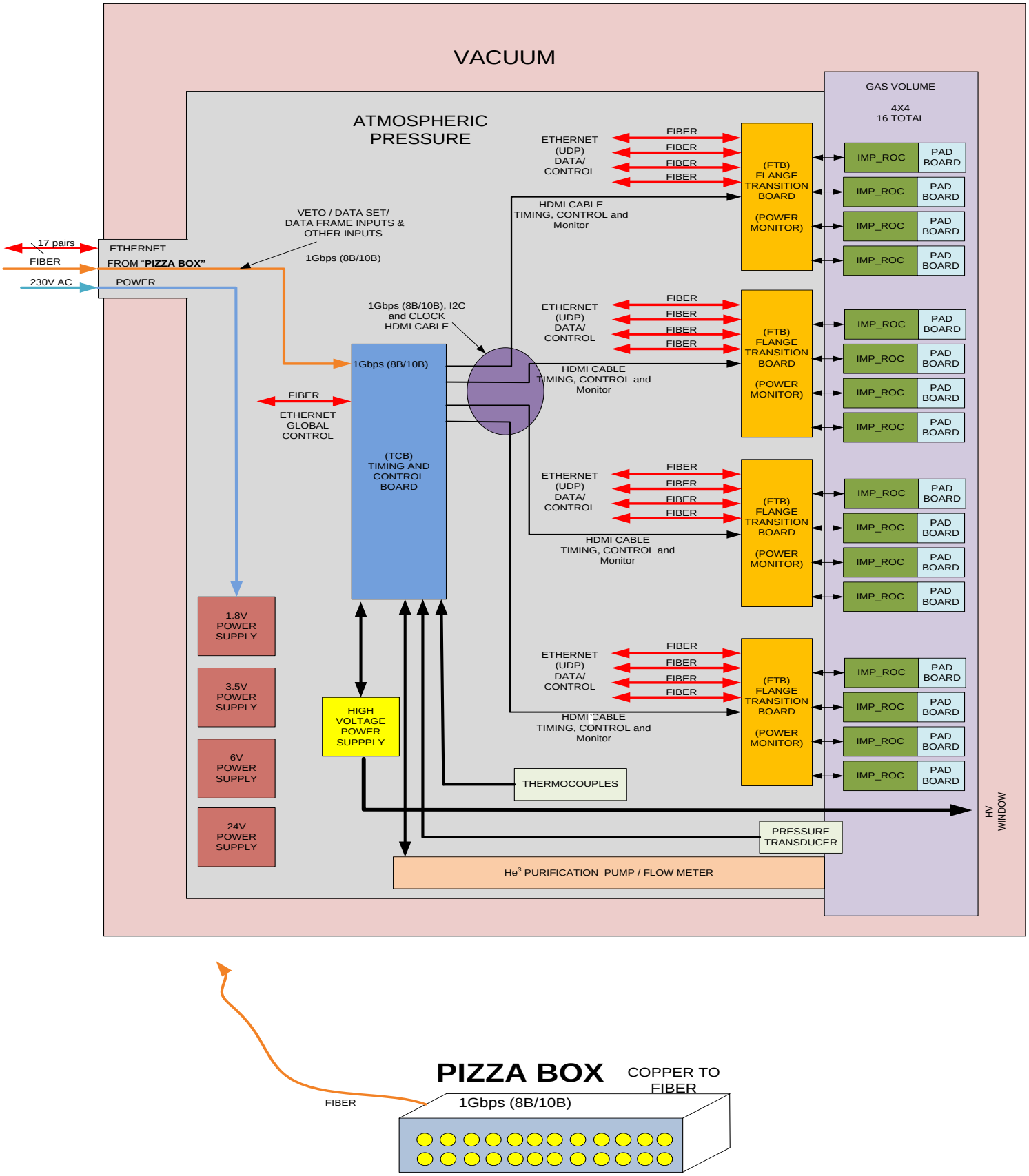


FIG 1.

As seen in Fig 1. the system contains the following

BOARDS

- Sixteen Pad Boards
- Sixteen IMP_ROC Boards
- Sixteen Rigid Flex Cables
  - There are four different versions of the rigid flex cables and each quadrant contains one of each.
- Four Flange Transition Boards
- One Master Timing and Control Board (MTC)
- One Pizza Box

System IO & Power

- Seventeen Gigabit Ethernet link
  - Sixteen links one for each IMP_ROC
  - One Link for the Master Timing and Control (MTC)
- One Unidirectional fiber link
  - An 8b10b link running at 1Gbps from the pizza box to the MTC this link is distributed to the all IMP_ROC boards and delivers timing , trigger & control information.
- Power INPUT 110 - 230V AC auto sensing input

PAD BOARD

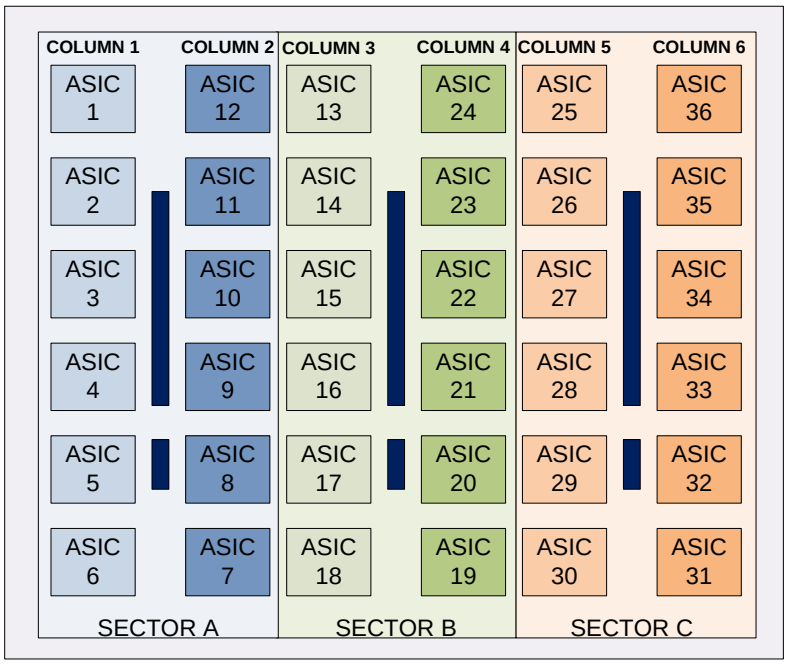


FIG 2.

