W-Si Digital Logic

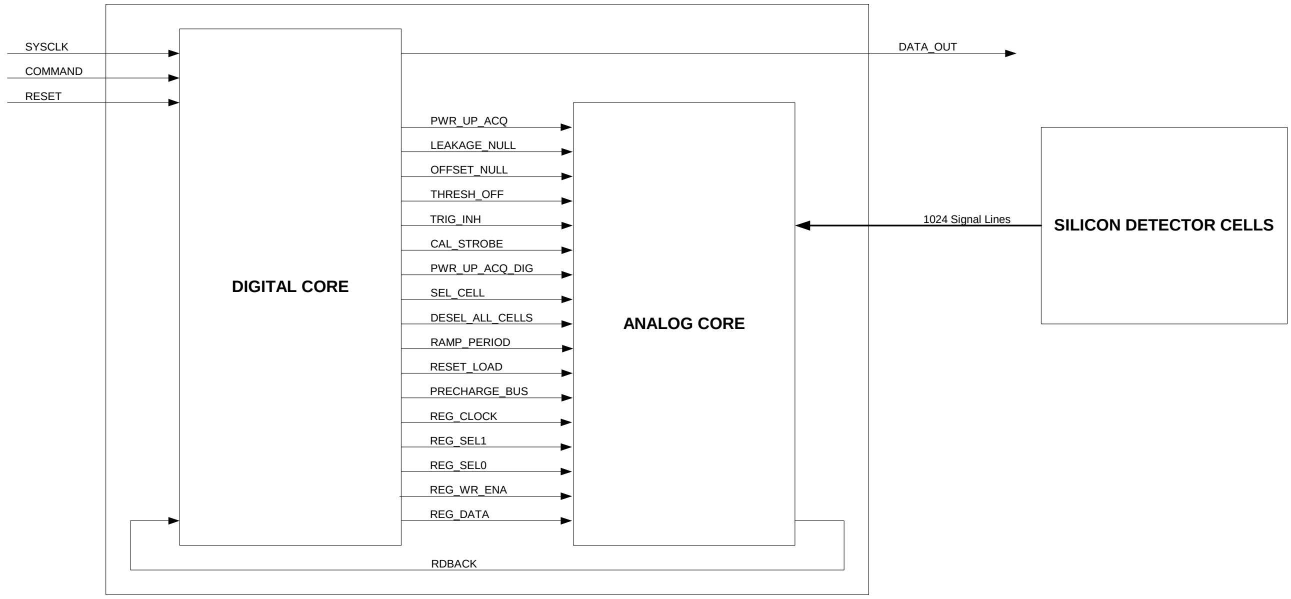
Version 0.07, 11/17/2005

1. PURPOSE

The purpose of this document is to briefly describe the design of the digital logic portion of the W-Si ASIC. This document should be read by those attempting to understand the VHDL source code of the digital core and those attempting to interface digitally to the ASIC.

2. OVERVIEW

The W_SI ASIC exists in order to amplify and store signals generated by the external silicon detector wafer. Once amplified a timestamp is added to these signals, they are stored temporarily and then transmitted upward to data processing logic. The W_SI ASIC has the ability to sample four distinct events in its analog memory, each of which is then digitized into a 13-bit data word for processing by the upstream logic.



W_SI ASIC Block Diagram

The ASIC is divided into two distinct sections, an analog array and a digital control core. The majority of the ASIC footprint is consumed by the analog array which consists of 1024 channels arranged in a grid of 32x32. Each of these channels consists of an amplifier block, 4 analog storage locations, a digitizer block and memory to store the 4 digitized values and their associated timestamps.

The digital core exists to provide an interface between the 1024 analog cells and the upstream processing logic. The following functions are provided by this block:

- Process commands and register updates received from upstream logic.
- Transmit register read data back to upstream logic.
- Generate clock & control signals needed by analog logic during data acquisition.
- Generate calibration control signals as required.
- Readout the sample data from each analog cell and transmit data to upstream logic.

The ASIC interfaces to the upstream logic through the following signals:

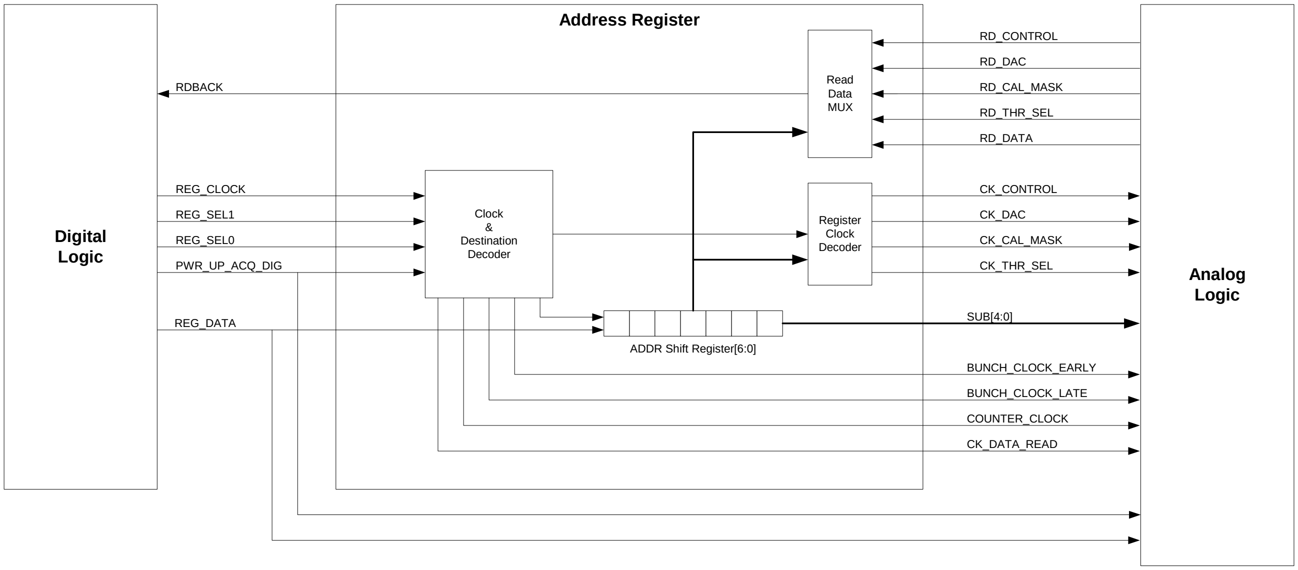
- SYSCLK: The master clock with a period of 42.125nS.
- RESET: A master reset which initializes then device when asserted high.
- COMMAND: A serial command link for command & register update frames.
- DATA_OUT: A serial response/data link for response & sample data.

