IOWA STATE UNIVERSITY
College of Engineering

Intermediate Frequency Data Aquisition Device (*IF_DAD*)

Team: SD_DEC23_20

Nathan Ayers, Matthew Caron, Michael Levin, and Rodrigo Romero

Advisor: Dr. Mohammad Tayeb Al Qaseer

Website

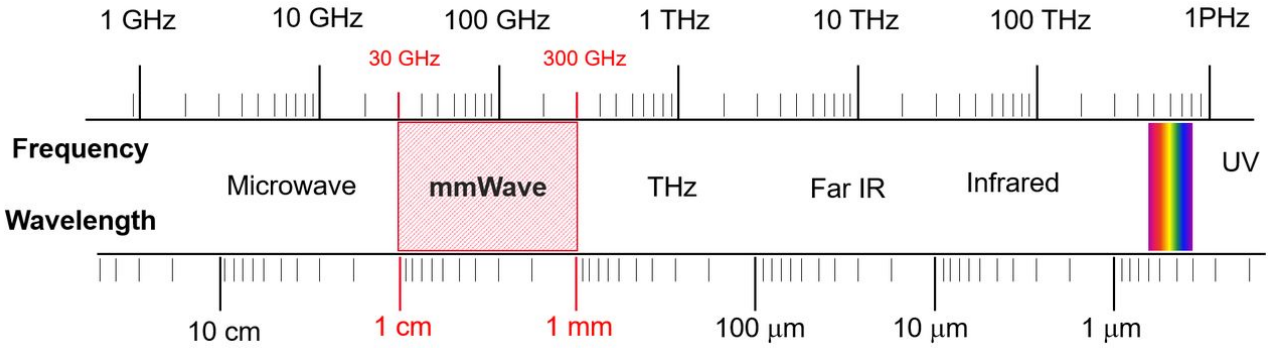
https://sddec23-20.sd.ece.iastate.edu/

Problem Statement

- Across all industries, the failure of “Safety-Critical” systems can result in large financial loss or even death.
- Iowa State Center for Non-Destructive Evaluation research promotes safety in industry by understanding the lifespan of such systems.
- Our client, Dr. Al Qaseer has requested that we design the Intermediate Frequency circuit for a Millimeter wave radar.
- Final design will be utilized and improved upon by researchers and students within the CNDE.

Millimeter waves

- Frequency range from 30 - 300 GHz, opposed to Microwaves range of 300 MHz - 30 GHz.
- Distinguished for capability to penetrate non-conductive materials (e.g., wood, concrete, fiber composites).
- Interact with the surface of conductive materials, can identify cracks, pitting.
- Higher frequencies are more sensitive to geometrical variations.



Design Requirements:

- Must interface with existing hardware/software that includes Up converters and Down converters, this will require a signal input and output.
- Our design will ideally output an IF (intermediate frequency) signal of 10 MHz signal, which will be “sped up” by the existing hardware.
- With the input signal that is returned from the output signal, we must convert that into its real and imaginary components.
- This converted data must be organized clearly into a file on the PC, so that Labview or Matlab may graph it.
- FPGA clock is 100 MHz.
- Communicating with PC to FPGA via a USB-C port.

Intended Users and Uses

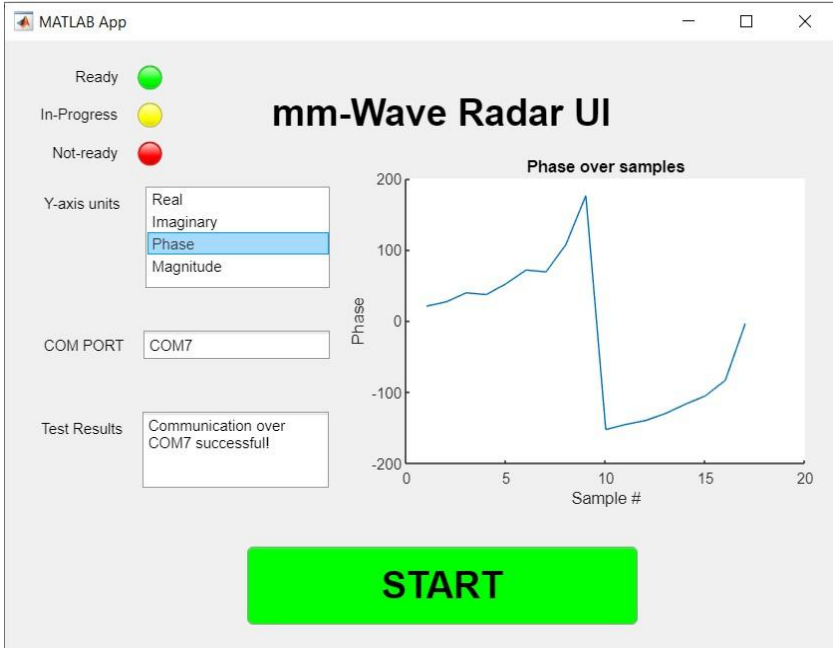
Users: Iowa State Researchers, and students.

Uses: Evaluating the integrity of a material without having to disassemble the product. This means detecting internal defects, cracks, and analyzing the material properties of insulating samples. While some imaging radars can be used to analyze metals, ours wouldn’t likely have those capabilities due to the refractive instability of microwaves on conductive materials.

