Channelization using RFNoC

GRCON 2017

Phil Vallance

Critical Mission Engineering

Company Info

CRITICAL MISSION ENGINEERING

- Engineering services in the areas of embedded hardware, firmware, software development and test.
- Specialties include: RF, DSP, FPGA, C/C++, Python, Linux.
- From initial operational concept, through engineering design & development, to field deployment.

Table of contents

Introduction

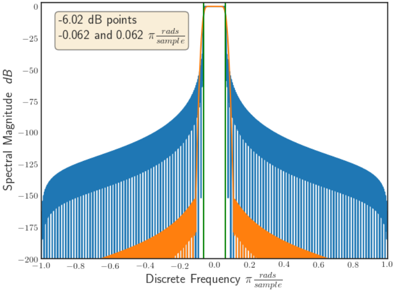
Introduction : Goals

- Review the derivation of the $M/2$ channelizer structure found in *A Versatile Multichannel Filter Bank with Multiple Channel Bandwidths* [Harris F.(2010)].
- Provide detailed FPGA implementation optimized for both the Xilinx FPGA architecture and the RFNoC framework.
- Detail implications using block-floating point FFT core.
- Describe current filter tap generation algorithm based on [Lubberhuizen(2010)].
- Discuss limitations and future work.

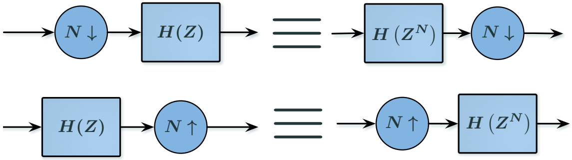
Channelizer Background

Channelizer Background : Motivation

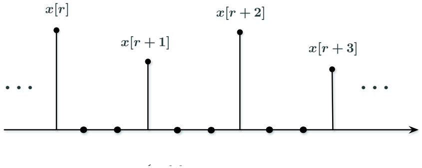
- Relaxes the filter transition bandwidth requirements.
- Allows perfect reconstruction of the channels.
- Efficient implementation.



Noble Identities

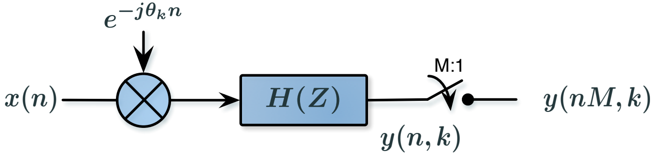


Time Expansion



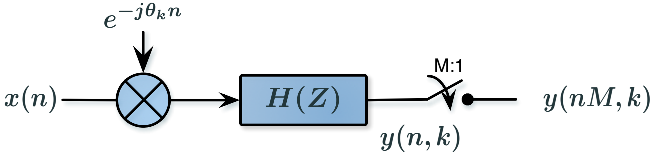
$$x_k[n] = \begin{cases} x[r], & n = Mr \\ 0, & n \notin M\mathbb{Z} \end{cases} \Rightarrow X(Z^M)$$

Channelizer Background : Channel Selector

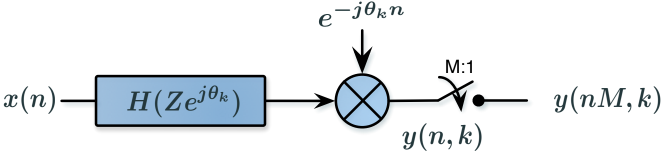


$$y(n,k) = \left[x(n)e^{-j\theta_k n}\right] * h(n) = \sum_{r=0}^{N-1} x(n-r)e^{-j\theta_k n}h(r)$$

Channelizer Background : Channel Selector

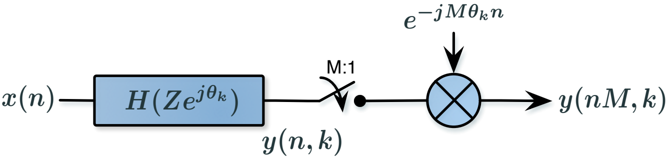


$$y(n,k) = \left[x(n)e^{-j\theta_k n}\right] * h(n) = \sum_{r=0}^{N-1} x(n-r)e^{-j\theta_k n}h(r)$$

