BeRadio
BeRadio

FPGA-based Software Defined Radio
Introduction and Objectives

- Introduction and SDR definitions
- Example SDR architectures
- SDR RF and DSP hardware considerations
- SDR DSP software considerations
- BeRadio hardware and FPGA firmware
- Higher performance options
- Compile/Load/Test an FPGA-based SDR design
What is an SDR?

**Software Defined Radio**

An SDR in general is a radio that has

- Primary functionality (mod/demodulation, filtering, etc) defined in firmware and/or software, the digital domain.
- DSP algorithms implemented in configurable hardware and/or PC software

An SDR is **NOT**

- A computer-controlled conventional radio
- A conventional radio with a GUI integrated into its front panel
- A SuperHet receiver with Audio Frequency -DSP
Advantages & Benefits of SDR

• Radio’s physical layer behavior is primarily defined in software
  • Reduce parts inventory
• Supports a broad range of frequencies, air interfaces, and application SW
  • Easily adapts to different network
• Implementing multiple radio standards
  • Flexible modulation formats & protocols
• Changing configurations to satisfy user requirements.
• An FPGA-based SDR platform can also reduce the high development costs of a full new ASIC-based solution
  • Exploit declining prices in computing components
SDR Examples

RFspace SDR-IQ™ Receiver

- 500 Hz – 30MHz
- 14-bit @ 67MSPS ADC
- Max Display BW 196 kHz
- USB 2.0 Full Speed (12Mb/s)
SDR Examples

Elbit Systems SDR-7200

• 30-512Mhz continuous frequency coverage
• RX: 14-bit @ 105MSPS ADC,
• Max data rates 1Msp/s over a single channel
• GPS built in
• Interfaces: Ethernet, USB, RS-232, Audio
Architecture Definitions

**DDC**: Digital Down Conversion

Sampling of the RF input using an ADC as a receiver

**DUC**: Digital Up Conversion

Generation of the RF waveform using a DAC as a transmitter
Architecture: DDC
Digital Down Conversion

- Digital Sampling configuration, Receive (DDC)
Architecture: DUC
Digital Up Conversion

- Digital Sampling configuration, Transmit (DUC)
Focus will be on Digital Down Conversion (DDC) receivers for the rest of this presentation.
SDR Front-end Considerations

- Analog Front End:
  - Attenuation
  - Amplification
- ADC
  - Selection
  - Clocking
  - Input Drive
- PCB layout considerations
SDR Front End: Attenuation & Amplification

- **Input Attenuation**
  - Helps prevent ADC overload
  - Can be switched in and out
  - Multiple small steps are practical

- **Input Amplification**
  - Improves Minimum Discernible Signal (MDS) at the expense of ADC overload
  - Can combine with ADC driver
  - Can improve receiver Noise Figure
    - Use shunt capacitor to form noise filter
  - **Don’t overdo it!** avoid over amplification, ADC clipping the signal
SDR Front End: ADC Selection

- Number of Bits
- Sample rate
- Analog Front-end bandwidth
- Noise floor, SFDR and jitter
- Clock input requirements
- Output type: serial, parallel, LVDS, CMOS
- Digital Feedback Reduction:
  - Alternate bit polarity, Randomizer, duty-cycle stabilization
ADC Selection: Baseline of terms

- **SFDR**: Spurious Free Dynamic Range
  - Ratio of the RMS value of the carrier frequency (max signal component) at the input of the ADC to the RMS value of the next largest noise or harmonic distortion component (referred to as a “spurious” or a “spur”) at its output.
  - SFDR measured in dBc (i.e. with respect to the carrier frequency amplitude) or in dBFS (i.e. with respect to the ADC's full-scale range).

- **SNR**: Signal to Noise Ratio
  - The ratio of the amplitude of the desired signal to the amplitude of noise signals at a given point in time.
  - Ideal SNR = 6.02N + 1.76dB
    
    Where N = number of bits of resolution
ADC Selection: Baseline of terms

- **DNL**: Differential Non Linearity
  - An ideal ADC output is divided into $2^n$ uniform steps. Any deviation from the ideal step width is the DNL.

- **INL**: Integral Non Linearity
  - The maximum deviation from the ideal slope of the ADC and is measured from the center of the step.
ADC Selection : Baseline of terms

- **OVERSAMPLING**: sampling a signal with a frequency significantly higher than twice the bandwidth or highest frequency of the signal being sampled.