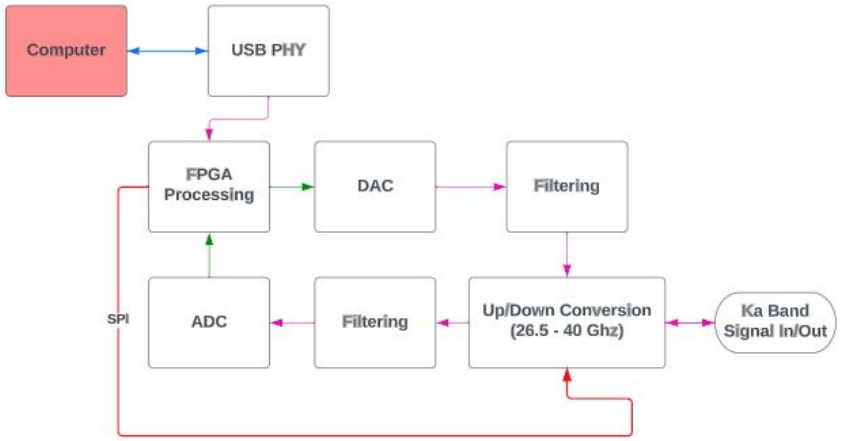
Design Approach

Computer Interface:

A Matlab program communicates with the FPGA and analyzes data received from the FPGA.



Design Approach Cont.



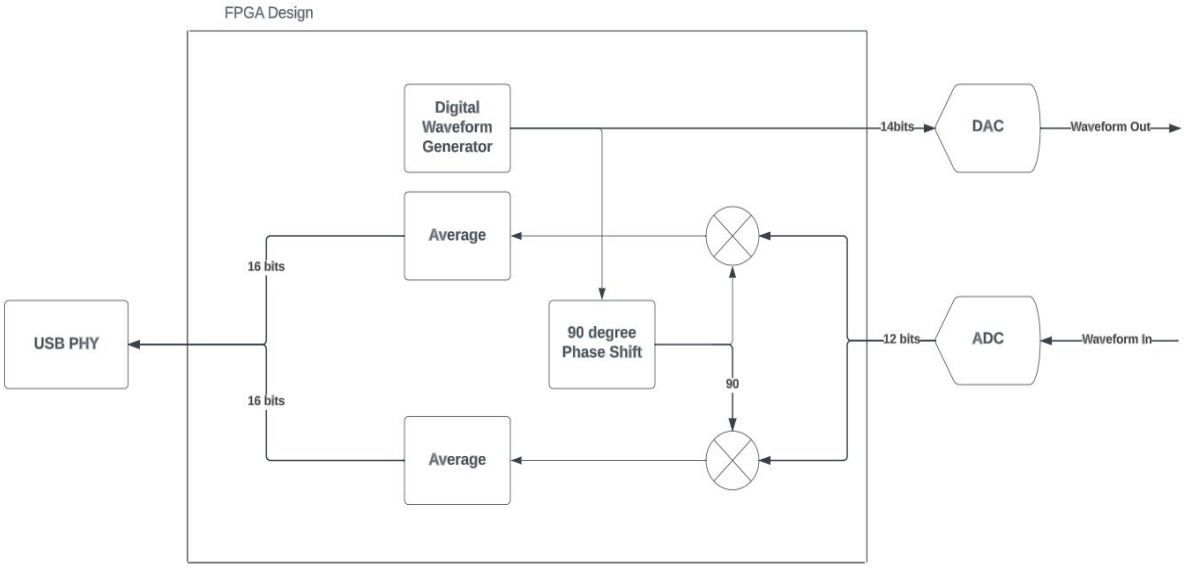
Hardware:

FPGA produces a 10 MHz sine wave. The DAC and up-converter to RF (26.5 - 40 GHz). Hardware then receives the signal, down converts it to the IF frequency, filters and sends it to the ADC where the FPGA processes and send data back to the UI

Design Approach Cont.

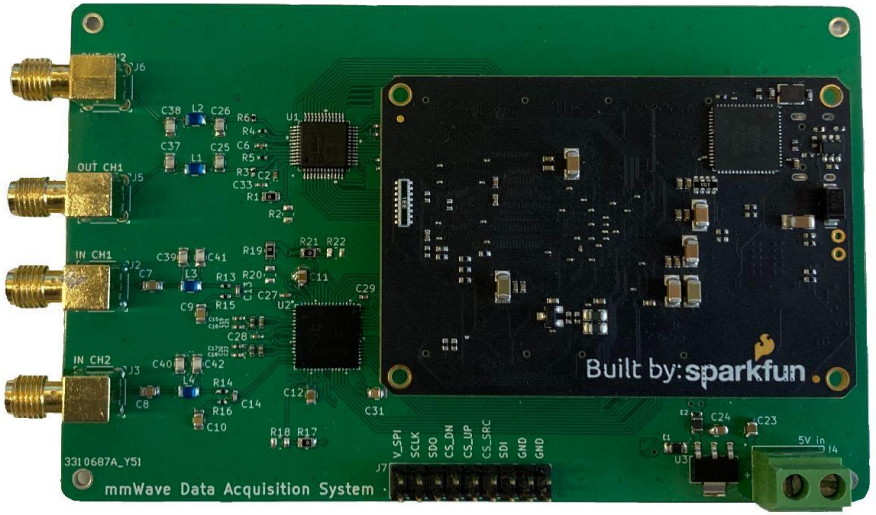
Digital Signal Processing (DSP):

The DSP system takes the raw data received by the ADC and splits it into real and imaginary parts based on the convolution of the original sine wave.