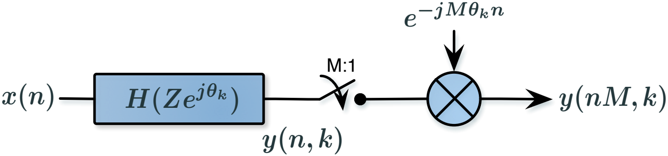


$$y(n,k) = \sum_{r=0}^{N-1} x(n-r)e^{-j\theta_k(n-r)}h(r) = e^{-j\theta_k n}\sum_{r=0}^{N-1} x(n-r)h(r)e^{j\theta_k r}$$

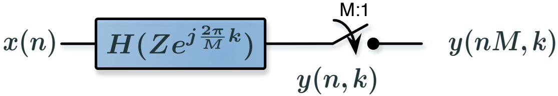
Channelizer Background : Channel Selector



Channelizer Background : Channel Selector



Restrict θ_k to a multiple of $\frac{2\pi}{M}$

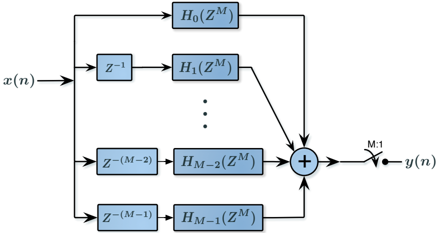


Channelizer Background : Filter Transformation

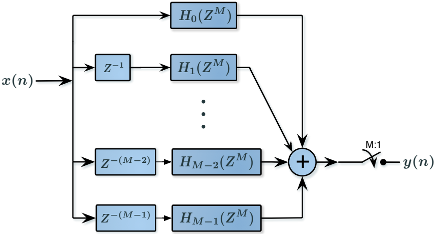
$$H(Z) = \sum_{n=0}^{N-1} h(n)Z^{-n} = \sum_{r=0}^{M-1} Z^{-r} H_r(Z^M)$$
$$= \sum_{r=0}^{M-1} Z^{-r} \sum_{n=0}^{(N/M)-1} h(r+NM)Z^{-nM}$$

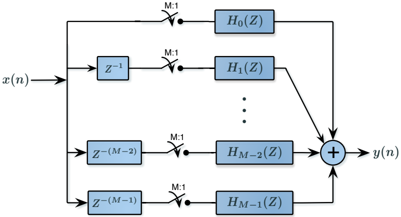
$$\begin{array}{rccccccc}
H(z) = & h(0) & + & h(M+0)Z^{-M} & + & h(2M+0)Z^{-2M} & + & \dots \\
& h(1)Z^{-1} & + & h(M+1)Z^{-(M+1)} & + & h(2M+1)Z^{-(2M+1)} & + & \dots \\
& h(2)Z^{-2} & + & h(M+2)Z^{-(M+2)} & + & h(2M+2)Z^{-(2M+2)} & + & \dots \\
& h(3)Z^{-3} & + & h(M+3)Z^{-(M+3)} & + & h(2M+3)Z^{-(2M+3)} & + & \dots \\
& \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\
& h(M-1)Z^{-(M-1)} & + & h(2M-1)Z^{-(2M-1)} & + & h(3M-1)Z^{-(3M-1)} & + & \cdots
\end{array}$$

Channelizer Background : Filter Transformation

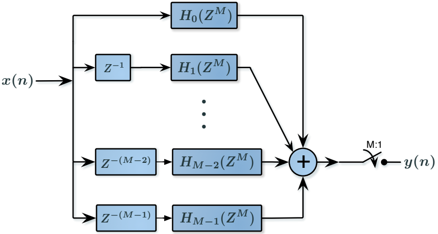


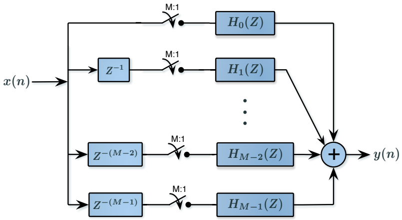
Channelizer Background : Filter Transformation

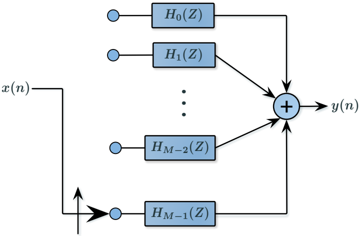




Channelizer Background : Filter Transformation





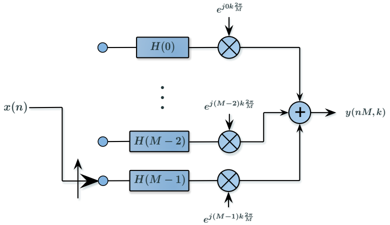


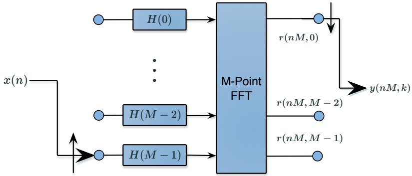
Channelizer Background : M Filter Transformation