- **UNDERSAMPLING** or **BANDPASS SAMPLING**: sample a bandpass filtered signal at a sample rate below the usual Nyquist rate but is still able to reconstruct the signal.
ADC Selection: Sample Rate

- We must block all input frequencies above $f_{\text{nyquist}}$.
- In practice, we oversample and filter.
  - Ideal SNR = $6.02N + 1.76\text{dB}$
  - $\text{SNR Improvement (dB)} = 10 \times \log \left( \frac{\text{Sampling Rate}}{2 \times \text{Signal Bandwidth}} \right)$
  - For a given input bandwidth, doubling the sampling frequency can increase the SNR by 3dB, and increase effective bits by $\frac{1}{2}$ bit.

- BeRadio LPF corner = 2.0MHz
- BeRadio LPF is 5 pole or $\sim30\text{dB/octave}$ rolloff
  - The attenuation at 5Mhz is $\sim39.8\text{dB}$
- Oversampling compensates for less-than-ideal filters
ADC Selection: Front-end bandwidth

- Determines the number of Nyquist zones useable when under sampling
- **Nyquist frequency** = half-sampling frequency (fs/2) is often called the “folding frequency”
- The front end BW of the LTC2225 for BeRadio is 575MHz
  - (115 Nyquist zones @10Mmps)
ADC Selection: Clocking

• Most important
• Minimize clock:
  • Phase noise
  • Jitter
  • Reflections
  • Digital noise pickup
  • Deviation from 50% duty cycle < (+/- 10%)
• Treat the clock as analog input
**ADC Selection: Clock Jitter**

- Clock jitter: amount of timing variability of the edge of the ADC sampling clock from cycle to cycle.

- **Example:**

  ADC clock of 200Mhz (5 nsec period)

  5 psec of jitter

  Jitter is 0.1% of the clock period

  Max SNR \( \text{Jitter} = 20 \log \left( \frac{1}{2\pi \cdot F_{\text{Signal}} \cdot t_{\text{Jitter}}} \right) \)

  60MHz signal:

  \[ \text{SNR}_{\text{Jitter}} = 20 \log \left( \frac{1}{2\pi \cdot 60 \cdot 10^6 \cdot 5 \cdot 10^{-12}} \right) = -54.5 \text{ dB} \]

  ADC performance at 60MHz is limited to 10 bits (6dB / bit)

  460MHz signal:

  \[ \text{SNR}_{\text{Jitter}} = 20 \log \left( \frac{1}{2\pi \cdot 460 \cdot 10^6 \cdot 5 \cdot 10^{-12}} \right) = -36.8 \text{ dB} \]

  ADC performance at 460MHz is limited to 6 bits (6dB / bit)
ADC Selection: Digital Feedback in ADCs

Digital Outputs couple into Analog Inputs
- Capacitive coupling
- Inductive coupling
- Ground currents

Output Code

| 00 0000 0000 0011 |
| 00 0000 0000 0010 |
| 00 0000 0000 0001 |
| 00 0000 0000 0000 |
| 11 1111 1111 1111 |
| 11 1111 1111 1110 |
| 11 1111 1111 1101 |

Worse for Small Signals!

Signal vs. Time

All Bits Change at Mid-Scale
ADC Selection: Alternate Bit Polarity Mode

Reduces Digital Feedback by Canceling Ground Plane Currents for Small Input Signals

Output Code

10 1010 1010 1001
10 1010 1010 1000
10 1010 1010 1011
10 1010 1010 1010
01 0101 0101 0101
01 0101 0101 0100
01 0101 0101 0111

Signal vs. Time

7x

7 Bits 0 to 1
7 Bits 1 to 0
at Mid-Scale

LTC proprietary technology
ADC Selection: Alternate Bit Polarity

Decoding Alternate Bit Polarity Mode Data

LTC2262-14

D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0

Encoded Data

FPGA or ASIC

D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0

DECODED DATA
ADC Selection: Randomize

Optional Digital Output Randomizer

The digital randomizer reduces digital noise and produces better SFDR performance

Data Bits are Exclusive OR’ed with the LTC2262’s LSB

To decode, the FPGA must apply the same XOR operation
ADC Selection:
Alternate Bit Polarity and Randomizer

