UNIVERSITY OF CALIFORNIA,
IRVINE

A Low Powered Single Board Multichannel Data Acquisition System for the ARIANNA Ultra-High Energy Neutrino Detector

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical and Computer Engineering

by

Anirban Samanta

Thesis Committee:
Professor Stuart Kleinfelder, Chair
Professor Steven Barwick
Professor Ozdal Boyraz

2014

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank my advisor Professor Stuart Kleinfelder, for his guidance, support, encouragement and trust. I would like to thank my committee member Professor Steven Barwick for his patience, trust and support. I am greatly thankful to him and my advisor for providing me invaluable financial support when I needed it the most, without which I would have never been able to complete my studies. I would like to thank my committee member Professor Ozdal Boyraz for his time, advice and knowledge.

I would like to thank other members of my research group: Dr. Corey Reed, Dr. Mahshid Roumi and Chris Persichilli. They made invaluable contributions towards ARIANNA and helped me to understand and complete my work. I would like to thank Tarun Prakash and Edwin Cheim for their work without which I would have never gotten the opportunity to finish my work.

I would like to thank my friend Arpita Neogi, who inspired me to start writing my thesis when I had writer's block and continues to inspire me with her excitement for life every day. I would like to thank my karate teachers Bruce Nguyen and Chad Eagan who taught me to be disciplined and how to find courage during tough times. I would like to thank my friends Bharathram Pattabhiraman, Prasanna P Raghavan, Bharath Natarajan, Prithviraj Vasanth, Pavan M Pattanada and Sandhya P Ravindran for helping me in my times of need. I am thankful to my roommate Achyuth Shivapuja for his understanding and support.

Finally I would like to thank my parents Barindra K Samanta and Swapna Samanta for their selfless love and support and for giving me the opportunity to chase my ambitions.

ABSTRACT OF THE THESIS

A Low Powered Single Board Multichannel Data Acquisition System for the
ARIANNA Ultra-High Energy Neutrino Detector

By

Anirban Samanta

Master of Science in Electrical and Computer Engineering

University of California, Irvine, 2014

Professor Stuart Kleinfelder, Chair

The ARIANNA experiment is an ultra-high energy neutrino detector designed to detect cosmogenic neutrinos. The experiment uses the Ross Ice Shelf to detect RF signals which are generated when the high energy neutrinos interact with the ice. The experiment plans to use a massive deployment of hundreds of antenna array stations to capture neutrino events over a large area. The stations use a data acquisition system to capture and store the data and subsequently transmit the data back to UCI. This thesis describes the design, functioning and results of the newest generation system which was deployed in the field in the 2014 season. The new system utilizes a new data digitization chip developed by the ARIANNA engineering group led by Dr. Stuart Kleinfelder. The new SST chip is a largely simplified and streamlined design compared to the previous generation chip and utilizes advanced circuitry to achieve fully synchronous operation while reducing the power usage by many factors. The new system makes substantial improvements over the previous generation system and introduces a number of new features, while providing a possible design solution for the final design requirements of the ARIANNA array systems.

Chapter 1: Introduction

1.1 The ARIANNA Experiment

The search for neutrinos has led to a number of different research experiments in the past with varying degrees of success, but due to the vast range of known or suspected neutrino energies, search for neutrinos is still a very compelling avenue for scientific study. The ARIANNA (Antarctic Ross Iceshelf Antenna Neutrino Array) experiment, first proposed in 2005 [1] by Dr. Steven W. Barwick, is designed to detect high energy cosmic neutrino events and increase the detection sensitivity of neutrinos with energies in excess of 10^{17} eV by at least one order of magnitude. ARIANNA plans to measure the Greisen, Zatsepin, and Kuzumin (GZK) neutrino flux [3], which is suspected to be responsible for absorption of high energy cosmic neutrinos. The GZK neutrino flux is known to be very small, but ARIANNA has a large enough aperture and collecting power to provide adequate statistics to determine the GZK flux cross-section [2].

ARIANNA is intended to establish the neutrino flux produced due to the interaction of cosmic rays with the Cosmic Microwave Background (CMB) [4]. ARIANNA is capable of surveying the southern hemisphere sky with high sensitivity and high resolution. GZK neutrinos interact with matter on earth at center of momentum energies of 100 TeV and thus ARIANNA can allow for study of and beyond the standard model at a much higher magnitude than that is currently available [2,4]. Other than GZK cosmogenic neutrinos, ARIANNA is also expected to be able to detect neutrinos produced directly by cosmic ray sources. [4].

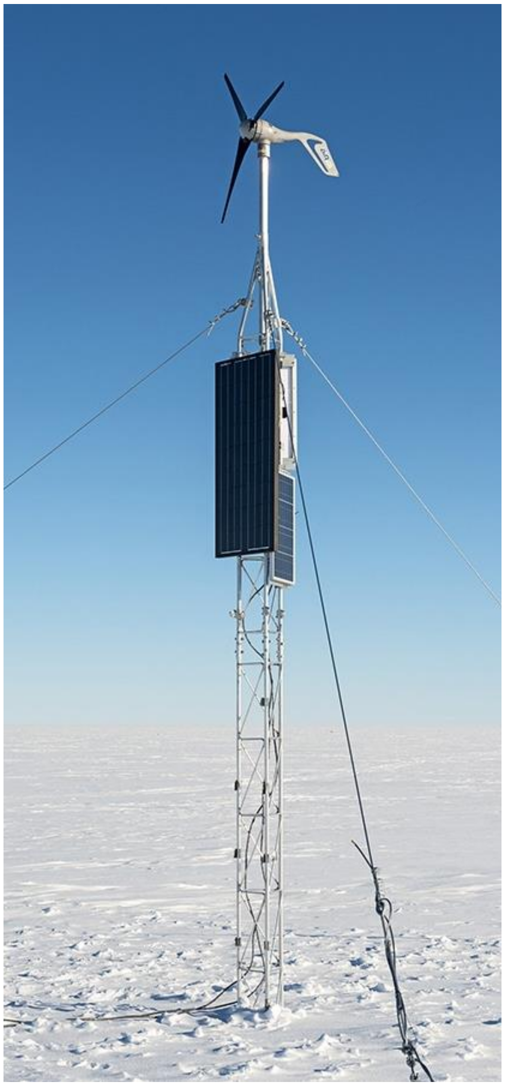


Figure 1-1: An ARIANNA station on the ice. Photo by Dr. Stuart Kleinfelder.

ARIANNA is an array of RF receiver stations at the Ross Ice Shelf in Antarctica. The interaction of neutrinos with the atoms in the ice produce radio signals via the Askaryan Effect [5], which are then captured by downward facing antennas placed in the ice. ARIANNA utilizes a number of remarkable properties of the ice shelf to achieve unprecedented sensitivity. The ice is relatively transparent to electromagnetic waves at the radio frequencies, and with the long attenuation length in the ice the surface antennas can pick up the signals reflected off the ice-water boundary which acts as a good mirror for the signals.

The close proximity of the site to the McMurdo base allows for a large array to be deployed and maintained. The large array with over six months of continuous operation, low energy threshold and a large view of the sky allows for a very high sensitivity to the GZK neutrinos.

1.2 Data Acquisition for ARIANNA HRA stations

The ARIANNA array is composed of Hexagonal Radio Arrays (HRA) which feature a high speed Data Acquisition system (DAQ). Dr. Stuart Kleinfelder and his research group designed the first generation DAQ chip, the Advanced Transient Waveform Digitizer (ATWD) and the overall system hardware. The ATWD was a single channel digitizing IC. The ATWD was part of a daughter card which was then installed in a 4 channel motherboard which then stored and remotely communicated the data to servers back at UCI. The system is shown in fig 1-2.

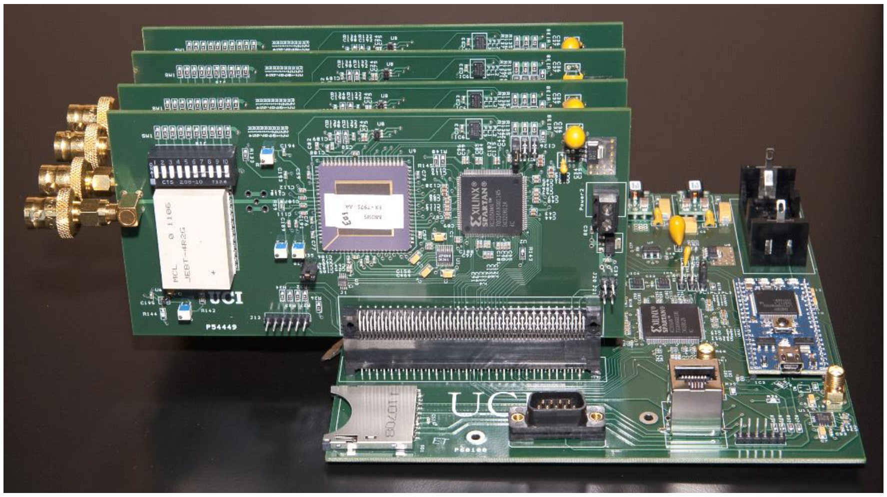


Figure 1-2: The previous generation ATWD based DAQ system. Photo taken by Dr. Stuart Kleinfelder.

The system found considerable success during the second deployment season in 2012, with a number of the systems deployed. For the current season a new Synchronous Sampling plus

Triggering (SST) chip was designed to simplify the ATWD chip and achieve major improvements [6]. The sampling in the new system is completely synchronous without the use of PLL or delays. The chip features stable working capacity with clock rates over 6 orders of magnitude. Design optimizations resulted in a near flat analog bandwidth. A 0.25 μm CMOS process allows large input voltages resulting in considerable reduction in power usage. It also features 256 samples and hold circuits, double that of the ATWD. It supports both single ended as well as differential trigger outputs apart from multiple trigger voltage outputs [7].

A complete new system based on the SST has been designed with the plan of completing the HRA in 2014. A new system board was necessary to be designed to accommodate the SST chip. The new system supports native single board 4-channel operation with reduced power usage and markedly improved system robustness. The remote system measurement capabilities were largely improved with more accurate current and voltage measurements and the inclusion of a digital temperature sensor also improves the ambient condition sensing capabilities. The firmware and the software were also changed significantly, now featuring true independent waveform digitization since the digitization stage is now controlled exclusively by the onboard FPGA.

In the following chapters of this thesis I discuss the development of the new system board. Chapter 2 describes the component selection, hardware design and the PCB design features used, chapter 3 describes the design of the new firmware, chapter 4 discusses the testing procedure and the test results, with chapter 5 being the conclusion for the thesis.

Chapter 2: Hardware Design

2.1 System Overview

The ARIANNA experiment uses the Antarctic ice to detect neutrino events. When the neutrinos interact with the ice it generates a RF signal which then reflects off of the lower surface of the ice shelf. Antennas placed facing downwards in the ice capture the RF signals. The RF signal is then amplified and passed onto the DAQ system. The overall information flow is illustrated in fig. 2-1.

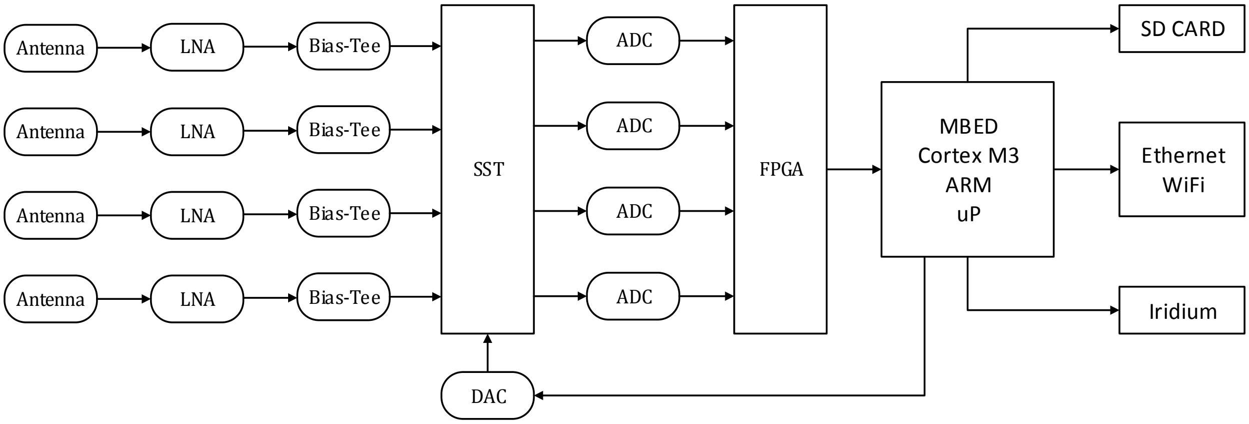


Figure 2-1: Data flow for the four channel system

The multi-antenna system uses four separate channels to determine angle of incidence through phase calculations. Each antenna forms an independent channel which feeds the analog information to the DAQ system. Voltage control through amplifiers and biasing is used in order to bring the incoming signals within the SST (Synchronous Sampling plus Triggering) chip's operational limits. The SST chip stores the incoming analog signals if it detects a neutrino event by comparing the analog signal samples to DAC value thresholds for each channel. The analog signal is then digitized using ADCs on the system board and passed onto an onboard FPGA, which is used to control the acquisition phase of the system and store the incoming digital data. An ARM Cortex-M3 based MBED microcontroller platform oversees the configuration, communications and housekeeping functions. The MBED controls when the data from the FPGA is collected and stored on the onboard SD card and transmitted over AFAR and Iridium networks back to servers at UCI. The MBED can also control when the system is functional by controlling clock generators and voltage supplies.

The system runs at 2 GHz using a 1GHz clock and takes data across 4 channels of 256 samples of 12 bit / sample data. The onboard FPGA runs the data acquisition phase independently, allowing for more freedom for the MBED platform to perform time critical functions. The system features a Secure Digital (SD) card storage for the collected information in parallel to Iridium and Wi-Fi communication systems.

2.2 Motherboard Design

The system board is designed with a number of requirements in mind. The transition to the SST based system means a lot of the previous system board features are made redundant. The new SST chip features 4 channel acquisition in a single chip. Hence the new system board was designed from the ground up to integrate 4 channels into a single system. The system results in a unified system board that incorporates the SST chip along with the data flow components, sensing components, communications components and power control circuitry.

2.2.1 Component Selection

2.2.1.1 Bias-Tee – Mini-Circuits TCBT-2R5G+

Bias-Tee is a three port device that is used to add DC power into a RF signal. It has two signal paths, one to pass the RF signal and one to pass DC power, the output is a combination of DC and RF signal. It provides excellent electrical isolation and voltage biasing of the RF signal needed for powering circuitry.

The bias-tee was selected based on ARIANNA's system requirements. The wideband device has a wide range of 20-2500 MHz. It also features a low typical insertion loss of 0.35 dB and excellent isolation of 44dB (typ.). It also features a much larger maximum current rating, allowing for a much robust system design. Another reason was the extremely small footprint of the device – 0.15" x 0.15" surface mount footprint. A small footprint was necessary due to the requirement of including four devices, one for each channel along with the accompanying circuitry such as bypass capacitors, termination, etc. The most important upgrade from the previous generation system board bias-tee is the reduction in footprint, 0.15" x 0.15" compared to 1.26" x 0.94" for the previous bias-tee.

2.2.1.2 Digital to Analog Convertor – Linear Technologies LTC2657

The Linear Technologies digital to analog convertor LTC2657 used is an I2C based octal 12bit rail-to-rail DAC chip. It features full-scale outputs of 2.5 V from an input of 2.7 V-5.5 V. Dedicated output buffers guarantees monotonic outputs. It features internal voltage reference resulting in lower reference error and hence results in markedly robust outputs.

The internal reference and the I2C protocol were the primary reason for the selection of the specific part. The availability of eight channels in a single chip was also a deciding factor.

The part features a typical zero-scale error of just 1mV. The part features selectable internal or external reference. The internal reference can be switched off by issuing command 0111b through the protocol interface. The device features asynchronous DAC update feature, which forces the outputs to be updated immediately. It also features a power-on-reset feature which allows the device to initialize to zero-scale or to mid-scale at power-up. A software power down feature is also supported by the device through a simple command issue. A particular I2C slave device address can be set by connecting the three configuration pins CA2, CA1 and CA0 to either Vcc, GND or left floating. A global address will also work if only one device is present on the entire I2C bus.

2.2.1.3 Analog to Digital Convertor – Texas Instruments ADS7886

The TI ADS7886 ADC is a 12-bit, 1 MSPS (Million Samples Per Second), capacitor based SAR (Successive Approximation Register) A/D convertor. It incorporates inherent sample and hold circuitry for the input signal. It features a 20Hz Micropower serial interface. It can be supplied through 2.35 V – 5.25 V supply. The low power device also features a power down feature further reducing the dissipated power when the system is used at lower conversion speeds. The device is controlled through two signals SCLK and $\overline{CS}$. On power down and subsequent power up sequence a full cycle of $\overline{CS}$ and SCLK is necessary before beginning conversion.

2.2.1.4 High Speed Oscillator – FXO-LC72

A high speed oscillator is used to generate the 1 GHz clock for the SST chip. The Fox Electronics XpressO series offers excellent low jitter, stable oscillators. The FXO-LC72 is used due to the excellent ambient performance, low noise and low cost. The LC72 also offers a control pin to shut down the chip if necessary to save further power. The clock output is an LVDS output.

2.2.1.5 Temperature Sensor – DS18B20

The DS18B20 is a digital thermometer that provides 9-bit/12-bit temperature measurements. The 1-wire bus data protocol needs just a single data line for communications with a µP and also allows the sensor to be powered from the same data line. With an operating temperature range of -55 °C to + 125 °C. The primary reason for selection is the 1-wire communication protocol.

2.2.1.6 FPGA – Xilinx Spartan 3AN XC3S50AN-4TQG144C

The Xilinx Spartan 3AN family combines FPGA along with non-volatile memory to produce a robust single chip solution for automated systems. Combining FPGA and Flash technology minimizes chip count, PCB traces and overall size while increasing system reliability. The non-volatile feature is also a necessary requirement of the project since it allows the configuration files to be stored onboard and programmed from the memory itself without needing extra circuitry and control. The selected IC is a QFP package IC, which was preferred due to backward compatibility to the previous designs and because any upgrade would lead to redundant resources and higher power usage.

2.2.1.7 Communications – Maxim MAX 13237

The Maxim MAX 13237 is a high speed (3Mbps) RS-232 Transceiver IC that features a single set of transmitter and receiver that is powered from 3.0 V - 5.5 V supply and features a flexible logic low voltage interface with enhanced electrostatic discharge (ESD) protection. The transmitters have a low-dropout transmitter output stage, delivering true RS-232 performance. The device features an auto-shutdown features that enters the device into a low-power shutdown mode when the RS-232 cable is disconnected or the devices driving the transmitter and receiver inputs are inactive for more than 30s.

2.2.1.8 Switched Regulators – Texas Instruments LMZ14202

The LMZ14202 power module is a DC-DC voltage regulator. Unlike linear regulators which regulate power by dissipating the unnecessary power switching regulators actually on-off the output power resulting in much superior conversion efficiency (90%) along with considerably lower heat generation and robustness.

The selected part offers exceptional power conversion efficiency, line and load regulation and output accuracy. The LMZ14202 can take an input ranging from 6.0 V to 42 V with an output range from 0.8 V to 6.0 V. The part features integrated shielded inductor and requires a simple PCB layout leading to low external component count. It has a control pin that allows the output to be controlled through a microcontroller. It supports thermal shutdown, input under-voltage lockout, output over-voltage protection, short-circuit protection, output current limit, and allows startup into a pre-biased output. It also features control over the switching frequency.

2.2.1.9 Oscillator – Abracon ASFLMB-80

The Abracon ASFLMB series features very low power consumption using 2nd generation MEMS technology with reduced jitter along with exceptional stability. An 80MHz model is used in the design in order to power the clock for the FPGA and data clocking features.