Substitute Z^{-1} with $Z^{-1}e^{j\theta}$ and Z^{-M} with $Z^{-M}e^{jkM(2\pi/M)}$

$$
\begin{aligned}
G(Z) &= \sum_{n=0}^{N-1} h(n)[e^{-j\theta}Z]^{-n} \\
&= h(0) + h(1)e^{j\theta}Z^{-1} + h(2)e^{2j\theta}Z^{-2} + \cdots + h(N-1)e^{j(N-1)\theta}Z^{-(N-1)}
\end{aligned}
$$

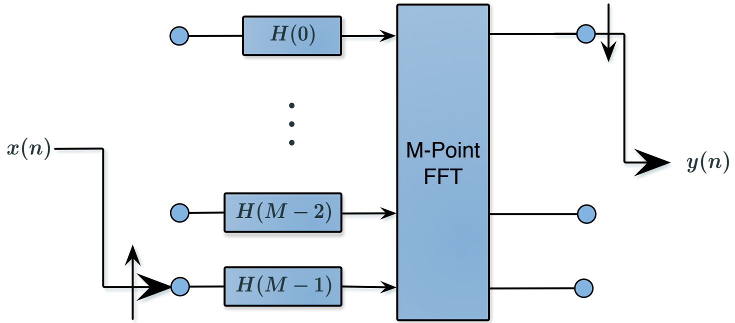
$$
\begin{array}{lcccccccc}
G(z) & = & h(0) & + & h(M+0)Z^{-M}e^{jkM(2\pi/M)} & + & h(2M+0)Z^{-2M}e^{jk2M(2\pi/M)} & + & \cdots \\
 & & h(1)Z^{-1}e^{jk(2\pi/M)} & + & h(M+1)Z^{-(M+1)}e^{jk(M+1)(2\pi/M)} & + & h(2M+1)Z^{-(2M+1)}e^{jk(2M+1)(2\pi/M)} & + & \cdots \\
 & & h(2)Z^{-2}e^{2jk(2\pi/M)} & + & h(M+2)Z^{-(M+2)}e^{jk(M+2)(2\pi/M)} & + & h(2M+2)Z^{-(2M+2)}e^{jk(2M+2)(2\pi/M)} & + & \cdots \\
 & & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \cdots \\
 & & h(M-1)Z^{-(M-1)}e^{(M-1)jk(2\pi/M)} & + & h(2M-1)Z^{-(2M-1)}e^{jk(2M-1)(2\pi/M)} & + & h(3M-1)Z^{-(3M-1)}e^{jk(3M-1)(2\pi/M)} & + & \cdots
\end{array}
$$





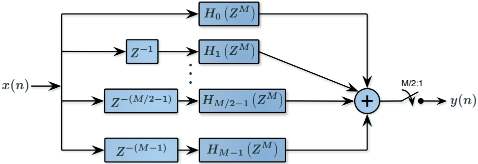
Channelizer Background

Standard Decimate by M Channelizer

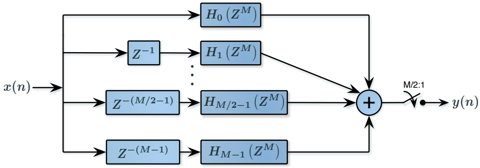


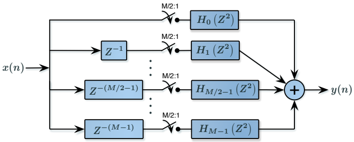
$$y(n,k) = \left[x(n)e^{-j\theta_k n}\right] * h(n) = \sum_{r=0}^{N-1} x(n-r)e^{-j\theta_k n}h(r)$$

Channelizer Background : $M/2$ Filter Transformation

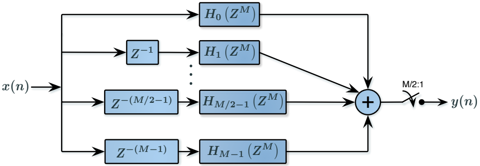


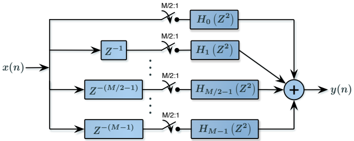
Channelizer Background : $M/2$ Filter Transformation

