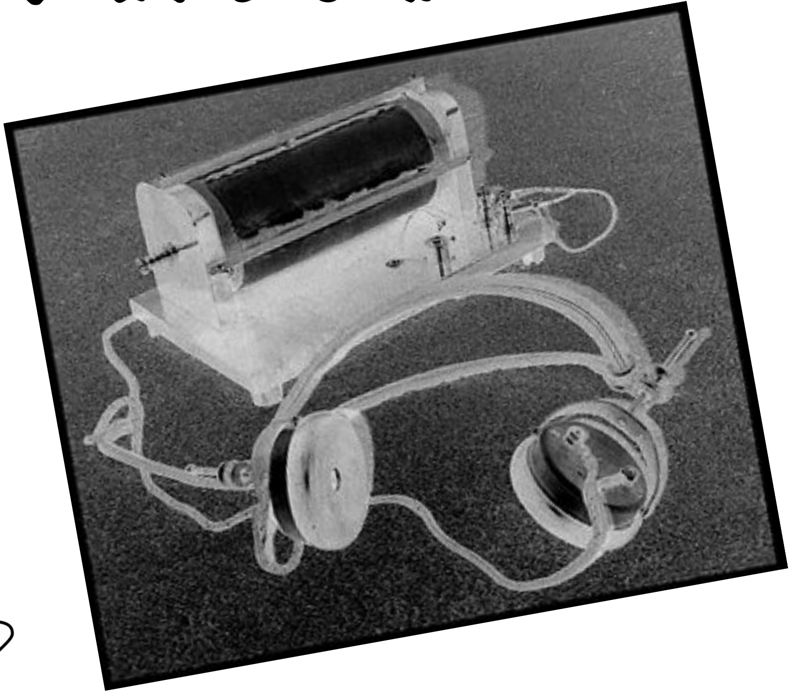
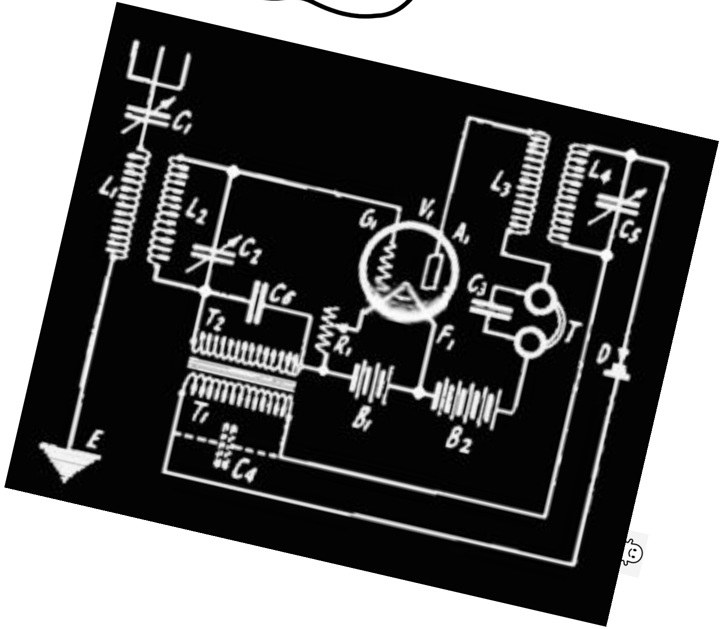
FPGA Software Defined Radio (SDR)

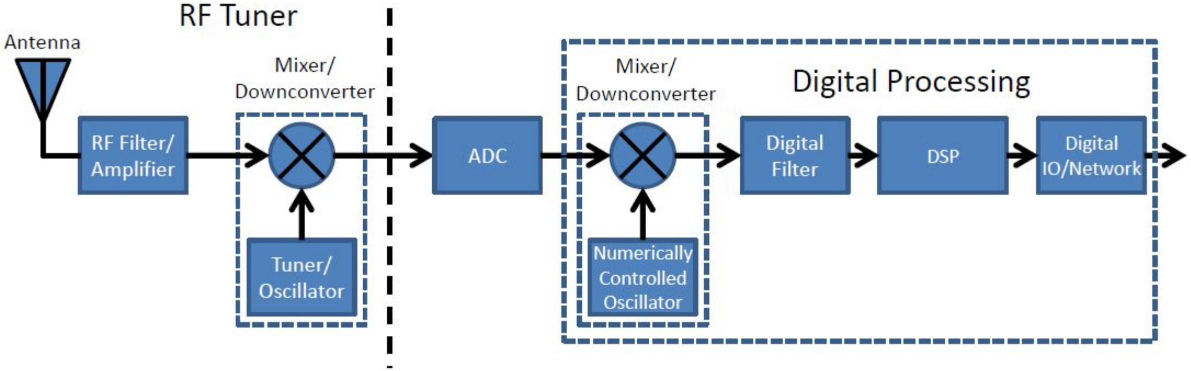
Colin Chaney and Charles Lindsay

THE 1920’S





Overview - What is a SDR?