2.2.1.10 Linear Regulators

A. Microchip MCP1824

The MCP1824 is a fixed/adjustable low dropout (LDO) linear regulator that provides high current and low output voltages. An adjustable version is used to provide for more control over the output voltage. It operates from 2.1 V - 6.0 V and provides output of 0.8 V - 5.0 V. The 300 mA output current capability, combined with the low output voltage capability, make the MCP1824 a good choice for new sub-1.8V output voltage LDO applications that have high current demands. The output can be stabilized easily using ceramic capacitors reducing the size and cost of the overall regulation circuitry. It also features short circuit current limiting and over-temperature protection.

B. Analog Devices ADP3339

The ADP3339 is high accuracy, low noise, low-dropout voltage regulator that works in the input range of 2.8 V – 6 V and delivers load current up to 1.5 A. The device features safety current limit and thermal overload protection.

2.2.2 Motherboard Schematic and Layout Design

The schematic and layout design was done in PCB Artist software from Advanced Circuits. The specific software was selected since we also use their fabrication facility for the system boards and the software provides a good interrelation to the fabrication process from the design phase. In this section I provide an explanation as to how the different circuitry is designed and where applicable, how to configure the different components.

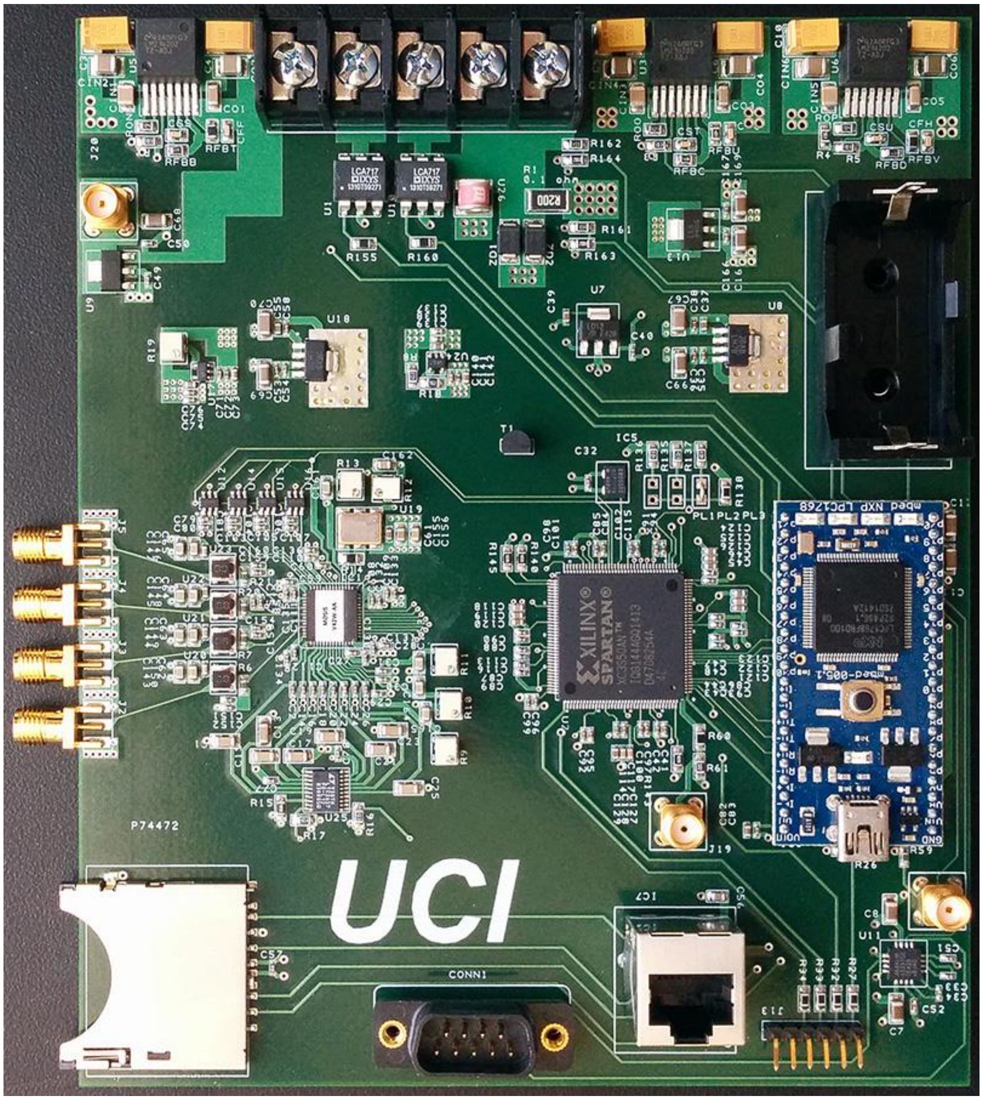


Figure 2-2: System board. Photo taken by Dr. Stuart Kleinfelder.

2.2.2.1 Overvoltage Protection

To protect the system from overvoltage conditions special protection mechanisms are adopted. The system has two zener diodes which are rated at 33 V with very fast response time of around 1 ns. Further protection is provided by a Gas Discharge Tube (GDT) rated at 60 V breakover voltage with a sparkover voltage of 500 V.

The ON Semiconductor 1SMB33AT zener diodes are designed to protect voltage sensitive components from high energy transients. These provide clamping capability and surge protection capability. The selected component also features very fast response time to protect against sudden voltage surges. The zener diode works in the zener breakdown mode to pass excess current from sensitive circuitry leading to a drop in voltage. Transient surges need to be avoided and hence the fast response time of <1 ns works fast to drop the excess voltages. The excess current is grounded and removed leading to a drop in voltage bringing the diode out of the breakdown condition.

Further protection is provided by using a GDT. GDTs work differently to zener diodes but has a similar effect on the electrical dynamics. When a surge occurs and reaches the sparkover value, the GDT becomes a virtual short. This is known as arc mode. The short removes the excess current through the GDT to ground and removes the voltage surge away from the equipment. At normal conditions the GDT remains in a high-impedance off-state condition. As the voltage increases into the glow region the gas ionizes due to the charge development. This gives rise to an avalanche effect and the device transitions into a short circuit condition once the Arc voltage is reached. A good GDT transitions into the Arc voltage quickly to prevent high transients. The selected component transitions to the arc mode in less than 500 mA. Once the energy drops below the arc voltage condition the GDT resets into the high-impedance mode, this is called the extinguishing voltage, holdover voltage or impulse reset voltage [8]. The GDT is designed to be a redundancy protection in the situation that the voltage goes beyond the zener diode breakdown, which may cause permanent damage to the diodes.

2.2.2.2 DC-DC Power Conversion

The Texas Instruments LMZ14202 is used to provide switched DC power for the MBED, the amplifiers and the system board circuits. The power source (solar or wind) provides a 12 V input power. The 12 V input line is the input to three sets of LMZ14202 ICs and the corresponding support circuitry. The 12 V line also provides power to the AFAR Ethernet Bridge. A sample circuitry is shown in fig. 2-3.

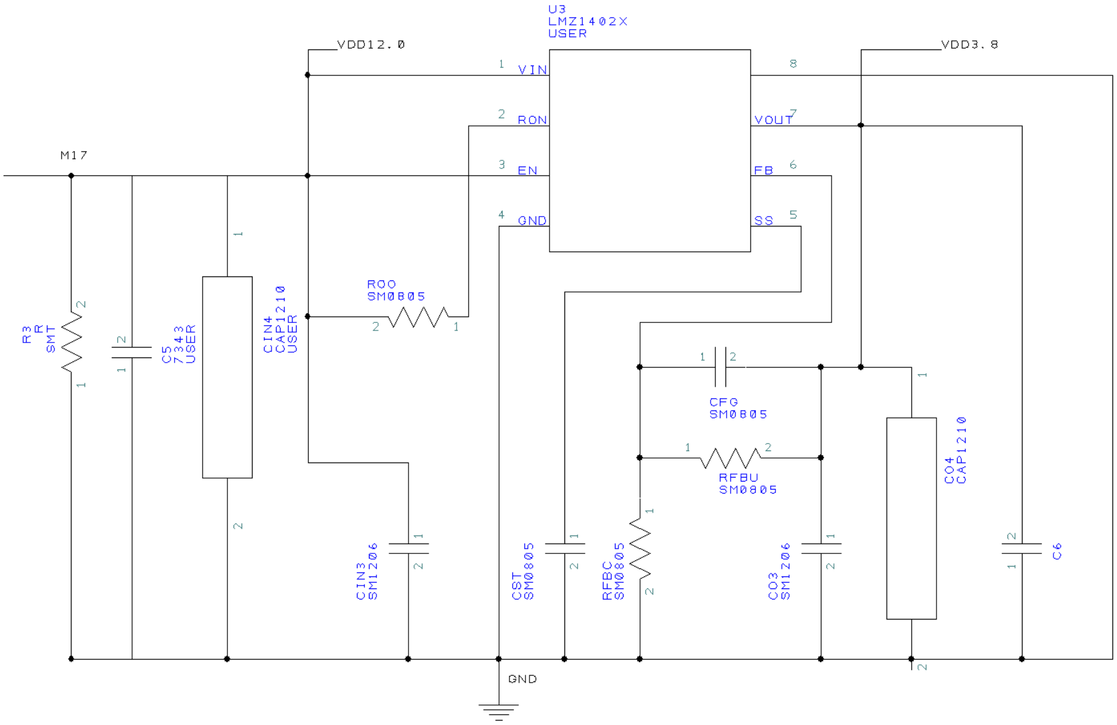


Figure 2-3: Circuit for the DC-DC convertor LMZ14202.

The ratio of R_{FBT} and R_{FBB} determines the output voltage according the following equation

$$V_{OUT} = 0.8\ V * (1 + R_{FBT} / R_{FBB}) \tag{2-1}$$

The rest of the component values can be selected directly from tables given in the component datasheet. A 100 µF capacitor is enough to stabilize the output however in order to improve the output stability further both mechanically and electrically two more bypass capacitors are added. A 100 µF tantalum capacitor provides mechanical protection and a 1 µF capacitor widens the frequency response of the bypass circuit. A similar approach is taken on the input side. The input side have three bypass capacitors, 22 µF, 10 µF and 1 µF capacitors. Other than the wider range of the bypass capacitors the input side also sports 50 V rated 22 µF tantalum capacitors compared to the 16 V rated 100 µF capacitors on the output side. This is in order to protect the regulators against power spikes from the power supply systems. The power supplies are known to spike up to 25 V so the capacitors are selected to protect against all such spikes. One of the regulators is used to provide 5.0 V supply to the MBED and the Iridium system. The other two output 3.8 V and are used to power the amplifiers and the system board. These two regulators also have the control pin (pin 3) connected to the MBED through pull-up resistors, and can be controlled by the MBED, allowing the system to power down.

The system was designed earlier however some changes have been made in the present format. The control pin connections were changed from control through FET to direct pin control from the microcontroller. Modifications were also made to the layout of the components to implement a significantly more compact layout. The overall design was also integrated into the single board system from the earlier separate power module system.

2.2.2.3 System Power

One of the DC-DC convertor provides power to the system board. A 3.8 V supply provides power to a 3.3 V regulator from which all power is derived from. A detailed map of the power system is shown in the fig. 2-4.

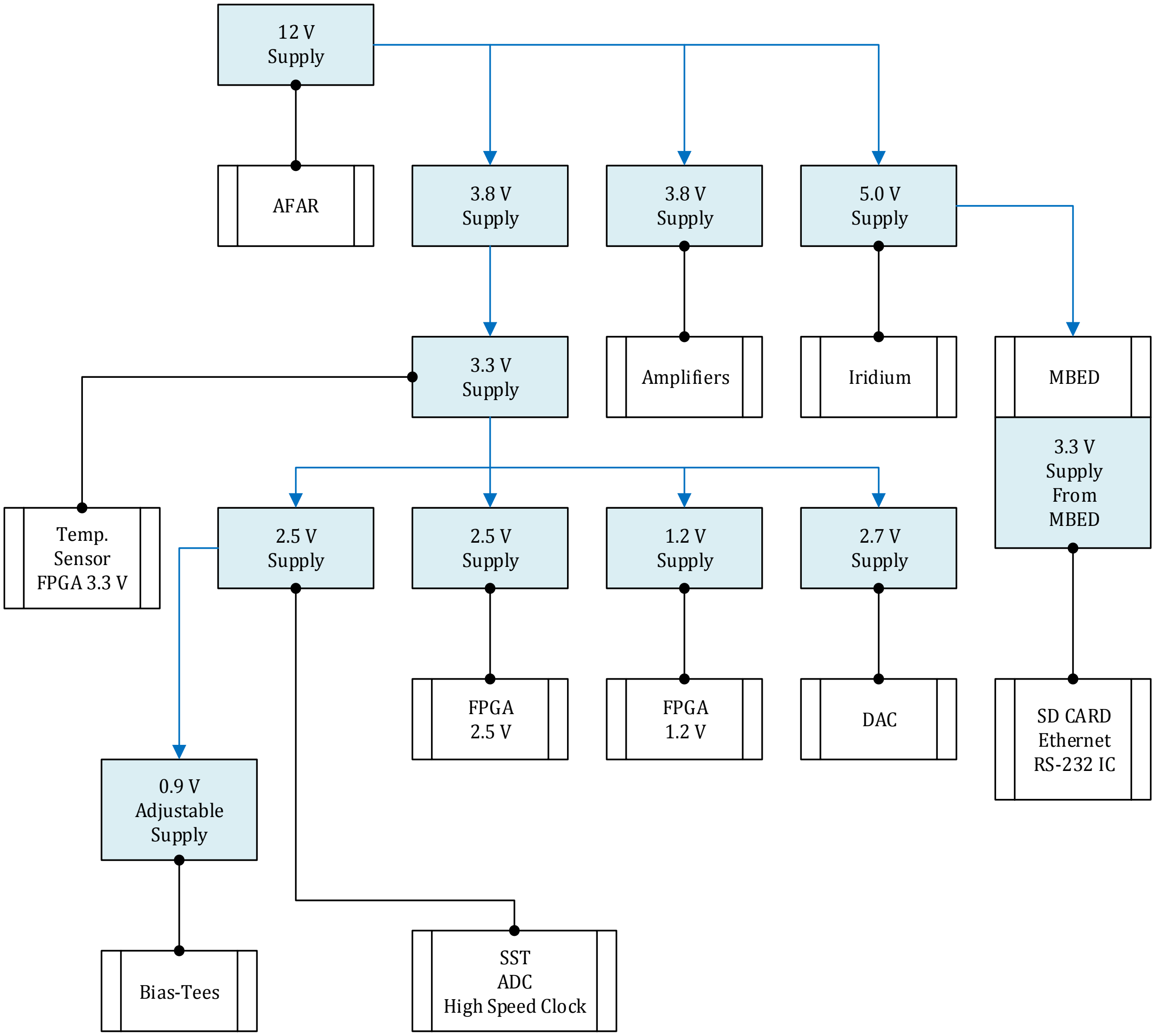


Figure 2-4: Power system overview

The 3.3 V supply powers linear regulators to generate other necessary voltages. It also supplies power to the temperature sensor. The FPGA needs 2.5 V and 1.2 V supplies aside from the 3.3 V. Couple of Texas Instruments LP3871EMPX LDO regulators are used to generate two 2.5 V supplies. One supply is used by the FPGA, while the other is used to generate the power supply for the SST. Separation of the power source is necessary for power source noise isolation between the digital and the analog circuits. The 2.5 V supply for SST also supplies power to the ADCs and the high-speed clock. This same supply also supplies power to an adjustable 0.9 V LDO voltage regulator which powers the bias-tees. A 1.2 V supply is generated from the 3.3 V for powering the FPGA outputs. Another voltage regulator generates a 2.7 V supply which is used exclusively by the DAC.

Another DC-DC regulator provides 3.8 V power exclusively to the amplifiers while another provides 5.0 supply which is used by the Iridium platform and the MBED. The 5.0 V supply is not controllable and is always ON. The MBED provides a regulated 3.3 V supply to be used by peripheral components connected to the MBED. This supply powers the Secure Digital (SD) card slot, the Ethernet jack and the RS-232 communications chip, all connected and controlled by the MBED.

2.2.2.4 Configuration of the FPGA

The FPGA needs 1.2 V for powering the internal core (V_{CCINT}) and 3.3 V for internal circuit biasing (V_{CCAUX}). The output driver supply (V_{CCO}) can be in the range of 1.2 V – 3.3 V depending on the I/O standard selected and each of the I/O banks can be independently powered and hence can have any I/O standard. In the current system three of the I/O banks (bank 0, 1, 2) of the FPGA runs on LVDS_25 standard and the corresponding V_{CCO} is selected to be 2.5 V. The MBED however runs on the LVDS_33 standard and hence all the I/O pins connected to the MBED are kept on one of the banks (bank 3) and the standard is selected to be LVDS_33 and the corresponding V_{CCO} is 3.3 V.

The FPGA supports a number of memory configurations for storage and also allows a number of different methods for loading the configuration into the FPGA internal memory. In order to reduce the peripheral circuitry and overall simplification the internal In-System flash memory is used to store the configuration. The type of memory system is determined by three pins on the FPGA - M0 (p38), M1 (p37) and M2 (p39). The different configurations are indicated in table 2-1.

Table 2-1: FPGA configuration modes

M [2:0]	Spartan-3AN
<0:0:0>	Master Serial Mode
<0:0:1>	Master SPI Mode
<0:1:0>	BPI Up
<0:1:1>	Internal Master SPI
<1:0:0>	NA
<1:0:1>	JTAG Mode
<1:1:0>	Slave Parallel Mode
<1:1:1>	Slave Serial Mode

Since we use the internal flash memory to configure the FPGA we selected the Internal Master SPI mode and hence set <M2:M1:M0> to be <0:1:1> as shown in table 2-1. During configuration the FPGA needs the M [2:0] pins as well as the DONE pin (p73), INIT_B pin (67), PUDC_B pin (p143) and the PROG_B pin (p144). The CCLK pin (p72) is also necessary during configuration if a slave configuration mode is used, however since we use a master mode this pin remains free. Once the configuration has been loaded onto the FPGA all the pins other than DONE and PROG_B pins are released and can be used for user I/O. Only the M1 pin has been reused as a user I/O pin in the current system.

While the DONE pin can be externally driven to keep the FPGA free for accepting new configurations, a necessary condition when multiple FPGAs are being configured in a daisy chain system. However if a single FPGA is used and if any of the dual pins (configuration pins and then user after I/O pins) are used for user I/O then it's a good practice to leave the DONE pin floating and force the pin through software configuration instead. This problem occurred in the current system and hence, on the hardware, although the circuit exists for externally driving the DONE pin, the connection was removed as a solution for the problem during testing of the system.

2.2.2.5 Configuration of the DAC

The decision to switch to the I2C interface from the previous generation system's SPI interface was made to simplify the firmware as the SPI line is left free for only data rather than component configurations. The DAC offers eight independent outputs which are all used by the MBED to set threshold values for the SST. The DAC features two pins which are used to decide the power-up voltage value, the PORSEL (p14) and REFLO (p1). Both are tied to the GND voltage (0 V) and hence the DAC resets to 0 V on power-up. The hardware update feature, set by the use of the pin $\overline{\text{LDAC}}$ (p8), is not being used and the control has been left completely with the MBED platform. The I2C interface uses two pins SCL (p10) and SDA (p11) to interact with the component. Both have been pulled-up to V_{CC} as required by the I2C interface specifications. SCL is the system clock and SDA is the data pin. Both can be controlled from the MBED since it provides in-built support for I2C. The connections are shown in fig. 2-5.