The Pad board is made up of 36 custom ASIC's for more information about the ASICs refer to the ASIC reference document. For this guide it important to note that the PAD board is divided into two types of groups columns & Sectors. Certain functions of the pad board can only be performed by column or by sector

PAD BOARD Sector functions (12 ASICS)

- ASIC reset
- ASIC SPI programming
- ASIC TEST PULSE
- Sector Disable

PAD BOARD Column functions (6 ASICS)

- 2.5V Regulator monitor control
  - Monitors voltage and current for both analog and digital supply
  - Analog and digital regulators can be disabled
- Analog monitor
  - Used by on board ADC for system diagnostics
  - Each column shares one analog bus
- Column disable

PAD BOARD CHANNEL MAP

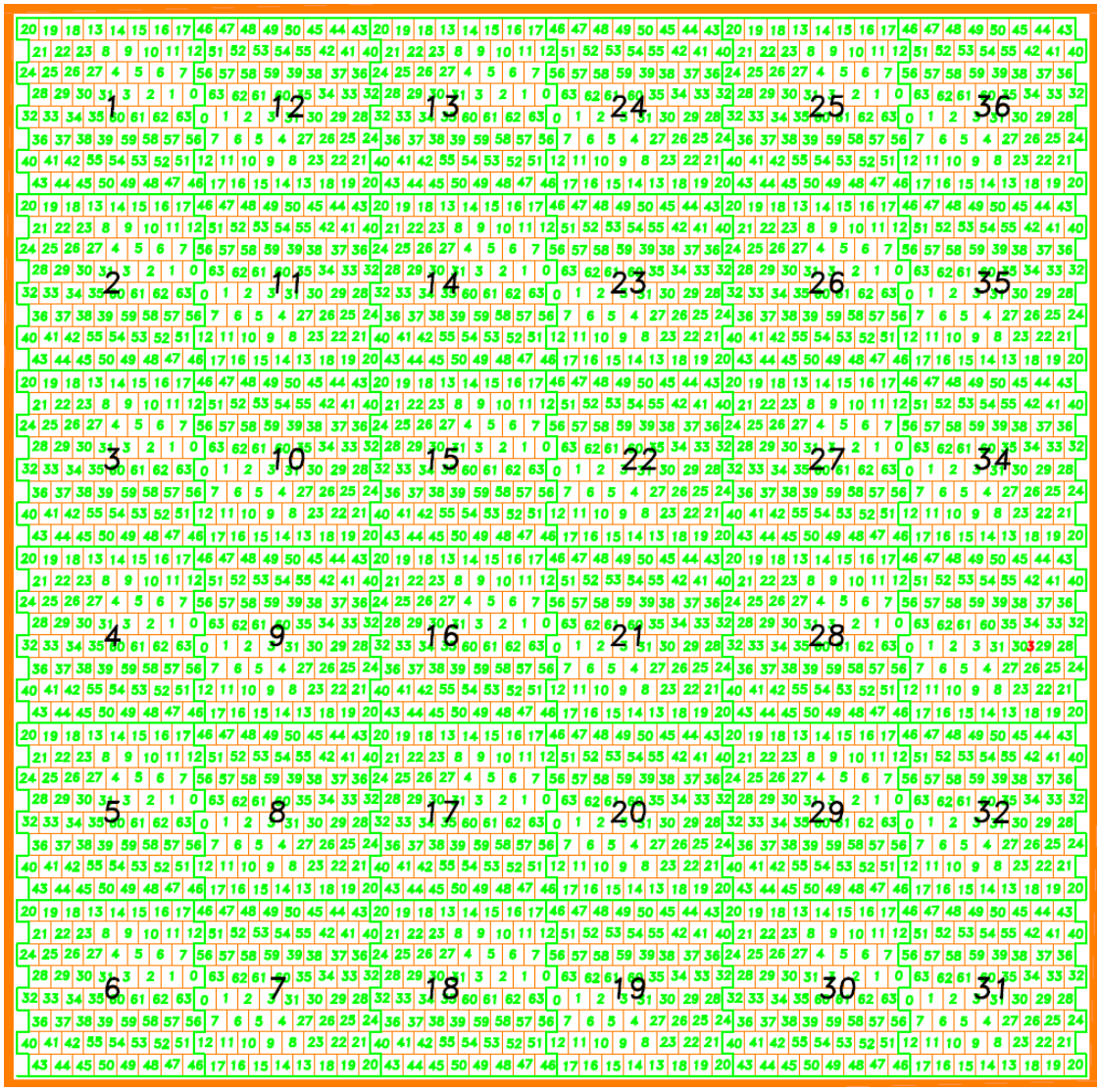


FIG 3.

Code used in C# to generate X,Y coordinates (48x48)

```
private uint[] PIXEL_MAP = {39, 38, 37, 36, 44, 45, 46, 47, 51, 52, 53, 54, 55, 59, 60, 61, 62, 63, 58, 57, 56, 48, 49, 50, 40, 41, 42, 43, 32, 33, 34, 35, 24, 25, 26, 27, 16, 17, 18, 19, 8, 9, 10, 0, 1, 2, 7, 6, 5, 4, 3, 15, 14, 13, 12, 11, 23, 22, 21, 20, 28, 29, 30, 31 };
private uint[] CHIP_MAP_Y = { 0, 8, 16, 24, 32, 40, 40, 32, 24, 16, 8, 0, 0, 8, 16, 24, 32, 40, 40, 32, 24, 16, 8, 0, 0, 8, 16, 24, 32, 40, 40, 32, 24, 16, 8, 0 };
private uint[] CHIP_MAP_X = { 0, 0, 0, 0, 0, 0, 8, 8, 8, 8, 8, 8, 16, 16, 16, 16, 16, 16, 24, 24, 24, 24, 24, 24, 32, 32, 32, 32, 32, 32, 40, 40, 40, 40, 40, 40};
private uint[] CHIP_MAP_fl = { 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1 };

public void PAD_MAP(uint chn,uint chp,out uint x,out uint y)
{  uint chn_x,chn_y;
   chn_y = (PIXEL_MAP[chn] / 8);
   chn_x = PIXEL_MAP[chn] - (chn_y * 8);
   if (CHIP_MAP_fl[chp] == 0)
      y =  CHIP_MAP_Y[chp] + (7-chn_y);
   else
      y = CHIP_MAP_Y[chp] + chn_y;
   if( CHIP_MAP_fl[chp] == 0)
      x = CHIP_MAP_X[chp] +chn_x;
   else
      x = (CHIP_MAP_X[chp] + (7-chn_x));
}
```