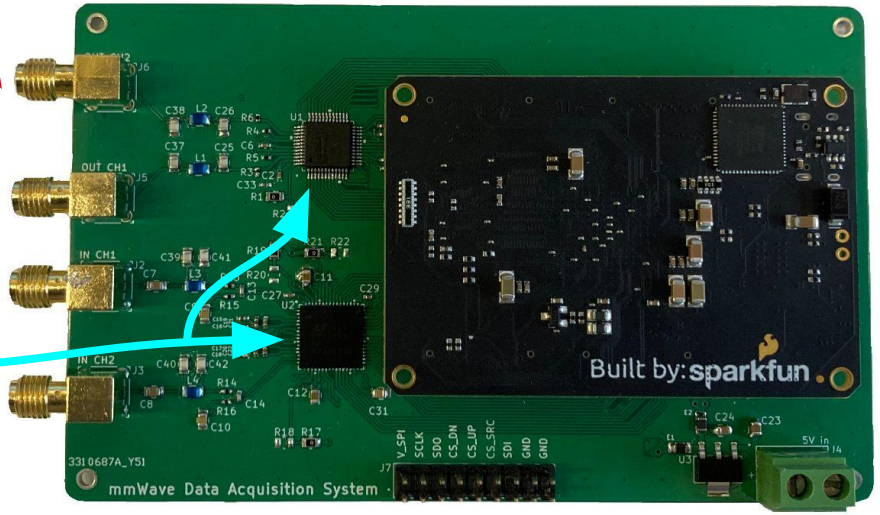
Technical Details:

- Four specific modules to implement, Hardware (ADC & DAC), SPI, DSP, and User Interface.
- HW, SPI, and DSP all require Vivado to program the FPGA (Xilinx Artix 7) in Verilog and C (via the Microblaze Processor)
- HW PCB interfaces with Alchitry AU FPGA development board, all HW, SPI, DSP code is loaded directly to the FPGA.
- UI requires MATLAB, specifically programmed in the MATLAB app designer



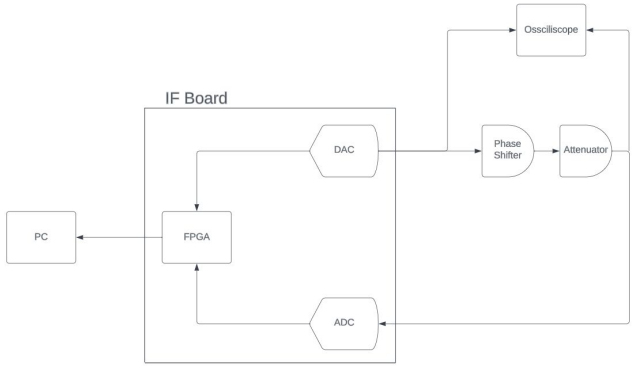
Standards:

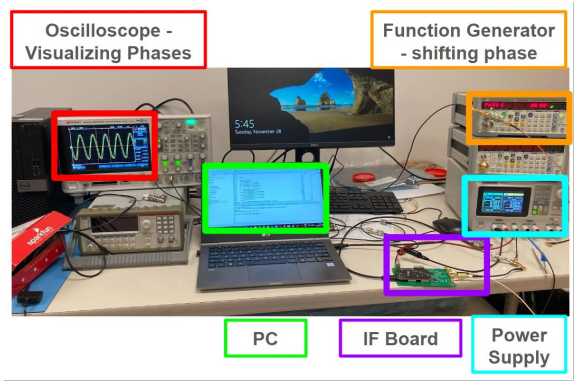
- PCB implemented by Matt with coaxial connectors at RF, and Millimeter-wave frequencies, we followed IEEE 287.1-2021, IEEE 287.2-2021, IEEE 287.3-2021.
- With presence of ADCs and DACs we had to be vigilant of terminology and test standards/protocol, so we paid close attention to IEEE 1658-2011, IEEE 1241-2010.



Testing

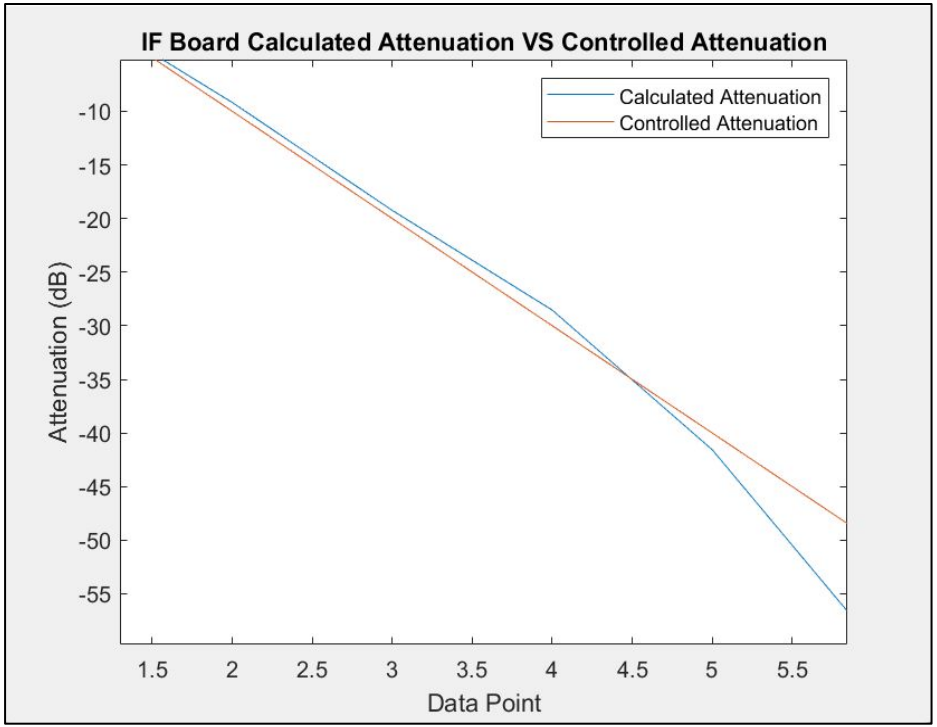
- Testing was done to emulate the phase and amplitude shifts the RF components introduce
- This testing consisted of inputting the sinusoidal IF waveform of 10 MHz into the reference of a function generator to artificially shift the phase of the IF output signal
- The output of the function generator was then fed back into the IF board for signal processing and data collection to the PC