LTC2261-14, 125Msps, Ain = 70MHz, -65dBFS
Averaged 128k Point FFTs

Improves SFDR of ADC by 10-15dB

ABP = Off, RAND = Off

ABP = On, RAND = On
# ADC Selection:
## LTC High Speed ADC for SDR’s

<table>
<thead>
<tr>
<th>Product</th>
<th>Bits</th>
<th>Speed (MHz)</th>
<th>Channels</th>
<th>SINAD (dB)</th>
<th>SFDR (dBFS)</th>
<th>Power (mW)</th>
<th>I/O</th>
<th>ABP</th>
<th>Randomizer</th>
<th>Dither</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTC2208</td>
<td>16</td>
<td>130</td>
<td>1</td>
<td>77.7</td>
<td>100</td>
<td>1250</td>
<td>Parallel CMOS, Parallel LVDS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LTC2217</td>
<td>16</td>
<td>105</td>
<td>1</td>
<td>81.2</td>
<td>100</td>
<td>1190</td>
<td>Parallel CMOS, Parallel LVDS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LTC2274</td>
<td>16</td>
<td>105</td>
<td>1</td>
<td>77.6</td>
<td>100</td>
<td>1300</td>
<td>Serial JESD204</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>LTM9001-AA</td>
<td>16</td>
<td>130</td>
<td>1</td>
<td>72</td>
<td>82</td>
<td>1650</td>
<td>Parallel CMOS, Parallel LVDS</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>LTC2195</td>
<td>16</td>
<td>125</td>
<td>2</td>
<td>76.8</td>
<td>90</td>
<td>432</td>
<td>Serial LVDS</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LTC2185</td>
<td>16</td>
<td>125</td>
<td>2</td>
<td>76.8</td>
<td>90</td>
<td>370</td>
<td>Parallel CMOS, Parallel DDR CMOS, Parallel DDR LVDS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LTC2165</td>
<td>16</td>
<td>125</td>
<td>1</td>
<td>76.8</td>
<td>90</td>
<td>194</td>
<td>Parallel CMOS, Parallel DDR CMOS, Parallel DDR LVDS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LTC2157-14</td>
<td>14</td>
<td>250</td>
<td>2</td>
<td>70</td>
<td>90</td>
<td>650</td>
<td>DDR LVDS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LTC2262-14</td>
<td>14</td>
<td>150</td>
<td>1</td>
<td>72.8</td>
<td>88</td>
<td>149</td>
<td>Parallel CMOS, Parallel DDR CMOS, Parallel DDR LVDS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LTC2145-14</td>
<td>14</td>
<td>125</td>
<td>2</td>
<td>73</td>
<td>90</td>
<td>189</td>
<td>Parallel CMOS, Parallel DDR CMOS, Parallel DDR LVDS</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>LTC2175-14</td>
<td>14</td>
<td>125</td>
<td>4</td>
<td>73.4</td>
<td>88</td>
<td>558</td>
<td>Serial LVDS</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>LTC2268-14</td>
<td>14</td>
<td>125</td>
<td>2</td>
<td>73.4</td>
<td>88</td>
<td>299</td>
<td>Serial LVDS</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
ADC Selection:

PCB Layout Considerations

- Ground plane required under signal layers: no two layer PCBs
- Separate analog and digital power with filters
- Stitch ground layers together with *many* vias
- Keep ADC outputs isolated from clock input
- Physical symmetry at input is very important
- Provide return paths for all current
- Top-layer flood >3x distance to ground plane
- Use LTC demo board layout as a starting point

Even at a low sample rate, ADC may respond to >GHz
Hardware DSP Considerations

• Sample rate and sample data width
• Decimation value
• Hardware DSP: filtering, demodulation
• Estimating FPGA resource requirements
• Selecting an FPGA
• Selecting a data interface to the PC
**Hardware DSP:** Baseline of terms

- **DECIMATION:** Combined operation of filtering and down sampling
  - reduce the sampling rate at the output of one system to match the input to a system operating at a lower sampling rate
  - A common motivation for decimation is reduced computational complexity, i.e. lower cost

- **FIR:** Finite Impulse Response:
  - a filter whose impulse response (or response to any finite length input) is of *finite* duration. The response settles to zero in finite time
  - FIR filters have a Linear Phase Response
Hardware DSP: Baseline of terms

• **CIC**: Cascaded Integrator Comb filter
  
  • CIC digital filters are computationally efficient implementations of hardware-efficient linear phase finite impulse response (FIR) digital filters.
  
