GNU Radio Zedboard Implementation with FPGA Acceleration

Khodamoradi, Hong, Pandit, Parikh

Overview

- FPGA communication blocks via gr_modtool
- WBFM functions separately created Vivado HLS
- Overall WBFM block needs to be verified in Vivado HLS

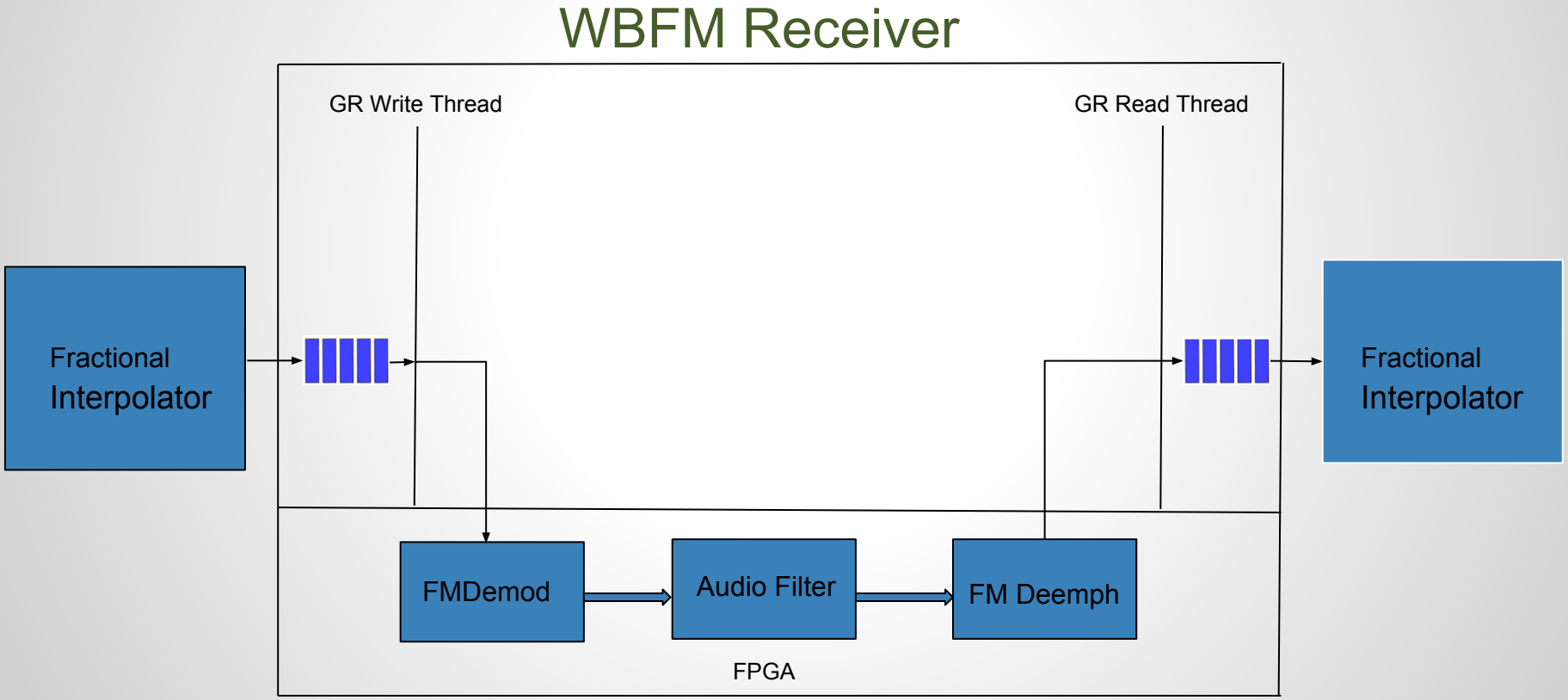
Accomplishments

- Used gr_modtool to create blocks to send data to and receive data from FPGA via Xillybus (Ali)
- Moved volk multiply, volk dot product, fast_atan2f, FIR, and IIR functions into C (Keyur, Nishant, Jun)
  - Still troubleshooting FIR
- Created unified test bench to evaluate all functions working together in WBFM block in hardware (Nishant, Ali)