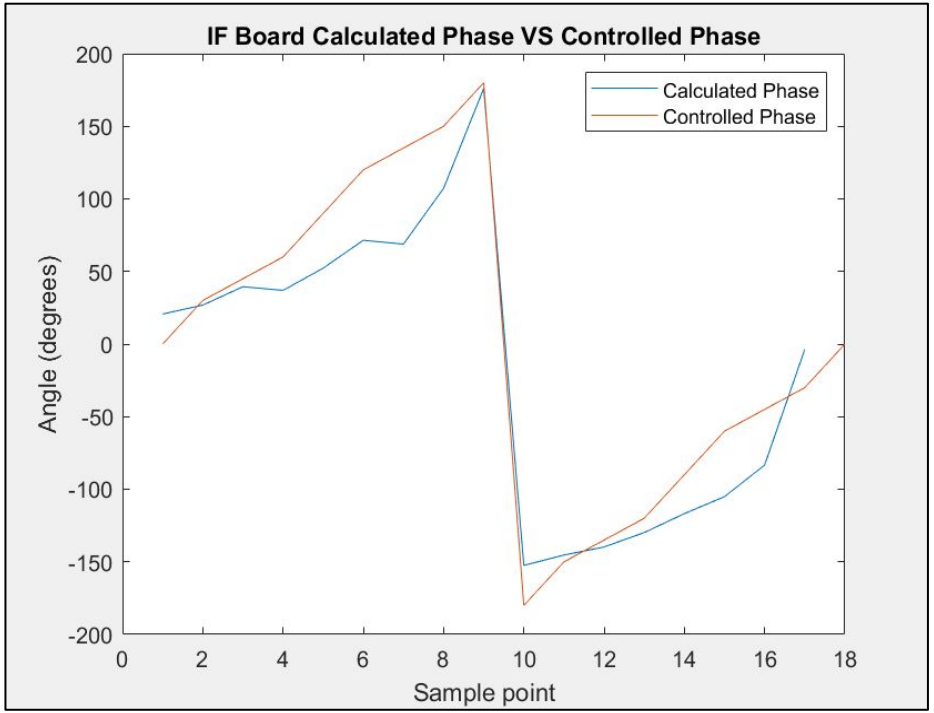


Results

As the IF output was attenuated from 0 dB to -50 dB the calculated attenuation from the IF board was accurately able to predict the actual controlled attenuation up to about - 40 dB



Results Cont.



As the phase of the IF output was swept from 0◦ to 360◦ the calculated phase from the IF board was able to predict the actual controlled phase difference