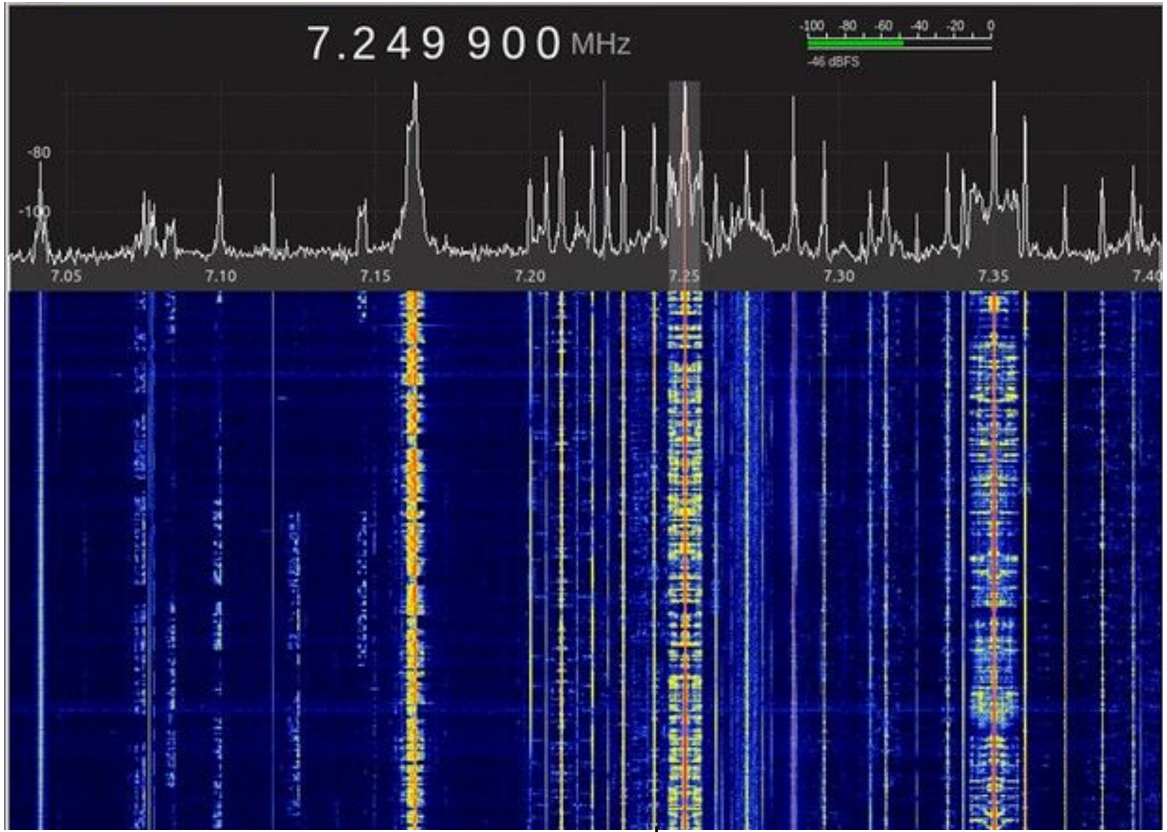
- Receive, demodulate and decode a signal in “real time” in the digital domain
- Allows for flexible radio configuration and interface with other digital systems

Our SDR

- RF simulated by function generator
- Demodulate AM and FM audio signals and play sound
- VGA monitor
  - FFT of received RF spectrum (0 - 500 kHz)
  - Audio signal

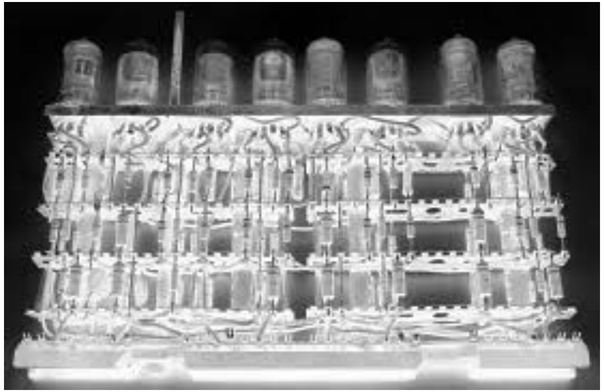
Stretch Goals

- Add front end to receive AM radio (540 - 1600 kHz)
- Waterfall display on VGA
- Demodulate FSK, BPSK, QPSK

