Microwave SQUID multiplexer Firmware Specification

Justus A. Brevik

1 introduction

The plan is to take the latest stable version of the ARCONS firmware (chan_snap_v3.mdl) and adapt it for the readout of the microwave SQUID resonator with flux ramp modulation. The ARCONS firmware is channelized by a digital down converter (DDC) and allows for the readout of 256 channels at 500 kHz bandwidth per channel. This high per-channel bandwidth is compatible with the 100–150 kHz modulation frequency of flux ramp modulation. The ARCONS firmware was designed to detect and analyze pulses in the timestream data, as described in section **??**. This step is incompatible with the microwave SQUID readout and will be replaced by a step that demodulates the flux ramp response of the SQUIDs and then outputs timestreams at a rate of around 20–50 kHz, given by the rate of the flux ramp signal.

For an overview of the hardware and an explanation of the MUSIC firmware for resonator readout please see: MUSIC description. For an overview of the ARCONS system please see: ARCONS description

2 firmware modification steps

I've broken the development of the firmware into several steps that add new functionality necessary for the implementation of flux ramp modulation. After each step is implemented

below the compiled firmware (.bof) file can be sent to me for immediate testing and feedback. The third step in section **??** is a first pass at defining the flux ramp demodulation algorithm and will probably be modified in the future.

In addition to the work described below, there seem to be a few timing issues that the ARCONS team has ignored while compiling their firmware. If possible it would be great if these could be properly sorted out.

2.1 IQ center

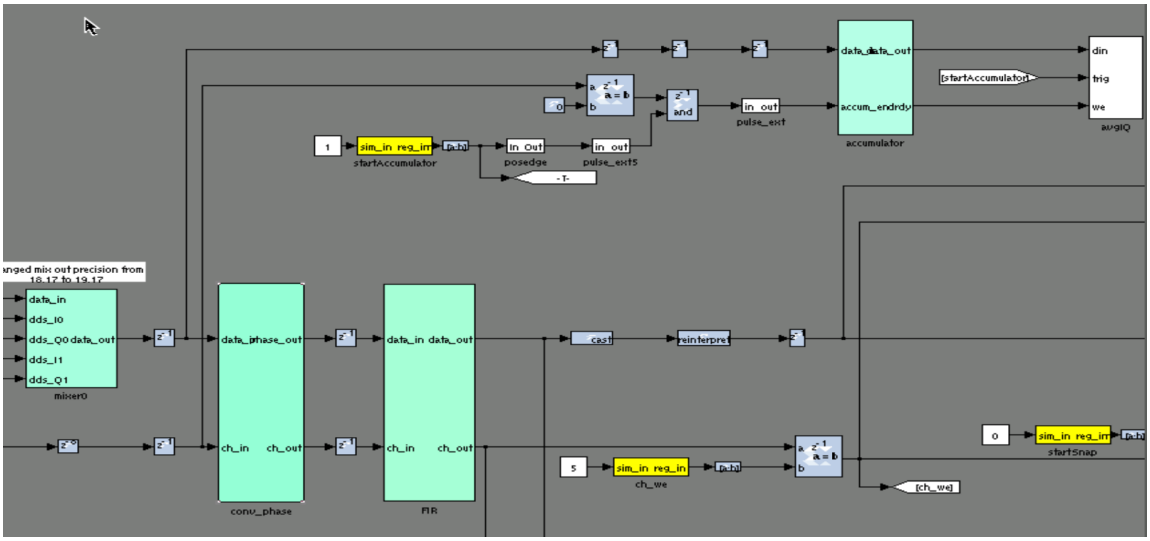

Figure 1: Portion of the chan_snap_v3 firmware that accumulates and outputs averaged IQ data (top) and the "conv_phase" block that centers IQ data and calculates the phase from I and Q (bottom left).