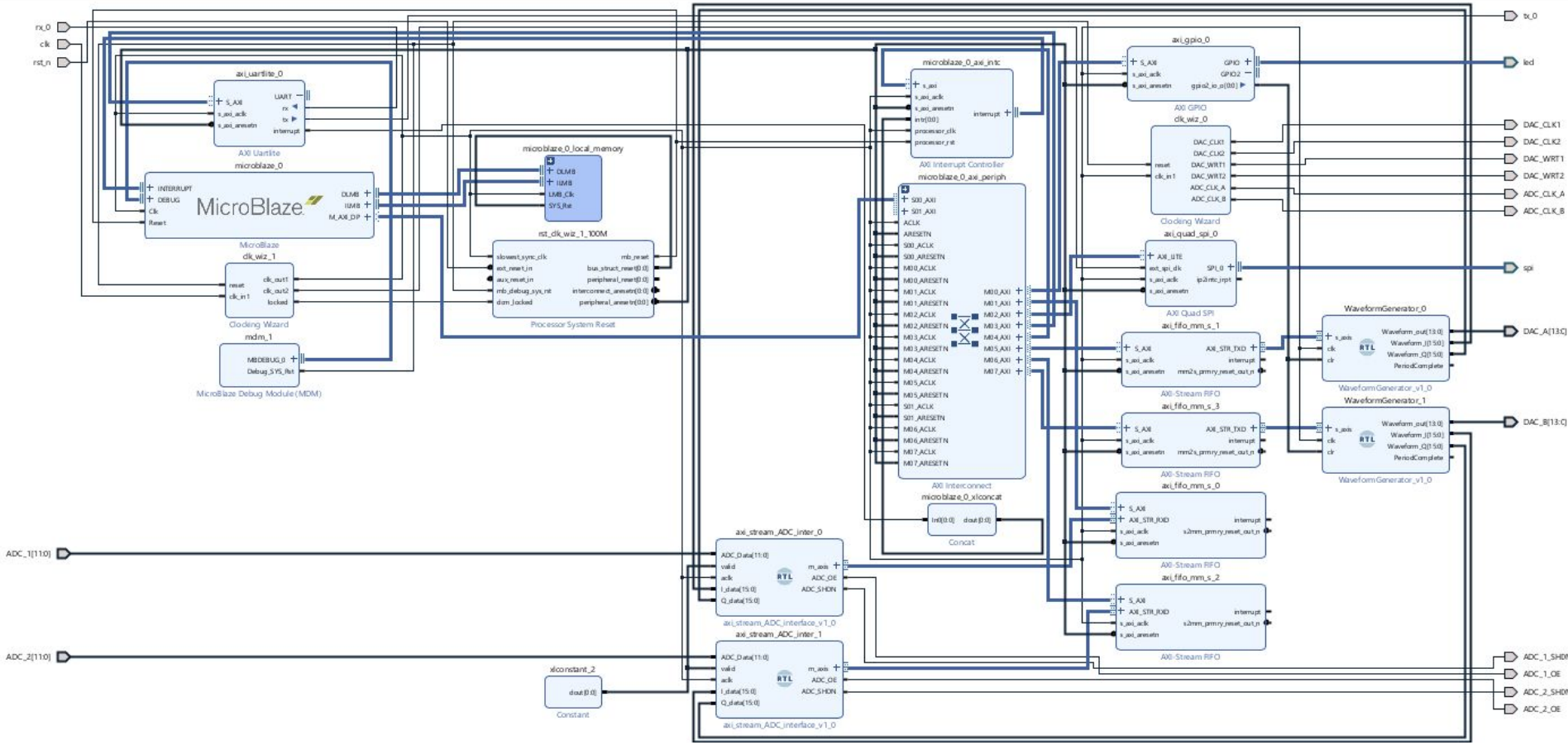
Conclusions

The designed intermediate frequency circuit works as expected under the specified conditions.

Future work includes the implementation of SPI to communicate the FPGA with upconverter and downconverter, as well as the quantification of the measured error compared to the theoretical results.

Questions?

