Adaptable Firmware for Microwave SQUID Readout on a Commercial Hardware Platform

Johnathon D. Gard1, D. T. Becker1, D. A. Bennett2, J. W. Fowler1,2, G. C. Hilton1, J. Imrek1, J. A. B. Mates1, K. M. Morgan1, C. D. Reinsteena, D. R. Schmidt2, D. S. Swetz3, J. N. Ullom1,2, L. R. Vale1, A. L. Wessels1
1 University of Colorado Boulder: Physics Department
2 National Institute of Standards and Technology
3 University of Colorado Boulder

ABSTRACT: As the size and scale of low temperature detector arrays continues to grow, the demands on the cryogenic multiplexing has dramatically increased. The microwave SQUID multiplexer is meant to address this issue by opening the possibility of multiple gigabits of readout bandwidth per coax pair. With this readout technique, complexity is moved from the cryogenic stages to the room temperature hardware and digital signal processing firmware. With the variety of microwave SQUID multiplexer designs that are being developed at NIST, the signal processing firmware must have sufficient agility to accommodate different numbers of channels, resonator bandwidths, and resonator spacings. The necessary flexibility is possible with the advent of high-performance ADCs and DACs integrated with field programmable gate arrays (FPGAs). Our firmware is implemented on a commercial, off-the-shelf data acquisition platform capable of manipulating up to four gigahertz of bandwidth. Depending on the application, we can modify the channelization module to achieve target resonator bandwidths and spacings. We will discuss the application space of microwave SQUID multiplexers and how that impacts the firmware modules that need to be implemented. This modular firmware architecture for microwave SQUID multiplexers can be ported to a wide variety of Xilinx FPGAs, including the current and future generations of Xilinx’s RFSoCs.

Microwave Multiplexing

Microwave multiplexing is a well-known technique in various sensor applications, such as cryogenic detectors where it is used to down-convert and sample multiple detector signals onto a single RF channel. This technique allows for the efficient readout of a large number of detectors by sharing a common RF channel and using a bank of mixers to down-convert and separate the individual detector signals. The key advantage of microwave multiplexing is its ability to handle large numbers of channels with a single RF channel, which is particularly useful in scenarios where the number of channels is much larger than the available RF bandwidth.

Channelization

Channelization is the ability to select specific bands of interest from the down-converted signals. It involves the process of selecting and isolating individual signal bands from the multiplexed signal for further processing. Channelization is achieved by applying filters, such as bandpass filters, to the multiplexed signal. These filters allow the desired frequency bands to pass through while attenuating the other bands, effectively isolating the signal components of interest.

Microwave Mux System Attributes

Below is a description of pertinent variables governing the key design choices of the firmware.

- **Chip Complexity**: This parameter indicates the complexity of the chip required to handle a given number of channels. A higher chip complexity means that more resources on the FPGA are needed to implement the firmware.
- **Chip Delay**: The delay associated with processing the signal through the chip, which affects the overall latency of the system.
- **Power Consumption**: The amount of power consumed by the chip. Power consumption is an important factor in the design of hardware platforms as it affects the cooling requirements and the overall system efficiency.

Microwave Multiplexer Application Space

One way to visualize how these attributes affect the firmware design is to plot channel count versus bandwidth. A chart of several hardware platforms’ bandwidths, presented as shaded areas, shows the possibilities for readout given a constant chip complexity and number of channels.

Resource Utilization

For above implementations:

- **LUT**: Logic elements used in the firmware.
- **FF**: Flip-flops used for data storage.
- **DSP**: Digital signal processing elements used for filtering or other DSP tasks.
- **BRAM**: Block RAM used for data storage.

CONCLUSION: With an ever-changing landscape of ADCs, DACs, and FPGAs capabilities it becomes necessary to write firmware that is adaptable across multiple platforms. This puts the requirement on the firmware to be highly parameterizable. When the firmware is coded in a modular fashion, the parameterization becomes easier. We have produced such an adaptable firmware design.

Acknowledgements

I would like to gratefully acknowledge support to the University of Colorado Boulder from the U.S. Department of Energy’s Nuclear Energy University Program (NEUP), National Institute of Standards and Technology Innovations in Measurement Science (IMS) and the National Aeronautics and Space Administration’s Strategic Astrophysics Technology (SAT) award.