The first step is a fairly small one and should allow the firmware developer to become familiar with the ARCONS firmware and the blocks it uses. In Figure **??** the portion of the firmware that accumulates I and Q timestreams is shown at the top of the figure leading in to the blocks labelled "accumulator" and "avgIQ". The data saved to this register is used

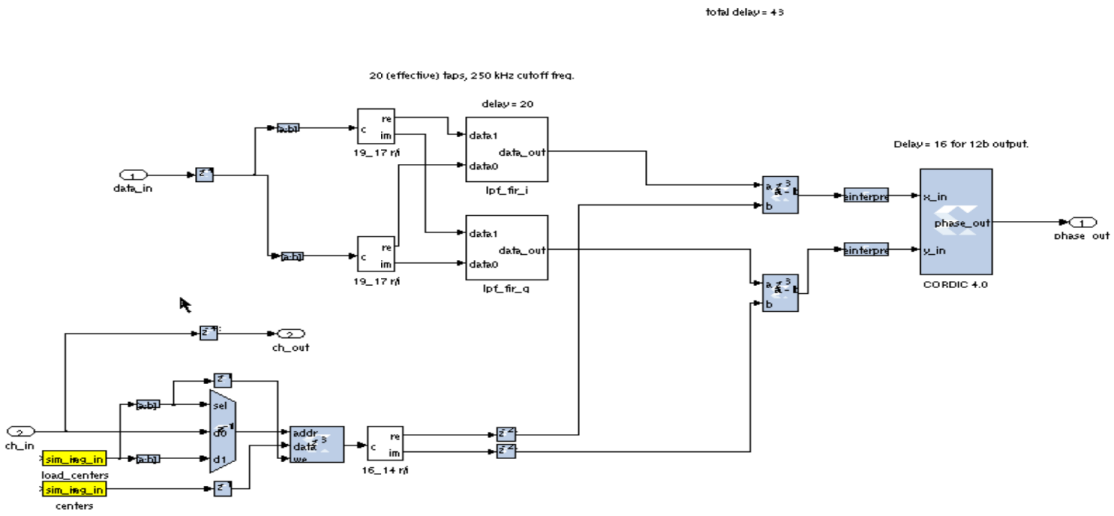

Figure 2: Inside the "conv_phase" block. Complex IQ centers are subtracted from "data_in", which is then passed to "CORDIC 4.0" for the phase calculation.

to configure the readout of the resonators. It records the location of the complex (I and Q) response of the resonator. The "conv_phase" block currently performs 2 steps shown inside the block in Figure **??**. First it subtracts a real and imaginary number from the complex I and Q data and then it passes the data to the "CORDIC 4.0" block to determine the phase in the complex plane from I and Q. The differencing step centers the resonator IQ loops on the origin so that the phase is calculated relative to the loop center.

The complex data routed to the "accumulator" and "avgIQ" blocks are done so prior to centering the IQ data. It would be very useful to have the "avgIQ" data centered before they are reported. This requires breaking the "conv_phase" block into two blocks. One block ("centerIQ") would center the complex IQ data and then pass it to a second block ("IQphase") that performs the CORDIC algorithm. The two taps that go off to the "accumulator" and "avgIQ" blocks would then be moved to the output of the "centerIQ" block.

2.2 flux ramp synchronization

The second step is to provide a synchronization signal to the firmware to keep track of the beginning of each flux ramp period for demodulation. The flux ramp signal will be provided by an external function generator that will take its timebase from the 10 MHz timing signal that is fed to the IF card on the ROACH. The IF card creates a 512 MHz clock that synchronizes the ADC/DAC card and the FPGA. The FPGA is clocked at 256 MHz. The flux ramp signal will be user-configurable and will have a rate ("ramp_rate") of 20–50 kHz, or 50–20 μs per ramp period. In addition to providing an output waveform for the flux ramp the function generator also outputs a sync signal that indicates the beginning of the flux ramp period. This is a TTL signal with a falling edge that is synchronous with the beginning of the ramp period.