  • CIC filters achieve sampling rate decrease (decimation) and sampling rate increase (interpolation) without using multipliers.
Hardware DSP: Data Width

- Dynamic range is determined by width and decimation
- Sampling dynamic range = ~6dB per bit
- Conversion gain = ~3dB per decimation by 2
- BeRadio sampling width = 12 bits
- BeRadio $f_s = 10$MHz
- BeRadio decimation = 200

approximate BeRadio dynamic range = 
6dB/bit * 12bits + 10\log_{10}(200) = 72 + 23 = 95$dB

Only in Theory!
Hardware DSP: Filtering and Demodulation

Split between FPGA firmware and software (PC)

**FPGA**
- Sample rate reduction
- Filtering
- Managing data transfer (packetization, formatting)

**Software**
- Filtering
- Demodulation
- User interface
Hardware DSP: 
Estimating FPGA Requirements

- Block out IP: DDR controller, Ethernet MAC, FFT, NCO
- Need transceivers, differential pairs, PLLs?
- I/O pin count will likely not be the critical factor
- If in doubt, choose larger device
- Use off-the-shelf evaluation boards for quick-start
  - BeMicro SDK + BeRadio
  - BeMicro SDK + UDPSDR-HF1
  - BeMicro SDK + UDPSDR-HF2/UDPSSDR-TX2
- Reduce device size after firmware is written
- Leave room for new features ← very important
Hardware DSP: Altera’s 28nm Product Portfolio

More Products Than Any Other Prior Node
## Hardware DSP: Selecting an FPGA

### Cyclone Low Cost FPGA, maximums

<table>
<thead>
<tr>
<th>Family</th>
<th>Geometry</th>
<th>LE</th>
<th>Mem</th>
<th>18x18</th>
<th>DSP</th>
<th>Diff</th>
<th>Xcvr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cyclone III</td>
<td>65nm</td>
<td>120K</td>
<td>7Mb</td>
<td>288</td>
<td>0</td>
<td>233</td>
<td>no</td>
</tr>
<tr>
<td>Cyclone III LS</td>
<td>65nm</td>
<td>200K</td>
<td>15Mb</td>
<td>396</td>
<td>0</td>
<td>181</td>
<td>no</td>
</tr>
<tr>
<td>Cyclone IV E</td>
<td>60nm</td>
<td>115K</td>
<td>7Mb</td>
<td>266</td>
<td>0</td>
<td>230</td>
<td>no</td>
</tr>
<tr>
<td>Cyclone IV GX*</td>
<td>60nm</td>
<td>150K</td>
<td>12Mb</td>
<td>360</td>
<td>0</td>
<td>216</td>
<td>8</td>
</tr>
<tr>
<td>Cyclone V E</td>
<td>28nm</td>
<td>301K</td>
<td>14Mb</td>
<td>342</td>
<td>144</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Cyclone V GX</td>
<td>28nm</td>
<td>301K</td>
<td>14Mb</td>
<td>342</td>
<td>144</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Cyclone V GT</td>
<td>28nm</td>
<td>301K</td>
<td>14Mb</td>
<td>342</td>
<td>144</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Cyclone V SE*</td>
<td>28nm</td>
<td>110K</td>
<td>6Mb</td>
<td>112</td>
<td></td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>Cyclone V SX*</td>
<td>28nm</td>
<td>110K</td>
<td>6Mb</td>
<td>112</td>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Cyclone V ST*</td>
<td>28nm</td>
<td>110K</td>
<td>6Mb</td>
<td>112</td>
<td></td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

*With some variant of Hard IP (PCIe, Memory controller, and/or HPS ARM processor)
## Hardware DSP: Selecting an FPGA