3. ANALOG BLOCK

Although a detailed description of the analog block is beyond the scope of this document, some understanding of the structure of the analog portion of the design is required to understand the configuration and data readout functions of the digital core. First the clock de-multiplexing logic contained within the analog block must be understood. Secondly there must be an understanding of how the digital core interfaces to the various configuration registers that exist within the analog portion of the ASIC. Lastly the structure of the readout shift register and the various memory locations which it shifts data from must be understood to make sense of the readout controller within the digital core.

3.1 Address Register

The interface between the analog and digital portions of the ASIC was designed in order to minimize the number of signals that must be passed between these two blocks. These design decisions include the local decoding of command header data within the analog core as well the multiplexing of the numerous clock lines required within the analog block of the design. Both the command decoding and clock de-multiplexing functions are contained within the address register block in the analog portion of the ASIC. The following diagram shows the structure of the address register block.



Address Register

The following table describes the operation of the address register block.

REG_SEL[1:0]	PWR_UP_ACQ	ADDR[6:4]	Operation
00	X	XXX	REG_DATA is shifted into 6-bit header register with REG_CLOCK.
01	X	00X	Nothing is done.
01	X	010	REG_CLOCK is passed to CK_DAC output to write/read to/from one of nine DACs. SUB[3:0] is used externally to decode DAC selection. RD_DAC data is passed to RDBACK.
01	X	011	REG_CLOCK is passed to CK_CONTROL output to write/read to/from control register. RD_CONTROL is passed to RDBACK.
01	X	10X	REG_CLOCK is passed to CK_CAL_MASK output to access calibration mask registers. SUB[4:0] is used externally to decode row selection. RD_CAL_MASK is passed to RDBACK.
01	X	11X	REG_CLOCK is passed to CK_THR_SEL output to access threshold select registers. SUB[4:0] is used externally to decode row selection. RD_THR_SEL is passed to RDBACK.
10	1	XXX	BUNCH_CLOCK_EARLY & COUNTER_CLOCK are asserted on every even assertion of REG_CLOCK. BUNCH_CLOCK_LATE is asserted on every odd assertion of REG_CLOCK.
10	0	XXX	COUNTER_CLOCK is driven by REG_CLOCK.
11	X	XXX	CK_DATA_READ is driven by REG_CLOCK to access data shift register. RD_DATA is passed to RDBACK.

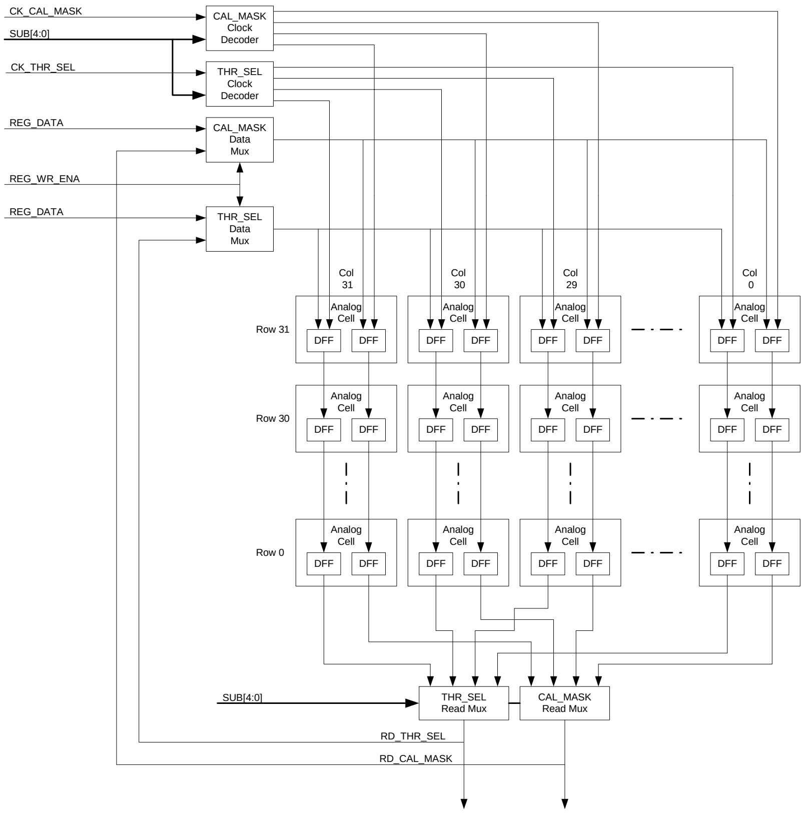
The data shifted into the ADDR shift register in the address block are the 6-bits which make up the register address in the command header. The structure of the command/response frames and their headers are described later in this document.

3.2 Cell Configuration Registers

As mentioned earlier the analog portion of the ASIC consists of a number of cells which are arranged into 32 rows each containing 32 columns. Each analog cell contains two configuration bits, one to enable/disable the calibration logic for the given cell and the second to select which of the two available thresholds should be used for the cell. The calibration mask bit for each of the 32 rows in a column is arranged into a single shift register to form a 32-bit calibration mask register. Each of the 32 columns in the design will have its own 32-bit calibration mask register. Similarly each of the 32 columns has a 32-bit threshold select register with a single bit for each of the row cells within the row.

In order to operate properly reads from these registers must be non-destructive. This allows the control software to read from each of the registers without the register loosing its current contents. A data selection mux at the input to the register selects between the read mode where data is recycled back into the register during shifting and the write mode where new data is input into the shift register with each clock assertion.

See the following diagram for a visual description of the register structure.