The ROACH board has several SMA-based GPIO ports that will accept an external sync signal. There are two SMAs (J12 and J13) located near the ZDOK ports and SD card slot that are available. From the CASPER wiki: "The signal input to this connector is terminated into 50 ohms and then turned into the LVDS differential pair GPIO_CLK0_P and GPIO_CLK0_N via an Analog Devices ADCMP605BCPE comparator and enters the FPGA at pins G15/G16." According to this CASPER wiki page: the J12 signal can be accessed through the "gpio" block using "ROACH: aux0_clk" for the "I/O group".

The sync signal from the function generator will be read in through J12 and turned into a TTL signal ("ramp_trigger") that will control the demodulation block described in section **??**.

In order to make this a useful independent firmware modification step we will need a way to test it before introducing the next modification. Ultimately we want the phase time stream output by the original "conv_phase" – now the "IQphase" block – that goes into the QDR and phase snapshot blocks to be synchronized with the flux ramp period. The

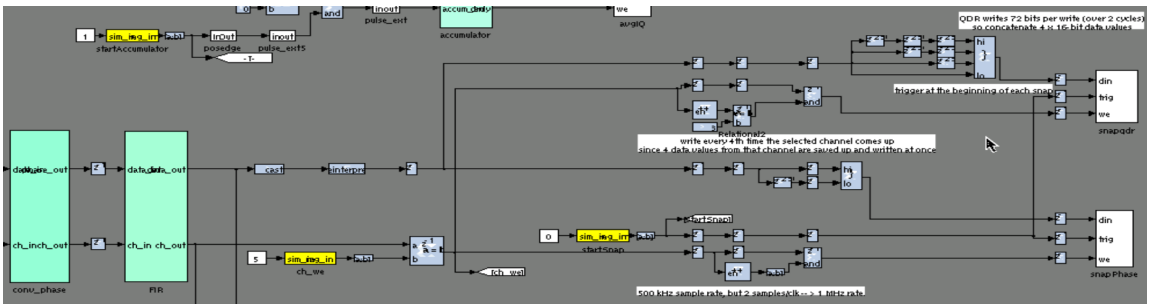

Figure 3: The QDR and phase snap blocks.

machinery for the snapshot blocks are shown in figure **??**. So, an additional step in this firmware modification is to modify the triggering of the snapshot block so that it waits for the "ramp_trigger" to initialize. Currently, the snapshot acquisition is triggered by setting the "startSnap" software register to 1. We want to wait for both the "startSnap" to go high and the falling edge of the "ramp_trigger" TTL signal. A "neg_edge" block on the output of the"ramp_trigger" will cause that input to go high for 1 clock cycle at the end of the ramp period. It should be sufficient to trigger the snap phase blocks by adding an AND logic block, which has inputs from "neg_edge" and "startSnap".

2.3 preliminary flux ramp demodulation

The basic operation of flux ramp modulation in a microwave SQUID multiplexer is shown in figure **??**. A sawtooth waveform (top panel) is supplied by a function generator and applied to a common flux ramp line that is inductively-coupled to all SQUIDs in the multiplexer. The linear ramp has been chosen to have a rate of 20 kHz or a 50 μs period and sweep through 5 flux quanta (Φ_0). The response of a SQUID to this linear ramp is shown by the sinusoidal signal in the bottom panel by the black line. For the choice of 5 quanta and 20 kHz ramp rate, the fundamental carrier frequency of the SQUID response will be at 100 kHz. In addition to the common flux ramp line, each SQUID is also inductively