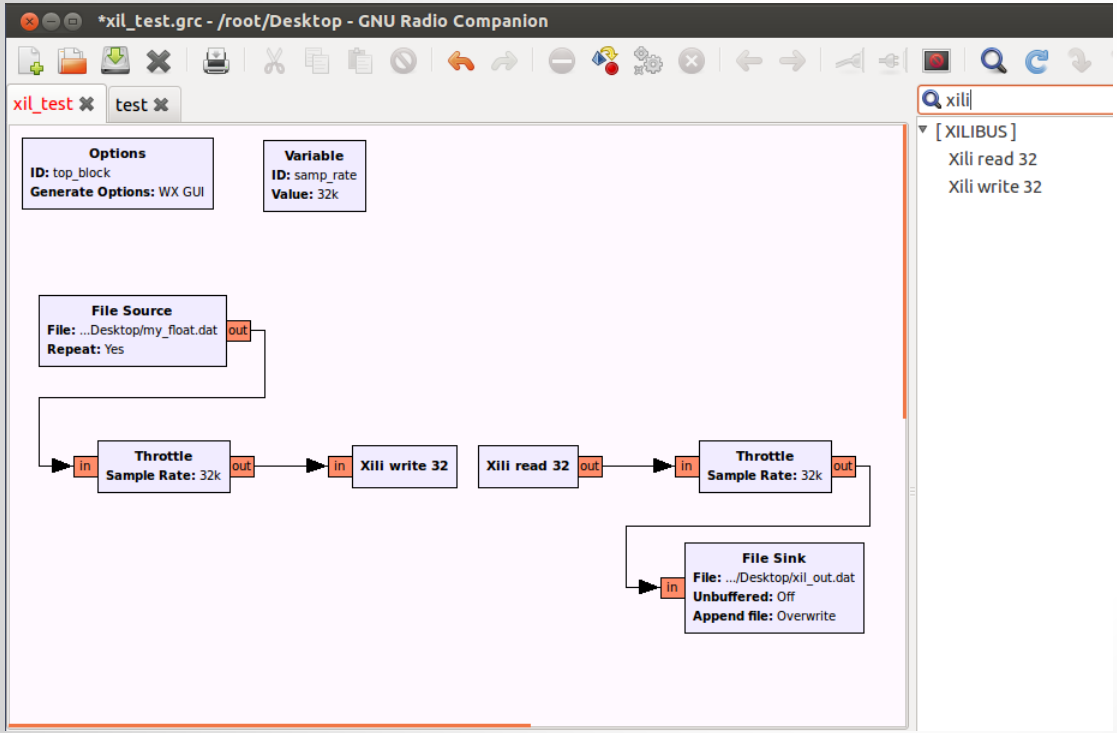
Current Progress

5/29/2015	Generate Test Data for each Function for HLS test benches	Ali	DONE
	C Function verification using HLS test data and test benches 1. volk_32fc_x2_multiply_conjugate_32fc // nishant 2. fast_atan2f // jun 3. volk_32f_x2_dot_prod_32f_a // done by nishant 4. IIR_filter // keyur	Team	DONE
	Synthesis C Functions into HLS IP	Team	In progress
	Host Code for write, read FPGA	Ali	DONE
	Integrate HLS IP into FPGA	Team	DONE
6/12/2015	FPGA End to End verification using Test Data	Team	In progress
	Optimizing hw-sw Performance	Team	In progress
	Final Demonstration	Team	In progress

Proposed Design



Xillybus blocks for GRC



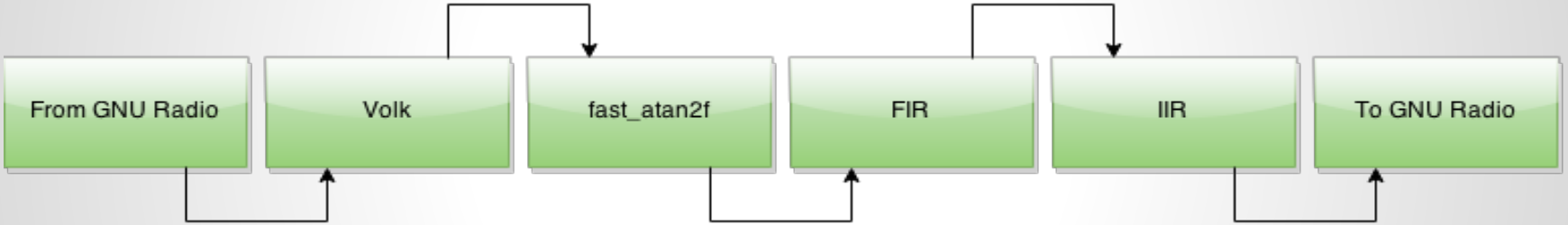

```
xili_write_32::sptr
xili_write_32::make()
{
  fd = open("/dev/xillybus_write_32", O_WRONLY);

  if (fd < 0) {
    printf("Failed to open write_32");
  }
  return gnuradio::get_initial_sptr
    (new xili_write_32_impl());
}

/*
 * The private constructor
 */
xili_write_32_impl::xili_write_32_impl()
  : gr::block("xili_write_32",
          gr::io_signature::make(1, 1, sizeof(float)),
          gr::io_signature::make(0, 0, 0))
{}

/*
 * Our virtual destructor.
 */
xili_write_32_impl::~xili_write_32_impl()
{
}
```

Function Sequence



Upcoming Two Weeks Sprint

- Insert FPGA block into GNU Radio Companion flow graph and test quality of playback
- Optimize FPGA Block
- Documentation

Quarter Plan

Goals left to complete this project

- Integrate Verilog code with GNU Radio
  - ensure data is being processed quickly enough
  - beware scheduling issues
- Listen to GNU Radio without skipping due to buffer overflows