IMP_ROC

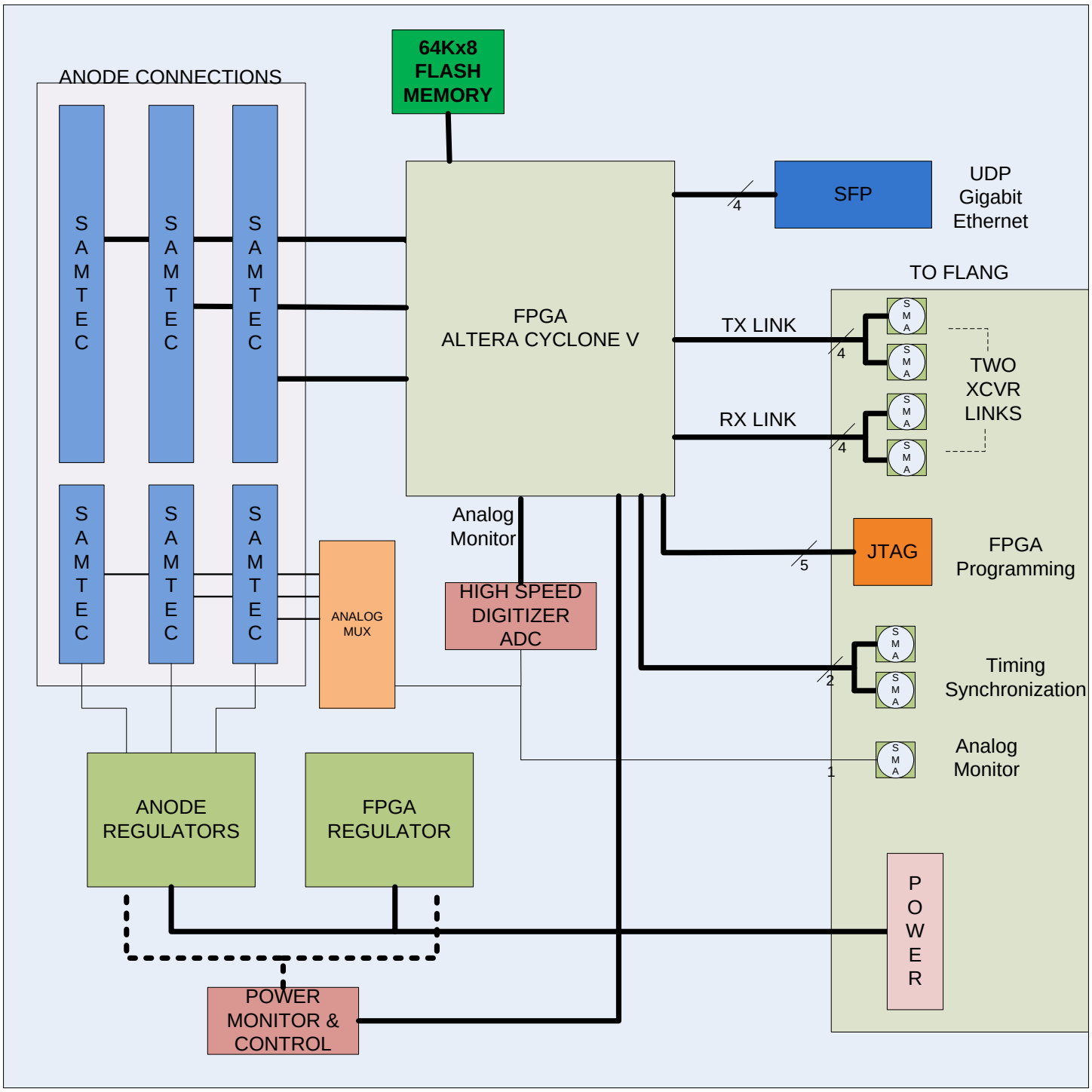

ANODE CONNECTIONS
SAMTEC
SAMTEC
SAMTEC
SAMTEC
SAMTEC
SAMTEC
64Kx8 FLASH MEMORY
FPGA ALTERA CYCLONE V
SFP
4
UDP Gigabit Ethernet
TO FLANG
TX LINK
RX LINK
4
4
SMA
SMA
SMA
SMA
TWO XCVR LINKS
Analog Monitor
JTAG
5
FPGA Programming
HIGH SPEED DIGITIZER ADC
ANALOG MUX
2
SMA
SMA
Timing Synchronization
1
SMA
Analog Monitor
ANODE REGULATORS
FPGA REGULATOR
POWER
POWER MONITOR & CONTROL

FIG 4.

IMP_ROC function

- Gigabit Ethernet link (UDP)
  - Programmable frame sizes (jumbo frames)
  - IP & MAC address determined by FTB + Rigid flex cable
- Gigabit timing link
  - 8b10b link used for synchronizing system with global commands
    - Time stamp reset
    - DAQ start and stop
    - Input signal from pizza box appended to sample data (with deterministic timing)
  - Recovered system clock
- Secondary Timing link
  - Differential clock received from MTC
  - Differential input signal received from MTC (serial interface)
- Cyclone V FPGA
  - Programmable over Ethernet or JTAG (JTAG port avalible on FTB)
- Control and read out 36 PAD board ASICS
- Monitor and control PAD board regulators
- 12 bit 40Msps ADC
  - Monitor ASIC analog output
  - On board voltage referance for ADC calibration
  - Adjustable sampling rate
  -
- On board flash memory 64Kx8
- Built-in test pulse generator

