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
RF DSP
RF DSP Chain

Sample and Mix

Band Pass

AM Demodulation

- ADC
- Minus Offset
- Mixer
- Local Oscillator

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

audio

audio
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
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 $\Delta f = 5$ 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
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 $s(t)$

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

Waterfall

Displaying Waveform

Audio to Speaker

Trigger
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
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
# Timeline

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

## Week of 11/11
- Create a bandpass filter for the IF and bandwidth of FM Radio (Charles)<br>- Demodulate the FM signal such that we can play the audio signal and display it on the monitor (Charles + Chaney)<br>- Project Checklist Meeting<br>- Desired Functions Met - Rest is just stretch goal

## Week of 11/18
- 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)<br>- Develop interface for RF front end including control of local oscillator and ADC read (Charles)<br>- Short Week

## Week of 11/25
- 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)<br>- Project Development ends

## Week of 12/02
- Select one or more of the stretch goals and try to implement them (Charles + Chaney)<br>- Debug all past points to make sure they are operating as expected (Charles + Chaney)<br>- Project Report and Checkoff

## Week of 12/09
- Make Report<br>- Film Checkoff