### Arria Midrange FPGA, maximums

<table>
<thead>
<tr>
<th>Family</th>
<th>Geometry</th>
<th>LE</th>
<th>Mem</th>
<th>18x18</th>
<th>DSP</th>
<th>PCIe</th>
<th>Diff</th>
<th>Xcvr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arria II GX</td>
<td>40nm</td>
<td>256K</td>
<td>12Mb</td>
<td>736</td>
<td>0</td>
<td>1</td>
<td>144</td>
<td>16</td>
</tr>
<tr>
<td>Arria II GZ</td>
<td></td>
<td>349K</td>
<td>16Mb</td>
<td>1040</td>
<td>0</td>
<td>1</td>
<td>179</td>
<td>24</td>
</tr>
<tr>
<td>Arria V GX</td>
<td>40nm</td>
<td>256K</td>
<td>12Mb</td>
<td>736</td>
<td>0</td>
<td>1</td>
<td>144</td>
<td>16</td>
</tr>
<tr>
<td>Arria V GZ</td>
<td></td>
<td>349K</td>
<td>16Mb</td>
<td>1040</td>
<td>0</td>
<td>1</td>
<td>179</td>
<td>24</td>
</tr>
<tr>
<td>Arria V GT</td>
<td></td>
<td>504K</td>
<td>7Mb</td>
<td>1156</td>
<td>2</td>
<td>160</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>Arria V GZ</td>
<td></td>
<td>504K</td>
<td>7Mb</td>
<td>1156</td>
<td>2</td>
<td>160</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>Arria V SX**</td>
<td>450K</td>
<td>39Mb</td>
<td>1139</td>
<td>1*</td>
<td>160</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arria V ST**</td>
<td>462K</td>
<td>25Mb</td>
<td>1068</td>
<td>2</td>
<td>160</td>
<td>30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Gen3 PCIe  
** with HPS ARM processor
## Hardware DSP: Selecting an FPGA

Stratix High Performance FPGA, maximums

<table>
<thead>
<tr>
<th>Family</th>
<th>Geometry</th>
<th>LE</th>
<th>Mem</th>
<th>18x18</th>
<th>DSP</th>
<th>PCIe</th>
<th>Diff</th>
<th>Xcvr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stratix III L</td>
<td>65nm</td>
<td>338K</td>
<td>20Mb</td>
<td>768</td>
<td>0</td>
<td>0</td>
<td>132</td>
<td>0</td>
</tr>
<tr>
<td>Stratix III E</td>
<td></td>
<td>254K</td>
<td>18Mb</td>
<td>768</td>
<td>0</td>
<td>0</td>
<td>112</td>
<td>0</td>
</tr>
<tr>
<td>Stratix IV E</td>
<td>40nm</td>
<td>813K</td>
<td>33Mb</td>
<td>1288</td>
<td>0</td>
<td>0</td>
<td>132</td>
<td>0</td>
</tr>
<tr>
<td>Stratix IV GX</td>
<td></td>
<td>531K</td>
<td>27Mb</td>
<td>1288</td>
<td>0</td>
<td>4</td>
<td>98</td>
<td>48</td>
</tr>
<tr>
<td>Stratix IV GT</td>
<td></td>
<td>531K</td>
<td>27Mb</td>
<td>1288</td>
<td>0</td>
<td>4</td>
<td>44</td>
<td>48</td>
</tr>
<tr>
<td>Stratix V E</td>
<td>28nm</td>
<td>952K</td>
<td>63Mb</td>
<td>352</td>
<td>0</td>
<td>0</td>
<td>210</td>
<td>0</td>
</tr>
<tr>
<td>Stratix V GX</td>
<td></td>
<td>952K</td>
<td>63Mb</td>
<td>352</td>
<td>4</td>
<td>0</td>
<td>210</td>
<td>66</td>
</tr>
<tr>
<td>Stratix V GS</td>
<td></td>
<td>695K</td>
<td>58Mb</td>
<td>1963</td>
<td>2</td>
<td>0</td>
<td>223</td>
<td>48</td>
</tr>
<tr>
<td>Stratix V GT</td>
<td></td>
<td>952K</td>
<td>63Mb</td>
<td>352</td>
<td>1</td>
<td>0</td>
<td>210</td>
<td>32</td>
</tr>
</tbody>
</table>
Hardware DSP:
Selecting an FPGA Data Path Requirements
• Data path is important once we enter the digital domain
  • Number of traces between ADC and FPGA
  • I/O pin requirements (at FPGA boundaries)
  • Bus width (internal to FPGA)
  • FPGA resources used (for multipliers, adders, accumulators, etc)
  • Speed and type of PC interface
• Speed vs Width can be traded off (sometimes)
• There are limits on speed, even inside the FPGA
Software DSP Considerations

- Minimum computer requirements
- SDR-to-PC communications bandwidth
- Display and demodulation bandwidth
- PC graphics requirements
Software DSP: Computer Requirements