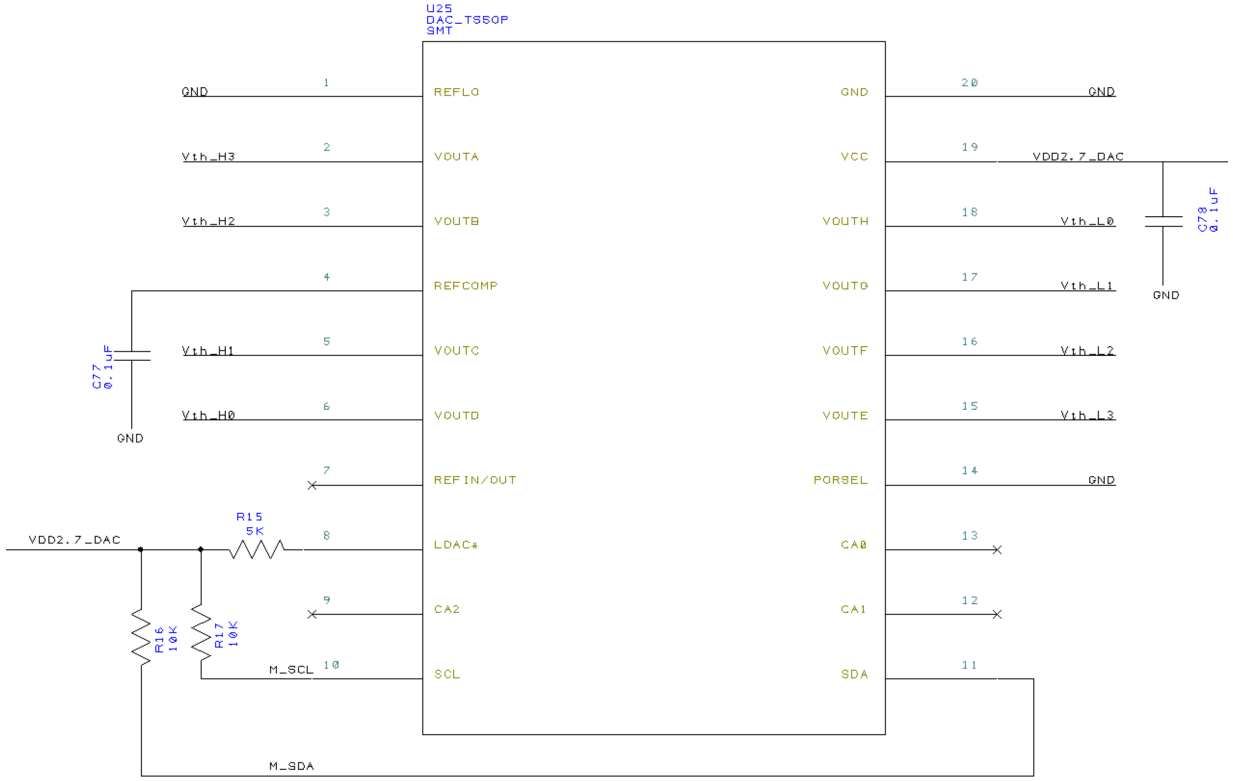


Figure 2-5: Schematic for the DAC

The I2C interface allows multiple devices to share the same bus, by allowing individual devices to be set to specific addresses. The address can be set by connections (V_{CC}, GND or FLOAT) to three address pins CA2 (p9), CA1 (p12) and CA0 (p13). In the design all the address pins have been left floating, which sets the address to be 1000001. Aside from the address that is set, a global address exists which is hardwired and sends the same instructions to all I2C devices on the bus. Given that there is only one device on the I2C bus, both addresses can be used for addressing the DAC in the present design.

Table 2-2: I2C addresses eligible for DAC configuration

CA2	CA1	CA0	Address
Float	Float	Float	1 0 0 0 0 0 1
Global Address			1 1 1 0 0 1 1

2.2.2.6 Design of the Analog Inputs

The analog data is sourced after the amplifier stage. Special care has been taken to load match the analog inputs in order to reduce signal loss. The PCB tracks carrying analog signals all need to be matched to 50 Ω and as such the tracks need to be designed as transmission lines. A microstrip line approach was decided upon. The calculations for the PCB tracks are as given below. The thickness of the trace was acquired from the manufacturer. The substrate thickness was also acquired from the manufacturer. Given a four layer PCB, only the top substrate thickness is considered since the second layer is the ground plane and in microstrip line structure the transmission line is formed between the conduction layer and the ground layer. The substrate material is standard FR4 material. The substrate material's dielectric constant or the relative permittivity value is necessary for the calculations. The constant was taken to be 4.5 as indicated by the manufacturer.

The equations for the calculation can be found in [9]. An online tool utilizing the same equations for the calculations was used to calculate the necessary trace width.

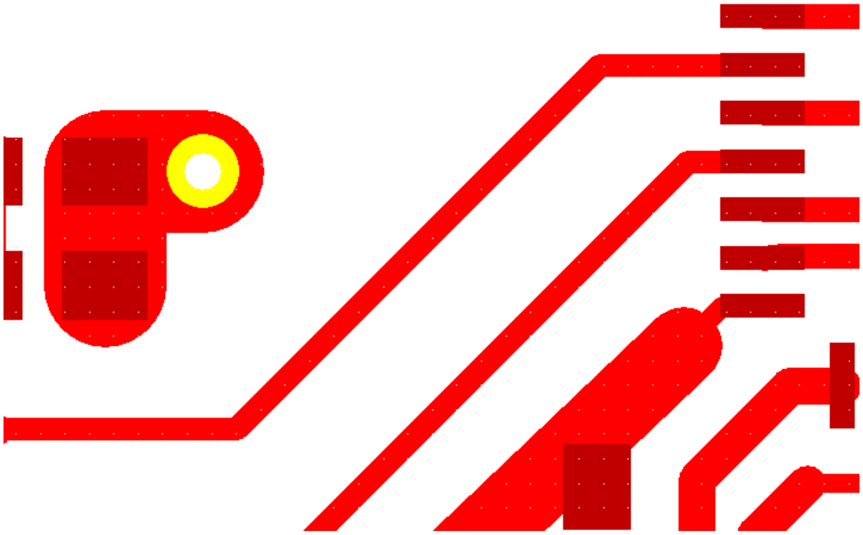


Figure 2-6: Angles in analog traces

Care was taken to keep the microstrip line traces as short as possible to reduce signal loss. The traces were also carefully designed to avoid corners as much as possible, with any corners kept to obtuse angles only as shown in fig 2-6, to reduce loss.

Bias-tees are used to bring the analog inputs in the working range of the SST inputs. The bias-tee injects DC power into RF signals. The output (RF+DC) must be properly matched and terminated to maintain signal integrity by eliminating reflections while reducing power consumption. The traditional approach of terminating with a matched resistance results in power loss due to constant dissipation. An alternative approach of terminating with matched resistance and capacitors is taken, shown in fig. 2-7. The 50 Ω matched resistance provides the impedance match needed to minimize reflections at the load and the capacitor’s function is to block the DC average current thus removing power consumption issues.

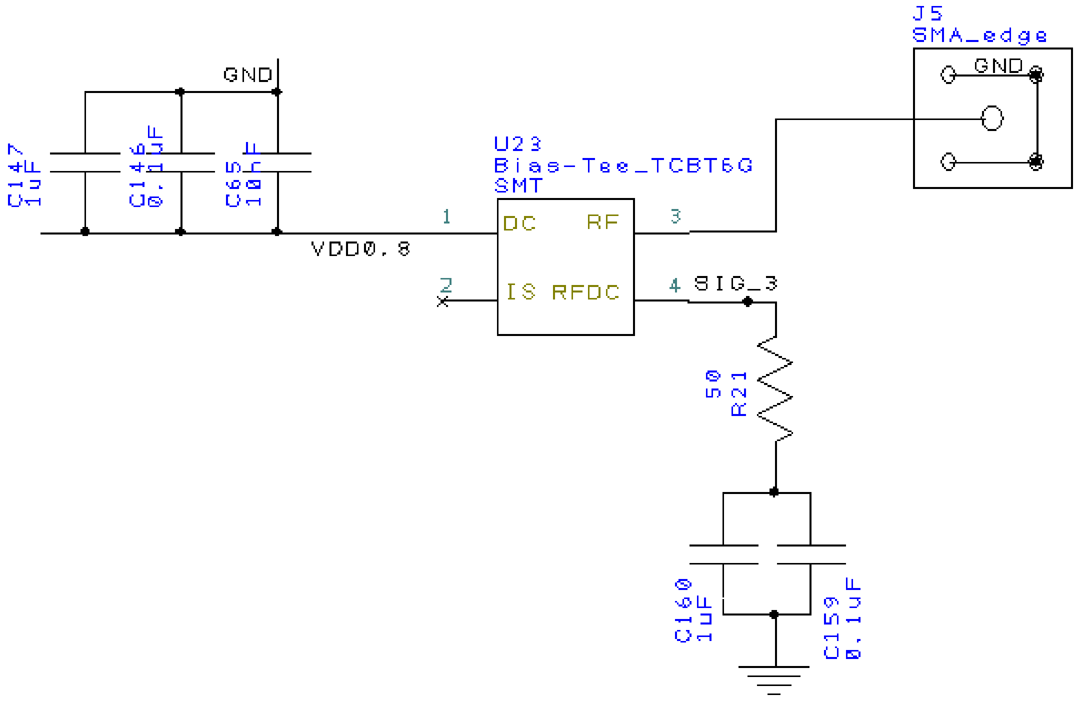


Figure 2-7: RF signal bus termination

2.2.2.6 Support circuitry for the SST

The SST requires 2.5 V supply which is provided using a separate power supply. Bypass capacitors are used for SST pins connecting to the V_{DD} and the GND for improving signal integrity. The traces are kept as short and wide as possible to reduce trace resistance and maintain signal integrity. The traces are narrowed near the SST pin to meet design rules and avoid shorts arising due to manufacturing errors. The SST uses a high-speed 1 GHz clock to synchronize its functions. The clock gives a LVDS output. These outputs need to be separated from all power traces or else risk contaminating the sources with high frequency noise. These traces are isolated as much as possible and separating ground traces introduced wherever isolation is difficult. The oscillator also has a control pin connected to the FPGA which allows the oscillator to be turned-off if necessary.

The SST has five bias voltages which are adjusted using potentiometers to maintain control over them. Four of the voltages are referenced to GND while the LVDS bias voltage is referenced to V_{DD}. The schematic is shown in fig. 2-8.

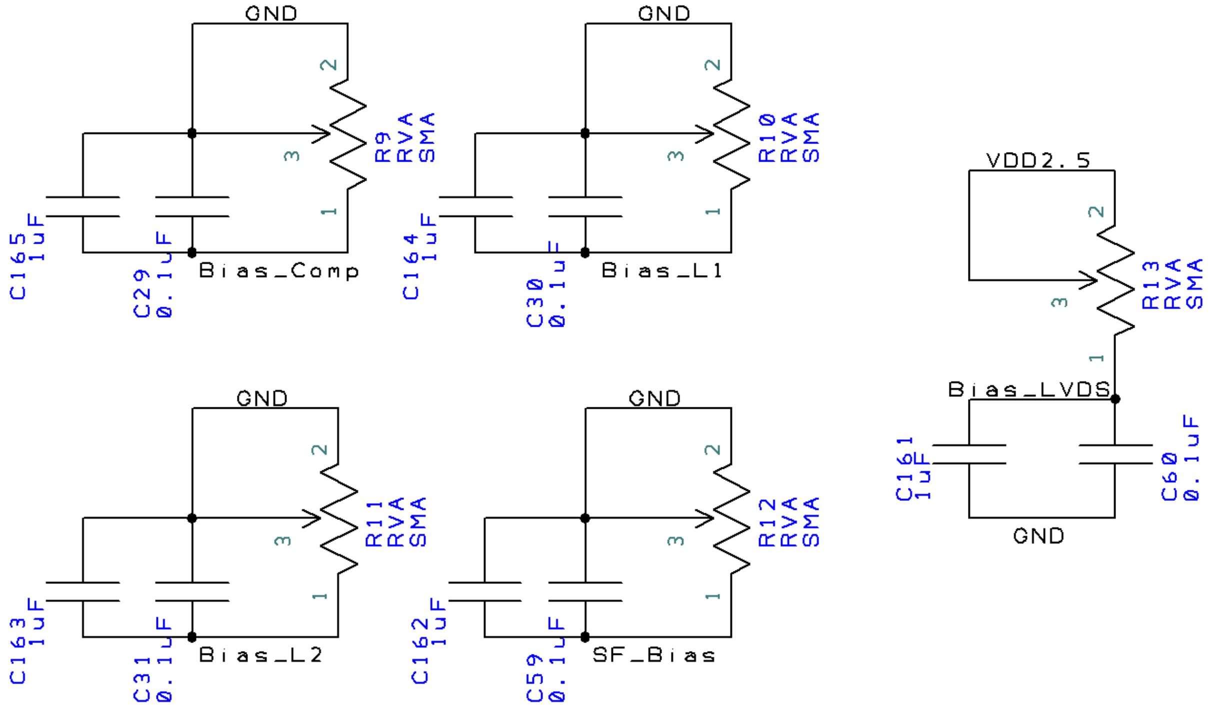


Figure 2-8: Potentiometer circuitry for bias voltage control

The analog outputs are connected to four ADCs (one per channel) which digitize the analog data before sending it to the FPGA. The rest of the connections are to the FPGA which are explained in the firmware section.

Chapter 3: Firmware Design

3.1 System Overview

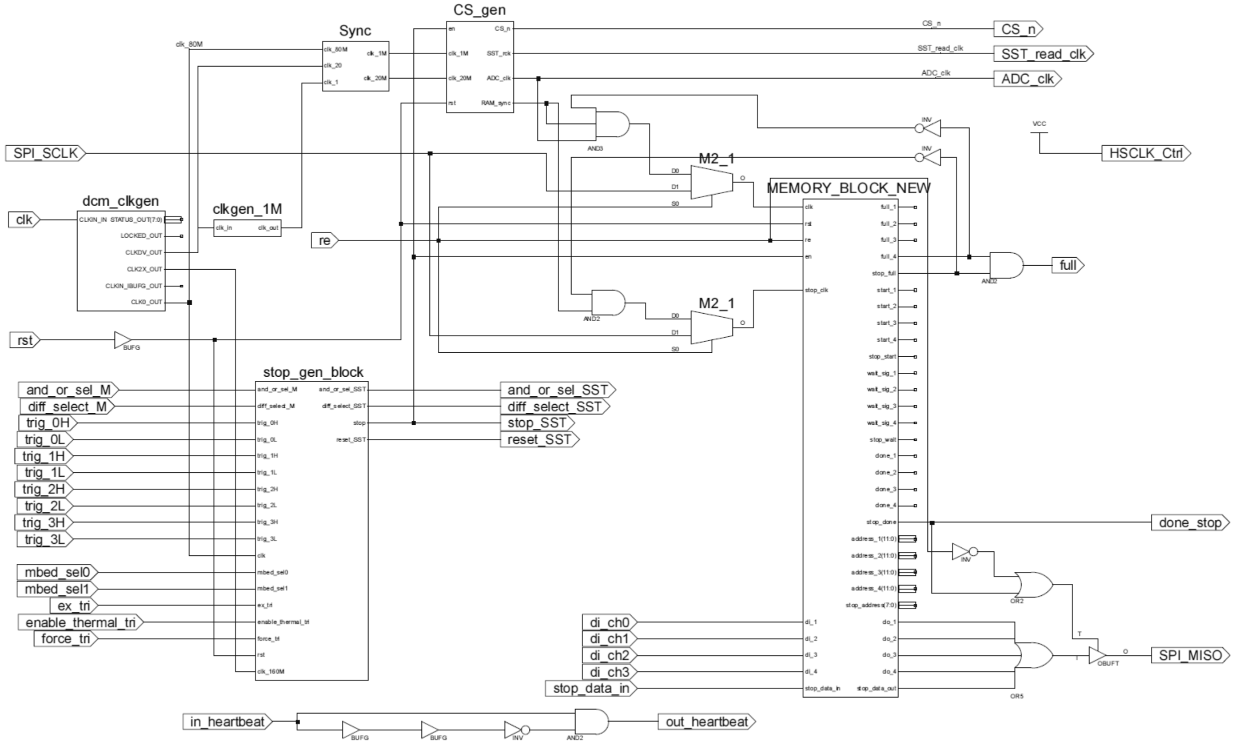


Figure 3-1: Firmware schematic design

The firmware for the new system was built from the ground-up to be a simpler and more automated system than the previous generation system. The new SST chip is simpler function wise than the previous generation ATWD chip, allowing for a simpler control architecture. The fig 3-2 shows the primary control logic structure for the firmware. The firmware is designed in Xilinx ISE environment with a mix of code and gates generating schematic blocks.

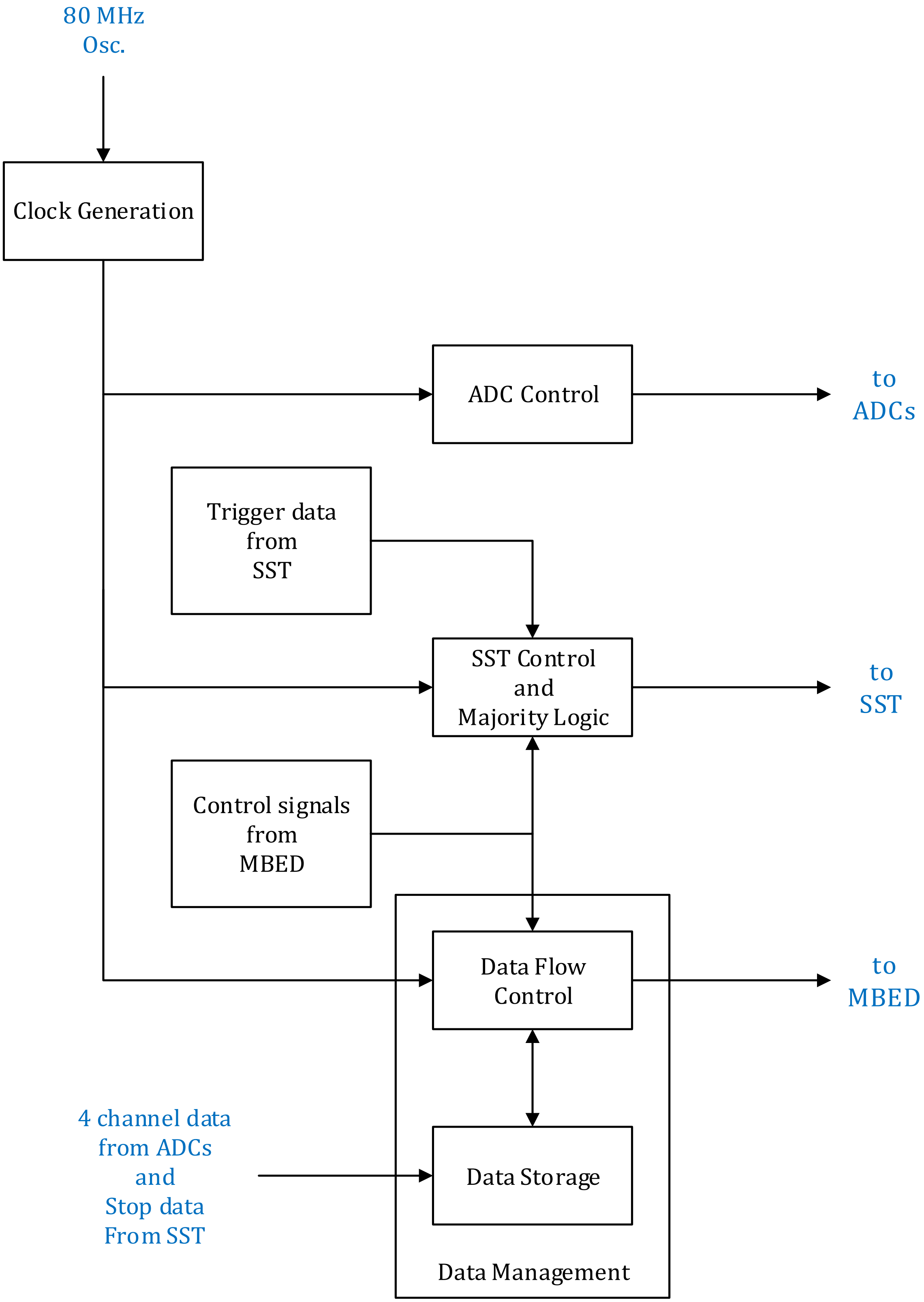


Figure 3-2: Logic architecture for the FPGA

3.2 Design of the Firmware

3.2.1 Clock Generation

The motherboard features an 80 MHz oscillator for all clocking functions in the FPGA. The FPGA takes the 80 MHz clock and generates all other clock frequencies necessary by using the in-built DCM (Digital Clock Manager) and subsequent logic.