Analog Block Register Structure

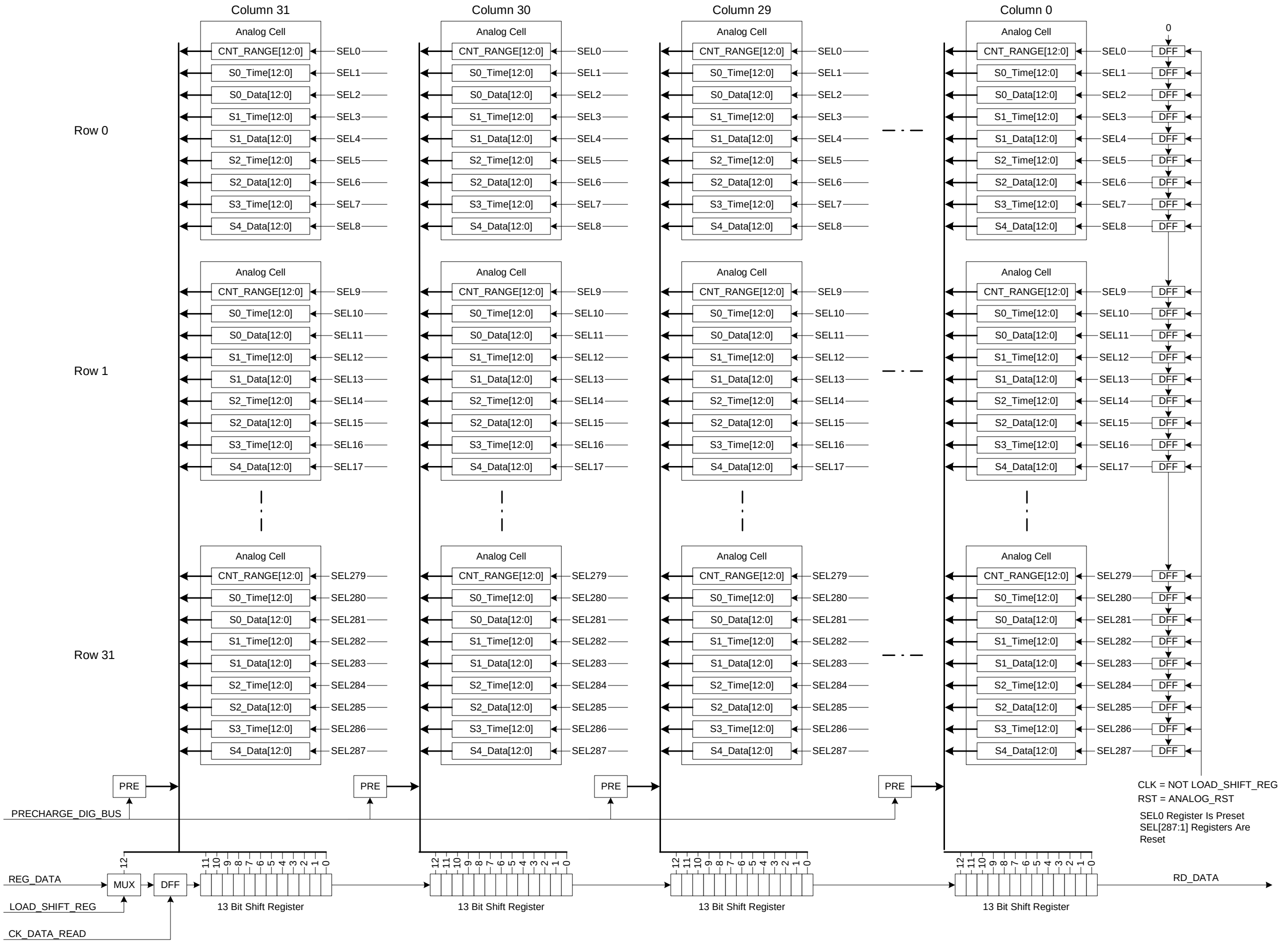
3.3 Sample Data Readout

The structure used to read out the sample values and their associated timestamps differs somewhat from the configuration register. Each cell contains 9 values that must be read out following a data acquisition cycle. These nine values are described below.

- CNT_RANGE: Range & Counter. This word contains a 4-bit event counter and a range bit for each of the four events.
- S0_TIME: Sample 0 timestamp.
- S0_DATA: Sample 0 data.
- S1_TIME: Sample 0 timestamp.
- S1_DATA: Sample 0 data.

- S2_TIME: Sample 0 timestamp.
- S2_DATA: Sample 0 data.
- S3_TIME: Sample 0 timestamp.
- S3_DATA: Sample 0 data.

The following diagram shows the structure of the readout logic associated with the 1024 analog cells contained in the ASIC.



Readout Data Structure

All of the 9 words in each of the 32-cells in a single column are connected together with a digital readout bus. This bus is then connected to a 13 bit shift register at the bottom of the column. At any given time one of the 9 words in one of the 32 row cells will be enabled allowing its data value to be output to the digital bus. This value will then be loaded from the digital bus into the readout shift register by asserting the LOAD_SHIFT_REG signal while asserting the CD_DATA_LOAD signal for a single clock. The loading of the shift register at the bottom of all of the 32 columns is performed in parallel. Before the next value can be read out onto the digital the PRECHARGE_DIG_BUS signal is asserted forcing all of the 13-data bits on each of the 32 digital busses to be set to a value of 0.

The current row/word which is connected to the digital bus is determined by the state of a 288 bit selection shift register with a single selection bit for each of the 9 words in each of the 32 row cells. All of the bits in this shift register are initialized before readout by the ANALOG_RST. Bit 0 of this shift register is initialized to a value of 1 while the remaining bits are initialized to a value of 0. Each time the LOAD_SHIFT_REG signal transitions from 1 to 0 the selection shift register is shifted one bit, selecting the next word/row in the sequence. A single selection shift register exists with its outputs connected to each of the 32 columns in the array.

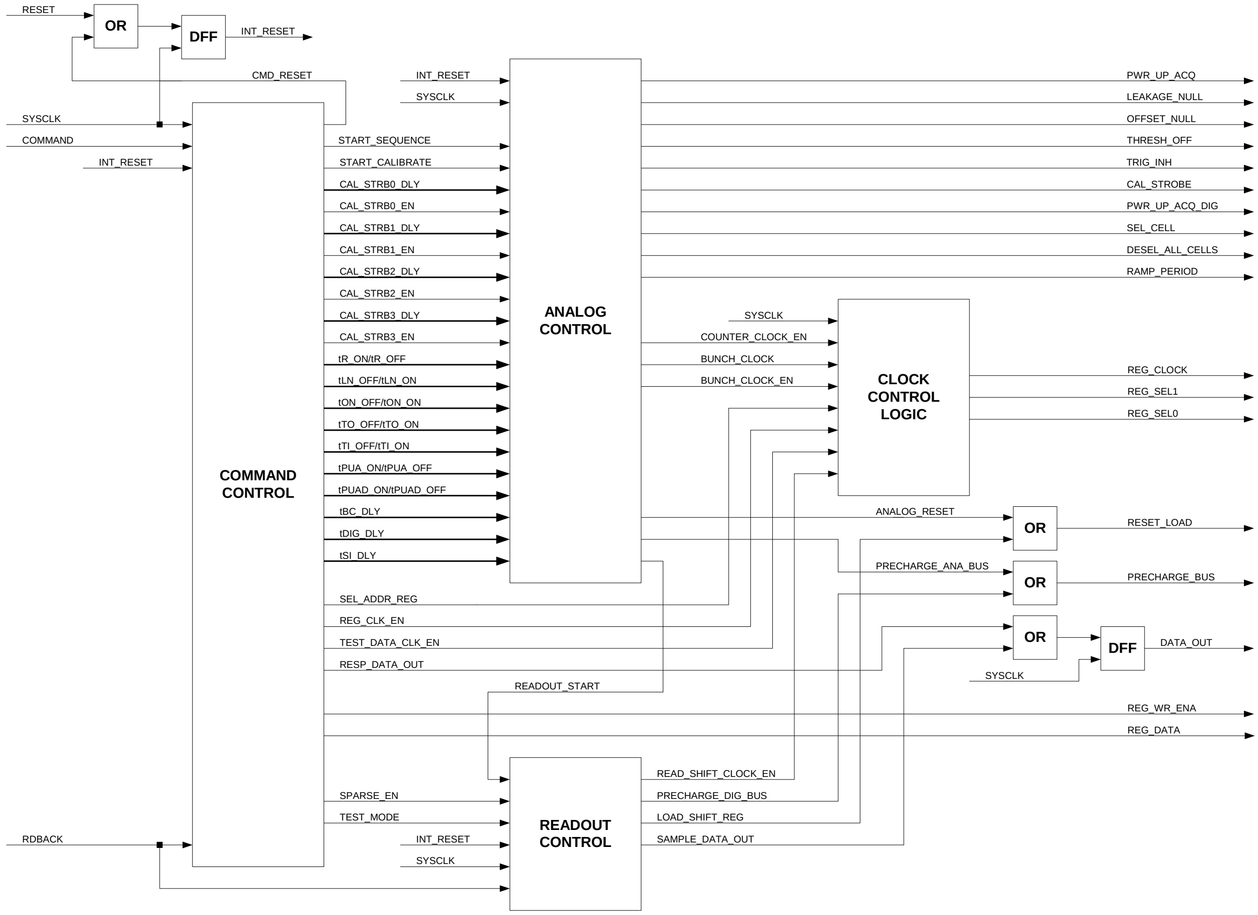
For each word that is read from the array the digital logic will pre-charge the digital bus, load the shift register and then shift out the 416 data bits previously loaded into the data shift register. The digital logic must give the digital bus time to settle for 10uS following the selection of a new value. This sequence will then repeat a total of 288 times in order to read out sample data from the entire array.