IMP_ROC REGISTER MAP

FPGA version 0x105

ADDRESS	NAME	R/W	BIT	Description
0	SYS_RESET	W	0	Set to reset entire system (AUTO Clears)
0	REG_RESET	W	1	Set to reset system registers (AUTO Clears)
0	UDP_RESET	W	2	Set to reset UDP ethernet state machine (AUTO Clears)
0	ALG_RESET	W	3	Set to reset all system state machines except UDP (AUTO Clears)
0	ERROR_CNT_RST	W	4	Set to reset all error counters (AUTO Clears)
1	ASIC RESET	R/W	2..0	Set to reset ASICS Bit 0 = Reset ASICS in sector A Bit 1 = Reset ASICS in sector B Bit 2 = Reset ASICS in sector C
1	TIME_STAMP_RST	R/W	4	Set to reset board time stamp
2	ASIC_DISABLE_A	R/W	11.0	Set to disable ASIC readout in Sector A (column 1-2) Bit 0 = Set to Disable Chip 1 Bit 1 = Set to Disable Chip 2 Bit 2 = Set to Disable Chip 3 . . Bit 11 = Set to Disable Chip 12
3	ASIC_DISABLE_B	R/W	11.0	Set to disable ASIC readout in Sector B (column 3-4) Bit 0 = Set to Disable Chip 7 Bit 1 = Set to Disable Chip 8 Bit 2 = Set to Disable Chip 9 . . Bit 11 = Set to Disable Chip 24
4	ASIC_DISABLE_C	R/W	11.0	Set to disable ASIC readout in Sector C (column 5-6) Bit 0 = Set to Disable Chip 25 Bit 1 = Set to Disable Chip 26 Bit 2 = Set to Disable Chip 27 . . Bit 11 = Set to Disable Chip 36
5	DSYNC_RATE	R/W	15..0	ASIC read clocks before a ASIC DSYNC is generated. (Default is = 0x0010) DO NOT CHANGE IF YOU DO BAD THINGS WILL HAPPEN....
5	TIME_STAMP CLOCK PERIOD	R/W	19..16	SET TO ADJUST TIME STAMP PERIOD 00 = 100nS 01 = 200nS 02 = 400nS 03 = 800nS 04 = 1.6uS 05 = 3.2uS OTHER = 100ns
6	SW_ASIC_CNTL_EN	R/W	0	Setting this bit will allow register control of ASIC input pins
7	SW_ASIC_FUNC_A	R/W	0	Allows controls of the FUNC_A pin on all ASICS ("SW_ASIC_CNTL_EN" register 6 must be enabled) (SPI WRITING OVERRIDES)
7	SW_ASIC_FUNC_B	R/W	1	Allows controls of the FUNC_B pin on all ASICS ("SW_ASIC_CNTL_EN" register 6 must be enabled)

7	SW_ASIC_MODE	R/W	5..4	Allows controls of the MODE pins on all ASICS (SPI WRITING OVERRIDES)
7	SW_ASIC_SER_IN	R/W	8	Allows controls of the serial in pins on all ASICS (SPI WRITING OVERRIDES)
7	SW_ASIC_CLK	R/W	9	Allows controls of the FUNC_B pin on all ASICS ("SW_ASIC_CNTL_EN" register 6 must be enabled) (SPI WRITING OVERRIDES)
8	SECTOR_DISABLE	R/W	2..0	SET TO DISABLE AN ENTIRE SECTOR Bit 0 = SECTOR A Bit 1 = SECTOR B Bit 2 = SECTOR C
8	ASIC_FIX_DISABLE	4	4	Set to disable ASIC reset function (fix for ver 2 of asic) use register 32 to set period.
8	ALG_DATA_TEST	R/W	5	System TEST MODE -- can be used for UDP rate tests Set to fake data from all ASIC channels
8	RATE_METER_WIN	R/W	6	Rate meter window select 0 = 100mS 1 = 1s Select rate meter window based and event rate.
9	ADC_PEAK_TRIGGER	R/W	15..0	Use ADC calibration data for LSB 1 LSB = ~492uV
9	ADC_PK_TIMEOUT	R/W	31..16	Not used
10	CTEST_PERIOD	R/W	23..0	Calibration test pulse (CTEST) period control CTEST_PERIOD = Period/ 100n
10	TIME_STAMP CLOCK PERIOD	R/W	29	Do not use set to "0"
10	CTEST_SW	R/W	30	Register control for CTEST "CTEST_CNTL" needs to be enabled 0= ctest low 1=ctest high
10	CTEST_CNTL	R/W	31	Set to disable CTEST generator & enable CTEST_SW
11	AMON_MUX_SEL	R/W	2..0	Analog monitor multiplexer selection 0 = Column 1 ASICs sector A 1 = Column 2 ASICs sector A 2 = Column 3 ASICs sector B 3 = Column 4 ASICs sector B 4 = Column 5 ASICs sector C 5 = Column 6 ASICs sector C 6= 0.9V Fixed voltage for ADC calibration 7 = 1.5V Fixed voltage for ADC calibration
12	SPI_WRITE	R/W	2..0	Set high to write ASIC in corresponding SECTOR Bit 0 = WRITE TO ASICs in SECTOR A Bit 1 = WRITE TO ASICs in SECTOR B Bit 2 = WRITE TO ASICs in SECTOR C
13	SPI_WRT_REG0	R/W	31..0	
14	SPI_WRT_REG1	R/W	31..0	
15	SPI_WRT_REG2	R/W	31..0	
16	SPI_WRT_REG3	R/W	31..0	
17	SPI_WRT_REG4	R/W	31..0	
18	SPI_WRT_REG5	R/W	31..0	
19	SPI_RDBK_REG0	R	31..0	
20	SPI_RDBK _REG1	R	31..0	
21	SPI_RDBK _REG2	R	31..0	
22	SPI_RDBK _REG3	R	31..0	
23	SPI_RDBK _REG4	R	31..0	
24	SPI_RDBK _REG5	R	31..0	
25	ASIC_PWR_DISABLE	R/W	11..0	Shut down regulators for ASIC SECTORS Set to disable 0 = Regulator on 1= regulator ff

				Bit 0 = Analog supply for column 1 sector A Bit 1 = Digital supply for column 1 sector A Bit 2 = Analog supply for column 2 sector A Bit 3 = Digital supply for column 2 sector A Bit 4 = Analog supply for column 3 sector B Bit 5 = Digital supply for column 3 sector B Bit 6 = Analog supply for column 4 sector B Bit 7 = Digital supply for column 4 sector B Bit 8 = Analog supply for column 5 sector C Bit 9 = Digital supply for column 5 sector C Bit 10 = Analog supply for column 6 sector C Bit 11 = Digital supply for column 6 sector C
26	S_PWR_MON_CONV	R/W	0	Set to start ASIC power monitor conversions results are in register 41-56. (user must clear)
27	HS_DATA_SELECT	R/W	3..0	Select type of data to be transmitted over the high speed link (high speed link data format described below) 0 = ASIC DATA 1 = TEST COUNTER 2 = ADC DATA
28	ADC_SAMPLE_RATE	R/W	23..0	Set to adjust ADC sample rate 0 = DNS 1 = 25nS 2 = 50nS . .
28	ADC_DISABLE	R/W	31	Set to disable and power down ADC
29	FPGA_F_OP_CODE	R/W	7..0	Set this to the proper FLASH OP-CODE (ALTERA EPCS) OP_CODES 0x02 = Write memory -- write enable must be set 0x03 = Read memory 0x04 = Disable Write Enable 0x05 = Read Status 0x06 = Write Enable (use before write @ bulk erase) 0xAB = Read Silicon ID (result is stored in status reg) 0xC7 = Erase bulk (write enable must be set)
29	FPGA_F_STRT_OP	R/W	8	Set this bit to start FPGA FLASH operation (user must clear) (ALTERA EPCS)
29	JTAG_MODE	R/W	31	Enable Flash programming mode or allow eeprom to be seen on JTAG scan.
30	FPGA_F_ADDR	R/W	23..0	Set for start address of the FPGA FLASH 256 byte operation (ALTERA EPCS)
31	FPGA_F_status	R	1..0	Status returned by FPGA FLASH after operation Bit 0 = Write in progress Bit 1 = Write enable latch bit set (ALTERA EPCS)
32	ASIC_FUNC FIX_PERIOD	R/W	31..0	Period of func A NOT used for new ASICS
33	EEPROM_ADDR	R/W	15..0	EEPROM ADDRESS (32bit words)
33	EERPROM_RD	R/W	16	Set then clear to generate a read operation result stored in register 35
33	EEPROM_WR	R/W	17	Set then clear to generate a write operation
33	EEPROM_ALG_RST	R/W	31	Set then clear to reload the values stored in the EEPROM to the FPGA
34	EEPROM_WR_DATA	R/W	31..0	Data to write to EEPROM.