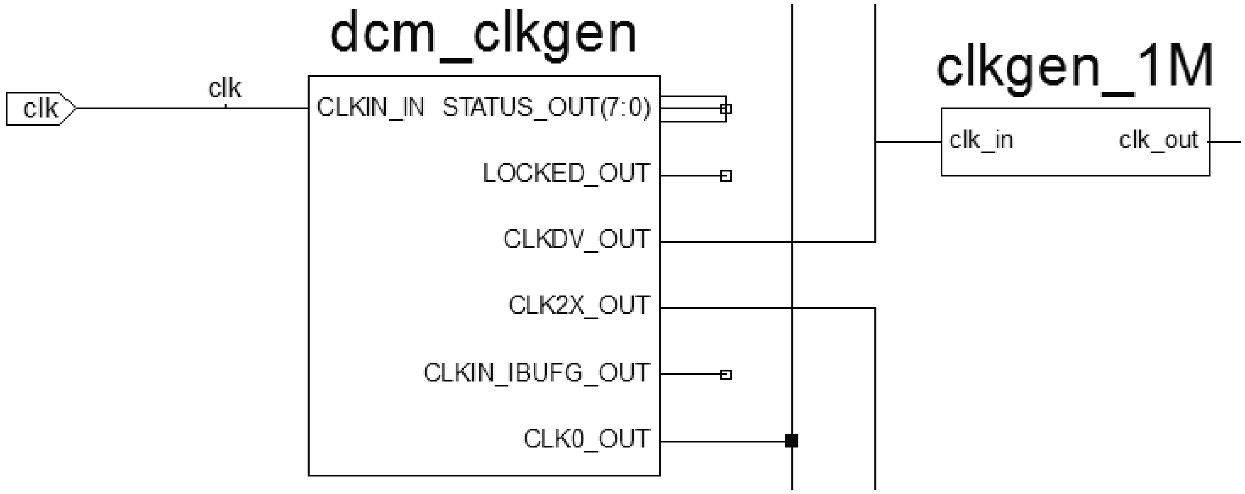


Figure 3-3: Clock generation logic

The in-built DCM is a Phase Locked Loop (PLL) based standard logic block that is accessible through the software. The 80 MHz oscillator frequency is taken and is divided by four to generate a 20 MHz clock and multiplied by two to produce a 160 MHz clock. The 80 MHz clock is also passed through. The DCM supports automatic duty cycle correction, phase shift and feedback. The 20 MHz frequency is then divided further in the clkgen_1M block to generate a 625 KHz clock. All clocks are 50% duty cycle.

3.2.2 Clock Synchronization

While the 625 KHz clock is generated from the 20 MHz clock further synchronization is necessary to have perfectly synchronized clocks due to duty cycle errors or inherent inefficiencies in the firmware logic implementation. This is achieved through further operations on the clock signals. The logic is shown in fig 3-4.

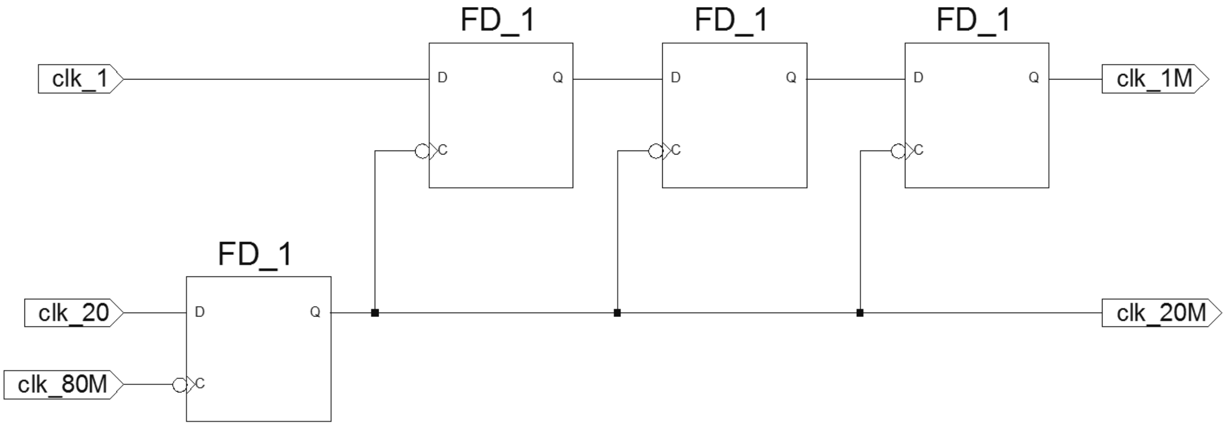


Figure 3-4: Clock synchronization logic

The synchronization is achieved through a network of D flip-flops. The 80 MHz clock is first used to synchronize the 20 MHz clock. A negative edge triggered D flip-flop (FD) is used, so the FD clocks the input only when it detects a negative going edge of the 80 MHz clock. The 20 MHz clock is then used to synchronize the 625 KHz clock in a similar fashion. Three flip-flop stages are used for synchronizing the 625 KHz clock to avoid metastability condition. The multiple stages are employed only for the slower clock since only the 20 MHz and the 625 KHz clock need to be synchronized as a necessity in the firmware as they are used together for generating the ADC control signals.

3.2.3 ADC Control Generation

The selected ADC works at 1 MSPS input speed but outputs 16 bit data with maximum readout speed of 20 MHz. ADC requires a chip select signal ($\overline{\text{CS}}$) and a read clock signal (SCLK). It takes analog input and gives a digital output (SDO). The $\overline{\text{CS}}$ and SCLK are generated on the FPGA. The $\overline{\text{CS}}$ signal is just the 625 KHz clock signal, shown in the fig. 3-5 as CS_n. The block takes four inputs – en (which is an enable signal), clk_1M (which is the 625 KHz clock), clk_20M (which is the 20 MHz clock) and rst (reset signal). The block provides the $\overline{\text{CS}}$ signal as CS_n, SST readout clock – SST_rck, ADC SCLK readout clock – ADC_clk and RAM_sync signal which is to synchronize the memory to the ADC readout.

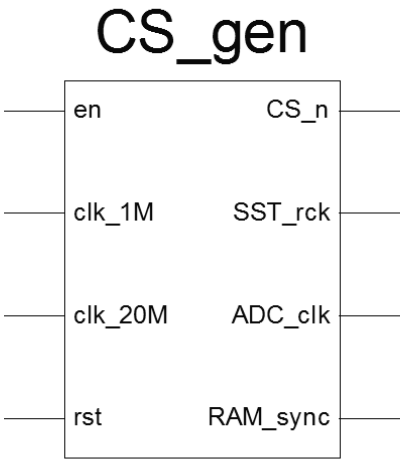


Figure 3-5: ADC control generation block

The ADC digitization begins at the falling edge of the CS_n signal, and the chip is put into tri-state condition when CS_n goes HIGH and stays that way until the next falling edge of CS_n. The ADC also requires a readout clock aside from the CS_n signal. After the first falling edge of CS_n has clocked the first bit out, subsequent bits are clocked out every falling edge of the ADC_clk signal for 16 clock cycles. The data word is composed of 4 leading zeros, followed by 12-bit data in MSB first format. The clocks have been designed to provide a simple interface for the ADC readout. The CS_n signal is a 1600 ns period (50% duty cycle) clock and hence provides 800 ns for the data conversion since the data is clocked out only during the clock logical ‘0’ condition. The readout clock is kept at 20 MHz or 50 ns clock period, which allows exactly 16 bits readout time i.e. 16 bits x 50 ns = 800 ns.

The ADC needs a dummy cycle of $\overline{\text{CS}}$ going low for more than ten SCLK falling edges to come out of power down mode which is every time the ADC is powered down or if the $\overline{\text{CS}}$ signal goes HIGH before a conversion is completed. So careful consideration is needed to be taken to avoid getting garbage data on the first read cycle. This is taken care of by driving a full dummy read cycle without clocking the actual analog data, and throwing away the ADC data during this cycle. Failure to perform this dummy read resulted in saturated data i.e. full scale ADC output value.

The CS_gen block also generates the SST read clock – SST_rck. The SST_rck is just the 625 KHz signal non-inverted. The code also generates a RAM_sync signal which is used to determine when there is useful data from the ADC. This signal is just the 625 KHz clock signal inverted but delayed by one clock cycle to account for the dummy ADC read, since only when the clock signal is low that the ADC output is meaningful, so the inverted signal is used to control the data storage.

3.2.4 SST Control and Majority Logic

The new SST requires three control signals for configuration. Two of these signals are provided by the MBED. The other signal is generated in the FPGA code using the trigger signals from the SST. The control signals generated by the MBED are used in the code to condition the channel data and then passed onto the SST. The signals are shown in fig. 3-6.

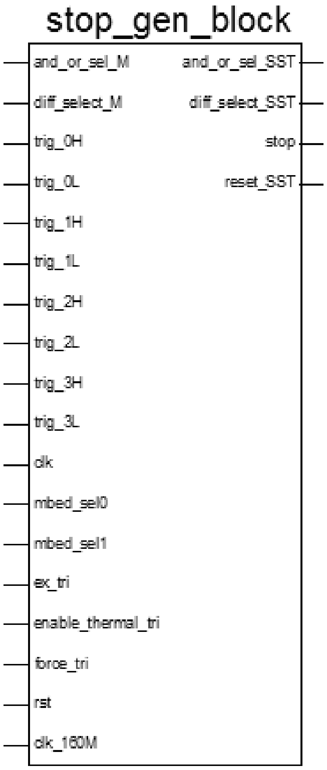


Figure 3-6: SST control logic block

The SST control inputs from the MBED are and_or_sel_M, diff_select_M, these are passed onto the outputs and_or_sel_SST and diff_select_SST respectively onto the SST. The trigger inputs from the SST are used according to the MBED controls to condition the channel data and then passed onto the majority logic block, which then decides if there is a trigger according to whether the system has been configured to trigger when there is four/three/two/any coincidental triggers. The trigger signals are as follows: Trigger_0_H, Trigger_0_L, Trigger_1_H, Trigger_1_L, Trigger_2_H, Trigger_2_L, Trigger_3_H and Trigger_3_L. The SST supports differential triggers or single ended triggers. The triggers in differential mode are handled in pairs where as in single ended mode all eight triggers have to be handled independently and that's where the and_or_sel signal becomes useful. The logic is shown in fig 3-8. The majority logic block, which determines how many channels need to be coincidental, was designed by Mahshid Roumi and also features external trigger and forced trigger inputs as well as the control switch signal for enabling thermal triggers in the trigger conditioning.

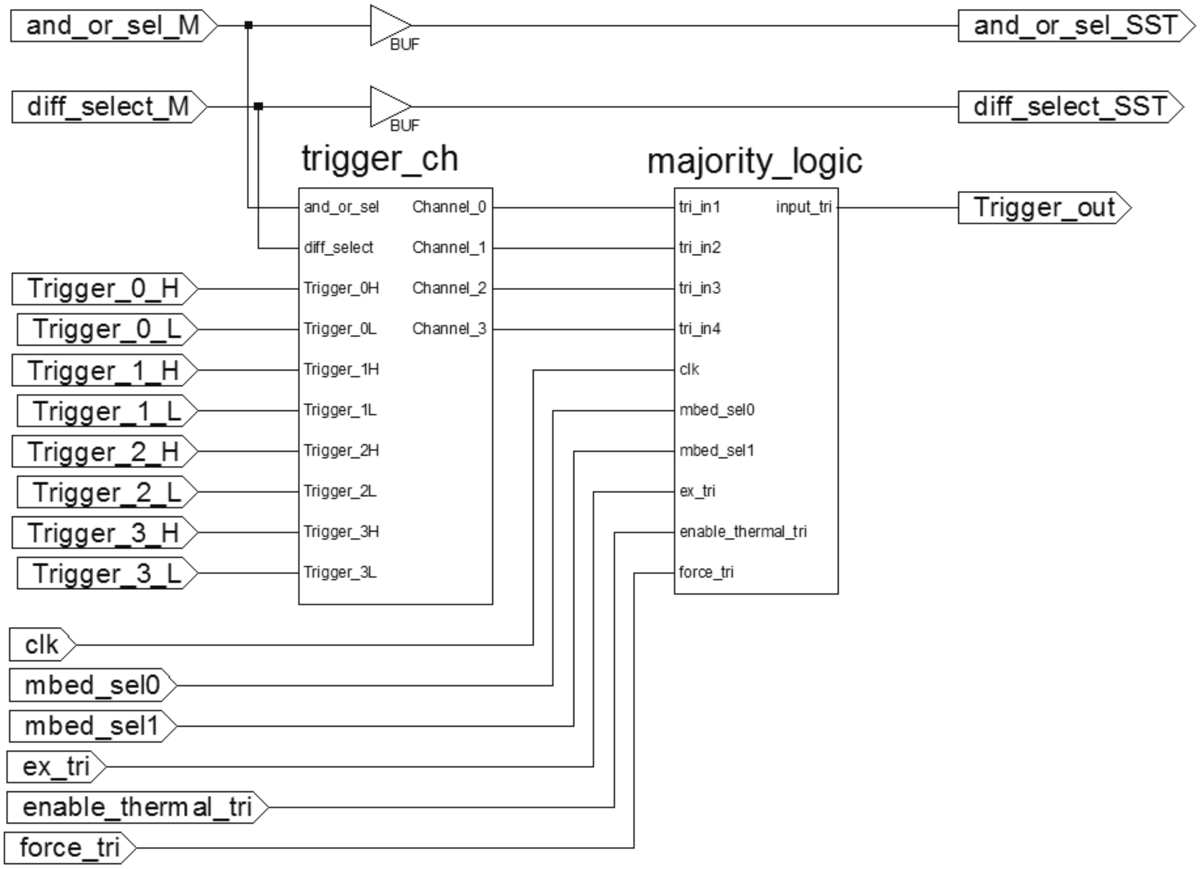


Figure 3-7: Signal conditioning and majority logic. The majority logic block was designed by Mahshid Roumi.

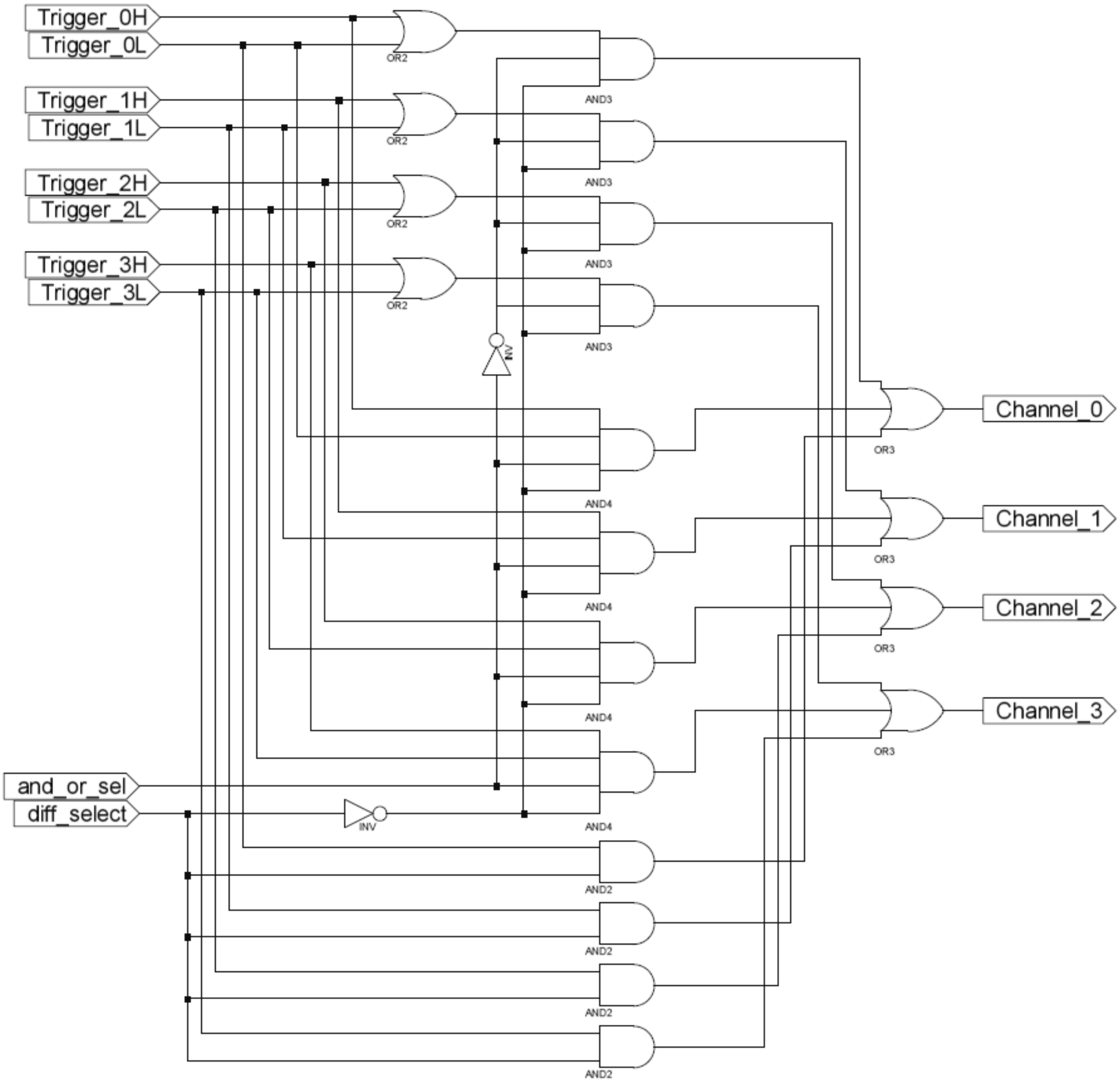


Figure 3-8: Trigger signal conditioning logic

In differential mode only the low triggers are used in each channel and that constitutes a trigger in that channel. In single ended mode, which is not used in current deployment, the high and low triggers are considered independent and further conditioning may be needed with the and_or_sel signal to form the channel trigger. If the majority logic decides a trigger has occurred then the majority_sys block (shown in fig. 3-9) outputs the trigger signal which is then used to generate a stop signal which is required by the SST to stop sampling the analog inputs further. The stop signal is delayed by a customizable delay logic. The delay is necessary in order to allow time for the incoming analog signal's tail to be sampled by the SST, since it's not known which part of the analog signal has crossed the set thresholds and generated the trigger and this uncertainty is why the delay needs to be customizable. The default delay is currently set to approx. 62 ns.