The digital logic also has the ability to shift a know data value into the readout shift register. This is provided to allow diagnostics to be performed on the ASIC to verify its proper operation. Following the shifting of test data into the readout shift register the digital logic can choose to not load the digital bus data into this data shift register during the readout process. The digital logic can also connect the RD_DATA signal to the REG_DATA signal allow the test data (or possible its inverse) to be re-cycled back into the shift register. This allows the same 416 data bit value to be repeated during each of the 288 readout operations.

4. DIGITAL BLOCK

The digital block of the ASIC provides an interface between the analog cells and the external data collection and processing logic. The digital block can receive and response to commands and register accesses provides a number of control & timing signals to the analog block and performs readout of the sample data following each acquisition cycle.

The digital block is broken up into 4 major sections as shown in the following diagram. These blocks include the command control block, the analog control block, the readout control block and the clock control block.



Digital Block

It is important to note a couple of details in the above diagram before moving onto the description of each individual sub-block. First the incoming reset is combined with a CMD_RESET signals before being registered and passed onto internal logic. The CMD_RESET signal is generated when a

RESET command is received by the command controller.

The ANALOG_RESET & LOAD_SHIFT_REG signals are combined together to create the multiplexed RESET_LOAD signal. This signal is decoded by the analog logic as ANALOG_RESET during the acquisition & digitization stages using the PWR_UP_ACQ_DIG signal to determine the mode.

The PRECHARGE_ANA_BUS & PRE_CHARGE_DIG_BUS signals are combined together to create the multiplexed PRECHARGE_BUS signal. Again PWR_UP_ACQ_DIG signal is used by the analog block to determine the mode of this signal.

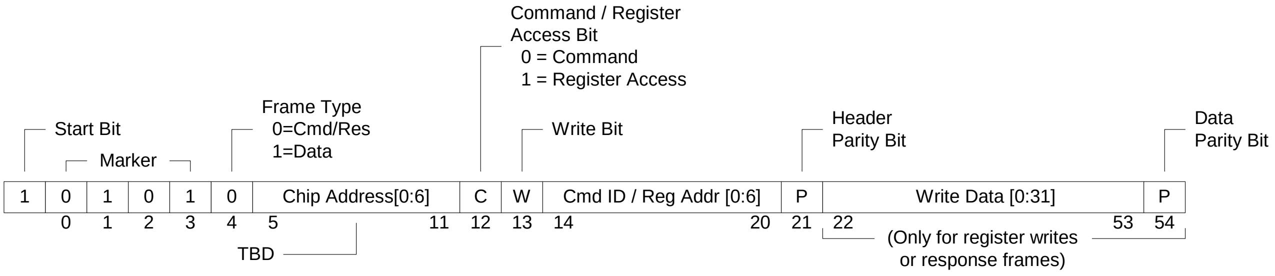
The top level source code which contains the sub-blocks, the multiplexing logic and the clock control logic can be found at the following location:

MEMORY_ARRAY_CONTROL.VHD

4.1 Command Control Block

The command control block serves as the communication interface between the external logic and the various portions of the digital block of logic. The primary function of this block is to receive command frames over the serial command link as well as send response frames back to the external logic when appropriate. In addition to receiving and transmitting frames it also has a number of shift registers for storing configuration bits and timing control settings.

The following diagram shows the structure of the command frames which the command control block receives as well as the response frames it has the ability to transmit back to the control logic.



Command / Response Frame

The command / response frame is designed so that the format of the command frame received is consistent with the response frame that is sent out. Also the response frame must have a similar format to the data frame which is sent out by the readout control logic over the same link. The structure of the first 13-bits (not including the start bit) of the response frame is identical to the data frame sent by the readout control logic. Bit 4 of the header is used to distinguish between a command/response frame and a data frame.

The start of the command frame is detected by a single start bit inserted before the first bit of the header data. The command link is normally contains a value of 0 when the link is idle.

Bits 5-11 of the command/response serve as a chip address. The LSB of this address is the first bit received or transmitted in the serial sequence. The function of this address field has yet to be determined. For now the chip is hardwired to believe its address is 0x00 and will respond to frames with the address field set to either 0x7F or 0x00. All response & data frames received from the ASIC will have this field set to 0x00.

Bit 12 (C-bit) of the command/response frame is used to distinguish between a command frame and a write / read frame. When this bit is set to 0 the command control logic will treat the received frame as a command and decode the intended command. When this bit is set to 1 the command control logic will treat the received frame as a register read or write. This bit will always bet set in the response frame since only registers reads are responded to.

Bit 13 (W-bit) of the command/response frame is used to distinguish between a register read or write. When set the command control logic treats the received frame as a register write, otherwise a register read is assumed. This bit will always bet clear in the response frame since only registers reads are responded to.

Bits 14-20 of the frame contain the register address / command id field. For command frames with the C bit clear this field identifies the command to be performed. When the C bit is set this field contains the address of the register that is being accessed. The response frame always contains a copy of the received address / command value received.

Bit 21 of the frame is the header parity bit. This bit contains the even parity calculation of all of the bits in the header (no including start bit). If the command control logic detects a mismatch between the received parity bit and its local calculation of header parity the command will be ignored and a header parity error bit in the local status register will be set.