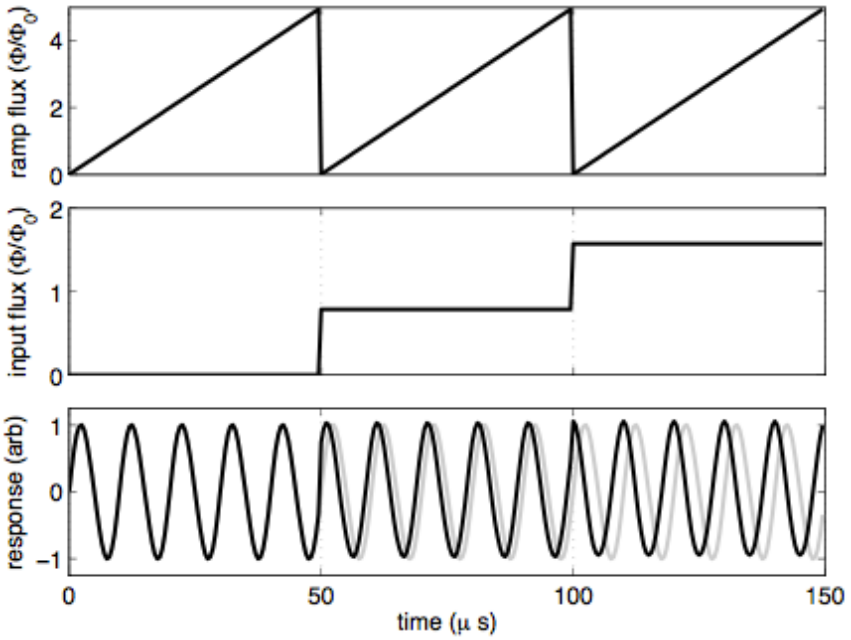

Figure 4: Flux ramp response of a microwave SQUID multiplexer channel.

coupled to the output of a voltage-biased TES. If a DC signal to this input line is stepped for each ramp period, as shown in the middle panel, then this will cause an offset in the phase (x-direction) of the SQUID response, as indicated by the shift between the gray and black lines. This is the signal of interest that needs to be extracted each frame; the phase offset of the sinusoid.

A continuous phase detection circuit would give the instantaneous phase shift and thus provide a continuous measurement of the input flux. However, the linear flux ramp cannot be continuously ramped ad infinitum and must reset to zero before continuing to ramp again. We need to keep track of where these reset periods occur since they introduce short-lived ($\sim 10\mu$s) transients in the SQUID response which we would like to blank out. This is specified by a user-configurable "blank_period" that specifies the number of samples acquired at 512 MHz that should be excluded from the beginning of each ramp period. After

blanking the initial transient we then measure the average phase shift over the remaining portion of the flux ramp period, e.g. from 10μs – 50μs for a 20 kHz ramp rate.