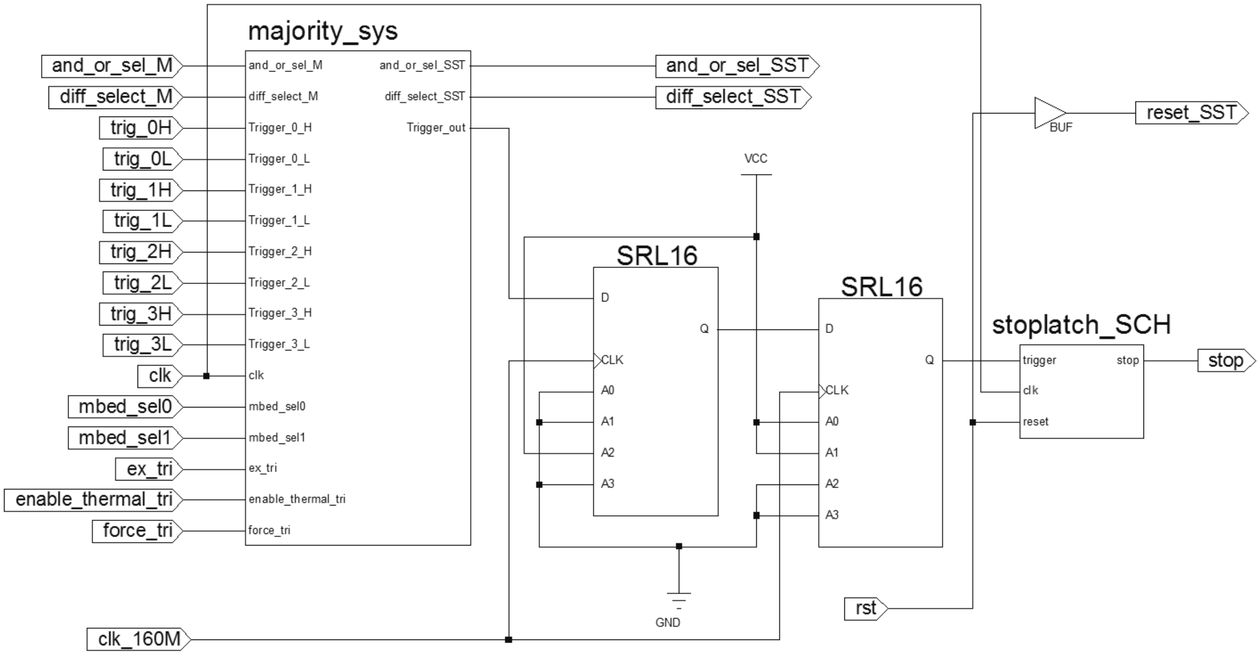


Figure 3-9: Stop signal generation and delay

The delay can be changed by changing the V_{CC} and GND connections to the shift registers in the firmware. The delay logic is comprised of two 16-bit long shift registers (SRL16). The synchronous shift registers operate on the asynchronous trigger signal resulting in some jitter at the shift register output. To reduce the jitter the shift registers are clocked by the faster 160 MHz clock to increase the adjustment resolution, which allows for a theoretical jitter of 6.25 ns. When the stoplatch_SCH block sees a trigger it sends a stop signal to the SST and the stop signal is reset only when a reset signal from the MBED comes in. The reset signal is also passed onto the SST at this stage. The stop signal also functions as an enable signal for the entire memory and ADC control logic of the firmware.

3.2.5 Data Flow Control

The firmware is designed to collect the data from the respective ADCs and store the data in the FPGA block memory system along with the stop data from the SST, and then inform the MBED that there is data available for collection. The system is then designed to wait until the MBED is interested in the data. When the MBED is ready to collect the data it sets a read enable signal and activates the SPI clock. The data is then read from the FPGA and put onto the SPI channel one channel at a time followed by the stop data. The system can be reset at this point as it will allow the SST to start sampling the analog inputs. The process is displayed in fig. 3-10 and the code schematic is shown in fig. 3-11.

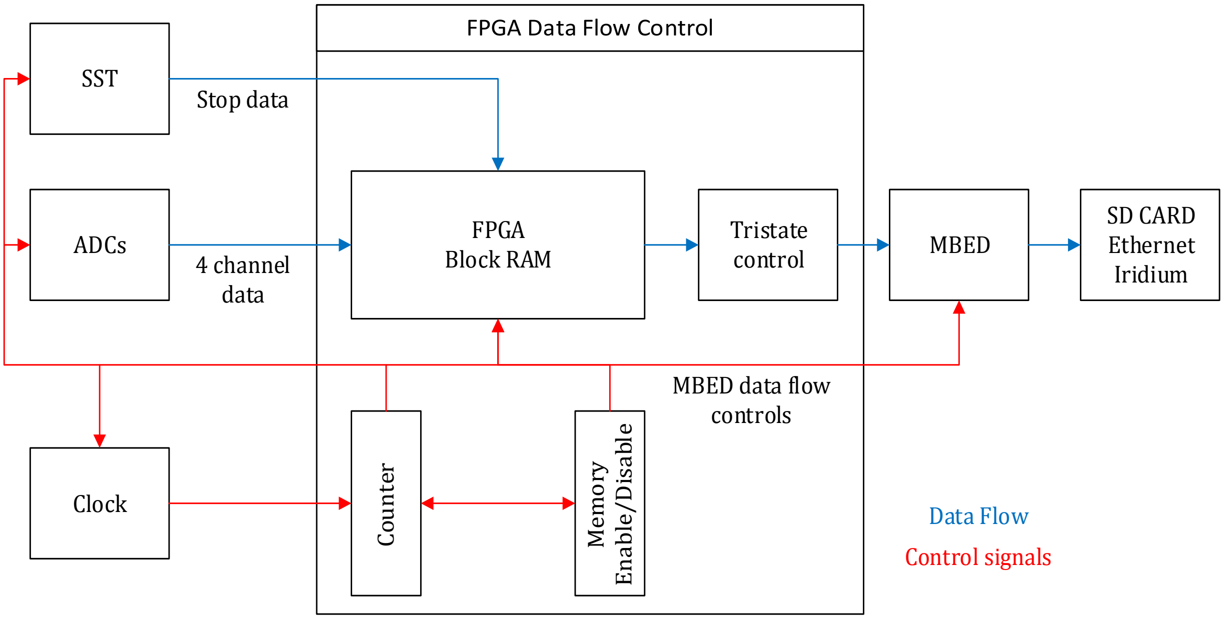


Figure 3-10: Data flow control overview

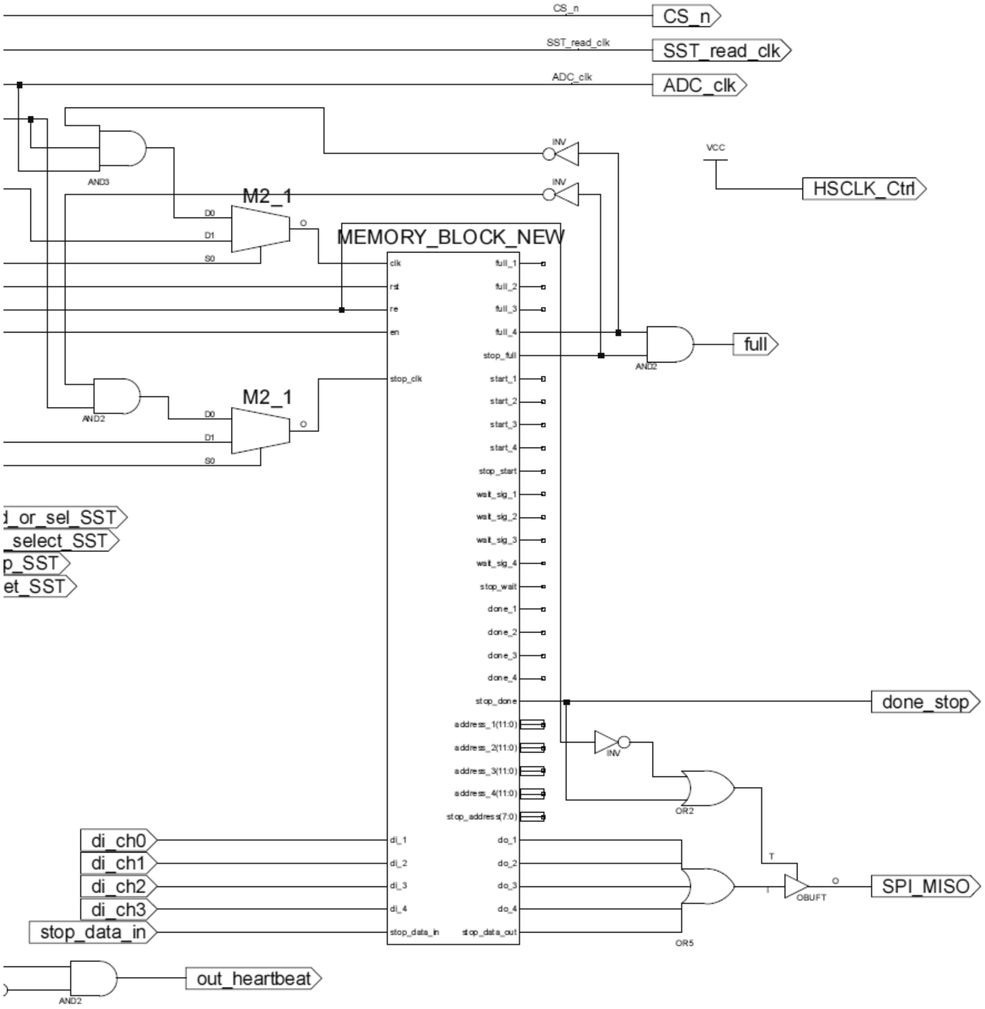


Figure 3-11: Memory storage logic and tri-state control

The logic shown in fig. 3-11 is what controls the memory storage and the tri-state control. The logic block takes ten inputs – two clocks for the channel data and the stop data, a read enable signal, a logic enable signal, reset, four SST data inputs and one stop data input. Two clocks are necessary since the channel data is clocked at a higher rate than the stop data from the SST. The gated logic as shown in the schematic is used to control the clock inputs. When data is being stored the channel data clock runs at 20 MHz and the stop data clock runs at 625 KHz, but when data is being read the clocks run at the SPI clock speed provided by the

MBED. The data inputs di_ch0, di_ch1, di_ch2 and di_ch3 are the four channels' data inputs from the ADCs, the stop_data_in is the data input from the SST stop data readout. There are numerous outputs in the logic block which are carryover outputs from the lower logic level and not all are necessary for the functioning of the logic but can be used for expanding the firmware functions or for testing the logic behavior. The outputs that are necessary are full_4, stop_full, do_1, do_2, do_3, do_4 and stop_data_out. The outputs full_4 and stop_full are used to determine when all the data from the ADCs and the SST has been collected, and subsequently send a flag signal to the MBED that the data is ready to be collected. The other outputs are the data outputs. The lower level logic makes sure that the data is readout in sequence so all the channels are put into an OR gate to collate the data into a single data stream. The tri-state logic is explained in 3.2.6. Fig. 3-12 shows the lower level logic structure of the memory control logic for the individual channels and fig. 3-13 shows how the individual channel logic is stitched together to form the overall storage system.

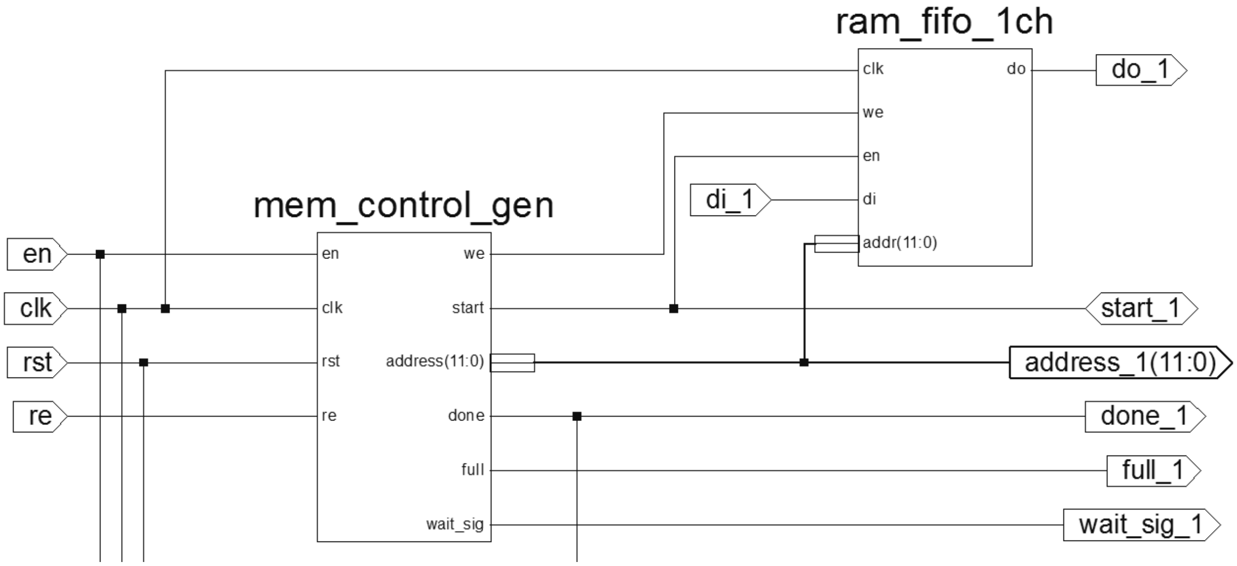


Figure 3-12: Memory logic for individual channels

The mem_control_gen logic block controls the memory instance ram_fifo_1ch which is configured as an instantiated block of block RAM. The control signals en, clk, rst, re as described earlier are passed onto the control logic block, which then generates address values and other intermediate control signals for the memory. The data input di_1 is the data from the first channel. The data from the ADC is 1 bit wide and 16 bits long with a total of 256 words of data per channel, making for a total data length of 4096 bits. The memory is configured as 1 bit wide and 4096 bits long. The addresses for each bit is generated by the control block. A full flag (full_1) is set once all the bits have been recorded to indicate as such.

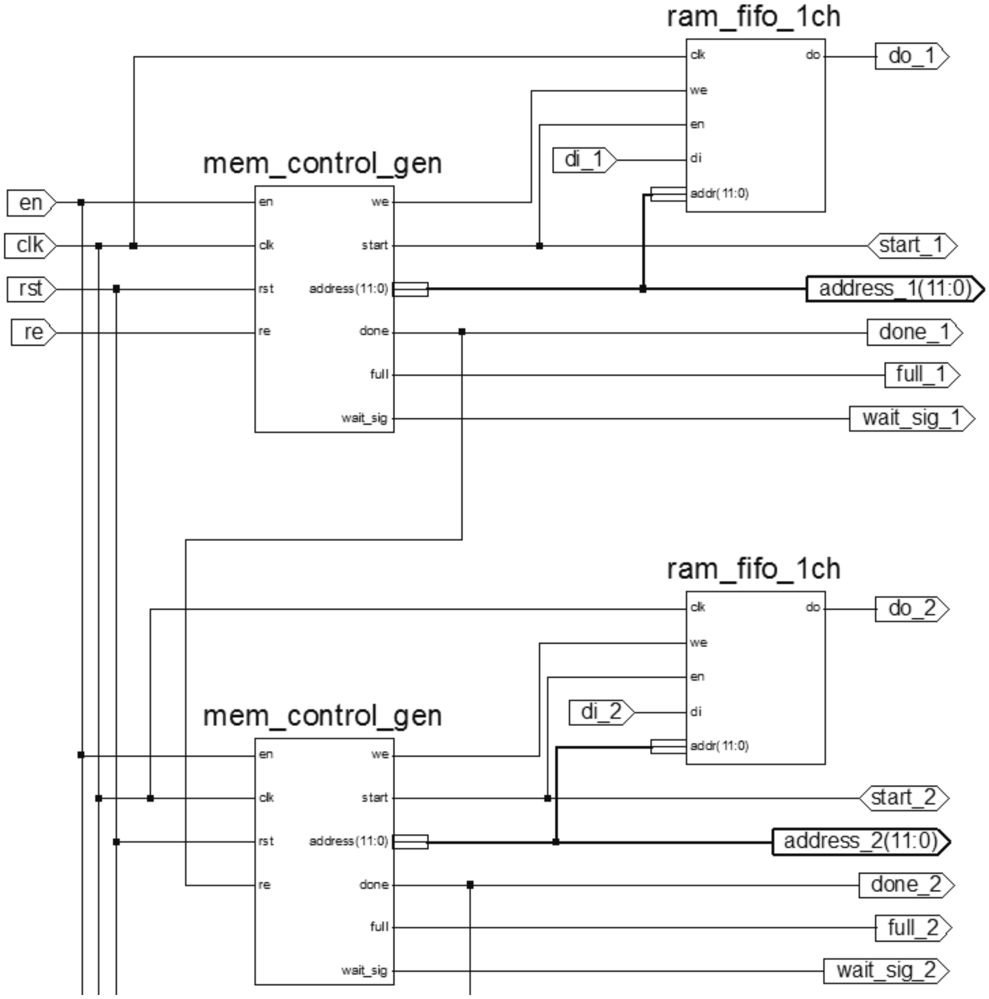


Figure 3-13: Memory logic interconnections

The control signals other than the read enable (re) is shared between the channels, this is so because the channels record data simultaneously. The second channel data input is shown as di_2. All the other channels and the stop data channel are all connected in a similar manner, with only the size and clock frequency of the stop data block being different to the ADC data. The stop data is 1 bit wide and 256 bits long, so the memory for the stop data is configured as such and the clock frequency is 625 KHz, same as the SST read clock.

The read process is different since the data needs to be sequenced. When the MBED is ready to read the data, it sets the read enable (re) HIGH and starts the SPI clock, which becomes the clock being fed to the memory system. The memory control starts counting the address from the beginning and the data is put on the output pin (do_1). Once the all the data has been read out, a done flag (done_1) is set which then becomes the read enable signal for the next channel, and the process is repeated for the next channel and so on for the other channels and the stop data. All the other channels also have the similar outputs which are also passed onto the higher logic level and can be used for testing purposes. The reset signal is used to reset all the control logic.

3.2.6 Tri-state control

The tri-state logic first shown in fig. 3-11 and again in fig. 3-14 is a necessary feature due to the way the FPGA uses the SPI interface. The SPI protocol lines are connected to the MBED and also to the SD card slot. When data is read out from the FPGA, it is simultaneously being written onto the SD card, however when data is not being read out, the FPGA keeps hold of the SPI line making it impossible for the MBED to communicate with the SD card.

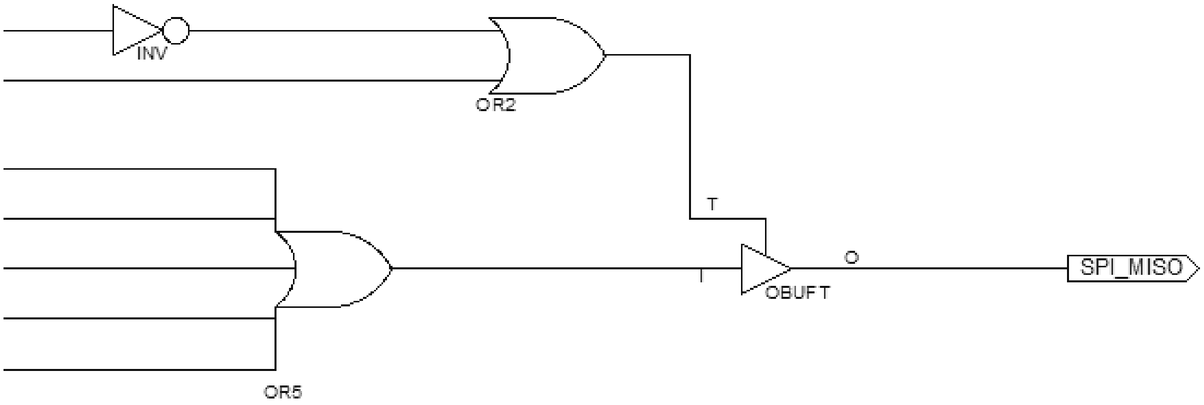


Figure 3-14: Tri-state control logic

The OBUFT logic shown in fig. 3-14 is a tri-state output buffer, which behaves as shown in table 3-1. The read enable signal, which stays LOW when data is not being read, is fed to the inverted input of the 2-input OR gate, so the output is tri-stated in all such conditions. The other input is connected to the last done flag from the memory system, so that as soon as data has been read out and the flag has been set, the output is also tri-stated and thus releasing the SPI line.