Bits 22-53 contain a 32-bit data word that is used only for register writes in the received frame. This field contains the value that will be shifted into the addressed register during the write operation. This field contains the data read from the addressed register in the transmitted response frame.

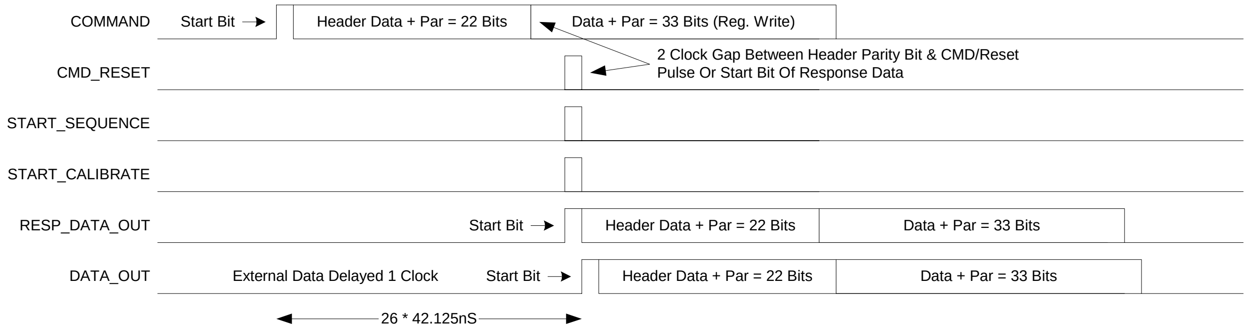
Bit 54 of the frame is the header parity bit that is used only for register writes in the received frame. This bit contains the even parity calculation of bits 22-53 in the received frame or the transmitted response frame. If the command control logic detects a mismatch between the received parity value and its local calculation a data parity error bit in the local status register will be set. The new data value will still be shifted into the addressed register.

Upon receiving the incoming frame the command control logic will shift the register address portion of the header into he address register contained in the analog portion of the design.

If the received frame is determined to be a command with the C bit set to 0, the command control logic will decode the command and set the appropriate control bit. If a “Reset” command is received the CMD_RESET signal will be asserted for a single clock forcing the internal reset line within the ASIC to be asserted. If a “Start Acquisition” command is detected the START_SEQUENCE signal to the analog control block will be asserted for a single clock. If a “Start Calibration” command is detected both the START_SEQUENCE & START_CALIBRATION signals to the analog control block will be asserted for a single clock.

For register read and write frames the command control logic will decode the address to determine if the register is local to the command control logic or contained in the analog block. If the register is local the address register will be updated. For accesses to the analog block of logic the proper control signals will be asserted and the analog block will use the value shifted into the address register to decode the appropriate register.

The following diagram shows the timing of the signals generated in response to either a command or register access frame. This timing is important to understand in order to properly line up the bunch clocks generated by the analog control logic.


COMMAND
Start Bit
Header Data + Par = 22 Bits
Data + Par = 33 Bits (Reg. Write)
2 Clock Gap Between Header Parity Bit & CMD/Reset
Pulse Or Start Bit Of Response Data
CMD_RESET
START_SEQUENCE
START_CALIBRATE
RESP_DATA_OUT
Start Bit
Header Data + Par = 22 Bits
Data + Par = 33 Bits
DATA_OUT
External Data Delayed 1 Clock
Start Bit
Header Data + Par = 22 Bits
Data + Par = 33 Bits
26 * 42.125nS

Command / Response Timing

The following table defines the address map of the ASIC.

Command	Cmd/Reg	Write Bit	Address	Data	
NoOp	0	1	0x00	NA	NA
Reset	0	1	0x01	NA	NA
Start-Acquisition	0	1	0x02	NA	NA
Start-Calibration	0	1	0x03	NA	NA
Unused	0	0	0x04 - 0x7F	NA	NA
Status Register	1	0	0x00	0 1 31 - 2	Command Parity Error Data Parity Error Unused, Read as 0
Configuration Register	1	1/0	0x01	0 1 31 - 2	TEST_DATA SPARSE_EN Unused, Read/Write Allowed For Testing
Unused	1	0	0x02 - 0x07	31 - 0	Unused, Read as 0
Timing Control Register 0	1	1/0	0x08	15 - 0 31 - 16	RESET On tR_ON[15:0] RESET Off tR_OFF[15:0]
Timing Control Register 1	1	1/0	0x09	15 - 0 31 - 16	LEAKAGE_NULL Off tLN_OFF15:0] LEAKAGE_NULL On tLN_ON[15:0]
Timing Control Register 2	1	1/0	0x0A	15 - 0 31 - 16	OFFSET_NULL Off tON_OFF[15:0] OFFSET_NULL On tON_ON[15:0]
Timing Control Register 3	1	1/0	0x0B	15 - 0 31 - 16	THRESH_OFF Off tTO_OFF[15:0] THRESH_OFF On tTO_ON[15:0]
Timing Control Register 4	1	1/0	0x0C	15 - 0 31 - 16	TRIG_INH Off tTI_OFF[15:0] TRIG_INH On tTI_ON[15:0]
Timing Control Register 5	1	1/0	0x0D	15 - 0 31 - 16	PWR_UP_ACQ On tPUA_ON[15:0] PWR_UP_ACQ Off tPUA_OFF[15:0]
Timing Control Register 6	1	1/0	0x0E	15 - 0 31 - 16	PWR_UP_ACQ_DIG On tPUAD_ON[15:0] PWR_UP_ACQ_DIG Off tPUAD_OFF[15:0]
Timing Control Register 7	1	1/0	0x0F	7 - 0 23 - 8 31 - 24	Start Init Delay tSI_DLY[7:0] Bunch Clock Delay tBC_DLY[15:0] Digitization Delay tDIG_DLY[7:0]
Cal Delay Register 0	1	1/0	0x10	11 - 0 12 15 - 13 27 - 16 28 31 - 29	Cal Strobe 0 Delay [11:0] Cal Strobe 0 Enable Unused Cal Strobe 1 Delay [11:0] Cal Strobe 1 Enable Unused
Cal Delay Register 1	1	1/0	0x11	11 - 0 12 15 - 13 27 - 16 28 31 - 29	Cal Strobe 2 Delay [11:0] Cal Strobe 2 Enable Unused Cal Strobe 3 Delay [11:0] Cal Strobe 3 Enable Unused
Unused	1	0	0x12 - 0x17	31 - 0	Unused, Read as 0
Test Data Load Register	1	1/0	0x18	31 - 0	Test data to load into shift registers. 13 writes to this register are required to fill 416-bit shift register with test data.
Unused	1	0	0x19 - 0x1F	NA	Unused, Read as 0
Event 0 Threshold Low DAC	1	1/0	0x20	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Event 1 Threshold Low DAC	1	1/0	0x21	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Ramp Threshold DAC	1	1/0	0x22	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Range Threshold DAC	1	1/0	0x23	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Calibration DAC	1	1/0	0x24	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Event Threshold Reference DAC	1	1/0	0x25	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Shaper BIAS DAC	1	1/0	0x26	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Default Analog DAC	1	1/0	0x27	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Event 0 Threshold High DAC	1	1/0	0x28	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Event 1 Threshold High DAC	1	1/0	0x29	31 - 0	DAC Setting, 8-Bit Register Repeat 4X
Unused	1	0	0x2A - 0x2F	NA	Unused, Read as 0
Control Register 8-bit register, repeat 4X	1	1/0	0x30	2 - 0 3 4 5 6 7	HOLD_TIME[2:0] CAL_RANGE EXTERNAL_DAC FORCE_LOW_GAIN LEAKAGE_NULL Disable Unused
Unused	1	0	0x31 - 0x3F	NA	Unused, Read as 0
Calibration Mask Registers (1 Per Col)	1	1/0	0x40 - 0x5F	31 - 0	1 bit per Row bit 0 = Row 0
Range Select Registers (1 Per Col)	1	1/0	0x60 - 0x7F	31 - 0	1 bit per Row bit 0 = Row 0