- Requirements vary widely – no simple answer
- Typical minimum is dual-core >2GHz CPU
- Memory is cheap, use lots of it! >4GB
- Gigabit Ethernet preferred over 10M/100M
- USB 2.0 or USB 3.0
- Whatever interface matches your radio!
  - Firewire, eSATA, PCIe?
Software DSP: SDR-to-PC Interface

- Selecting a non-limiting interface
- Account for conversion gain
  - 16-bit samples become 24-bit after decimation by 256
- Account for communications overhead
  - Bit encoding on Ethernet makes GbE 100MByte/s
- Account for other overhead
  - Packet overhead: headers, checksums
  - Channel overhead: collisions, holdoff

Always leave some margin!
Software DSP:
SDR-to-PC: Example Bandwidth Calculation

UDP/SDR-HF1 Receiver Data Flow
Software DSP: PC DSP Considerations

- DSP is very compute intensive
  - May include FFT, inverse FFT, convolution, etc
  - PC CPUs are general purpose processors, not DSPs
- Don’t expect to process 200MHz of bandwidth
  - 80MByte/s possible (40MHz of 8-bit samples) with high-end PC
- CPU horsepower follows Moore’s Law
- So does FPGA density and capability

  - SDR performance increases over time!
Software DSP: PC Graphics Requirements

- Dynamic graphical displays demand resources
  - Panadaptor (spectrum display)
  - Waterfall (spectral waterfall display)
- Modern GPU (Graphical Processing Unit) is really a DSP engine
  - Use OpenGL to program DSP functions using GPU
  - Example: cuSDR for the openHPSDR project
- GPU is often more powerful DSP than CPU
  - NVIDIA GK110 GPU on 28nm, 7.1 billion transistors
  - Intel 8-core Itanium Poulson on 32nm, 3.1 billion transistors
15 Minute Break
BeMicro SDK Evaluation Board
- Features / Layout

- MicroSD card on back
- Micron Mobile DDR 512 Mbit
- LTC Power Supplies
- 10/100 Ethernet PHY
- 8 User LEDs
- CPU Reset
- Temperature Sensor
- USB-Blaster
- Altera 60nm Cyclone IV E FPGA with 22k LEs
- DIP Switch
BeRadio Hardware

Antenna/Filter  Anti-Alias Filter  ADC Driver  ADC

BeRadio Add-On Board

BeMicro

Ant Filter 0dB  Anti-Alias 2 MHz  30dB  ADC 10 MSPS

Audio DAC 50 kSPS  Reconstruction Filter  0dB

NIOs

CIC LPF 2/3  \sqrt{F + Q^2}  FIR LPF 5 kHz

PHONES

BeMicro Add-On Board

Audio DAC  Reconstruction Filter  Headphone Driver

Analog Path  Digital Path  Analog Block  Digital Block

0°  90°  180°  270°
BeRadio Hardware: Antenna/Filter

- Resonant circuit with wire antenna pickup
  - Four digitally-selectable frequency ranges
    - 660kHz, 781kHz, 1000kHz, 1720kHz
  - Selection based on NCO frequency
  - Wire extension for experimentation or external input
- Emitter-follower buffer amplifier
  - Reduces antenna impedance
  - Low Q provides mild “peaking”
BeRadio Hardware: Antenna/Filter

EXT WIRE ANT

SRF=5.8MHz
TYCO 3615A221K

ANT1 Wire Antenna

L3 220uH

RF VDD

C10 39pF 50V
C11 75pF 50V
C12 150pF 50V
C9 100nF 16V
C8 100nF 16V
C7 1nf 50V
C47 1nf 50V

R3 49K9 1%
R1 10K 1%
R2 49K9 1%
R4 1K 1%
R5 10K 1%
R6 10K 1%

U2A FSA266K8X
U2B FSA266K8X

C7 1nF 50V
C13 100nF 16V

GND

VCC

C14 100nF 16V
C16 100nF 16V

A

MMBT6429

A

A
BeRadio Hardware: Anti-Alias Filter

- Nyquist frequency is 5MHz
- Many signals in 2nd Nyquist zone (5-10MHz)
  - 31m, 41m, 49m, 60m shortwave broadcast
  - 40m amateur
  - Marine, aeronautical
BeRadio Hardware: Anti-Alias Filter