Table 3-1: Truth table of the OBUFT logic

Inputs		**Outputs**
T	**I**	**O**
1	X	Z
0	1	1
0	0	0

3.2.7 Heartbeat signal

The system features a fast pulse generator called as ‘heartbeat’ pulser. It was previously designed by Liang Zou and employs a simple edge detector mechanism, included here for completion of the design functions. The logic is shown in fig. 3-15.

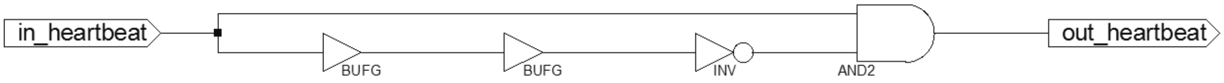


Figure 3-15: Heartbeat pulse generator

The input in_heartbeat is a regular square pulse from the MBED, which is fed into two paths – one is an unchanged copy of the input, while the other goes through a series of buffer delays and is then inverted. The resulting signals are then put to an AND gate and the output is given as the heartbeat pulse. The buffer delay determines the width of the pulse, and the inverting and then AND operation forms the edge detection operation.

Chapter 4: Test Results

4.1 Setting up the board

The board has a number of hardware configuration and tuned voltages. These are shown in fig 4-1, also shown are the easiest test probe locations.

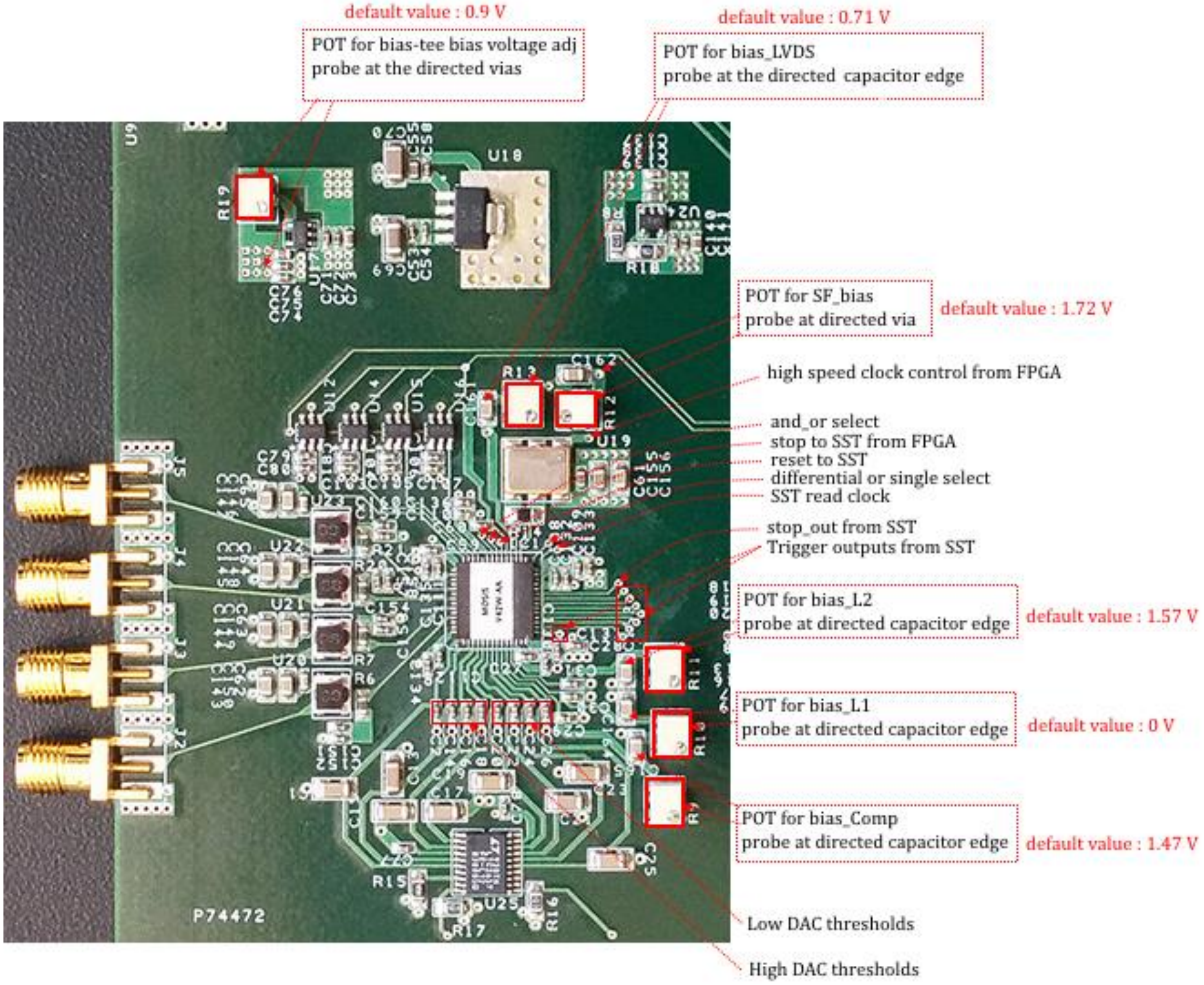


Figure 4-1: Bias voltage configurations and test probe locations

The bias voltages are set by setting the potentiometers as shown. The bias voltages are critical for stable SST functioning. The thresholds are set from the DAC through the MBED.

The high thresholds are set at approx. 1.1 V and the low threshold is set 400 mV. Only thermal triggers are used and the SST is set to trigger if either thresholds are crossed.

4.2 Test results

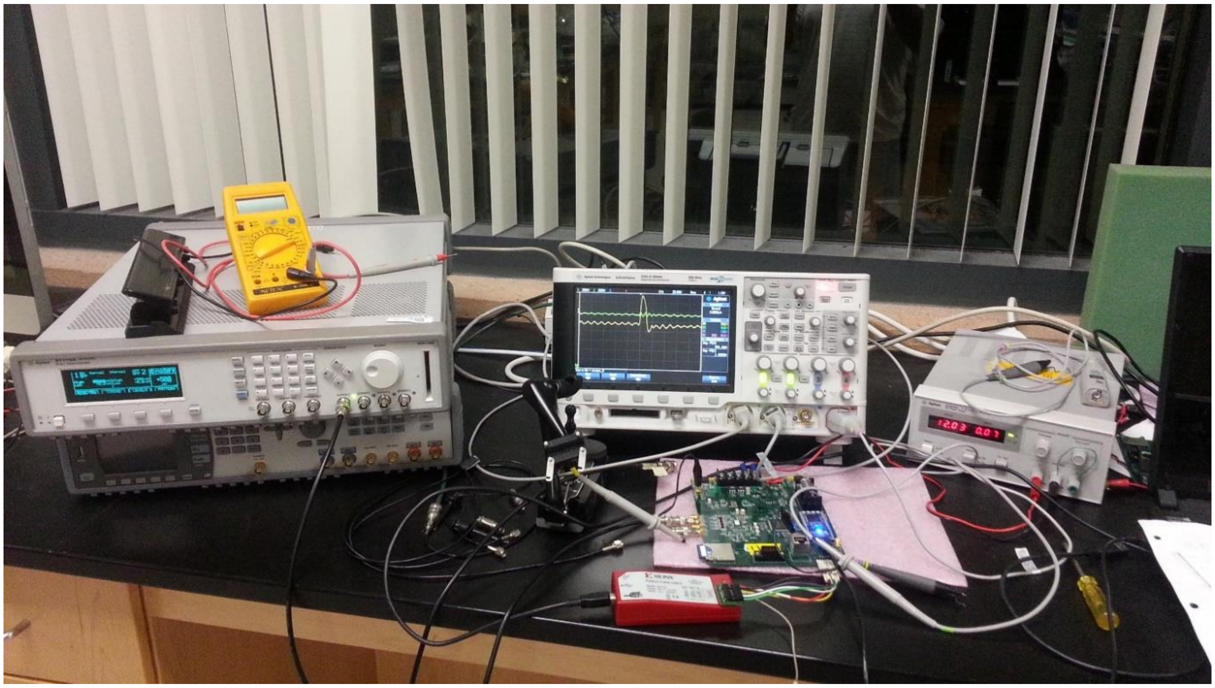
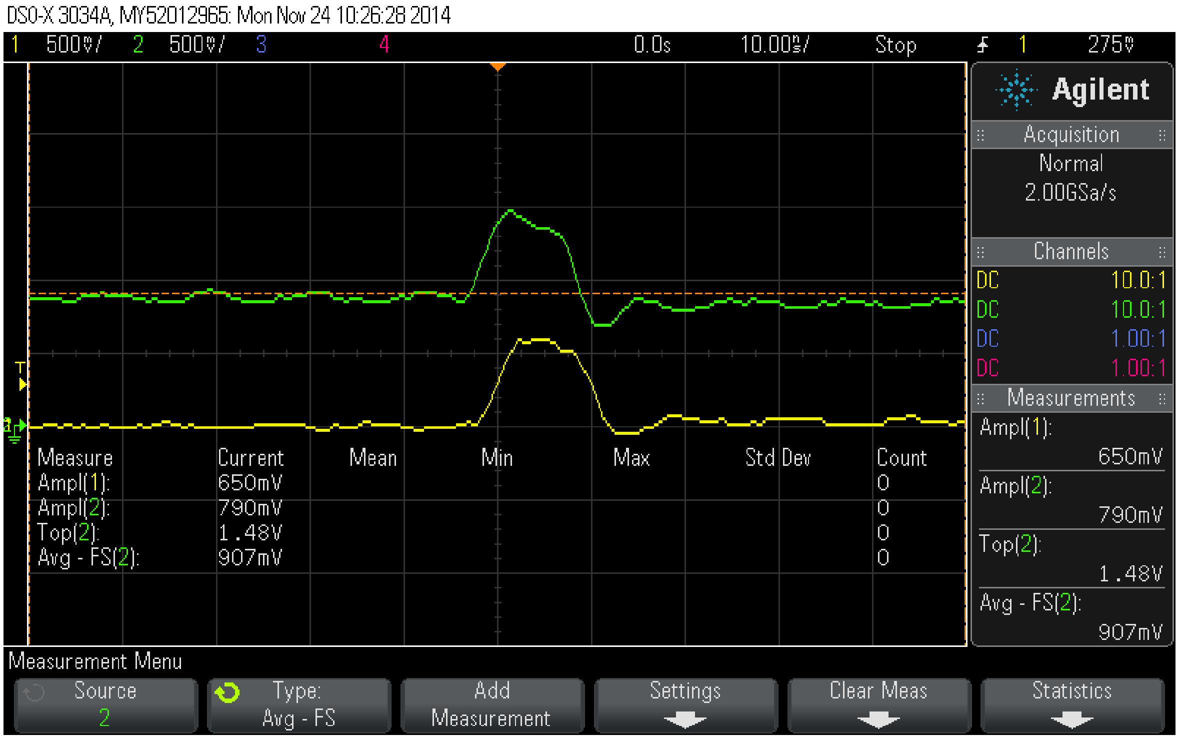


Figure 4-2: Test setup

The basic hardware test setup consists of a pulse generator, an oscilloscope, a multimeter and a power supply. The basic functions and the signals can be checked in this setup. More detailed testing and analysis on the collected data and functions has been done by other members of the ARIANNA collaboration.

The analog signal after the amplifier stage is put to a bias-tee which shifts the signal by the DC power injected. The SST works at the bias voltage of 900 mV. This function is shown in the oscilloscope screen in fig. 4-2. The input signal is generated by a pulse generator and the signal path is probed before and after the bias-tee.



DS0-X 3034A, MY52012965: Mon Nov 24 10:26:28 2014

Measure	Current	Mean	Min	Max	Std Dev	Count
Ampl(1):	650mV					0
Ampl(2):	790mV					0
Top(2):	1.48V					0
Avg - FS(2):	907mV					0

Acquisition
Normal
2.00GSa/s

Channels

DC	10.0:1
DC	10.0:1
DC	1.00:1
DC	1.00:1

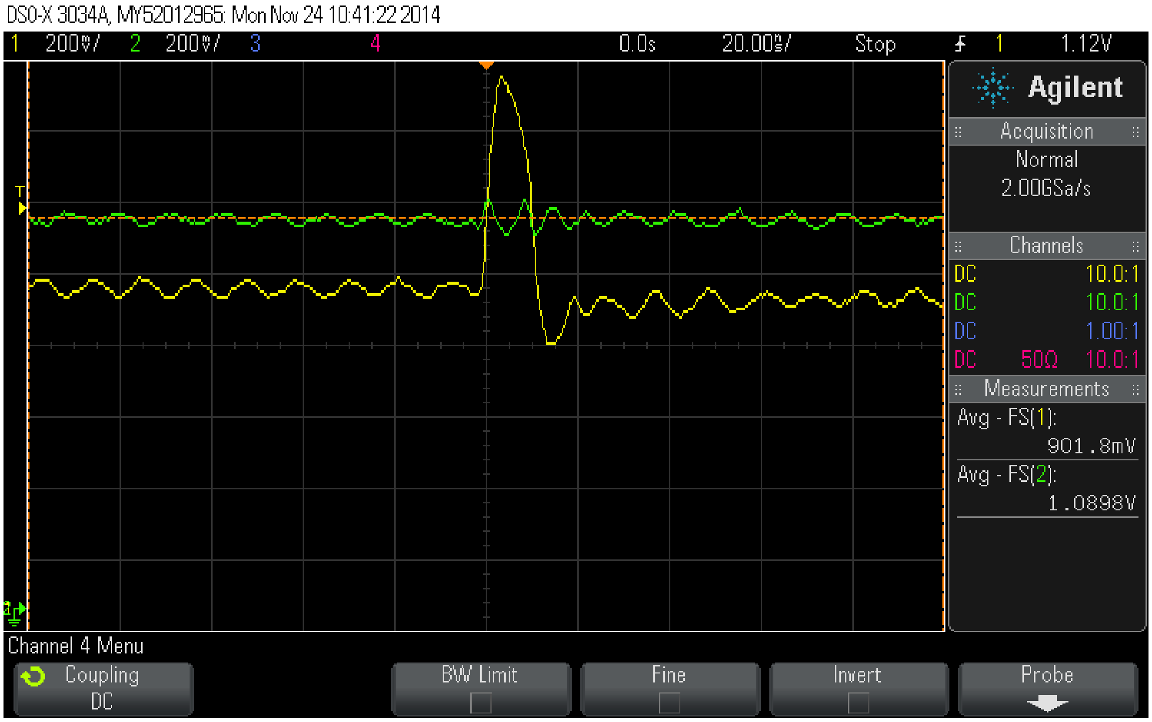
Measurements
Ampl(1): 650mV
Ampl(2): 790mV
Top(2): 1.48V
Avg - FS(2): 907mV

Measurement Menu

Source: 2
Type: Avg - FS

Figure 4-3: Input (yellow) and bias-tee shifted signal (green)

The shifted signal (in green) is seen to have the average at approx. 900 mV. The slight variation is due to noise. The signal is then sampled by the SST and checked against set thresholds. If a signal crosses the thresholds and depending on the trigger signal modes, differential or single ended, the SST then sends out a trigger signal. The input signal and the high threshold is shown in fig. 4-4. Both plots are at the same voltage range, and shows the input crossing the threshold.



DSO-X 3034A, MY52012965: Mon Nov 24 10:41:22 2014

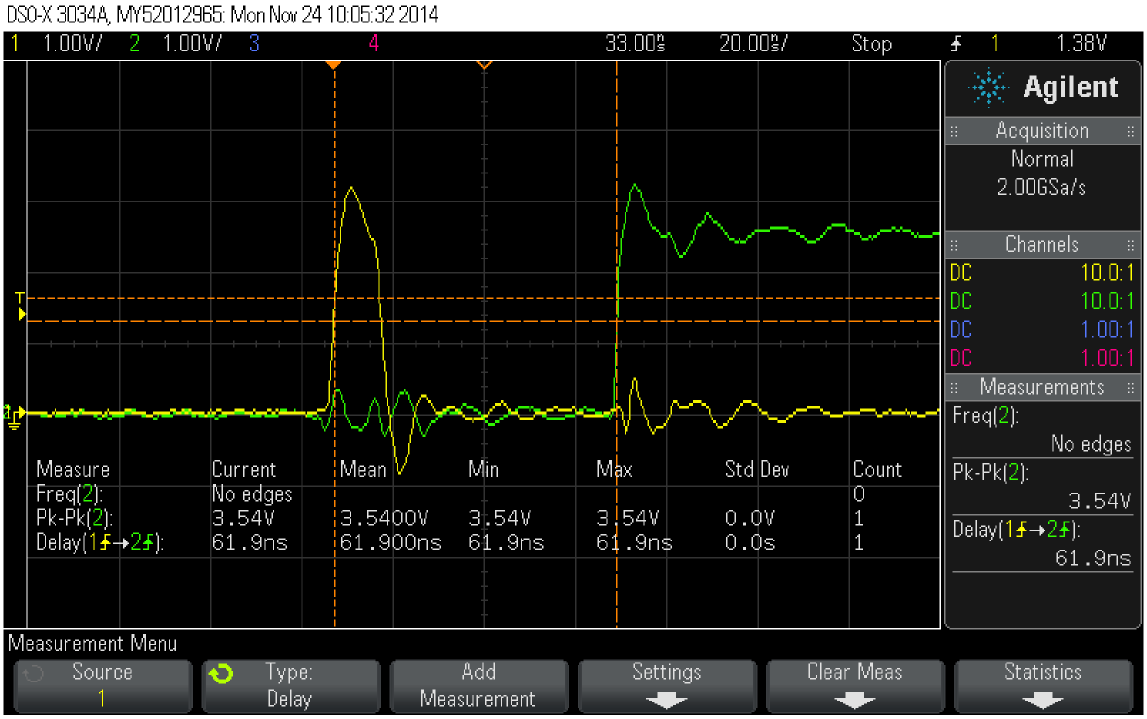
```
1  200㎷/   2  200㎷/   3   4        0.0s    20.00ª/    Stop   ⨍  1   1.12V
```

Acquisition: Normal, 2.00GSa/s

Channels:
DC 10.0:1
DC 10.0:1
DC 1.00:1
DC 50Ω 10.0:1

Measurements:
Avg - FS(1): 901.8mV
Avg - FS(2): 1.0898V

Figure 4-4: Input signal (yellow) crossing the HIGH threshold (green)



DSO-X 3034A, MY52012965: Mon Nov 24 10:05:32 2014

```
1  1.00V/   2  1.00V/   3   4        33.00ª    20.00ª/    Stop   ⨍  1   1.38V
```

```
Measure          Current    Mean        Min      Max      Std Dev   Count
Freq(2):         No edges                                           0
Pk-Pk(2):        3.54V      3.5400V     3.54V    3.54V    0.0V      1
Delay(1⨍→2⨍):    61.9ns     61.900ns    61.9ns   61.9ns   0.0s      1
```

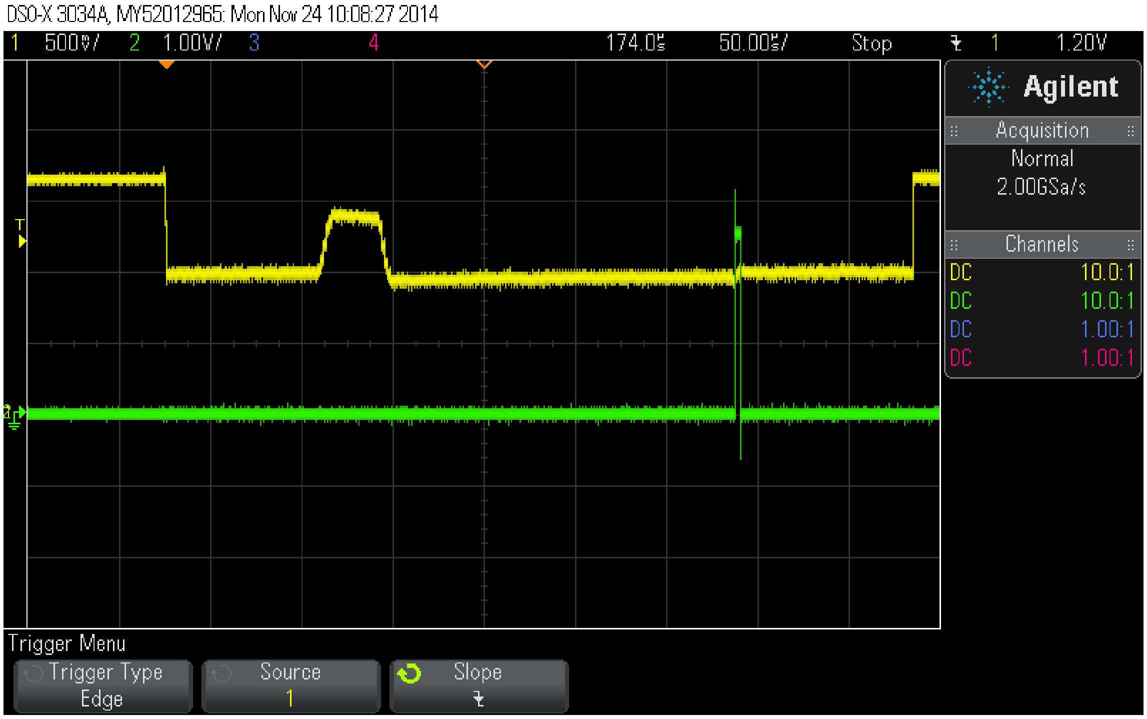
Acquisition: Normal, 2.00GSa/s

Channels:
DC 10.0:1
DC 10.0:1
DC 1.00:1
DC 1.00:1

Measurements:
Freq(2): No edges
Pk-Pk(2): 3.54V
Delay(1⨍→2⨍): 61.9ns

Figure 4-5: Trigger (yellow) to Stop signal (green) delay

The local UCI radio station KUCI 88.9 results in a noise of approximately 88.9 MHz being picked up by the DSO scope. The trigger generated by the SST is then sent to the FPGA which decides whether to stop the SST depending on majority logic configuration. It adds a programmable delay as explained in chapter 3. This delay is shown in fig. 4-5. Once the SST has been stopped, the FPGA issues the SST read clock, which then starts to read out the analog values stored in the SST.