35	EEPROM_RD_BK DATA	R	31..0	Value returned from EEPROM after a EEPROM_RD
36	UDP_PACKET_SIZE	R/W	11..0	Set UDP DATA PACKET SIZE. The value in this register is based on 16 bit data words. The maximum value is 0xf00 and the minimum value is 0x0ff -- if a value is set out of range a default value of 0x1f8 is set
37	UDP_PACKET_TIME OUT	R/W	31..0	This register sets the timeout value for the UDP ENGINE to start to transmit a partial UDP packet. 1 LSB = 12.5ns (defalut value is 0x01000)
38	USER_PKT_HEADER_	R/W	23..0	24 bits of the UDP USER PACKET HEADER
39	SCRATCH_PAD	R/W	31..0	Register scratch pad -- safe register to play with
40	HS_DATA_DISABLE	R/W	0	Set to Disable high speed UDP data transfers
41	PW_S1_VCC_TEMP	R	31..0	
42	PW_S1_V12	R	31..0	
43	PW_S1_V34	R	31..0	
44	PW_S1_C12	R	31..0	
45	PW_S1_C34	R	31..0	
46	PW_S2_VCC_TEMP	R	31..0	
47	PW_S2_V12	R	31..0	
48	PW_S2_V34	R	31..0	
49	PW_S2_C12	R	31..0	
50	PW_S2_C34	R	31..0	
51	PW_S2_VCC_TEMP	R	31..0	
52	PW_S2_V12	R	31..0	
53	PW_S2_V34	R	31..0	
54	PW_S2_C12	R	31..0	
55	PW_S2_C34	R	31..0	
56	PW_S13_VCC_VCC	R	31..0	
57	OVERFLOW_ERR_A	R	31..0	
58	OVERFLOW_ERR_B	R	31..0	
59	OVERFLOW_ERR_C	R	31..0	
60				
100	VERSION ID	R	31 ..0	FIRMWARE VERSION ID
128-191	FPGA_F_WR MEMORY	R/W	31..0	Storage for Writing FPGA EPCS 256 bytes
192-255	FPGA_F_RD MEMORY	R/W	31..0	Storage for read back of FPGA EPCS 256 bytes

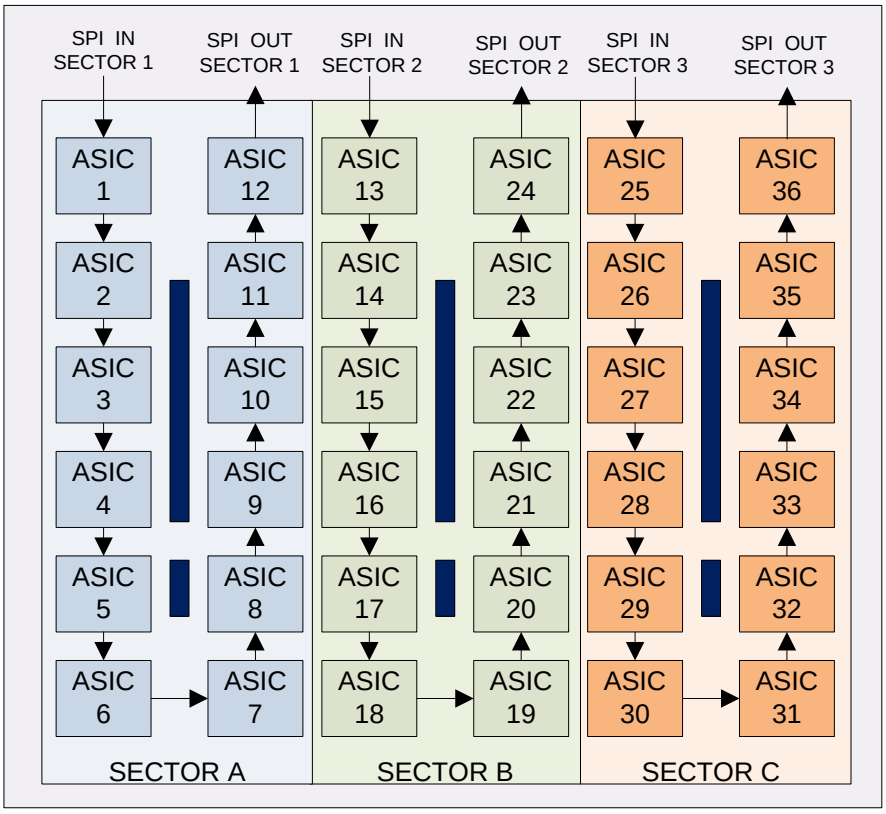
Rigid Flex Cable

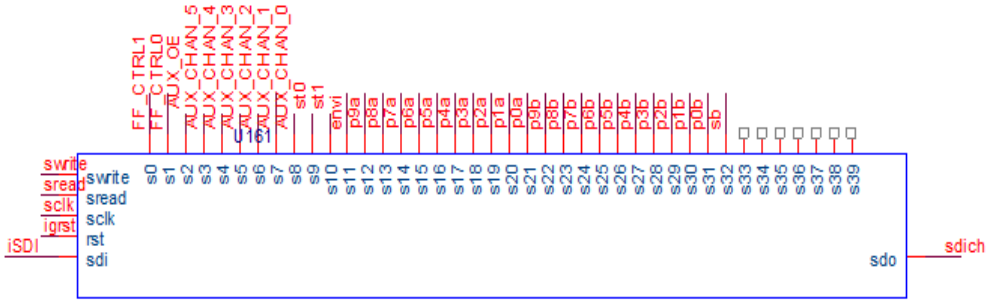
Flange Transition Board

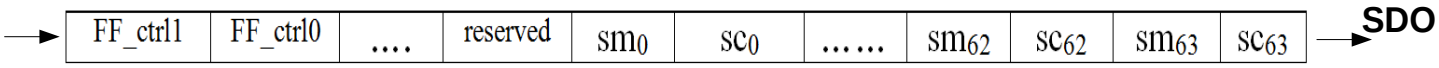
(FTB)