Appendix

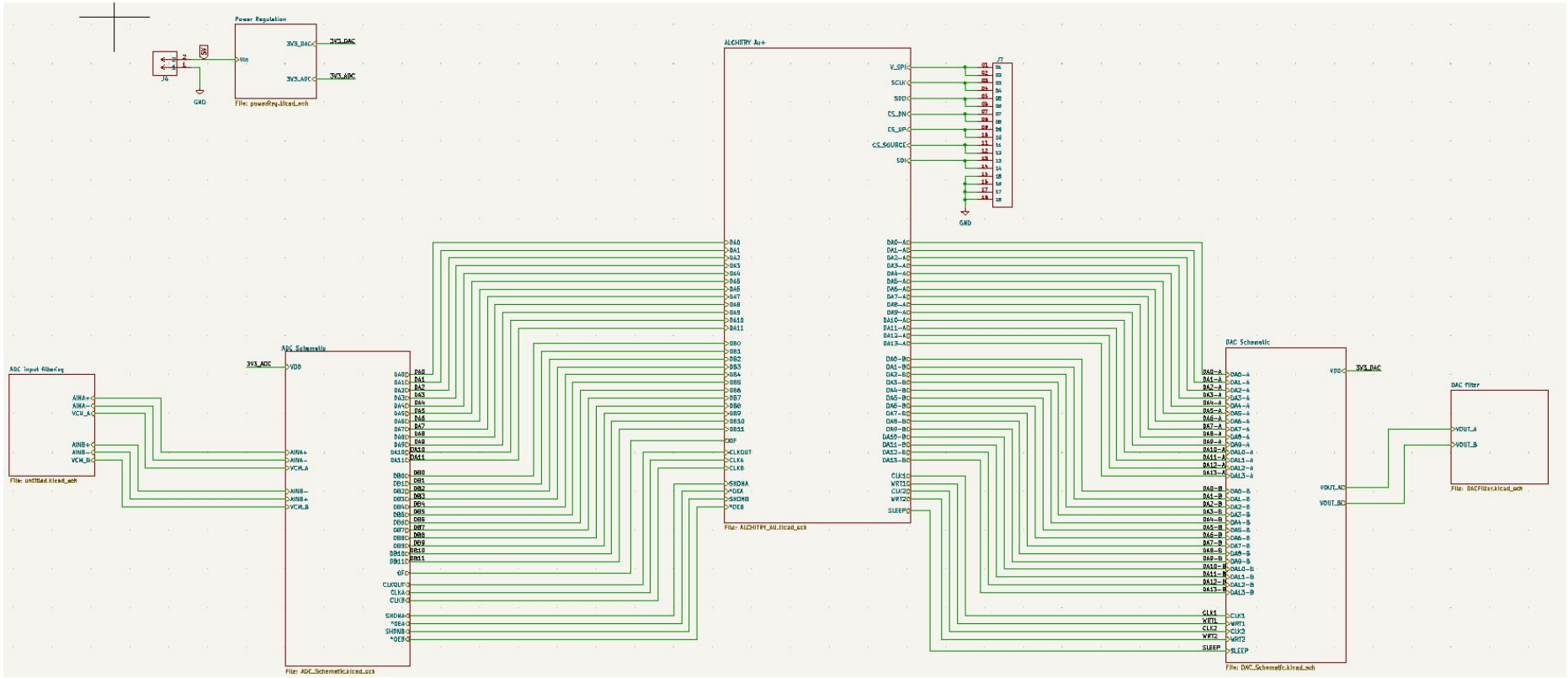

rx_0
clk
rst_n
axi_uartlite_0
AXI Uartlite
microblaze_0
MicroBlaze
clk_wiz_1
Clocking Wizard
mdm_1
MicroBlaze Debug Module (MDM)
microblaze_0_local_memory
rst_clk_wiz_1_100M
Processor System Reset
microblaze_0_axi_intc
AXI Interrupt Controller
microblaze_0_axi_periph
AXI Interconnect
microblaze_0_xlconcat
Concat
axi_gpio_0
AXI GPIO
clk_wiz_0
Clocking Wizard
axi_quad_spi_0
AXI Quad SPI
axi_fifo_mm_s_1
axi_fifo_mm_s_3
axi_fifo_mm_s_0
axi_fifo_mm_s_2
AXI-Stream FIFO
WaveformGenerator_0
WaveformGenerator_1
WaveformGenerator_v1_0
axi_stream_ADC_inter_0
axi_stream_ADC_inter_1
axi_stream_ADC_interface_v1_0
xlconstant_2
Constant
tx_0
led
DAC_CLK1
DAC_CLK2
DAC_WRT1
DAC_WRT2
ADC_CLK_A
ADC_CLK_B
spi
DAC_A[13:0]
DAC_B[13:0]
ADC_1[11:0]
ADC_2[11:0]
ADC_1_SHDN
ADC_1_OE
ADC_2_SHDN
ADC_2_OE

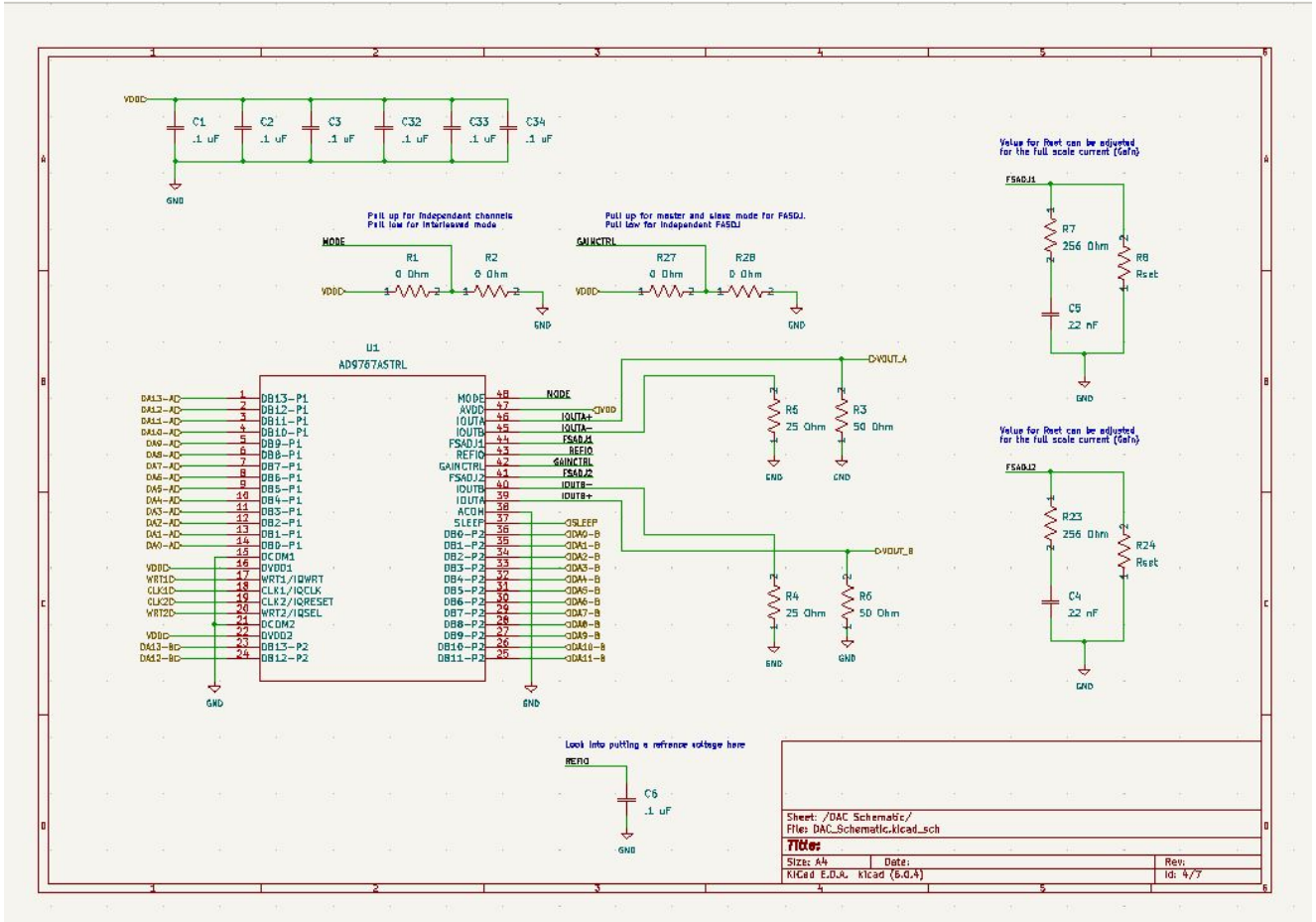
```
always @ (negedge aclk) begin
    I_temp <=  I_data* ADC_Data;
    //I_temp <= I_data;
end
always @ (negedge aclk) begin
    Q_temp <= Q_data* ADC_Data;
    //Q_temp <= Q_data;
end
```