DSO-X 3034A, MY52012965: Mon Nov 24 10:08:27 2014

1 500mV/ 2 1.00V/ 3 4 174.0µs 50.00µs/ Stop 1 1.20V

Acquisition: Normal 2.00GSa/s

Channels:
DC 10.0:1
DC 10.0:1
DC 1.00:1
DC 1.00:1

Trigger Menu: Trigger Type Edge, Source 1, Slope

Figure 4-6: Analog SST output (yellow) with stop data bits (green)

The yellow plot in fig. 4-6 shows the analog pulse as it appears at the SST output. The stop bits shown in green are bits indicating the location where the last of the 256 SST samples are read out. This indicates the end of the SST data and is two bits long. The data after the stop bits are values from the previous sampling and are to be discarded during data analysis.

The ADCs then digitize the data and stores the data in the FPGA. The data is then collected by the MBED using the SPI interface. The MBED sends out the SPI clock which then clocks out the data. This operation is shown fig. 4-7. The displayed signal resolution is limited by the scope sampling resolution resulting in the jagged appearance.

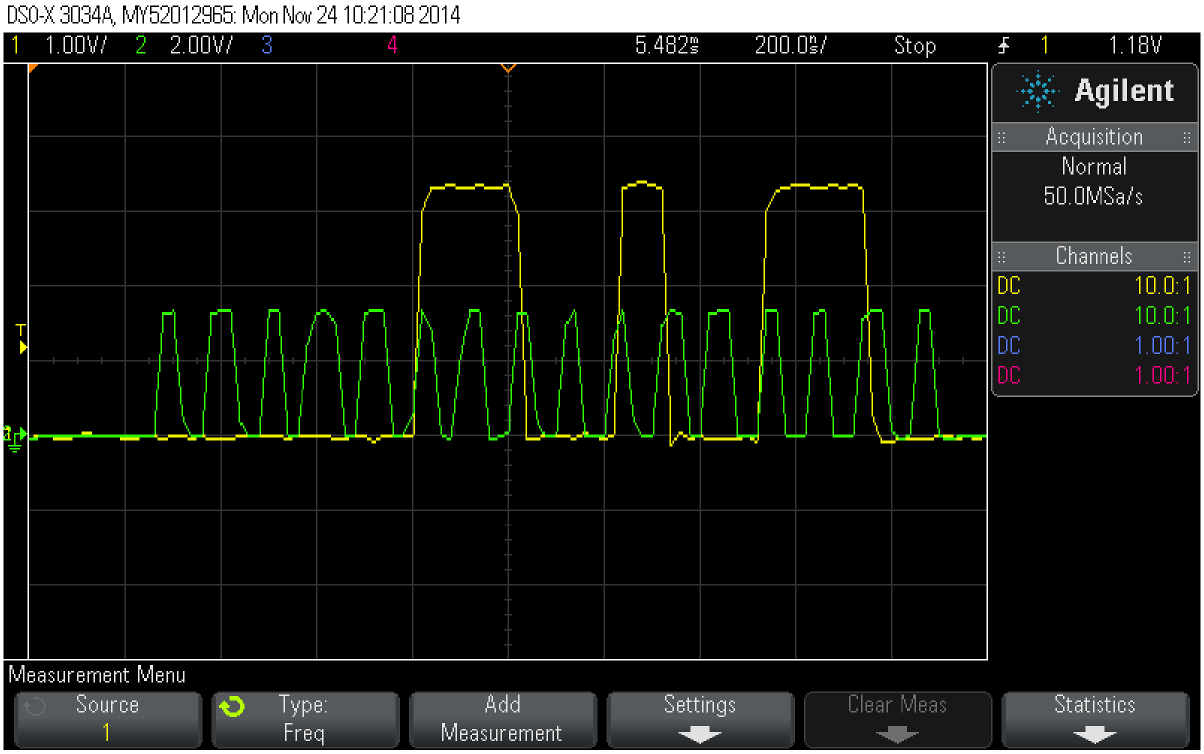


Figure 4-7: Data output at the SPI line

The address is propagated at the rising edge of the clock (green) and the data (yellow) is read at the falling edge, thus the data interpreted from the above plot is 0000 0110 0100 1100. The leading 0s are discarded since they are the dummy 0s introduced by the ADC. The actual 12 bits that form the data is 0110 0100 1100. The above figure represents one sample of the data from a channel without an input and thus represents the fixed pattern noise (FPN) which was found to be stable in further studies by Corey Reed as shown in fig. 4-8.

4.3 Further Test Results

A number of detailed tests and analysis were performed by others in the collaboration. An analysis on the data allows for a detailed study of the data and to determine any major noise sources. FPN characterization was done by Corey Reed, and the system displayed excellent FPN performance.

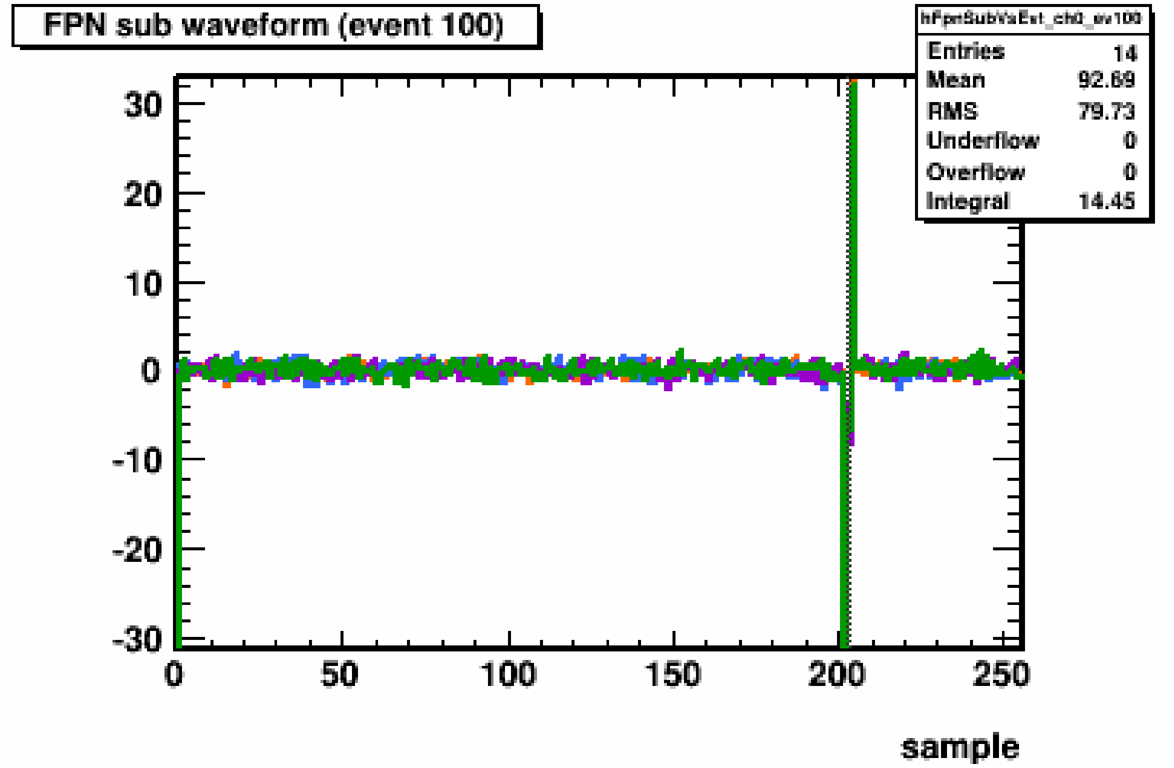


Figure 4-8: FPN subtracted data seen to be stable. Plot generated by Corey Reed.

The spike around sample 200 represents the stop bits. FPN represents the inherent signal pattern if a DC signal is applied at the input of the sensing system. A stable FPN allows for a simple FPN subtraction, and indicates a well-designed system, making it easy to analyze the data, since FPN calibration is not necessary or is straightforward.

Another important study to determine the stability of the system was a study of the stop bits. The stop bits as explained earlier indicates where the data ends for a particular sample. The presence of stop bits is extremely important for the integrity of the collected data. A detailed study on the stop bits was performed by Chris Persichilli.

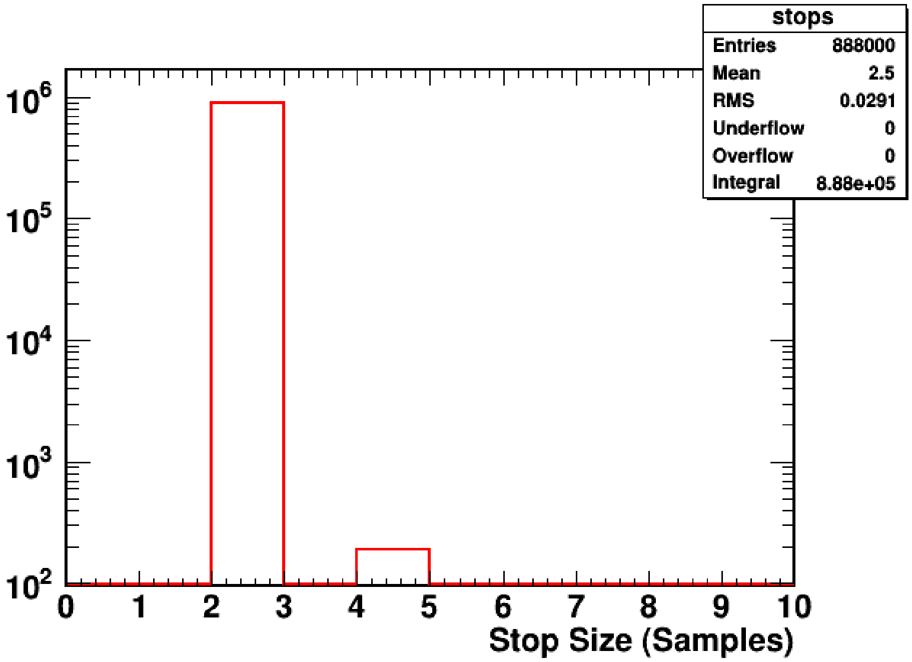


Figure 4-9: Long term study of stop bits. Plot generated by Chris Persichilli.

The board was set to trigger on forced triggers from the MBED and data was collected over a long time frame. A total of 888000 events were collected and the above plot shows the histogram for all the stop bits. The study found that sometimes the stop bits are 4-bits long but most are 2-bits long, which is the standard stop length. The 4-bits long stops are an expected observation and do not indicate errors, they occur due to the stop pointer slipping from clock synchronization errors. This results in two data bits being overwritten with the stop bit. A 4-bit stop event is four orders of magnitude less likely to occur than a proper 2-bit stop event as can be seen from the plot, making it an extremely rare occurrence.

Chapter 5: Conclusion

A new high speed, four channel DAQ system was designed and fabricated. The new system represents a lot of new features and improvements over the previous system, in line with the requirements of the ARIANNA project. The new SST chip allows for a single board system as opposed to the previous generation's daughter card and motherboard setup. The new system features 4 channels, 256 samples per channel with almost double the analog bandwidth as the previous system. It also features a lower FPN, fully synchronous clocked performance. The new system board also represents major improvements over the previous system with typical power usage reduced by a factor of 3 approx. It introduces onboard digital temperature measurement and markedly more accurate voltage and power monitoring. It also features much improved system robustness with integrated voltage converters, with more robust components, simpler firmware and software all while maintaining compatibility with the old systems. A number of the new systems are set to be deployed in the field in the 2014 deployment season and represents a candidate prototype for full scale deployment.

References

[1] Barwick S W 2005 "Exploratory Measurement of Attenuation and Reflection Properties of the Ross Ice Shelf at Radio Frequencies between 100 MHz and 1 GHz", Proposal to the National Science Foundation SGER Program

[2] Barwick S W 2007 J. Physics, ARIANNA: A New Concept for UHE Neutrino Detection

[3] Greisen K 1966 Phys. Rev. Lett. 16 748; Zatsepin G T and Kuzmin V A 1966 JETP Lett. 4 78

[4] Barwick, S., for ARIANNA Collaboration, 2011, ICRC proc., ARIANNA: A New Concept for High Energy Neutrino Detection

[5] G. A. Askaryan, JETP 14, 441 (1962); 21, 658 (1965).

[6] S. A. Kleinfelder, E. Cheim, T. Prakash, "The SST Fully-Synchronous Multi-GHz Analog Waveform Recorder with Nyquist-Rate Bandwidth and Flexible Trigger Capabilities," Conference Record of the 2003 IEEE Nuclear Science Symposium, 2014, in press.

[7] S. A. Kleinfelder for the ARIANNA Collaboration, "Design and Performance of the ARIANNA Hexagonal Radio Array Systems", 2014, in press

[8] "First Principles of a Gas Discharge Tube (GDT) Primary Protector" Rev.2, Bourns

[9] IPC-2141A (2004), "Design Guide for High-Speed Controlled Impedance Circuit Boards."

[10] Hanson, J., for the ARIANNA Collaboration. "Ross Ice Shelf Thickness, Radio-frequency Attenuation and Reflectivity: Implications for the ARIANNA UHE Neutrino Detector." Proceedings of the 2011 International Cosmic Ray Conference, Beijing, China.

[11] S. A. Kleinfelder, "A multi-GHz, multi-channel transient waveform digitization integrated circuit," IEEE Nuclear Science Symposium Conference Record, 2002.

[12] S. A. Kleinfelder, S. W. Chiang, W. Huang "Multi-GHz Synchronous Waveform Acquisition With Real-Time Pattern-Matching Trigger Generation" IEEE Tran. On Nuclear Science, Vol. 60, No. 5, 2013

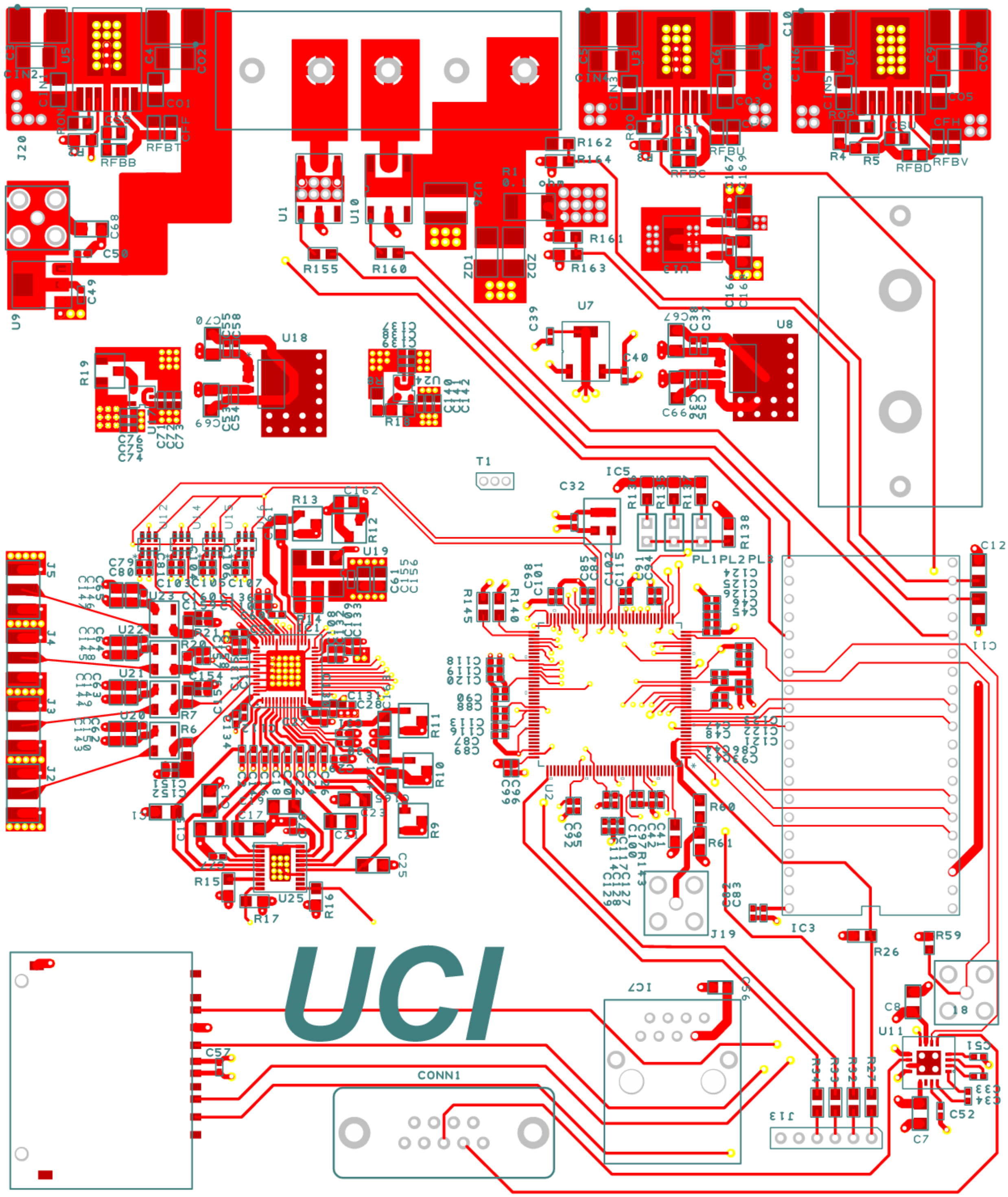
[13] C. Reed, et al., "Performance of the ARIANNA Neutrino Telescope Station," Proc. Int. Cosmic Ray Conf., Rio De Janeiro, July 2013.

[14] M. Roumi, "Advanced Pattern-Matching Trigger System for the ARIANNA High-Energy Neutrino Detector," Dissertation, U.C. Irvine, 2014.