Master Timing & Control Board (MTC)

ASIC SPI REGISTERS PACKING



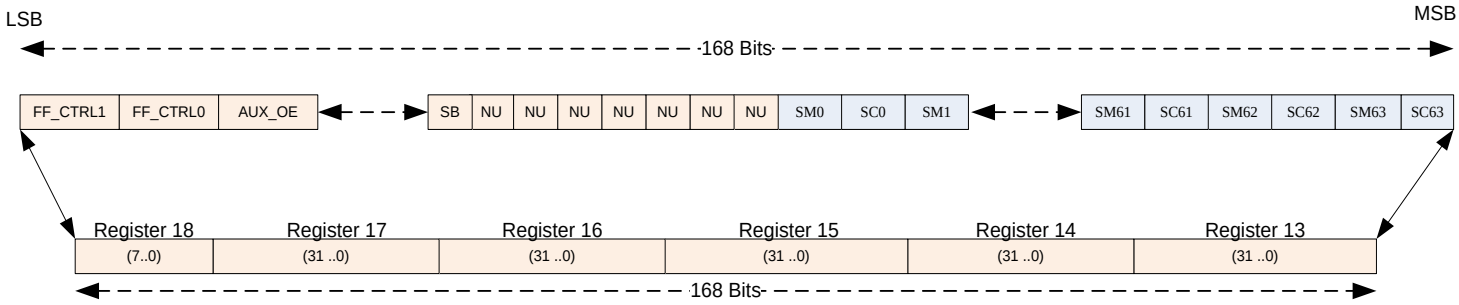




SMx = Channel Mask 0 = Channel on 1= channel off
SCx = Calibration Enable 0 = Calibration off 1= Calibration on
NU = Not Used (RESERVED)
SB = Output buffer enable 0 = disable 1= enable
pxb = DAC Threshold PB (9..0) 10 bits
pxa = DAC Test Pulse PA (9..0) 10 bits
Envi = Analog to digital converter enable 0 = enable 1 = stop
STx = Peaking time set ST(0-1) n=00→0. 5µs,
n=01→1.0µs,
n=10→2.0µs,
n=11→ 4.0µs
AUX_CHNx = Analog monitor Channel select
AUX_OE = Analog Monitor function control / output enable
Auxiliary output control oe, a_5, a_4, a_3, a_2, a_1, a_0 (see table below)

OE	a_5	a_4	a_3	a_2	a_1	a_0	
0	0	0	0	0	0	0	Auxiliary output is disabled
1	x	x	x	x	x	x	Channel output enable (00000=ch0)
0	0	0	0	0	0	1	Threshold DAC output enable
0	0	0	0	0	1	0	Test DAC output (full scale) enable
0	0	0	0	0	1	1	Test pulse output enable

FF_CTRLx = Fifo Flag Control 00 = Completely Empty (**DO NOT USE ANY OTHER VALUE** !!!!!)