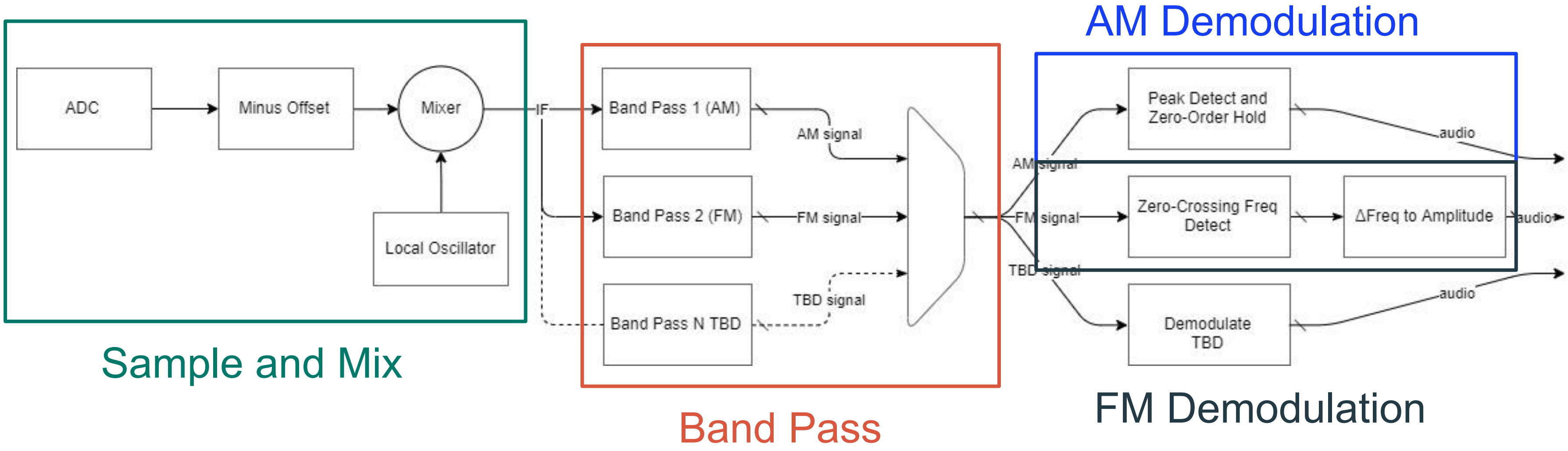


gqrx.dx

RF DSP



RF DSP Chain


AM Demodulation
ADC
Minus Offset
Mixer
Local Oscillator
IF
Band Pass 1 (AM)
Band Pass 2 (FM)
Band Pass N TBD
AM signal
FM signal
TBD signal
AM signal
FM signal
TBD signal
Peak Detect and Zero-Order Hold
Zero-Crossing Freq Detect
ΔFreq to Amplitude
Demodulate TBD
audio
audio
audio
Sample and Mix
Band Pass
FM Demodulation

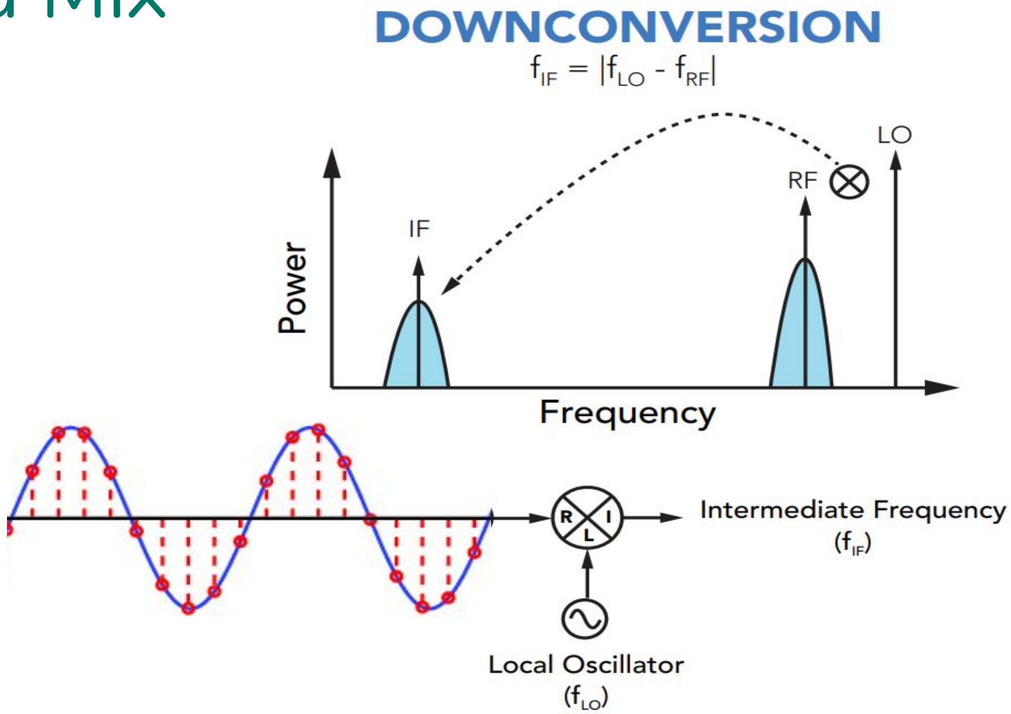
Sample and Mix

Nyquist freq = 500 KHz

- Limited by Nexys 4 DDR ADC
- With front end could be 5 MHz

IF = 455 KHz

Local oscillator synthesized on FPGA



medium.com

Band Pass

AM FIR

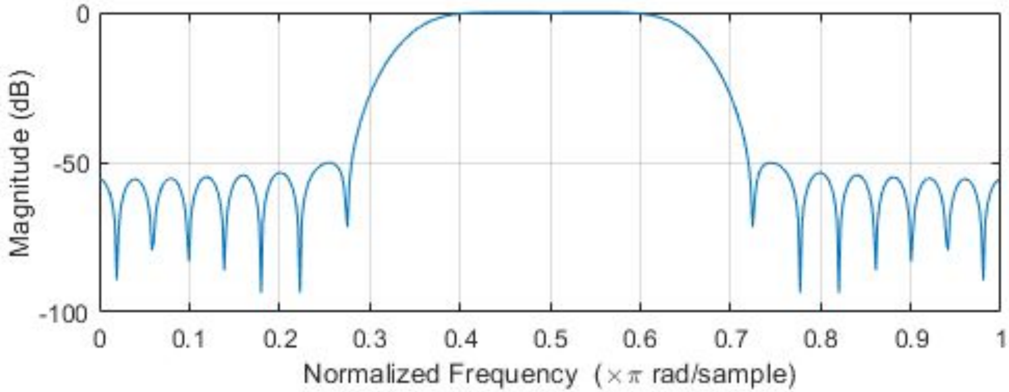
- f_c = 455 kHz (IF)
- f_m = 5 kHz

FM FIR

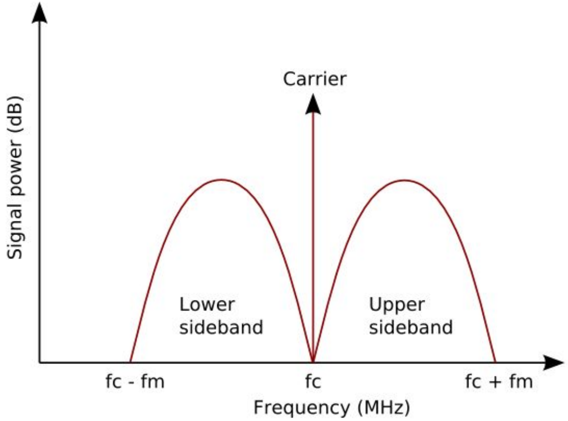
- f_c = 455 kHz (IF)
- f_m = 8 kHz
    - Carson Bandwidth Rule, for signal with peak Δf = 5 kHz & max audio = 3 kHz