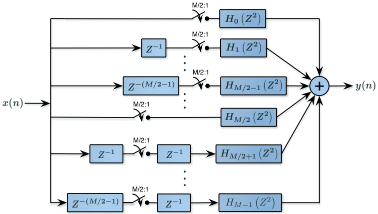




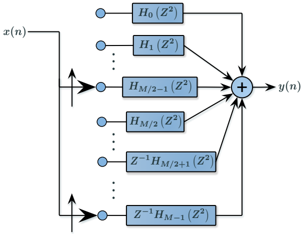
Channelizer Background : $M/2$ Filter Transformation

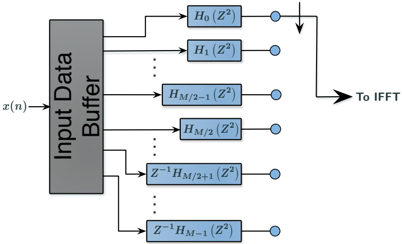






Channelizer Background : $M/2$ Filter Transformation





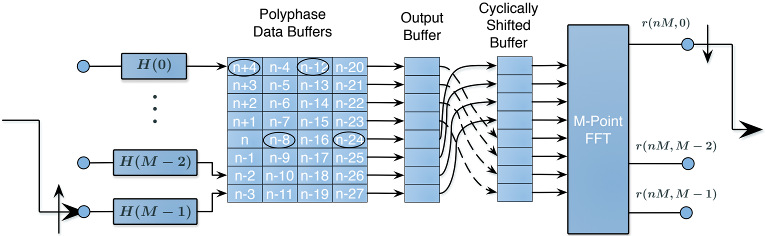
$$H(Z) = \sum_{r=0}^{M/2-1} Z^{-r} H_r(Z^M) + Z^{-(r+M/2)} H_{r+M/2}(Z^M)$$

where

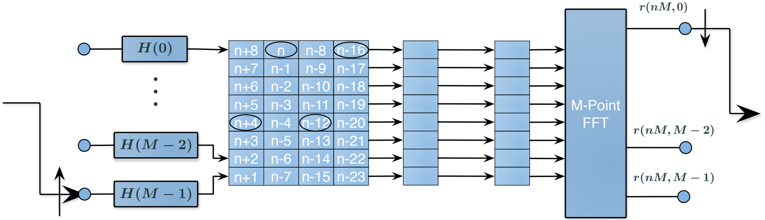
$$H_r(Z^M) = \sum_{n=0}^{(N/M)-1} h(r+nM) Z^{-nM}$$

- Delay in Polyphase filter bank effectively implements a circular shift of $M/2$
- This shifting of origin will need to be compensated before PFB outputs are sent to IFFT.