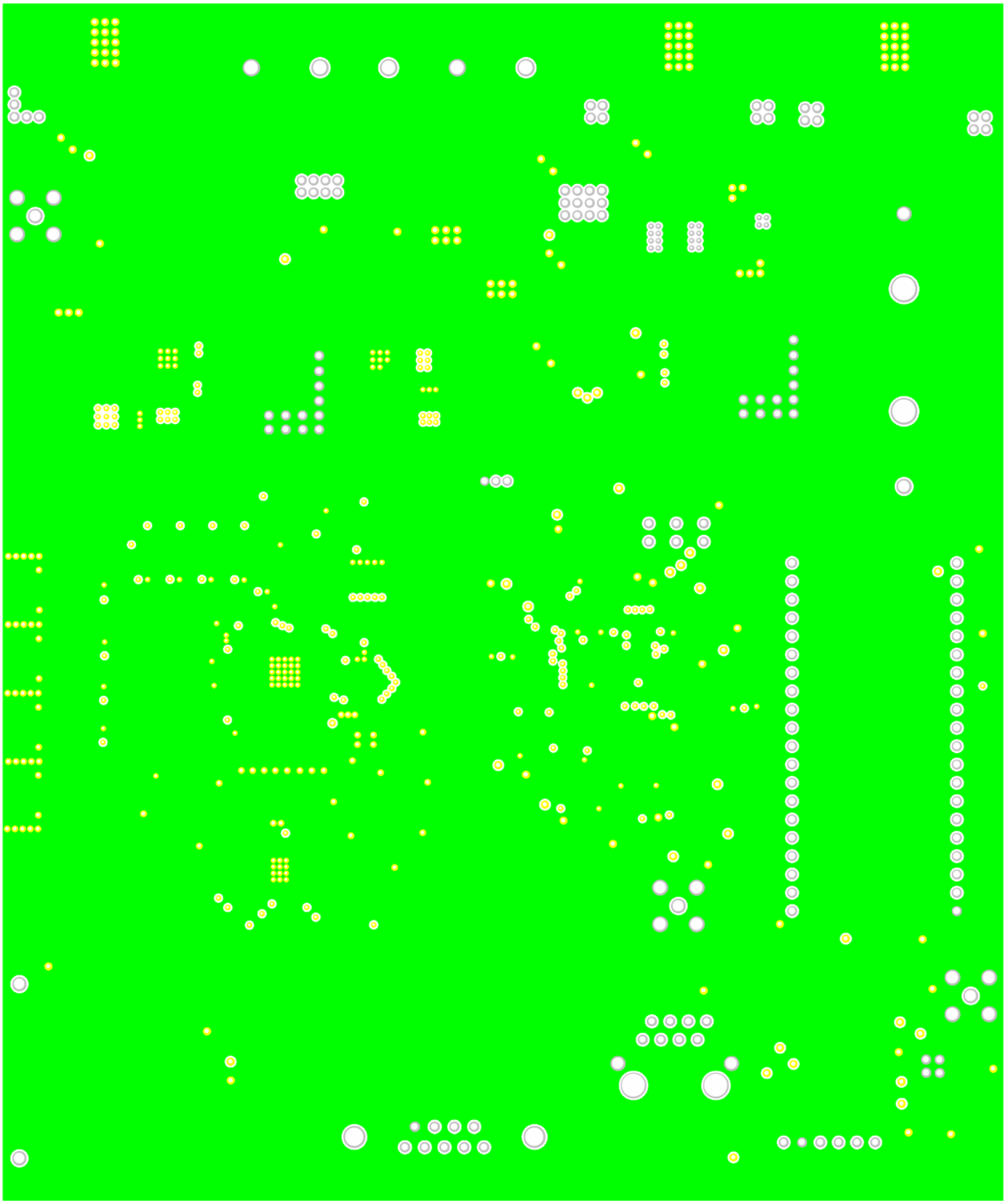
[15] Liang Zou, "A high speed and multichannel Data Acquisition System for the ARIANNA High Energy Neutrino Detector," Dissertation, U.C.Irvine, 2013.

Appendix A: Motherboard Layout

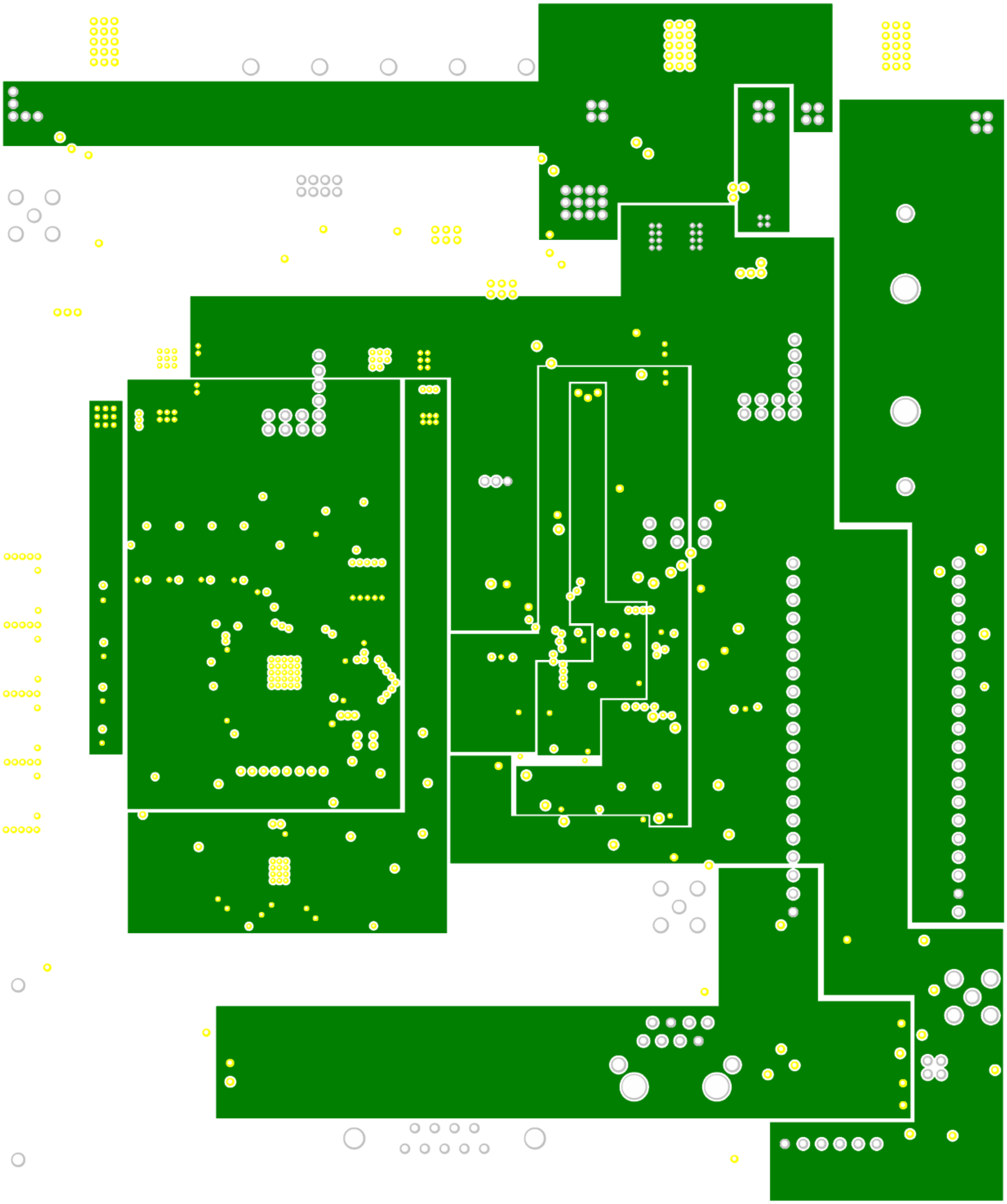
Layer: Top Copper with Silkscreen



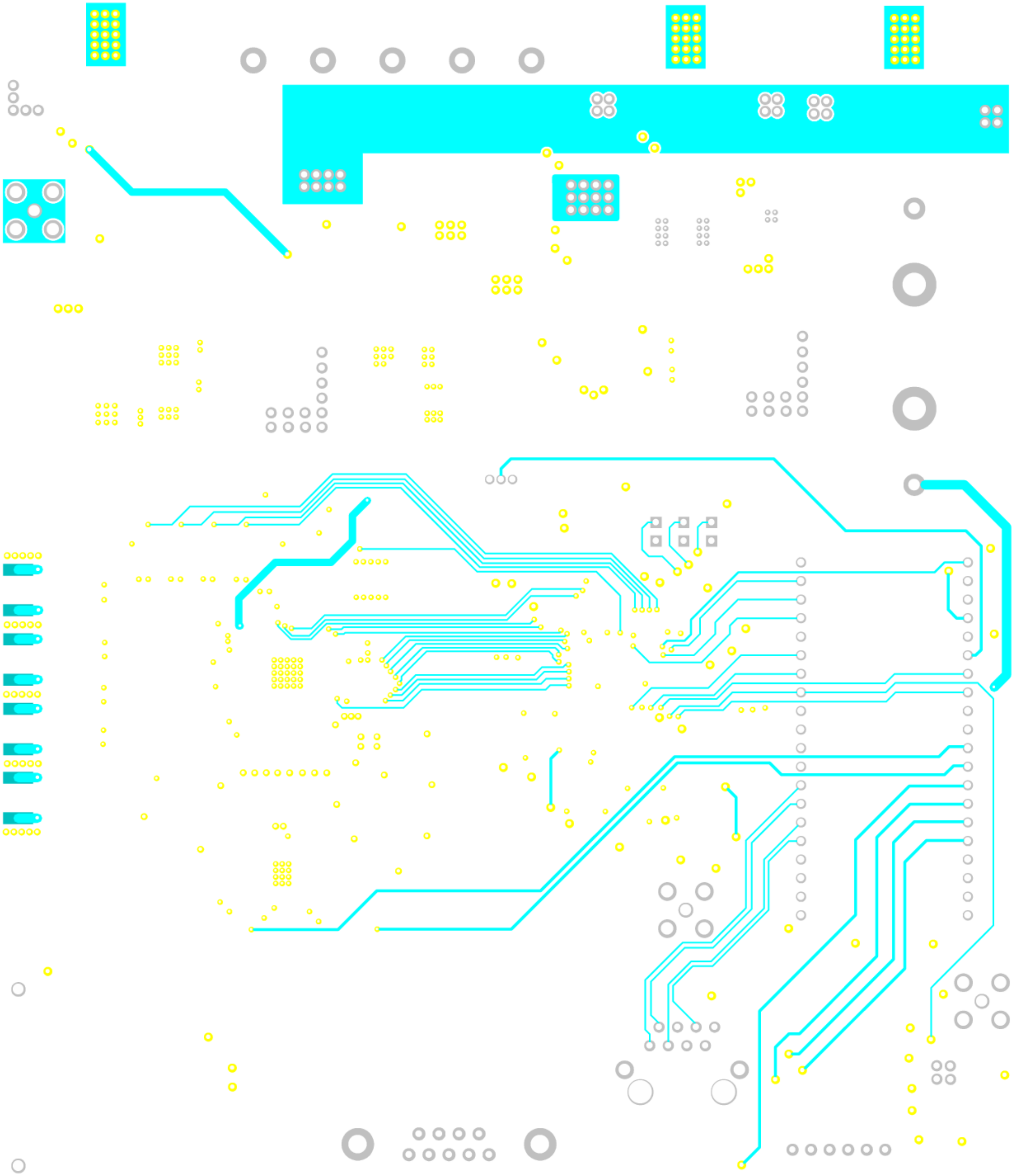
Layer: Ground Layer



Layer: Power Layer



Layer: Bottom Copper



Appendix B: Source Code

Block: CS_gen.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity CS_gen is
Port (
-- en is the chip enable signal
-- clk_1M is the slower clock
-- clk_20M is the faster clock
-- rst is the reset signal from the MBED
-- ADC_clk is the ADC read clock output to the ADC
-- SST-rck is the SST read clock output to the SST
-- CS_n is the active low chip select to the ADC
-- RAM_sync is the memory synchronizing signal
        en : in  STD_LOGIC;
        clk_1M : in  STD_LOGIC;
        clk_20M : in  STD_LOGIC;
        rst : in  STD_LOGIC;
        ADC_clk : out  STD_LOGIC;
        SST_rck : out STD_LOGIC;
        CS_n : out  STD_LOGIC;
        RAM_sync : out  STD_LOGIC);
end CS_gen;

architecture Behavioral of CS_gen is
-- declaring local signals used in the code for conditioning
        signal enable1 : STD_LOGIC := '0';
        signal enable2 : STD_LOGIC := '0';
begin
-- use the chip enable signal to condition a switching signal
enable1_proc : process(clk_1M, en, rst) begin
        if rst = '1' then
```

```
              enable1 <= '0';
        else
              if en = '1' then
                if rising_edge(clk_1M) then
                        enable1 <= '1';
                 end if;
              else
                enable1 <= '0';
              end if;
        end if;
end process;

-- use the first level switching signal to generate another switching signal
enable2_proc : process(clk_1M, enable1, rst) begin
        if rst = '1' then
              enable2 <= '0';
        else
              if enable1 = '1' then
                if rising_edge(clk_1M) then
                        enable2 <= '1';
                end if;
              else
                enable2 <= '0';
              end if;
        end if;
end process;

-- signals to start at the first switching signal
SST_signal_gen : process (enable1, rst) begin
        if rst = '1' then
              CS_n<= '1';
              SST_rck <= '1';
              ADC_clk <= '0';
```

```
    else
        if enable1 = '0' then
            CS_n<= '1';
            SST_rck <= '1';
            ADC_clk <= '0';
        else
            SST_rck <= clk_1M;
            CS_n<= clk_1M;
            ADC_clk <= clk_20M;
        end if;
    end if;
end process;

-- signals to start at the second switching signal
Others_signal_gen : process (enable2, rst) begin
    if rst = '1' then
       RAM_sync <= '0';
    else
       if enable2 = '0' then
        RAM_sync <= '0';
       else
       RAM_sync <= not clk_1M;
       end if;
    end if;
end process;
end Behavioral;
```

Block: clkgen_1M.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity clkgen_1M is
 Port (
-- clk_in is the input clock
-- clk_out is the output clock
                  clk_in : in  STD_LOGIC;
                  clk_out: out STD_LOGIC);
end clkgen_1M;

architecture Behavioral of clkgen_1M is
-- declaring local signals for counting and signal conditioning
   signal temporal: STD_LOGIC := '0';
   signal counter : integer range 0 to 15 := 0;
begin
-- count the input clock 16 times and then switch, giving rise to
-- a divide by 32 clock as output
   frequency_divider: process (clk_in,counter,temporal) begin
      if rising_edge(clk_in) then
         if (counter = 15) then
            temporal <= NOT(temporal);
            counter <= 0;
         else
            counter <= counter + 1;
         end if;
      end if;
   end process;
      clk_out <= temporal;
end Behavioral;
```

Block: stop_latch.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity stop_latch is
Port (
-- trigger_in is the incoming trigger signal
-- clk is the clock signal to which the code is synchronized
-- rst is the reset signal to clear the latch
-- stop is the stop signal to the SST to stop sampling
        trigger_in : in  STD_LOGIC;
        clk : in  STD_LOGIC;
        rst : in  STD_LOGIC;
        stop : out  STD_LOGIC);
end stop_latch;

architecture Behavioral of stop_latch is
-- local signal for signal switching
        signal enable_stop : STD_LOGIC := '0';
begin
-- as soon as a rising edge is detected in the trigger
-- the stop is generated
stop_gen : process (rst, trigger_in) begin
        if rst = '1' then
              enable_stop <= '0';
        else
              if rising_edge (trigger_in) then
                enable_stop <= '1';
              end if;
        end if;
end process;

-- the generated stop signal is passed onto the output
stop_proc : process (enable_stop) begin
        stop <= enable_stop;
end process;
end Behavioral;
```

Block: mem_control_gen.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.all;
entity mem_control_gen is
Port (
-- en is the enable signal
-- rst is the reset signal
-- re is the read enable signal to determine whether to read or write
-- start is an intermediate in/out signal for waiting on input
-- clk is the system clock
-- address is 12 bit integer output used by the memory
-- full is a flag signal to indicate memory full
-- done is a flag signal to indicate process completion
-- wait_sig is used to throttle the memory when waiting
-- we is write enable output used by the memory
        en : in STD_LOGIC;
        rst : in  STD_LOGIC;
        re : in  STD_LOGIC;
        start : inout  STD_LOGIC;
        clk : in STD_LOGIC;
        address : out  STD_LOGIC_VECTOR (11 downto 0);
        full : out  STD_LOGIC;
        done : out  STD_LOGIC;
        wait_sig : out  STD_LOGIC;
        we : out STD_LOGIC);
end mem_control_gen;

architecture Behavioral of mem_control_gen is
-- local signals used to condition and generate the outputs
        signal addr : std_logic_vector (11 downto 0) := "000000000000";
        signal full_f : std_logic := '0';
        signal done_f : std_logic := '0';
        signal wait_f : std_logic := '0';
begin

-- count to 4095 from 0 and then wait until read enable is set
-- upon read enable being set
-- count to 4095 from 0 and then wait until reset
main_proc : process(clk, rst, re, start, en, full_f, done_f) begin
        if rst = '1' then
              addr <= "000000000000";
              full_f <= '0';
              done_f <= '0';
              wait_f <= '0';
        else
              if falling_edge(clk) then
                if en = '1' then
                        if start = '1' then
                                case re is
                                  when '0' => if addr < "111111111111" then -- 4095
                                                addr <= addr + 1;
                                                else
                                                        addr <= "000000000000";
                                                        full_f <= '1';
                                                        wait_f <= '1';
                                                end if;
                                  when '1' => if addr < "111111111111" then -- 4095
                                                addr <= addr + 1;
                                                else
                                                        addr <= "000000000000";
                                                        done_f <= '1';
                                                        wait_f <= '1';
                                                end if;
                                  when others => null;
                                  end case;
```

```
       else
        addr <= "000000000000";
            if re = '0' and full_f = '1' then
                wait_f <= '1';
            elsif re = '1' and done_f = '1' then
                wait_f <= '1';
            else
                wait_f <= '0';
            end if;
            end if;
        else
            wait_f <= '1';
        end if;
       end if;
    end if;
end process;

-- passign the conditioned signals onto the outputs
sig_gen : process(addr, full_f,done_f,wait_f,re) begin
    address <= addr;
    full <= full_f;
    done <= done_f;
    wait_sig <= wait_f;
    start <= not wait_f;
    we <= not re;
end process;

end Behavioral;
```

Block: ram_fifo_1ch.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_unsigned.all;
entity ram_fifo_1ch is
Port (
-- clk is the system clock
-- we is the write enable signal for the memory
-- en is an enable signal for the entire memory unit
-- di is the data input
-- do is the data output
-- addr is the 12 bit address integer
        clk : in  STD_LOGIC;
        we : in std_logic;
        en : in std_logic;
        di : in  STD_LOGIC;
        do : out  STD_LOGIC;
        addr : in  STD_LOGIC_VECTOR (11 downto 0));
end ram_fifo_1ch;

architecture Behavioral of ram_fifo_1ch is
-- declaring the RAM type to enforce blockRAM usage
type ram_type is array (0 to 4095) of std_logic;
signal RAM: ram_type;
attribute ram_style: string;
attribute ram_style of RAM : signal is "block";
begin
-- the data is sampled at the rising clock edge
-- and put onto the output always
-- data is written onto the memory only if the we control is high
-- otherwise outputs a '0' if the memory is OFF
-- but the the memory is clocked. This is done to avoid
-- unknown memory output conditions
-- generates a read_first memory as can be checked from the summary report
process (clk, en, we) begin
        if rising_edge(clk) then
              if en = '1' then
                do <= RAM(conv_integer(addr));
                if we = '1' then
                        RAM(conv_integer(addr)) <= di;
                end if;
              else
                do <= '0';
              end if;
        end if;
end process;

end Behavioral;
```