Wide band FM (radio broadcast)
FM radio stations in 88 - 108 MHz range have BW > 200 kHz and multiple sidebands

FIR bandpass filter response



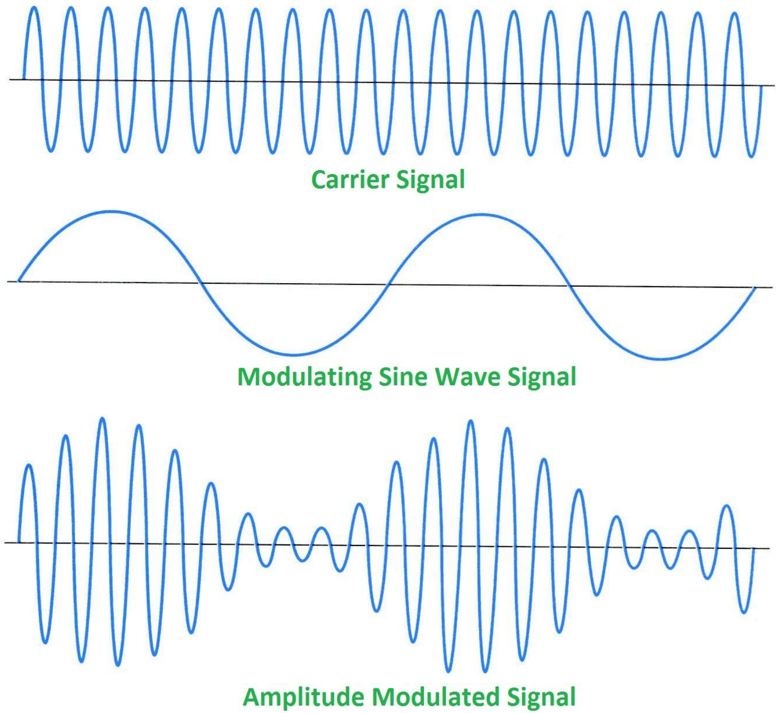
AM/FM Spectrum

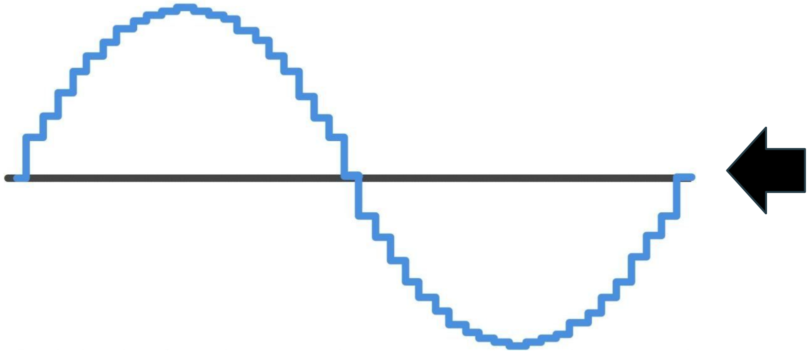


AM Demodulation

Peak Detection

- Results in reconstructed digitized audio signal sampled at 455 kHz
- Will be down sampled to generate 48 kHz audio





FM Demodulation

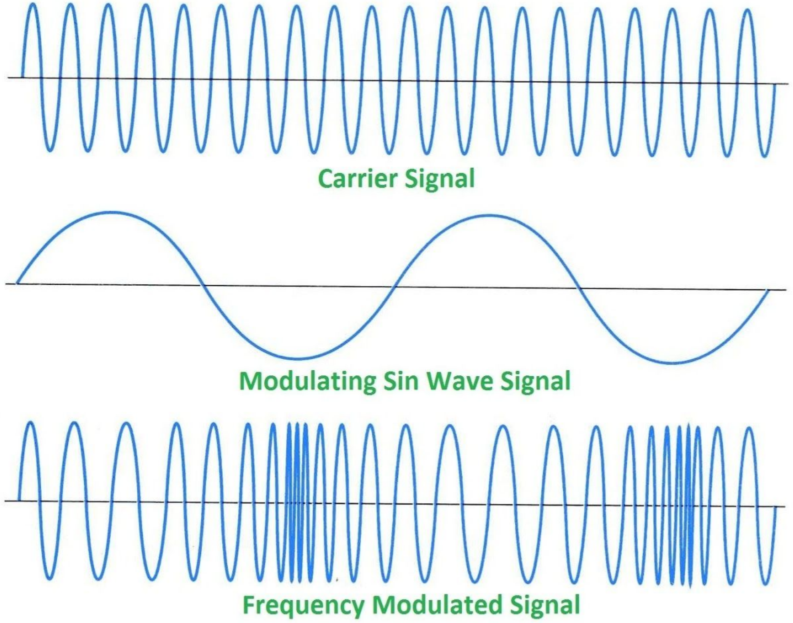
Frequency Detect

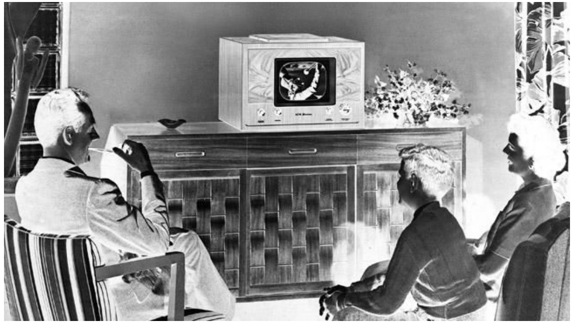
- Identify period between zero crossings to detect instantaneous freq = $f_s(t)$