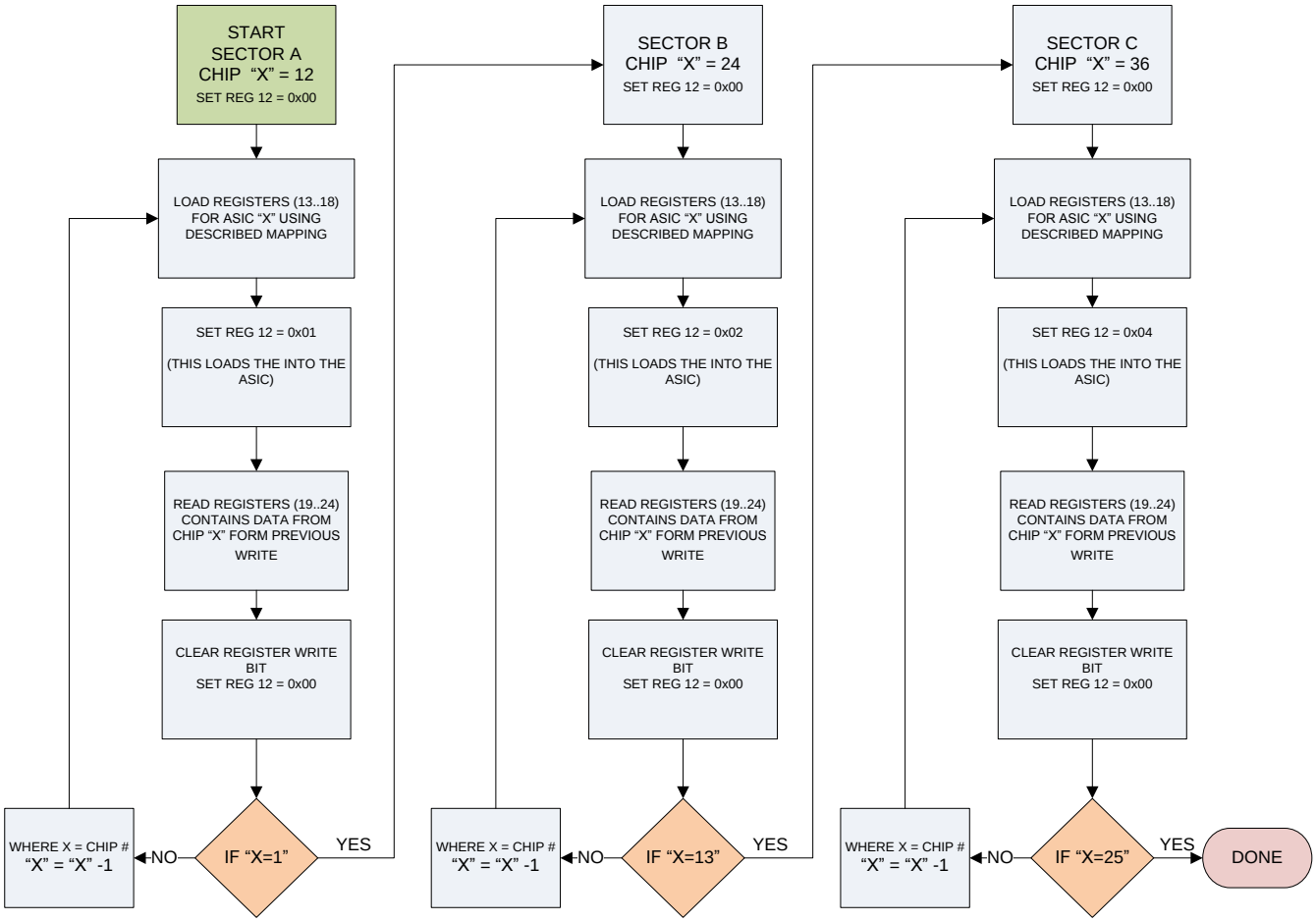
SPI WRITE REGISTERS

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG.13	SM48	SC48	SM40	SC49	SM50	SC50	SM51	SC51	SM52	SC52	SM53	SC53	SM54	SC54	SM55	SC55	SM56	SC56	SM57	SC57	SM58	SC58	SM50	SC59	SM60	SC60	SM61	SC61	SM62	SC62	SM63	SC63
REG.14	SM32	SC32	SM33	SC33	SM34	SC34	SM35	SC35	SM36	SC36	SM37	SC37	SM38	SC38	SM30	SC39	SM40	SC40	SM41	SC41	SM42	SC42	SM43	SC43	SM44	SC44	SM45	SC45	SM46	SC46	SM47	SC47
REG.15	SM16	SC16	SM17	SC17	SM18	SC18	SM19	SC19	SM20	SC20	SM21	SC21	SM22	SC22	SM23	SC23	SM24	SC24	SM25	SC25	SM26	SC26	SM27	SC27	SM28	SC28	SM29	SC29	SM30	SC30	SM31	SC31
REG.16	SM0	SC0	SM1	SC1	SM2	SC2	SM3	SC3	SM4	SC4	SM5	SC5	SM6	SC6	SM7	SC7	SM8	SC8	SM9	SC9	SM10	SC10	SM11	SC11	SM12	SC12	SM13	SC13	SM14	SC14	SM15	SC15
REG.17	AUX0	ST0	ST1	ENVI	P9A	P8A	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A	P9B	P8B	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	SB	NU	NU	NU	NU	NU	NU	NU
REG.18	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	FFC1	FFC0	AUOE	AUX5	AUX4	AUX3	AUX2	AUX1

SPI READ REGISTERS

BIT	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG.19	SM48	SC48	SM40	SC49	SM50	SC50	SM51	SC51	SM52	SC52	SM53	SC53	SM54	SC54	SM55	SC55	SM56	SC56	SM57	SC57	SM58	SC58	SM50	SC59	SM60	SC60	SM61	SC61	SM62	SC62	SM63	SC63
REG.20	SM32	SC32	SM33	SC33	SM34	SC34	SM35	SC35	SM36	SC36	SM37	SC37	SM38	SC38	SM30	SC39	SM40	SC40	SM41	SC41	SM42	SC42	SM43	SC43	SM44	SC44	SM45	SC45	SM46	SC46	SM47	SC47
REG.21	SM16	SC16	SM17	SC17	SM18	SC18	SM19	SC19	SM20	SC20	SM21	SC21	SM22	SC22	SM23	SC23	SM24	SC24	SM25	SC25	SM26	SC26	SM27	SC27	SM28	SC28	SM29	SC29	SM30	SC30	SM31	SC31
REG.22	SM0	SC0	SM1	SC1	SM2	SC2	SM3	SC3	SM4	SC4	SM5	SC5	SM6	SC6	SM7	SC7	SM8	SC8	SM9	SC9	SM10	SC10	SM11	SC11	SM12	SC12	SM13	SC13	SM14	SC14	SM15	SC15
REG.23	AUX0	ST0	ST1	ENVI	P9A	P8A	P7A	P6A	P5A	P4A	P3A	P2A	P1A	P0A	P9B	P8B	P7B	P6B	P5B	P4B	P3B	P2B	P1B	P0B	SB	NU	NU	NU	NU	NU	NU	NU
REG.24	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	0X00	FFC1	FFC0	AUOE	AUX5	AUX4	AUX3	AUX2	AUX1

SPI Programming flow chart



ETHERNET SETUP

- **DEVICE IP 192.168.1XY.001 MTC = 192.168.010.001**
  - **X = FLANGE ID (0 <->3)**
  - **Y = CABLE ID (0<->3)**
- **DEVICE MAC: AABBCCDDEEXY**
  - **X = FLANGE ID (0 <->3)**
  - **Y = CABLE ID (0<->3)**
- **SYSTEM KEY = 0xDEADBEEF**
- **IMP_ROC /MTC ETHERNET PORTS**
  - **32000 write port -- Used to write registers**
  - **32001 read request port -- Used to read registers**
  - **32002 response port -- Used in respond to a read request**
  - **32003 high speed data port -- Used to receive alert & high speed data**

Ethernet Address MAP

IMP ROC 0 192.168.100.001 MAC aa.bb.cc.dd.ee.00	IMP ROC 1 192.168.101.001 MAC aa.bb.cc.dd.ee.01	IMP ROC 4 192.168.110.001 MAC aa.bb.cc.dd.ee.10	IMP ROC 5 192.168.111.001 MAC aa.bb.cc.dd.ee.11
IMP ROC 2 192.168.102.001 MAC aa.bb.cc.dd.ee.02	IMP ROC 3 192.168.103.001 MAC aa.bb.cc.dd.ee.03	IMP ROC 6 192.168.112.001 MAC aa.bb.cc.dd.ee.12	IMP ROC 7 192.168.113.001 MAC aa.bb.cc.dd.ee.13
IMP ROC 8 192.168.120.001 MAC aa.bb.cc.dd.ee.20	IMP ROC 9 192.168.121.001 MAC aa.bb.cc.dd.ee.21	IMP ROC 12 192.168.130.001 MAC aa.bb.cc.dd.ee.30	IMP ROC 13 192.168.131.001 MAC aa.bb.cc.dd.ee.31
IMP ROC 10 192.168.122.001 MAC aa.bb.cc.dd.ee.22	IMP ROC 11 192.168.123.001 MAC aa.bb.cc.dd.ee.23	IMP ROC 14 192.168.132.001 MAC aa.bb.cc.dd.ee.32	IMP ROC 15 192.168.133.001 MAC aa.bb.cc.dd.ee.33

View From Back Of Detector

Packet Example Write Register

32000 write port -- Used to write registers

Write register PACKET FORMAT = to port 32000

Single register write
SYSTEMKEY(31..0) ,ADDRESS(16..0),data(31..0), 0xFFFF

Multi Register write
SYSTEMKEY(31..0) ,ADDRESS(16..0),data(31..0),
ADDRESS(16..0),data(31..0),……., 0xFFFF