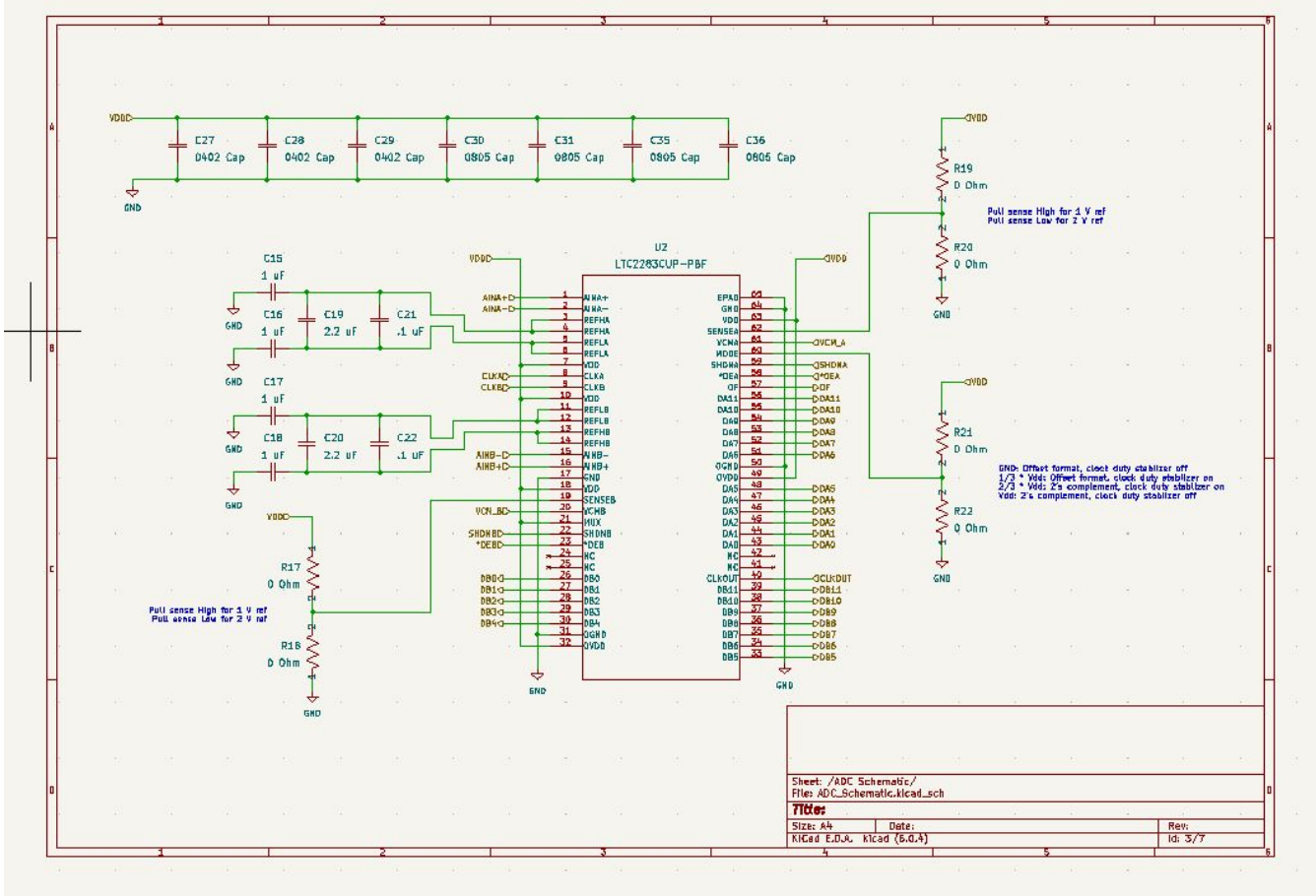
```
// iterate iter by one or reset if iter is greater than max_address
// if the iterator went though all of the values flash the period complete output
always @ (negedge clk) begin
    if (~s_axis_tvalid && (iter < (max_address - 1'b1))) begin
        PeriodComplete <= 1'b0;
    end else begin
        PeriodComplete <= 1'b1;
    end

    if (~s_axis_tvalid) begin
        iter = (iter < max_address) ? (iter + 1'b1): 9'h000;
    end else begin
        iter = latency;
    end
end
```

```
// convert signed to unsigned and clock out waveform data
always @ (posedge clk) begin
    if (~s_axis_tvalid) begin
        Waveform_out <= (storageData[iter] + 14'h2000) & 14'h3FFF;
    end
end
```


Power Regulation
3V3_DAC
3V3_ADC
Vin
GND
File: powerReg.kicad_sch
ALCHITRY Au+
V_SPI
SCLK
SDO
CS_DN
CS_UP
CS_SOURCE
SDI
GND
File: ALCHITRY_AU.kicad_sch
ADC Schematic
3V3_ADC
VDD
CLKOUT
CLKA
CLKB
SHDNA
*OEA
SHDNB
*OEB
File: ADC_Schematic.kicad_sch
ADC input filter/reg
AINA+
AINA-
VCM_A
AINB+
AINB-
VCM_B
File: untitled.kicad_sch
DAC Schematic
3V3_DAC
VDD
CLK1
WRT1
CLK2
WRT2
SLEEP
VOUT_A
VOUT_B
File: DAC_Schematic.kicad_sch
DAC filter
VOUT_A
VOUT_B
File: DACFilter.kicad_sch