$$f_s(t) = f_c(t) + k \cdot s(t)$$

- Convert $f_s(t)$ to audio s(t)

$$s(t) = \frac{f_s(t) - f_c(t)}{k}$$





TOO SPOOKY

VGA Output

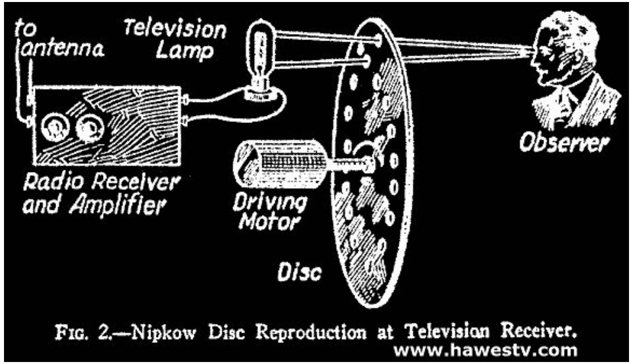


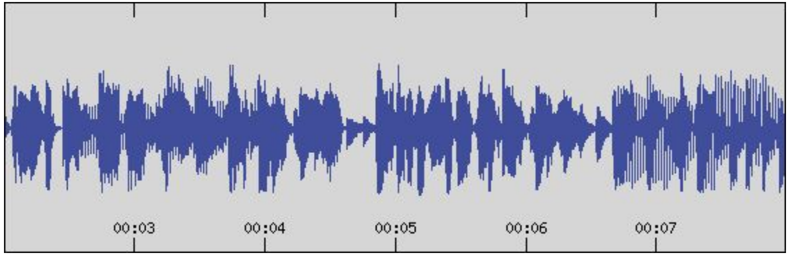
FIG. 2.—Nipkow Disc Reproduction at Television Receiver.
www.hawestv.com