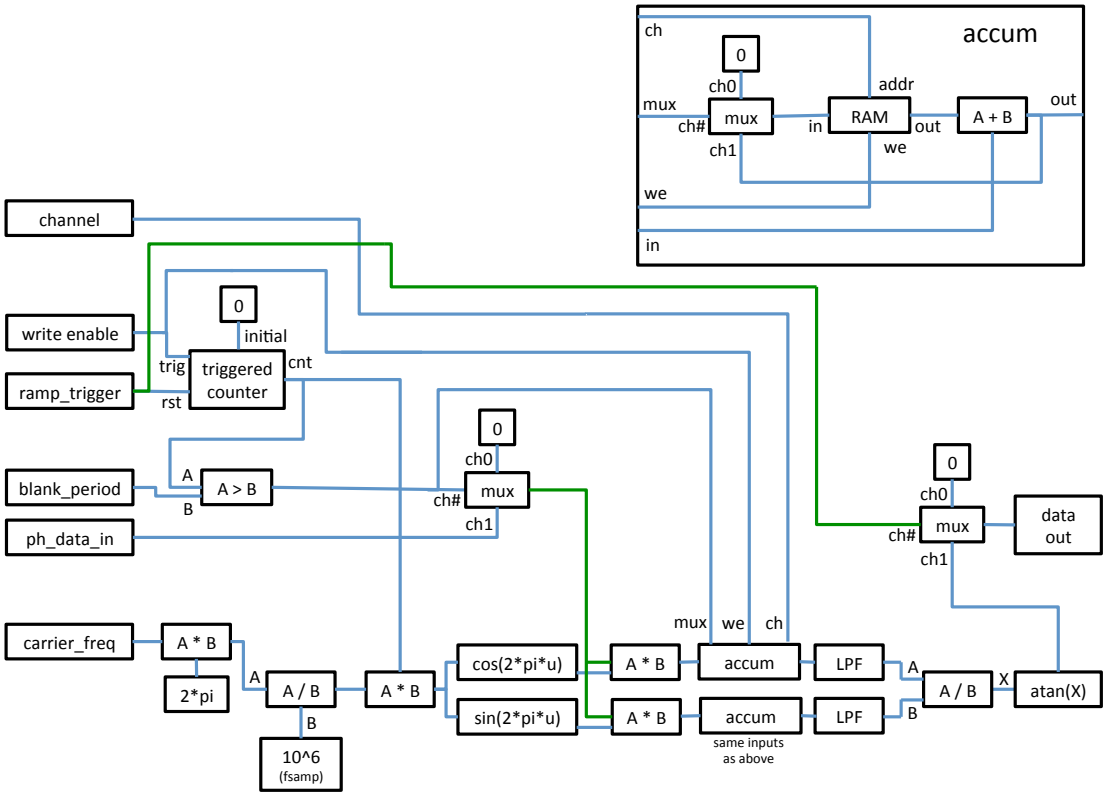

Figure 5: A schematic for the flux ramp demodulation firmware.

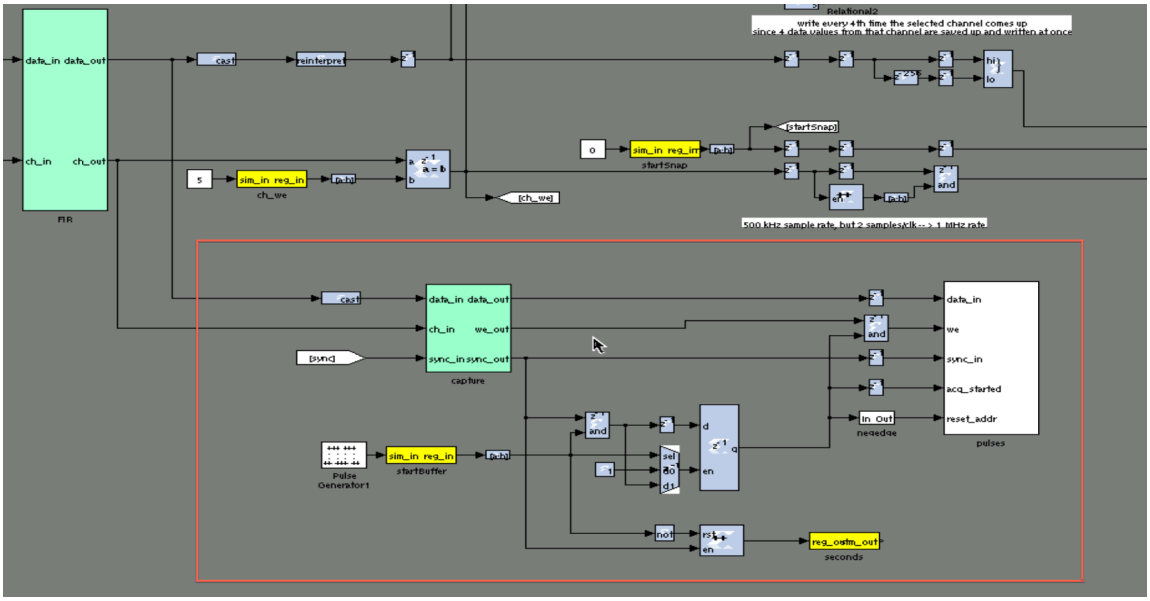

Figure 6: The pulse detection and analysis block outlined in red.