VDDC
C1 .1 uF
C2 .1 uF
C3 .1 uF
C32 .1 uF
C33 .1 uF
C34 .1 uF
GND
Pull up for independent channels
Pull low for interleaved mode
MODE
R1 0 Ohm
R2 0 Ohm
Pull up for master and slave mode for FASDJ.
Pull low for independent FASDJ
GAINCTRL
R27 0 Ohm
R28 0 Ohm
U1
AD9767ASTRL
DB13-P1
DB12-P1
DB11-P1
DB10-P1
DB9-P1
DB8-P1
DB7-P1
DB6-P1
DB5-P1
DB4-P1
DB3-P1
DB2-P1
DB1-P1
DB0-P1
DCOM1
DVDD1
WRT1/IQWRT
CLK1/IQCLK
CLK2/IQRESET
WRT2/IQSEL
DCOM2
DVDD2
DB13-P2
DB12-P2
MODE
AVDD
IOUTA
IOUTB
FSADJ1
REFIO
GAINCTRL
FSADJ2
IOUTB
IOUTA
ACOM
SLEEP
DB0-P2
DB1-P2
DB2-P2
DB3-P2
DB4-P2
DB5-P2
DB6-P2
DB7-P2
DB8-P2
DB9-P2
DB10-P2
DB11-P2
DVDD
IOUTA+
IOUTA-
IOUTB+
IOUTB-
DVOUT_A
DVOUT_B
R6 25 Ohm
R3 50 Ohm
R4 25 Ohm
R6 50 Ohm
Value for Rset can be adjusted for the full scale current (Gain)
FSADJ1
R7 256 Ohm
R8 Rset
C5 22 nF
FSADJ2
R23 256 Ohm
R24 Rset
C4 22 nF
Look into putting a refrence voltage here
REFIO
C6 .1 uF
Sheet: /DAC Schematic/
File: DAC_Schematic.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.4)
Id: 4/7


C27 0402 Cap
C28 0402 Cap
C29 0402 Cap
C30 0805 Cap
C31 0805 Cap
C35 0805 Cap
C36 0805 Cap
VDDC
GND
C15 1 uF
C16 1 uF
C19 2.2 uF
C21 .1 uF
C17 1 uF
C18 1 uF
C20 2.2 uF
C22 .1 uF
R17 0 Ohm
R18 0 Ohm
Pull sense High for 1 V ref
Pull sense Low for 2 V ref
U2
LTC2283CUP-PBF
R19 0 Ohm
R20 0 Ohm
R21 0 Ohm
R22 0 Ohm
Pull sense High for 1 V ref
Pull sense Low for 2 V ref
GND: Offset format, clock duty stabilizer off
1/3 * Vdd: Offset format, clock duty stabilizer on
2/3 * Vdd: 2's complement, clock duty stabilizer on
Vdd: 2's complement, clock duty stabilizer off
Sheet: /ADC Schematic/
File: ADC_Schematic.kicad_sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (6.0.4)
Id: 3/7

SPI

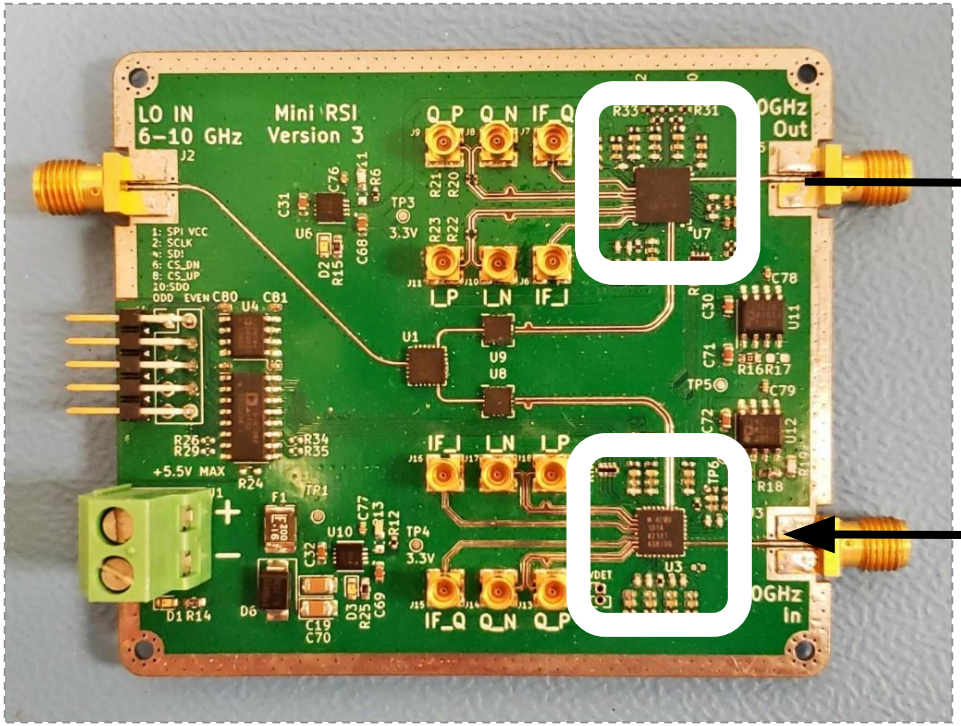
Required pins:

SCLK

SDIN

SDO

SEN.



Upconverter

Downconverter