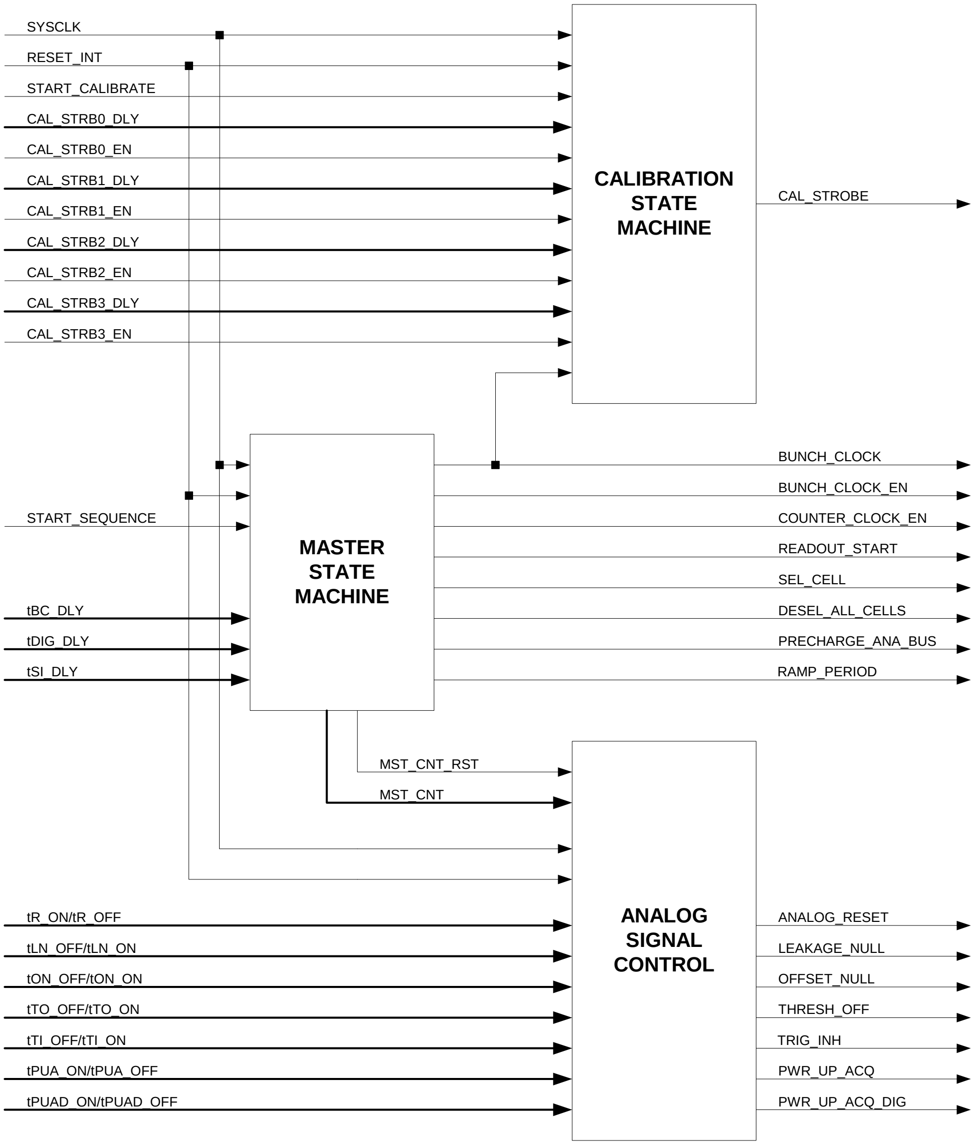
ASIC Address Map

The source code for the command control block can be found at the following location:

COMMAND_CONTROL.VHD & REG_RW_32.VHD

4.2 Analog Control Block

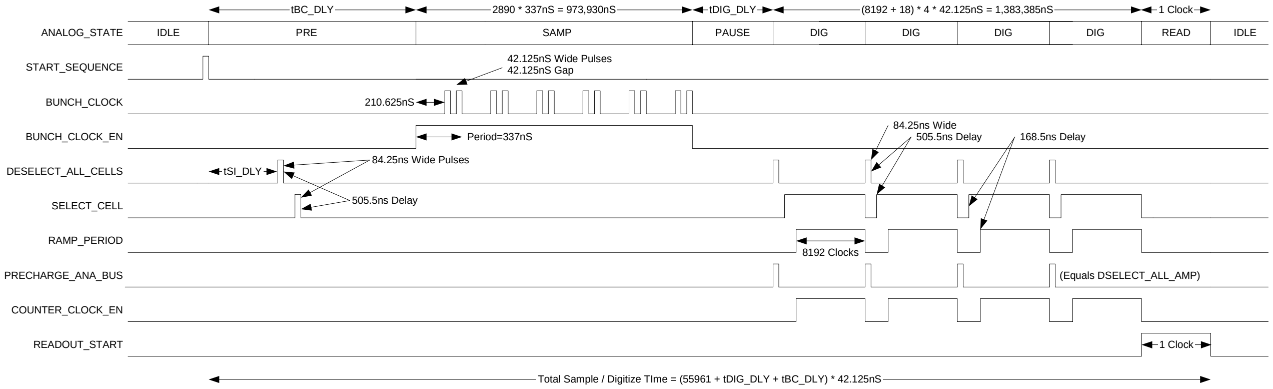
The analog control block provides a number of control signals to the analog block in the ASIC. There are three sub-blocks which exist in the analog control block. The primary block is the master state machine which controls the overall operation of the analog control block and keeps a number of counters which are used by the other two blocks.