Channelizer Background : Origin Compensation

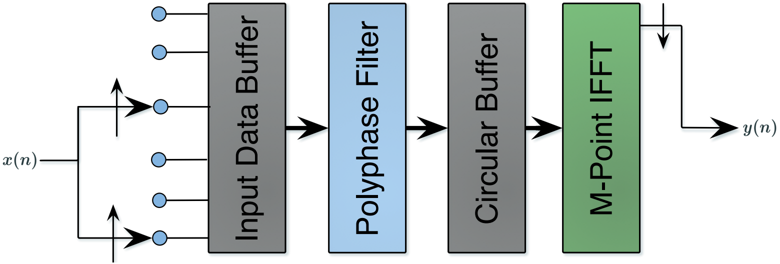


Channelizer Background : Origin Compensation



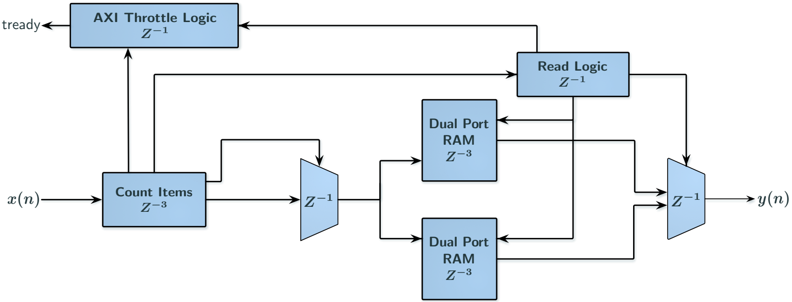
- Need to implement *Circular Shift*

Channelizer Background : System Diagram



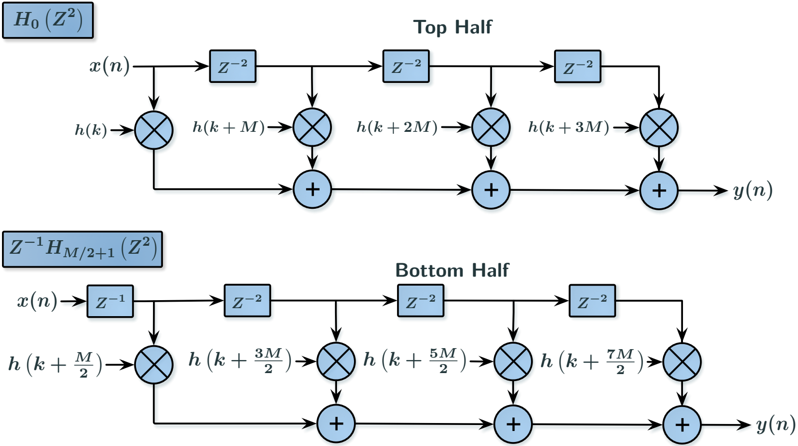
Hardware Implementation

Hardware Implementation : Input Buffer

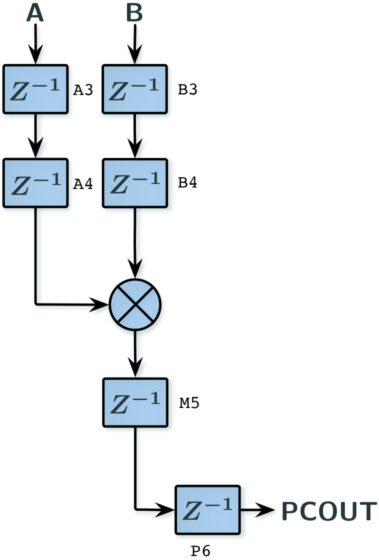


- Utilizes AXI flow to conform to the RFNoC interface.
- Provides a ping-pong buffer interface for continuous streaming.
- Each dual-port RAM is read twice. (Produces the time domain sequence required by the PFB)

Hardware Implementation : Polyphase Filter Bank

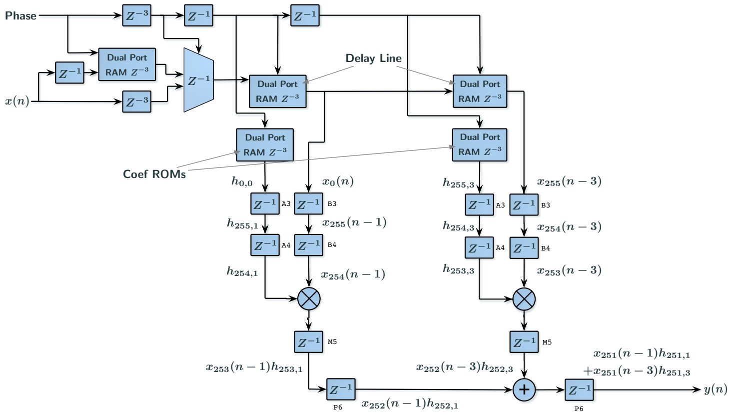


Hardware Implementation : DSP48



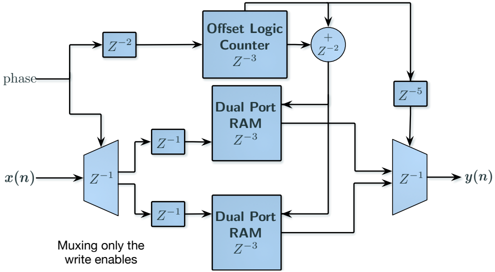
- Take advantage of Systolic Architecture of PFB.
- Fully pipeline DSP internals to maximize FMax.
- Utilize dedicated routing for optimal placement.

Hardware Implementation : PFB Final Implementation



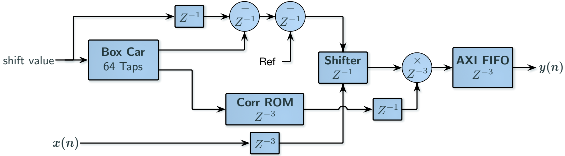
- Utilizes built in routing of Xilinx architecture
- Samples can be streamed continuously.
- Achieves high FMax value > 500 MHz.