- 2 MHz 5-pole filter provides some extra “room”
- 30dB/octave yields nearly 40dB attenuation at 5MHz
BeRadio Hardware: ADC Driver

- LTC6409 ADC driver provides 30dB gain
  - Inefficient antenna benefits from high gain
  - High GBW product, usable up to 100MHz
  - Very low noise, $1.1 \text{nV/} \sqrt{\text{Hz}}$
  - Input common mode range includes ground
- Drives ADC through source termination resistors / input capacitor
- Use $V_{\text{cm}}$ to center signal in ADC input range
BeRadio Hardware: ADC Driver

GAIN = 30dB

[Diagram with components and connections, including resistors R10, R13, R7, R8, capacitors C27, C28, C29, C30, C24, C15, C17, and the LTC6409CUDB IC.]
BeRadio Hardware: ADC LTC2225

• From RF to digital
  • The LTC2225 is a 12-bit ADC @ 10MSPS
• Note separate filtering on Vdd and Ovdd
• Input capacitor helps smooth S&H transients
BeRadio Hardware:
ADC LTC2225 : 12-Bit @ 10Mps

ADC LTC2225CUH

3P3V

C40
100nF
16V

C39
1nF
50V

C38
100nF
16V

C37
1nF
50V

C36
100nF
16V

C35
1nF
50V

C34
1uF
16V

C33
100nF
16V

C32
2.2uF
6.3V

C31
1uF
16V

C26
2.2uF
6.3V

L8
FB_MMZ2012R601A

L7
FB_MMZ2012R601A

ADC_OVDD

ADC_MODE

ADC_CS\_STABIL\_EN\_L
BeRadio Hardware: DAC

- From digital to audio
  - DAC is 12-bit @ 50KSPS: **LTC2641-12**
- Use SPI to send low data rate audio
- 3kHz LP reconstruction filter follows DAC
- LT6200 drives headphones
BeRadio Hardware: DAC
BeRadio Hardware: Reconstruction Filter

- 3kHz LP reconstruction filter follows ADC
- Down 37db @ 45kHz (50KHz sample rate – 5KHz BW)
BeRadio Hardware: Headphone Driver

- LT6200 drives headphones
- Capacitive coupling at output
BeRadio Firmware

Numerically Controlled Oscillator
USB interface
NIOS CPU

Filter/Decimator
AM Demodulator
FIR LPF
Multipliers

BeRadio Firmware
BeRadio Nonboard Board
BeRadio Nonboard Board
Antenna Filter 0 dB
Anti-Alias 2 MHz 20 dB
ADC 10 MSPS

Anti-Alias 2 MHz 20 dB
ADC 10 MSPS

Audio DAC 50 KSPS
Reconstruct 5 kHz 0 dB

NIOS

ANALOG PATH
ANALOG BLOCK
DIGITAL PATH
DIGITAL BLOCK

 Altera EP4CE22 FPGA

CIC LPF 200
FIR LPF 5 kHz

PHONES
BeRadio Firmware: NCO

- Generates two digitized streams of data values
- One stream represents sampled sine, other cosine
- Uses table lookup for sine and cosine values
  - For maximum performance use wide path with filtering
- Sample rate is matched to the input data rate

What do we use these two data streams for?
BeRadio Firmware: Multipliers

- Sample data * cosine = I (in phase or REAL) component
- Sample data * sine = Q (quadrature or IMAGINARY) component
- This implements a complex mixer
- Tune the local oscillator to center the signal of interest around 0 Hz

The I and Q data represent the down-converted input signal in complex form
BeRadio Firmware: Cascaded Integrator Comb Filter

- CIC filter acts as Low Pass Filter (LPF)
- Simultaneously filters and reduces sample rate (decimation)
- More stages
  - Steeper filter roll-off (steeper slope)
  - More FPGA resources
  - Independent of decimation ratio
- Higher decimation ratio
  - Lower output sample rate
  - Smaller Bandwidth
    - (lower frequency cutoff)
BeRadio Firmware: AM Demodulator

- I and Q: rectangular representation
- Magnitude and phase: polar representation
  - How do we convert from rectangular to polar?
- Root of the sum of the squares is the magnitude
- Magnitude is the demodulated output
BeRadio Firmware: FIR LPF