Analog Control Block

4.2.1 Master State Machine

The master state machine is initiated when the START_SEQUENCE signal is asserted by the command control block. The operation of the master state machine is shown in the following diagram.



Sample & Digitize Timing

Two counters exist in the analog control block. Both of these counters are held in reset while the state machine is in the IDLE state. The master counter is enabled for all other states and continues to count through all state transitions. The sub counter is reset to 0 as the state machine transitions between states.

Upon detecting the assertion of the START_SEQUENCE control signal the state machine enters the PRE state for a period of time defined by the tBC_DLY timing control value. The tBC_DLY register defines the sub counter value at which the master state machine should transition to the SAMP state. While in this state the state machine waits a period of time before initializing the analog cells by asserting the DESEL_ALL_CELLS followed by the assertion of SEL_CELL. The sub counter value at which the DESEL_ALL_CELLS signals should be asserted is defined by the tSI_DLY value. The DESEL_ALL_CELLS value is registered so the assertion will take affect on the next clock edge.

In the SAMP state the state machine asserts a number of bunch clocks to the analog logic. These clocks are aligned with the bunch collisions in the detector with two pulses being asserted for each bunch received. The period of the bunch clock is 337nS with the first bunch clock period aligned with the beginning of the SAMP state. For each 337nS period the bunch clock signal repeats the following sequence:

1. Low for 210.625nS
2. High for 42.125nS (Used as BUNCH_CLOCK_EARLY by the analog block)
3. Low for 42.125nS

4. High For 42.125nS(Used as BUNCH_CLOCK_LATE by the analog block)

This sequence is repeated 2890 times. In the detector 2880 bunches are expected. 10 extra bunch clocks are generated by the analog control state machine in order to provide some flexibility in the alignment of the generated clocks with the actual bunches.

Once 2890 bunch clock periods pass the master state machine will transition to the PAUSE state. In this state no control signals are generated by the master state machine. The sub counter value at which the state machine transitions to the DIG state is defined the by the tDIG_DLY register value.

Following the PAUSE state the master state machine will go to the DIG state 4 times, once for each of the four stored samples. The master state machine will stay in the DIG state for 4106 clocks with the following sequence:

1. DESELECT_ALL_CELLS & PRECHARGE_ANA_BUS asserted for 2 clocks.
2. All signals low for 12 clocks.
3. SELECT_CELL asserted for 4 clocks.
4. SELECT_CELL, RAMP_PERIOD & COUNTER_CLOCK_EN asserted for 8192 clocks.

After this sequence has repeated 4 times the master state machine will enter the READ state. The master state machine will stay in the READ state for a single clock with the READOUT_START signal asserted. After that the master state machine returns to the IDLE state, all counters are reset and all control signals go to their idle state.

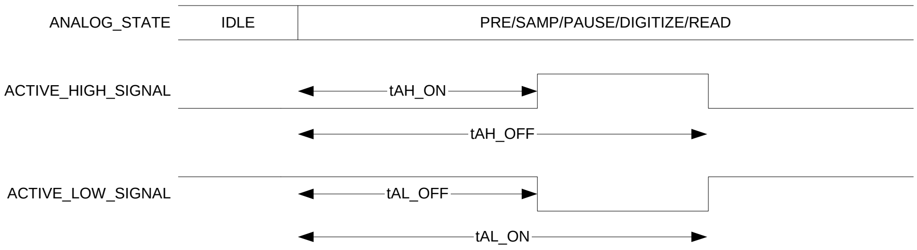
As mentioned in the previous text there are three register values that affect the operation of the master state machine. The default setting of each of these values is included below:

- tSI_DLY = 0x47 = 72 clocks = 3,033nS
- tBC_DLY = 0x017B = 380 clocks = 16,007.5nS
- tDIG_DLY = 0xED = 238 clocks = 10,025.75nS

4.2.2 Analog Signal Control

While the master state machine is out of the IDLE state and the master counter is enabled the analog control signal logic will assert & de-assert a number of control signals. The assertion and de-assertion time of each of these signals are independently configurable by setting a timing register. The set of control signals consist of a mix of active high and active low signals. For active high signals the signal is turned on when the master counter reaches the tON value associated with that signal and is turned off when the master counter reaches the tOFF value. The timing of active low signals is similar as shown in the following diagram. Each output signal is registered so its assertion / de-assertion takes affect on the clock edge at which the master counter value equals the ON/OFF register value.

The following diagram shows how the timing settings affect the two types of signals.



Control Signal Timing

The following table defines each of the signals controlled in the analog signal control block as well as their default values.

Signal	Type	Register	Default	Clocks	Delay
ANALOG_RESET	Active High	tR_ON tR_OFF	0x0007 0x00ED	8 238	337nS 10,0025.75nS
LEAKAGE_NULL	Active Low	tLN_OFF tLN_ON	0x0002 0x5C42	3 23,619	126.375nS 994,950.375nS
OFFSET_NULL	Active Low	tON_OFF tON_ON	0x0164 0x5C42	357 23,619	15,038.625nS 994,950.375nS
THRESHOLD_OFF	Active Low	tTO_OFF tTO_ON	0x016B 0x5C42	364 23,619	15,333. 5nS 994,950.375nS
TRIGGER_INH	Active Low	tTI_OFF tTI_ON	0x0183 0x5C42	388 23,619	16,344. 5nS 994,950.375nS
PWR_UP_ACQ	Active High	tPUA_ON tPUA _OFF	0x0009 0x5C42	10 23,619	421.25nS 994,950.375nS
PWR_UP_ACQ_DIG	Active High	tPUAD_ON tPUAD_OFF	0x0009 0xDD02	10 56,579	421.25nS 2,383,390.375nS

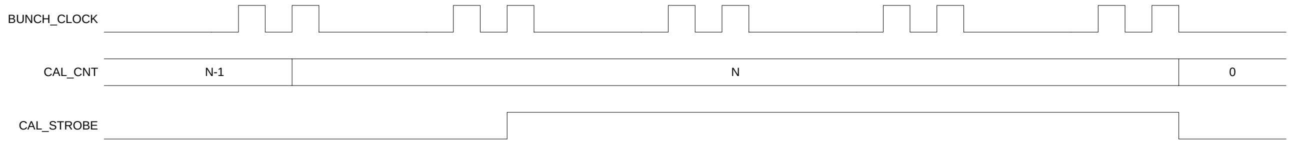
4.2.3 Calibration State Machine

The third sub-block which exists in the analog control block is the calibration state machine. This block of code is responsible for generating up to 4 calibration pulses at configured times during the sampling sequence. The calibration state machine is a slave to the master state machine and is only enabled while the master state machine is in the SAMP state. Also the calibration state machine is not enabled unless the START_CALIBRATE signal is asserted at the same time as the

START_SEQUNCE signal by the command control logic.