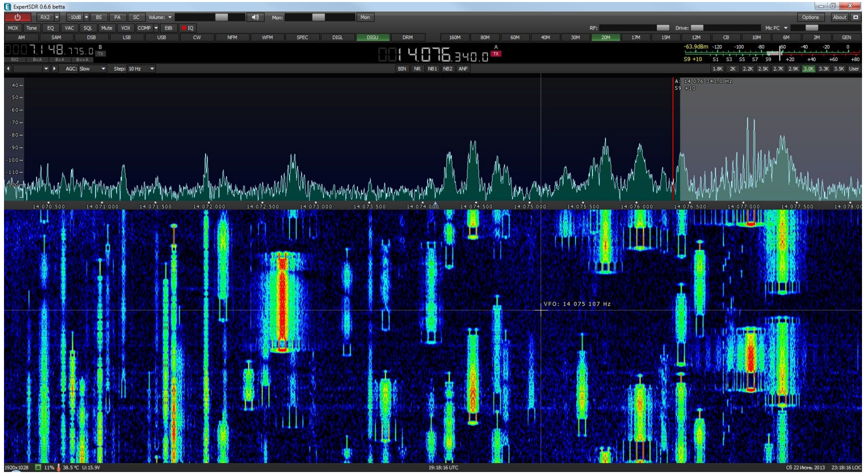
VGA Options

1-) Display the Audio Waveform

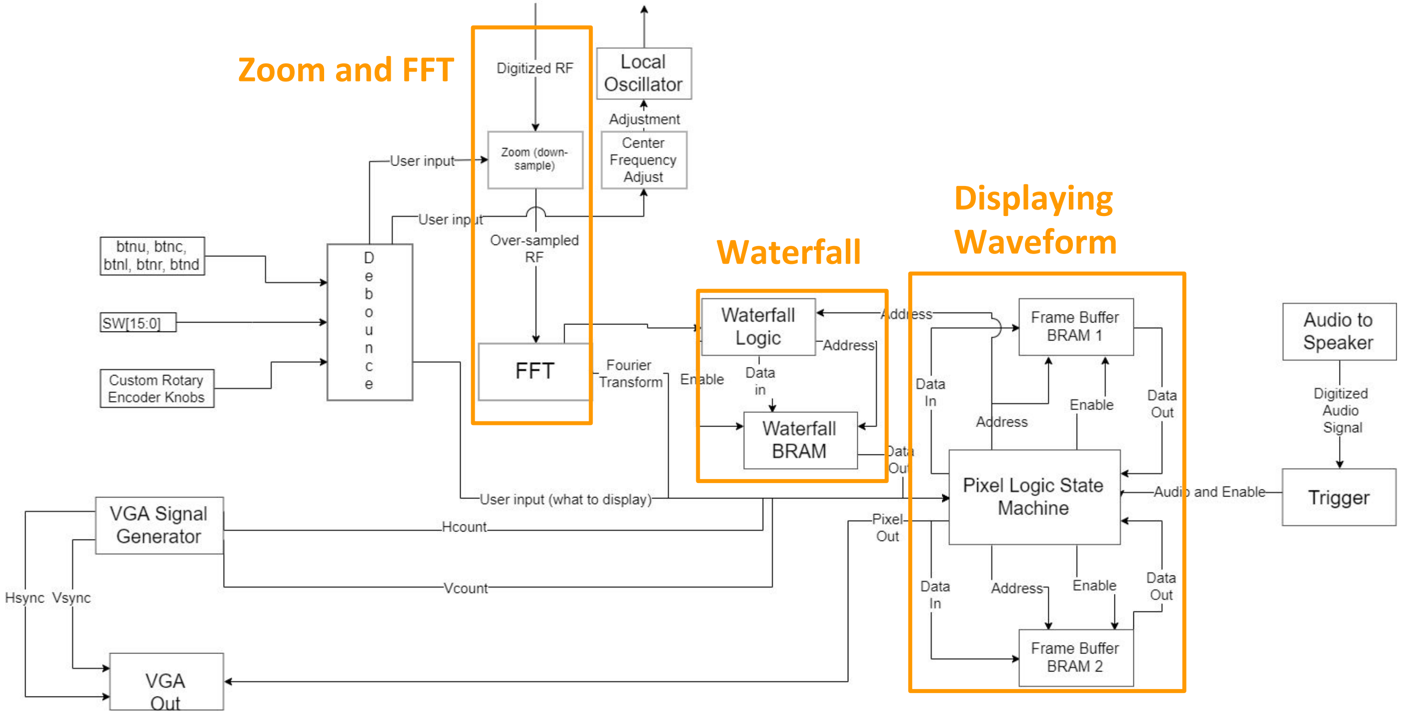
2-) Display full frequency spectrum

3-) Narrow band frequency spectrum with waterfall display



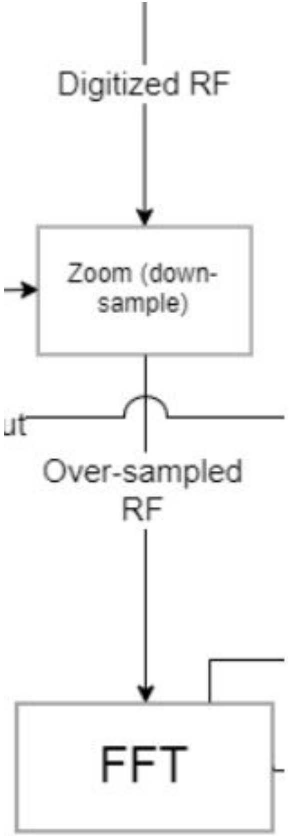


FFT and VGA Out



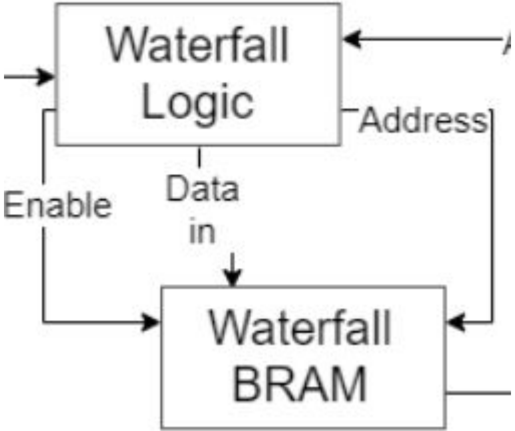
Zoom and FFT

- Zoom
  - Limit the FFT frequency by oversampling
  - Change FFT magnitudes we display, depending on desired user range
- FFT
  - Used to display the frequency spectrum of our signal
  - Shows the different radio signals we can demodulate
  - Work with module given to us on piazza

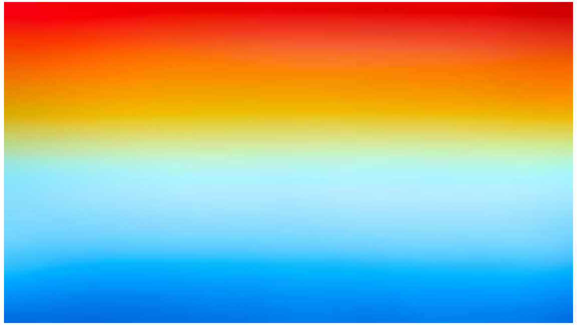


Waterfall Display

- Stores Magnitude of FFT over time
  - Shows how signals grow and decay
- Have a color gradient to display how strong the signal is
- Most memory intensive part of the system
  - Reduce memory usage by limiting to our IF + Bandwidth
  - Can also reduce memory usage by limiting how far back we look at data



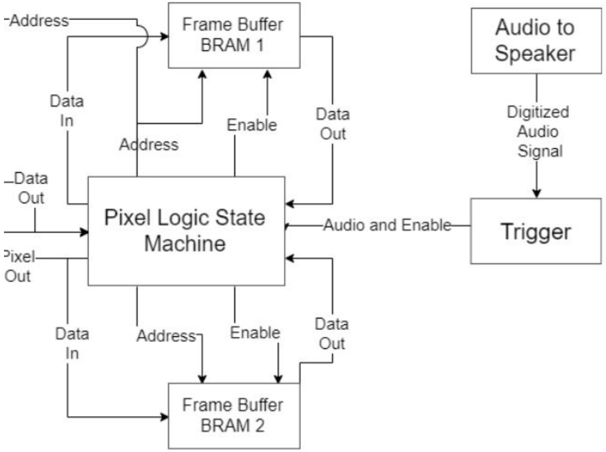
1 magnitude



0 magnitude

Displaying the Audio Waveform

- Most complex module
- Will display different waveforms depending on user input
- Trigger from audio input to make signal stay the same
- Has two BRAMs for frame buffering
- Display from centerline with adjustable height