Hardware Implementation : Circular Buffer



- Utilizes ping-pong buffering to allow streaming.
- Offset counter logic offsets the read pointer by $M/2$ samples every other block of M samples.

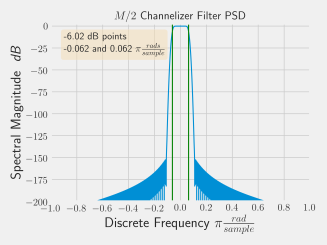
Hardware Implementation : Exp Shifter



- Using block floating point IFFT (common exponent for block of samples).
- Averages the exponent of 64 consecutive IFFT frames.
- Current block is shifted by the integer difference between the current exponent and the moving average.
- Enforces headroom in output to allow for *loud*, bursty transmissions.

Filter Generation

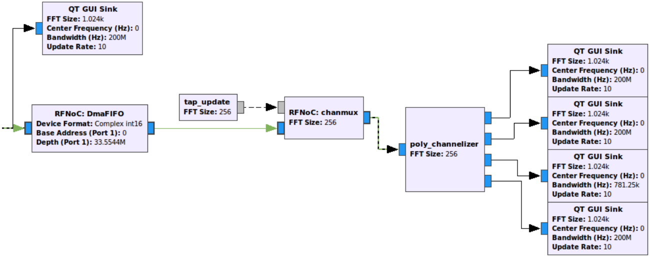
Filter Generation



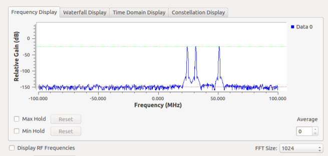
- Based on the work of Wessel Lubberhuizen found in [Lubberhuizen(2010)].
- Using root raised erf functions to perform filter coefficient calculation.
- Stable even for very narrow filter bandwidths.
- Simple Calculation.
- Not as flexible as Remez.
- Provides flat passband and low stopband with linear phase.

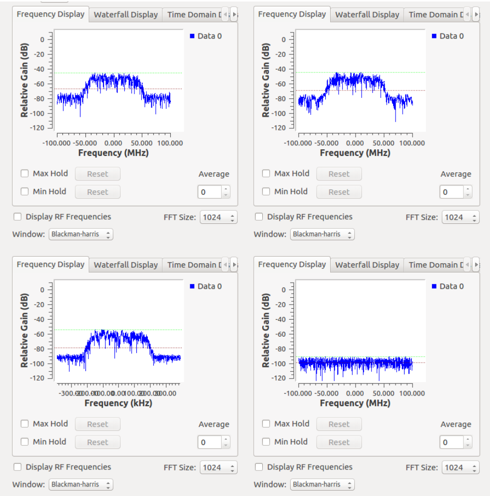
GNURadio Software Component / Results

GNURadio Software Component / Results



- This setup is using a 256 channel channelizer.
- Four sub-bands specified in the mask vector argument to *poly_channelizer* block.
- The *tap_update* python block configures the taps at start-up and when number of channels is adjusted.
- *chanmux* is the block controller for the RFNoC block.
- The *tap_update* is being converted to C++ and integrated in *chanmux*.


Frequency Display
Waterfall Display
Time Domain Display
Constellation Display
Data 0
Relative Gain (dB)
Frequency (MHz)
Max Hold
Min Hold
Reset
Average
Display RF Frequencies
FFT Size: 1024


Frequency Display
Waterfall Display
Data 0
Relative Gain (dB)
Frequency (MHz)
Frequency (kHz)
Max Hold
Min Hold
Reset
Average
Display RF Frequencies
FFT Size: 1024
Window: Blackman-harris

Future Work

Future Work

- Word sizes larger than 16 bit I/Q samples, current head space is diminishing DR.
- Narrow-band signals would also benefit.
- Second task is to convert the python code found in the *tap_update* module in to C++.

Questions?

References i

Harris F., McGwier R., Egg B.
A versatile multichannel filter bank with multiple channel bandwidths.
In *CrownCom 5th International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 1–5, Cannes, France, 2010.

Lubberhuizen, Wessel.
Near perfect reconstruction polyphase filterbank.
`http://www.mathworks.com/matlabcentral/fileexchange/15813-near-perfect-reconstruction-polyphase-filterbank`, 2010.