- Eliminates out-of-band (above 5KHz) noise
- More effective filter than CIC
- FIR requires more resources than CIC
- Greatly improves quality of audio output
BeRadio Firmware: USB Interface

• USB port is not used to transfer I/Q data
• USB port is used to load FPGA and serial PROM
• USB supplies +5V power
• Optional low-bandwidth serial control channel
  • Set NCO frequency
  • Scale audio output
  • Select antenna input filter center frequency
BeRadio Firmware: Altera NIOS II CPU

A general-purpose RISC processor core

- Full 32-bit instruction set
- 32 general-purpose registers
- 32 interrupt sources
- External interrupt controller interface
- Single-instruction $32 \times 32$ multiply and divide producing a 32-bit result
- Dedicated instructions for computing 64-bit and 128-bit products of multiplication
BeRadio Firmware: NIOSII CPU implementation

- Very simple three-command interface
  - I command to set channel increment
  - C command to set the channel
  - V command to scale the audio volume
- Commands are all two-byte pairs
  - ASCII character for command
  - One-byte binary argument
- BeRadio is slave only
  - BeRadio echoes command back to PC
BeRadio Firmware: Timing Constraints

Timing constraints specified in the Synopsys Design Constraints (SDC)

ADC clocking options:
• on-board 10 MHz osc
• or a clock generated from the FPGA.

Current design uses the on-board oscillator.

Whichever clock source is chosen, a copy of that clock is sent back to the FPGA to be used as a sample clock.
BeRadio Firmware: Timing Constraints -

The design also defines the clock output for the DAC SPI interface. Thus, the SDC file defines six (6) clocks total:

- **clk50**: The original 50 MHz input clock from the BeMicro SDK
- **adc_clk_in**: The 10 MHz output clock generated by a PLL going to the BeRadio (currently not used)
- **adc_clk**: 10 MHz virtual clock defined at the ADC clock input
- **adc_clk_out**: 10 MHz clock input to the FPGA from the BeRadio board
- **dac_sclk**: 5 MHz clock generated for SPI interface, defined at the output of the FPGA register
- **P26**: 5 MHz SPI interface clock at the FPGA output port
The SDRstick™ Family

- BeRadio
  - 100kHz – 4MHz receiver
- UDPSDR-HF1
  - 100kHz – 30MHz receiver
- UDPSDR-HF2
  - 100KHz – 55MHz receiver
- UDPSDR-TX2
  - 100kHz – 55MHz 500mW transmitter
  - Companion to HF2
HF1 changes from BeRadio

- Sampling rate increased to 80MSPS from 10MSPS
- Sampling data width increased to 14 bits (LTC2249)
- SMA jack for external antenna replaces MW antenna
- LNA gain reduced to 17.5dB
- 30MHz front-end LPF
- Anti-Aliasing filter corner frequency raised to 30MHz
SDRstick™ HF1
SDRstick™ HF2

- HF2 changes from HF1
  - Sampling rate increased to 122.88MSPS
  - Sampling data width increased to 16 bits (LTC2208)
  - LNA gain raised to 20dB (LTC6400-20)
  - 31dB step attenuator added
  - External GPSDO synchronization input
  - External LVDS sampling clock input
  - Extremely low phase-noise oscillator, -152dBc/Hz
  - Customization available for specific applications
SDRstick™ TX2

- TX2 companion transmitter for HF2 receiver
  - Plugs directly into HF2
  - DUC with 14-bit DAC @ 122.88MSPS
  - 500mW wideband power amplifier
  - On-board T/R switching
  - On-board SMPS supply powers BeMicro/HF2/TX2
  - Power requirements 13.8VDC@500mA
References & Additional Reading

• Digital Signal Processing 101: Everything you need to know to get started
  • Michael Parker / Altera corporation

• Understanding Digital Signal Processing
  • Richard G. Lyons

• Understanding CIC Compensation Filters
  • Altera: Application Note 445

• SiGe Differential Amplifier Drives High Speed ADCs at Hundreds of MHz
  • Linear Technology: Kris Lokere and Adam Shou
  • LTMag-V17N02-01-LTC6400-Lokere.pdf
20 Minute Break
BeRadio Lab
Resources

Arrow Electronics
– http://arrow.com

Altera Corporation
– http://altera.com

Linear Technology Corporation
– http://linear.com

Zephyr Engineering, Inc
– Standard SDR products and custom SDR consulting
– http://www.zpci.com/