Address

Frame Buffer BRAM 1

Audio to Speaker

Data In

Address

Enable

Data Out

Digitized Audio Signal

Data Out

Pixel Logic State Machine

Audio and Enable

Trigger

Pixel Out

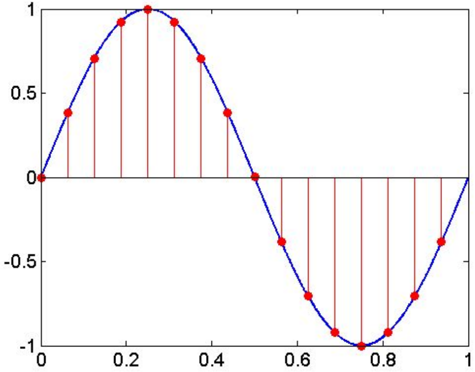
Data In

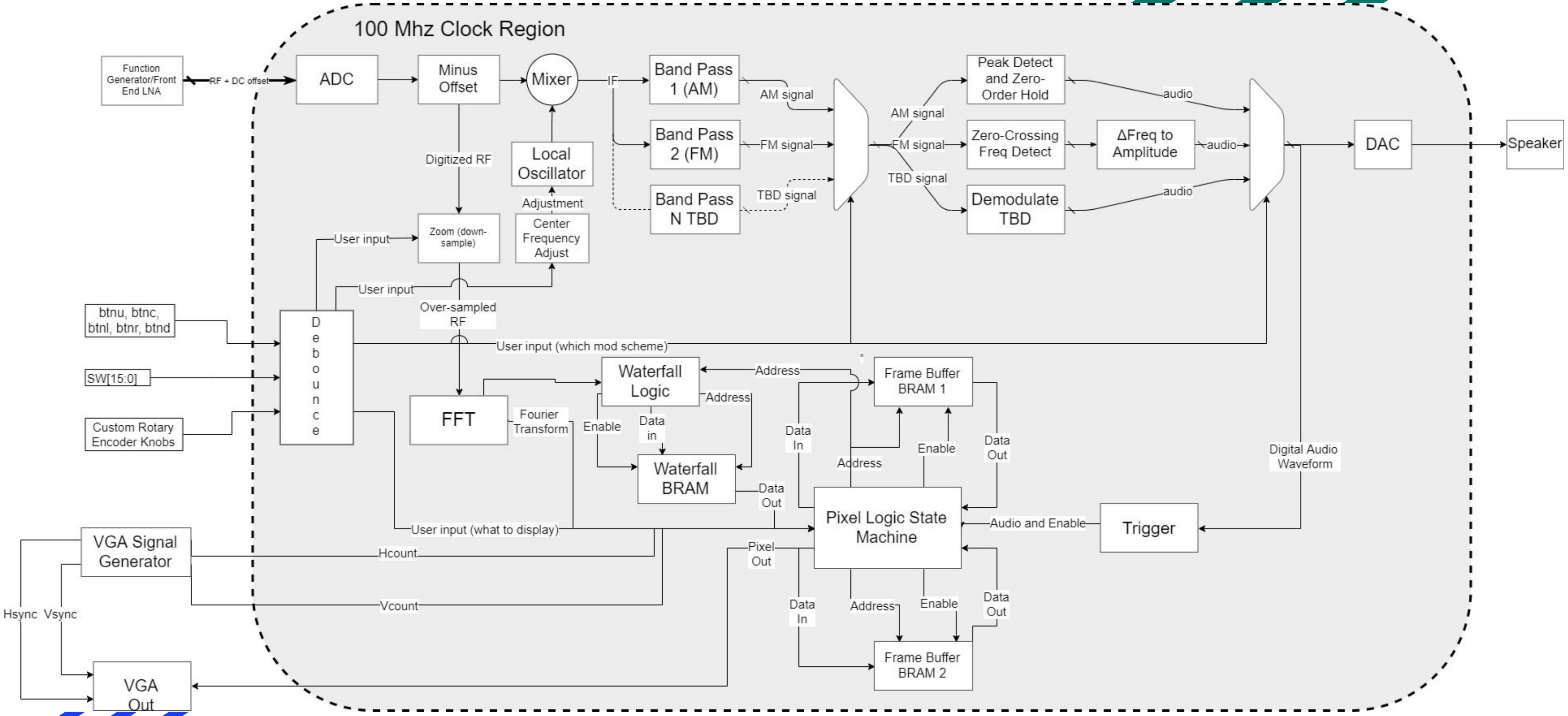
Address

Enable

Data Out

Frame Buffer BRAM 2





100 Mhz Clock Region

Function Generator/Front End LNA

RF + DC offset

ADC

Minus Offset

Mixer

IF

Band Pass 1 (AM)

Band Pass 2 (FM)

Band Pass N TBD

AM signal

FM signal

TBD signal

Peak Detect and Zero-Order Hold

Zero-Crossing Freq Detect

ΔFreq to Amplitude

Demodulate TBD

audio

DAC

Speaker

Digitized RF

Local Oscillator

Adjustment

Center Frequency Adjust

Zoom (down-sample)

User input

Over-sampled RF

btnu, btnc, btnl, btnr, btnd

SW[15:0]

Custom Rotary Encoder Knobs

Debounce

User input (which mod scheme)

FFT

Fourier Transform

Waterfall Logic

Address

Enable

Data in

Waterfall BRAM

Data Out

Frame Buffer BRAM 1

Data In

Enable

Data Out

Pixel Logic State Machine

Audio and Enable

Trigger

Digital Audio Waveform

User input (what to display)