Example: write 0x1234 to register 3 UDP PACKET

```
0000   aa bb cc dd 01 01  00 26 18 71 dd e9 08 00 45 00
0010   00 28 09 07 00 00 80 11 e1 64 0a 14 1e 32 0a 14
0020   1e 00 7d 00 7d 00 00 14 50 7f de ad be ef 00 03
0030   00 00 12 34 ff ff
```

Packet Example Read Register

32001 read request port -- Used to read registers
32002 response port -- Used by system in respond to a read request

BLK_READ = 0 read 1 register
BLK_READ = 1-31 (read up to 32 sequential registers)

READ register PACKET FORMAT = to port 32001

SYSTEMKEY(31..0) ,ADDRESS(16..0), BLK_READ(16..0),0xFFFF

Example: read register 3 UDP PACKET

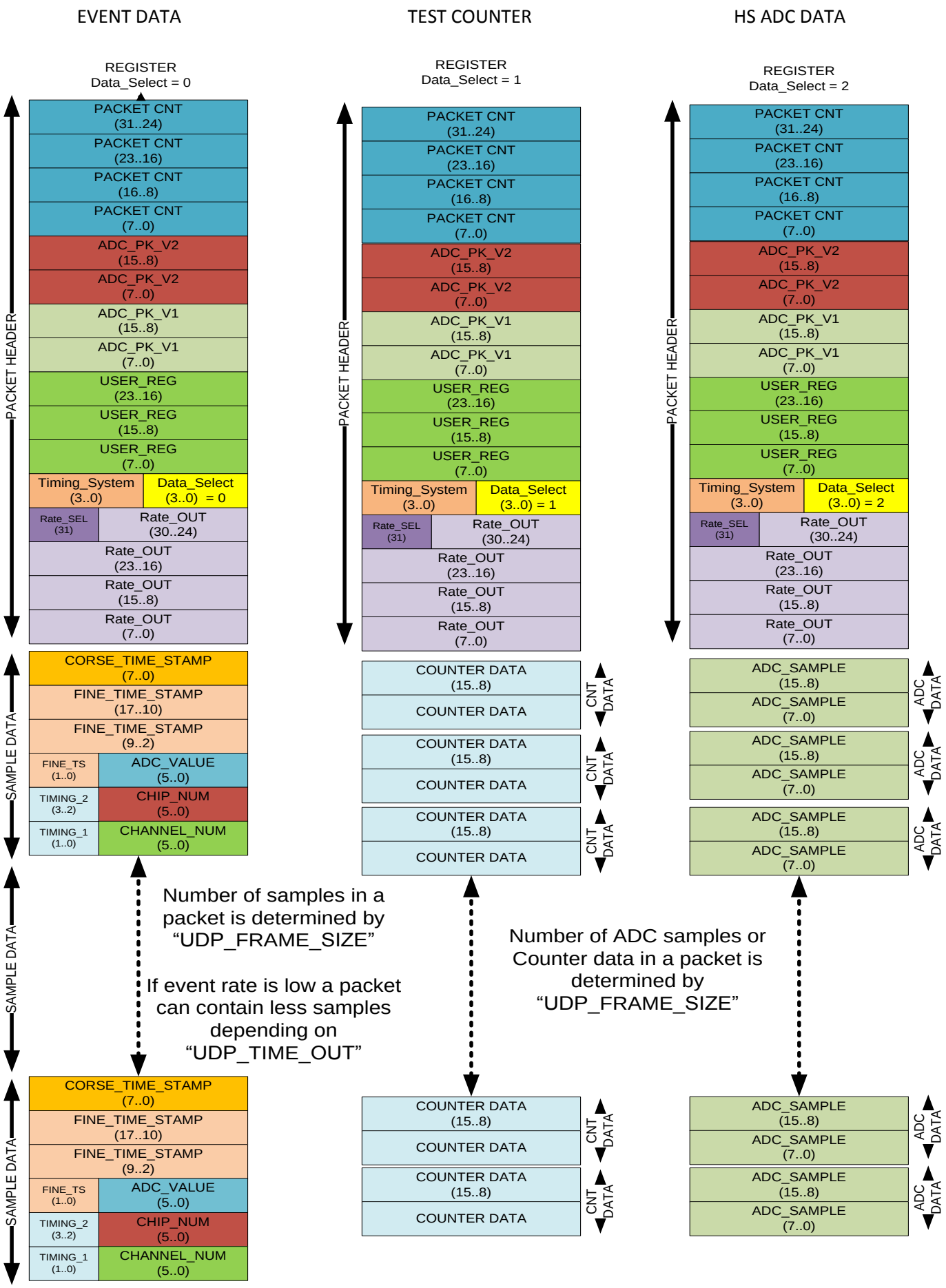
Write to port 3001

```
0000   aa bb cc dd 01 01   00 26 18 71 dd e9 08 00 45 00
0010   00 28 09 0d 00 00 80 11 e1 5e 0a 14 1e 32 0a 14
0020   1e 00 7d 00 7d 01 00 14 50 7f de ad be ef 00 03
0030   00 00 00 00 ff ff
```

response on port 32002

```
0000   00 26 18 71 dd e9 aa bb cc 01 01 00 08 00 45 00
0010   00 2e 3d aa 00 00 80 11 ac bb 0a 14 1e 00 0a 14
0020   1e 32 7d 00 7d 02 00 1a 00 00 00 03 00 00 00 00
0030   00 00 00 00 00 00 00 00 00 00 00 00
```

UDP PACKET FORMAT



Packet header description

PACKET_CNT(31..0) = UDP packet counter -- increments on every packet sent over high speed link port (32003)

ADC_PK_V2(15..0) = MCA ADC data peak # 2

ADC_PK_V1(15..0) = MCA ADC data peak # 1

USER_REG(23..0) = User data appended to data stream -- data set from register 38

Timing_system(3..0) = 4 bits received from pizza box appended to UDP header

Data_select(3..0) = Set by register 27 "HS_DATA_SELECT" to select one of the three data streams

Rate_sel(0) = Rate meter window 0 = 100mS 1 = 1s -- set by register 8 bit 6

Rate_OUT = Number of samples in rate meter window

Register start sequence

set registers 2,3,4 = chip disable

Set Register 5 DSYNC_RATE = 0x10

TIME_STAMP CLOCK PERIOD = 0x00 100ns

set register 7 = set bit 0 to 1

set register 8 = sector disable, ASIC FIX Enable , testmode , rate meter window

set register 9 = ADC trigger