Each of the possible 4 calibration pulses has a delay value and enable bit associated with it. The delay setting defines the bunch clock count at which the calibration pulse should be asserted. The enable bit defines if the calibration pulse should be asserted at all.

The following diagram describes the operation and timing of the calibration state machine.



Calibration Timing

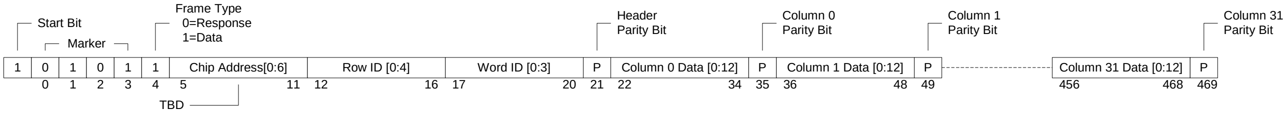
In the previous diagram N is the current setting of the calibration delay register. When the bunch clock count reaches N the calibration pulse is asserted. The pulse remains asserted for 3 bunch clock periods (~1uS) and is de-asserted. At that point the bunch clock counter is reset. This sequence is repeated 4 times, once for each of the calibration pulse settings. The state machine starts with the pulse 0 settings selected and moves on as each count is reached. The enable setting of each pulse only controls whether or not the CAL_STROBE is asserted. The delay setting will still cause the calibration state machine to transition. This means that if the pulse 0 delay is set to 7 and is disabled and the pulse 1 delay is set to 0 and it is enabled the first calibration pulse will occur at bunch clock 11.

The source code for the analog control block can be found at the following location:

ANALOG_CONTROL.VHD

4.3 Readout Control Block

The readout control block is responsible for reading data from each of the 1024 analog cells contained in the ASIC. Within each cell are 9 13-bit values that must be read. As described earlier in this document each of the 9 values is read out a row at a time. Each value read out is a 13 bit value and the shift register has a single location for each of the 32 columns. This results in a 416-bit shift register. As described earlier in the document there is a ~12uS time period that is required to select the next row / word and shift its value into the readout shift register. Due to this delay the readout controller shifts each row/word combination out in separate data frames. A header is attached to the front of the shifted data to identify which row & word is currently contained in the transmitted frame. In order to provide data protection the readout controller will insert a single parity bit after each 13-bit value is shifted out of the readout register. This is achieved by gapping the shift clock for once cycle every 13 bits as data is read from the readout shift register. The format of the data frame sent out by the readout control block is shown in the following diagram.



Data Frame

As noted earlier the format of the data frame is similar the command response frame. Bit 4 of the header is set to ‘1’ to identify the transmitted frame as a data frame. The chip address field will contain the address of the chip sending out data (0x00 for now). The row ID field identifies which of the 32-rows is the source of the data in the frame. The word ID field identifies which of the 9 word value is the source of the data in the frame and is decoded as follows:

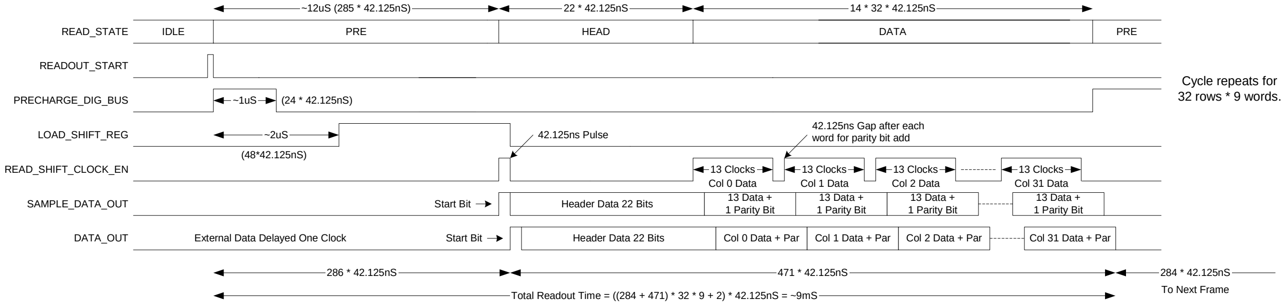
- 0x0 = Count / Range ID Register
- 0x1 = Sample 0 Timestamp
- 0x2 = Sample 0 Data
- 0x3 = Sample 0 Timestamp
- 0x4 = Sample 0 Data
- 0x5 = Sample 0 Timestamp
- 0x6 = Sample 0 Data

- 0x7 = Sample 0 Timestamp
- 0x8 = Sample 0 Data

Following the word ID is the header parity bit. This bit contains the calculated even parity of the header data not including the start bit. Following the header are the 32 13-bit values read from the shift register. An even parity value is calculated for each of the 13-bit values read. The value frame column 0 is read LSB first followed by column 1 and so on until the data from column 31 is read.

Once a row of data is read the next word / row combination is selected with the following readout sequence: Row0_Word0 … Row0_Word1 … Row0_Word2 … Row0_Word3 … Row0_Word4 … Row0_Word5 … Row0_Word6 … Row0_Word7 … Row0_Word8 … Row1_Word0 … Row1_Word1 … all the way up to Row31_Word9.

The timing of the signals generated by the readout control logic and the timing of the transmitted frame is shown in the following diagram.



Readout Timing

Although the timing of the transmitted data frame is predictable it is recommended that the start bit is used to determine the beginning of the data frame.

The source code for the readout control block can be found at the following location:

READOUT_CONTROL.VHD

4.4 Clock Control Block

The clock control block is responsible for multiplexing the numerous local clocks into a single REG_CLOCK signal which is de-multiplexed in the analog logic by the address register logic. The following table describes how the REG_CLOCK, REG_SEL1 & REG_SEL0 signals are generated from the various inputs.

SEL_ADDR_REG	0	1	0	0	0	0	0
REG_CLK_EN	0	X	1	0	0	0	0
BUNCH_CLOCK_EN	0	X	X	1	0	0	0
COUNTER_CLOCK_EN	0	X	X	X	1	0	0
TEST_DATA_CLK_EN	0	X	X	X	X	1	0
READ_SHIFT_CLK_EN	0	X	X	X	X	X	1
REG_SEL1 Output	0	0	0	1	1	1	1
REG_SEL0 Output	0	0	1	0	0	1	1
REG_CLOCK Output	0	not sysclk	not sysclk	BUNCH_CLOCK	not sysclk	not sysclk	not sysclk

The inversion of sysclk is used to avoid glitches on the REG_CLOCK output of this logic due to all of the control signals being aligned to the positive edge of sysclk. This results in register & readout data being aligned to the negative edge of sysclk during register reads and writes and during the readout process. Logic in the command controller and readout controller exists to deal with this alignment. The counter clock which is used during digitization also becomes the inversion of sysclk.

The source code for the clock control block is contained in the top level source code which can be found at the following location:

MEMORY_ARRAY_CONTROL.VHD