VGA Signal Generator

Hcount

Vcount

Pixel Out

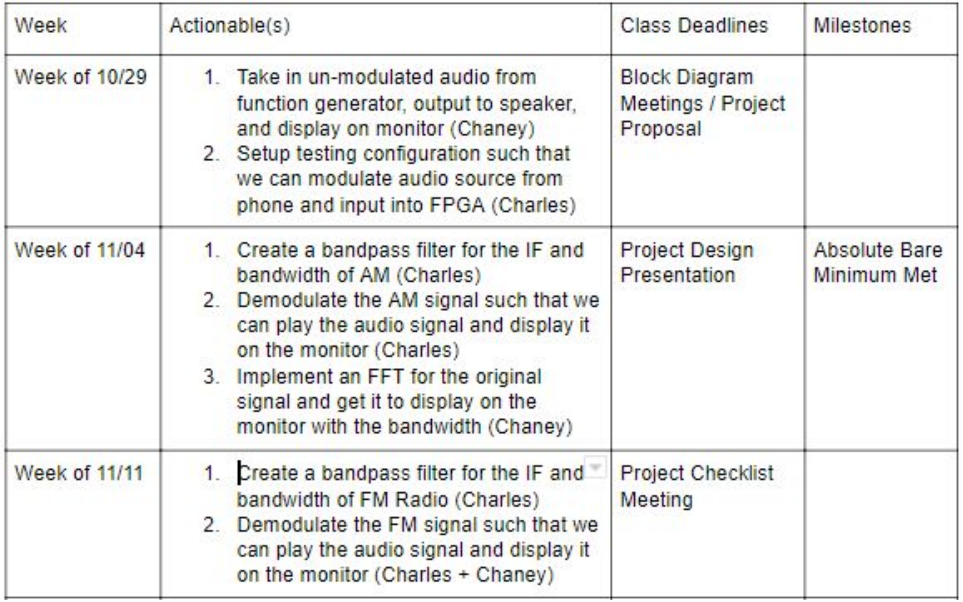
Hsync

Vsync

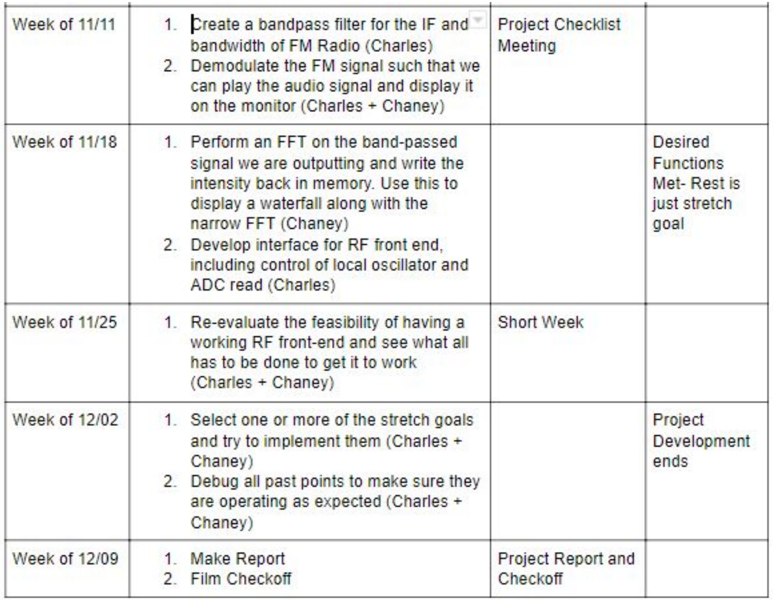
VGA Out

Frame Buffer BRAM 2

Timeline



Week	Actionable(s)	Class Deadlines	Milestones
Week of 10/29	1. Take in un-modulated audio from function generator, output to speaker, and display on monitor (Chaney) 2. Setup testing configuration such that we can modulate audio source from phone and input into FPGA (Charles)	Block Diagram Meetings / Project Proposal	
Week of 11/04	1. Create a bandpass filter for the IF and bandwidth of AM (Charles) 2. Demodulate the AM signal such that we can play the audio signal and display it on the monitor (Charles) 3. Implement an FFT for the original signal and get it to display on the monitor with the bandwidth (Chaney)	Project Design Presentation	Absolute Bare Minimum Met
Week of 11/11	1. Create a bandpass filter for the IF and bandwidth of FM Radio (Charles) 2. Demodulate the FM signal such that we can play the audio signal and display it on the monitor (Charles + Chaney)	Project Checklist Meeting	



Week of 11/11	1. Create a bandpass filter for the IF and bandwidth of FM Radio (Charles) 2. Demodulate the FM signal such that we can play the audio signal and display it on the monitor (Charles + Chaney)	Project Checklist Meeting	
Week of 11/18	1. Perform an FFT on the band-passed signal we are outputting and write the intensity back in memory. Use this to display a waterfall along with the narrow FFT (Chaney) 2. Develop interface for RF front end, including control of local oscillator and ADC read (Charles)		Desired Functions Met- Rest is just stretch goal
Week of 11/25	1. Re-evaluate the feasibility of having a working RF front-end and see what all has to be done to get it to work (Charles + Chaney)	Short Week	
Week of 12/02	1. Select one or more of the stretch goals and try to implement them (Charles + Chaney) 2. Debug all past points to make sure they are operating as expected (Charles + Chaney)		Project Development ends
Week of 12/09	1. Make Report 2. Film Checkoff	Project Report and Checkoff	